Sparse Source Localization Using Perturbed Arrays via Bi-Affine Modeling

Ali Koochakzadeh^{a,*}, Piya Pal^a

^a*Dept. of Electrical and Computer Engineering, University of Maryland*

Abstract

Non-uniform spatial sampling geometries, such as nested and coprime arrays, are provably capable of localizing $O(M^2)$ sources using only M sensors. However, such guarantees require the physical locations of the sensors to satisfy certain constraints, as dictated by the corresponding array geometries. In this paper, we consider the scenario when these constraints may be violated, leading to unknown perturbations on the locations of sensors. Such perturbations can have detrimental effect on the performance of virtual array based direction-of-arrival (DOA) estimation algorithms, since the perturbed virtual array will no longer be a uniform linear array (ULA). We propose a novel self-calibration approach for underdetermined DOA estimation with such arrays, that makes extensive use of the redundancies (or repeated elements) in the virtual array. Assuming small perturbations, and a sparse grid-based model for the DOAs, we extract a novel “bi-affine” model (affine in the perturbation variable, and linear in the source powers) from the covariance matrix of the received signals. The redundancies in the co-array are then exploited to eliminate the nuisance perturbation variable, and reduce the bi-affine problem to a linear underdetermined (sparse) problem in source powers, from which the DOA’s can be exactly recovered under suitable conditions. This reduction is derived for both ULA and a newly-introduced robust version of coprime arrays, when the covariance matrix of the received signals is exactly known. Our approach is compared and

*Corresponding author

Email addresses: alik@umd.edu (Ali Koochakzadeh), ppal@umd.edu (Piya Pal)

contrasted with recently developed algorithms for blind gain and phase calibration (BGPC), whose signal model is fundamentally different from ours. We also provide an iterative algorithm to jointly solve for the DOAs and perturbation values when we can only estimate the covariance matrix using a finite number of snapshots.

Keywords: Coprime Sampling, Co-array, Bi-Affine Model, Array Perturbations, Sparse Source Localization

1. Introduction

Directions-of-Arrival (DOA) estimation of energy emitting sources using sensor arrays finds important application in problems ranging from target localization in radar system to speech enhancement using microphone arrays. In recent times, new sparse array geometries, such as coprime [1] and nested arrays [2], have been proposed that are capable of identifying $O(M^2)$ sources using just M sensors, exploiting the enhanced degrees of freedom offered by their difference co-arrays (or, virtual arrays) [3]. In order to exploit the enhanced degrees of freedom, so far, two main approaches for DOA estimation have been proposed: 1) Subspace methods 2) Sparsity based methods. In the former approach, which is based on the MUSIC algorithm, the subspace properties of the spatially smoothed co-array manifold is used to estimate the DOAs [4]. However, in the latter approach, the range of all possible directions is discretized into a grid, and then the DOA estimation problem is reformulated as a sparse representation problem [5, 6, 7, 8]. We review this approach in more details in Section 2.

It is well known that array imperfections such as gain and/or phase error, perturbations in sensor locations, and mutual coupling, can significantly degrade the performance of DOA estimation algorithms [9], [10]. This is mainly due to the strong dependence of these algorithms on the accurate knowledge of the underlying array manifold. In this paper, we consider the sensor location error as the only imperfection associated with the physical array, i.e., we

assume that the sensor locations are perturbed from their nominal positions. The problem of DOA estimation using such perturbed arrays has been well studied for more than two decades. Existing approaches mostly treat the perturbations as unknown but deterministic parameters, and then estimate these parameters jointly with the DOAs. Classical methods such as [10], [11], [12] [13], resolve array uncertainties using eigenstructure-based methods, or variants of the maximum-likelihood approach. Recently, [14] proposed a unified framework for different kind of array imperfections, and proposed a Bayesian approach for array calibration and DOA estimation. However, these approaches mostly work for an overdetermined signal model (fewer sources than sensors), primarily because many of them consider a uniform linear array.

In recent times, the problem of blind gain and phase calibration (BGPC) has been formulated as a bilinear problem [15], which in turn, can be recast as a convex optimization problem, using the idea of “lifting” [16, 17, 18]. However, such a formulation does not consider the concept of co-array, and, hence their guarantees are not applicable for an underdetermined signal model where the number of sources can possibly be $O(M^2)$.

In contrast, the authors in [19], studied the effect of co-array geometry on the BGPC problem and proposed a new self-calibration algorithm for nested arrays in presence of gain/phase errors. Their approach builds on and extends the method in [12], which was originally proposed for a ULA. However, in this paper, we consider perturbations in sensor locations, which gives rise to a signal model, which is distinctly different from that considered in [19]. In BGPC problems, the gain and/or phase of the sensors are unknown, and the goal is to resolve both unknown gain and/or phase and the DOAs. In our case, we assume that the phase and gain of the signals received from the sensors are ideal, but the location of the sensors are perturbed. We will compare the signal model defined for gain/phase error, which has been studied in [19], against sensor location error in Section 2.2, and establish important differences between them.

Since the self calibration algorithm developed in [19] cannot be directly applied to our case, we follow a different approach in this paper. We assume that

the perturbations are small, so that we can approximate the coarray manifold using its first order Taylor series expansion. This formulation leads to a “bi-affine” model, which is linear in source powers, and affine in the perturbation variable. We show that it is possible to recover the DOAs even in presence of the nuisance perturbation variables, via a clever elimination of variables. By exploiting the pattern of repeating elements, it is possible to reduce the said bi-affine problem to a linear underdetermined (sparse) problem in source powers, which can be efficiently solved using ℓ_1 minimization. We establish precise conditions under which such reduction is possible, for both ULA and a robust version of coprime arrays.

The paper is organized as follows. In Sec. 2 we compare and contrast different kinds of array imperfections (gain/phase error, sensor location perturbation) in terms of their effects on the difference co-array. In Sec. 3, we introduce the bi-affine model for DOA estimation with perturbed sensors. Sec. 4, establishes a transformation under which we can write the bi-affine problem as a linear problem in source powers, via elimination of the unknown perturbation variable. The specific details of this transformation depend on the array geometry. In Sec. 5, we review an iterative algorithm proposed in [20] to jointly solve for DOAs and source powers when we only have an estimate of the covariance matrix. Numerical simulations are conducted in Sec. 6. Sec. 7 concludes the paper.

Notation: Throughout this paper, matrices are represented by upper case bold letters, and vectors by lower case bold letters. The symbol x_i represents the i th entry of a vector $\mathbf{x}$. The symbol j denotes the imaginary unit $\sqrt{-1}$. The symbols $(\cdot)^*$, $(\cdot)^T$, $(\cdot)^H$ stand for the conjugate, transpose, and hermitian, respectively. The symbols $\circ$, $\odot$, $\otimes$ represent the Hadamard product, Khatri-Rao product, and Kronecker product, respectively. The symbol $\|\cdot\|_F$ denotes the matrix Frobenius norm and $\text{vec}(\cdot)$ represents the vectorized form of a matrix.

2. Signal Model for Gain/Phase Error vs Location Errors

Consider a linear array of M antennas impinged by K narrow-band sources with unknown directions of arrival (DOA) $\boldsymbol{\theta} \in \mathbb{R}^K$, $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_K]^T$. Let $\mathbf{y}[l] \in \mathbb{C}^M$ be the vector of signals received by the M antennas, $\mathbf{x}[l] \in \mathbb{C}^K$ represent the emitted signals from K sources, and $\mathbf{w}[l]$ be the additive noise (all corresponding to the l th time snapshot). The source signals are assumed to be zero mean, and pairwise uncorrelated, and the noise vector is zero mean, i.i.d. with variance σ_w^2 , and uncorrelated from the signal. We do not make any specific assumptions on the distribution of the signal or noise.

The sensors are designed to be at the nominal locations $\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_M$, where $\tilde{d}_m \in \mathbb{R}$ for $1 \leq m \leq M$, and $\tilde{d}_m = Dd_m$. Here, $d_m \in \mathbb{Z}$, and D is the minimum inter-element spacing of the array, which is typically chosen to be $D = \lambda/2$, λ being the carrier wavelength of the narrowband sources. Note that d_m are the normalized sensor locations (and $\tilde{d}_m$ are the actual sensor locations). In the sequel, we will use the normalized locations as we introduce the perturbed array model. In this paper, we consider two different array geometries: uniform linear array (ULA), and coprime array. In a ULA, we have $d_m = m - 1$, for $m = 1, \dots, M$. A coprime array, however, is comprised of two different ULAs with spacings N_1 and N_2 , where N_1 and N_2 are coprime numbers. We will review the coprime arrays in more detail in Sec. 2.1.2. To simplify the notations, we designate a spatial frequency $\omega_i = \frac{2\pi D}{\lambda} \sin \theta_i$ corresponding to each direction of arrival θ_i for $1 \leq i \leq K$. Choosing $D = \lambda/2$, we have $\omega_i = \pi \sin \theta_i$. Also, let $\boldsymbol{\omega} = [\omega_1, \dots, \omega_K]^T$ be the vector of spatial frequencies associated with the K sources.

