Fast Unit-Modulus Least Squares with Applications in Beamforming

John Tranter, Nicholas D. Sidiropoulos*, Xiao Fu, and A. Swami

***Abstract*—Unit-modulus least squares (ULS) problems arise in many applications, including phase-only beamforming, sensor network localization, synchronization, phase retrieval and radar code design. ULS formulations can always be recast as unit-modulus quadratic programs (UQPs), to which semi-definite relaxation (SDR) can be applied, and is often the state-of-the-art approach. However, SDR lifts the problem dimension (i.e., the number of variables) from N to N^2, which drastically increases the memory burden and computational cost when the problem size is already large – e.g., when designing phase-only beamformer weights for massive multiple-input-multiple-output (MIMO) systems. This work focuses on scalable first-order algorithms for the ULS problem and some of its variants. It advocates using simple gradient projection (GP) as a starting point for solving the ULS problem, establishes global convergence of GP to a Karush-Kuhn-Tucker (KKT) point for this NP-hard problem, and bounds its iteration complexity. Then it proposes ULS extensions tailored to reflect practical beamformer design objectives, bringing in and exploiting new degrees of freedom to improve the beampattern designs. Simple variants of GP are proposed to deal with these extended ULS problems. Simulations are used to showcase the effectiveness of the proposed algorithms in both the plain ULS problem and in the context of phase-only beamforming.**

***Index Terms*—Unit-modulus least squares, unit-modulus quadratic programming, MaxCut, constant modulus beamforming, per-antenna power constraint, massive MIMO.**

I. Introduction

Unit-modulus least squares (ULS) optimization problems have many contemporary engineering applications. Signal processing applications of ULS include sensor network localization [2], phase retrieval [3, 4], and radar code design [5–7]. Phase-only beamforming restrains the power of each antenna to be a constant and uses the phases of the weights associated with the antenna elements to form the desired beampattern. Designing such beamformers can also be posed as a ULS problem. Phase-only arrays have gained renewed attention recently [8–11]; the driving force behind this resurgence is potential uses in massive multiple-input-multiple-output (MIMO) systems, where it is costly and impractical to employ a separate power amplifier for each antenna, and there is a need to control the peak-to-average power ratio (PAPR) as well.

Although the ULS/UQP problem is non-convex and NP-hard in its general form [12] due to the unit-modulus constraints, it is also a special case of non-convex quadratically constrained quadratic programming (QCQP), to which a popular relaxation technique known as semi-definite (rank) relaxation (SDR) [13] can be readily applied. When SDR returns a rank-one solution, this is optimal for the original problem as well. When SDR returns a higher-rank solution, many approaches can be used to obtain a feasible solution to the ULS problem, e.g., randomization [13].

Although SDR has been successively applied to a large variety of applications, a major drawback is that it is not well-suited for large-scale problems. If the original ULS/UQP problem has N optimization variables, SDR requires lifting the problem to N^2 dimensions. As a result, the resulting computational complexity is $\mathcal{O}(N^7)$ flops if a general-purpose interior-point solver is used. Additionally, this approach is quite inefficient in terms of memory usage. Storage of N^2 variables may be impractical and costly when N is large. On the practitioner's side, many modern applications of ULS/UQP do have a large number of variables and require scalable optimization methods in terms of both computational resources and memory. For example, MIMO communication systems have been of interest for over 15 years, due to the performance improvements that they enable. Arrays with multiple antennas enable higher data rates, as well as longer reach and improved link reliability [14, 15]. At present, *massive* MIMO systems promise to play a major role in the evolution toward fifth-generation (5G) wireless technology. One of the emerging 5-G paradigms envisions equipping each base station with many more antennas than the number of active users in its service cell [16]. Applications such as radar code design may likewise entail very high-dimensional optimization, especially in high-resolution scenarios with many antennas. Because of the high-dimensional nature of these problems, many of the previously developed approximation methods like SDR become impractical. Therefore, there is substantial motivation to find more memory-efficient and computationally advantageous alternatives for dealing with the ULS problem.

Our work is motivated by the problem of phase-only beamformer design for massive MIMO systems, where computationally heavy algorithms like SDR are no longer practical due to their high computational and memory costs. A desired spatial beampattern is typically synthesized by appropriately picking the modulus and phase of each element of the complex-valued beamforming vector. This corresponds to using a separate phase shifter and power amplifier for each antenna, which is costly. In many applications, especially

* Corresponding author. Original manuscript submitted to *IEEE Trans. on Signal Processing* Feb. 10, 2016; revised December 8, 2016. Earlier conference version of part of this work appears in *Proc. EUSIPCO 2016* [1]. J. Tranter, N. Sidiropoulos, and X. Fu are with the Department of Electrical and Computer Engineering, University of Minnesota—Twin Cities, Minneapolis, MN, 55455 USA; e-mail: (trant004,nikos,xfu)@umn.edu. A. Swami is with the Army Research Laboratory, Adelphi, MD, 20783 USA; e-mail: a.swami@ieee.org. J. Tranter and X. Fu were supported in part by ARO STIR W911NF-15-1-0384.

those involving 'massive' antenna arrays and/or small form-factor / low power mobile devices, it is preferable to use a single power amplifier for all antennas, and rely on per-antenna phase shifters to steer the beam in the direction(s) of interest. This gives rise to a constant-modulus constraint on the beamforming vector. Several algorithms have been proposed for phase-only beamforming, such as phase perturbation methods [17, 18]. Thompson [5] utilized a gradient-search method (with an explicit angle parametrization of the unit-modulus constraint) for adaptively adjusting the phase shifters. Smith [19] proposed a combination of conjugate gradient and Newton algorithms for phase-only adaptive nulling, where optimization is performed over the N-dimensional unit-modulus torus (with an angle paramaterization of the unit-modulus constraint). The conjugate gradient algorithm is employed to first approach the solution, followed by multiple Newton refinements. The algorithm requires calculation and inversion of the Hessian matrix at every iteration, and thus is not ideal for large-scale problems. Choi and Sarkar [20] devised a direct data domain least squares algorithm, which adaptively adjusts the phase weights from snapshots of complex voltages at each antenna element via a conjugate gradient algorithm (also with an angle parametrization).

A common denominator of early approaches is that they employ an explicit angle parametrization of the unit-modulus constraint that enables classic unconstrained optimization methods such as gradient descent to be applied to this non-convex and (NP-)hard problem. SDR [13] is a much more advanced approach for this type of problem, and often exhibits superior performance in practice. A closely related phase-only beamforming formulation employing a linearly constrained minimum variance (LCMV) criterion has been considered in Lu *et al.* [6], who proposed to apply SDR as its approximation.

Motivated by applications in unit-modulus (constant-envelope) radar code design, Soltanalian and Stoica [7] proposed a monotonically error-bound improving technique (MERIT), as well as a power method-like iteration for UQP optimization problems. MERIT is comparatively complex, but it provides a sub-optimality guarantee that is sometimes tighter than that provided by SDR. The power-like iteration in [7] can be used to improve any initial estimate at a relatively low (second-order) cost, but the ultimate result depends a lot on initialization.

Contributions: In this paper, we propose several low-complexity first-order algorithms for ULS/UQP and certain extended ULS/UQP problems that arise in beamforming. We begin with the classical ULS problem and propose to employ a simple gradient projection (GP) algorithm to handle it when the problem size is large. The motivation is twofold:

- First-order methods are well-suited for large-scale problems since they avoid computing the Hessian and its inverse in each iteration, and thus have much lower per-iteration complexity relative to interior-point methods. They also do not lift the problem dimension as SDR does, and thus a lot of memory and computational resources can be saved. Gradient-based methods can also exploit signal and data sparsity, which is critical when dealing with big problems.
- Projection to the unit-modulus manifold is easy – this operation can be expressed in closed form and requires almost negligible computational cost.

On the other hand, applying GP to nonconvex constraints may seem naive – there is no guarantee of decreasing the cost, and one may even risk divergence. Despite these apparent difficulties, using some nice properties of the ULS cost function such as Lipschitz continuity of its gradient, we show that the solution sequence of the proposed algorithm globally converges to a Karush-Kuhn-Tucker (KKT) point of the original NP-hard problem. Here, global convergence refers to convergence of the whole solution sequence produced by an iterative algorithm, as opposed to convergence of subsequences of the solution iterates. Subsequence convergence results (e.g., every limit point is a KKT point) are more often encountered in the literature, however these are weaker compared to global converegence. In addition, we carefully analyze the iteration complexity of getting to a KKT point using the proposed GP algorithm. Our analysis shows that a metric that measures the optimality gap between the current iterate and a KKT point shrinks to at most $\mathcal{O}(1/T)$ after T iterations.

Our second contribution lies in algorithms that are tailored for phase-only array beamforming for massive MIMO systems. We consider a scenario where a unit-modulus complex weight vector should be designed for a large number of antennas such that a pre-specified transmit beampattern (e.g., "pencil" beams or sector beams) is synthesized. The weights are constrained to have the same modulus since a common power amplifier is used to drive all transmit antennas, and one relies only on antenna phase shifters to realize the desired spatial beampattern. Rather than sticking to plain ULS, more appropriate application-specific formulations are introduced that take into account additional degrees of freedom in global scaling and the spatial phase response, which yield enhanced performance in terms of beampattern synthesis accuracy. Following the insight of GP for ULS, alternating optimization algorithms that iterate between updating the antenna weights and the scaling and phase response factors are proposed. The update of the weight vector is a simple GP step, as for plain ULS; the subproblem with respect to the other variables can be solved to optimality. Convergence properties of these algorithms are also discussed. We exemplify the comparative advantages of these first-order methods (both in terms of cost minimization and runtime complexity) relative to existing popular approaches such as SDR and MERIT.

An earlier conference version of part of this work appears in [1]. Relative to [1], this journal version includes more mature convergence analysis (global convergence, iteration complexity) that is also more broadly applicable; proofs (which were entirely missing from [1]); and a comprehensive suite of experiments.

Notation We use $\mathbf{X}$, $\mathbf{x}$ and x to denote a matrix, a vector, and a scalar, respectively; ^{H}, * and ^{T} are used to denote Hermitian, conjugate and transpose operators, respectively; $\lambda_{\max}(\mathbf{X})$ denotes the largest eigenvalue of $\mathbf{X}$; $^{-1}$ and † denotes the inverse and pseudo-inverse operator, respectively; $\mathbf{x} \circledast \mathbf{y}$ is the element-wise (Hadamard) product of $\mathbf{x}$ and $\mathbf{y}$;

Diag($\mathbf{x}$) is a diagonal matrix that holds the elements of $\mathbf{x}$ to be the diagonal elements.

II. Motivation, Problem Statement and Prior Art

A. Motivation and Problem Formulation

In receive beamforming, complex-valued weights are applied to the signals coming from the different antennas before combining them, to create a desired spatial beampattern that enhances signals from specified directions of interest and attenuates those from other directions. In transmit beamforming, a common information-bearing signal is fed to multiple antennas, using a different complex weight (corresponding to separate power amplification and carrier phase shift) per antenna. Transmit beamforming can be *unicast* (pointing to a single receiver of interest) or *multicast* (pointing to multiple receivers, interested in a common information stream). In both cases, the objective is to steer power in the direction(s) of interest while mitigating interference to other users within transmission range. Although many classical beamforming scenarios apply both magnitude and phase weighting to the antenna elements, there are many advantages to phase-only beamforming [6, 21, 22], especially for massive MIMO systems where phase-only beamforming is important from a hardware complexity, cost, and form factor (size/weight) point of view, as it does not require a separate power amplifier for each up/down conversion chain, it can alleviate peak-to-average power ratio (PAPR) problems, and improve energy-efficiency [8–11].

