Array Design Based on the Worst-Case Cramér-Rao Bound to Account for Multiple Targets

Kokke, Costas A.; Coutino, Mario; Heusdens, Richard; Leus, Geert

DOI
10.1109/IEEECONF59524.2023.10476966

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 57th Asilomar Conference on Signals, Systems, and Computers

Array Design Based on the Worst-Case Cramér-Rao Bound to Account for Multiple Targets

Costas A. Kokke[‡], Mario Coutino[*], Richard Heusdens[†‡], Geert Leus[‡]
[*]Radar Technology, Netherlands Organisation for Applied Scientific Research, The Hague, The Netherlands
[†]Netherlands Defence Academy, Den Helder, The Netherlands
[‡]Signal Processing Systems, Delft University of Technology, Delft, The Netherlands

2023 57th Asilomar Conference on Signals, Systems, and Computers | 979-8-3503-2574-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/IEEECONF59524.2023.10476966

***Abstract*—Sensor selection is a useful method to help reduce computational, hardware, and power requirements while maintaining acceptable performance. Although minimizing the Cramér-Rao bound has been adopted previously for sparse sensing, it did not consider multiple targets and unknown target directions. We propose to tackle the sensor selection problem for direction of arrival estimation using the worst-case Cramér-Rao bound of two uncorrelated equal power sources on planar arrays. We cast the problem as a convex semi-definite program and retrieve the binary selection by randomized rounding. We illustrate the proposed method through numerical examples related to planar arrays. We show that our method selects a combination of edge and center elements, which contrasts with solutions obtained by minimizing the single-target Cramér-Rao bound.**

***Index Terms*—array processing, Cramér-Rao bound, multi-target estimation, sensor selection, sparse sensing**

I. Introduction

One of the main functions of a radar system is angle of arrival (AoA) estimation. To realize the acquisition of spatial data and the application of beamforming, modern radars employ antenna arrays. While the aperture of the array is the main contributor to angular resolution, its density controls the suppression outside of the main beam and when sufficient prevents spatial aliasing.

It should come as no surprise then, that having an array with both a large aperture and a large density is ideal. Both the aperture and the density however, come with a cost. The cost increase can be in terms of hardware, amount of space required, and the amount of throughput and processing that is required. Additionally, the increased need for hardware and processing increases the amount of power required to drive the system.

As a result, several works have focused on answering the question: *is it possible to design an array with a reduced number of sensors while preventing spatial aliasing and achieving sufficient suppression of out-of-beam emitters and reflectors?* One effective method of reducing the amount of data processing while maintaining acceptable performance is compressed sensing [1]–[3]. While CS methods indeed reduce the amount of data that needs to be processed, it does not relieve us fully of the hardware requirements. Sparse array design and sensor selection however can help us reduce more hardware.

Sensor selection as a means of designing a sparse array uses the idea of a candidate set of sensors. From this candidate set of sensors, we would like to select a subset that performs better than any other subset of the same size, which is determined using a metric or cost of choice. The candidate set of sensors can be physically present or not. When they are physically present, one can think about a scenario where there are many antennas, which are cheap, but not so many frontends, which are expensive. The frontends can then be switched to different antennas based on the sensor selection procedure results. If the sensors in the candidate set are not all physically present, sensor selection is similar to the sensor placement problem, where the space in which the sensors can be placed is discretized.

To perform the sensor selection, convex optimization has been used to some success, for example by relaxing a non-convex problem to a convex one [4]–[6]. Convex optimization has the immediate benefits of having well-studied optimization methods, including many off-the-shelf solvers, as well as guarantees of reaching a global optimum. In this work, we also take the approach of formulating a relaxed convex program as part of our proposed method.

To evaluate the quality of the selected subset of sensors, we need a metric. For this work we choose the Cramér-Rao bound (CRB) as it is agnostic to the estimation method used, making it especially relevant for sensor selection during the array design phase. Some works have already used the CRB for sensor selection [7]–[10], but planar arrays with multiple unknown targets have not yet been considered. This is what we will discuss in this work.

In Section II we present the signal model, its associated multi-target CRB and discuss the specific selection that we would like to find using our method. The proposed cost function and method are presented in Sections III and IV, where we also show the derivations to obtain the final convex semidefinite program. We present the results of our simulations as verification of our proposed method in Section V and we conclude in Section VI with some discussion.

