Article:
Cai, J., Liu, W., Zong, R. et al. (2018) Sparse array extension for non-circular signals with subspace and compressive sensing based DOA estimation methods. Signal Processing, 145. pp. 59-67. ISSN: 0165-1684

https://doi.org/10.1016/j.sigpro.2017.11.012

Sparse Array Extension for Non-Circular Signals with Subspace and Compressive Sensing Based DOA Estimation Methods

Jingjing Cai[a,*], Wei Liu[b,**], Ru Zong[a], Bin Wu[a]

[a] *Department of Electronic and Engineering, Xidian University, Xi'an, 710071, China*
[b] *Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, S1 4ET, United Kingdom*

Abstract

The virtual array generation process based on typical sparse arrays is studied for a mixture of circular and non-circular impinging signals. It consists of two sub-arrays: one is the traditional difference co-array and the other one is the new sum co-array. The number of consecutive virtual array sensors is analysed for the nested array case, but it is difficult to give a closed-form result for a general sparse array. Based on the extended covariance matrix of the physical array, two classes of direction of arrival (DOA) estimation algorithms are then developed, with one based on the subspace method and one based on sparse representation or the compressive sensing (CS) concept. Both the consecutive and non-consecutive parts of the virtual array can be exploited by the CS-based method, while only the consecutive part can be exploited by the subspace-based one. As a result, the CS-based solution can have a better performance than the subspace-based one, though at the cost of significantly increased computational complexity. The two classes of algorithms can also deal with the special case when all the signals are noncircular. Simulation results are provided to verify the performance of the proposed algorithms.

Keywords: Non-circular signal, sparse array, direction of arrival estimation, subspace, compressive sensing.

[*] Corresponding author: Tel.: +86-29-88204759; fax: +86-29-88204759.
[**] Corresponding author: Tel.: +44-114-2225813; fax: +44-114-2225834.
Email addresses: jjcai@mail.xidian.edu.cn (Jingjing Cai), w.liu@sheffield.ac.uk (Wei Liu), zongru@xidian.edu.cn (Ru Zong), bwu@xidian.edu.cn (Bin Wu)

1. Introduction

Recently, the sparse array concept combined with co-array equivalence has attracted significant interest in the community [1]. There are two representative sparse array examples: co-prime arrays [2, 3, 4] and nested arrays [5, 6]. Sparse arrays can form a larger aperture given the same number of antennas and more importantly provide much more degrees of freedom (DOFs) for direction of arrival (DOA) estimation than traditional uniform arrays [7, 8, 3, 9, 10, 11, 12]. However, to our best knowledge, the DOA estimation problem for such sparse arrays has not been properly studied yet to exploit the possible non-circularity of the impinging signals.

On the other hand, DOA estimation based on noncircular signals for traditional uniform array structures has been studied widely. Practical examples for noncircular signals include those generated by modulation schemes such as binary phase shift keying (BPSK), minimum shift keying (MSK), Gaussian MSK (GMSK), pulse amplitude modulation (PAM), and unbalance quadrature PSK (UQPSK), etc. Although the DOA estimation methods for noncircular signals have much higher computational complexity, they can achieve a better performance in terms of resolution and robustness against circular noise [13]. For a uniform linear array (ULA) with noncircular signals, assuming the total number of physical sensors is M, then it can be deduced that the virtual steering vector index range obtained by vectorizing both the covariance matrix and the pseudo covariance matrix is $C_u = 2M - 1$ and the number of DOAs that can be resolved is $D_u = C_u - 1$. However, this result is not applicable to the sparse array case and the virtual array aperture generated for non-circular signals will be studied in detail in this paper.

So far the DOA estimation algorithms proposed for noncircular signals based on uniformly spaced arrays can be divided into two categories: the subspace-based [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25] and the compressive sensing (CS)-based [26, 27, 13, 28]. As we will see later, although they can be applied to the sparse array case directly, they can only exploit part of the increased DOFs provided by the system and will not achieve the best possible estimation performance. Moreover, some of these algorithms can only deal with noncircular signals and cannot be applied to the general case with a mixture of circular and noncircular signals; the latter case can occur when some of the sources are emitting noncircular signals, such as BPSK signals, while some others are emitting circular ones, such as quadrature phase shift keying (QPSK) or quadrature amplitude modulation (QAM) signals. In this work, we will propose two classes of sparse

array DOA estimation algorithms for a mixture of circular and noncircular signals: one is subspace-based and one is CS-based. Both classes of algorithms can also deal with the special case when all the impinging signals are noncircular.

For the subspace-based methods, traditionally an extended covariance matrix is first constructed, which is actually the covariance matrix of the extended signal vector obtained by stacking the original array signal vector and its conjugate together [14, 16, 20, 15, 17]. It can be divided into four sub-matrices and the first two sub-matrices are used for the following DOA estimation process and they provide the same number of DOFs for a ULA in the general case; we call this case equally divided. However, for sparse arrays, these two sub-matrices will generate different number of DOFs and we call this case unequally divided, where two much larger sub-matrices can be constructed, but with different dimensions and they help build a much larger extended covariance matrix, as will be shown later. One effective low-complexity algorithm based on uniformly spaced arrays for noncircular sources is the polynomial rooting method [14]. Applying this algorithm to sparse arrays leads to an algorithm called unequal length (UL) algorithm. To increase the number of exploited DOFs further, an unequal length plus (ULP) algorithm is then proposed in this paper.

For the CS-based methods, the direct covariance matrix (DCM) based algorithm studied in [26] can be used for sparse arrays, but it turns out that the maximum number of sources it can identify is limited by the DOFs of the difference co-array. To overcome this limitation, we propose an extended covariance matrix (ECM) based algorithm with sparse representation, which can identify a much larger number of sources. Both the proposed ULP and ECM algorithms can make effective use of the DOFs provided by the sum co-array as well as those of the difference co-array. However, the number of DOFs exploited by the ECM algorithm is a little lager than that of the ULP algorithm, because the sparse representation based algorithm can utilize not only the consecutive but also the nonconsecutive virtual sensors, while the subspace-based algorithm can only use the consecutive ones.

This paper is organized as follows. The data model for noncircular signals based on a general sparse array is presented in Sec. 2, where the consecutive virtual sensors generated by the difference co-array and the sum co-array are analyzed. The subspace-based DOA estimation algorithms are proposed in Sec. 3, while the CS-based ones are introduced in Sec. 4. Simulation results are provided in Sec. 5, with conclusions drawn in Sec. 6.

2. Data Model

Circularity is an important property of random variables [14, 29, 30], and a zero-mean complex random variable x is said to be circular at the second order, if its elliptic covariance is zero, i.e. $E[xx] = 0$, where $E(\cdot)$ is the mathematical expectation; otherwise, it is noncircular. This property will be used in constructing the pseudo covariance matrix of the received array signals in the following.

Now consider K far-field stationary and uncorrelated narrowband signals impinging on the M-sensor sparse array from DOA angles θ_k, $k = 1, 2, \ldots, K$ and $\theta_k \in [-90°, 90°]$, which are corrupted by additive circular white Caussian noise. The power of these signals are $\eta_1, \ldots, \eta_K$, and among them K_n signals are noncircular, while the remaining K_c of them are circular, with $K_c + K_n = K$. Note that for our proposed methods, we do not need to know the value of K_c or K_n. Define the unit inter-element spacing as d, which is equal to half wavelength $\lambda/2$, and positions of the whole set of array sensors can be expressed as

$$P = \{p_1, p_2, \ldots, p_m, \ldots, p_M\} \cdot d \tag{1}$$

Suppose the signals are uncorrelated with the noise, which is zero-mean with covariance matrix $\sigma^2\mathbf{I}_M$, where $\mathbf{I}_M$ is the $M \times M$ identity matrix. The received array signal vector $\mathbf{x}(t)$ at time index t can be expressed as:

$$\mathbf{x}(t) = \sum_{k=1}^{K} \mathbf{a}(\theta_k)s_k(t) + \mathbf{n}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) \;\; t = 1, 2, \ldots, N \tag{2}$$

where $s_k(t)$ is the kth zero-mean source signal, N is the number of snapshots, $\mathbf{a}(\theta_k)$ is the steering vector corresponding to the kth signal, given by

$$\mathbf{a}(\theta_k) = [e^{-j(2\pi p_1 dsin\theta_k/\lambda)}, \ldots, e^{-j(2\pi p_M dsin\theta_k/\lambda)}]^T \tag{3}$$

with $(\cdot)^T$ denoting the transpose operation. The steering matrix $\mathbf{A}$ and the signal vector $\mathbf{s}(t)$ are formed by

$$\begin{aligned} \mathbf{A} &= [\mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_K)] \\ \mathbf{s}(t) &= [s_1(t), \ldots, s_K(t)]^T \end{aligned} \tag{4}$$

