Design of Large Effective Apertures for Millimeter Wave Systems Using a Sparse Array of Subarrays

Anant Gupta, Student Member, IEEE, Upamanyu Madhow, Fellow, IEEE, Amin Arbabian, Senior Member, IEEE, and Ali Sadri

Abstract—We investigate synthesis of a large effective aperture using a sparse array of subarrays. We employ a multi-objective optimization framework for placement of subarrays within a prescribed area dictated by form factor constraints, trading off the smaller beam width obtained by spacing out the subarrays against the grating and side lobes created by sparse placement. We assess the performance of our designs for the fundamental problem of bearing estimation for one or more sources, comparing performance against estimation-theoretic bounds. Our tiled architecture is motivated by recent progress in low-cost hardware realizations of moderately sized antenna arrays (which play the role of subarrays) in the millimeter wave band, and our numerical examples are based on 16-element (4 × 4) subarrays in the 60 GHz unlicensed band.

Index Terms—Millimeter wave radar, estimation bounds, compressive estimation, gridless super-resolution, sparse subarray design, multi-objective optimization.

I. INTRODUCTION

MANY sensing and situational awareness applications (e.g., radar imaging for vehicles and drones) require highly directional, electronically steerable beams. Reducing beam width requires expansion of antenna aperture. This is typically accomplished by filling the aperture with antenna elements spaced at half the carrier wavelength or less, in order to avoid grating lobes. However, this approach does not scale well with aperture size since the cost, power consumption and design complexity increases with number of antenna elements.

In this paper, we investigate the problem of synthesizing narrow beams using a tiled architecture, with a sparse set of subarrays spread over a large physical aperture. Each subarray is a relatively compact antenna array with a moderate number of elements at sub-wavelength spacing. This is a modular design, in which each subarray can be controlled by a radio frequency integrated circuit (RFIC) of moderate complexity, with multiple RFICs tiled to build up large aperture arrays with a much larger number of elements compared to that of a single subarray. The resulting array is “sparse” because, while the total number of antenna elements summed across subarrays is large, this number is far smaller than that for a classical design with antennas at sub-wavelength spacing spanning the entire physical aperture. The sidelobes and grating lobes resulting from such spatial undersampling must therefore be controlled in order for our proposed “array of subarrays” to be useful. Our goal here is to determine the placement of a given number of subarrays over a physical aperture in order to optimize multiple beam attributes, including beam width, maximum sidelobe level, and directivity.

While our framework is general, the design of millimeter wave arrays is of particular interest to us, because the small carrier wavelength enables synthesis of narrow beams using relatively compact apertures. As a running example throughout this paper, we consider the design of a 60 GHz array of subarrays created by placing 8 subarrays over an aperture size of 10 cm by 10 cm (20λ × 20λ for wavelength λ = 0.5 cm), where each subarray has 4 × 4 elements arranged in uniform rectangular grid with 0.5λ horizontal spacing and 0.6λ vertical spacing. The total number of antenna elements in such designs is 128, which is an order of magnitude smaller than the 1600 elements required to cover the entire aperture at half-wavelength spacing. In addition to optimizing beam characteristics, our design framework also accounts for practical placement constraints consistent with existing prototype subarrays. For example, the subarrays need to be aligned along their axes, assuming that all elements have unidirectional linear polarization. Also, each subarray tile occupies extra physical area on the plane, which must be accounted for in the placement procedure.

A. Contributions

Our contributions are summarized as follows:

- We formulate the problem of subarray placement as multi-objective optimization of key performance measures such as beam width (BW), maximum sidelobe level (MSLL), eccentricity (ecc) and directivity (GD),

\[
\begin{align*}
\text{Minimize} & \quad BW(C, w), \text{MSLL}(C, w), \text{ecc}(C, w) \\
\text{Maximize} & \quad GD(C, w) \\
\text{subject to} & \quad AoS(C)
\end{align*}
\]

where \( C \) is \( N_s \times 2 \) Subarray center position matrix, \( w \) is \( N \times 1 \) beamsteering weight vector and \( AoS(C) \) are
physical constraints to avoid overlapping subarrays. Note
that orientation of subarrays is not an optimization variable
in this architecture, since the polarization of the elements
has to be aligned for beamforming. We consider minimiza-
tion of a weighted linear combination of the objectives,
focusing mainly on beamwidth \( B \) and maximum sidelobe
level \( MSLL \). The configuration \( C \) that we optimize over is
characterized by a set of discrete-valued variables, and the
number of possible values for these variables is combinatori-
ally explosive. Furthermore, we do not have closed form
expressions for the performance measures as a function of
\( C \), hence significant computation is required to evaluate
the cost function for each configuration. In order to control
the complexity, we use geometric heuristics to eliminate
similar configurations in the first stage of our algorithm,
and then employ a second stage of refinement using small
perturbations around the first stage solution. We numeri-
cally explore the Pareto front for (1) by sweeping through
the weights, and illustrate how the beam attributes depend
on the weights.

- We evaluate our designs using estimation-theoretic bench-
marks for two-dimensional (2D) direction of arrival (DoA)
estimation. At low signal-to-noise ratio (SNR), large side-
lobes can lead to large errors in the DoA estimate. At high
SNR, on the other hand, the DoA estimation error is gov-
erned by beam width. We derive a Ziv-Zakai bound (ZZB),
which captures the effect of both large and small estimation
errors, for DoA estimation for specular paths. The ZZB
exhibits a distinct transition in its behavior from low to
high SNR, tending at high SNR to the Cramer-Rao bound
(CRB), which captures the effect of small errors around
the true parameter value. Thus, we use the ZZB transition
SNR as a measure of efficacy of sidelobe reduction, and
the CRB as a measure of efficacy of beam width reduction.

- We report in detail on two array designs, \( A_1 \) with primary
emphasis on reducing beamwidth and \( A_2 \) based on joint
optimization of beamwidth and maximum sidelobe level.
These designs are compared against two benchmark ar-
drays. The first is termed a “compact array,” with subarrays
packed closely together: this is expected to have worse
beam width but smaller sidelobes than our sparse designs.
The second is termed a “naive array,” obtained by placing
subarrays in diamond pattern to obtain beamwidth equiva-
 lent to that of sparse array \( A_2 \). Some illustrative numerical
results for our running example are as follows. The sparse
design \( A_1 \) is 11 dB better than the compact array in terms
of CRB, while degrading less than 1 dB in terms of ZZB
threshold. The sparse design \( A_2 \) is 4 dB better in terms of
CRB than the “naive array,” while also having a better ZZB
threshold.

- We investigate DoA estimation performance numerically
using a state of the art algorithm for off-grid estimation. The
impact of the higher sidelobes due to sparse placement, and
hence that of our optimization procedure, is more evident
when estimating DoA in the presence of multiple interfer-
ting targets. We show that, depending on the strength of
interferers, our optimized arrays achieve better estimation
accuracy than the “compact” and “naive” benchmark arrays
at moderate to high SNR due to a combination of sharper
beamwidth and lower sidelobes. We also show that the
efficacy of DoA estimation using our sparse designs, and
the associated benchmarks, is maintained when we employ
compressive measurements.

B. Related Work

There is a rich body of work on sparsifying linear arrays,
including minimum redundancy arrays [1], genetic optimiza-
tion [2], joint Cramér Bound and sidelobe level optimiza-
tion [3], and simulated annealing [4]. Most popular design strate-
gies try to find an element pattern which minimizes beamwidth,
along with some notion of DoA ambiguities such as sidelobe
level or probability of DoA outlier. Recent approaches like
Nested 2D arrays [5] and H-arrays [6] utilize the idea of “dif-
fERENCE CO-ARRAY” to reduce the number of redundant spacings
and maximize the randomness of element positions, so that the
number of spatial frequencies being sampled by the array is
maximized.