Let $\boldsymbol{\zeta} \in \mathbb{C}^M$ be a vector of unknown parameters associated with array imperfections, such as gain/phase, or sensor location errors. The received samples at the time instant l can be written as

$$\mathbf{y}[l] = \mathbf{A}(\boldsymbol{\omega}, \boldsymbol{\zeta})\mathbf{x}[l] + \mathbf{w}[l] \quad (1)$$

in which $\mathbf{A}(\boldsymbol{\omega}, \boldsymbol{\zeta}) = [\mathbf{a}(\omega_1, \boldsymbol{\zeta}), \dots, \mathbf{a}(\omega_K, \boldsymbol{\zeta})]$ denotes the array manifold, and $\mathbf{a}(\omega_i, \boldsymbol{\zeta}) \in \mathbb{C}^M$ is the steering vector for the i th source. In the absence of array

imperfections ($\boldsymbol{\zeta} = \mathbf{0}$), the m th element of the steering vector corresponding to direction θ_i is given by $a_m(\omega_i, \mathbf{0}) = e^{jd_m\omega_i}$. In the following subsections, we will first review the concept of a virtual array by considering the covariance matrix for the unperturbed problem [2]. Subsequently, we will discuss and distinguish the signal models corresponding to two different kinds of array imperfections: (i) gain/phase error, and (ii) sensor location error.

2.1. Virtual array in the absence of array imperfections

In the absence of array imperfections ($\boldsymbol{\zeta} = \mathbf{0}$), we can write the covariance matrix of the received signals as

$$\mathbf{R}_y = E(\mathbf{y}\mathbf{y}^H) = \mathbf{A}_0(\boldsymbol{\omega})\mathbf{R}_x(\mathbf{A}_0(\boldsymbol{\omega}))^H + \sigma_w^2\mathbf{I} \quad (2)$$

where $\mathbf{R}_x = E(\mathbf{x}\mathbf{x}^H)$ is the covariance matrix of the sources, and $\mathbf{A}_0(\boldsymbol{\omega}) = \mathbf{A}(\boldsymbol{\omega}, \mathbf{0})$. Assuming that the sources are uncorrelated, i.e., $\mathbf{R}_x$ is diagonal, following [2] the vectorized form of the covariance matrix can be written as

$$\mathbf{z} = \mathbf{A}_{\text{KR},0}(\boldsymbol{\omega})\tilde{\mathbf{p}} + \sigma_w^2 \text{vec}(\mathbf{I}), \quad (3)$$

where $\mathbf{A}_{\text{KR},0}(\boldsymbol{\omega}) = \mathbf{A}_0(\boldsymbol{\omega})^* \odot \mathbf{A}_0(\boldsymbol{\omega})$ is the difference co-array, $\tilde{\mathbf{p}} = [p_1, p_2, \dots, p_K]$ is the diagonal of $\mathbf{R}_x$, and $\mathbf{z} = \text{vec}(\mathbf{R}_y)$. The $(m + (m' - 1)M, i)$ -th element of $\mathbf{A}_{\text{KR},0}(\boldsymbol{\omega})$ is given by $e^{j\omega_i(d_m - d_{m'})}$. Therefore, each column of $\mathbf{A}_{\text{KR},0}(\boldsymbol{\omega})$ is characterized by the difference co-array:

$$S_{\text{ca}} = \{d_m - d_{m'}, 1 \leq m, m' \leq M\}$$

Define $M_{\text{ca}} = |S_{\text{ca}}|$ to be the number of distinct elements in the unperturbed virtual coarray, and also let $M'_{\text{ca}} = \frac{M_{\text{ca}} - 1}{2}$ be the size of the positive half of the coarray.

Weight Function: Let $w(k)$ be the number of repetitions of the lag k in the difference coarray, i.e., number of pairs of indices (i, j) such that $d_i - d_j = k$. The function $w(k)$ is known as the weight function of the coarray [2]. In later sections, we will make extensive use of $w(k)$ to eliminate the unwanted perturbation variable in our proposed biaffine model. We will consider array

geometries such that $w(k) > 1$ for an adequate number of lags k contained in S_{ca} . The weight functions corresponding to ULA and robust coprime array (defined later) are illustrated in Figure 1, in which $N_1 = 4, N_2 = 9, M = 32$.

2.1.1. ULA and its Coarray

For a ULA, we have $d_m = m - 1$, with $1 \leq m \leq M$. The coarray corresponding to a ULA is given by

$$S_{\text{ca}}^{\text{ULA}} = \{m | m = -(M-1), \dots, M-1\}$$

Therefore, for a ULA we have $M_{\text{ca}} = 2M - 1$, and $M'_{\text{ca}} = M - 1$.

2.1.2. Coprime Array and its Coarray

In a coprime array, we choose two coprime numbers N_1, N_2 ($N_1 < N_2$), and place the sensors on two ULAs with inter-element spacings N_1 , and N_2 . In the original coprime array introduced in [4], the first ULA is comprised of N_2 sensors located at the locations $d_n^{(1)} = nN_1$, and the second ULA has $2N_1$ sensors which are located at the $d_m^{(2)} = mN_2$, where $0 \leq n \leq N_2 - 1, 1 \leq m \leq 2N_1 - 1$, and we have $d_i = d_i^{(1)}$ for $1 \leq i \leq N_2$, and $d_j = d_{j-N_2}^{(2)}$ for $N_2 + 1 \leq j \leq 2N_1 + N_2 - 1$. This choice leads to a virtual array such that

$$S_{\text{ca}}^{\text{Coprime}} \supset \{n | n = -N_1N_2, \dots, N_1N_2\}$$

Hence, $M'_{\text{ca}} > N_1N_2$, and $M_{\text{ca}} > 2N_1N_2 - 1$. In other words, the virtual coarray has $O(N_1N_2)$ elements, although the physical array has only $M = 2N_1 + N_2 - 1$ sensors. This increased degrees of freedom makes it possible to resolve more sources than the available number of sensors. In the following sections, we will see what happens to the coarray when we have imperfections in the coprime array.

2.2. Perturbed Sensor Array and its Virtual Array

Assume that the sensors are not located at their nominal positions, and their perturbed locations are given by $d_1 + \delta_1, d_2 + \delta_2, \dots, d_M + \delta_M$, in which

$\boldsymbol{\delta} \in \mathbb{R}^M$, $\boldsymbol{\delta} = [\delta_1, \delta_2, \dots, \delta_M]^T$ is an unknown perturbation vector. In this case, we have $a_m(\omega_i, \boldsymbol{\delta}) = e^{j(d_m + \delta_m)\omega_i}$, from which we can rewrite the array manifold $\mathbf{A}(\boldsymbol{\omega}, \boldsymbol{\delta}) \in \mathbb{C}^{M \times K}$ as

$$\mathbf{A}(\boldsymbol{\omega}, \boldsymbol{\delta}) = \mathbf{A}_0(\boldsymbol{\omega}) \circ \mathbf{P}(\boldsymbol{\omega}, \boldsymbol{\delta}), \quad (4)$$

where $\mathbf{A}_0(\boldsymbol{\omega}) = \mathbf{A}(\boldsymbol{\omega}, \mathbf{0})$ is the unperturbed array manifold, and $\mathbf{P}(\boldsymbol{\omega}, \boldsymbol{\delta}) \in \mathbb{C}^{M \times K}$ is a perturbation matrix whose (m, i) th element is given by $e^{j\omega_i \delta_m}$. The covariance matrix of the received signals can therefore be written as

$$\mathbf{R}_y = E(\mathbf{y}\mathbf{y}^H) = \mathbf{A}(\boldsymbol{\omega}, \boldsymbol{\delta})\mathbf{R}_x(\mathbf{A}(\boldsymbol{\omega}, \boldsymbol{\delta}))^H + \sigma_w^2 \mathbf{I}, \quad (5)$$

and similar to Sec. 2.1, we can write the vectorized form of the covariance matrix to get

$$\mathbf{z} = \mathbf{A}_{\text{KR}}(\boldsymbol{\omega}, \boldsymbol{\delta})\tilde{\mathbf{p}} + \sigma_w^2 \text{vec}(\mathbf{I}), \quad (6)$$