The covariance matrix $\mathbf{R}_{xx}$ of the received array signal is given by

$$\begin{aligned} \mathbf{R}_{xx} &= E\{\mathbf{x}(t)\mathbf{x}^H(t)\} = \sum_{k=1}^{K} \eta_k \mathbf{a}(\theta_k)\mathbf{a}^H(\theta_k) + \delta^2\mathbf{I}_M \\ &= \mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^H + \delta^2\mathbf{I}_M \end{aligned} \tag{5}$$

where $(\cdot)^H$ denotes the Hermitian transpose operation, and the source covariance matrix $\mathbf{\Gamma}$ is diagonal with

$$\mathbf{\Gamma} = E[s(t)s^H(t)] = \begin{bmatrix} \eta_1, 0, \cdots, 0 \\ 0, \eta_2, \cdots, 0 \\ \vdots, \vdots, \ddots, \vdots \\ 0,\ 0, \cdots, \eta_K \end{bmatrix} \tag{6}$$

The pseudo covariance matrix is defined as

$$\begin{aligned} \mathbf{R}_{xx^*} &= E\{\mathbf{x}(t)\mathbf{x}^T(t)\} = \sum_{k=1}^{K} \rho_k e^{j\psi_k} \eta_k \mathbf{a}(\theta_k)\mathbf{a}^T(\theta_k) \\ &= (\mathbf{A}^*\mathbf{V}^*)\mathbf{\Gamma}(\mathbf{A}^*\mathbf{V}^*)^H \end{aligned} \tag{7}$$

where $(\cdot)^*$ denotes the conjugate operation, and $\mathbf{V}$ is a $K \times K$ diagonal matrix. Its first K_n diagonal elements are $\rho_k e^{j\psi_k}$ and last $(K-K_n)$ diagonal elements are 0, where ρ_k and ψ_k are the noncircularity rate and phase of the kth noncircular signal. For circular signals, $\rho_k = 0$, and it is $0 < \rho_k \leq 1$ for noncircular signals. For strictly noncircular signals, such as BPSK signals, we have $\rho_k = 1$. If all the signals are circular, only $\mathbf{R}_{xx}$ is non-zero valued. If the signals are a mixture of circular and noncircular signals, as studied in this work, both $\mathbf{R}_{xx}$ and $\mathbf{R}_{xx^*}$ are non-zero valued, and we can define the extended covariance matrix as

$$\mathbf{R} = E\{\mathbf{x}_{et}(t)\mathbf{x}_{et}^H(t)\} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^*\mathbf{V}^* \end{bmatrix} \mathbf{\Gamma} \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^*\mathbf{V}^* \end{bmatrix}^H \tag{8}$$

based on the extended signal vector $\mathbf{x}_{et}$

$$\mathbf{x}_{et} = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}^*(t) \end{bmatrix} \tag{9}$$

Two separate virtual arrays can be generated by vectorizing $\mathbf{R}_{xx}$ and $\mathbf{R}_{xx^*}$, and the resultant difference co-array vector $\mathbf{r}_{xx}$ and sum co-array vector $\mathbf{r}_{xx^*}$ are respectively given by

$$\begin{aligned} \mathbf{r}_{xx} &= \sum_{k=1}^{K} \eta_k \mathbf{a}_{xx}(\theta_k) + [\delta^2, \ldots, \delta^2]^T = \mathbf{A}_{xx}\mathbf{v}_{xx} + [\delta^2, \ldots, \delta^2]^T \\ \mathbf{r}_{xx^*} &= \sum_{k=1}^{K_n} \rho_k e^{j\psi_k} \eta_k \mathbf{a}_{xx^*}(\theta_k) = \mathbf{A}_{xx^*}\mathbf{v}_{xx^*} \end{aligned} \tag{10}$$

with the corresponding steering vectors $\mathbf{a}_{xx}(\theta_k)$ and $\mathbf{a}_{xx^*}(\theta_k)$ given by ($\otimes$ denotes the Kronecker product)

$$\begin{aligned}\mathbf{a}_{xx}(\theta_k) &= \mathbf{a}^*(\theta_k) \otimes \mathbf{a}(\theta_k) \\ \mathbf{a}_{xx^*}(\theta_k) &= \mathbf{a}(\theta_k) \otimes \mathbf{a}(\theta_k)\end{aligned} \tag{11}$$

The corresponding steering matrices and the virtual source vectors are given by

$$\begin{aligned}\mathbf{A}_{xx} &= [\mathbf{a}_{xx}(\theta_1), \ldots, \mathbf{a}_{xx}(\theta_K)] \\ \mathbf{A}_{xx^*} &= [\mathbf{a}_{xx^*}(\theta_1), \ldots, \mathbf{a}_{xx^*}(\theta_K)] \\ \mathbf{v}_{xx} &= [\eta_1, \ldots, \eta_K]^T \\ \mathbf{v}_{xx^*} &= [\rho_1 e^{j\phi_1}\eta_1, \ldots, \rho_{K_n} e^{j\phi_{K_n}}\eta_{K_n}, 0, \ldots, 0]^T\end{aligned} \tag{12}$$

Assume p_i, p_j, p_u and p_v are arbitrary values chosen from the set P in (1). Clearly the values of $\mathbf{r}_{xx}$ are related to $(p_i - p_j)$, while those of $\mathbf{r}_{xx^*}$ are related to $(p_u + p_v)$.

Now, we analyze the DOFs provided by both co-arrays. Suppose the consecutive values generated by $(p_i - p_j)$ is C_d, while it is C_s for $(p_u + p_v)$. Define $\breve{C}_d$ and $\breve{C}_s$ as the corresponding total number of different integers including both consecutive and nonconsecutive ones. For the two typical sparse arrays, the co-prime array and the nested array[2, 3, 4, 5, 6], the number of consecutive difference co-array virtual sensors is $C_d = 2M_1M_2 + 2M_2 - 1$ in the nested array case using $M = M_1 + M_2$ sensors, and $C_d = 2M_1M_2 + 2M_1 - 1$ in the co-prime array case using $M = 2M_1 + M_2 - 1$ sensors, where M_1 and M_2 are the associated parameters of the nested array and the co-prime array. The total number of difference co-array virtual sensors of the nested array is $\breve{C}_d = C_d$, but it is difficult to give a general result for the co-prime array.

However, to our best knowledge, for the number of consecutive sum co-array sensors, it has not been addressed yet. Due to complexity of the problem, it is difficult to give an analysis for the co-prime array case, not to mention the more general one, but the consecutive integers generated by $(p_u + p_v)$ can be analyzed for the two-level nested arrays. In such an array, the first $M_1 + 1$ elements of the array are consecutive segment from 1 to $M_1 + 1$, with an inter-element spacing d, and then the next $M_2 - 1$ elements have a spacing of $(M_1 + 1)d$. When these elements are added together by the sum operation, the first $M_1 + 1$ elements are shifted to the positions of the next $M_2 - 1$ elements, making all the consecutive elements and shifted ones consecutive. Since the last element of P is $p_M = (M_1 + 1)M_2$, the number

of consecutive virtual sensors generated by the sum co-array is given by $C_s = M_1M_2 + M_1 + M_2$. The number of total sum co-array sensors is $M_2 - 1$ more than C_s, i.e. $\breve{C}_s = M_1M_2 + M_1 + 2M_2 - 1$. For example, when $M_1 = M_2 = 2$, we have $p_M = 6$, $C_d = 11$, $C_s = 8$, $\breve{C}_d = 11$ and $\breve{C}_s = 9$.

One note is that C_d is normally much lager than C_s, since a sparse array is usually designed by maximizing the number of consecutive elements generated by $(p_i - p_j)$, while the number of consecutive elements generated by $(p_i + p_j)$ is ignored in this process. For example, we have $C_d - C_s = M_1M_2 + M_2 + M_1 - 1 > 0$ for a nested array, and clearly C_d is much greater than C_s.

In practice, for a finite number of snapshots, the covariance and pseudo covariance matrices can be estimated as

$$\begin{aligned} \hat{\mathbf{R}}_{xx} &= (1/N)\sum_{t=1}^{N} \mathbf{x}(t)\mathbf{x}^H(t) \\ \hat{\mathbf{R}}_{xx^*} &= (1/N)\sum_{t=1}^{N} \mathbf{x}(t)\mathbf{x}^T(t), \end{aligned} \tag{13}$$

and the corresponding $\mathbf{r}_{xx}$ and $\mathbf{r}_{xx^*}$ are then approximated by $\hat{\mathbf{r}}_{xx}$ and $\hat{\mathbf{r}}_{xx^*}$.

In the next two sections, based on the model in (10), we develop two different DOA estimation methods, which can all effectively exploit the information carried by both the difference and the sum co-arrays.

3. Subspace Based DOA estimation: the Unequal Length Plus (ULP) Algorithm

Some DOA estimation algorithms based on the subspace method have been proposed for various scenarios with noncircular signals [14, 16, 20, 15, 17, 22, 23, 24, 25], but they are all applied to uniformly spaced arrays, where the number of steering vector indexes in the difference covariance matrix and the pseudo/sum covariance matrix are the same. One algorithm using polynomial rooting proposed in [14] could also be adapted to the sparse array case, and we call the adapted algorithm the unequal length (UL) algorithm, as the dimensions of the two newly constructed sub-matrices are different, which will be seen in the following. The specific processing procedure is introduced below firstly. We also derive the maximum number of signals the array can resolve based on the length of the difference co-array and sum co-array. However, we will see later that this UL algorithm resolves less signals than this maximum number. So we further modify the UL algorithm and

propose a so-called unequal length plus (ULP) algorithm which can fully exploit the potential of the sparse array to resolve the maximum number of signals.