A closely related sparse array design methodology is the
sensor selection problem, wherein a smaller subset of individual
antenna positions is to be chosen from a predefined grid. By
employing certain surrogate measures, near-optimal arrays can
be obtained in polynomial time using standard convex relaxation
methods [7], [8]. Array thinning methods such as these are well
known to avoid complicated nonlinear optimizations for linear
case [9].

Most existing techniques, however, assume that antenna ele-
ments can be placed freely. Hence, they do not apply in our set-
ning, where element placement within subarrays is constrained.
The prior work most similar to our is [10], which investigates
design of linear arrays with two and three subarrays. However,
the focus there is on performance criteria for comparing a
number of sensible designs in a far smaller design space, rather
than searching over a large space of possibilities as we do here.

Our performance evaluation requires implementation of DoA
estimation algorithms. Classical subspace-based algorithms
such as MUSIC [11] and ESPRIT [12], as well as their extensions
to arrays of subarrays such as [13]–[16], rely on regular array
geometries for efficient computation. Recently developed super-
resolution algorithms such as Basis Pursuit Denoising (BPDN)
[17] and Newtonized Orthogonal Matching Pursuit (NOMP)
[18] are both more general and have better performance. It
is worth noting that [19] shows that, for large arrays, BPDN
and other sparse estimation techniques with compressive mea-
surements outperform subspace-based methods. In this paper,
we employ NOMP in our numerical experiments, since we
have found it to provide better performance than BPDN at
lower complexity. We also show that the performance trends are
unchanged under compressive measurements, consistent with
recent general theory [20].

C. Outline

We first describe the beam attributes to be optimized while
designing the sparse arrays and discuss constraints for the
optimization in Section II. The geometric heuristics and design approach are described in detail in Section III. We then provide a brief review of estimation bounds for 2D bearing estimation and discuss their utility for analyzing the sparse arrays in Section IV. Numerical results are provided in Section V. These include exploration of the Pareto front, and comparison of example designs against benchmarks in terms of both beam characteristics and DoA estimation. We show in Section VI that the performance trends hold for compressive DoA estimation, and conclude in Section VII.

II. SPARSE SUBARRAY DESIGN

We formulate the array design problem in terms of jointly optimizing multiple beam parameters that are expected to affect DoA estimation performance.

A. Beam Pattern Basics

We use the directional cosines

\[ u = \sin(\theta) \cos(\phi), \quad v = \sin(\theta) \sin(\phi) \]

(2)
to represent the DoA of target. The elevation \( \theta \) and azimuth \( \phi \) angles are measured from the broadside direction (perpendicular to the baseline array plane). The 2D beampattern \( R(u, v) \) in direction \((u, v)\) when the beam is steered towards the broadside is given by

\[ R(u, v) = \frac{1}{N^2} \left| \sum_{i=1}^{N} e^{j(k(ud_i + vd_i))} \right|^2 \]

(3)

where \( N \) is the number of array elements; \([d_x^i, d_y^i]^T \triangleq d_i \) are the 2D co-ordinates of arrays elements and \( k = \frac{2\pi}{\lambda} \) is the wavenumber. We assume isotropic antenna elements with ideal steering weights and far-field sources with normalized response. The term “subarray” refers to the subset of elements with uniform half-wavelength spacing, while “super-array” refers to the placement of these subarrays, which is described by the subarray centers. Since the elements in a subarray are fixed, the array element locations, \( D \) can be expressed in terms of the subarray centers, \( C \), as \( D = C \times 1_{N_e} + D_c \times 1_{N_c} \), where \( D_c \) is the fixed \( 2 \times N_e \) matrix containing the subarray element coordinates with respect to its center, \( N_e \) is the number of subarrays, \( N_c \) is number of elements in individual subarrays, \( 1_n \) is an \( n \times 1 \) column vector of ones, and \( \otimes \) denotes the Kronecker product.

When beamforming in a general direction \((u_0, v_0)\) (broadside corresponds to \((u_0, v_0) = (0, 0)\)), the beam pattern is given by

\[ R(u_0, v_0)(u, v) = R(u - u_0, v - v_0) \]

(4)

For Direction of Arrival (DoA) estimation, the ideal beam should have small beamwidth with minimal sidelobes and high directivity. We consider the following beam attributes, some of which depend on the steering direction \((u_0, v_0)\), as key performance metrics to be optimized:

- **2D beamwidth (BW):** Although the main beam of non-uniform array has non-trivial shape in 2D, we approximate it as an ellipse to define beamwidth. We evaluate the 2D beamwidth in terms of the 3-dB beamwidths along the major and minor axes of this ellipse, denoted by \( \text{BW}_{\text{max}} \) and \( \text{BW}_{\text{min}} \) respectively. The mean squared error of DoA estimation depends on the sum of these beamwidths (see Appendix A), hence we define beamwidth as \( \text{BW}_{\text{DoA}} = \sqrt{\text{BW}_{\text{max}}^2 + \text{BW}_{\text{min}}^2} \).

- **Maximum sidelobe level (MSLL):** Is the relative level of the strongest sidelobe in the beampattern with respect to the main lobe, \( \text{MSLL} = 10 \log(R_{\text{max}}/R_{\text{msl}}) \). Thus, \( R_{\text{max}} \) and \( R_{\text{msl}} \) are the largest and second largest magnitude local maxima of beampattern \( R(u, v) \) given by

\[ R_{\text{max}}(u_0, v_0) = \max_{u,v} R_{u_0,v_0}(u, v) \]

\[ = R_{u_0,v_0}(u_0, v_0) = R(0, 0) \]

\[ R_{\text{msl}} = \max_{u,v} R_{u_0,v_0}(u^*, v^*) \]

s.t. \((u^*, v^*) \neq (u_0, v_0)\),

\[ R_{u_0,v_0}(u^*, v^*) \geq R_{u_0,v_0}(D_c(u^*, v^*)) \]

where \( D_c(u^*, v^*) = \{(u, v) : |u - u^*| < \epsilon, |v - v^*| < \epsilon\} \) denotes \( \epsilon \)-neighborhood. Note that \( R_{\text{max}} \) does not depend on steering direction \((u_0, v_0)\), but \( R_{\text{msl}} \) might.

- **Directivity (\(\text{G}_D\)):** The directivity is the ratio of main lobe power to average power, \( \text{G}_D = 10 \log\left(\frac{\text{BW}_{\text{max}}^2}{R_{\text{avg}}}\right) \). The average power does not have a closed form expression for general planar arrays, and is evaluated in \((u, v)\) domain by the integral [21]:

\[ R_{\text{avg}} = \frac{2}{4\pi} \int_{-1}^{1} \int_{-1}^{1} \frac{R_{u_0,v_0}(u, v)}{\sqrt{1-u^2-v^2}} \, du \, dv \]

- **Eccentricity (\(\text{ecc}\)):** is a measure of the asymmetry of the main beam. We add this additional parameter to suppress the trivial linear placement solution, \( \text{ecc} = \sqrt{1 - (\text{BW}_{\text{min}}/\text{BW}_{\text{max}})^2} \).

For non-uniform planar arrays, none of these beam parameters have a closed form expression [22], hence they must be computed numerically. In our simulations, we compute these beam attributes using a beampattern over a \(512 \times 512\) grid in UV space as shown in Fig. 2.
where

\[
f(C, w) = \alpha \text{BW}(C, w) + \beta \text{MSLL}(C, w) + \gamma \text{GCC}(C, w) \tag{6}
\]

is the weighted cost function in terms of the normalized objective functions and \([\alpha, \beta, \gamma]\) are weights that can be used to sweep through the optimal surface for this optimization. We show some example arrays obtained for different choices of weights in Table I.