II. Signal Model and Problem Statement

We start from a candidate set of array elements, from which we will select a subset of elements that performs better than

The work is part of a project funded by the Netherlands Organisation for Applied Scientific Research (TNO) and the Netherlands Defence Academy (NLDA).

any other subset of the same size according to our chosen cost function. The signal model for the full array, i.e., the candidate set, is given by

$$\begin{aligned}\boldsymbol{X} &= \sum_{k=1}^{K} \boldsymbol{a}(\boldsymbol{\omega}_k)\boldsymbol{s}_k^{\mathrm{T}} + \boldsymbol{E}\,,\\ &= \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{S}^{\mathrm{T}} + \boldsymbol{E} \in \mathbb{C}^{N\times T}\,,\end{aligned}$$

where

$$\begin{aligned}[\boldsymbol{a}(\boldsymbol{\omega}_k)]_n &= \mathrm{e}^{\mathrm{j}\boldsymbol{r}_n^{\mathrm{T}}\boldsymbol{\omega}_k}\,,\\ \boldsymbol{A}(\boldsymbol{\omega}) &= \begin{bmatrix}\boldsymbol{a}(\boldsymbol{\omega}_1) & \cdots & \boldsymbol{a}(\boldsymbol{\omega}_K)\end{bmatrix} \in \mathbb{C}^{N\times K}\,,\\ \boldsymbol{S} &= \begin{bmatrix}\boldsymbol{s}_1 & \cdots & \boldsymbol{s}_K\end{bmatrix} \in \mathbb{C}^{T\times K}\,,\\ \boldsymbol{\omega} &= \begin{bmatrix}\boldsymbol{\omega}_1^{\mathrm{T}} & \cdots & \boldsymbol{\omega}_K^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in [0,2\pi)^{DK\times 1}\,,\\ \mathrm{vec}(\boldsymbol{E}) &\sim \mathcal{CN}\big(\boldsymbol{0},\sigma^2\boldsymbol{I}\big),\end{aligned}$$

with $\boldsymbol{\omega}_k$ the D-dimensional target angle of the kth source (note that the actual AoAs are a function of this), $\boldsymbol{r}_n$ the D-dimensional position of the nth array element, and D the number of dimensions of the array, i.e., $D = 1$ for linear arrays and $D = 2$ for planar arrays. For the remainder, we assume that the matrix $\boldsymbol{R} = \boldsymbol{S}^{\mathrm{T}}\boldsymbol{S}/T$ is diagonal (orthogonal sources), which is for instance a fair assumption if T is large and the sources are independent.

Selecting M array elements from the full candidate set can be expressed as $\boldsymbol{Y} = \boldsymbol{\Phi}_{\boldsymbol{p}}\boldsymbol{X} \in \mathbb{C}^{M\times T}$, where

$$\boldsymbol{\Phi}_{\boldsymbol{p}} \in \{0,1\}^{M\times N}\,,\quad \boldsymbol{\Phi}_{\boldsymbol{p}}^{\mathrm{T}}\boldsymbol{\Phi}_{\boldsymbol{p}} = \mathrm{diag}(\boldsymbol{p})\,,\quad \boldsymbol{\Phi}_{\boldsymbol{p}}\boldsymbol{\Phi}_{\boldsymbol{p}}^{\mathrm{T}} = \boldsymbol{I}\,,$$

and $\boldsymbol{p} = \begin{bmatrix}p_1 & \cdots & p_N\end{bmatrix}^{\mathrm{T}} \in \{0,1\}^N$ is a binary selection vector where the M ones indicate the corresponding selected array elements, and zeros indicate the elements that are not selected.

Finding the optimal binary $\boldsymbol{p}$ for the array element selection problem can in general be expressed as

$$\min_{\boldsymbol{p}}\ f(\boldsymbol{p})\,,\quad \text{s.t.}\ \ \boldsymbol{1}^{\mathrm{T}}\boldsymbol{p} = M\,,\ \boldsymbol{p} \in \{0,1\}^N\,, \tag{1}$$

where $f(\boldsymbol{p})$ is the considered cost function.

III. The Worst-Case Multi-Target Cramér-Rao Bound

The CRB seems a logical choice for a cost function, since it serves as an indication of the theoretically achievable performance of an estimator. So we need a selection dependent CRB for multi-target AoA estimation. Let

$$\boldsymbol{D}_k = \frac{\partial \boldsymbol{a}(\boldsymbol{\omega}_k)}{\partial \boldsymbol{\omega}_k} \in \mathbb{C}^{N\times D}\,, \tag{2}$$

$$\boldsymbol{D} = \begin{bmatrix}\boldsymbol{D}(\boldsymbol{\omega}_1) & \cdots & \boldsymbol{D}(\boldsymbol{\omega}_K)\end{bmatrix} \in \mathbb{C}^{N\times DK}\,. \tag{3}$$

For notational convenience, let $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{\omega})$, and $\boldsymbol{P} = \mathrm{diag}(\boldsymbol{p})$. For the derivation of the CRB, we take $\boldsymbol{\theta} = \begin{bmatrix}\mathrm{Re}\{\boldsymbol{s}^{\mathrm{T}}\} & \mathrm{Im}\{\boldsymbol{s}^{\mathrm{T}}\} & \boldsymbol{\omega}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, for our unknown parameter vector, where $\boldsymbol{s} = \mathrm{vec}(\boldsymbol{S})$. Note that the parameter σ^2 is also unknown, but it is known that it is uncorrelated to the parameters in $\boldsymbol{\theta}$ [11], [12], so it does not need to be considered here.