Firstly, we construct two new vectors $\mathbf{r}_d$ and $\mathbf{r}_s$ using the elements of the difference co-array $\mathbf{r}_{xx}$ and the sum co-array $\mathbf{r}_{xx^*}$ in (10), as given below. The repeated elements in $\mathbf{a}_{xx}$ and $\mathbf{a}_{xx^*}$ of (11) are merged, and we then order the elements according to an increasing steering vector index, similar to that of a standard uniform linear array, and at last these consecutive virtual steering vector elements are used to construct the two new steering vectors $\mathbf{a}_d$ and $\mathbf{a}_s$ in (15), respectively. The length of these two new vectors will be exactly C_d and C_s, respectively. Note that all the nonconsecutive elements in $\mathbf{a}_{xx}$ and $\mathbf{a}_{xx^*}$ are excluded from the construction of (15).

$$\begin{aligned}\mathbf{r}_d &= \sum_{k=1}^{K} \eta_k \mathbf{a}_d(\theta_k) + [\delta^2, \ldots, \delta^2]^T = \mathbf{A}_d \mathbf{v}_d + [\delta^2, \ldots, \delta^2]^T \\ \mathbf{r}_s &= \sum_{k=1}^{K} \rho_k e^{j\psi_k} \eta_k \mathbf{a}_s(\theta_k) = \mathbf{A}_s \mathbf{v}_s\end{aligned} \tag{14}$$

with

$$\begin{aligned}&\mathbf{a}_d(\theta_k) = [e^{-j(2\pi \frac{C_d-1}{2} d sin\theta_k/\lambda)}, \ldots, e^{j(2\pi \frac{C_d-1}{2} d sin\theta_k/\lambda)}]^T \\ &\mathbf{a}_s(\theta_k) = [e^{-j(2\pi d \Delta p sin\theta_k/\lambda)}, \ldots, e^{-j(2\pi(\Delta p + C_s - 1) d sin\theta_k/\lambda)}]^T \\ &\mathbf{A}_d = [\mathbf{a}_d(\theta_1), \ldots, \mathbf{a}_d(\theta_K)] \\ &\mathbf{A}_s = [\mathbf{a}_s(\theta_1), \ldots, \mathbf{a}_s(\theta_K)] \\ &\mathbf{v}_d = \mathbf{v}_{xx} \\ &\mathbf{v}_s = \mathbf{v}_{xx^*}\end{aligned} \tag{15}$$

where Δp is the first number in the consecutive sequence of values produced by $(p_i + p_j)$.

The l-th element of $\mathbf{r}_d$ can be expressed as

$$\mathbf{r}_d(l) = \eta_k e^{-j[2\pi(\frac{C_d+1}{2} - l) d sin\theta_k/\lambda]} \quad l = 1, ..., C_d \tag{16}$$

and the lth element of $\mathbf{r}_s$ is given by

$$\mathbf{r}_s(l) = \rho_k e^{j\psi_k} \eta_k e^{-j[2\pi(l + \Delta p - 1) d sin\theta_k/\lambda]} \quad l = 1, ..., C_s \tag{17}$$

In our discussions so far, we have treated the elements in $\mathbf{r}_d$ and $\mathbf{r}_s$ as the received virtual signal for an array with steering vectors given by

$\mathbf{a}_d(\theta_k)$ and $\mathbf{a}_s(\theta_k)$, respectively. On the other hand, since those elements are correlation values of the received signals by the original physical array, we can use them directly to construct a new set of difference covariance matrix and sum/pseudo covariance matrix of an equivalent ULA with dimensions $(C_d+1)/2$ and $(C_s+1)/2$, respectively. In order to ensure the number of sensors of the virtual ULA to be an integer, the value of C_d and C_s should be odd. C_d always meets this requirement, but C_s may not. So we will use (C_s-1) instead of C_s when C_s is even.

Following a similar relationship as the structure of $\mathbf{R}$ in (8), we construct a new extended covariance matrix $\mathbf{R}_u$ using the elements of $\mathbf{r}_d$ and $\mathbf{r}_s$ as follows

$$\mathbf{R}_u = \begin{bmatrix} \mathbf{r}_d(L_1) & \dots & \mathbf{r}_d(2L_1-1) & \mathbf{r}_s(1) & \dots & \mathbf{r}_s(L_2) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{r}_d(1) & \dots & \mathbf{r}_d(L_1) & \mathbf{r}_s(L_1) & \dots & \mathbf{r}_s(L_1+L_2-1) \\ \mathbf{r}_s^*(1) & \dots & \mathbf{r}_s^*(L_1) & \mathbf{r}_d(L_1) & \dots & \mathbf{r}_d(L_1-L_2+1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{r}_s^*(L_2) & \dots & \mathbf{r}_s^*(L_1+L_2-1) & \mathbf{r}_d(L_1+L_2-1) & \dots & \mathbf{r}_d(L_1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2^*\mathbf{V}^* \end{bmatrix} \mathbf{\Gamma} \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2^*\mathbf{V}^* \end{bmatrix}^H \tag{18}$$

where $\mathbf{r}_d(i)$ or $\mathbf{r}_s(i)$ denotes the ith elements of the vector $\mathbf{r}_d$ or $\mathbf{r}_s$, $\mathbf{A}_1$ and $\mathbf{A}_2$ are the steering matrices

$$\begin{aligned} \mathbf{A}_1 &= [\mathbf{a}_1(\theta_1), \dots, \mathbf{a}_1(\theta_K)] \\ \mathbf{A}_2 &= [\mathbf{a}_2(\theta_1), \dots, \mathbf{a}_2(\theta_K)] \\ \mathbf{a}_1(\theta_k) &= [e^{-j(2\pi d sin\theta_k/\lambda)}, \dots, e^{-j(2\pi L_1 d sin\theta_k/\lambda)}]^T \\ \mathbf{a}_2(\theta_k) &= [e^{-j(2\pi d sin\theta_k/\lambda)}, \dots, e^{-j(2\pi L_2 d sin\theta_k/\lambda)}]^T \end{aligned} \tag{19}$$

We can see that the elements used in $\mathbf{r}_d$ for constructing $\mathbf{R}_u$ is $\mathbf{r}_d(i)$, $i = 1, \cdots, 2L_1-1$. Since the maximum number of elements contained in $\mathbf{r}_d$ is C_d, we have $2L_1-1 = C_d$ and then $L_1 = (C_d+1)/2$. In the same way, the range of elements used in $\mathbf{r}_s$ is $\mathbf{r}_s(i)$, $i = 1, \cdots, L_1+L_2-1$, and since the maximum number of elements contained in $\mathbf{r}_s$ is C_s, we have $L_1+L_2-1 = C_s$,

and then $L_2 = C_s + 1 - L_1$. In summary,

$$\begin{aligned} L_1 &= (C_d + 1)/2 \\ L_2 &= C_s + 1 - L_1 \end{aligned} \tag{20}$$

The matrix $\mathbf{R}_u$ can be divided into four blocks accordingly

$$\mathbf{R}_u = \begin{bmatrix} \mathbf{R}_1 \, \mathbf{R}_2 \\ \mathbf{R}_3 \, \mathbf{R}_4 \end{bmatrix} \tag{21}$$

where $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4$ are matrices with dimensions $L_1 \times L_1$, $L_1 \times L_2$, $L_2 \times L_1$, and $L_2 \times L_2$, separately.

Applying eigen-decomposition to $\mathbf{R}_u$, we obtain a K-dimensional signal subspace $\mathbf{U}_s$ and an $(L_1 + L_2 - K)$-dimensional subspace $\mathbf{U}_n$. Similar to [14], DOA estimation can be performed by minimizing the following cost function:

$$\mathbf{J}(\theta, \psi) = \mathbf{b}^H(\theta, \psi)\mathbf{U}_n\mathbf{U}_n^H\mathbf{b}(\theta, \psi) = \mathbf{q}^H\mathbf{M}\mathbf{q} \tag{22}$$

where the extended steering vector $\mathbf{b}$, vector $\mathbf{q}$ and matrix $\mathbf{M}$ are given by

$$\begin{aligned} \mathbf{b}(\theta, \psi) &= \begin{bmatrix} \mathbf{a}_1(\theta) \\ \mathbf{a}_2^*(\theta)\rho e^{j\psi} \end{bmatrix}, \\ \mathbf{q} &= \begin{bmatrix} 1 \\ \rho e^{j\psi} \end{bmatrix}, \\ \mathbf{M} &= \begin{bmatrix} \mathbf{a}_1^H(\theta)\mathbf{U}_{n1}\mathbf{U}_{n1}^H\mathbf{a}_1(\theta) \; \mathbf{a}_1^H(\theta)\mathbf{U}_{n1}\mathbf{U}_{n2}^H\mathbf{a}_2^*(\theta) \\ \mathbf{a}_2^T(\theta)\mathbf{U}_{n2}\mathbf{U}_{n1}^H\mathbf{a}_1(\theta) \; \mathbf{a}_2^T(\theta)\mathbf{U}_{n2}\mathbf{U}_{n2}^H\mathbf{a}_2^*(\theta) \end{bmatrix}, \end{aligned} \tag{23}$$

with ρ and ψ being the possible noncircularity ratio and phase of signals, $\mathbf{U}_{n1}$ being an $L_1 \times (L_1 + L_2 - K)$ matrix and $\mathbf{U}_{n2}$ an $L_2 \times (L_1 + L_2 - K)$ matrix. These two matrices are obtained by unequally splitting $\mathbf{U}_n$ as follows,

$$\mathbf{U}_n = \begin{bmatrix} \mathbf{U}_{n1} \\ \mathbf{U}_{n2} \end{bmatrix} \tag{24}$$

It can be proved that the true values of θ are obtained when the determinant of the matrix $\mathbf{M}$ becomes zero [14]. In this way, the noncircularity ratio ρ and phase ψ have been separated from the signal direction θ, so that DOA estimation can be performed independent of the noncircularity coefficients. Now we can use the polynomial rooting method to find the DOA information [14].