Since the cost function \(f(C, w)\) is evaluated numerically from its beampattern for specific values of \((C, w)\), exploring the entire solution space is computationally infeasible. This discrete-valued nature of the optimization variables leads to a combinatorial problem without closed form objectives and constraints. Furthermore, beam characteristics in general depend on the steering weights \(w\), which in turn depend on the direction \((u_0, v_0)\) in which we are steering. We therefore employ two key simplifications:

- We remove the dependence of the cost function on beamforming direction, and hence on steering weights, by computing the objectives based on an expanded beam pattern, as discussed in Section II-C.
- We employ geometric heuristics to cut down the solution space to a reasonable size, as described in Section III.

C. Invariance to Beamforming Direction

The cost function in (6) is evaluated using the beampattern \(R(u - u_0, v - v_0)\), which depends on the beamsteering direction \((u_0, v_0)\). It would be prohibitively expensive to evaluate the beam attributes over all such beampatterns for finding the optimal \((C, w)\). However, for arrays with isotropic elements, we can define an Expanded Beam pattern (EBP) which subsumes beampatterns of all steering direction in a Region of Interest (ROI) [24]. Suppose that our maximum steering angle in the ROI is \(\theta_{\text{max}}\). From (2), we see that \((u_0, v_0)\) lies within a circle of radius \(\sin \theta_{\text{max}}\). On the other hand, sidelobes can appear at any \((u, v)\) within a circle of radius 1. It is easy to see, therefore, that \((u - u_0, v - v_0)\) is guaranteed to lie within a circle of radius \(1 + \sin \theta_{\text{max}}\). We can therefore compute beam attributes using the following EBP:

\[
R_{\rho}(\tilde{u}, \tilde{v}) = \frac{1}{N^2} \left| \sum_{i=1}^{N} e^{j k \rho (\tilde{u} d_i^x + \tilde{v} d_i^y)} \right|^2
\]

\[
\rho = 1 + \sin(\theta_{\text{max}}) \tag{7}
\]

Fig. 4 shows the EBP, \(R_{1.5}(\tilde{u}, \tilde{v})\) for ROI with \(\theta_{\text{max}} = 30^\circ\), and the beampattern for the steering angle \(((u_0, v_0) = (0.3, 0.4))\). The shape of the main beam is preserved under the transformation (7), hence beam width and eccentricity can be directly evaluated from (7). The MSLL evaluated from EBP is a worst-case value, corresponding to an argument \(\rho(\tilde{u}, \tilde{v})\) which, in principle, might not correspond to a feasible value of \((u - u_0, v - v_0)\) in (4). However, the maximum sidelobe always lies within the main lobe of the subarray beam pattern, so that physically implausible values of \(\rho(\tilde{u}, \tilde{v})\) do not correspond to large local maxima of the EBP.
With the introduction of the EBP, we can, without loss of generality, assume that the main beam is being steered towards broadside, setting \( w = 1_N \). Our problem now reduces to finding the optimal configuration \( C^* \) as follows,

\[
C^* = \arg \min_C f(C, 1_N) \quad \text{subject to} \quad \text{AoS}(C) \tag{8}
\]

### III. Placement Optimization

In order to optimize the placement, we need to evaluate the cost function over all array configurations. The number of configurations depends on the allowed form factor, and the size of the subarray module, and an exhaustive search over all configurations is computationally infeasible: for example, the number of configurations for a discrete grid of size \( 20 \times 20 \) is of the order \( 10^{20} \). We therefore propose a two-stage approach, first performing a combinatorial search on a reduced search space, and then obtaining the final solution by searching over perturbations around the solution from the first stage.

#### A. Combinatorial Search

We reduce the solution space by removing geometrically “similar” arrays. We employ the covariance of the element positions, and pairwise element separations, as measures of similarity. The choice of covariance of element positions \( \Sigma(C) \) as similarity metric is motivated by its inverse proportionality to Cramér Rao Bound on accuracy of DoA estimation (see Section IV-A1). However, array configurations with similar array covariance but diverse beam attributes also exist: Fig. 5 shows an example of two array configurations with different shapes but the same covariance. The arrays have similar beamwidth but their MSLL levels are different. We observe that the variance of pairwise element distances \( \psi(C) \) is different for these arrays, and use it as an indicator for these large scale deviations. This allows us to reduce the dimensionality of the solution space from \( 2 \times N_s \) down to 3 array shape parameters: the eigenvalues \( (\lambda_1, \lambda_2) \) of \( \Sigma(C) \) and \( \psi(C) \).

1) Subarray Placement Algorithm: We construct a prefix tree dictionary to find feasible solutions using a breadth first search based enumeration technique. The element position covariance for an array of subarray can be uniquely represented by the covariance of its subarray centers, \( \Sigma_D = \Sigma_C + \Sigma_{D_e} \).
Hence the super-array center covariance can be used instead of that of the full array in the Dictionary search algorithm. Each node in the tree stores a subarray center position, and the path from root to a node at the \( n \)th layer of prefix tree represents a unique configuration of \( n \) subarrays. The subarray centers are constrained to lie on a fixed set of discrete grid points \( G \). The algorithm is described in Algorithm 1. We briefly discuss the key steps below.

- **INITIALIZE**: In order to allow for sufficient exploration, we employ multiple random initializations \( C^i_1 \) of the root node being placed on \( N_{\text{init}} \) different locations on the grid. (For example, circular configurations cannot be obtained if the root subarray is fixed at the center).
- **LIST**: Define the operator \( J_G : G^{|C^i_1|} \to G^{|V_i|} \) which maps the set of subarrays centers \( C^i_1 \) to the set of \( |V_i| \) vacant gridpoints in \( G \) available for placement of next subarray which are not blocked by the subarrays already placed at \( C^i_1 \). This operator also accounts for additional surface area occupied by the subarray module apart from the physical antenna elements (see Appendix C for details).
- **APPEND**: The \((n+1)\)th subarray configuration is constructed from the vacancies, \( C^i_n \times V_i \) where \( \times \) denotes cartesian product of sets. A temporary dictionary \( C^{n+1} \) is formed by inserting \( |V_i| = \kappa \) child nodes for each node in the \( n \)th layer.
- **PRUNE**: Nodes corresponding to “similar” configurations are deleted based on the array shape parameters
  1. Find eigenvalues (\( \lambda_1, \lambda_2 \)) of the subarray center covariance matrix, \( \Sigma(C) \) and variance of array separations, \( \psi(C) = \mathbb{E}[(l_{ij} - \mathbb{E}[l_{ij}])^2] \), where, \( l_{ij} \) denotes the distances between \( i \)th and \( j \)th elements.
  2. Enumerate unique configurations by binning the \( (\lambda_1, \lambda_2, \psi) \) triplets over a 3-D grid with resolution \( \tau \) and randomly picking one configuration from each bin (see Appendix D for criteria to choose \( \tau \)).

This procedure is repeated until the number of dictionary atoms reach the desired number of subarrays, \( n = N_s \). All arrays in the dictionary \( C \) obtained from this algorithm satisfy the AoS(\( C \)) constraint by construction. This simplifies the constrained multi-objective optimization in (5) to

\[
C^* = \arg\min_{C \in \mathcal{C}} f(C, 1_N) \tag{9}
\]

### B. Iterative Placement Refinement

In the second stage, we try to improve the cost function (9) by applying small local perturbations (within a bin of the grid \( G \)) to the subarray positions obtained from the combinatorial search in the first stage, as described in Algorithm 2.