By combining the results from [10]–[12], the $DK \times DK$ block of the selection dependent CRB matrix corresponding to $\boldsymbol{\omega}$, is given by

$$\begin{aligned}&\mathrm{CRB}_{\boldsymbol{\omega}\boldsymbol{\omega}}^{-1}(\boldsymbol{\omega},\boldsymbol{p}) =\\ &\quad \frac{2T}{\sigma^2}\,\mathrm{Re}\Big\{\boldsymbol{D}^{\mathrm{H}}\boldsymbol{P}\Big(\boldsymbol{I} - \boldsymbol{A}\big(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{P}\boldsymbol{A}\big)^{-1}\boldsymbol{A}^{\mathrm{H}}\Big)\boldsymbol{P}\boldsymbol{D} \circ \tilde{\boldsymbol{R}}\Big\}\,,\end{aligned}$$

where $\tilde{\boldsymbol{R}} = \boldsymbol{R} \otimes \boldsymbol{J}_D$, and $\boldsymbol{J}_D$ is a $D \times D$ matrix of ones. Because each target angle is in general given by a length-D vector $\boldsymbol{\omega}_k$, the CRB for one of the targets is given by a $D \times D$ matrix. The selection dependent CRB matrix for the angle of arrival of one of the targets is then proportional to

$$\begin{aligned}&\mathrm{CRB}_{\boldsymbol{\omega}_k\boldsymbol{\omega}_k}^{-1}(\boldsymbol{\omega},\boldsymbol{p}) \propto \boldsymbol{F}_k(\boldsymbol{\omega},\boldsymbol{p}) =\\ &\quad \mathrm{Re}\Big\{\boldsymbol{D}_k^{\mathrm{H}}\boldsymbol{P}\Big(\boldsymbol{I} - \boldsymbol{A}\big(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{P}\boldsymbol{A}\big)^{-1}\boldsymbol{A}^{\mathrm{H}}\Big)\boldsymbol{P}\boldsymbol{D}_k\Big\}\,.\end{aligned} \tag{4}$$

To obtain a scalar objective, we propose to use

$$f(\boldsymbol{p}) = \max_{k,\boldsymbol{\omega}}\ \mathrm{tr}\big(\boldsymbol{F}_k^{-1}(\boldsymbol{\omega},\boldsymbol{p})\big)\,, \tag{5}$$

which will optimize the worst performing target and worst performing set of target angles.

IV. Array Element Selection Method

To evaluate (5), we need the number of targets and their AoAs, which are in general not available. To address this issue, we took inspiration from a widely-used idea in waveform, filter and beamformer design in unknown multi-target scenarios: peak sidelobe (PSL) suppression.

When observing a specific filter response when the beam is aimed in a particular direction, we observe sidelobes in the directions outside of the beam. These sidelobes can change when the beam is aimed in different directions and therefore, if we want to find the PSL level, we should consider all mainlobe directions. The PSL can be considered as the maximum interference that one target can inflict upon another, a worst-case two-target metric. For a given array element selection $\boldsymbol{p}$, we can formulate finding the PSL as

$$\mathrm{PSL} = \max_{\boldsymbol{\omega}\in[0,2\pi)^{2D\times 1}}\ \mathrm{SL}(\boldsymbol{\omega},\boldsymbol{p})\,, \tag{6}$$

where $\mathrm{SL}(\boldsymbol{\omega},\boldsymbol{p})$ is the sidelobe level at $\boldsymbol{\omega}_2$ when the beam is aimed at $\boldsymbol{\omega}_1$. Note that the PSL implicitly assumes two equi-power targets. Since the PSL is completely filter/estimator dependent, we propose to make use of the estimator-agnostic properties of the CRB to construct a performance metric for the unknown multi-target case similar to (6). Here we advocate the use of the CRB metric in (5) with only two targets with equal power.

Combining (1) and (5), we then obtain the following optimization problem:

$$\begin{aligned}\min_{\boldsymbol{p}}\ \max_{\substack{k\in\{1,2\}\\ \boldsymbol{\omega}\in[0,2\pi)^{2D\times 1}}}\ &\mathrm{tr}\big(\boldsymbol{F}_k^{-1}(\boldsymbol{\omega},\boldsymbol{p})\big)\\ \text{s.t.}\ \ &\boldsymbol{1}^{\mathrm{T}}\boldsymbol{p} = M\,,\ \boldsymbol{p} \in \{0,1\}^N\,.\end{aligned} \tag{7}$$

It is not immediately obvious how we can optimize this efficiently. To do so, we propose to relax the above formulation and exploit the characteristics of the two-target CRB expression. For the remainder, we will assume $D = 2$, the planar array case.