For simplicity, let us consider a uniform linear array (ULA) comprising N antennas with equidistant spacing ($\lambda/2$, where λ is the wavelength of the carrier frequency) (our formulation can handle any beamforming scenario with a known array manifold). Let $\boldsymbol{\theta}$ denote an $M \times 1$ vector representing a discretization of the angle space, i.e.,

$$\boldsymbol{\theta} = \left[0, \frac{2\pi}{M}, \frac{4\pi}{M}, \ldots, \frac{2(M-1)\pi}{M}\right]. \tag{1}$$

In a ULA scenario, the $N \times 1$ steering vectors have Vandermonde structure $\mathbf{a}(\theta_i) = [1, e^{-j\theta_i}, e^{-j2\theta_i}, \ldots, e^{-j(N-1)\theta_i}]^T$. We can then construct our design matrix $\mathbf{A} = [\mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_M)]^H$, and formulate the following ULS optimization problem

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{C}^N} \quad & \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 \\ \text{subject to} \quad & |w_i|^2 = 1, i = 1, \ldots, N, \end{aligned} \tag{2}$$

where w_i is the i-th element of the beamforming weight vector $\mathbf{w}$, and $\mathbf{y}$ denotes our desired target response (see section III-B for details regarding the choice of $\mathbf{y}$). Figures 1 and 2 illustrate an example multicast beampattern aimed towards two users in the directions $\{-20^o, +20^o\}$; note that the geometry of the ULA scenario will result in symmetry about the axis defined by the ULA.

In this work, we begin by considering Problem (2), motivated by phase-only beamformer design for arrays comprising many antennas. Problem (2) has numerous other applications, so our results are of broader interest; but phase-only beamforming is fertile ground for pertinent extensions as well. We propose two custom formulations that take into account additional degrees of freedom in global scaling and the spatial phase response, resulting in better beampattern synthesis accuracy.

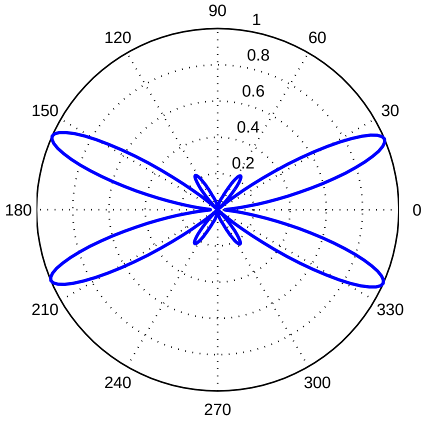

Fig. 1: Polar beam pattern example, $M = 360$, $N = 16$ for angles $\{-20^o, +20^o\}$

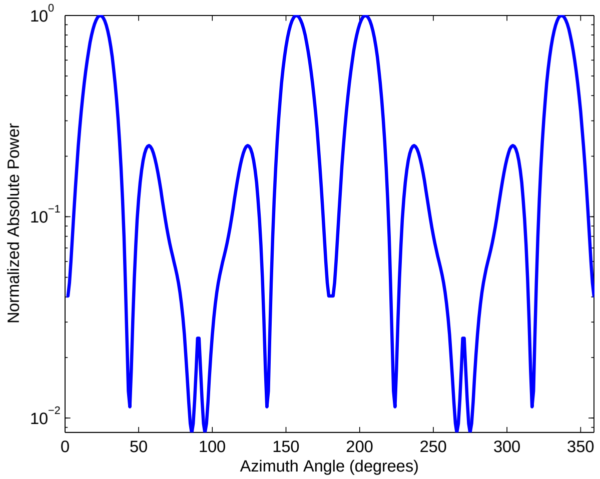

Fig. 2: $|\mathbf{a}^H(\theta_i)\mathbf{w}|^2$ for $M = 360$, $N = 16$ for angles $\{-20^o, +20^o\}$

B. Prior Art: Semidefinite Relaxation and Others

It is well-known that the ULS problem in (2) can be transformed to the following UQP [13]

$$\begin{aligned} \min_{\bar{\mathbf{w}}\in\mathbb{C}^{N+1}} \quad & \bar{\mathbf{w}}^H\mathbf{R}\bar{\mathbf{w}} \\ \text{subject to} \quad & |\bar{w}_i|^2 = 1, i = 1, \ldots, N+1, \end{aligned} \tag{3}$$

where

$$\mathbf{R} := \begin{bmatrix} \mathbf{A}^H\mathbf{A} & -\mathbf{A}^H\mathbf{y} \\ -\mathbf{y}^H\mathbf{A} & \mathbf{y}^H\mathbf{y} \end{bmatrix}, \quad \bar{\mathbf{w}} := \begin{bmatrix} \mathbf{w} \\ t \end{bmatrix}. \tag{4}$$

Conversely, UQP can likewise be expressed as ULS if $\mathbf{R}$ is positive semidefinite. To see this, consider the square root decomposition of a positive semidefinite matrix $\mathbf{R} = \mathbf{R}^{H/2}\mathbf{R}^{1/2}$ and the following partitioning:

$$\mathbf{R}^{1/2} = [\mathbf{A}, -\mathbf{y}].$$

Consequently, we have

$$\begin{aligned}\bar{\mathbf{w}}^H\mathbf{R}\bar{\mathbf{w}} =& \begin{bmatrix}\mathbf{w}^H e^{-j\theta} & e^{-j\theta}\end{bmatrix}\begin{bmatrix}\mathbf{A}^H\mathbf{A} & -\mathbf{A}^H\mathbf{y} \\ -\mathbf{y}^H\mathbf{A} & \|\mathbf{y}\|_2^2\end{bmatrix}\begin{bmatrix}\mathbf{w}e^{j\theta} \\ e^{j\theta}\end{bmatrix} \\ =& \mathbf{w}^H\mathbf{A}^H\mathbf{A}\mathbf{w} - \mathbf{y}^H\mathbf{A}\mathbf{w} - \mathbf{w}^H\mathbf{A}^H\mathbf{y} + \|\mathbf{y}\|_2^2. \quad (5)\end{aligned}$$

Even if $\mathbf{R}$ is not positive semidefinite, it is easy to see that, owing to the unit-modulus constraint, one can add diagonal loading to make it positive semidefinite, without changing the problem.

To deal with UQP, the most popular approach is arguably semidefinite relaxation (SDR) [13]. Consider the UQP problem

$$\begin{aligned}\min_{\mathbf{w}\in\mathbb{C}^N} \quad & \mathbf{w}^H\mathbf{R}\mathbf{w} \qquad (6) \\ \text{subject to} \quad & |w_i|^2 = 1, \quad i = 1, \ldots, N.\end{aligned}$$

Writing $\mathbf{w}^H\mathbf{R}\mathbf{w} = \text{Trace}(\mathbf{w}^H\mathbf{R}\mathbf{w}) = \text{Trace}(\mathbf{R}\mathbf{w}\mathbf{w}^H)$, we can define a matrix $\mathbf{W} := \mathbf{w}\mathbf{w}^H$ and equivalently write (6) as

$$\begin{aligned}\min_{\mathbf{W}} \quad & \text{Trace}(\mathbf{R}\mathbf{W}) \qquad (7) \\ \text{subject to} \quad & \mathbf{W}_{ii} = 1, \quad i = 1, \ldots, N \\ & \mathbf{W} = \mathbf{w}\mathbf{w}^H.\end{aligned}$$

The constraint $\mathbf{W} = \mathbf{w}\mathbf{w}^H$ is equivalent to $\mathbf{W} \succeq \mathbf{0}$ (positive semi-definite) *and* rank$(\mathbf{W}) = 1$. The rank$(\mathbf{W}) = 1$ constraint is non-convex; dropping it yields a relaxation form that is convex. However, the optimal $\mathbf{W}_o$ of the relaxed problem will (in general) not be rank one. To extract a feasible solution of (6) from $\mathbf{W}_o$, one can follow the randomization procedure in [23].

SDR has been very successful in dealing with nonconvex QCQP problems including UQP. However, in the era of massive antenna arrays, implementing SDR is challenging. First, SDR lifts the problem dimension to N^2, which means that a general-purpose interior point method needs $\mathcal{O}(N^7)$ flops and $\mathcal{O}(N^2)$ memory to solve the corresponding SDP. These demands make SDR unappealing for applications like massive MIMO systems – although there are faster solvers for special SDP problems like (7) [24], their flop count is still heavy, and the memory cost cannot be reduced. In the following sections, we will propose algorithms that do not lift the problem dimension and only use first-order information for handling the ULS/UQP problem. Consequently, the computational and memory burdens are substantially alleviated.

Before moving on, we note that earlier work on phase-only beamforming [5, 19] considered an explicit $w_i = e^{j\theta_i}$ parametrization of the unit modulus constraint, and proposed using unconstrained derivative-based methods, such as gradient descent and Newton's method to address the resulting optimization problem. Algorithms in this genre are generally simpler than SDR. However, such explicit parametrization changes the cost function from a nice convex quadratic to a non-convex one, and does not seem to work well in our experience.

III. Proposed algorithms

A. Gradient Projection (GP)

Different from earlier attempts, we propose keeping the unit-modulus constraint and using *projected* gradient descent (or, gradient projection (GP)) instead of unconstrained gradient descent. The details are provided in Algorithm 1.

Algorithm 1 Gradient Projection

```
1: Initialization: Set $k = 0$, $\alpha = \frac{\beta}{\lambda_{\max}(\mathbf{A}^H\mathbf{A})}$, $\beta \in (0,1)$, $\mathbf{w}^{(0)} = e^{j\angle(\mathbf{A}^\dagger\mathbf{y})}$
2: Repeat
3:     $\boldsymbol{\zeta}^{(r+1)} = \mathbf{w}^{(r)} + \alpha\mathbf{A}^H\left(\mathbf{y} - \mathbf{A}\mathbf{w}^{(r)}\right)$;    (Gradient)
4:     $\mathbf{w}^{(r+1)} = e^{j\angle(\boldsymbol{\zeta}^{(r+1)})}$;    (Projection)
5:     $r = r + 1$;
6: until convergence
```

Algorithm 1 is nothing but a GP algorithm – what is special is that the projection step involves a non-convex set – the element-wise unit modulus constraint, in particular. In Algorithm 1, α is the step size along the opposite direction of the gradient. The motivation of Algorithm 1 is simple: gradient descent has the advantage of scalability, and is able to exploit data (i.e., $\mathbf{A}$ and $\mathbf{y}$) sparsity. In addition, it is much more memory-efficient relative to SDR. These traits are well-suited for large-scale problems. In addition, projection onto a unit modulus constraint admits a closed-form solution (cf. line 4 in Algorithm 1), and the entire procedure can be carried out very efficiently. On the other hand, the concern is that projection onto a non-convex set may in fact increase the cost value, and thus tends to be problematic in terms of optimization.