Firstly, define $z = e^{-j(2\pi d sin\theta/\lambda)}$. We have

$$\begin{aligned} \mathbf{a}_1(z) &= [z, \ldots, z^{L_1}]^T \\ \mathbf{a}_2(z) &= [z, \ldots, z^{L_2}]^T \end{aligned} \tag{25}$$

and the matrix $\mathbf{M}$ becomes a function of z. We estimate the DOAs by finding the values of z at which the determinant of matrix $\mathbf{M}$ equals zero. As a result, the DOA estimation problem is transformed into a polynomial rooting problem. The polynomial of z can be written as $m_1 m_4 - m_2 m_3 = 0$, where

$$\begin{aligned} m_1 &= \mathbf{a}_1^T(1/z)\mathbf{U}_{n1}\mathbf{U}_{n1}^H\mathbf{a}_1(z), \\ m_2 &= \mathbf{a}_1^T(1/z)\mathbf{U}_{n1}\mathbf{U}_{n2}^H\mathbf{a}_2(1/z), \\ m_3 &= \mathbf{a}_2^T(z)\mathbf{U}_{n2}\mathbf{U}_{n1}^H\mathbf{a}_1(z), \\ m_4 &= \mathbf{a}_2^T(z)\mathbf{U}_{n2}\mathbf{U}_{n2}^H\mathbf{a}_2(1/z). \end{aligned} \tag{26}$$

After obtaining the roots z_n of this polynomial, we can then calculate the DOA estimates using

$$\theta_k = arcsin(\lambda/(2\pi d)arg(z_n)). \tag{27}$$

The order of the polynomial is from $(2 - L_1 - L_2)$ to $(L_1 + L_2 - 2)$ with the number of roots being $2L_1 + 2L_2 - 4$. Due to the symmetry property of the polynomial coefficients, the roots appear in reciprocal conjugate pairs. Either one can be used for calculating the DOA result, since they have the same angle in the complex plane [14]. So we obtain $D_{ul} = L_1 + L_2 - 2 = C_s - 1$ possible DOAs using this method, and all of them can be noncircular signals or at most $(C_d - 1)/2$ circular DOAs with a mixture of circular and noncircular signals. Here we can derive the condition $C_s > (C_d + 1)/2$ for this algorithm to work, which should be met by the array setting; otherwise, $\mathbf{R}_u$ can not be constructed. Fortunately this condition can be met by all nested and co-prime arrays.

The noncircularity phase ψ_k can also be estimated. Recalling that the minimum of the quadratic form in (22) is obtained when this form is equal to the smallest eigenvalue of $\mathbf{M}$, ψ_k can be obtained by examining the eigenvector associated with this eigenvalue. The smallest eigenvalue being equal to zero, and the corresponding DOA θ_k being known, we then have the expression for the associated eigenvector as

$$\mathbf{q} = \begin{bmatrix} 1 \\ -\frac{a_2^T(\theta_k)\mathbf{U}_{n2}\mathbf{U}_{n1}^H a_1(\theta_k)}{a_1^H(\theta_k)\mathbf{U}_{n1}\mathbf{U}_{n1}^H a_1(\theta_k)} \end{bmatrix}. \tag{28}$$

Comparing this with (22), we can determine the noncircularity phase corresponding to the DOA θ_k as

$$\psi_k = \pi - arg(a_2^T(\theta_k)\mathbf{U}_{n2}\mathbf{U}_{n1}^H a_1(\theta_k)). \tag{29}$$

In theory, C_d elements could construct a $(C_d+1)/2 \times (C_d+1)/2$ matrix at most, which can resolve $(C_d-1)/2$ signals; C_s elements could construct a $(C_s+1)/2 \times (C_s+1)/2$ matrix at most, which can resolve $(C_s-1)/2$ signals. Adding these two together, the maximum number of signals that can be resolved theoretically can be finally derived as $(C_s + C_d)/2 - 1$. Note that the maximum number of sources that can be resolved by the UL algorithm (C_s-1) is obviously less than the theoretical maximum value $(C_d+C_s)/2-1$. Next, we modify the UL algorithm and propose the ULP DOA estimation algorithm, by which the theoretically maximum number of resolvable signals can be achieved.

First, we construct another extended covariance matrix as follows

$$\tilde{\mathbf{R}}_u = \begin{bmatrix} \mathbf{R}_4^* & \mathbf{R}_3^* \\ \mathbf{R}_2^* & \mathbf{R}_1^* \end{bmatrix} \tag{30}$$

Then, a new cost function $\tilde{\mathbf{J}}$ and matrix $\tilde{\mathbf{M}}$ can be constructed in a similar way as before, where $\tilde{\mathbf{M}}$ contains four polynomials $\tilde{m}_1$,$\tilde{m}_2$,$\tilde{m}_3$ and $\tilde{m}_4$. The degrees of these four polynomials are from $(-L_2+1)$ to (L_2-1), from $(2-L_1-L_2)$ to 0, from 0 to (L_1+L_2-2) and from $(-L_1+1)$ to (L_1-1), separately. The true values of θ can also be obtained by setting the determinant of the matrix $\tilde{\mathbf{M}}$ equal to zero, i.e. $\tilde{m}_1\tilde{m}_4 - \tilde{m}_2\tilde{m}_3 = 0$.

It's clear that the order of the polynomials $\tilde{m}_1$, $\tilde{m}_4$, $\tilde{m}_2$, and $\tilde{m}_3$ is the same as that of the m_4, m_1, m_3, and m_2, separately. That means the maximum number of resolvable signals by solving the cost function $\mathbf{J}$ or $\tilde{\mathbf{J}}$ is the same. However, we can increase this number by adding the two cost functions together as follows

$$\mathbf{J}(\theta,\psi) + \tilde{\mathbf{J}}(\theta,\psi) = \mathbf{q}^H(\mathbf{M}+\tilde{\mathbf{M}})\mathbf{q} \tag{31}$$

Similarly, the DOA estimation result can be obtained by setting the determinant of the matrix $\mathbf{M}+\tilde{\mathbf{M}}$ to zero, i.e.

$$(m_1+\tilde{m}_1)(m_4+\tilde{m}_4) - (m_2+\tilde{m}_2)(m_3+\tilde{m}_3) = 0 \tag{32}$$

We call this new solution the ULP algorithm. The degrees of the polynomials $(m_1+\tilde{m}_1)$ and $(m_4+\tilde{m}_4)$ are the same, which are all from $min[-L_1+$

$1, -L_2+1]$ to $max[L_1-1, L_2-1]$, where $min[\cdot]$ and $max[\cdot]$ are the operations to obtain the minimum and the maximum values, respectively. Because $L_1 - L_2 = C_d - C_s > 0$, the degrees can be written as from $(-L_1+1)$ to (L_1-1). The degrees of the polynomials $(m_2+\tilde{m}_2)$ and $(m_3+\tilde{m}_3)$ are from $(2-L_1-L_2)$ to 0 and from 0 to (L_1+L_2-2), separately. Then, the degree of $(m_1+\tilde{m}_1)(m_4+\tilde{m}_4)$ is from $2(-L_1+1)$ to $2(L_1-1)$ with a length $4(L_1-1)$, while the degree of $(m_2+\tilde{m}_2)(m_3+\tilde{m}_3)$ is from $(2-L_1-L_2)$ to (L_1+L_2-2) with a length $2(L_1+L_2-2)$. Hence the degree of the whole polynomial is half that of $(m_1+\tilde{m}_1)(m_4+\tilde{m}_4)$ and half that of $(m_2+\tilde{m}_2)(m_3+\tilde{m}_3)$ added together, which is $3L_1+L_2-4 = C_d+C_s-2$. That means there are (C_d+C_s-2) roots in total. Since they appear in reciprocal pairs, there are $D_{ulp} = (C_d+C_s)/2-1$ possible DOAs. Same as the UL algorithm, all of these DOAs can be from noncircular sources, or at most $(C_d-1)/2$ of them are circular signals with the rest being noncircular.