Firstly, let us introduce the following notation:

$$\bar{\bar{\boldsymbol{Z}}} = \boldsymbol{D}_k^{\mathrm{H}} \boldsymbol{P} \boldsymbol{D}_k \tag{8}$$
$$= \begin{bmatrix} \sum_{n=1}^{N} [\boldsymbol{r}_n]_1^2 p_n & \sum_{n=1}^{N} [\boldsymbol{r}_n]_1 [\boldsymbol{r}_n]_2 p_n \\ \sum_{n=1}^{N} [\boldsymbol{r}_n]_1 [\boldsymbol{r}_n]_2 p_n & \sum_{n=1}^{N} [\boldsymbol{r}_n]_2^2 p_n \end{bmatrix},$$

$$\bar{\boldsymbol{Z}}_1(\Delta\boldsymbol{\omega}) = \boldsymbol{A}^{\mathrm{H}} \boldsymbol{P} \boldsymbol{D}_1 \tag{9}$$
$$= \begin{bmatrix} \sum_{n=1}^{N} [\boldsymbol{r}_n]_1 p_n & \sum_{n=1}^{N} [\boldsymbol{r}_n]_1 p_n \mathrm{e}^{\mathrm{j}\boldsymbol{r}_n^{\mathrm{T}} \Delta\boldsymbol{\omega}} \\ \sum_{n=1}^{N} [\boldsymbol{r}_n]_2 p_n & \sum_{n=1}^{N} [\boldsymbol{r}_n]_2 p_n \mathrm{e}^{\mathrm{j}\boldsymbol{r}_n^{\mathrm{T}} \Delta\boldsymbol{\omega}} \end{bmatrix}^{\mathrm{H}},$$

$$\bar{\boldsymbol{Z}}_2(\Delta\boldsymbol{\omega}) = \boldsymbol{A}^{\mathrm{H}} \boldsymbol{P} \boldsymbol{D}_2 \tag{10}$$
$$= \begin{bmatrix} \sum_{n=1}^{N} [\boldsymbol{r}_n]_1 p_n \mathrm{e}^{\mathrm{j}\boldsymbol{r}_n^{\mathrm{T}} \Delta\boldsymbol{\omega}} & \sum_{n=1}^{N} [\boldsymbol{r}_n]_1 p_n \\ \sum_{n=1}^{N} [\boldsymbol{r}_n]_2 p_n \mathrm{e}^{\mathrm{j}\boldsymbol{r}_n^{\mathrm{T}} \Delta\boldsymbol{\omega}} & \sum_{n=1}^{N} [\boldsymbol{r}_n]_2 p_n \end{bmatrix}^{\mathrm{T}},$$

$$\boldsymbol{Z}(\Delta\boldsymbol{\omega}) = \boldsymbol{A}^{\mathrm{H}} \boldsymbol{P} \boldsymbol{A} \tag{11}$$
$$= \begin{bmatrix} M & \sum_{n=1}^{N} p_n \mathrm{e}^{\mathrm{j}\boldsymbol{r}_n^{\mathrm{T}} \Delta\boldsymbol{\omega}} \\ \sum_{n=1}^{N} p_n \mathrm{e}^{-\mathrm{j}\boldsymbol{r}_n^{\mathrm{T}} \Delta\boldsymbol{\omega}} & M \end{bmatrix},$$

where $\Delta\boldsymbol{\omega} = \boldsymbol{\omega}_2 - \boldsymbol{\omega}_1$. From these expressions and (4) it is clear that $\boldsymbol{F}_k(\boldsymbol{\omega}, \boldsymbol{p})$ only depends on $\Delta\boldsymbol{\omega}$ and can be rewritten as

$$\boldsymbol{F}_k(\Delta\boldsymbol{\omega}, \boldsymbol{p}) = \mathrm{Re}\Big\{\bar{\bar{\boldsymbol{Z}}} - \bar{\boldsymbol{Z}}_k^{\mathrm{H}}(\Delta\boldsymbol{\omega}) \boldsymbol{Z}^{-1}(\Delta\boldsymbol{\omega}) \bar{\boldsymbol{Z}}_k(\Delta\boldsymbol{\omega})\Big\}.$$

We can further show that $\bar{\boldsymbol{Z}}_k(\Delta\boldsymbol{\omega}) = \bar{\boldsymbol{Z}}_k^*(-\Delta\boldsymbol{\omega})$, $\boldsymbol{Z}(\Delta\boldsymbol{\omega}) = \boldsymbol{Z}^*(-\Delta\boldsymbol{\omega})$, $\bar{\boldsymbol{Z}}_1(\Delta\boldsymbol{\omega}) = \boldsymbol{P}\bar{\boldsymbol{Z}}_2^*(\Delta\boldsymbol{\omega})$ and $\boldsymbol{Z}(\Delta\boldsymbol{\omega}) = \boldsymbol{P}\boldsymbol{Z}^*(\Delta\boldsymbol{\omega})\boldsymbol{P}$, where $\boldsymbol{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is a permutation matrix. Because of this

$$\boldsymbol{F}_k(\Delta\boldsymbol{\omega}, \boldsymbol{p}) = \boldsymbol{F}_k(-\Delta\boldsymbol{\omega}, \boldsymbol{p}), \tag{12}$$
$$\boldsymbol{F}_1(\Delta\boldsymbol{\omega}, \boldsymbol{p}) = \boldsymbol{F}_2(\Delta\boldsymbol{\omega}, \boldsymbol{p}), \tag{13}$$