The $(m + (m' - 1)M, k)$ -th element of $\mathbf{A}_{\text{KR}}(\boldsymbol{\omega}, \boldsymbol{\delta})$ is given by $e^{j\omega_k(d_m + \delta_m - d_{m'} - \delta_{m'})}$. The difference coarray corresponding to the perturbed array is then

$$S_{\text{ca}}^\delta = \{d_m + \delta_m - d_{m'} - \delta_{m'}, 1 \leq m, m' \leq M\}$$

Grid based model: In this paper, we consider a grid-based model for the DOAs [6, 20], where the range of possible directions $[-\pi/2, \pi/2]$, is quantized into N_θ grid points. We can then construct a *grid based* array manifold $\mathbf{A}_{\text{grid}}(\boldsymbol{\delta}) \in \mathbb{C}^{M \times N_\theta}$, where each column of this matrix is a steering vector corresponding to a particular direction on the grid. The grid-based co-array manifold is given by $\mathbf{A}_{\text{ca}}(\boldsymbol{\delta}) = \mathbf{A}_{\text{grid}}(\boldsymbol{\delta})^* \odot \mathbf{A}_{\text{grid}}(\boldsymbol{\delta})$. *Clearly, this co-array manifold only depends on $\boldsymbol{\delta}$, and the structure of the array.* We can now rewrite (6) as

$$\mathbf{z} = \mathbf{A}_{\text{ca}}(\boldsymbol{\delta})\mathbf{p} + \sigma_w^2 \text{vec}(\mathbf{I}), \quad (7)$$

where the non-zero elements of $\mathbf{p} \in \mathbb{C}^{N_\theta}$ are equal to the corresponding elements of $\tilde{\mathbf{p}}$. The locations of the non zero elements of $\mathbf{p}$ can be used to recover the DOAs.

The model derived in this subsection for sensor location errors is distinctly different from the model based on gain/phase error, which we discuss next.

2.3. Covariance matrix with unknown sensor gain & phase

The problem of co-array based DOA estimation with phase and gain errors has been thoroughly studied in [19] for ULA and nested arrays. In this Section, we will repeat some of the results from [19] to distinguish it from the problem considered in this paper. The received signal at an array with gain and phase errors is given by

$$\mathbf{y}[l] = \mathbf{\Phi}\mathbf{\Psi}\mathbf{A}(\omega)\mathbf{x}[l] + \mathbf{w}[l]$$

where $\mathbf{\Phi} = \text{diag}(e^{j\phi_1}, \dots, e^{j\phi_M})$, $\mathbf{\Psi} = \text{diag}(\psi_1, \dots, \psi_M)$ are the respective phase and gain errors, in which $\psi_m > 0$ for $1 \leq m \leq M$. We can stack the vectors $\mathbf{y}[l]$ for $l = 1, \dots, L$ into the columns of $\mathbf{Y} \in \mathbb{C}^{M \times L}$ (and similarly for $\mathbf{X}$ and $\mathbf{W}$) to get

$$\mathbf{Y} = \mathbf{\Lambda}\mathbf{A}(\omega)\mathbf{X} + \mathbf{W}, \quad (8)$$

in which $\mathbf{\Lambda} = \mathbf{\Phi}\mathbf{\Psi}$. This problem is known as the Blind Gain and Phase Calibration (BGPC), which has been studied recently in [15, 18]. However, these results do not cast the problem in coarray domain, and hence do not address the possibility of resolving more sources than the number of sensors.

The authors in [19], on the other hand, cast the BGPC problem in the co-array domain by using properties of the covariance matrix $\mathbf{R}_{\mathbf{y}}$ for nested arrays and ULA. Let us first write the covariance matrix of the received signal

$$\mathbf{R}_{\mathbf{y}} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{A}(\omega)\mathbf{R}_{\mathbf{x}}(\mathbf{A}(\omega))^H\mathbf{\Psi}^H\mathbf{\Phi}^H + \sigma_w^2\mathbf{I} \quad (9)$$

The (m, n) th element of $\mathbf{R}_{\mathbf{y}}$ is then given by

$$R_{m,n} = \psi_m\psi_n e^{j(\phi_m - \phi_n)} \left(\sum_{k=1}^K p_k e^{j(d_m - d_n)\omega_k} + \delta_{m,n}\sigma_w^2 \right)$$

with $\delta_{m,n} = 1$ if $m = n$, and $\delta_{m,n} = 0$ otherwise.

Remark 1. Comparing the covariance matrices corresponding to sensor location errors, and gain/phase errors, which are respectively given by (5) and (9), we notice some important differences. In (5), the array imperfections appear

in the form of the matrix $\mathbf{P}(\boldsymbol{\omega}, \boldsymbol{\delta})$ which forms a Hadamard product with the array-manifold. However, in (9), the gain and phase errors are captured in the matrix $\mathbf{A}$ which gets multiplied to the array manifold matrix. Due to this, the phase error in BGPC problems is independent of the directions of arrival of sources, whereas, in presence of location errors, the phase error in each entry of the array manifold is a function of the DOAs. Therefore, the approach in [19] cannot be directly used to resolve the sensor location errors in our case.

3. Formulation as a Bi-Affine problem

3.1. The Bi-Affine Model

In this Section, we derive a bi-affine model from the covariance matrix (5) corresponding to a sensor array with perturbed locations. Our main assumption is that $\boldsymbol{\delta}$ is small enough so that we can approximate the coarray manifold $\mathbf{A}_{\text{ca}}(\boldsymbol{\delta})$ using the first order Taylor series expansion as follows:

$$(\mathbf{A}_{\text{ca}}(\boldsymbol{\delta}))_{(m'-1)M+m,i} \simeq e^{j(d_m-d'_m)\omega_i} (1 + (\delta_m - \delta_{m'})j\omega_i),$$

which can be also written in the matrix form as

$$\mathbf{A}_{\text{ca}}(\boldsymbol{\delta}) \simeq \mathbf{A}_{\text{ca},0} + \boldsymbol{\Delta} \mathbf{A}_{\text{ca},0} \Upsilon \quad (10)$$

where $\mathbf{A}_{\text{ca},0}$ denotes the unperturbed co-array manifold, $\Upsilon = j \text{diag}(\omega_1, \dots, \omega_{N_\theta})$, and $\boldsymbol{\Delta} \in \mathbb{R}^{M^2}$ is given as

$$\begin{bmatrix} \text{diag}(\boldsymbol{\delta}) - \delta_1 \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \text{diag}(\boldsymbol{\delta}) - \delta_2 \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \text{diag}(\boldsymbol{\delta}) - \delta_M \mathbf{I} \end{bmatrix} \quad (11)$$

Therefore, (7) can be approximated as

$$\mathbf{z} = (\mathbf{A}_{\text{ca},0} + \boldsymbol{\Delta} \mathbf{A}_{\text{ca},0} \Upsilon) \mathbf{p} + \sigma_w^2 \text{vec}(\mathbf{I}), \quad (12)$$

To suppress the effect of noise, we discard the 0th lag of the co-array, and only keep the elements of $\mathbf{z}$ corresponding to the positive half of the co-array to

obtain ¹:

$$\mathbf{z}^u = (\mathbf{A}_{\text{ca},0}^u + \mathbf{\Delta}^u \mathbf{A}_{\text{ca},0}^u \mathbf{\Upsilon}) \mathbf{p} \quad (13)$$

where $\mathbf{A}_{\text{ca},0}^u$ is the unperturbed co-array manifold with rows corresponding to the positive lags, where we retain repeated rows (that correspond to the same lag in the virtual array). The matrix $\mathbf{\Delta}^u$ is constructed from $\mathbf{\Delta}$ by retaining only the rows corresponding to those of $\mathbf{A}_{\text{ca},0}^u$. Notice that unlike the approach in [4, 2], we also keep the repeated rows of $\mathbf{A}_{\text{ca},0}^u$, since their corresponding rows in $\mathbf{z}^u$ may not be repeated due to the presence of perturbations. Hence, this redundancy in the rows of $\mathbf{A}_{\text{ca},0}^u$ can help us to get more information on the perturbations.

Inspired by the so-called *bilinear* model in the literature, which arises in various problems such as the blind gain and phase calibration (BGPC) problem [15], we call the model given in (13) a “*bi-affine*” model, since $\mathbf{z}^u$ is a linear function of $\mathbf{p}$ and affine function of $\mathbf{\delta}$. Given the covariance matrix $\mathbf{R}_y$ (or equivalently $\mathbf{z}^u$), the goal is to recover the sparse vector $\mathbf{p}$ from the bi-affine model defined in (12).