We wish to show that the solution sequence of $\mathbf{w}^{(r)}$ converges to a KKT point. To that end, let us consider a real-valued parametrization of the ULS problem [13], i.e.,

$$\tilde{\mathbf{y}} = \begin{bmatrix}\text{Re}\{\mathbf{y}\} \\ \text{Im}\{\mathbf{y}\}\end{bmatrix}, \tilde{\mathbf{w}} = \begin{bmatrix}\text{Re}\{\mathbf{w}\} \\ \text{Im}\{\mathbf{w}\}\end{bmatrix}, \ \tilde{\mathbf{A}} = \begin{bmatrix}\text{Re}\{\mathbf{A}\} & -\text{Im}\{\mathbf{A}\} \\ \text{Im}\{\mathbf{A}\} & \text{Re}\{\mathbf{A}\}\end{bmatrix},$$

leading to

$$\begin{aligned}&\min_{\tilde{\mathbf{w}}\in\mathbb{R}^{2N}} \ (1/2)\left\|\tilde{\mathbf{y}} - \tilde{\mathbf{A}}\tilde{\mathbf{w}}\right\|_2^2 \\ &\text{subject to } \tilde{w}_n^2 + \tilde{w}_{N+n}^2 = 1, \ n = 1, \ldots, N.\end{aligned} \qquad (8)$$

Note that Problems (2) and (8) are equivalent, and applying GP to Problem (8) results in exactly the same algorithm as in Algorithm 1. Therefore, in the following, we will analyze convergence of GP applied to Problem (8) for notational simplicity. The KKT conditions of Problem (2) are

$$\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\tilde{\mathbf{w}} - \tilde{\mathbf{A}}^T\tilde{\mathbf{y}} + 2\boldsymbol{\lambda} \circledast \tilde{\mathbf{w}} = \mathbf{0}, \qquad (9a)$$

$$\tilde{w}_n^2 + \tilde{w}_{N+n}^2 = 1, \ n = 1, \ldots, N, \qquad (9b)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{2N}$ is a Lagrangian multiplier. Let us define

$$\begin{aligned}Q(\tilde{\mathbf{w}}^{(r+1)}, \tilde{\mathbf{w}}^{(r)}, \boldsymbol{\lambda}^{(r+1)}) =& \|\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\tilde{\mathbf{w}}^{(r)} - \tilde{\mathbf{A}}\mathbf{y} \\ &+ 2\boldsymbol{\lambda}^{(r+1)} \circledast \tilde{\mathbf{w}}^{(r+1)}\|_2^2.\end{aligned}$$

We first show that

Lemma 1: Assume that $\mathbf{A}$ and $\mathbf{y}$ are bounded. Then, $Q(\tilde{\mathbf{w}}^{(r+1)}, \tilde{\mathbf{w}}^{(r)}, \boldsymbol{\lambda}^{(r+1)}) \to 0$ implies that $\tilde{\mathbf{w}}^{(r)}$ approaches a KKT point of Problem (8).

The proof can be found in Appendix A. According to Lemma 1, the speed of $Q(\tilde{\mathbf{w}}^{(r+1)}, \tilde{\mathbf{w}}^{(r)}, \boldsymbol{\lambda}^{(r+1)})$ approaching zero can be used as a measure of the iteration complexity. Based on this observation, we provide the following theorem:

Theorem 1: Assume that $\alpha < 1/L$, where $L = \lambda_{\max}(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}})$. Then, Algorithm 1 has the following convergence properties:

a) (Global Convergence) The whole solution sequence $\{\tilde{\mathbf{w}}^{(r)}\}$ converges to a set $\mathcal{K}$ which consists of all the KKT points of Problem (8).

b) (Iteration Complexity) Assume that $\epsilon > 0$ is some small number, and T is the number of iterations that are needed for $Q(\tilde{\mathbf{w}}^{(r+1)}, \tilde{\mathbf{w}}^{(r)}, \boldsymbol{\lambda}^{(r+1)}) \leq \epsilon$ to hold for the first time. Then, there exists a constant v such that

$$\epsilon \leq \frac{v}{T-1};$$

that is, the algorithm converges to a KKT point at least sublinearly.

The proof can be found in Appendix B. Theorem 1 assures that the solution sequence converges to a meaningful point. In the proof, one can see that when $\alpha < 1/L$, GP in fact reduces the objective at each iteration, which is another desired property in practice. More interestingly, the *b)* part gives a time complexity bound for GP to shrink the gap between the current iterate to a KKT point – it shows that for a certain measurement of the gap (i.e., the Q-function in Lemma 1) to reach ϵ, the number of iterations needed is at most $\mathcal{O}(1/\epsilon)$. We should mention that this bound is based on worst-case analysis and the algorithm may perform much faster in practice. We also note that the convergence of GP-like methods for manifold-constrained QP was also considered in [25–27], where specific QP problems, e.g., sparse principal component analysis, were considered. The insights of the convergence proofs in [25–27] can be modified to show part a) of Theorem 1. Our new proof of part a), however, takes a successive upper bound minimization viewpoint, and thus can potentially cover many more applications and more general majorization minimization algorithms than GP.

B. Auto-scaling formulation

Algorithm 1 is simple and effective in addressing general ULS/UQP problems, as we will show in Sec. IV. However, in the beamformer design problem, there is a subtle factor that greatly affects the performance, namely, the scaling of $\mathbf{y}$. Consider the following choice of $\mathbf{y}$ for receive beamformer design:

$$y_i = \begin{cases} 1 & \text{if } i \in \mathcal{J}, \\ 0 & \text{otherwise}, \end{cases} \tag{10}$$

where $\mathcal{J}$ denotes the set containing the indices i corresponding to the direction(s) of interest in $\boldsymbol{\theta}$. The $\mathbf{y}$ in (10) specifies the beampattern that we want to produce. However, note that

$$|\mathbf{a}^H(\theta_i)\mathbf{w}| = |\sum_{j=1}^{N} a_j^*(\theta_i)w_j| \leq \sum_{j=1}^{N} |a_j^*(\theta_i)w_j| = N, \tag{11}$$

and, from the Cauchy-Schwarz inequality, the maximum is achieved if and only if $\mathbf{w} = e^{j\epsilon}\mathbf{a}(\theta_i)$. Thus there is an upper bound on the array gain in any direction, and if we ask for the highest possible gain in one direction we completely lose all degrees of freedom to shape the beampattern in other directions. In practice, it is more common that multiple angles are of interest, and the 'optimal scaling' of $\mathbf{y}$ is unclear under such circumstances. In general, what we often desire is a good *relative* beampattern that concentrates power in the directions of interest; to that end, we can introduce an additional scaling variable $s \in \mathbb{C}$ to obtain

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{C}^N, s\in\mathbb{C}} \quad & \|\mathbf{y} - s\mathbf{A}\mathbf{w}\|_2^2 \\ \text{subject to} \quad & |w_i|^2 = 1, i = 1, \ldots, N. \end{aligned} \tag{12}$$

Variable s can be regarded as an 'automatic normalization' factor, which addresses the aforementioned scaling issue. Note that, by separability, we may compute and substitute the optimal s as a function of $\mathbf{w}$:

$$s_{opt} = \frac{\mathbf{w}^H\mathbf{A}^H\mathbf{y}}{\|\mathbf{A}\mathbf{w}\|^2}, \tag{13}$$

Using (13), we can 'embed' the tuning of s in each iteration of Algorithm 1, leading to Algorithm 2. This modified algorithm is an instance of alternating optimization (AO) with respect to (w.r.t.) $\mathbf{w}$ and s. For the subproblem w.r.t. $\mathbf{w}$, we solve it inexactly by taking a projected gradient step. Since s can be easily computed, Algorithm 2 has only a marginal complexity increase relative to Algorithm 1, but can better address the scaling issue in beampattern design.

Algorithm 2

```
Initialization: Set $k = 0$, $\beta \in (0,1)$, $\mathbf{w}^{(0)} = e^{j\angle(\mathbf{A}^\dagger\mathbf{y})}$
Repeat
    $s^{(r+1)} = (\mathbf{w}^{(r)})^H\mathbf{A}^H\mathbf{y}/\|\mathbf{A}\mathbf{w}^{(r)}\|_2^2$
    $\alpha^{(r+1)} = \beta/\lambda_{\max}(|s^{(r+1)}|^2\mathbf{A}^H\mathbf{A})$
    $\boldsymbol{\zeta}^{(r+1)} = \mathbf{w}^{(r)} + \alpha^{(r+1)}(s^{(r+1)})^*\mathbf{A}^H(\mathbf{y} - s^{(r+1)}\mathbf{A}\mathbf{w}^{(r)})$;
    $\mathbf{w}^{(r+1)} = e^{j\angle(\boldsymbol{\zeta}^{(r+1)})}$;
    $k = k + 1$
until convergence
```

It is interesting to investigate the convergence properties of Algorithm 2. This algorithm can be considered as an *inexact* alternating optimization approach; i.e., we alternate between solving subproblems with respect to s and $\mathbf{w}$, while fixing the other variable. During the updates, the subproblem with respect to s is optimally solved, while the subproblem with respect to $\mathbf{w}$ is not solved to optimality at each iteration – we only update $\mathbf{w}$ using a single iteration of gradient projection, for efficiency. Fortunately, this type of two-block inexact alternating optimization can be shown to converge to a KKT point. Specifically, we show that

Proposition 1: (Global Convergence) Assume that $\alpha^{(r)} < 1/\lambda_{\max}(|s^{(r)}|^2\mathbf{A}^H\mathbf{A})$ for all r. Then, the whole solution sequence produced by Algorithm 2 converges to a set $\mathcal{K}$ which consists of all the KKT points of Problem (12).

The proof of Proposition 1 is relegated to Appendix C. The insight of the proof is to treat the optimization procedure as alternating upper bound minimization, and use the continuity of the cost functions to establish asymptotic convergence.

C. Additional degrees of freedom in transmit beamforming

At this point it is worth highlighting an important difference between transmit and receive beamforming. In the receive beamforming scenario, we may be required to control the phase response as a function of $\boldsymbol{\theta}$, e.g., for phase coherence or constructive combining of specular multipath components. In transmit (including multicast) beamforming, however, it is sufficient to specify a desired magnitude for each direction or general channel vector of interest, as the receiver will have to perform any necessary phase estimation/correction anyway, due to local oscillator phase mismatch. We can mathematically model this situation by considering the modified optimization problem

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{C}^N,\boldsymbol{\theta}\in\mathbb{R}^M,s\in\mathbb{C}} \quad & \|\mathbf{y}\circledast e^{j\boldsymbol{\theta}} - s\mathbf{A}\mathbf{w}\|_2^2 \\ \text{subject to} \quad & |w_i| = 1, i = 1,\ldots,N, \end{aligned} \tag{14}$$

where $\boldsymbol{\theta}$ represents the additional degrees of (phase response) freedom. Let $\mathbf{u} = e^{j\boldsymbol{\theta}}$; we can equivalently express (14) as

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{C}^N,\mathbf{u}\in\mathbb{C}^M,s\in\mathbb{C}} \quad & \|\mathbf{Y}\mathbf{u} - s\mathbf{A}\mathbf{w}\|_2^2 \\ \text{subject to} \quad & |w_i|^2 = 1, i = 1,\ldots,N, \\ & |u_i|^2 = 1, i = 1,\ldots,M, \end{aligned} \tag{15}$$

where $\mathbf{Y} = \text{Diag}(\mathbf{y})$. Performing alternating optimization on $\mathbf{w}$, $\mathbf{u}$ and s (projecting $\mathbf{w}$ and $\mathbf{u}$ onto the unit-modulus space after each iteration), we obtain the following algorithm:

Algorithm 3

```
1:  Initialization: Set $k = 0$, obtain initial $\mathbf{w}^{(0)}$ from Algorithm 2, initialize $\mathbf{u}^{(0)} = \boldsymbol{\xi}^{(0)} = \mathbf{1}$, $\beta \in (0,1)$, $\alpha_2 = \beta/\lambda_{\max}(\mathbf{Y}^H\mathbf{Y})$
2:  Let $\mathcal{J} = \{i : y_i \neq 0\}$, $\mathbf{Y} = \text{Diag}(\mathbf{y})$, $\tilde{\mathbf{Y}} = \text{Diag}(\mathbf{y}(\mathcal{J}))$
3:  Repeat
4:      $s^{(r+1)} = (\mathbf{w}^{(r)})^H\mathbf{A}^H\mathbf{Y}\mathbf{u}^{(r)}/\|\mathbf{A}\mathbf{w}^{(r)}\|_2^2$
5:      $\alpha_1^{(r+1)} = \beta/\lambda_{\max}(|s^{(r+1)}|^2\mathbf{A}^H\mathbf{A})$
6:      $\boldsymbol{\zeta}^{(r+1)} = \mathbf{w}^{(r)} + \alpha_1^{(r+1)}(s^{(r+1)})^*\mathbf{A}^H(\mathbf{Y}\mathbf{u}^{(r)} - s^{(r+1)}\mathbf{A}\mathbf{w}^{(r)})$;
7:      $\mathbf{w}^{(r+1)} = e^{j\angle(\boldsymbol{\zeta}^{(r+1)})}$;
8:      $\boldsymbol{\xi}^{(r+1)} = \boldsymbol{\xi}^{(r)}$;
9:      $\boldsymbol{\xi}^{(r+1)}(\mathcal{J}) = \mathbf{u}^{(r)}(\mathcal{J}) - \alpha_2\tilde{\mathbf{Y}}^H(\tilde{\mathbf{Y}}\mathbf{u}^{(r)}(\mathcal{J}) - s^{(r+1)}\mathbf{A}(\mathcal{J},:)\mathbf{w}^{(r+1)})$;
10:     $\mathbf{u}^{(r+1)} = e^{j\angle(\boldsymbol{\xi}^{(r+1)})}$;
11:     $k = k + 1$
12: until convergence
```

Note that for the $\mathbf{u}$ update, only the elements corresponding to the non-zero values of $\mathbf{y}$ are of interest. As such, the set $\mathcal{J}$ has been defined as the set of indices where $y_i \neq 0$ and $\mathbf{A}(\mathcal{J},:)$ denotes a matrix comprised of the rows of $\mathbf{A}$ corresponding to the indices in $\mathcal{J}$. Thus, only the elements of $\mathbf{u}(\mathcal{J})$ are updated (steps 6 and 7 of the algorithm). Algorithm 3 can also be interpreted as an alternating upper bound minimization algorithm, and thus decreases its cost value monotonically.

Remark 1: One may also consider the conceptually simpler formulation

$$\begin{aligned} \min_{\mathbf{z}\in\mathbb{C}^{M+N}} \quad & \|\mathbf{B}\mathbf{z}\|_2^2 \\ \text{subject to} \quad & |z_i|^2 = 1, i = 1,\ldots,M+N, \end{aligned} \tag{16}$$

where

$$\mathbf{B} := \left[\begin{array}{c:c} \mathbf{Y} & -s\mathbf{A} \end{array}\right], \quad \mathbf{z} := \left[\begin{array}{c} \mathbf{u} \\ \hdashline \mathbf{w} \end{array}\right]. \tag{17}$$

Algorithm 2 can be directly applied to this formulation. However, the step size for GP in (16) is a function of the largest eigenvalue of $\mathbf{B}^H\mathbf{B}$; depending on the scaling of $\mathbf{Y}$ and $\mathbf{A}$, the freedom that Algorithm 3 provides (with respect to employing different step sizes for the $\mathbf{w}$ and $\mathbf{u}$ updates) yields faster convergence, as demonstrated in the supplementary material that accompanies this paper.

Remark 2: Typically, first-order methods feature low per-iteration complexity, but the number of iterations required to converge may be high. For convex optimization, a common trick for reducing the number of iterations it to employ Nesterov's acceleration scheme, known as 'extrapolation' [28]. Extrapolation can also be used in Algorithms 1-3. For example, applying extrapolation to (2), the resulting algorithm is defined by the following updates (with $\mathbf{w}^{(1)} = \mathbf{z}^{(1)}$):

$$\begin{aligned} \mathbf{z}^{(r+1)} &= \mathbf{w}^{(r)} - \alpha\nabla f(\mathbf{w}^{(r)}) \\ &= \mathbf{w}^{(r)} \; - \alpha\mathbf{A}^H(\mathbf{y} - \mathbf{A}\mathbf{w}^{(r)}) \\ \mathbf{w}^{(r+1)} &= \gamma^{(r)}\mathbf{z}^{(r+1)} + (1-\gamma^{(r)})\mathbf{z}^{(r)} \end{aligned} \tag{18}$$

with

$$\Gamma^{(0)} = 0, \quad \Gamma^{(r)} = \frac{1+\sqrt{1+4(\Gamma^{(r-1)})^2}}{2}, \quad \gamma^{(r)} = \frac{\Gamma^{(r-1)}}{\Gamma^{(r+1)}}. \tag{19}$$

The update for $\mathbf{z}^{(r+1)}$ is effectively identical to a simple gradient step; the $\mathbf{w}^{(r+1)}$ update then leverages the momentum from the previous iteration to descend further in the direction of $\mathbf{z}^{(r)}$. The derivation of the sequences defined in (19) can be found in [29]. Empirically, using extrapolation provides measurable gains in terms of runtime complexity, as we will illustrate in Section IV.

IV. Simulations

We first test the performance of Algorithm 1 by considering the generative signal model $\mathbf{y} = \mathbf{A}\mathbf{w} + \mathbf{n}$, where $\mathbf{n}$ is circularly symmetric zero-mean unit-variance i.i.d. Gaussian noise. The elements of $\mathbf{A} \in \mathbb{C}^{M\times N}$ are drawn from $\mathcal{N}_{\mathbb{C}}(0,\mathbf{I})$; each $\mathbf{w}$ is also drawn from $\mathcal{N}_{\mathbb{C}}(0,\mathbf{I})$, but is then projected onto the unit-modulus torus. To find our estimate $\hat{\mathbf{w}}$, we can consider the ULS formulation as in (2). The noise variance for the desired SNR was obtained via the relationship

$$\text{SNR} = 10\log_{10}\frac{\mathbb{E}\left[\text{trace}(\mathbf{A}^H\mathbf{A}\mathbf{w}\mathbf{w}^H)\right]}{M\sigma^2}. \tag{20}$$

The algorithms under test are compared both in terms of mean squared error and runtime performance; the results are all averaged over 100 independent Monte Carlo trials. In the simulations, we fix the step size $\alpha = 0.999 \times (1/L)$, where $L = \lambda_{\max}(\mathbf{A}^H\mathbf{A})$.

In this section, we mainly use the the FastSDR algorithm (FastSDR (w/Rand)) [24] as a baseline. FastSDR solves the relaxed SDP using a very efficient block coordinate descent algorithm and is state-of-the-art. In the simulations, we use 1000 randomization trials after solving the relaxed SDP. The result of MERIT that was reported to exhibit good performance in solving UQP [7] is also presented when applicable. Note that under the described generative model, the result given by (2) is in fact a maximum likelihood estimator of $\mathbf{w}$ (MLE) under the model $\mathbf{y} = \mathbf{A}\mathbf{w} + \mathbf{n}$. Therefore, in the simulations, the Cramér-Rao bound (CRB) of this generative model is also used as a baseline. The CRB with respect to the angles in $\mathbf{w}$ can be shown to be (see Appendix D for derivation)

$$\text{CRB} = \frac{\sigma^2}{2} \left[\text{Re} \left\{ \text{diag}(\mathbf{w})^H \mathbf{A}^H \mathbf{A} \text{diag}(\mathbf{w}) \right\} \right]^{-1}. \qquad (21)$$

Remark 3: Note that using the CRB to evaluate the performance of algorithms for non-convex and NP-hard *estimation* problems is common, but bringing in the CRB to assess algorithmic performance for pure optimization (or, *design*) problems unrelated to estimation seems like an unexplored idea. For ULS/UQP, SDR provides a generally unattainable (optimistic) lower bound on the least squares cost for *each instance* of this NP-hard problem, and we can take the average of these lower bounds as a bound on the average least squares cost. This, however, tells us nothing about how far the design variables are from the optimal ones. An alternative to using the SDR lower bound as a gauge is to *think* of our design problem as arising from maximum likelihood (ML) estimation for the generative signal model $\mathbf{y} = \mathbf{A}\mathbf{w}_o + \mathbf{n}$, where $\mathbf{A}$ is given, $\mathbf{w}_o$ is known to have unit-modulus elements but is otherwise unknown, and $\mathbf{n}$ is circularly symmetric i.i.d. Gaussian. ML estimation for this model boils down to ULS, and the associated CRB provides a lower bound on the variance of unbiased estimators of $\mathbf{w}_o$. Under certain conditions ($M >> N$, appropriate signal to noise ratio), ML approaches the CRB, which makes the latter predictive of ML performance. This way, the CRB can serve as a benchmark on the average attainable distance of the design variables from their optimal settings for our design problem. Of course, this bound will only be valid if we generate design problem instances from the given generative signal model, i.e., if we indeed draw desired response patterns from $\mathbf{y} = \mathbf{A}\mathbf{w}_o + \mathbf{n}$. Note that any $\mathbf{y}$ can be written in this way if we allow for low-enough signal to noise ratio in the generative model, but then the CRB will not be as predictive of the average attainable performance – albeit still a lower bound. This gives us an alternative way to explore algorithm performance for our NP-hard design problem.

Figures 3 and 4 compare the above algorithms for the described general ULS/UQP for $N = 2, \ldots, 200$. Here, we set $M = 144$ and SNR= 10dB, and observe the mean-squared-error (MSE) of the angles of the estimated $\hat{\mathbf{w}}$ as our performance measure. We see that both GP and its accelerated counterpart (GP accel; cf Remark 2) exhibit better MSE performance relative to FastSDR and MERIT in this simulation, which is rather encouraging. Furthermore, GP is approximately 10 times faster than FastSDR for all N considered and is more than 1000 times faster compared to MERIT when the number of antennas $N \geq 25$. One thing that we noticed is that MERIT does exhibit better MSE performance relative to FastSDR when N increases, but it is not as scalable as FastSDR and GP for larger N's – we were only able to test MERIT for $N \leq 75$ since it is too slow for larger N's. The accelerated implementation of GP provides a twofold improvement over GP when N is large, as shown in Fig. 4. GP and the extrapolated version of GP also approach the CRB quite closely, which means that the estimation result is nearly optimal (although the estimator is biased). Another remark is that FastSDR is known to be a surprisingly fast algorithm for approximating UQP, which has demonstrated around 1000 times speed improvement from using interior point methods in some applications [2], and our results show that GP is even more promising in dealing with ULS/UQP.

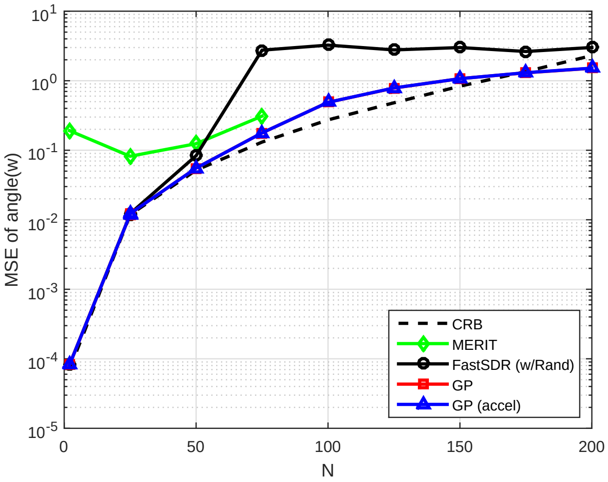

Fig. 3: The MSEs of Algorithm 1 and FastSDR under various N's; SNR= 10dB.

Figures 5-6 show the results under various M's when N is fixed to 144. One can see that GP and its accelerated version always approach the CRB, while FastSDR has a large gap between its estimation MSEs and the CRB when $M \leq 1,000$. In terms of runtime, the proposed algorithms are faster than FastSDR for all M, especially when $M \leq 1,000$.