Fig. 6 shows a sample of how costs are minimized using sequential refinement over \( N_s = 8 \) subarrays. After running few iterations, the final array has 0.8 dB lower MSLL, while keeping other beam attributes relatively unchanged.

### C. Computational Complexity

The complexity of this approach is dominated by the construction of the prefix tree dictionary \( \mathcal{C} \) in the first stage. Our algorithm progresses by growing leaf nodes of the tree in a breadth first fashion until a tree depth of \( N_s \) is reached. In each step, the key bottleneck lies in pruning the temporary dictionary \( \mathcal{C}^n \) which lists all possible vacancies for every leaf node of existing tree. The number of operations and space can be upper bounded as follows (see Appendix E for details):

\[
T_{\text{Algorithm 1}} = \sum_{n=1}^{N_s-1} \mathcal{C}^{n+1} \leq \sum_{n=1}^{N_s-1} |G||E_{\text{max}}^n| \sim O(|G|^3 N_s^4)
\]

\[
S_{\text{Algorithm 1}} = \max_{n \in [1,N_s]} |\mathcal{C}^n| \leq |E_{\text{max}}^n| \sim O(|G|^3 N_s^3)
\]
The resulting dictionary is independent of cost function weights \( [\alpha, \beta, \gamma] \), and represents a thinned version of the entire design space. Thus, optimized arrays for different weights can be evaluated by solving (9). The solution of (9) has \( O(|C|) \) time complexity, assuming that cost evaluation for each array configuration takes constant time. The second stage of the algorithm requires cost evaluation of fine perturbations over an oversampled bin of size \( |B| \) around each subarray, repeated \( N_{ref} \) times. This stage requires \( T_{\text{Algorithm2}} = |B|N_{ref}N_s \) operations using constant space, which is significantly lower compared to first stage. Hence, the overall algorithm complexity is \( O(|G|^4N_s^4) \) in time and \( O(|G|^3N_s^3) \) in space.

We note that the polynomial complexity of proposed algorithm is significantly better compared to exhaustive search, which exhibits exponential complexity \( O(|G|^N_s) \). For our running example, the proposed algorithm’s complexity is \( T_{\text{Algorithm1}} = (400 \times 8)^4 \approx 10^{14} \) compared to the complexity of exhaustive search, \( T_{\text{Exhaustive}} = (400)^8 > 10^{20} \).

IV. ESTIMATION-THEORETIC BENCHMARKS

We now seek to evaluate the efficacy of our sparse designs for the canonical application of 2D DoA estimation. We compare different array designs in terms of estimation-theoretic bounds as well as simulated performance using a super-resolution algorithm. For clarity in exposition, from here onwards we overload \( \mathbf{u} \triangleq [u, v] \) to denote the DoA.

A. Signal Model

We model the received signal from \( K \) sources in the scene with distinct DoAs \( \Theta = \{\theta_1, \theta_2, \ldots, \theta_K\} \) as

\[
x = \sum_{j=1}^{K} \alpha_j s(\theta_j) + z \tag{10}
\]

where \( s(\theta_j) = [e^{j\kappa_1^T d_1}, \ldots, e^{j\kappa_N^T d_N}]^T \) is the array response, \( z = [z_1, \ldots, z_N]^T \) is complex white noise such that \( \mathbb{E}(zz^H) = \sigma^2 I_N \), and \( \{\alpha_j\}_{j=1}^{K} \) are complex gains which are unknown deterministic constants. The joint probability density of received signal conditioned on \( (\Theta, \{\alpha_j\}_{j=1}^{K}) \) is given by,

\[
p(x|\Theta, \alpha) = \prod_{\mathbf{u}_j \in \Theta} \frac{1}{\pi N \exp \left( -\frac{||x - \alpha_j s(\mathbf{u}_j)||^2}{\sigma^2} \right)} \tag{11}
\]

For any DoA estimator \( \hat{\Theta} \), the covariance of estimation error is defined as,

\[
\mathbf{C}(\hat{\Theta}) = \mathbb{E} \left[ \sum_{i=1}^{K} (\mathbf{u} - \hat{\mathbf{u}}_i)(\mathbf{u} - \hat{\mathbf{u}}_i)^T \right]
\]

\( \mathbf{C} \) can be geometrically interpreted by its trace \( \sqrt{\text{tr}(\mathbf{C})} \) which represents the expected overall Root mean square error (RMSE) in DoA estimation (see Appendix A). We use this measure to compare the performance of array designs in Section V. For single source case \( (K = 1) \), the joint maximum likelihood estimator of \( \mathbf{u} \) and \( \alpha \) yields a noncoherent estimator for \( \mathbf{u} \) as follows:

\[
\hat{\mathbf{u}}_{ML} = \arg \max_{\mathbf{u}} |s(\mathbf{u})^H \mathbf{x}|^2 \tag{12}
\]

For this case, we derive the Cramer Rao (CRB) and Ziv-Zakai (ZZB) bounds on \( \mathbf{R}_e \) to assess the best possible estimation accuracy of different designs. Although derived for single source case, we use these bounds for multiple source case as well to compare DoA estimation performance.

1) Cramér Rao Bound: The Bayesian Cramér Rao Bound for this signal model is given by [24]:

\[
\mathbf{CRB}(\mathbf{R}_e) = (\mathbf{J}_F + \mathbf{J}_P)^{-1} \tag{13}
\]

where \( \mathbf{J}_F, \mathbf{J}_P \) denote the Fisher Information Matrix (FIM) contributions from the observation and the prior distribution of DoA respectively.

\[
(\mathbf{J}_F)_{ij} = -\mathbb{E}_{x,u} \left[ \frac{\partial^2 l(x|u)}{\partial u_i \partial u_j} \right], \quad (\mathbf{J}_P)_{ij} = -\mathbb{E}_u \left[ \frac{\partial^2 l(u)}{\partial u_i \partial u_j} \right]
\]

where \( l(x|u) \) and \( l(u) \) are the conditional log likelihood and prior log likelihoods, respectively. In addition, the following regularity condition needs to be satisfied,

\[
\mathbb{E}_{x,u} \left[ \frac{\partial l(x|u)}{\partial u} \right] = 0
\]

\[
\mathbb{E}_{x,u} \left[ jk \sum_{i=1}^{N} d_i (x_i e^{jku^T d_i} - x_i^* e^{-jku^T d_i}) \right] = 0
\]

\[
jk \left( \sum_{i=1}^{N} d_i - \alpha^* \sum_{i=1}^{N} d_i \right) = 0
\]

In order to always satisfy this condition, we enforce the array element positions to be centered i.e., \( \sum_{i=1}^{N} d_i = 0 \).

For a single source, the FIM is given by,

\[
\mathbf{J}_F = -\frac{1}{\sigma^2} \mathbb{E} \left[ \frac{\partial s(\mathbf{u})^H}{\partial \mathbf{u}} \frac{\partial s(\mathbf{u})}{\partial \mathbf{u}} \right] = 2k^2 \gamma D^T D \tag{14}
\]

which depends only on the element positions, \( D \) and Signal to Noise ratio (SNR) \( (\gamma = |\alpha|^2/\sigma^2) \). Assuming the DoA prior to be uniformly distributed in the ROI \((\theta \leq 30^\circ)\), the prior FIM simplifies to \( \mathbf{J}_P = 1.343 I_2 \).