4. Source Localization: Bi-Affine to Linear Transformation

Under the grid-based model, the DOAs can be estimated from the support of the sparse vector $\mathbf{p}$ that is a solution to the bi-affine system of equations (13). In general, (13) can admit multiple solutions in the variables $(\mathbf{\delta}, \mathbf{p})$. While the column rank of a matrix describing a linear system of equations determines if it admits a unique solution, to the best of our knowledge, no such general condition exists for a bi-affine (or even bi-linear) system which can be used as a test for existence of unique solution.

¹For ease of exposition, we only consider the positive half of the co-array and demonstrate how to eliminate the spurious variable $\mathbf{\delta}$. However, with straightforward modifications to the proposed technique, it is also possible to incorporate the negative half to generate more augmented equations and use them for eliminating $\mathbf{\delta}$.

In this Section, we will derive a transformation such that *we can extract a linear system of equations in the variable $\mathbf{p}$* , from (13) by eliminating the variable δ . In particular (13) can be reduced to an underdetermined linear system of equations of the form

$$\mathbf{h} = \mathbf{G}\mathbf{p} \quad (14)$$

where $\mathbf{G}$ is a fat matrix, whose size and structure depends on the array geometry. Hence, the bi-affine system of equations will indeed admit a unique solution in $\mathbf{p}$ (although, not necessarily in δ) if (14) yields a unique K -sparse solution. This transformation of the bi-affine problem into a linear problem will be shown to be possible under appropriate conditions on the array geometry, M (number of sensors), and K (number of sources).

4.1. Elimination of Variables Using Co-Array Redundancies

We derive the aforementioned transformation for two different array geometries: uniform linear array (ULA), and a robust version of coprime array (introduced later). The basic idea is to use the pattern of repeated elements in the unperturbed co-array manifold which is specified by the weight function $w(k)$ to equate certain elements of $\mathbf{R}_{\mathbf{y}}$, thereby eliminating δ .

Let R_{mn} denote the (m, n) th element of $\mathbf{R}_{\mathbf{y}}$. Given an integer $k \in S_{\text{ca}}$, we define the following notations:

$$f_k := \sum_{i=1}^{N_\theta} e^{jk\omega_i} p_i \quad (15)$$

$$\lambda_k := \sum_{i=1}^{N_\theta} e^{jk\omega_i} j\omega_i p_i \quad (16)$$

Using these notations, we can rewrite (13) as

$$R_{mn} = f_k + \lambda_k(\delta_m - \delta_n) \quad (17)$$

where $1 \leq m, n \leq M$, and $k = d_m - d_n$, $k > 0$.

Notice that f_k and λ_k are themselves linear functions of the unknown sparse vector $\mathbf{p}$. If the lag k in the co-array S_{ca} repeats at least twice, (i.e. $w(k) \geq 2$),

Figure 1: The weight function corresponding to ULA and robust-coprime array for $M = 32$, $N_1 = 4$, $N_2 = 9$.

then this redundancy can be exploited to eliminate variables as follows. If $w(k) \geq 2$, we must have $d_i - d_j = d_m - d_n = k$, for some $1 \leq i, j, m, n \leq M$. In this case, we have

$$\begin{aligned} R_{ij} &= f_k + \lambda_k(\delta_i - \delta_j) \\ R_{mn} &= f_k + \lambda_k(\delta_m - \delta_n), \end{aligned}$$

The variable f_k can be easily eliminated by subtracting these equations, leading to

$$R_{ij} - R_{mn} = \lambda_k(\delta_i - \delta_j - \delta_m + \delta_n).$$

A very similar idea can be used (with some additional computations) to eliminate δ from the M^2 equations of the form (17) for both ULA and a robust version of the co-array. Recall that co-array redundancies are also used to calibrate sensors with unknown gain and phase errors [19]. However, since we are concerned with sensor position errors, our signal model fundamentally differs from that considered in [19]. Consequently, as discussed in Sec. 2.2, we cannot exploit the co-array redundancies in the same way as done in [19]. We need to adopt a slightly more involved approach to eliminate the undesirable variable δ , the details of which depend on the geometry of the physical array.

4.2. Uniform Linear Array

In a ULA, we have $d_m - d_n = m - n$. Hence, we can rewrite the equation (17) as

$$R_{mn} = f_{m-n} + \lambda_{m-n}(\delta_m - \delta_n) \quad (18)$$

For a given $1 \leq k \leq M - 2$, define

$$\bar{r}_k = \sum_{j=2}^{k+1} R_{j,j-1} \quad (19)$$

$$\beta_k = \frac{R_{k+2,2} - R_{k+1,1}}{R_{k+2,k+1} - R_{2,1}} \quad (20)$$

The following theorem summarizes our main result for the ULA:

Theorem 1. *For a Uniform Linear Array (ULA) containing M antennas with perturbed locations, the bi-affine model (13) derived from the signal covariance matrix can be reduced to the form*

$$\mathbf{C}\mathbf{f} = \mathbf{h} \quad (21)$$

where $\mathbf{f} \in \mathbb{C}^{M-2} = [f_1, f_2, \dots, f_{M-2}]^T$ and for every $1 \leq k \leq M - 2$, the elements of $\mathbf{C} \in \mathbb{C}^{(M-3) \times (M-2)}$ and $\mathbf{h} \in \mathbb{C}^{M-2}$ are given by

$$C_{k-1,1} = -k\beta_k \quad (22)$$

$$C_{k-1,k} = 1 \quad (23)$$

$$h_{k-1} = R_{k+1,1} - \beta_k \bar{r}_k \quad (24)$$

for every $2 \leq k \leq M - 2$. The remaining elements of $\mathbf{C}$ are zero. This transformation holds for almost all $\boldsymbol{\delta} \in \mathbb{R}^M$.

Proof. The proof can be found in Appendix A. □

Remark 2. Notice that the elements of $\mathbf{f}$ are linear functions of the sparse vector $\mathbf{p}$. In particular, from (15), we have

$$\mathbf{f} = \mathbf{B}_U \mathbf{p}$$

where the elements of $\mathbf{B}_U \in \mathbb{C}^{M-2, N_\theta}$ are given by $[\mathbf{B}_U]_{m,i} = e^{jm\omega_i}$. Therefore (21) can be written as the following system of underdetermined equations (since $N_\theta \gg M$)

$$\mathbf{h} = \mathbf{C}\mathbf{B}_U\mathbf{p} \quad (25)$$

This system can admit a unique sparse solution, representing the true source powers, if the Kruskal Rank of $\mathbf{C}\mathbf{B}_U$ is at least $2K$. Since $\text{kruskal-rank}(\mathbf{C}\mathbf{B}_U) \leq M - 3$, this implies $\|\mathbf{p}\|_0 = K < (M - 3)/2$ is a sufficient condition for the true sparse $\mathbf{p}$ to be the unique solution. In practice, by exploiting the fact that $\mathbf{p}$ is non negative, a larger number of sources may be uniquely recovered. We will study the phase-transition behavior of ℓ_1 minimization algorithms to solve (25) to determine such an empirical relation between K and M .

Remark 3. The matrix $\mathbf{C}$ and the vector $\mathbf{h}$, are only functions of the elements of covariance matrix $\mathbf{R}_y$, and are not explicit functions of the unknown parameter δ . The constructive proofs in Appendices A,B demonstrate the details of eliminating the variable δ from our bi-affine equations.

4.3. Robust Coprime Array

In order to eliminate δ from the equations (13), we need to have $w(k) > 1$. As evident from Fig. 1b, that in the original coprime array, $w(k) = 1$ for some values of k , due to which we cannot apply our simplifications. Hence, we will consider an extended version of coprime array, defined as follows.

Definition 1 (Robust Coprime Array). A robust coprime array contains $M = 4N_1 + 2N_2 - 2$ sensors, whose normalized locations are given by

$$\{(i-1)N_1, 1 \leq i \leq 2N_2 + 1\} \cup \{(j-1)N_2, 1 \leq j \leq 4N_1\}$$

In other words, we extend the usual coprime array by doubling the number of sensors in each ULA. Therefore, we have $M = 4N_1 + 2N_2 - 2$ sensors in the robust coprime array. By adding these additional sensors, we are ensured that every lag between $-N_1N_2, \dots, N_1N_2$ is repeated at least twice. We will

use these repetitions to resolve the unknown perturbations. Moreover, we will discard the lags beyond the aforementioned range. In the sequel, we will denote $M'_{\text{ca}} = N_1 N_2$, representing the number of positive integers in this range.