Starting from Figure 7, we test the algorithms that are tailored for beamforming, i.e., Algorithm 2 and Algorithm 3. We test the algorithms versus the number of antennas ranging from $N = 2$ to $N = 200$ with the angle space discretized into 36 and 144 regions (resulting in $\mathbf{A}$ of dimension $36 \times N$ and $144 \times N$, respectively). The ULA scenario is considered, with rows of $\mathbf{A}$ admitting the Vandermonde structure $[1, e^{j\theta_i}, e^{j2\theta_i}, \ldots, e^{j(N-1)\theta_i}]$ for each row i in $\mathbf{A}$. 100 independent Monte Carlo trials were performed for each N, and a $K = 2$ receiver multicast beamforming scenario is considered. We impose per-antenna power constraints on all the antennas, resulting in unit-modulus constraints on w_i. The two user angles are randomly drawn in each problem instance, with each $\mathbf{y}$ constructed in accordance with (10).

Figures. 7-8 present a simulation of phase-only beamform-

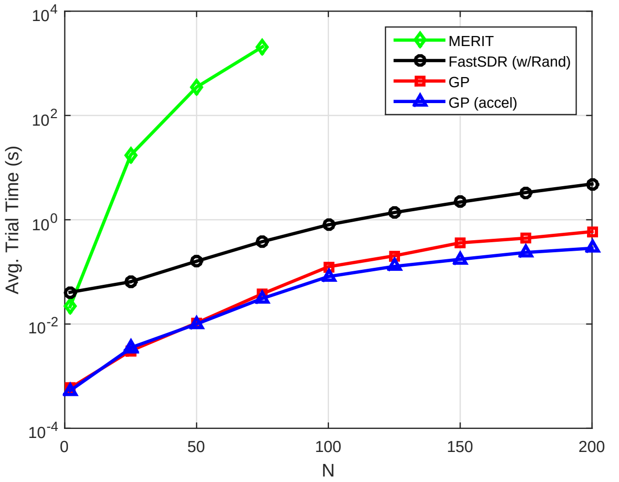

Fig. 4: Run times of Algorithm 1 and FastSDR under various N's; SNR= 10dB.

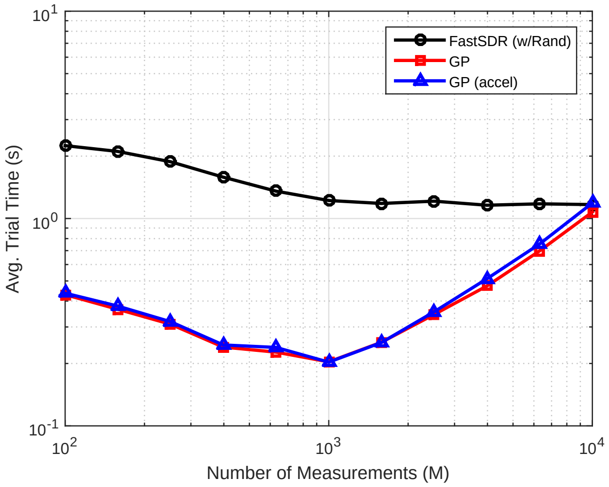

Fig. 6: Runtime comparison for simulation in Fig. 5, $N = 144$, $M = 10^2, \dots, 10^4$, SNR = 10 dB.

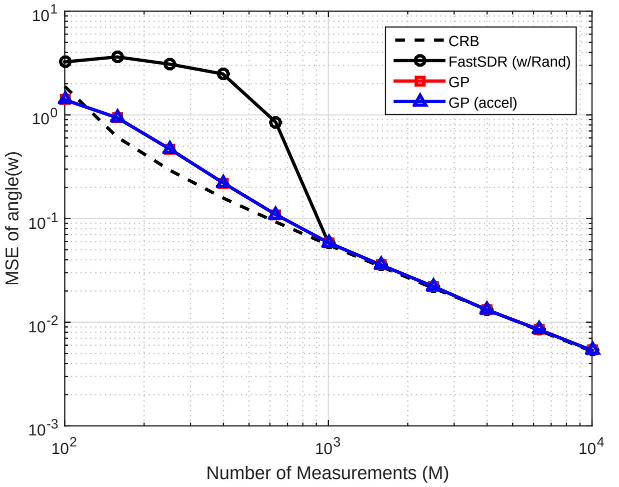

Fig. 5: CRB/MSE comparison for $M = 10^2, \dots, 10^4$, SNR = 10 dB.

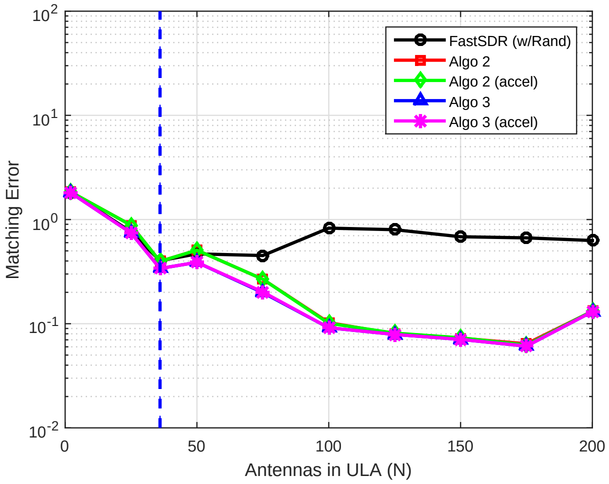

Fig. 7: Matching error for Vandermonde $\mathbf{A} \in \mathbb{C}^{36\times N}$, $N = 2, \dots, 200$.

ing for a ULA, where $\mathbf{A}$ is Vandermonde ($\mathbf{A} \in \mathbb{C}^{36\times N}$ and $\mathbf{A} \in \mathbb{C}^{144\times N}$, respectively). We compare Algorithms 2-3 with FastSDR in this simulation. We plot the values of cost functions associated with FastSDR, Algorithm 2, and Algorithm 3, respectively. These cost values reflect the matching errors between the designed and desired beam patterns, and thus are meaningful in the context of beamformer design. As shown in Fig. 7, Algorithm 2 performs comparably with FastSDR for $N \leq M$, and increasingly outperforms FastSDR for $N > M$ in terms of the matching errors. Note that it is challenging to incorporate an automatic scaling factor in SDR, and this may explain the performance of FastSDR – it also serves as evidence that adding s to the formulation is much helpful in this special ULS problem. Algorithm 3 yields even slightly lower costs compared to that of Algorithm 2 since it explores additional degrees of freedom.

The companion plots illustrating runtime complexity are shown in Figs. 9 and 10. The proposed algorithms clearly outperform FastSDR. The performance gap widens as the number of antennas increases. Further, the accelerated version of the proposed algorithms (i.e., with extrapolation) perform similarly to their unaccelerated counterparts in terms of cost, and we see modest gains in terms of runtime performance (1.1-1.5 times faster).

As expected, Algorithm 3 performs worse than Algorithm 2 in terms of runtime (particularly as the number of antennas increases) since it has one more block to update. However, even though we are only introducing $K = 2$ additional degrees of freedom (since the additional freedom to conveniently set the phase response can only be exercised at angles where the target magnitude response is nonzero), we observe a noticeable difference in matching error performance between the two proposed algorithms. To test a case with more degrees of freedom, we next consider a *sector beamforming* scenario with many more non-zero entries in $\mathbf{y}$.

Consider the scenario where rather than attempting to

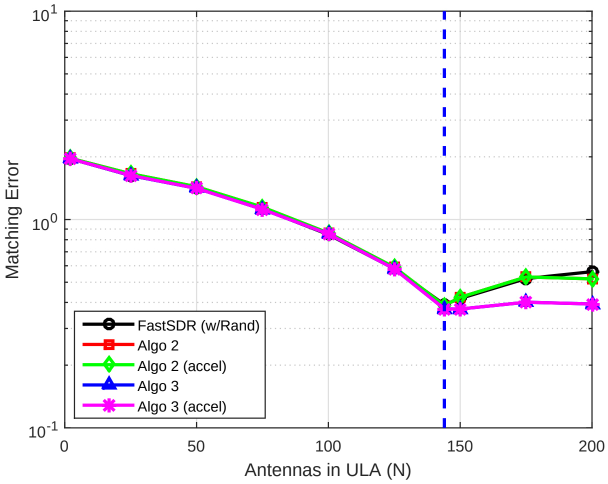

Fig. 8: Matching error for Vandermonde $\mathbf{A} \in \mathbb{C}^{144 \times N}$, $N = 2, \dots, 200$.

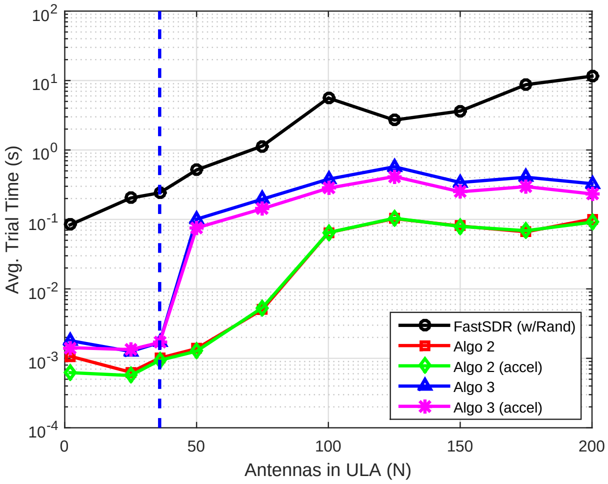

Fig. 9: Trial runtime comparison for Vandermonde $\mathbf{A} \in \mathbb{C}^{36 \times N}$, $N = 2, \dots, 200$

design a beamformer that confines the energy to selected discrete directions (i.e., pencil beamformer), a symmetric wedge spanning $[-40^o, 40^o]$ and its reflection is chosen as the desired magnitude response. For $M = 144$, this is equivalent to $\mathcal{J} = \{1 \dots 18, 55 \dots 90, 127 \dots 144\}$, with $y_i = 1$ for all $i \in \mathcal{J}$. Figure 11 is a polar plot of the resulting transmit beampattern. In this scenario, we again have $N + K$ degrees of freedom in (15), but now $K >> 2$; as such, it is reasonable to expect improved performance relative to (12) compared to the previously considered 'pencil beam' scenario. Indeed, we observe a substantial improvement for Algorithm 3, as shown in Figure 12. To summarize, the alternating optimization algorithm for the formulation in (15) provides performance gains (in terms of least squares cost), even with the introduction of very few additional degrees of freedom (K). Further, these improvements become increasingly pronounced as K increases (as in the sector beamforming scenario).