After obtaining the DOA θ_k, the circularity phase ψ_k can also be calculated in the same way as the UL algorithm. The vector $\mathbf{q}$ can be constructed as

$$\mathbf{q} = \begin{bmatrix} 1 \\ -\frac{a_2^T(\theta_k)\mathbf{U}_{n2}\mathbf{U}_{n1}^H a_1(\theta_k)+a_1^T(\theta_k)\mathbf{U}_{n1}\mathbf{U}_{n2}^H a_2(\theta_k)}{a_1^H(\theta)\mathbf{U}_{n1}\mathbf{U}_{n1}^H a_1(\theta_k)+a_2^H(\theta_k)\mathbf{U}_{n2}\mathbf{U}_{n2}^H a_2(\theta_k)} \end{bmatrix}. \tag{33}$$

Then, ψ_k is given by

$$\psi_k = \pi - arg(a_2^T(\theta_k)\mathbf{U}_{n2}\mathbf{U}_{n1}^H a_1(\theta_k) + a_1^T(\theta_k)\mathbf{U}_{n1}\mathbf{U}_{n2}^H a_2(\theta_k)). \tag{34}$$

4. Compressive Sensing Based DOA Estimation: The Extended Covariance Matrix (ECM) Algorithm

In this section, we propose a CS-based algorithm which, different from the subspace-based one, does not require consecutiveness of the virtual sensors, so that we can utilize not only the consecutive elements but also the nonconsecutive ones generated by the co-arrays.

Similar to (14), we can use all the consecutive and nonconsecutive elements in $\mathbf{r}_{xx}$ and $\mathbf{r}_{xx^*}$ to construct two vectors $\breve{\mathbf{r}}_d$ and $\breve{\mathbf{r}}_s$ by merging all the repeated elements

$$\begin{aligned} \breve{\mathbf{r}}_d &= \breve{\mathbf{A}}_d\mathbf{v}_d + [\delta^2, \ldots, \delta^2]^T \\ \breve{\mathbf{r}}_s &= \breve{\mathbf{A}}_s\mathbf{v}_s \end{aligned} \tag{35}$$

where

$$\begin{aligned}
&\breve{\mathbf{a}}_d(\theta_k) = [e^{-j(2\pi(p_1-p_M)dsin\theta_k/\lambda)}, \ldots, e^{-j(2\pi\frac{C_d-1}{2}dsin\theta_k/\lambda)}, \\
&\ldots, e^{j(2\pi\frac{C_d-1}{2}dsin\theta_k/\lambda)}, \ldots, e^{j(2\pi(p_M-p_1)dsin\theta_k/\lambda)}]^T \\
&\breve{\mathbf{a}}_s(\theta_k) = [e^{-j(2\pi d2p_1 sin\theta_k/\lambda)}, \ldots, e^{-j(2\pi d\Delta psin\theta_k/\lambda)}, \\
&\ldots, e^{-j(2\pi(\Delta p+C_s-1)dsin\theta_k/\lambda)}, \ldots, e^{-j(2\pi 2p_M dsin\theta_k/\lambda)}]^T \\
&\breve{\mathbf{A}}_d = [\breve{\mathbf{a}}_d(\theta_1), \ldots, \breve{\mathbf{a}}_d(\theta_K)] \\
&\breve{\mathbf{A}}_s = [\breve{\mathbf{a}}_s(\theta_1), \ldots, \breve{\mathbf{a}}_s(\theta_K)]
\end{aligned} \tag{36}$$

Note that elements from $e^{-j(2\pi\frac{C_d-1}{2}dsin\theta_k/\lambda)}$ to $e^{j(2\pi\frac{C_d-1}{2}dsin\theta_k/\lambda)}$ are consecutive in $\breve{\mathbf{a}}_d$, while other elements are nonconsecutive ones. Similarly, the elements from $e^{-j(2\pi d\Delta psin\theta_k/\lambda)}$ to $e^{-j(2\pi(\Delta p+C_s-1)dsin\theta_k/\lambda)}$ are consecutive ones in $\breve{\mathbf{a}}_s$, and the others are nonconsecutive ones.

The elements of $\breve{\mathbf{r}}_d$ have the same expressions as those of $\mathbf{r}_d$ in (16), while the elements of $\breve{\mathbf{r}}_s$ have the same expressions as those of $\mathbf{r}_s$ in (17), and the only difference is their range. With the same reason as in the subspace-based algorithm, $\breve{C}_s$ should be odd, so we also use $(\breve{C}_s - 1)$ instead of $\breve{C}_s$ when it is even, and the maximum number of DOAs to be resolved can be deduced as $(\breve{C}_d + \breve{C}_s)/2 - 1$, theoretically.

As shown in (35), by the original covariance matrix and pseudo covariance matrix, we can construct two virtual subarrays with the received signal vector given by $\breve{\mathbf{r}}_d$ and $\breve{\mathbf{r}}_s$, steering matrix given by $\breve{\mathbf{A}}_d$ and $\breve{\mathbf{A}}_s$, and source signal vector given by $\mathbf{v}_d$ and $\mathbf{v}_s$, respectively. Ignoring the effect of noise, we can rewrite (35) as follows

$$\begin{aligned}
\breve{\mathbf{r}}_d &= \breve{\mathbf{A}}_d(\Theta)\mathbf{v}_d \\
\breve{\mathbf{r}}_s &= \breve{\mathbf{A}}_s(\Theta)\mathbf{v}_s
\end{aligned} \tag{37}$$

where Θ represents the K real DOAs values. Define $\hat{\Theta}$ as the discrete angle set that covers all possible directions of the incident signals, which contains K_g potential incident angles $\hat{\theta}_1, \ldots, \hat{\theta}_{K_g}$, $\breve{\mathbf{A}}_d(\hat{\Theta})$ and $\breve{\mathbf{A}}_s(\hat{\Theta})$ as the overcomplete dictionaries with structures corresponding to the estimated data model $\hat{\breve{\mathbf{r}}}_d$ and $\hat{\breve{\mathbf{r}}}_s$, $\hat{\mathbf{v}}_d$ and $\hat{\mathbf{v}}_s$ as the spatial distributions of $\mathbf{v}_d$ and $\mathbf{v}_s$, with K_g elements in each of them, which have non-zero values only at locations of the impinging signals.

Then, the CS-cased DOA estimation problem can be formulated as

$$\begin{aligned} &min \quad \|\hat{\mathbf{v}}\|_1 \\ &subject\ to \quad \hat{\mathbf{v}}(i) \geq \|[\hat{\mathbf{v}}_d(i), \hat{\mathbf{v}}_s(i)]\|_2 \\ &where \quad \|\hat{\breve{\mathbf{r}}}_d - \breve{\mathbf{A}}_d(\hat{\Theta})\hat{\mathbf{v}}_d\|_2 \leq \beta_d \\ &\qquad\quad \|\hat{\breve{\mathbf{r}}}_s - \breve{\mathbf{A}}_s(\hat{\Theta})\hat{\mathbf{v}}_s\|_2 \leq \beta_s \end{aligned} \tag{38}$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ are the l_1-norm and l_2-norm, separately, $\hat{\mathbf{v}}_d(i)$, $\hat{\mathbf{v}}_s(i)$ and $\hat{\mathbf{v}}(i)$ are the ith elements of $\hat{\mathbf{v}}_d$, $\hat{\mathbf{v}}_s$, and $\hat{\mathbf{v}}$, corresponding to direction $\hat{\theta}_i$ in the set $\hat{\Theta}$, β_d and β_s are the reconstruction error bounds of $\hat{\breve{\mathbf{r}}}_d$ and $\hat{\breve{\mathbf{r}}}_s$ respectively. A rough estimate for the values of β_d and β_s can be obtained following the method in [26].

The noncircularity phase can also be estimated. Define $\hat{\mathbf{v}}_{sk}$ as the kth peak value of $\hat{\mathbf{v}}_s$. Then, ψ_k can be estimated as

$$\psi_k = arg(\hat{\mathbf{v}}_{sk}) \tag{39}$$

Note that the angles $\{\theta_k\}_{k=1}^K$ and $\{\psi_k\}_{k=1}^K$ have been paired automatically. We call the above solution the direct covariance matrix (DCM) algorithm.

Define the following two parameters

$$\begin{aligned} L_d &= (\breve{C}_d + 1)/2 \\ L_s &= (\breve{C}_s + 1)/2 \end{aligned} \tag{40}$$

The DOFs of $\hat{\breve{\mathbf{r}}}_d$ and $\hat{\breve{\mathbf{r}}}_s$ are $(2L_d - 1)$ and $(2L_s - 1)$. As using $(2L_d - 1)$ elements can construct an $L_d \times L_d$ matrix at most and for $2(L_s - 1)$ elements it is $L_s \times L_s$. For a linear array, these two sequences of elements correspond to an L_d and L_s length array, and they can resolve $(L_d - 1)$ and $(L_s - 1)$ signals, respectively. The total number of resolvable signals is determined by the longer virtual array of the two, which is $D_{dcm} = L_d - 1 = (\breve{C}_d - 1)/2$, and this number is also equal to the maximum number of circular signals that can be resolved. That means the signals can all be noncircular, or circular, or a mixture of them. To increase this number, we further propose the following ECM sparse representation algorithm.