and as such we equivalently reformulate (7) as

$$\min_{\boldsymbol{p}} \max_{\Delta\boldsymbol{\omega} \in \mathcal{D}_+} \quad \mathrm{tr}\big(\boldsymbol{F}_1^{-1}(\Delta\boldsymbol{\omega}, \boldsymbol{p})\big)$$
$$\text{s.t.} \quad \mathbf{1}^{\mathrm{T}}\boldsymbol{p} = M, \ \boldsymbol{p} \in \{0,1\}^N,$$

simplifying our problem. The set $\mathcal{D}_+$ is the set of all angle difference vectors that we want to consider in our optimization. It should not contain differences that are too small, since $\boldsymbol{Z}(\Delta\boldsymbol{\omega})$ will become ill-conditioned, but also because a very small angle difference might lead to unresolvable targets. Also, recall that due to the symmetry given by (12), $\mathcal{D}_+$ can be halved.

To further handle the maximization, we propose that $\mathcal{D}_+$ is a discrete set. For each entry of the set, we can then consider a constraint to ensure that the maximum cost is minimized. This leads to

$$\min_{\boldsymbol{p}, \boldsymbol{C}} \quad \mathrm{tr}(\boldsymbol{C})$$
$$\text{s.t.} \quad \mathbf{1}^{\mathrm{T}}\boldsymbol{p} = M, \ \boldsymbol{p} \in \{0,1\}^N$$
$$\boldsymbol{F}_1(\Delta\boldsymbol{\omega}, \boldsymbol{p}) \succeq \boldsymbol{C}^{-1}, \ \forall \Delta\boldsymbol{\omega} \in \mathcal{D}_+.$$

By separating the constraints into parts that do and do not depend on $\Delta\boldsymbol{\omega}$ by introduction of a variable $\boldsymbol{G}$, and application of the Schur complement, we can obtain the following binary semi-definite program (SDP):

$$\min_{\boldsymbol{p}, \boldsymbol{C}, \boldsymbol{G}} \quad \mathrm{tr}(\boldsymbol{C})$$
$$\text{s.t.} \quad \mathbf{1}^{\mathrm{T}}\boldsymbol{p} = M, \ \boldsymbol{p} \in \{0,1\}^N, \ \begin{bmatrix} \bar{\bar{\boldsymbol{Z}}} - \boldsymbol{G} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{C} \end{bmatrix} \succeq \mathbf{0} \tag{14}$$
$$\begin{bmatrix} \boldsymbol{G} & \bar{\boldsymbol{Z}}_1^{\mathrm{H}}(\Delta\boldsymbol{\omega}) \\ \bar{\boldsymbol{Z}}_1(\Delta\boldsymbol{\omega}) & \boldsymbol{Z}(\Delta\boldsymbol{\omega}) \end{bmatrix} \succeq \mathbf{0}, \ \forall \Delta\boldsymbol{\omega} \in \mathcal{D}_+,$$

where all the matrix inequalities are linear in all the variables, see (8), (9) and (11).

This means that, except for the binary constraint, (14) is a convex program. To handle this final obstacle, we propose to use a continuous variable $\tilde{\boldsymbol{p}}$, instead of $\boldsymbol{p}$ in the SDP:

$$\min_{\tilde{\boldsymbol{p}}, \boldsymbol{C}, \boldsymbol{G}} \quad \mathrm{tr}(\boldsymbol{C})$$
$$\text{s.t.} \quad \mathbf{1}^{\mathrm{T}}\tilde{\boldsymbol{p}} = M, \ \tilde{\boldsymbol{p}} \in [0,1]^N, \ \begin{bmatrix} \bar{\bar{\boldsymbol{Z}}} - \boldsymbol{G} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{C} \end{bmatrix} \succeq \mathbf{0}$$
$$\begin{bmatrix} \boldsymbol{G} & \bar{\boldsymbol{Z}}_1^{\mathrm{H}}(\Delta\boldsymbol{\omega}) \\ \bar{\boldsymbol{Z}}_1(\Delta\boldsymbol{\omega}) & \boldsymbol{Z}(\Delta\boldsymbol{\omega}) \end{bmatrix} \succeq \mathbf{0}, \ \forall \Delta\boldsymbol{\omega} \in \mathcal{D}_+.$$

To obtain our desired binary vector $\boldsymbol{p}$, we finally perform a randomized rounding procedure on the optimal $\tilde{\boldsymbol{p}}$ [5]. The optimal $\tilde{\boldsymbol{p}}$ can be obtained by any off-the-shelf solver for convex SDPs.

Though we derived the optimization problem here for planar arrays ($D = 2$), similar problems can be derived for the $D = 1$ and $D = 3$ cases, where the cost also only depends on the difference of AoAs and the symmetries in (12) and (13) also apply. The problem has been derived for linear arrays in [10]. Since the complexity of SDPs primarily scales in the size and number of LMIs [13], the problem is greatly reduced in complexity in the case of $D = 1$: In that case $\Delta\omega$ is a scalar and all the linear matrix inequalities (LMI) are two-by-two instead of four-by-four.