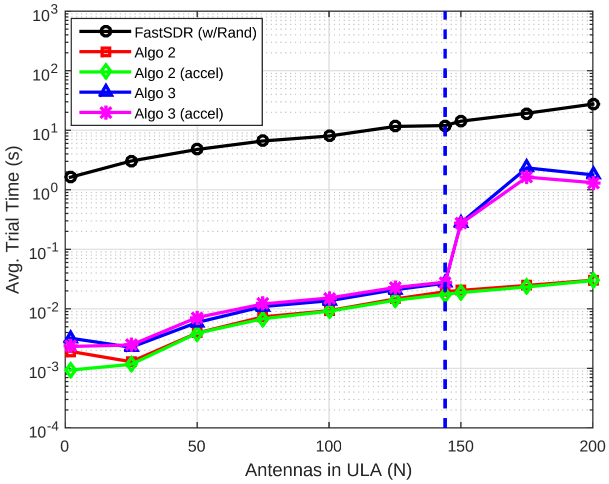

Fig. 10: Trial runtime comparison for Vandermonde $\mathbf{A} \in \mathbb{C}^{144 \times N}$, $N = 2, \dots, 200$

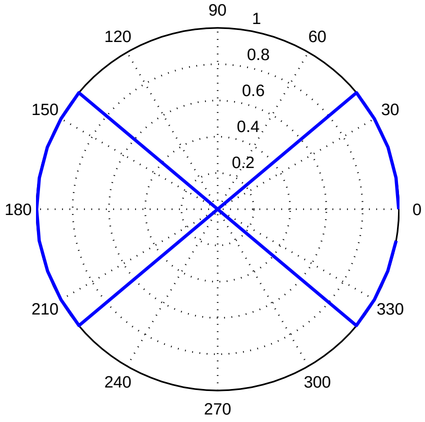

Fig. 11: Sector beamforming illustration, $M = 144$, $\mathcal{J} = \{1 \dots 18, 55 \dots 90, 127 \dots 144\}$

Although the beamforming examples discussed so far employed a ULA (Vandermonde steering vectors), it is natural to consider the case where $\mathbf{A}$ is complex Gaussian, modeling a Rayleigh fading scenario. The simulations shown in Figures 7-10 are repeated for $\mathbf{A}$ which is circularly symmetric Gaussian with unit variance. As shown in Figures 13 and 14, all the proposed algorithms outperform FastSDR in terms of matching error. Again, accelerated methods demonstrate modestly improved average trial times with respect to their plain versions (cf. Figures 15-16.)

V. Conclusion

In this paper we have considered the ULS/UQP problem and certain extensions that arise in phase-only beamforming, with emphasis on emerging massive MIMO applications. To

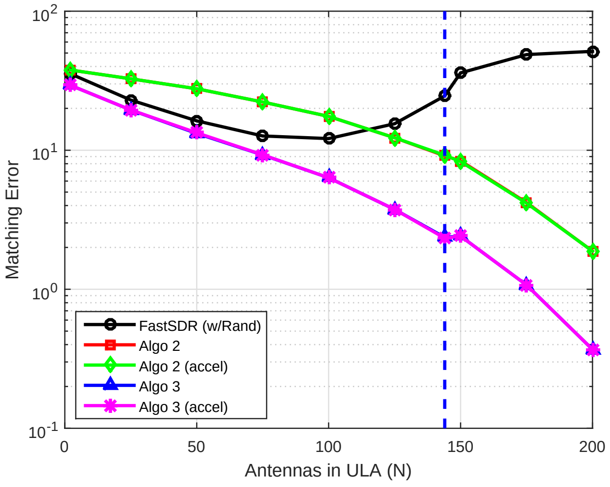

Fig. 12: Matching error comparison for sector case, $M = 144$, $N = 2, \dots, 200$

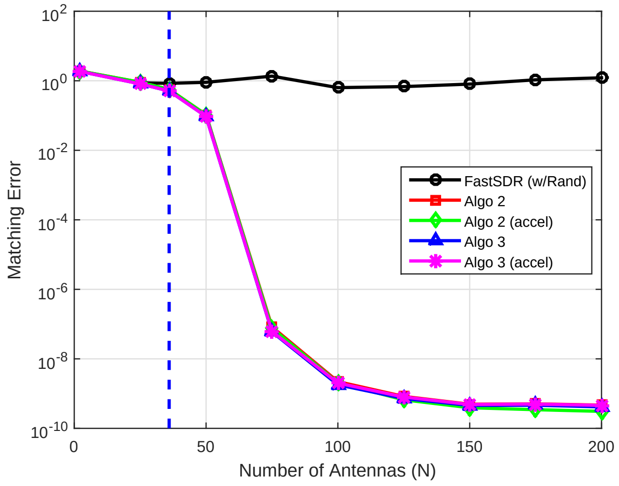

Fig. 13: Matching error comparison for Gaussian $\mathbf{A} \in \mathbb{C}^{36 \times N}$, $N = 2, \dots, 200$

circumvent the scalability issues of using SDR for handling ULS, a gradient projection-based algorithm has been proposed. Convergence properties of the algorithm have been carefully studied. To enhance the performance of large-scale beamformer design, two variants of the plain-vanilla ULS formulation have been proposed, which introduce more degrees of freedom to the design problem. Generalizations of the GP algorithm have been proposed, and their convergence properties have been discussed as well. The proposed algorithms have been carefully compared against state-of-art methods such as SDR and MERIT, and have been found to perform at least as well in terms of accuracy, and even better in several scenarios at significantly lower runtime complexity.

VI. Acknowledgement

The authors would like to thank Prof. Mojtaba Soltanalian, University of Illinois at Chicago, for kindly providing his code for the MERIT algorithm.

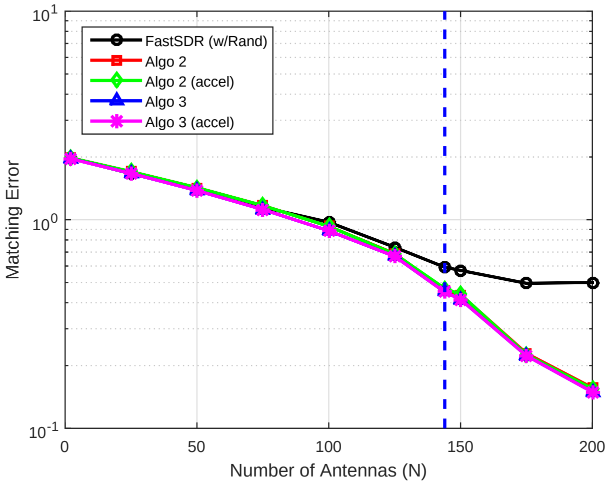

Fig. 14: Matching error comparison for Gaussian $\mathbf{A} \in \mathbb{C}^{144 \times N}$, $N = 2, \dots, 200$

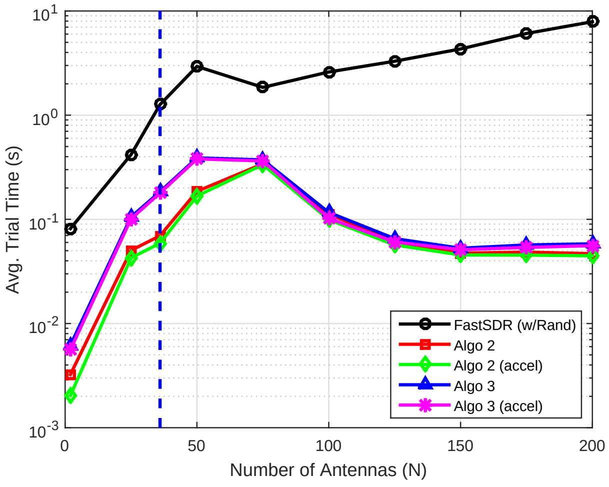

Fig. 15: Trial runtime comparison for Gaussian $\mathbf{A} \in \mathbb{C}^{36 \times N}$, $N = 2, \dots, 200$

Appendix A
Proof of Lemma 1

The gradient projection step can be written as follows:

$$\tilde{\mathbf{w}}^{(r+1)} \in \arg\min_{\tilde{\mathbf{w}} \in \mathcal{W}} \; f(\tilde{\mathbf{w}}^{(r)}) + \langle \nabla f(\tilde{\mathbf{w}}^{(r)}), \tilde{\mathbf{w}} - \tilde{\mathbf{w}}^{(r)} \rangle + \frac{1}{2\alpha} \|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}^{(r)}\|_2^2, \qquad (22)$$

where we define

$$\mathcal{W} = \{\tilde{\mathbf{w}} \mid \tilde{w}_i^2 + \tilde{w}_{N+i}^2 = 1, \; i = 1, \dots, N\}.$$

By arranging terms, one can verify that the solution of Problem (22) can be obtained via the following equivalent form:

$$\tilde{\mathbf{w}}^{(r+1)} \in \arg\min_{\tilde{\mathbf{w}} \in \mathcal{W}} \; \left\| \tilde{\mathbf{w}} - \left(\tilde{\mathbf{w}}^{(r)} - \alpha \nabla_{\tilde{\mathbf{w}}} f\left(\tilde{\mathbf{w}}^{(r)} \right) \right) \right\|_2^2,$$

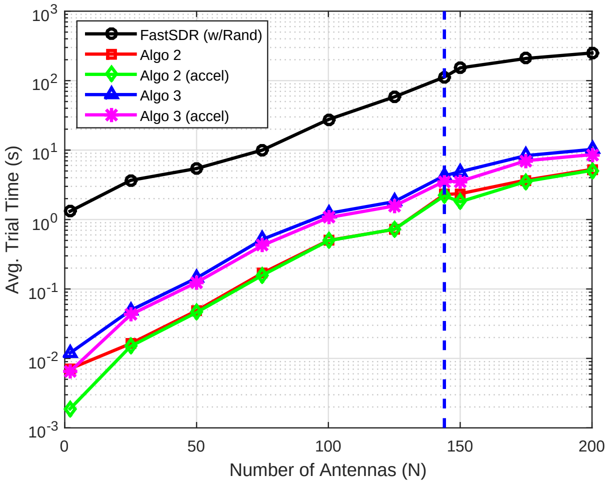

Fig. 16: Trial runtime comparison for Gaussian $\mathbf{A} \in \mathbb{C}^{144\times N}$, $N = 2, \ldots, 200$

which is exactly a GP step. The above implies that $\tilde{\mathbf{w}}^{(r+1)}$ satisfies the KKT conditions of the right hand side (RHS) of Problem (22), i.e.,

$$\mathbf{0} = \nabla f(\tilde{\mathbf{w}}^{(r)}) + \frac{1}{\alpha}(\tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}) + 2\tilde{\mathbf{w}}^{(r+1)} \circledast \boldsymbol{\lambda}^{(r+1)} \tag{23a}$$

$$\Rightarrow \|\nabla f(\tilde{\mathbf{w}}^{(r)}) + 2\tilde{\mathbf{w}}^{(r+1)} \circledast \boldsymbol{\lambda}^{(r+1)}\|_2^2 = \frac{1}{\alpha^2}\|\tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\|_2^2. \tag{23b}$$

From the above, one can see that

$$\begin{aligned}
&\|\nabla f(\tilde{\mathbf{w}}^{(r)}) + 2\tilde{\mathbf{w}}^{(r+1)} \circledast \boldsymbol{\lambda}^{(r+1)}\|_2^2 \to 0\\
\Rightarrow &\|\tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\|_2^2 \to 0\\
\Rightarrow &\|\nabla f(\tilde{\mathbf{w}}^{(r)}) + 2\tilde{\mathbf{w}}^{(r)} \circledast \boldsymbol{\lambda}^{(r+1)}\|_2^2 \to 0
\end{aligned}$$

which implies that the KKT conditions are satisfied since $\tilde{\mathbf{w}}^{(r)}$ is always feasible. Note that to show the last equation above, it is required that the elements of $|\boldsymbol{\lambda}^{(r+1)}|$ be bounded as $r \to \infty$, which can be readily seen from (23a) and the assumption that $\mathbf{A}$ and $\mathbf{y}$ are bounded.