Notations and Definitions: For the ease of notation, assume every variable with superscript $(\cdot)^{(1)}$ to be associated with the first sub-array (with spacing N_1), and every variable with superscript $(\cdot)^{(2)}$ to be associated with the second sub-array (with spacing N_2). Therefore, we have $d_i^{(1)} = (i-1)N_1$, $d_j^{(2)} = (j-1)N_2$, where $1 \leq i \leq 2N_2 + 1$, and $1 \leq j \leq 4N_1$. Moreover, let $R_{ij}^{(12)}$ denote the covariance between the received signal on the i th sensor of first ULA and j th sensor of the second ULA, and $R_{ij}^{(21)} = (R_{ij}^{(12)})^*$. Similarly, $R_{ii'}^{(1)}$ (resp. $R_{jj'}^{(2)}$) denotes the covariance between the received signal on the first (resp. second) sub-array on its i th and i' th (resp. j th and j' th) sensors. This indexing also holds for $\delta_i^{(1)}, \delta_j^{(2)}$. We also define several quantities which will be later used to state our main theorem on robust coprime array. Firstly, define the quantities

$$\bar{r}_i^{(1)} := \sum_{i'=2}^i R_{i',i'-1} \quad (26)$$

$$\bar{r}_j^{(2)} := \sum_{j'=2}^j R_{j',j'-1}. \quad (27)$$

$$\alpha := \frac{2\bar{r}_{N_1+1}^{(2)} - \bar{r}_{2N_1+1}^{(2)}}{2\bar{r}_{N_2+1}^{(1)} - \bar{r}_{2N_2+1}^{(1)}} \quad (28)$$

We also define the vectors $\beta_{cp} \in \mathbb{C}^{M'_{\text{ca}}}$ and $\mathbf{h}_{cp} \in \mathbb{C}^{M'_{\text{ca}}}$ as follows. From the properties of the difference set of a coprime array, each index k in the range $1 \leq k \leq M'_{\text{ca}}$ is necessarily of one of the following four forms:

$$k = \begin{cases} d_i^{(1)} - d_j^{(2)}, & 1 \leq i \leq N_2, 1 \leq j \leq 2N_1 \\ d_j^{(2)} - d_i^{(1)}, & 1 \leq i \leq N_2, 1 \leq j \leq 2N_1 \\ d_i^{(1)} - d_{i'}^{(1)}, & 1 \leq i, i' \leq N_2 \\ d_j^{(2)} - d_{j'}^{(2)}, & 1 \leq j, j' \leq 2N_1 \end{cases}$$

In other words, k can either be a self-difference of sensors belonging to the same sub-array, or a cross-difference of sensors belonging to different sub-arrays. If

k happens to be both a self-difference and a cross-difference, we consider k as a self-difference within its corresponding sub-array. Moreover, if k happens to be a cross-difference of type $k = d_i^{(1)} - d_j^{(2)}$ for some i, j , and also we have $k = d_{\hat{j}}^{(2)} - d_{\hat{i}}^{(1)}$ for some other $\hat{i}$ and $\hat{j}$, we consider it as a cross-difference of the former type.

The elements of β_{cp} and $\mathbf{h}_{cp}$ are then given as follows. Here, the indices i, i', j , and j' vary over the ranges $1 \leq i, i' \leq N_2, 1 \leq j, j' \leq 2N_1$, and $\bar{i} = i + N_2, \bar{j} = j + N_1$.

$$[\beta_{cp}]_k = \begin{cases} \frac{\alpha(R_{i\bar{j}}^{(12)} - R_{ij}^{(12)})}{\alpha(\bar{r}_i^{(1)} - \bar{r}_i^{(1)} - \bar{r}_{N_2+1}^{(1)}) - (\bar{r}_j^{(2)} - \bar{r}_j^{(2)} - \bar{r}_{N_1+1}^{(2)})}, & k = d_i^{(1)} - d_j^{(2)} \\ -\frac{\alpha(R_{i\bar{j}}^{(12)} - R_{ij}^{(12)})^*}{\alpha(\bar{r}_i^{(1)} - \bar{r}_i^{(1)} - \bar{r}_{N_2+1}^{(1)}) - (\bar{r}_j^{(2)} - \bar{r}_j^{(2)} - \bar{r}_{N_1+1}^{(2)})}, & k = d_j^{(2)} - d_i^{(1)} \\ \frac{R_{k+2,2}^{(1)} - R_{k+1,1}^{(1)}}{R_{k+2,k+1}^{(1)} - R_{2,1}^{(1)}} & k \neq N_2, \quad k = d_i^{(1)} - d_{i'}^{(1)} \\ \frac{R_{k+2,2}^{(2)} - R_{k+1,1}^{(2)}}{R_{k+2,k+1}^{(2)} - R_{2,1}^{(2)}} & k \neq N_1, \quad k = d_j^{(2)} - d_{j'}^{(2)} \end{cases} \quad (29)$$

$$[\mathbf{h}_{cp}]_k = \begin{cases} R_{ij}^{(12)} - \beta_k \bar{r}_i^{(1)} + \alpha^{-1} \beta_k \bar{r}_j^{(2)}, & k = d_i^{(1)} - d_j^{(2)} \\ (R_{ij}^{(12)})^* + \beta_k \bar{r}_i^{(1)} - \alpha^{-1} \beta_k \bar{r}_j^{(2)} & k = d_j^{(2)} - d_i^{(1)} \\ R_{i+1,1}^{(1)} - \beta_k \bar{r}_i^{(1)} & k \neq N_2, \quad k = d_i^{(1)} - d_{i'}^{(1)} \\ R_{j+1,1}^{(2)} - \beta_k \bar{r}_j^{(2)} & k \neq N_1, \quad k = d_j^{(2)} - d_{j'}^{(2)} \end{cases} \quad (30)$$

Furthermore, $[\mathbf{h}_{cp}]_{N_1} = [\mathbf{h}_{cp}]_{N_2} = 0$. Based upon the above definitions, let us also define a matrix $\mathbf{C}_{cp} \in \mathbb{C}^{M'_{ca} \times M'_{ca}}$ such that $[\mathbf{C}_{cp}]_{k,k} = 1, 1 \leq k \leq M'_{ca}, k \neq N_1, N_2$, and $[\mathbf{C}_{cp}]_{N_1, N_1} = [\mathbf{C}_{cp}]_{N_2, N_2} = 0$. Its remaining entries satisfy:

$$[\mathbf{C}_{cp}]_{k, N_1} = \begin{cases} -(i-1)\beta_k, & k = d_i^{(1)} - d_j^{(2)} \\ (i-1)\beta_k, & k = d_j^{(2)} - d_i^{(1)} \\ -(i-1)\beta_k, & k = d_i^{(1)} - d_{i'}^{(1)}, k \neq N_1 \end{cases} \quad (31)$$

$$[\mathbf{C}_{cp}]_{k, N_2} = \begin{cases} \alpha^{-1}(j-1)\beta_k, & k = d_i^{(1)} - d_j^{(2)} \\ -\alpha^{-1}(j-1)\beta_k, & k = d_j^{(2)} - d_i^{(1)} \\ -(j-1)\beta_k, & k = d_j^{(2)} - d_{j'}^{(2)}, k \neq N_2 \end{cases} \quad (32)$$

Equipped with the above definitions, we state our main result on robust coprime arrays as the following theorem:

Theorem 2. *For a robust coprime array, the bi-affine formulation (13) can be reduced to $\mathbf{C}_{cp}\mathbf{f}_{cp} = \mathbf{h}_{cp}$, where $\mathbf{f}_{cp} = [f_1, f_2, \dots, f_{M'_{ca}}]^T$ and $\mathbf{C}_{cp} \in \mathbb{C}^{M'_{ca} \times M'_{ca}}$, $\mathbf{h}_{cp} \in \mathbb{C}^{M'_{ca}}$ are previously defined. This transformation holds for almost all $\boldsymbol{\delta} \in \mathbb{R}^M$.*

Proof. The proof can be found in the Appendix B. $\square$

Remark 4. *Recall that $\mathbf{f}_{cp} = \mathbf{B}_{cp}\mathbf{p}$, where the elements of $\mathbf{B}_{cp} \in \mathbb{C}^{M'_{ca} \times N_\theta}$ are given by $[\mathbf{B}_{cp}]_{m,i} = e^{jm\omega_i}$. Hence, from Theorem 2, we obtain $\mathbf{C}_{cp}\mathbf{B}_{cp}\mathbf{p} = \mathbf{h}_{cp}$, which can admit a unique sparse solution in $\mathbf{p}$ if $\mathbf{C}_{cp}\mathbf{B}_{cp}$ has kruskal rank of $O(M'_{ca})$. In future, we will characterize the exact kruskal rank of $\mathbf{C}_{cp}\mathbf{B}_{cp}$. However, in Sec. 6, we experimentally show that ℓ_1 minimization can resolve larger number of sources for coprime arrays, compared to ULA.*

5. Iterative Algorithm for finite snapshots and noise

While our main results show that it is fundamentally possible to eliminate the nuisance variable $\boldsymbol{\delta}$ and solve for DOAs, they are derived under the assumption that the ideal covariance matrix $\mathbf{R}_Y$ is available. In practice however, we can only estimate $\mathbf{R}_Y$ using a finite number of snapshots. For the estimated covariance matrix, the technique for variable elimination in the proofs of Theorems 1 and 2 may not be robust (although it works perfectly for the ideal covariance matrix). This prompts us to use an iterative algorithm (derived earlier in [20] to jointly estimate $\mathbf{p}$ and $\boldsymbol{\delta}$ in presence of finite snapshots. In order to recover $\boldsymbol{\delta}$, we need to assume that one of sensors locations is exactly known. This is because of the fact that the perturbed coarray manifold only depends on the pairwise differences of the sensor locations, i.e., adding a fixed bias to all the sensor locations would not change the $\mathbf{A}(\boldsymbol{\delta})$ and $\mathbf{z}$. Hence, without loss of generality, we assume $\delta_1 = 0$.