V. Simulation Results

To verify the proposed method and quantify its performance, we have performed a number of simulations comparing our method to random selection and selection using the single-target CRB. The single-target CRB optimization is given by

$$\min_{\tilde{\boldsymbol{p}}, \boldsymbol{C}, \boldsymbol{G}} \quad \mathrm{tr}(\boldsymbol{C})$$
$$\text{s.t.} \quad \mathbf{1}^{\mathrm{T}}\tilde{\boldsymbol{p}} = M, \ \tilde{\boldsymbol{p}} \in [0,1]^N$$
$$\begin{bmatrix} \bar{\bar{\boldsymbol{Z}}} - \boldsymbol{G} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{C} \end{bmatrix} \succeq \mathbf{0}, \ \begin{bmatrix} \boldsymbol{G} & \bar{\boldsymbol{z}}_1^{\mathrm{T}} \\ \bar{\boldsymbol{z}}_1 & M \end{bmatrix} \succeq \mathbf{0},$$

where $\bar{\boldsymbol{z}}_1 = \begin{bmatrix} \sum_{n=1}^{N} [\boldsymbol{r}_n]_1 p_n & \sum_{n=1}^{N} [\boldsymbol{r}_n]_2 p_n \end{bmatrix}$, i.e., the first row of $\bar{\boldsymbol{Z}}_1(\Delta\boldsymbol{\omega})$.

The following parameters are fixed, except where noted: 100 randomized rounding trials, $M = \frac{1}{2}N$, $\mathcal{D}_+$ consists of 500 vector samples with uniformly sampled magnitudes between $\Delta\omega_{\text{min}}$ and π and uniformly sampled directions. In practice, $\Delta\omega_{\text{min}}$ would be determined by the system requirements and/or

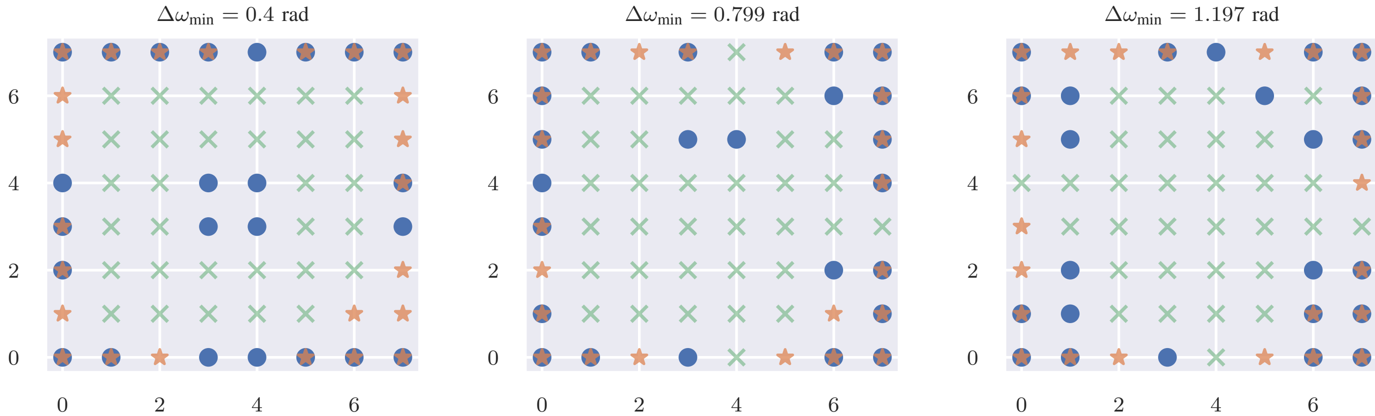


Fig. 1: Example of planar array selections for different minimum separations. The blue dots indicate array elements by the proposed method, red stars are array elements selected by the single source CRB optimization, and green crosses indicate those that are not selected by either method.

prior knowledge of the radar scene, for example. For the candidate array, we pick a square uniform planar array (UPA). Let $\sqrt{N} = N_x = N_y$, where N_x and N_y are the amount of columns and rows of the UPA, respectively.

In Fig. 1, we show a number of selection examples for the eight-by-eight planar array candidate set. The blue dots, of which there are twenty-four, indicate array elements selected by the proposed method, red stars are array elements selected by the single source CRB optimization, and green crosses indicate those that are not selected by either method. By varying $\Delta\omega_{\min}$, we can obtain different selections that are more or less distinct from the single-target optimization result. As $\Delta\omega_{\min}$ increases, the results become more and more similar to the single-target optimization result. Similar to the linear array results in [10], we see a mix of edge and center elements being selected for small $\Delta\omega_{\min}$, with the center selections 'moving outward' as $\Delta\omega_{\min}$ gets larger. This implies that the worst case happens mostly for smaller differences between targets, which makes intuitive sense. For smaller $\Delta\omega_{\min}$, we don't see notably different results. Note that while we expect some symmetry in the selection (this due to our metric only depending on the angle differences), we may not always obtain this in the final selection, due to the randomized rounding step.