2) Ziv-Zakai Bound: The CRB is a local bound, which accounts for estimation performance dependent on mainbeam, hence it is only useful at high SNR. In order to better characterize the estimation performance of Sparse arrays at low SNR, we calculate the Ziv-Zakai Bound (ZZB) which incorporates the effect of sidelobes and predicts the threshold behavior. For any directional vector \( \alpha = [\cos \xi, \sin \xi]^T \), the ZZB is given by [25]

\[
\alpha^T \mathbf{R}_e \alpha \geq \int_0^{\infty} \int_{-\infty}^{\infty} \mathcal{V}(\cdot) \max_{\delta \in \Delta} \left\{ \int A(u, \delta) P_e(u, \delta) du \right\} dh
\]

where, \( A(u, \delta) = \min\{p(u), p(u + \delta)\}, \mathcal{V}(\cdot) \) is the valley filling function and \( P_e(u, \delta) \) is error probability of the following
vector parameter binary detection problem,
\[ H_0 : \hat{u} = u; \quad Pr(H_0) = \frac{1}{2}, x \sim p(x|u) \]
\[ H_1 : \hat{u} = u + \delta; \quad Pr(H_1) = \frac{1}{2}, x \sim p(x|u + \delta) \]
This error probability can be lower bounded by the following optimal non-coherent detector:
\[
\text{Decide}(u) = \begin{cases} 
    u & \text{if } \rho_1 > \rho_2 \\
    u + \delta & \text{if } \rho_1 < \rho_2
\end{cases}
\]
where \( \rho_1 = |x^H s(u)| \) and \( \rho_2 = |x^H s(u + \delta)| \). Given \( u = u_0, \rho_1, \rho_2 \) are rician distributed with scale parameter \( s = \sigma^2/M \) and non-centrality parameter \( \nu = |\alpha|N, |\alpha R(\delta)|N \) respectively where \( R(\delta) = R(\delta_x, \delta_y) \) is the beampattern from (3). The error probability is given by [26]
\[
P_{nc}(u, \delta) = \frac{1}{2} \left( Pr(\rho_1 < \rho_2 | u) + Pr(\rho_1 > \rho_2 | u + \delta) \right)
\]
\[
= Pr(\rho_1 < \rho_2 | u) \\
= Q_1(a, b) - \frac{1}{2} e^{-\frac{a^2 + b^2}{4}} I_0(ab)
\]
where,
\[
a = \sqrt{\frac{\gamma N}{2} \left( 1 - \sqrt{1 - |R(\delta)|^2} \right)}
\]
\[
b = \sqrt{\frac{\gamma N}{2} \left( 1 + \sqrt{1 - |R(\delta)|^2} \right)}
\]
which is not a function of \( u \). For ROI in our case, the maximum error \( h^{(\text{max})} = (a^T \delta)^{(\text{max})} = 1 \). Note that for a uniformly distributed DoA in spherical coordinates (\( \theta, \phi \)), the distribution of \( u \) is not uniform. However for simplicity of analysis, we make the assumption that \( u \) is uniformly distributed on a circular disc. Hence, the ZZB expression simplifies to
\[
a^T R_e a \geq \int_0^1 \int_{\delta=\hat{\delta}} A(u) dudP_{nc}(\delta) \quad \text{hhdh}
\]
\[
Z \bar{Z} \bar{B}(a^T R_e a) = \int_0^1 \int_{\delta=\hat{\delta}} P_{nc}(\delta) \quad \text{hhdh}
\]
The maximum error probability over all directions \( \delta \) cannot be expressed as closed form expression. However, due to the monotonicity of Marcum’s Q function, \( Q_1(.) \) and Bessel function of 0\( \text{th} \) order, \( I_0(.) \), the error probability in (16) is maximized only when \( R(\delta) \) is maximized. Therefore, for each values of \( h \), we compute the max\( a^T \delta = h \) \( |R(\delta)| \) numerically by searching over a discrete set of points on the line segment \( a^T \delta = h \) and substitute in (17).

B. DoA Estimation Algorithm

Grid-based sparse estimation for a set of DoAs models the the received signal (10) as follows:
\[
x = S(\Psi) b + z
\]
where \( S(\Psi) = [s(u_1) \ldots s(u_{|\Psi|})] \) contains the array response at discretized set of DoAs \( u_i \in \Psi \) as columns. The nonzero entries in \( b \) point to presence of target in the corresponding DoA in \( \Psi \). The DoA and gain pair \((\hat{u}_i, \hat{\alpha}_i)\)\( _{i=1}^{K} \) can be estimated by jointly minimizing the residual power,
\[
T(\hat{u}, \hat{\alpha}) = \left\| x - \sum_{j=1}^{K} \hat{a}_j s(\hat{u}_j) \right\|^2
\]
The NOMP algorithm summarized below provides a two stage estimator:

1) Detection: Using precomputed \( S(\Psi) \), coarse estimates of DoA and complex gain are obtained
\[
\hat{u} = \arg \max_{u \in \Psi} |s(u)^H x|^2
\]
\[
\hat{\alpha} = s(u)^H x / N
\]
2) Refinement: The estimates are refined using the Newton method:
\[
\hat{u}' = \hat{u} - (H_N T(\hat{u}, \hat{\alpha}))^{-1} \nabla T(\hat{u}, \hat{\alpha}) \quad (19)
\]
\[
\hat{\alpha}' = s(\hat{u}')^H x / N \quad (20)
\]
where \( H_N T \) and \( \nabla T \) denote the Hessian and gradient of \( T(u, \alpha) \) with respect to \( u \) at current estimate \((\hat{u}, \hat{\alpha})\) (see [27] for details).

The algorithm is repeated with the residual signal, \( r = x - \hat{\alpha}' s(\hat{u}') \) to estimate other DoAs. The refinement steps are repeated after each new detection for all DoAs in a cyclic manner for few rounds to improve accuracy.

The algorithm yields \( K_{\text{est}} \) DoA estimates, with estimation performance degrading when \( K_{\text{est}} \) does not match the true number of DoAs, \( K_{\text{DoA}} \). Hence, in order to evaluate the arrays independent of such errors, we implement both algorithm where \( K = K_{\text{DoA}} \) is known.

We also run extensive simulations with another state of the art algorithm, BPDN [17], with default parameters and 5 refinement stages. The computational complexity of BPDN is significantly higher than that of NOMP. The grid \( \Psi \) needs to be adapted at each iteration for BPDN, depending on the DoAs, whereas it remains fixed for NOMP \( S(\Psi) \), and can be precomputed, which makes it suitable for faster implementation in large arrays. Since the NOMP algorithm also yields somewhat better estimation accuracy than BPDN, we only present results obtained with NOMP here.

V. NUMERICAL RESULTS

A. Design of Arrays

Using the combinatorial search algorithm, we create a search space of array configurations, \( C \in \mathcal{C} \) of size \( |\mathcal{C}| = 657,000 \) for \( N_{\text{sub}} = 8 \) subarrays. Fig. 7(a) shows the values of two major objectives, MSLL, BN over this space. The beamwidth is improved by spreading out the subarrays, but this typically leads to a deterioration in the MSLL. Fig. 7(a) clearly shows the Pareto front corresponding to the multiple solutions of (9), corresponding to different relative weights, trading off these opposing
objectives. Fig. 7(b) shows the weighted cost function obtained for an example set of weights \((\alpha = 0.1, \beta = 0.5, \gamma = 0.1)\) as a function of the eigenvalues of the array covariance matrix. We observe that for a similar set of eigenvalues, many of the solutions we explore (shown in yellow) are substantially worse in terms of the weighted cost than the solutions shown in blue from which we choose our solutions, indicating the complexity of the optimization landscape. Choosing one more variable, the variance of the element locations \(\psi(C)\), for binning is therefore crucial for exploring this landscape more thoroughly. Fig. 7(c) shows the array obtained using the proposed algorithm, and its beampattern. We mark the locations of \(B\), and of two other Pareto optimal designs, \(A_1\) and \(A_2\) (to be discussed shortly) along the Pareto front in Fig. 7(a).