In [20], we proposed an iterative approach based on alternating minimization which is summarized in Fig. 2. The minimization with respect to $\mathbf{p}$ is performed

```

1:  $\boldsymbol{\delta}^{(0)} \leftarrow \mathbf{0}, i \leftarrow 0$ 
2: repeat
3:    $\mathbf{p}^{(i)} \leftarrow \arg \min_{\mathbf{p}} \left\{ \|\mathbf{A}_{\text{ca}}(\boldsymbol{\delta}^{(i)})\mathbf{p} - \mathbf{z}\|_F^2 \right\}$  s.t.  $\|\mathbf{p}\|_0 \leq K$ 
4:    $\boldsymbol{\delta}^{(i+1)} \leftarrow \arg \min_{\boldsymbol{\delta}} \left\{ \|\mathbf{A}_{\text{ca}}(\boldsymbol{\delta})\mathbf{p}^{(i)} - \mathbf{z}\|_F^2 \right\}$ 
5:    $i \leftarrow i + 1$ 
6: until Convergence
7:  $\hat{\boldsymbol{\delta}} \leftarrow \boldsymbol{\delta}^{(i)}, \hat{\mathbf{p}} \leftarrow \mathbf{p}^{(i)}$ 

```

Figure 2: Algorithm for jointly estimating the perturbations and the source directions

by solving a sparse recovery problem which could be done using LASSO or OMP. In order to update the vector $\boldsymbol{\delta}$, in the second stage of the iterative algorithm, an error function defined as $E(\boldsymbol{\delta}) = \|\mathbf{A}_{\text{ca}}(\boldsymbol{\delta})\mathbf{p} - \mathbf{z}\|_F^2$ is minimized through running T iterations of gradient descent

$$\boldsymbol{\delta} \leftarrow \boldsymbol{\delta} - \mu \frac{\partial E(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}}, \quad (33)$$

in which $\mu > 0$ is the step size parameter and the m th component of $\frac{\partial E(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}}$ for $m \geq 2$ is obtained by

$$\frac{\partial E(\boldsymbol{\delta})}{\partial \delta_m} = \text{tr} \left((-2\mathbf{z}\mathbf{p}^T + 2\mathbf{A}_{\text{ca}}(\mathbf{0})\mathbf{p}\mathbf{p}^T) \frac{\partial \mathbf{P}_{\text{ca}}(\boldsymbol{\delta})}{\partial \delta_m} \right), \quad (34)$$

and $\frac{\partial E(\boldsymbol{\delta})}{\partial \delta_1} = 0$, where $\mathbf{P}_{\text{ca}}(\boldsymbol{\delta})$ is given by $\mathbf{P}^*(\boldsymbol{\omega}, \boldsymbol{\delta}) \odot \mathbf{P}(\boldsymbol{\omega}, \boldsymbol{\delta})$ for $\boldsymbol{\omega}$ evaluated on the grid points.

Remark 5. In our earlier paper [20], we have shown that the algorithm summarized in Fig. 2 is guaranteed to converge to a local minimum. We showed that the convergence is subject to success of both the LASSO and the gradient descent stages. For the LASSO stage to succeed, the sparsity of $\mathbf{p}$ should be high enough so that the recovering a unique sparse solution be guaranteed. For the gradient descent stage to succeed, we need to have $\mu < \frac{2}{C}$ where

$$C = 2\pi M \sqrt{2M} \| -\mathbf{z}\mathbf{p}^T + \mathbf{A}_{\text{ca}}(\mathbf{0})\mathbf{p}\mathbf{p}^T \|.$$

6. Simulations

In this section, we conduct three different sets of numerical experiments to validate our theoretical claims. In all the simulations, we assume $N_\theta = 200$ points on the grid. The DOAs are chosen uniformly between -60° and 60° , and assigned to the closest point on the grid. The perturbations are assumed to be $\boldsymbol{\delta} = \alpha \boldsymbol{\delta}_0$ (Notice that, following the model given in Section 2, the sensor locations and the perturbations are normalized with respect to half of the wavelength $\lambda/2$). In the first and second sets of simulations, $\boldsymbol{\delta}_0$ is a fixed vector with $|\boldsymbol{\delta}|_\infty \leq 0.5$, and is drawn from a uniform distribution. However, for the third set of simulations, we choose $\boldsymbol{\delta}_0$ randomly in each trial.

In the first set of simulations, we assume that we know the covariance matrix $\mathbf{R}_y$, and that the model defined in Sec. 3 holds exactly. In this case, we use the approach proposed in the proofs of Theorems 1, and 2 to eliminate the perturbations and recover the source powers. We compare our method against running ℓ_1 minimization on the covariance matrix, assuming that the coarray manifold is unperturbed (which will lead to a basis mismatch). In other words, we compare our approach with the solution of the following problem:

$$\min_{\mathbf{p}} \|\mathbf{p}\|_1 \quad \text{s.t.} \quad \mathbf{A}_{\text{ca},0}^u \mathbf{p} = \mathbf{z}^u$$

As demonstrated in Fig. 3, our approach as described in the proofs of Theorems 1 and 2, exactly recovers the true supports for both ULA and coprime arrays when $K = 10 < M = 32$ (The blue line corresponding to the true solution and green line corresponding to our method, match exactly). When $K = 35 > M$, ULA cannot recover the true DOAs, while the robust coprime array perfectly identifies the support (compare Fig. 3f and Fig. 3h, where $\alpha = 0.5$). In the third experiment, we will empirically study the relationship between K and M (for both ULA and coprime) that ensures perfect recovery of DOAs in the form of a phase transition diagram.

In the second set of simulations, we use an estimated covariance matrix using a finite number of snapshots. In our simulations, we consider both ULA and

Figure 3: Recovered powers using the approach proposed in Theorems 1, 2. In each plot, X-axis shows the directions on the grid in degrees, and Y-axis shows the power corresponding to each direction on the grid.

robust coprime array geometries. For the robust coprime array with $M = 16$, we pick $N_1 = 2, N_2 = 5$, and for $M = 32$ we choose $N_1 = 4, N_2 = 9$, which correspond to Fig. 4a, and Fig. 4b, respectively. We use the algorithm described in Sec. 5, which is labeled as “Calibration” in the figure. “LASSO” refers to naively running LASSO on the vectorized covariance matrix assuming an unperturbed dictionary, i.e., solving the following problem:

$$\min_{\mathbf{p}} \|\mathbf{p}\|_1 + \lambda \|\mathbf{A}_{\text{ca},0}^u \mathbf{p} - \mathbf{z}^u\|_2^2$$

in which λ is a regularization parameter. For each value of α , we show the average value of root-mean-square error (RMSE) over $N_{\text{tests}} = 100$ Monte-Carlo runs, where $\text{RMSE} \triangleq \sqrt{\sum_{i=1}^{N_{\text{tests}}} \frac{\|\hat{\mathbf{p}} - \mathbf{p}\|^2}{N_{\text{tests}}}}$, and $\hat{\mathbf{p}}$ is the recovered vector of source powers. We observe that proposed iterative algorithm can recover source powers and is able to resolve the perturbations. We noticed in our simulations that the proposed approach (calibration) may sometimes perform worse than the simple LASSO (for instance, when $\alpha > 0.7$ in Fig. 4b). We observed this happens because the iterative algorithm finds a local optimum instead of the global optimum, thereby failing to recover the true source powers. A theoretical analysis of this phenomenon will be a topic of future research.