Fig. 2 shows the results of more extensive simulations for different candidate set sizes and sparsities. The CRB that is plotted is $\max_{\Delta\boldsymbol{\omega}\in\mathcal{D}_+} \operatorname{tr}\big(\boldsymbol{F}_1^{-1}(\Delta\boldsymbol{\omega},\boldsymbol{p})\big)$, with $\Delta\omega_{\min} = 3.5N_x^{-1}$. We see there that the optimization is successful in optimizing the worst-case two-target CRB and is thus outperforming the other methods. The shaded regions indicate the worst and best performing randomized rounding results. While the shaded region is generally slim for our proposed method, there is clearly some merit to performing multiple rounding trials and picking the best result. It should be noted, that the differences in performance may not be very large, and that our proposed method is more complex to compute than the others. Whether this trade-off is worthwhile depends on the specifics of the application. During offline optimization however, where the computation of the array selection is done prior to deployment, the increased computational complexity is less of an issue.

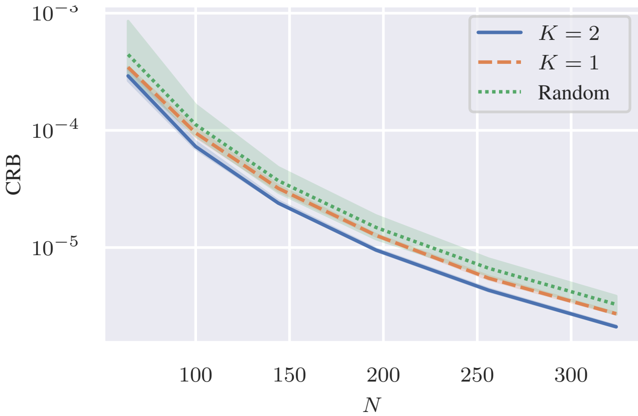
Finally, we show two example matched filter responses where the mainlobe is aimed at $\left[\frac{\pi}{4} \quad \frac{\pi}{4}\right]^{\mathrm{T}}$, $N = 64$, and $M = 24$. We see that compared to the single target method, for the proposed method, there are less prominent sidelobes near the mainlobe and the peak sidelobe level is lower, at the cost of mainlobe width.

VI. Conclusions

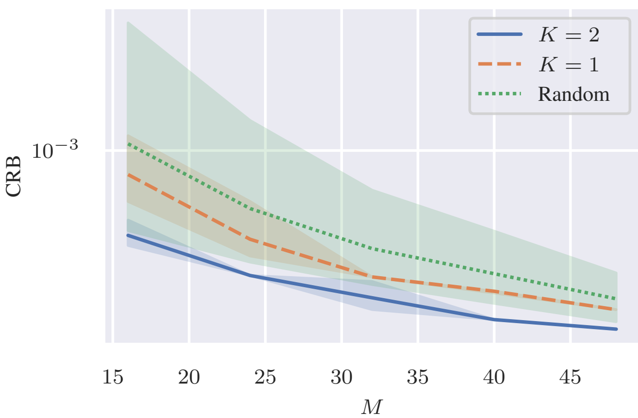
We showed our method is successful in optimizing the worst-case two-target CRB for planar arrays. The worst-case two-target CRB functions as an estimator agnostic surrogate for peak sidelobe suppression, which makes it useful for modern estimators which do not produce sidelobes like, for example, a matched filter receiver. We have shown that using the worst-case two-target CRB also leads to PSL reduction by inspecting some example array responses, without the need to include sidelobe suppresing constraints or costs explicitly.

The proposed method is a convex semidefinite program which, after randomized rounding, produces a binary selection vector. Both steps can be efficiently solved [5], [13]. Though we have shown only results for square UPAs, the method does not require a specific candidate set structure. In fact, it will work for any candidate set of planar array elements by adapting (8), (9) and (11) for that particular set.

We have shown that our method outperforms random selection and single-target CRB optimization in terms of the worst-case two-target CRB through simulation. This shows our method is successful in optimizing the metric. Further, simulations have shown that the variation in performance due to the randomized rounding step is small, but not always insignificant, warranting choosing the best from multiple rounding trials. Through example array element selections it appears that center elements of the array are useful in distinguishing closely spaced targets. As targets become more separated, the benefit of the center elements decreases.



(a) Results for varying candidate set sizes. $M = \frac{N}{2}$.



(b) Results for varying sparsities. $N = 64$.

Fig. 2: Worst-case two-target CRB performance of a number of different methods using different planar candidate array sizes and sparsities.