As we change the relative values of weights \(\alpha, \beta\) and \(\gamma\) in exploring the Pareto front, we can make the following observations regarding the corresponding subarray placements:

- For a larger relative value of \(\alpha\) (more importance given to beamwidth), the subarrays are widely distributed over the available aperture area.
- When we increase the relative value of \(\beta\) to suppress MSLL, the array becomes restricted to a smaller area.
- The relative weight of \(\gamma\), which corresponds to the objective of reducing eccentricity, affects both the shape of the main lobe and the positions of the sidelobes. For positive \(\gamma\), the solution is not expected to lie on the BW-MSLL Pareto front boundary. Indeed, Fig. 7(a) shows that the solutions from first stage dictionary search lie slightly away from this boundary. However, the second stage iterative refinement reduces this gap.

Based on these observations, we set the weights to obtain following sample array configurations:

1. \(A_1\): Primary emphasis is given towards minimizing beamwidth by setting \(\alpha = 1, \beta, \gamma = 0.1\).
2. \(A_2\): In this case, we emphasize all beam attributes by setting all weights equal to 1.

We compare these array designs against two simple array configurations, 1) a “compact” array where subarrays are placed together such that overall element pattern becomes a uniform rectangular array, 2) a “naive” array where subarrays are spread along a diamond shape such that its resultant beamwidth is equal to that of \(A_2\). Table I lists the weights and resulting beam attributes of these arrays.

Fig. 8 shows the array designs obtained using our optimization approach and their beam patterns. The \(A_2\) array has a sharp beamwidth and only 2.6 dB worse MSLL compared to the compact array. On the other hand, a naive sparse array with circular arrangement of subarrays yields 2.6 dB higher MSLL compared to \(A_2\) for similar beamwidth. Our designs \(A_1, A_2\) exhibit several small sidelobes (the highest sidelobe for \(A_2\) is \(-10.2\) dB), whereas the naive array exhibits fewer but more pronounced sidelobes. Since large sidelobes and grating lobes can cause large errors in DoA estimation, we expect our designs to yield better estimation performance, which is borne out by the results presented in the next section.

We note that the sidelobe levels and locations are primarily dependent on the super-array (i.e., the location of subarray centers \(C\)), while the subarray pattern mainly affects the large-scale beampattern. Therefore, we obtain similar solutions, with similar beam characteristics, using our optimization approach with small variations in the subarray element patterns (e.g., replacement of a rectangular pattern with a plus pattern). Such results are not reported here due to space constraints.

### B. Comparison of Estimation Performance

We evaluate the arrays based on their DoA estimation accuracy at different SNRs for both single and multiple source cases. We use the RMSE in estimating DoA for comparison which is given by \(\epsilon = \sqrt{\mathbb{E}[(\hat{\theta} - \theta_0)^2]}\) = \(\sqrt{\text{tr}(R_e)/2}\).

1. Estimation Bounds:
   - The Cramér Rao Bound is evaluated using (13), \(CRB(\epsilon) = \sqrt{\text{tr}(CRB(R_e))/2}\). The Ziv Zakai
Fig. 8. Beam patterns (Bottom row) for designed (Left half) & benchmarking (right half) arrays.

Fig. 9. Comparison of estimation theoretic bounds for arrays.

Bound is evaluated using (17),

$$ZZB(\hat{\epsilon}) = \sqrt{ZZB(a_1^TR_\epsilon a_1) + ZZB(a_2^TR_\epsilon a_2)}$$

where $a_1$, $a_2$ denote the directions of maximum and minimum beamwidths of the array. We also computed the ML estimation (MLE) error by Monte Carlo simulation using (12) with an overcomplete dictionary of array responses. Fig. 9 shows the CRB, ZZB and MLE curves for all the arrays. CRB is proportional to beamwidth (see Appendix B for details). The ZZB bound converges to CRB at the so-called “ZZB threshold” SNR: when the SNR is below this threshold, far-ambiguities in DoA estimation caused by large sidelobes dominate the MSE. The tradeoff between beamwidth and MSLL is thus expected to translate to one between CRB (better with smaller beamwidth) and ZZB threshold (worse with larger MSLL). Thus, as expected, “A1” array achieves the lowest CRB, followed by “naive” and “A2” with equal CRB, while “Compact” array has the largest beamwidth and hence highest CRB. The trend in MSLL is weakly reflected in the ZZB thresholds (for a single target, sidelobes do limited damage): the degradation in ZZB threshold, relative to that of the compact array for the optimized arrays (A1, A2) is less than 1 dB, while the gain in CRB due to smaller beamwidth is 4 dB and 2 dB, respectively. The MLE error curve also agrees with the threshold behavior predicted by ZZB. We see in the next set of results, however, that the size of the sidelobes becomes much more important when we consider multiple targets.

2) Estimation Algorithm Performance: We obtain DoA estimates using the NOMP algorithm [18], [27] with a known number of sources to compare the best case performance of these arrays. (The NOMP algorithm also performs as well as the brute force MLE for a single target discussed earlier—results omitted here.) The RMSE is evaluated across $N = 1024$ DoAs uniformly sampled over the ROI (spherical cap of half angle 30°). For evaluating the estimation performance in presence of multiple targets, ($K = 5$), we compute the RMS error in the DoA estimate for a primary target fixed at broadside, while interfering targets are distributed uniformly in ROI at separation of $\Delta u \geq 0.16$ or $\Delta \theta \geq 9.2^\circ$ away from primary target. This separation is imposed because the estimation problem is ill-posed for DoAs in close proximity. For uniform arrays, the minimum separation is typically defined with respect to the DFT bin size (e.g. $\Delta_{DFT} = 2\pi/L$ for an $L$-element linear array). Since this quantity cannot be defined for non-uniform planar arrays, we choose a minimum separation halfway between the RMS beamwidths of the “compact” and sparse arrays, to capture the effect of both local errors and far ambiguity errors due to sidelobes.
In addition to RMSE vs SNR curves, we also analyze the distribution of error magnitudes. The complementary cumulative distribution (CCDF) of the estimation errors is used to compare the “outage probability” corresponding to too large an error, which captures the impact of large sidelobes.

- With multiple sources, the estimation accuracy is degraded by interference from other sources, and RMSE does not converge to the single-target CRB. Fig. 10 shows the estimation performance for strong and weak interference.
  1) Weak interference: When the interfering sources are 6 dB weaker than primary target, sparse arrays offer more than 5 dB SNR gain compared to compact arrays for $SNR > -5$ dB. Also, the difference between $A_2$, naive array widens to about 1 dB in the threshold region indicating the benefit of suppressing sidelobes.
  2) Strong interference: When interfering sources have same magnitude, RMSE severely degrades for both arrays with high sidelobes ($A_1$, Naive) as well as arrays with high beamwidth (Compact). On the other hand, $A_2$ has lowest RMSE at $SNR > -5$ dB because of the dual benefit of small beamwidth and lower sidelobes.

- The increase in estimation errors at high SNR is attributed to ambiguity errors from sidelobes, hence the overall sidelobe suppression for the arrays can be compared using the distribution of these error magnitudes. Fig. 11 shows the CCDF curves of all arrays at $SNR = -5$ dB. The initial curvature of these curves (RMSE up to $-22$ dB) is expected to depend on local errors, hence the rate of change follows same order as CRB which is $A_1 > A_2 = naive > compact$. But the curvature reverses order at higher RMSE indicating the tradeoff with far-errors. We can see that $A_2$ achieves lower outage probability in both scenarios (e.g. for RMSE threshold set to $-15$ dB) as it strikes a balance between near and far errors. In contrast, both $A_1$ and naive exhibit high outage probability due to frequent far ambiguity errors caused by higher sidelobes.