In the third set of experiments, we study the phase transition diagram of ℓ_1 minimization algorithm applied on the linear underdetermined system of equations obtained from Theorems 1, 2. In these simulations, we assume that the covariance matrix is known exactly, i.e., we have infinite number of snapshots. We consider a trial successful if $\|\mathbf{p} - \hat{\mathbf{p}}\|_F \leq \epsilon$, where $\hat{\mathbf{p}}$ is the recovered vector of powers, and $\epsilon = 10^{-3}$. The white pixels in the plots of Fig. 5 show the problem settings under which performing ℓ_1 minimization on the linear system derived in Theorems 1, and 2 can always recover the true solution. We simulate each case 100 times and show the probability of success with a gray-scale pixel. In these plots, we only show the cases where M is an even number (because the robust-coprime array can only have even number of sensors). Moreover, for each M , we find N_1, N_2 such that $M = 4N_1 + 2N_2 - 2$ and $N_1 N_2$ is the maximum possible number. For coprime arrays, we find that even in presence of pertur-

(a) $M = 16$

(b) $M = 32$

Figure 4: RMSE of the recovered sources vs perturbations for different arrays and different number of sensors and sources.

bation, white pixels exist in the region where number of sources is greater than number of sensors. This shows that robust coprime array is capable of resolving more sources than the number of sensors, even in the presence of perturbations.

7. Conclusion

In this paper, we investigated the robustness of coprime arrays to unknown perturbations on the locations of sensors. We assumed that the perturbations are small and developed a bi-affine model in terms of the unknown perturbations and the source powers. We used the redundancies of the difference coarray to eliminate the nuisance variables, and reduce the bi-affine problem to a linear underdetermined (sparse) problem in source powers, which can be solved using ℓ_1 minimization. We derived this reduction for both ULA and a robust version of coprime arrays. Our simulations showed that if the ideal covariance matrix of the received signals is available, the source powers can be accurately recovered using our proposed approach, thereby validating the theoretical claims in Theorems 1 and 2. We also showed the region (in terms of K and M) under which the bi-affine problem has a unique solution, in the form of a phase transition diagram. When only a finite number of snapshots are available, we proposed an iterative algorithm to jointly recover perturbations and source powers, which shows satisfactory performance in our simulations.

Acknowledgments

This work was supported in parts by the University of Maryland, College Park, and the Department of Defense.

(a) ULA

(b) Coprime

Figure 5: Phase transition plots

Appendix A. Proof of Theorem 1

Proof. Let k, i_k be integers such that $1 \leq k \leq M - 2$ and $1 \leq i_k \leq M - k - 1$.

We have

$$R_{i_k+k+1, i_k+k} = f_1 + \lambda_1(\delta_{i_k+k+1} - \delta_{i_k+k}) \quad (\text{A.1})$$

$$R_{i_k+1, i_k} = f_1 + \lambda_1(\delta_{i_k+1} - \delta_{i_k}) \quad (\text{A.2})$$

Similarly,

$$R_{i_k+k+1, i_k+1} = f_k + \lambda_k(\delta_{i_k+k+1} - \delta_{i_k+1}) \quad (\text{A.3})$$

$$R_{i_k+k, i_k} = f_k + \lambda_k(\delta_{i_k+k} - \delta_{i_k}) \quad (\text{A.4})$$

Subtracting (A.1) from (A.2), and also (A.3) from (A.4) we obtain

$$\beta_k = \frac{R_{i_k+k+1, i_k+1} - R_{i_k+k, i_k}}{R_{i_k+k+1, i_k+k} - R_{i_k+1, i_k}} = \frac{\lambda_k}{\lambda_1} \quad (\text{A.5})$$

Here, we assumed that $\delta \in \mathbb{R}^M$ is such that

$$\delta_{i_k+k+1} - \delta_{i_k+k} - \delta_{i_k+1} + \delta_{i_k} \neq 0, \quad (\text{A.6})$$

This can be violated only on a set of measure zero in $\mathbb{R}^M$. Hence, the following results will hold for *almost all* δ .

Notice that for a fixed k , different instances of equation (A.5) corresponding to different values of i_k in the range $1 \leq i_k \leq M - k - 1$ are actually identical, as long as the observed covariance matrix is exact, and the assumption (A.6) holds for all i_k . Hence, in the sequel, we will assume $i_k = 1$.

From and (18) and (19), we can verify that

$$\bar{r}_k = kf_1 + \lambda_1(\delta_{k+1} - \delta_1) \quad (\text{A.7})$$

From (A.4), (A.7) we get, for $2 \leq k \leq M - 2$,

$$\beta_k (kf_1 - \bar{r}_k) = f_k - R_{k+1,1}$$

which can be expressed as

$$C_{k-1,1}f_1 + C_{k-1,k}f_k = h_{k-1}, \quad 2 \leq k \leq M - 2 \quad (\text{A.8})$$

where $C_{k-1,1}$, $C_{k-1,k}$, and h_{k-1} are given in (22), (23), and (24). We can express (A.8) in a more compact and explicit form as

$$\mathbf{C}\mathbf{f} = \mathbf{h} \quad (\text{A.9})$$

in which

$$\mathbf{C} = \begin{bmatrix} C_{1,1} & C_{1,2} & 0 & \cdots & 0 \\ C_{2,1} & 0 & C_{2,3} & \cdots & 0 \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ C_{M-3,1} & 0 & 0 & \cdots & C_{M-3,M-2} \end{bmatrix}$$

$$\mathbf{h} = [h_1 \ h_2 \ \cdots \ h_{M-3}]^T$$

□

Appendix B. Proof of Theorem 2

Proof. It can be easily verified that

$$\bar{r}_i^{(1)} = (i-1)f_{N_1} + (\delta_i^{(1)} - \delta_1^{(1)})\lambda_{N_1} \quad (\text{B.1})$$

$$\bar{r}_j^{(2)} = (j-1)f_{N_2} + (\delta_j^{(2)} - \delta_1^{(2)})\lambda_{N_2} \quad (\text{B.2})$$

For each lag k , one of the following four possibilities can happen: 1) $k = d_i^{(1)} - d_j^{(2)}$, 2) $k = d_j^{(2)} - d_i^{(1)}$, 3) $k = d_i^{(1)} - d_{i'}^{(1)}$, 4) $k = d_j^{(2)} - d_{j'}^{(2)}$.

1. $k = d_i^{(1)} - d_j^{(2)}$: We consider the case where the lag k is generated taking the difference between the i th sensor from the first sub-array and the j th sensor from the second sub-array.

Since, we have doubled the number of sensors of each ULA, for each $1 \leq i \leq N_2$, and $1 \leq j \leq 2N_1$ so that $d_i^{(1)} - d_j^{(2)} = N_1i - N_2j = k$, the sensors indexed by $\bar{i} = i + N_2, \bar{j} = j + N_1$ also create the same lag k . Therefore, we can rewrite the equations (B.1) and (B.2) for $\bar{i}, \bar{j}$. Subtracting those equations from (B.1) and (B.2), we get

$$\bar{r}_{\bar{i}}^{(1)} - \bar{r}_i^{(1)} = N_2f_{N_1} + (\delta_{\bar{i}}^{(1)} - \delta_i^{(1)})\lambda_{N_1} \quad (\text{B.3})$$

$$\bar{r}_{\bar{j}}^{(2)} - \bar{r}_j^{(2)} = N_1f_{N_2} + (\delta_{\bar{j}}^{(2)} - \delta_j^{(2)})\lambda_{N_2} \quad (\text{B.4})$$

We also know that

$$R_{ij}^{(12)} = f_k + (\delta_i^{(1)} - \delta_j^{(2)})\lambda_k \quad (\text{B.5})$$

$$R_{i\bar{j}}^{(12)} = f_k + (\delta_i^{(1)} - \delta_{\bar{j}}^{(2)})\lambda_k \quad (\text{B.6})$$

We know that the $(N_2 + 1)$ th element of the first ULA, and the $(N_1 + 1)$ th element of the second ULA happen to be the same sensor on the coprime array (both at the normalized location $N_1 N_2$). For this particular sensor, we can write equations (B.1) and (B.2).

$$\bar{r}_{N_2+1}^{(1)} = N_2 f_{N_1} + (\delta_{N_2+1}^{(1)} - \delta_1^{(1)})\lambda_{N_1} \quad (\text{B.7})$$

$$\bar{r}_{N_1+1}^{(2)} = N_1 f_{N_2} + (\delta_{N_1+1}^{(2)} - \delta_1^{(2)})\lambda_{N_2} \quad (\text{B.8})$$

where $\delta_{N_2+1}^{(1)} = \delta_{N_1+1}^{(2)}$ since they are the same sensor, and also $\delta_1^{(1)} = \delta_1^{(2)}$ for the same reason.