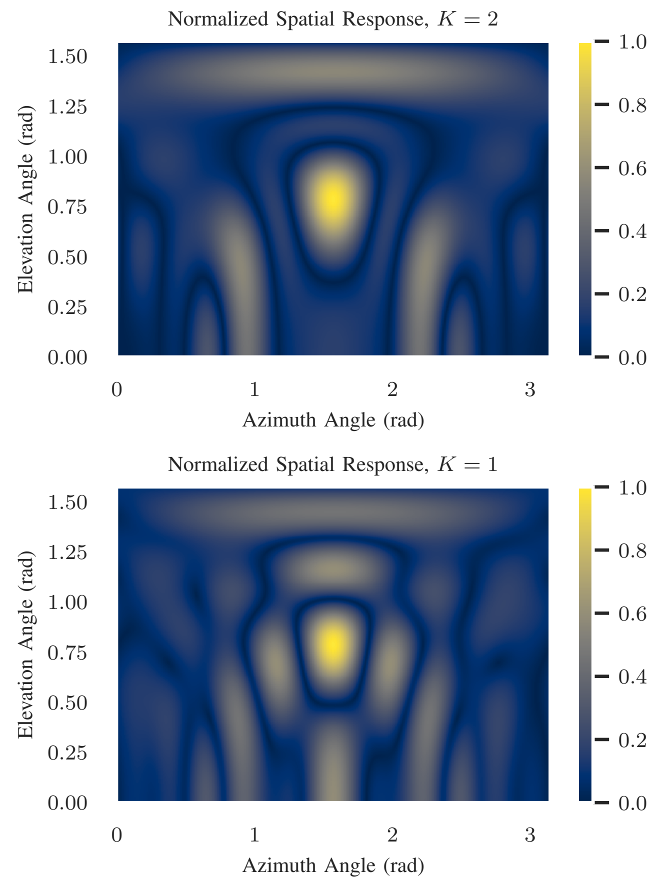


Fig. 3: Angle response of the proposed method and the single-target CRB optimization method, respectively. The mainlobe is aimed at $\begin{bmatrix} \frac{\pi}{4} & \frac{\pi}{4} \end{bmatrix}^{\mathrm{T}}$, $N = 64$, and $M = 24$.

For future work, we would like to investigate alternative directions for solving the original minimax problem, e.g., methods which do not require sampling the feasible region of the maximizer and thus do not result in a large number of constraints in the SDP. Furthermore, we would like to adapt the method to correlated sources and to performing the selection on both transmit and receive.

References

[1] J. H. Ender, "On compressive sensing applied to radar," *Signal Processing*, vol. 90, no. 5, pp. 1402–1414, May 2010.

[2] E. Tohidi, M. Radmard, S. M. Karbasi, H. Behroozi, and M. M. Nayebi, "Compressive sensing in MTI processing," in *2015 3rd International Workshop on Compressed Sensing Theory and Its Applications to Radar, Sonar and Remote Sensing (CoSeRa)*, Jun. 2015, pp. 189–193.

[3] Kumar Vijay Mishra and Yonina C. Eldar, "Sub-Nyquist Radar: Principles and Prototypes," in *Compressed Sensing in Radar Signal Processing*, 1st ed., A. De Maio, Y. C. Eldar, and A. M. Haimovich, Eds. Cambridge University Press, Sep. 2019, pp. 1–48.

[4] S. Joshi and S. Boyd, "Sensor Selection via Convex Optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, Feb. 2009.

[5] S. P. Chepuri and G. Leus, "Sparsity-Promoting Sensor Selection for Non-Linear Measurement Models," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 684–698, Feb. 2015.

[6] S. Sedighi, B. Shankar, M. Soltanalian, and B. Ottersten, "One-Bit DoA Estimation via Sparse Linear Arrays," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, May 2020, pp. 9135–9139.

[7] H. Godrich, A. P. Petropulu, and H. V. Poor, "Sensor Selection in Distributed Multiple-Radar Architectures for Localization: A Knapsack Problem Formulation," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 247–260, Jan. 2012.

[8] V. Roy, S. P. Chepuri, and G. Leus, "Sparsity-enforcing sensor selection for DOA estimation," in *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. St. Martin, France: IEEE, Dec. 2013, pp. 340–343.

[9] E. Tohidi, M. Coutino, S. P. Chepuri, H. Behroozi, M. M. Nayebi, and G. Leus, "Sparse Antenna and Pulse Placement for Colocated MIMO Radar," *IEEE Trans. Signal Process.*, vol. 67, no. 3, pp. 579–593, Feb. 2019.

[10] C. A. Kokke, M. Coutino, L. Anitori, R. Heusdens, and G. Leus, "Sensor Selection for Angle of Arrival Estimation Based on the Two-Target Cramér-Rao Bound," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–5.

[11] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramer-Rao bound," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 5, pp. 720–741, May 1989.

[12] A. Dogandzic and A. Nehorai, "Cramer-Rao bounds for estimating range, velocity, and direction with an active array," *IEEE Trans. Signal Process.*, vol. 49, no. 6, pp. 1122–1137, Jun. 2001.

[13] L. Vandenberghe and S. Boyd, "Semidefinite Programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, Mar. 1996.