Therefore, depending on the expected magnitude of interferers either one of the designed arrays with suitable sidelobe suppression can be selected. For a desired beamwidth reduction our design algorithm yields an array superior to a naively designed sparse array.

VI. COMPRESSIVE ESTIMATION

We now evaluate the arrays for sparse estimation using compressive measurements at each subarray given by:

$$y = \Phi x$$

where $x$ is the full measurement from (18) and $\Phi = \text{diag}(\Phi_1, \ldots, \Phi_{N_s})$ is the $M \times N$ measurement matrix consisting of the subarrays measurement matrices as its block diagonals. Each subarray takes $M_i$ compressive measurements with an independent $\Phi_i \in \mathbb{C}^{M_i \times N_s}$ whose elements are chosen uniformly and independently from QPSK samples $\{\pm 1, \pm j\}$. In addition, columns of $\Phi_i$ have unit norm to preserve signal norm on average ($\mathbb{E}[|\Phi S(u)|^2] = |S(u)|^2$) while scaling
uniform over $u \in \mathbb{R}$ is the error covariance matrix. Assuming that $R$ is an isometry $M > M/N$, the CRB is given by $\text{MSE} \simeq \frac{1}{2 \text{SNR}} \text{tr}(R_c)$. Comparing with Fig. 9, we observe that the estimation algorithms preserve the same characteristics as with full measurements with approximately 6 dB SNR penalty as expected ($N/M = 4$).

**VII. Conclusions**

Our results demonstrate that trading off beam width versus side lobes when synthesizing a large effective aperture does indeed produce performance gains in bearing estimation. Compared to a compact placement of subarrays, an optimized sparse placement produces smaller mean squared error because of its smaller beam width. Compared to a naive sparse placement, the control of side lobes via our optimized placement produces the most significant gains when estimating the bearing for multiple sources.

Exploring the application of our framework for communications, where transmit and receive beamforming gains are fixed by the number of elements, but control of beam width and side lobes affects interference, is an interesting direction. In the context of sensing, our work may be viewed as design of an individual sensor which can be placed within a more comprehensive architecture, such as a network of sensors for localization and tracking.

In our tiled architecture, the locations of antenna elements in a subarray are fixed once we specify the location of the subarray center. This constraint makes it difficult to adapt the extensive literature on sparse array optimization, which typically considers elements that can be freely placed, for our present purpose. The difficulty is compounded by the lack of closed form expressions for beam attributes of interest. However, since the objectives are only mildly dependent on the configuration of elements within a subarray, it might be possible to simplify the optimization problem, and possibly adapt ideas from the literature on sparse array optimization. This is an interesting direction for future research, especially given the importance of tiled architectures in realizing a large aperture leveraging low-cost hardware for subarrays with a moderate number of elements.

**Appendix A**

**Mean Square Error in 2D DOA Estimation**

For 2D DOA estimation, the error along any given angle $\xi$ is given by $a^T R_c a$, where $a = [\cos \xi, \sin \xi]^T$ is the directional cosine and $R_c$ is the error covariance matrix. Assuming that $\{q_i, q_i, i = 1, 2\}$, denote the eigenvalues and eigenvectors of $R_c$, the MSE averaged over $a$ (assume $\xi$ uniform over $[0, 2\pi]$) is given by

$$\text{MSE} = \mathbb{E}_a \left[ a^T R_c a \right] = \nu_1 \mathbb{E}_a \left[ |a^T q_1|^2 \right] + \nu_2 \mathbb{E}_a \left[ |a^T q_2|^2 \right]$$

$$= \nu_1 \frac{||q_1||^2}{2} + \nu_2 \frac{||q_2||^2}{2} = \frac{\nu_1 + \nu_2}{2} = \frac{1}{2} \text{tr}(R_c)$$

where we have used $\mathbb{E}[\cos^2 \xi] = \mathbb{E}[\sin^2 \xi] = \frac{1}{2}$.

**Appendix B**

**2D Beamwidth & CRB**

We define 2D beamwidth using the Taylor series expansion of beampattern $R_{\text{m}}(u)$ around mainlobe $R_{\text{m}}(0)$. Since the beampattern around the main lobe, and hence the beamwidth, is invariant to beamforming direction (see II-C), we assume $u_0 = 0$ without loss of generality, and drop the subscript: $R_{\text{m}}(u) \equiv R(u)$. By taking the derivatives of (3), the Taylor series
expansion up to second order is obtained as

$$R(u) \approx R(0) - \frac{k^2}{N} u^T D^T D u$$  \hspace{1cm} (22)

We define Half Power Beam Contour (HPBC) as the closed contour around mainbeam with \( \{ u: R(u) = 0.5 R(0) \} \), which is approximated as an ellipse using (22) as follows:

$$u^T D^T D u = \frac{N}{2k^2} R(0)$$  \hspace{1cm} (23)

Consider the eigendecomposition of \( D^T D \) given by,

$$D^T D = \lambda_1 p_1 p_1^T + \lambda_2 p_2 p_2^T \quad (\lambda_2 \geq \lambda_1)$$

The eigenvectors \( p_2, p_1 \) correspond to major and minor axis of HPBC ellipse respectively, and depend only on the element positions.

Fig. 14 shows the mainlobe of a beam and the dotted shaded region represents its HPBC ellipse. \( \hat{u}_o, \hat{u}_1^l, \hat{u}_2^l \) correspond to unit vectors in the direction of main beam, vertex and co-vertex of the HPBC where, \( \hat{u} = \{ u, v, \sqrt{1 - u^2 - v^2} \} \) is the unit vector towards directional cosine \( u = [u, v] \). These can be expressed as,

$$\hat{u}_o = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \hat{u}_1 = \begin{bmatrix} \sin \theta \cos \phi_{\text{max}} \\ \cos \theta \end{bmatrix}, \hat{u}_2 = \begin{bmatrix} \sin \varphi \cos \phi_{\text{min}} \\ \sin \varphi \sin \phi_{\text{min}} \end{bmatrix}$$

(24)

where \( \phi_{\text{max}}, \phi_{\text{min}} \) are perpendicular azimuthal angles and \( \theta, \varphi \) are the maximum and minimum beamwidth angles subtended from the major and minor axis of this ellipse to the mainbeam.

$$BW_{\text{max}} = \theta = \left( \frac{360}{\pi} \right) \cos^{-1} (\hat{u}_o \hat{u}_1)$$  \hspace{1cm} (25)

$$BW_{\text{min}} = \varphi = \left( \frac{360}{\pi} \right) \cos^{-1} (\hat{u}_o \hat{u}_2)$$  \hspace{1cm} (26)

Substituting the major and minor axis from (24) in (23), we obtain

$$\lambda_1 \sin^2 \theta = \frac{N}{2k^2} R(0) \Rightarrow \sin \theta \approx BW_{\text{max}} \propto 1/\sqrt{\lambda_1}$$

$$\lambda_2 \sin^2 \varphi = \frac{N}{2k^2} R(0) \Rightarrow \sin \varphi \approx BW_{\text{min}} \propto 1/\sqrt{\lambda_2}$$

That is, the beamwidths along extremal directions are inversely proportional to the square roots of the eigenvalues of \( D^T D \).