We first combine $\breve{\mathbf{r}}_d$ and $\breve{\mathbf{r}}_s$ into an extended vector $\breve{\mathbf{r}}$, and obtain the following model

$$\breve{\mathbf{r}} = \breve{\mathbf{A}}_{ds}(\Theta) \cdot \mathbf{T} \cdot \mathbf{v} \tag{41}$$

where

$$\begin{aligned}
&\breve{\mathbf{r}} = \left[\begin{array}{c} \breve{\mathbf{r}}_d \\ \breve{\mathbf{r}}_s \end{array}\right] \\
&\breve{\mathbf{A}}_{ds}(\Theta) = \left[\begin{array}{c} \breve{\mathbf{A}}_d(\Theta) \\ \breve{\mathbf{A}}_s(\Theta) \end{array}\right] \\
&\mathbf{T} = \left[\begin{array}{cc} \mathbf{J}_K & \mathbf{J}_K \end{array}\right] \\
&\mathbf{v} = \left[\begin{array}{c} \mathbf{v}_d \\ \mathbf{v}_s \end{array}\right] \\
&\quad = [\eta_1, \ldots, \eta_K, \rho_1 e^{j\psi_1}\eta_1, \ldots, \rho_{K_n} e^{j\psi_{K_n}}\eta_{K_n}, 0, \ldots, 0]^T
\end{aligned} \tag{42}$$

where $\mathbf{J}_K$ is a $K \times K$ matrix with ones on its anti-diagonal. The last $K - K_n$ elements of $\mathbf{v}$ are zero-valued. In detail, we have

$$\begin{aligned}
&\mathbf{v}(k) = \eta_k && k = 1, \ldots, K \\
&\mathbf{v}(K+k) = \rho_k e^{j\psi_k}\eta_k && k = 1, \ldots, K_n \\
&\mathbf{v}(K+k) = 0 && k = K_n + 1, \ldots, K
\end{aligned} \tag{43}$$

where $\mathbf{v}(i)$ denotes the ith element of $\mathbf{v}$. Then, we go on to define the corresponding parameters $\hat{\breve{\mathbf{r}}}$, $\breve{\mathbf{A}}_{ds}(\hat{\Theta})$ and $\hat{\mathbf{v}}$, and formulate the following problem

$$\begin{aligned}
&min \quad \|\hat{\mathbf{u}}\|_1 \\
&subject\ to \quad \hat{\mathbf{u}}(i) \geq \|[\hat{\mathbf{v}}(i), \hat{\mathbf{v}}(K_g + i)]\|_2 \\
&where \quad \|\hat{\breve{\mathbf{r}}} - \breve{\mathbf{A}}_{ds}(\hat{\Theta}) \cdot \mathbf{T} \cdot \hat{\mathbf{v}}\|_2 \leq \beta
\end{aligned} \tag{44}$$

where $\breve{\mathbf{A}}_{ds}(\hat{\Theta})$ is the new overcomplete dictionary, $\hat{\mathbf{v}} = [\hat{\mathbf{v}}_d^T \hat{\mathbf{v}}_s^T]^T$ with dimension $2K_g \times 1$, $\hat{\mathbf{u}}$ is a spatial distribution vector of dimension $K_g \times 1$, β is the new threshold value which can be calculated as $\beta = \sqrt{\beta_d^2 + \beta_s^2}$.

Note that the l_1 norm is used here as an approximation to the l_0 norm for sparsity maximization. Since the l_1 norm penalizes lager weight coefficients more heavily than smaller ones, the reweighted l_1 norm minimisation method can be employed for a closer approximation to the l_0 norm[31, 32, 33, 34]. Using the reweighted l_1 norm minimisation method, a lager weighting term is introduced to those coefficients with smaller non-zero values and a smaller weighting term to those coefficients with larger non-zero values. This weighting term will change according to the resultant coefficients at each iteration.

For the jth iteration, the problem (44) can be modified as follows:

$$\begin{aligned} &min \quad \sum_{n=1}^{N} \zeta^j(n)|\hat{\mathbf{u}}^j(n)| \\ &subject\ \ to \quad \hat{\mathbf{u}}^j(i) \geq \|[\hat{\mathbf{v}}^j(i), \hat{\mathbf{v}}^j(N+i)]\|_2 \\ &where \quad \|\hat{\tilde{\mathbf{r}}} - \breve{\mathbf{A}}_{ds}(\hat{\Theta}) \cdot \mathbf{T} \cdot \hat{\mathbf{v}}^j\|_2 \leq \beta^i \end{aligned} \tag{45}$$

where the superscript j indicates the value of the corresponding parameters at the jth iteration, and $\zeta^j(n)$ is the reweighting term for the nth row of coefficients, given by $\zeta^j(n) = (|\hat{\mathbf{u}}^{j-1}(n)| + \gamma)^{-1}$. Here, $\gamma > 0$ is required to provide numerical stability to prevent $\zeta^j(n)$ becoming infinity at the current iteration if the value of a weight coefficient is zero at the previous iteration.

The iteration processes are described as follows:

1. For the first iteration ($j = 1$), calculate the initial value $|\hat{\mathbf{u}}|$ by solving (44).
2. Set $j = j + 1$. Use the value of the last $(|\hat{\mathbf{u}}^{j-1}(n)| + \gamma)^{-1}$ to calculate $\zeta^j(n)$, and then find $\hat{\mathbf{v}}^j$ and $\hat{\mathbf{u}}^j$ by solving the problem in (45).
3. Repeat the last step until the positions of non-zero values of the weight coefficients do not change any more for some number of iterations.

This reweighted scheme can be applied to the DCM algorithm in the same way and in our simulations both the DCM and the ECM algorithms are used with their reweighted versions as the non-reweighted ones cannot give a meaningful result in some of the scenarios considered.

The above constrained l_1- norm minimization problems can be solved using cvx, a package for specifying and solving convex problems [35, 36].

The noncircularity phase can be estimated in the same way as in the DCM case. Defining $\hat{\mathbf{v}}_k$ as the kth peak value of $\hat{\mathbf{v}}(N+i)$, then ψ_k can be estimated as

$$\psi_k = arg(\hat{\mathbf{v}}_k) \tag{46}$$

The extended array can be considered as the combination of two ULAs with length L_d and L_s, respectively, which can resolve $D_{ecm} = L_d + L_s - 2 = (\breve{C}_d + \breve{C}_s)/2 - 1$ signals. These DOAs can all be noncircular, or at most $(\breve{C}_d - 1)/2$ circular with the rest being noncircular. Because the CS-based algorithm can utilize the consecutive and nonconsecutive virtual sensors produced by both co-arrays, the number of resolvable signals by ECM is larger than that by ULP. Interestingly, if we only consider the consecutive virtual sensors, the number of resolvable signals by these two algorithms will be exactly the same.

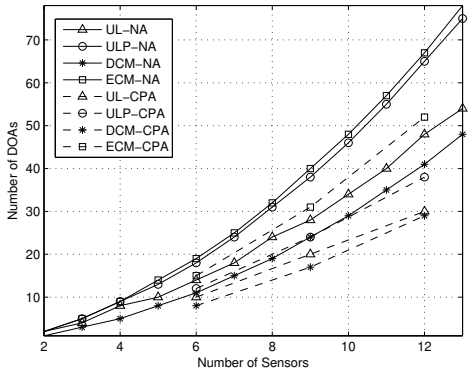


Figure 1: Number of resolvable sources by different algorithms using different sparse array structures.

5. Comparison and Simulation Results

5.1. Number of Resolvable Sources by Different Algorithms

Based on the two commonly used sparse array structures, i.e. the co-prime array (CPA) and the nested array (NA), we give a comparison of the number of resolvable sources by the proposed algorithms. Nested arrays using UL and ULP algorithms are denoted as UL-NA, ULP-NA, respectively, while co-prime array related algorithms are correspondingly denoted as UL-CPA and ULP-CPA. On the other hand, for the DCM and ECM algorithms, we have DCM-NA, ECM-NA, DCM-CPA and ECM-CPA. Define M_1 and M_2 as the two parameters in the co-prime array or the nested array. There are four different algorithms to be examined, and the number of resolvable signals has been provided in previous sections for the four cases, which is $D_{ul} = C_s - 1$, $D_{ulp} = (C_d + C_s)/2 - 1$, $D_{dcm} = (\breve{C}_d - 1)/2$ and $D_{ecm} = (\breve{C}_d + \breve{C}_s)/2 - 1$ for the UL, ULP, DCM and ECM algorithms, respectively. The results are shown in Fig. 1, and the corresponding array settings are listed in Tab. 5.1.

It can be seen that the ECM algorithm is better than the DCM algorithm, while the ULP algorithm is better than the UL algorithm. Since the CS-based algorithm can exploit not only the consecutive virtual sensors, but also the non-consecutive ones, the number of DOAs that can be resolved by the ECM algorithm is greater than that of the ULP algorithm, especially for co-prime arrays. For example, when the sensor number $M = 6$, the number of resolvable DOAs by different algorithms is $D_{ul} = 14, D_{ulp} = 18, D_{dcm} = 11$, and $D_{ecm} = 19$ for the nested array, and

Table 1: DOFs of different sparse array configurations.