Appendix B
Proof of Theorem 1

Let us prove the $a)$ part first. The proof follows the insight of the convergence proof of the successive upper bound minimization (SUM) algorithm in [30], with modifications to accommodate the nonconvex constraints. Let us first relate the gradient projection algorithm to SUM. Note that the cost function of Problem (8) has Lipschitz continuous gradients and $L = \lambda_{\max}(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}})$ is its smallest Lipschitz constant. Then, we have

$$\begin{aligned}
f(\tilde{\mathbf{w}}^{(r)}) \leq u(\tilde{\mathbf{w}}; \tilde{\mathbf{w}}^{(r)}) = f(\tilde{\mathbf{w}}^{(r)}) + \langle \nabla f(\tilde{\mathbf{w}}^{(r)}), \tilde{\mathbf{w}} - \tilde{\mathbf{w}}^{(r)}\rangle&\\
+ \frac{1}{2\alpha}\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}^{(r)}\|_2^2, \quad \forall \tilde{\mathbf{w}},&
\end{aligned}$$

since $\alpha < 1/L$. So $u(\tilde{\mathbf{w}}; \tilde{\mathbf{w}}^{(r)})$ is an upper bound on $f(\tilde{\mathbf{w}})$. Note that

$$f(\tilde{\mathbf{w}}^{(r)}) = u(\tilde{\mathbf{w}}^{(r)}; \tilde{\mathbf{w}}^{(r)}) \tag{24}$$

$$\nabla f(\tilde{\mathbf{w}}^{(r)}) = \nabla u(\tilde{\mathbf{w}}^{(r)}; \tilde{\mathbf{w}}^{(r)}). \tag{25}$$

As we have seen in (22), the GP algorithm can be considered as solving the following upper-bound problem:

$$\tilde{\mathbf{w}}^{(r+1)} \in \arg\min_{\tilde{\mathbf{w}} \in \mathcal{W}} \; u(\tilde{\mathbf{w}}; \tilde{\mathbf{w}}^{(r)}).$$

Such a procedure is the so-called majorization minimization (MM) algorithm or SUM, and by the properties of MM, we have

$$\begin{aligned}
f(\tilde{\mathbf{w}}^{(r)}) &= u(\tilde{\mathbf{w}}^{(r)}; \tilde{\mathbf{w}}^{(r)}) && \text{(26a)}\\
&\geq u(\tilde{\mathbf{w}}^{(r+1)}; \tilde{\mathbf{w}}^{(r)}) && \text{(26b)}\\
&\geq f(\tilde{\mathbf{w}}^{(r+1)}); && \text{(26c)}
\end{aligned}$$

i.e., the cost function decreases at each iteration. Assume that there is a subsequence $\{r_j\}_j$ that converges to a limit point, i.e., $\tilde{\mathbf{w}}^{(r_j)} \to \tilde{\mathbf{w}}^\star$. We have

$$\begin{aligned}
u(\tilde{\mathbf{w}}; \tilde{\mathbf{w}}^{(r_j)}) &\geq u(\tilde{\mathbf{w}}^{(r_j+1)}; \tilde{\mathbf{w}}^{(r_j)}) && \text{(27a)}\\
&\geq f(\tilde{\mathbf{w}}^{(r_j+1)}) && \text{(27b)}\\
&\geq f(\tilde{\mathbf{w}}^{(r_{j+1})}) && \text{(27c)}\\
&= u(\tilde{\mathbf{w}}^{(r_j+1)}; \tilde{\mathbf{w}}^{(r_j+1)}), && \text{(27d)}
\end{aligned}$$

where (27c) holds since $r_{j+1} \geq r_j + 1$ (as r_j indexes a subsequence). Taking $j \to \infty$, and by the continuity of $u(\cdot)$ we see that

$$u(\tilde{\mathbf{w}}; \tilde{\mathbf{w}}^\star) \geq u(\tilde{\mathbf{w}}^\star; \tilde{\mathbf{w}}^\star), \quad \forall \tilde{\mathbf{w}} \in \mathcal{W}. \tag{28}$$

The above means that there exists a $\boldsymbol{\lambda} \in \mathbb{R}^{2N}$ where $\lambda_i = \lambda_{N+i}$ for $i = 1, \ldots, N$ such that $\tilde{\mathbf{w}}^\star$ satisfies

$$\nabla u(\tilde{\mathbf{w}}^\star; \tilde{\mathbf{w}}^\star) + 2\boldsymbol{\lambda}^\star \circledast \tilde{\mathbf{w}}^\star = \mathbf{0}, \;\; \tilde{\mathbf{w}}^\star \in \mathcal{W},$$

since $\tilde{\mathbf{w}}^\star$ is a minimizer of $u(\tilde{\mathbf{w}}; \tilde{\mathbf{w}}^\star)$ over $\mathcal{W}$. By (25), we have

$$\nabla f(\tilde{\mathbf{w}}^\star) + 2\boldsymbol{\lambda}^\star \circledast \tilde{\mathbf{w}}^\star = \mathbf{0}, \;\; \tilde{\mathbf{w}}^\star \in \mathcal{W}.$$

Therefore, every limit point of $\{\tilde{\mathbf{w}}^{(r)}\}_r$ is a KKT point of Problem (8). In addition, since $\tilde{\mathbf{w}}^{(r)}$ lives in a compact set, we further claim that the whole solution sequence (instead of every convergent subsequence) converges to $\mathcal{K}$ which consists of all KKT points of (8). Indeed, suppose that $\{\tilde{\mathbf{w}}^{(r)}\}$ does not converge to $\mathcal{K}$. Since $\tilde{\mathbf{w}}$ lives in a compact set, there exists a subsequence indexed by r_j converging to a point $\mathbf{z}$ such that $d(\mathbf{z}, \mathcal{K}) \geq \gamma$ where $\gamma > 0$. However, we have just shown that every limit point is a KKT point, which is a contradiction. Therefore, we have $d(\tilde{\mathbf{w}}^{(r)}, \mathcal{K}) \to 0$.

Now we show that the $b)$ part holds. Due to the Lipschitz continuity of $\nabla f(\tilde{\mathbf{w}})$, we have

$$\begin{aligned}
f(\tilde{\mathbf{w}}^{(r+1)}) \leq f(\tilde{\mathbf{w}}^{(r)}) + \langle \nabla f(\tilde{\mathbf{w}}^{(r)}), \tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\rangle&\\
+ \frac{L}{2}\|\tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\|_2^2.&
\end{aligned} \tag{29}$$

We also have

$$\begin{aligned}&\langle \nabla f(\tilde{\mathbf{w}}^{(r)}), \tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\rangle + \frac{1}{2\alpha}\|\tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\|_2^2 \\ \leq &\langle \nabla f(\tilde{\mathbf{w}}^{(r)}), \tilde{\mathbf{w}}^{(r)} - \tilde{\mathbf{w}}^{(r)}\rangle + \frac{1}{2\alpha}\|\tilde{\mathbf{w}}^{(r)} - \tilde{\mathbf{w}}^{(r)}\|_2^2, \ \forall \tilde{\mathbf{w}} \in \mathcal{W},\end{aligned}$$

since $\tilde{\mathbf{w}}^{(r+1)}$ is a minimizer of $u(\tilde{\mathbf{w}}; \tilde{\mathbf{w}}^{(r)})$ over $\mathcal{W}$ (also see (22)). The above implies that

$$\begin{aligned}&\langle \nabla f(\tilde{\mathbf{w}}^{(r)}), \tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\rangle + \frac{1}{2\alpha}\|\tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\|_2^2 \leq \mathbf{0} \\ \Rightarrow &\langle \nabla f(\tilde{\mathbf{w}}^{(r)}), \tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\rangle \leq -\frac{1}{2\alpha}\|\tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\|_2^2.\end{aligned} \tag{30}$$

Plugging (30) into (29), we have

$$f(\tilde{\mathbf{w}}^{(r+1)}) - f(\tilde{\mathbf{w}}^{(r)}) \leq \left(\frac{L}{2} - \frac{1}{2\alpha}\right)\|\tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\|_2^2. \tag{31}$$

Summing up over $r = 0$ to $r = T - 1$, we have

$$f(\tilde{\mathbf{w}}^{(T)}) - f(\tilde{\mathbf{w}}^{(0)}) \leq \sum_{r=0}^{T-1}\left(\frac{L}{2} - \frac{1}{2\alpha}\right)\|\tilde{\mathbf{w}}^{(r+1)} - \tilde{\mathbf{w}}^{(r)}\|_2^2$$

Following (23b), the above means that

$$\begin{aligned}-f(\tilde{\mathbf{w}}^{(T)}) + f(\tilde{\mathbf{w}}^{(0)}) \geq \sum_{r=0}^{T-1}\left(-\frac{L}{2} + \frac{1}{2\alpha}\right)\alpha^2\|\nabla f(\tilde{\mathbf{w}}^{(r)}) \\ + 2\tilde{\mathbf{w}}^{(r+1)} \circledast \boldsymbol{\lambda}^{(r+1)}\|_2^2.\end{aligned}$$

By the definition of T, we have

$$\begin{aligned}&\left(\frac{1}{2\alpha} - \frac{L}{2}\right)\alpha^2\epsilon \\ \leq &\frac{\sum_{r=0}^{T-1}\left(\frac{1}{2\alpha} - \frac{L}{2}\right)\alpha^2\|\nabla f(\tilde{\mathbf{w}}^{(r)}) + 2\tilde{\mathbf{w}}^{(r+1)} \circledast \boldsymbol{\lambda}^{(r+1)}\|_2^2}{T-1} \\ \leq &\frac{f(\tilde{\mathbf{w}}^{(0)}) - f(\tilde{\mathbf{w}}^{(T)})}{T-1} \\ \Rightarrow &\epsilon \leq \frac{v}{T-1},\end{aligned}$$

where

$$v = \frac{f(\tilde{\mathbf{w}}^{(0)}) - f^\star}{\alpha^2\left(\frac{1}{2\alpha} - \frac{L}{2}\right)},$$

where $f^\star$ denotes the *global optimal* value of Problem (8)

Appendix C
Proof of Proposition 1

Let us denote

$$f(\mathbf{w}, s) = \|\mathbf{y} - s\mathbf{A}\mathbf{w}\|_2^2.$$

We can then define the surrogate upper bound

$$\begin{aligned}g(\mathbf{w}; \mathbf{w}^{(r)}, s^{(r)}) = f(\mathbf{w}^{(r)}, s^{(r)}) + \langle \nabla f(\mathbf{w}^{(r)}, s^{(r)}), \mathbf{w} - \mathbf{w}^{(r)}\rangle \\ + \frac{1}{2\alpha^{(r)}}\|\mathbf{w} - \mathbf{w}^{(r)}\|_2^2.\end{aligned}$$

It follows that

$$f(\mathbf{w}, s^{(r)}) \leq g(\mathbf{w}; \mathbf{w}^{(r)}, s^{(r)}), \ \forall \mathbf{w} \tag{32}$$

$$f(\mathbf{w}^{(r)}, s^{(r)}) = g(\mathbf{w}^{(r)}; \mathbf{w}^{(r)}, s^{(r)}) \tag{33}$$

$$\nabla_{\mathbf{w}} f(\mathbf{w}^{(r)}, s^{(r)}) = \nabla_{\mathbf{w}} g(\mathbf{w}^{(r)}; \mathbf{w}^{(r)}, s^{(r)}), \tag{34}$$

where the first inequality is due to the fact that $\alpha^{(r)} < 1/\lambda_{\max}(|s^{(r)}|\mathbf{A}^H\mathbf{A})$. Our updates can therefore be expressed as

$$\mathbf{w}^{r+1} = \arg\min_{|w_i|=1} g(\mathbf{w}; \mathbf{w}^{(r)}, s^{(r)}) \tag{35a}$$

$$s^{(r+1)} = \arg\min_{s} f(\mathbf{w}^{(r+1)}, s) = \frac{(\mathbf{w}^{(r+1)})^H\mathbf{A}^H\mathbf{y}}{\|\mathbf{A}\mathbf{w}^{(r+1)}\|^2}. \tag{35b}$$

The objective function decreases monotonically because the following holds:

$$\begin{aligned}f(\mathbf{w}^{(r)}, s^{(r)}) &= g(\mathbf{w}^{(r)}; \mathbf{w}^{(r)}, s^{(r)}) &(36a)\\ &\geq g(\mathbf{w}^{(r)}; \mathbf{w}^{(r+1)}, s^{(r)}) &(36b)\\ &\geq f(\mathbf{w}^{(r+1)}, s^{(r)}) &(36c)\\ &\geq f(\mathbf{w}^{(r+1)}, s^{(r+1)}), &(36d)\end{aligned}$$

where (36a) follows (33). (36b) is obtained because of (35a), (36c) holds due to the property in (32), and (36d) is obtained by the fact that the subproblem w.r.t. s is optimally solved via (35b).

