Inexact Alternating Optimization for Phase Retrieval in the Presence of Outliers

Cheng Qian, Xiao Fu, *Member, IEEE*, Nicholas D. Sidiropoulos, *Fellow, IEEE*, Lei Huang, *Senior Member, IEEE*, and Junhao Xie, *Senior Member, IEEE*

***Abstract*—Phase retrieval has been mainly considered in the presence of Gaussian noise. However, the performance of the algorithms proposed under the Gaussian noise model severely degrades when grossly corrupted data, i.e., outliers, exist. This paper investigates techniques for phase retrieval in the presence of heavy-tailed noise, which is considered a better model for situations where outliers exist. An ℓ_p-norm ($0 < p < 2$) based estimator is proposed for fending against such noise, and two-block inexact alternating optimization is proposed as the algorithmic framework to tackle the resulting optimization problem. Two specific algorithms are devised by exploring different local approximations within this framework. Interestingly, the core conditional minimization steps can be interpreted as iteratively reweighted least squares and gradient descent. Convergence properties of the algorithms are discussed, and the Cramér–Rao bound (CRB) is derived. Simulations demonstrate that the proposed algorithms approach the CRB and outperform state-of-the-art algorithms in heavy-tailed noise.**

***Index Terms*—Phase retrieval, iterative reweighted least squares (IRLS), gradient descent, impulsive noise, Cramér-Rao bound (CRB).**

I. Introduction

PHASE *retrieval* aims at recovering a signal $\mathbf{x} \in \mathbb{C}^N$ from only the magnitude of linear measurements. This is an old problem [2], [3] that has recently attracted renewed and growing interest. Phase retrieval arises in many fields, such as X-ray crystallography, coherent diffraction imaging, and optical imaging and astronomy, where the detectors only record the intensity information, because phase is very difficult and expensive to measure.

The mathematical description of the phase retrieval problem is simple: given the measuring matrix $\mathbf{A} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_M]^H \in \mathbb{C}^{M \times N}$ and the measurement vector $\mathbf{y} \in \mathbb{R}^M$, where $\mathbf{a}_m \in \mathbb{C}^N$, and

$$\mathbf{y} = |\mathbf{A}\mathbf{x}|,$$

find $\mathbf{x} \in \mathbb{C}^N$. Early attempts to solve the phase retrieval problem can be traced back to the 1970s, where techniques such as Gerchberg-Saxton (GS) [2], Fienup [3] and their variants were proposed. These algorithms have been empirically shown to work well under certain conditions, although little had been known regarding their convergence properties from a theoretical point of view. Recently, some new algorithms along this line of work were proposed in [4], where the convergence issue is better studied.

In recent years, more modern optimization-based approaches for phase retrieval have been proposed. For example, Candès proposed a semidefinite relaxation approach known as the *PhaseLift* algorithm [5] and proved that exact recovery is possible with high probability in the noiseless case. Hand recently studied the robustness of PhaseLift and showed that it can tolerate a constant fraction of arbitrary errors [6]. Wirtinger-Flow (WF) [7] is a more recent approach that combines a good statistical initialization with a computationally light gradient-type refinement algorithm. The combination works very well when $\mathbf{A}$ is i.i.d. Gaussian. Following [7], conceptually similar approaches, namely, the truncated WF (TWF) [8] and truncated amplitude flow (TAF) algorithms [9], [10] were proposed to handle more challenging scenarios. In the recent work in [11], [12], the authors proposed a least-squares *feasible point pursuit* (LS-FPP) approach that aims to solve the same optimization problem as LS PhaseLift in the presence of noise. Simulations in [11] indicate that LS-FPP approaches the Cramér-Rao bound (CRB) for the Gaussian measurement model considered in [11].

Some phase retrieval algorithms were originally developed under an exact measurement model, and subsequently treated the noisy case by replacing equality constraints with relaxed inequality constraints [5]–[13]. Most of the existing algorithms were explicitly or implicitly developed under a Gaussian noise model. In certain applications, a subset of the measurements may be corrupted much more significantly than the others, and heavy-tailed noise may be encountered as well [14]–[16]. One representative example is high energy coherent X-ray imaging using a charge-coupled device (CCD), where the impulsive noise originates from X-ray radiation effects on the CCD, and the density of impulses increases with the intensity of X-ray radiation or CCD exposure time [16], [17]. Under such circumstance, modeling the noise as Gaussian is no longer appropriate.

Manuscript received October 4, 2016; revised March 20, 2017, June 6, 2017, and July 14, 2017; accepted July 14, 2017. Date of publication August 15, 2017; date of current version September 26, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xavier Mestre. The work of N. D. Sidiropoulos was supported by NSF CIF-1525194. *(Corresponding author: Nicholas D. Sidiropoulos.)*

C. Qian, X. Fu, and N. D. Sidiropoulos are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: alextoqc@gmail.com; xfu@umn.edu; nikos@umn.edu).

L. Huang is with the College of Information Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: dr.lei.huang@ieee.org).

J. Xie is with the Department of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: xj@hit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2017.2740166

In recent years, robust phase retrieval algorithms, e.g., [14], [15], [18], have been proposed to handle outliers. The framework in [15] considered an undersampled phase retrieval model corrupted with Laplacian-distributed outliers, but the approach was designed specifically for sparse $\mathbf{x}$. Variations of TWF to handle outliers were also considered. For example, the work in [18] replaces the sample mean that is used in TWF by sample median in the initialization and truncated update, which exhibits robustness to outliers under certain conditions [18]. However, as TWF, this approach still heavily relies on the assumption that Gaussian measurements are employed.