	NA	NA	CPA	CPA
M	(M_1, M_2)	(C_d, C_s) $(\breve{C}_d, \breve{C}_s)$	(M_1, M_2)	(C_d, C_s) $(\breve{C}_d, \breve{C}_s)$
2	$(1, 1)$	$(3, 3)$ $(3, 3)$	$--$	$--$ $--$
3	$(1, 2)$	$(7, 5)$ $(7, 5)$	$--$	$--$ $--$
4	$(2, 2)$	$(11, 9)$ $(11, 9)$	$--$	$--$ $--$
5	$(2, 3)$	$(17, 11)$ $(17, 13)$	$--$	$--$ $--$
6	$(3, 3)$	$(23, 15)$ $(23, 17)$	$(2, 3)$	$(15, 11)$ $(17, 15)$
7	$(3, 4)$	$(31, 19)$ $(31, 21)$	$--$	$--$ $--$
8	$(4, 4)$	$(39, 25)$ $(39, 27)$	$--$	$--$ $--$
9	$(4, 5)$	$(49, 29)$ $(49, 33)$	$(3, 4)$	$(29, 21)$ $(35, 29)$
10	$(5, 5)$	$(59, 35)$ $(59, 39)$	$--$	$--$ $--$
11	$(5, 6)$	$(71, 41)$ $(71, 45)$	$--$	$--$ $--$
12	$(6, 6)$	$(83, 49)$ $(83, 53)$	$(4, 5)$	$(47, 31)$ $(59, 47)$
13	$(6, 7)$	$(97, 55)$ $(97, 61)$	$--$	$--$ $--$

it is $D_{ul} = 10, D_{ulp} = 12, D_{dcm} = 8$, and $D_{ecm} = 15$ for the co-prime array. With the increase of the total number of sensors, the ECM algorithm and the ULP algorithm will be more and more advantageous than the other two.

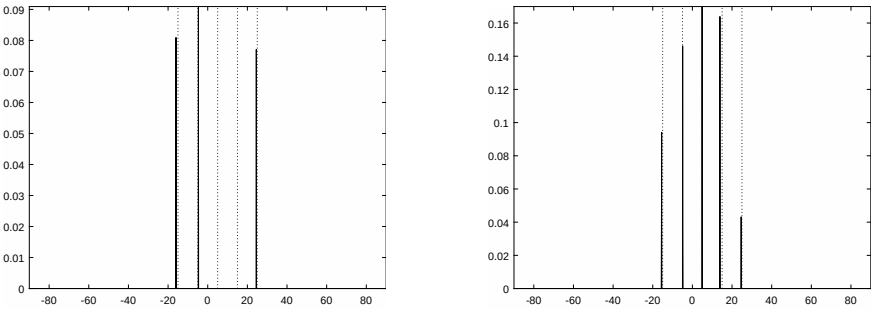
5.2. Computational Complexity of Different Algorithms

The UL and ULP algorithms are subspace-based, while the DCM and ECM algorithms are based on sparse representation solved by iterative convex optimistion methods. So the computational complexity of the former two will be much lower than the latter ones in theory. The computational complexity of the UL and ULP algorithms are mainly on subspace derivations, which are of $O\{[(C_d + 1)/2 + C_s + 1]^3\}$ for UL and $O\{2[(C_d + 1)/2 + C_s + 1]^3\}$ for ULP. The computational complexity for solving (38) and (44) through second-order cone programming is $O[(2K_g)^3]$. Assume J is the total number of iterations, and then the computational complexity of (45) is $O[J(2K_g)^3]$. It is clear that the computational complexity of UL and ULP is mainly dependent on the number of second-order co-array sensors C_d and C_s, while it is mainly determined by the searching grade size K_g for DCM and ECM. As K_g is always much greater than C_d and C_s, the DCM and ECM algorithms will always have larger computational complexity than the UL and ULP algorithms.

5.3. Simulation Results

In this part, simulations are performed to demonstrate the performance of the derived algorithms, i.e. UL, ULP, DCM and ECM. When using CS-based algorithms, the full angle range from -90° to 90° is discretized with a step size of 0.1°, and the number of reweighted iterations is set to be three.

In the first set of simulations, a nested array of $M = 3$ sensors is considered. The parameters of the sparse array is set according to Sec. 5.1 as $M_1 = 1$ and $M_2 = 2$. In this situation, the number of signals can be resolved by the four algorithms (UL-NA, ULP-NA, DCM-NA and ECM-NA) is $D_{ul} = 4$, $D_{ulp} = 5$, $D_{dcm} = 3$ and $D_{ecm} = 5$, respectively. There are five BPSK signals impinging on the array from directions $(-15^\circ, -5^\circ, 5^\circ, 15^\circ, 25^\circ)$. The number of data snapshots is 2000 and the signal-to-noise ratio (SNR) is $10dB$. The simulation results show that the UL-NA algorithm has failed to generate any meaningful result, because the number of roots on the unit circle is less than 10 in this case. The number of roots that are on the unit circle of the ULP-NA is exactly 10, and the DOA estimation results are $(-14.78^\circ, -4.23^\circ, 5.34^\circ, 9.48^\circ, 14.95^\circ)$, very close to the real ones. The results for the DCM-NA and ECM-NA algorithms are shown in Fig. 5.3,



a.DCM-NA. b.ECM-NA.

Figure 2: DOA estimation results of DCM-NA and ECM-NA.

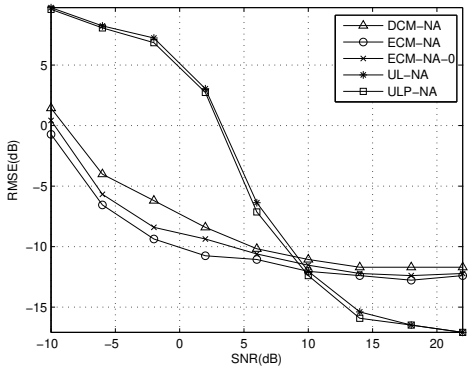


Figure 3: DOA estimation results with a varying SNR.

from which we can see that the DCM-NA can only resolve three DOAs while ECM-NA can resolve all of them effectively.

In the second set of simulations, a nested array with $M = 6$ sensors is considered, and the parameters of the nested array are $M_1 = M_2 = 3$. In theory, the number of signals that can be resolved is 14, 18, 11 and 19 for the UL-NA, ULP-NA, DCM-NA and ECM-NA algorithms, respectively. Six signals arrives from directions $(-25°, -15°, -5°, 5°, 15°, 25°)$. Among them, two are circular, three are BPSK signals, and one is UQPSK signal. The number of snapshots is 2000. The SNR varies from -10dB to 22dB with a step size of $4dB$. The performance of the estimators is obtained by 500 Monte-Carlo simulations, and the root-mean-square error (RMSE) of DOA estimates is shown in Fig. 3, where the legend $ECM - NA - 0$ denotes the unweighted ECM scheme in (44).

It can be seen that when the SNR is low, the performance of the CS-based

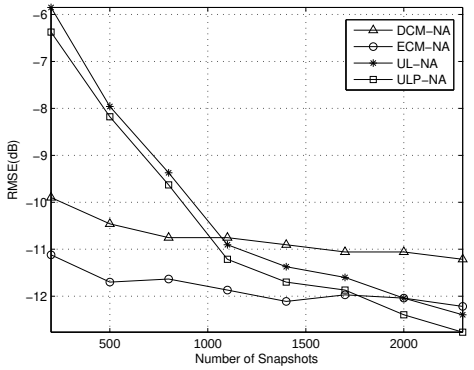


Figure 4: DOA estimation results with a varying snapshot number.

algorithms is much better than the subspace-based ones. When the SNR is greater than 9dB, the subspace-based algorithms start to outperform the CS-based ones. The estimation accuracy of the ECM-NA algorithm is always better than the DCM-NA, while the accuracy of the ULP-NA is better than the UL-NA. Since the number of DOFs exploited by the ECM-NA is the largest, in theory it should always have the best performance. The main reason why it was outperformed by the subspace-based algorithms when SNR is greater than 9dB is that the performance of the CS-based algorithms is dependent on the chosen grid size for the full angle range and the larger the grid size, the better their performance. For example, we have tried a step size of 0.05° instead of 0.1° and the CS-based algorithms were only outperformed by the subspace-based ones when the SNR is larger than 11dB, but at the cost of significantly increased computational complexity. This is the so-called off-grid problem and many methods have been proposed to tackle it and most of them can be applied here directly [37, 38, 39, 40]. Moreover, as expected, the reweighted scheme of ECM in (45) always performs better than the unweighted one in (44), especially for low SNR ranges.

In the third set of simulations, we fix the SNR at 10 dB and change the snapshot number. All the other parameters are the same as before. The results are shown in Fig. 4.

It can be seen that RMSEs of the CS-based algorithms do not change very much for the considered range of snapshot number, but those of the subspace-based algorithms varies a lot for different number of snapshots. When the number of snapshots is greater than 1700, the estimation accuracy of the subspace-based algorithms outperforms the CS-based one.