Assume that $\{r_j\}$ signifies the index set of a convergent subsequence, and that $\{\mathbf{w}^{(r_j)}, s^{(r_j)}\}$ converges to $(\mathbf{w}^\star, s^\star)$. Then, we have

$$\begin{aligned}g(\mathbf{w}; \mathbf{w}^{(r_j)}, s^{(r_j)}) &\geq g(\mathbf{w}^{(r_j+1)}; \mathbf{w}^{(r_j)}, s^{(r_j)}) &(37a)\\ &\geq f(\mathbf{w}^{(r_j+1)}, s^{(r_j)}) &(37b)\\ &\geq f(\mathbf{w}^{(r_j+1)}, s^{(r_j+1)}) &(37c)\\ &\geq f(\mathbf{w}^{(r_{j+1})}, s^{(r_{j+1})}) &(37d)\\ &= g(\mathbf{w}^{(r_{j+1})}; \mathbf{w}^{(r_{j+1})}, s^{(r_{j+1})}), &(37e)\end{aligned}$$

where (37b) holds because of (32) (i.e., $g(\mathbf{w}; \mathbf{w}^{(r)}, s^{(r)})$ upper bounds $f(\mathbf{w}, \mathbf{s}^{(r)})$ for all $\mathbf{w}$), (37d) is obtained by the fact that $r_{j+1} \geq r_j + 1$ since r_j indexes a subsequence. Taking $j \to \infty$ and by the continuity of $g(\cdot)$, we see that

$$g(\mathbf{w}; \mathbf{w}^\star, s^\star) \geq g(\mathbf{w}^\star; \mathbf{w}^\star, s^\star). \tag{38}$$

The inequality in (38) means that $\mathbf{w}^\star$ is a blockwise minimizer of $g(\mathbf{w}; \mathbf{w}^\star, s^\star)$. Therefore, it satisfies the partial KKT condition w.r.t. $\mathbf{w}$, i.e.,

$$\nabla_{\mathbf{w}} g(\mathbf{w}^\star; \mathbf{w}^\star, s^\star) + 2\boldsymbol{\lambda}^\star \circledast \mathbf{w}^\star = \mathbf{0}. \tag{39}$$

By (34), we have

$$\nabla_{\mathbf{w}} f(\mathbf{w}^\star, s) + 2\boldsymbol{\lambda}^\star \circledast \mathbf{w}^\star = \mathbf{0}. \tag{40}$$

Similarly, by the update rule in (35b), we have

$$f(\mathbf{w}^{(r_j)}, s) \geq f(\mathbf{w}^{(r_j)}, s^{(r_j)}),$$

and thus

$$f(\mathbf{w}^\star, s) \geq f(\mathbf{w}^\star, s^\star).$$

Then, the argument for $s^\star$ satisfying the KKT conditions follows. Therefore, every limit point of the solution sequence is a KKT point. We also notice that both s and $\mathbf{w}$ live in compact sets. Therefore, repeating the arguments as in Theorem 1, one can show that the whole solution sequence converges to $\mathcal{K}$, which completes the proof.

Appendix D
Derivation of Cramér-Rao Bound (21)

Consider the generative signal model $\mathbf{y} = \mathbf{A}\mathbf{w}+\mathbf{v}$, where $\mathbf{w}$ is constrained to the unit-modulus torus and $\mathbf{v} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. We can explicitly parameterize in terms of the vector of angles $\boldsymbol{\theta}$ as $\mathbf{y} = \mathbf{A}e^{j\boldsymbol{\theta}} + \mathbf{v}$, hence $\mathbf{y} \sim \mathcal{N}(\mathbf{A}e^{j\boldsymbol{\theta}}, \sigma^2\mathbf{I})$. The Fisher Information Matrix for this model can be expressed as [31]

$$[\mathbf{F}]_{k,l} = \frac{2}{\sigma^2}\Re\left\{\left(\frac{\partial \mathbf{A}e^{j\boldsymbol{\theta}}}{\partial\theta_k}\right)^H\left(\frac{\partial \mathbf{A}e^{j\boldsymbol{\theta}}}{\partial\theta_l}\right)\right\}, \qquad (41)$$

where parameter vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_N]^T$. We then obtain the derivative with respect to each θ_i as $\frac{\partial \mathbf{A}e^{j\boldsymbol{\theta}}}{\partial\theta_i} = je^{j\theta_i}\mathbf{a}_i$, where $\mathbf{a}_i$ denotes the i-th column of $\mathbf{A}$. It follows that

$$[\mathbf{F}]_{k,l} = \frac{2}{\sigma^2}\mathrm{Re}\left\{e^{-j\theta_k}\mathbf{a}_k^H\mathbf{a}_l e^{j\theta_l}\right\}, \qquad (42)$$

from which we can construct the Fisher Information Matrix as

$$\mathbf{F} = \frac{2}{\sigma^2}\mathrm{Re}\left\{\mathrm{Diag}\left(e^{-j\boldsymbol{\theta}}\right)\mathbf{A}^H\mathbf{A}\mathrm{Diag}\left(e^{j\boldsymbol{\theta}}\right)\right\}. \qquad (43)$$

Thus, the CRB on $\boldsymbol{\theta}$ can be compactly written as $(\mathbf{w} = e^{j\boldsymbol{\theta}})$

$$\mathrm{CRB} = \mathbf{F}^\dagger = \frac{\sigma^2}{2}\left[\mathrm{Re}\left\{\mathrm{Diag}(\mathbf{w})^H\mathbf{A}^H\mathbf{A}\mathrm{Diag}(\mathbf{w})\right\}\right]^\dagger. \qquad (44)$$

References

[1] J. Tranter, N. Sidiropoulos, X. Fu, and A. Swami, "Fast unit-modulus least squares with applications in transmit beamforming," in *Proc. 24th EUSIPCO Conference,* Aug. 29 – Sep. 2, Budapest, Hungary, 2016.

[2] X. Fu, F. Chan, W.-K. Ma, and H.-C. So, "A complex-valued semidefinite relaxation approach for two-dimensional source localization using distance measurements and imperfect receiver positions," *Signal Processing (ICSP) Proceedings, IEEE*, vol. 2, pp. 1491–1494, 2012.

[3] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.

[4] I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, MaxCut and complex semidefinite programming," *Mathematical Programming A*, vol. 149, pp. 47–81, February 2015.

[5] P. Thompson, "Adaptation by direct phase-shift adjustment in narrowband adaptive antenna systems," *Trans. Antennas Propagat., IEEE*, vol. 24, no. 5, pp. 756–760, 1976.

[6] C.-J. Lu, W.-X. Sheng, Y.-B. Han, and X.-F. Ma, "A novel adaptive phase-only beamforming algorithm based on semidefinite relaxation," *Phased Array Systems Technology, 2013 IEEE International Symposium on*, pp. 617–621, October 2013.

[7] M. Soltanalian and P. Stoica, "Designing unimodular codes via quadratic optimization," *Trans. on Signal Processing, IEEE*, vol. 62, no. 5, pp. 1221–1234, March 2014.

[8] J. Pan and W. K. Ma, "Constant envelope precoding for single-user large-scale miso channels: Efficient precoding and optimal designs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 982–995, Oct 2014.

[9] S. K. Mohammed and E. G. Larsson, "Single-user beamforming in large-scale miso systems with per-antenna constant-envelope constraints: The doughnut channel," *IEEE Transactions on Wireless Communications*, vol. 11, no. 11, pp. 3992–4005, 2012.

[10] ——, "Per-antenna constant envelope precoding for large multi-user mimo systems," *IEEE Transactions on Communications*, vol. 61, no. 3, pp. 1059–1071, 2013.

[11] ——, "Constant-envelope multi-user precoding for frequency-selective massive mimo systems," *IEEE Wireless Communications Letters*, vol. 2, no. 5, pp. 547–550, 2013.

[12] S. Zhang and Y. Huang, "Complex quadratic optimization and semidefinite programming," *SIAM Journal on Optimization*, vol. 16, no. 3, pp. 871–890, 2006.

[13] Z.-Q. Luo, W.-K. Ma, A.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *Signal Processing Magazine, IEEE*, vol. 27, no. 3, pp. 20–34, May 2010.

[14] I. Telatar, "Capacity of multi-antenna Gaussian channels," *Eur. Trans. Telecommun.*, vol. 10, no. 6, pp. 585–596, Nov.-Dec. 1999.

[15] X. Zhang, A. Molisch, and S.-Y. Kung, "Variable-phase-shift-based RF-baseband codesign for MIMO antenna selection," *Trans. on Signal Processing, IEEE*, vol. 53, no. 11, pp. 4091–4103, November 2005.

[16] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, "What will 5G be?" *Journal on Selected Areas in Commun., IEEE*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[17] C. Baird and G. Rassweiler, "Adaptive sidelobe nulling using digitally controlled phase-shifters," *Trans. Antennas Propagat., IEEE*, vol. AP-24, pp. 638–649, 1976.

[18] T. Ismail and M. Dawoud, "Null steering in phased arrays by controlling the element positions," *Trans. Antennas Propagat., IEEE*, vol. 39, pp. 1561–1566, 1991.

[19] S. Smith, "Optimum phase-only adaptive nulling," *Trans. on Signal Processing, IEEE*, vol. 47, no. 7, pp. 1835–1842, July 1999.

[20] W. Choi and T. Sarkar, "Phase-only adaptive processing based on a direct data domain least squares approach using the conjugate gradient method," *Trans. Antennas Propagat., IEEE*, vol. 10, no. 6, pp. 585–596, Nov.-Dec. 2004.

[21] T. Ismail and Z. Hamici, "Array pattern synthesis using digital phase control by quantized particle swarm optimization," *Trans. on Antennas and Propagation, IEEE*, vol. 58, no. 6, pp. 2142–2145, June 2010.

[22] T. Sinhamahapatra, A. Ahmed, G. Mahanti, N. Pathak, and A. Chakrabarty, "Design of discrete phase-only dual-beam array antennas with minimum dynamic range ratio," *Applied Electromagnetics Conference, IEEE*, pp. 1–4, December 2007.

[23] N. Sidiropoulos and Z.-Q. Luo, "A semidefinite relaxation approach to MIMO detection for high-order QAM constellations," *Signal Processing Letters, IEEE*, vol. 13, no. 9, pp. 525–528, September 2006.

[24] Z. Wen, D. Goldfarb, S. Ma, and K. Scheinberg, "Roy by row methods for semidefinite programming," *Dept. of IEOR, Columbia University, Tech. Rep.*, April 2009.

[25] N. Boumal, "Nonconvex phase synchronization," *arXiv preprint arXiv:1601.06114*, 2016.

[26] R. Luss and M. Teboulle, "Conditional gradient algorithms for rank-one matrix approximations with a sparsity constraint," *SIAM Review*, vol. 55, no. 1, pp. 65–98, 2013.

[27] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, "Generalized power method for sparse principal component analysis," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 517–553, 2010.

[28] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$," *Dokl. Acad. Nauk SSSR*, pp. 543–547, 1983.

[29] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[30] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[31] S. Kay, *Fundamentals of Signal Processing: Estimation Theory*. Upper Saddle River, New Jersey: Prentice Hall, 1993.