We can write similar equations for $(2N_2 + 1)$ th sensor of first ULA, and $(2N_1 + 1)$ th sensor of second ULA, which again happen to be the same sensor.

$$\bar{r}_{2N_2+1}^{(1)} = 2N_2 f_{N_1} + (\delta_{2N_2+1}^{(1)} - \delta_1^{(1)})\lambda_{N_1} \quad (\text{B.9})$$

$$\bar{r}_{2N_1+1}^{(2)} = 2N_1 f_{N_2} + (\delta_{2N_1+1}^{(2)} - \delta_1^{(2)})\lambda_{N_2}, \quad (\text{B.10})$$

in which $\delta_{2N_2+1}^{(1)} = \delta_{2N_1+1}^{(2)}$.

From (B.7), (B.8), (B.9), (B.10), we get

$$\frac{2\bar{r}_{N_1+1}^{(2)} - \bar{r}_{2N_1+1}^{(2)}}{\lambda_{N_2}} = \frac{2\bar{r}_{N_2+1}^{(1)} - \bar{r}_{2N_2+1}^{(1)}}{\lambda_{N_1}} \quad (\text{B.11})$$

and from equation, we obtain (28)

$$\alpha = \frac{\lambda_{N_2}}{\lambda_{N_1}} = \frac{2\bar{r}_{N_1+1}^{(2)} - \bar{r}_{2N_1+1}^{(2)}}{2\bar{r}_{N_2+1}^{(1)} - \bar{r}_{2N_2+1}^{(1)}}$$

From equations (B.5), (B.6) from (B.3), (B.4), we get

$$\begin{aligned}\delta_i^{(1)} - \delta_j^{(2)} - \delta_i^{(1)} + \delta_j^{(2)} &= \frac{\bar{r}_i^{(1)} - \bar{r}_i^{(1)} - N_2 f_{N_1}}{\lambda_{N_1}} \\ &\quad - \frac{\bar{r}_j^{(2)} - \bar{r}_j^{(2)} - N_1 f_{N_2}}{\lambda_{N_2}} \\ &= \frac{R_{ij}^{(12)} - R_{ij}^{(12)}}{\lambda_k},\end{aligned}$$

whereby (29) follows, where $[\boldsymbol{\beta}_{cp}]_k := \frac{\lambda_k}{\lambda_{N_1}}$.

Now, using (B.1), (B.2), we can write

$$\delta_i^{(1)} - \delta_j^{(2)} = \frac{\bar{r}_i^{(1)} - (i-1)f_{N_1}}{\lambda_{N_1}} - \frac{\bar{r}_j^{(2)} - (j-1)f_{N_2}}{\alpha\lambda_{N_1}}$$

and

$$R_{ij}^{(12)} = f_k + \beta_k(\bar{r}_i^{(1)} - (i-1)f_{N_1} - (\bar{r}_j^{(2)} - (j-1)f_{N_2})\alpha^{-1}) \quad (\text{B.12})$$

which is linear in terms of elements of $\mathbf{f}$, and hence it is linear in terms of $\mathbf{p}$.

By varying the indices i and j in the range $1 \leq i \leq N_2$ and $1 \leq j \leq 2N_1$ in (B.12), we obtain the corresponding rows of the system of equations

$$\mathbf{C}_{cp}\mathbf{f} = \mathbf{h}_{cp}.$$

2. $k = d_j^{(2)} - d_i^{(1)}$:

In this case, (B.5), (B.6) should be rewritten as

$$(R_{ij}^{(12)})^* = R_{ji}^{(21)} = f_k + (\delta_j^{(2)} - \delta_i^{(1)})\lambda_k \quad (\text{B.13})$$

$$(R_{ij}^{(12)})^* = R_{ji}^{(21)} = f_k + (\delta_j^{(2)} - \delta_i^{(1)})\lambda_k \quad (\text{B.14})$$

We can repeat the math accordingly to get

$$\begin{aligned}(R_{ij}^{(12)})^* &= f_k + \beta_k((\bar{r}_j^{(2)} - (j-1)f_{N_2})\alpha^{-1} \\ &\quad - (\bar{r}_i^{(1)} - (i-1)f_{N_1}))\end{aligned}$$

3. $k = d_i^{(1)} - d_{i'}^{(1)}$ and $k = d_j^{(2)} - d_{j'}^{(2)}$: These cases can be handled similar to ULA, following the same lines of math as in the proof of Theorem 1.

□

References

- [1] P. P. Vaidyanathan, P. Pal, Sparse sensing with co-prime samplers and arrays, *Signal Processing, IEEE Transactions on* 59 (2) (2011) 573–586.
- [2] P. Pal, P. Vaidyanathan, Nested arrays: a novel approach to array processing with enhanced degrees of freedom 58 (8) (2010) 4167–4181.
- [3] R. T. Hoor, S. Kassam, et al., The unifying role of the coarray in aperture synthesis for coherent and incoherent imaging, *Proceedings of the IEEE* 78 (4) (1990) 735–752.
- [4] P. Pal, P. Vaidyanathan, Coprime sampling and the music algorithm, in: *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE)*, 2011 IEEE, IEEE, 2011, pp. 289–294.
- [5] Y. D. Zhang, M. G. Amin, B. Himed, Sparsity-based doa estimation using co-prime arrays, in: *Acoustics, Speech and Signal Processing (ICASSP)*, 2013 IEEE International Conference on, IEEE, 2013, pp. 3967–3971.
- [6] P. Pal, P. Vaidyanathan, Correlation-aware techniques for sparse support recovery, in: *Statistical Signal Processing Workshop (SSP)*, 2012 IEEE, IEEE, 2012, pp. 53–56.
- [7] P. Pal, P. Vaidyanathan, On application of lasso for sparse support recovery with imperfect correlation awareness, in: *Signals, Systems and Computers (ASILOMAR)*, 2012 Conference Record of the Forty Sixth Asilomar Conference on, IEEE, 2012, pp. 958–962.
- [8] P. Pal, P. Vaidyanathan, Correlation-aware sparse support recovery: Gaussian sources, in: *Acoustics, Speech and Signal Processing (ICASSP)*, 2013 IEEE International Conference on, IEEE, 2013, pp. 5880–5884.
- [9] L. Seymour, C. Cowan, P. Grant, Bearing estimation in the presence of sensor positioning errors, in: *Acoustics, Speech, and Signal Processing*,

- IEEE International Conference on ICASSP'87., Vol. 12, IEEE, 1987, pp. 2264–2267.
- [10] Y.-M. Chen, J.-H. Lee, C.-C. Yeh, J. Mar, Bearing estimation without calibration for randomly perturbed arrays, *Signal Processing, IEEE Transactions on* 39 (1) (1991) 194–197.
 - [11] A. J. Weiss, B. Friedlander, Array shape calibration using sources in unknown locations-a maximum likelihood approach, *Acoustics, Speech and Signal Processing, IEEE Transactions on* 37 (12) (1989) 1958–1966.
 - [12] A. Paulraj, T. Kailath, Direction of arrival estimation by eigenstructure methods with unknown sensor gain and phase, in: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85.*, Vol. 10, IEEE, 1985, pp. 640–643.
 - [13] M. Viberg, A. L. Swindlehurst, A bayesian approach to auto-calibration for parametric array signal processing, *Signal Processing, IEEE Transactions on* 42 (12) (1994) 3495–3507.
 - [14] Z.-M. Liu, Y.-Y. Zhou, A unified framework and sparse bayesian perspective for direction-of-arrival estimation in the presence of array imperfections, *Signal Processing, IEEE Transactions on* 61 (15) (2013) 3786–3798.
 - [15] Y. Li, K. Lee, Y. Bresler, A unified framework for identifiability analysis in bilinear inverse problems with applications to subspace and sparsity models, *arXiv preprint arXiv:1501.06120*.
 - [16] E. J. Candes, T. Strohmer, V. Voroninski, Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming, *Communications on Pure and Applied Mathematics* 66 (8) (2013) 1241–1274.
 - [17] A. Ahmed, B. Recht, J. Romberg, Blind deconvolution using convex programming, *Information Theory, IEEE Transactions on* 60 (3) (2014) 1711–1732.

- [18] S. Ling, T. Strohmer, Self-calibration and biconvex compressive sensing, arXiv preprint arXiv:1501.06864.
- [19] K. Han, P. Yang, A. Nehorai, Calibrating nested sensor arrays with model errors.
- [20] A. Koochakzadeh, P. Pal, Sparse source localization in presence of co-array perturbations, SampTA, 2015.