Finally, we give a brief comparison of the running time for different

Table 2: Running time for different algorithms.

	UL	ULP	DCM	ECM
Running Time	$0.0282s$	$0.0324s$	$149.2352s$	$175.4136s$

algorithms as an indicator of their computational complexity. The number of snapshots is fixed at 2000 and the average processing time for the four different algorithms are listed in Tab. 5.3. It can be seen from the table that the time used by the CS-based algorithms is more than 5000 times that of the subspace-based ones, highlighting a common issue of the CS-based DOA estimation algorithms in comparison with subspace-based ones.

6. Conclusion

The virtual array generation process based on typical sparse arrays has been studied for a mixture of circular and non-circular impinging signals by analysing both the covariance matrix and the pseudo covariance matrix of the physical array. Two sub-arrays can be created by the matrices: one is the traditional difference co-array and the other one is the new sum co-array. The number of consecutive virtual array sensors has been provided for the nested array, but it is difficult to give a closed-form result for a general sparse array. By arranging the elements of the extended covariance matrix of the physical array in different ways, two classes of DOA estimation algorithms have been developed. The first is subspace-based and two algorithms are developed: one is called UL and one is called ULP; the second class is CS-based and again two algorithms are developed, which are called DCM and ECM, respectively. Both the consecutive and non-consecutive parts of the virtual array can be exploited by the CS-based algorithms, while only the consecutive part is exploited by the subspace-based ones. As a result, the CS-based solution can have a better performance than the subspace-based one, although at the cost of a significantly increased computational complexity. Simulation results have been provided to verify the performance of the proposed algorithms.

7. Acknowledgement

This work was supported by the National Natural Science Foundation of China (61405150 and 61628101).

References

[1] H. L. Van Trees, *Optimum Array Processing, Part IV of Detection, Estimation, and Modulation Theory*. New York: Wiley, 2002.

[2] P. Pal and P. P. Vaidyanathan, "Co-prime sampling and the music algorithm," in *IEEE Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop(DSP/SPE)*, Sedona, AZ, January 2011, pp. 289–294.

[3] P. P. Vaidyanathan and P. Pal, "Sparse sensing with co-prime samplers and arrays," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 573–586, Feb. 2011.

[4] Y. M. Zhang, M. G. Amin, and B. Himed, "Sparsity-based DOA estimation using co-prime arrays," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013, pp. 3967–3971.

[5] P. Pal and P. P. Vaidyanathan, "Nested arrays: a novel approch to array processing with enhanced degrees of freedom," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4167–4181, Aug. 2010.

[6] Z. B. Shen, C. X. Dong, Y. Y. Dong, G. Q. Zhao, and L. Huang, "Broadband DOA estimation based on nested arrays," *International Journal of Antennas and Propagation*, vol. 2015, 2015.

[7] P. Pal and P. P. Vaidyanathan, "Coprime sampling and the MUSIC algorithm," in *Proc. IEEE Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE)*, Sedona, AZ, Jan. 2011, pp. 289–294.

[8] C.-L. Liu and P. P. Vaidyanathan, "Remarks on the spatial smoothing step in coarray MUSIC," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1438–1442, Sep. 2015.

[9] Q. Shen, W. Liu, W. Cui, S. L. Wu, Y. D. Zhang, and M. Amin, "Low-complexity direction-of-arrival estimation based on wideband co-prime arrays," *IEEE Trans. Audio, Speech and Language Processing*, vol. 23, pp. 1445–1456, September 2015.

[10] S. Qin, Y. D. Zhang, and M. G. Amin, "Generalized coprime array configurations for direction-of-arrival estimation," *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1377–1390, March 2015.

[11] J. J. Cai, D. Bao, and P. Li, "Doa estimation via sparse recovering from the smoothed covariance vector," *Journal of Systems Engineering and Electronics*, vol. 27, no. 3, pp. 555–561, June 2016.

[12] J. J. Cai, W. Liu, R. Zong, and Q. Shen, "Doa estimation via sparse recovering from the smoothed covariance vector," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 480–484, April 2017.

[13] X. M. Yang, G. J. Li, and Z. Zheng, "Doa estimation of noncircular signal based on sparse representation," *Wireless Personal Communications*, vol. 82, pp. 2363–2375, February 2015.

[14] P. Charge, Y. Wang, and J. Saillard, "A non-circular sources direction finding method using polynomial rooting," *Signal Processing*, vol. 81, pp. 1765–1770, 2001.

[15] H. Abeida and J. Delmas, "Music-like estimation of direction of arrival for noncircular sources," *IEEE Transactions on Signal Processing*, vol. 54, no. 7, pp. 2678–2690, JULY 2006.

[16] J. Liu, Z. Huang, and Y. Zhou, "Extended 2q-music algorithm for noncircular signals," *Signal Processing*, vol. 88, pp. 1327–1339, 2008.

[17] H. Abeida and J. Delmas, "Statistical performance of music-like algorithms in resolving noncircular sources," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4317–4329, SEPTEMBER 2008.

[18] F. F. Gao, A. Nallanathan, and Y. Wang, "Improved music under the coexistence of both circular and noncircular sources," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3033–3038, 2008.

[19] Z. T. Huang, Z. M. Liu, J. Liu, and Y. Y. Zhou, "Performance analysis of music for non-circular signals in the presence of mutual coupling," *IET radar, sonar & navigation*, vol. 4, no. 5, pp. 703–711, 2010.

[20] J. Steinwandt, F. Roemer, M. Haardt, and G. Galdo, "R-dimensional ESPRIT-type algorithms for strictly second-order non-circular sources and their performance analysis ," *IEEE Transactions on Signal Processing*, vol. 62, no. 18, pp. 4824–4838, September 2014.

[21] H. Chen, C. P. Hou, W. Liu, W. P. Zhu, and M. N. S. Swamy, "Efficient two-dimensional direction of arrival estimation for a mixture of circular and noncircular sources," *IEEE Sensors Journal*, vol. 16, no. 8, pp. 2527–2536, April 2016.

[22] F. F. Gao, A. Nallanathan, and Y. D. Wang, "Improved music under the coexistence of both circular and noncircular sources," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3033–3038, 2008.

[23] J. Steinwandt, F. Roemer, and M. Haardt, "Esprit-type algorithms for a received mixture of circular and strictly non-circular signals," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Brisbane, QLD, Australia, April 2015.

[24] ——, "Analytical performance assessment of esprit-type algorithms for coexisting circular and strictly non-circular signals," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Shanghai, China, March 2016.

[25] J. Steinwandt, F. Roemer, M. Haardt, and G. Galdo, "Deterministic Cramr-Rao Bound for Strictly Non-Circular Sources and Analytical Analysis of the Achievable Gains ," *IEEE Transactions on Signal Processing*, vol. 64, no. 17, pp. 4417–4431, September 2016.

[26] Z. Liu, Z. Huang, Y. Zhou, and J. Liu, "Direction-of-arrival estimation of noncircular signals via sparse representation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, pp. 2690–2698, July 2012.

[27] J. Steinwandt, C. Steffens, M. Pesavento, and M. Haardt, "Sparsity-aware direction finding for strictly non-circular sources based on rank minimization," in *Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop*, PCU-Rio, Brazil, 2016.

[28] J. Steinwandt, F. Roemer, and M. Haardt, "Sparsity-based direction-of-arrival estimation for strictly non-circular sources," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Shanghai, China, March 2016.

[29] B. Picinbono, "On circularity," *IEEE Transactions on Signal Processing*, vol. 42, pp. 3473–3482, December 1994.

[30] J. Lacoume, "Complex random variables and signals," *Traitement du signal*, vol. 15, pp. 535–544, 1998.

[31] E. Candés, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, 2008.

[32] G. Prisco and M. DÚrso, "Maximally sparse arrays via sequential convex optimizations," *IEEE Antennas and Wireless Propagation Letters*, vol. 11, pp. 192–195, 2012.

[33] B. Fuchs, "Synthesis of sparse arrays with focused or shaped beampattern via sequential convex opeimizations," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 7, pp. 3499–3503, 2012.

[34] B. Zhang and W. Liu, and X. Gou, "Compressive sensing based sparse antenna array design for directional modulation," *IET Microwaves, Antennas and Propagation*, vol. 11, no. 5, pp. 634–641, 2017.

[35] C. Research, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," http://cvxr.com/cvx, September 2012.

[36] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph$_$dcp.html.

[37] D. Malioutov, M. Çetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," vol. 53, no. 8, pp. 3010–3022, Aug. 2005.

[38] Z. Yang, L. Xie, and C. Zhang, "Off-grid direction of arrival estimation using sparse bayesian inference," *IEEE Trans. on Signal Process.*, vol. 61, no. 1, pp. 38–43, Jan. 2013.

[39] T. Zhao and A. Nehorai, "Sparse direction of arrival estimation using co-prime arrays with off-grid targets," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 26–29, Jan. 2014.

[40] T. Zhao, P. Yang, and A. Nehorai, "Joint sparse recovery method for compressed sensing with structured dictionary mismatches," *IEEE Trans. on Signal Process.*, vol. 62, no. 19, pp. 4997–5008, Oct. 2014.