**Relation to CRB:** Using (14), the error covariance matrix is lower bounded by

$$R_e \geq \text{CRB} = J^{-1} = \frac{N}{2k^2\gamma} (D^T D)^{-1}$$

$$= \frac{N}{2k^2\gamma} \left( \frac{1}{\lambda_1} p_1 p_1^T + \frac{1}{\lambda_2} p_2 p_2^T \right) \quad \text{(using (} \lambda_2 \geq \lambda_1 \text{))}$$

Using Appendix A, the MSE can be lower bounded by

$$\text{MSE} \geq \text{CRB} = \frac{1}{2} \text{tr}(J^{-1}) = \frac{N}{4k^2\gamma} \left( \frac{1}{\lambda_1} + \frac{1}{\lambda_2} \right)$$

$$= \frac{N}{4k^2\gamma} (\sin^2(\theta) + \sin^2(\varphi))$$

$$\propto \frac{(BW_{\text{D}})^2}{\text{SNR}}$$

where \( BW_{\text{D}} = \sqrt{BW_{\text{max}}^2 + BW_{\text{min}}^2} = \sqrt{\theta^2 + \varphi^2} \) is defined as MSE beamwidth (\( \sin \theta \approx \theta \) for small angles \( \theta \)).

**APPENDIX C**

**VACANCY SEARCH OPERATOR \( \mathcal{T} \)**

Our reference subarray module shown in Fig. 15 occupies space in addition to antenna elements. In order to keep element polarizations aligned, these modules can be placed in either up (\( \theta' \)) or down (\( \theta'' \)) pose. We outline a procedure to list the vacant gridpoints \( V_i = \mathcal{T}(C_i^n) \) where the new subarray can be placed without overlapping with already placed \( \text{dormant} \) subarrays at \( C_i^n \). We define the subarray state as the center \( c \) of the element pattern and its pose \( \nu \), since vacancies depend on both parameters.

$$\hat{c} = \{ c, \nu \} \forall c \in C_i^n, V_i$$

The pose variable \( \nu \in \{ \nu^u, \nu^d, \nu^l \} \) denotes whether subarray can be placed in up only (\( \nu^u \)), down only (\( \nu^d \)) or free pose (\( \nu^l \)), either up or down) at the location \( c \). For a given set of \( \text{dormant} \) subarray states, \( C_i^n \) we identify all vacant states \( V_i \) for placing the new subarray. Once a new subarray is placed, the states of all dormant subarrays are updated (e.g., a free pose may switch to an up pose if the down pose becomes infeasible).

**APPENDIX D**

**PERTURBATION OF ARRAY**

In order to design a bin size for pruning array configurations, we analyze the effect of perturbing the location of a single array element on the eigenvalues of the array covariance matrix. Consider a small perturbation \( \nu = [\nu_x, \nu_y] \) added to \( i \)th array element position: \( d_i = d_i + \nu \). The covariance for the perturbed
array is

\[ \Sigma_D = (D^T D + \nu^T \nu + 2d^2) / N = \Sigma_D + \mathcal{G} + 2\mathcal{H} \]

where

\[ \mathcal{G} = \frac{1}{N} \begin{bmatrix} v_x^2 & v_x v_y & v_y^2 & v_x d_x & v_y d_y \\ v_x v_y & v_y^2 & v_x d_x & v_y d_y \end{bmatrix}, \quad \mathcal{H} = \frac{1}{N} \begin{bmatrix} v_x d_x & v_y d_x \\ v_x d_y & v_y d_y \end{bmatrix} \]

Using Weyl’s inequality [28] for real symmetric matrices, the eigenvalue perturbation is bounded as

\[ |\lambda_i - \lambda| \leq \|\mathcal{G} + 2\mathcal{H}\|_2 \leq \|\mathcal{G}\|_2 + 2\|\mathcal{H}\|_2 \leq \|\mathcal{G}\|_F + 2\|\mathcal{H}\|_F \]

The Frobenius norms of \( \mathcal{G}, \mathcal{H} \) are

\[ \|\mathcal{G}\|_F = (v_x^2 + v_y^2) / N \]

\[ \|\mathcal{H}\|_F = R_i \sqrt{(v_x^2 + v_y^2) / N} \]

where \( R_i = \sqrt{d_x^2 + d_y^2} \) is the distance of the \( i^{th} \) element from the array center. Hence, the overall variation of eigenvalues with variation \( \Delta_e = \sqrt{(v_x^2 + v_y^2)} \) of the \( i^{th} \) element is

\[ |\lambda_i - \lambda_i| \leq \frac{(2R_i + \Delta_e)}{N} \Delta_e \]

(27)

Thus, the eigenvalues are more sensitive to perturbations in the locations of elements further from the center. For perturbations within one grid size used in our placement search algorithm, the eigenvalue of the subarray center covariance can vary at most by \( \frac{(2R_i + \sqrt{2}\Delta_e)}{N_x} \sqrt{\tau\Delta_g} \), and we use this as a guideline for discretizing the eigenvalues for removing geometrically similar configurations.

**Appendix E**

**Algorithm Complexity**

The number of operations at the \( n^{th} \) iteration of Algorithm 1 is given by

\[ T_n = |\mathcal{E}^{n+1}| = \sum_{n=1}^{N_x-1} \sum_{i=1}^{C^n} |V_i| \leq |G||\mathcal{E}^n| \]

(28)

where \( \mathcal{E}^n \) denotes the set of leaf nodes at the \( n^{th} \) level of prefix tree, and \( |V_i| \) is the number of vacancies for the \( i^{th} \) leaf node. The vacancies are a subset of a grid with cardinality \( |G| = (2R_{max})^2 \) where \( R_{max} \) is the radius of the aperture and \( \Delta_g \) is the grid resolution.

The maximum number of array configurations (which correspond to leaves of the prefix tree) in any given iteration is bounded by the maximum number of unique triplets \((\lambda_1, \lambda_2, \psi)\), which can be expressed as follows:

\[ |\mathcal{E}^n| \leq |\mathcal{E}^{n+1}| = \frac{\lambda_{max}^1 \lambda_{max}^2 \psi_{max}}{\tau_{min}} \]

(29)

where \( \lambda_{max}^1, \lambda_{max}^2, \psi_{max} \) represent the maximum value of each parameter and \( \tau_{min} = \frac{2\Delta_g}{N_x} \) is the minimum bin resolution (using (27)). The eigenvalues of the array covariance are bounded by the maximum aperture radius \( R_{max}^1 < R_{max}^2 \) and the array separation variance can be bounded as

\[ \psi = \mathbb{E}[(l_{ij} - \mathbb{E}[l_{ij}])^2] \leq R_{max}^2 \]

Substituting these upper bounds in (29) and (28), we obtain

\[ T_n \leq |G||\mathcal{E}^n| \leq \frac{N^3}{\tau_{min}^3} \frac{R_{max}^2}{2\Delta_g^2} \]

Note that the upper bound derived here is conservative, as can be readily verified through simulations. Fig. 16 shows the time and space complexity bounds over \( N_x = 8 \) iterations for a sample run of the algorithm with \(|G| = 1600 \) grid points. We observe that in practice, computational complexity increases up to \( N_x = 4 \) but starts reducing afterwards. The latter is because of the reduction in the number of vacancies in the aperture as the space occupied by the existing subarray modules increases.

**References**


DESIGN OF LARGE EFFECTIVE APERTURES FOR MILLIMETER WAVE SYSTEMS USING A SPARSE ARRAY OF SUBARRAYS 6497

(2008) and Proc. IEEE 5th Int. Conf.


IEEE Antennas Propag. Mag.


Anant Gupta received the B.Tech. degree in elec-

tronics and electrical communication engineering and the M.Tech. degree in telecommunication systems engineering from IIT Khargpur, Khargpur, India, in 2013. He received the M.S. degree in electrical and computer engineering from the University of California at Santa Barbara (UCSB), Santa Barbara, CA, USA, in 2016. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, UCSB. His research interests include wireless sensing, statistical signal processing and machine learning.


Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on November 08,2020 at 22:01:35 UTC from IEEE Xplore. Restrictions apply.