Another important aspect of phase retrieval is how noise enters the measurement model. Our previous work [11] considered the noise model $\mathbf{y} = |\mathbf{A}\mathbf{x}|^2 + \mathbf{n}$. Another noise model that is frequently considered in the literature [2]–[4] is

$$\mathbf{y} = |\mathbf{A}\mathbf{x}| + \mathbf{n}. \tag{1}$$

Notice the subtle but important difference between the two models: whether noise is added to the magnitude or the squared magnitude. The choice hinges on the experimental setup, including the measurement apparatus; but (1) is more widely adopted by experimentalists.

Contributions: We consider the phase retrieval problem under the model in (1) in the presence of impulsive noise, and focus on designing robust algorithms to handle it. To fend against impulsive noise, we adopt the ℓ_p-fitting ($0 < p < 2$) based estimator that is known to be effective in dealing with outliers, and devise two optimization algorithms using two-block inexact alternating optimization. Specifically, the two algorithms both solve local majorization-based approximate subproblems for one block, instead of exactly solving the conditional block minimization problem to optimality. Interestingly, starting from different majorizations, the resulting solutions turn out equivalent to iteratively reweighted least squares and gradient descent, respectively. Unlike the existing inexact and exact alternating optimization frameworks that mostly operate with convex constraints, the proposed algorithms work with a unit-modulus nonconvex constraint, so convergence analysis seems difficult. Nevertheless, we prove convergence of the proposed algorithms to a Karush-Kuhn-Tucker (KKT) point by exploiting the two-block structure of the problem. We also derive computationally light implementations using Nesterov-type and stochastic gradient updates. In order to assist in performance analysis and experimental design, we derive the CRB for the model in (1) and under different parameterizations, in the presence of Laplacian and Gaussian noise. Curiously, although related CRBs for different noisy measurement models have been previously derived in [11], [19]–[22], to the best of our knowledge, there is no available CRB for the model in (1) – and our work fills this gap. The proposed algorithms are validated by extensive simulations. The simulations show that our approaches outperform the state-of-the-art algorithms in the presence of impulsive noise.

A preliminary conference version of part of this work has been submitted to EUSIPCO 2016 [1]. The conference version includes one of the two basic algorithms, the Laplacian CRB, and limited simulations. The second algorithm, Nesterov acceleration and stochastic gradient-type updates, additional CRB results, and, most importantly, proof of convergence of the iterative algorithms are all provided only in this journal version, which naturally also includes more comprehensive simulation results.

Notation: Throughout the paper, we use boldface lowercase letters for vectors and boldface uppercase letters for matrices. Superscripts $(\cdot)^T$, $(\cdot)^*$, $(\cdot)^H$, $(\cdot)^{-1}$ and $(\cdot)^\dagger$ represent transpose, complex conjugate, conjugate transpose, matrix inverse and pseudo-inverse, respectively. The $\mathrm{Re}\{\cdot\}$ and $\mathrm{Im}\{\cdot\}$ denote the real part and imaginary part. $\mathbb{E}[\cdot]$ is the expectation operator, $|\cdot|$ is the absolute value operator, $||\cdot||_F$ is the Frobenius norm, $||\cdot||_p$ is the vector ℓ_p-norm, whose definition is $||\mathbf{x}||_p = (\sum_{i=1}^N |x_i|^p)^{1/p}$, $\odot$ is the element-wise product, and $\mathrm{diag}(\cdot)$ is a diagonal matrix with its argument on the diagonal. δ_{ij} denotes the Kronecker delta function, and $\angle(\cdot)$ takes the phase of its argument. $\mathbf{0}_m$, $\mathbf{1}_m$, and $\mathbf{I}_m$ stand for the $m \times 1$ all-zero vector, $m \times 1$ all-one vector, and $m \times m$ identity matrix, respectively. Furthermore, $\mathrm{trace}(\cdot)$ and $\partial a/\partial x$ denote the trace and partial derivative operators, respectively.

II. Proposed Algorithms

A. AltIRLS

Let us first consider the noiseless case where $\mathbf{y} = |\mathbf{A}\mathbf{x}|$. Effectively, it can also be written as

$$\mathbf{y} \odot \mathbf{u} = \mathbf{A}\mathbf{x} \tag{2}$$

where $\mathbf{u} = e^{j\angle(\mathbf{A}\mathbf{x})}$ is an auxiliary vector of unknown unit-modulus variables with its mth component being $u_m = e^{j\angle(\mathbf{a}_m^H\mathbf{x})}$. In the presence of impulsive noise, ℓ_p-(quasi)-norm has be recognized as an effective tool for promoting sparsity and fending against outliers [23]–[25]. Therefore, we propose to employ an ℓ_p-fitting based estimator instead of using the ℓ_2-norm, which has the form of

$$\min_{|\mathbf{u}|=\mathbf{1},\mathbf{x}} \sum_{m=1}^{M} \left(|y_m u_m - \mathbf{a}_m^H\mathbf{x}|^2 + \epsilon\right)^{p/2} \tag{3}$$

where $0 < p < 2$ is chosen to down-weigh noise impulses (i.e., outliers), and $\epsilon > 0$ is a small regularization parameter (e.g., $\epsilon \in [10^{-8}, 10^{-6}]$) that keeps the cost function within its differentiable domain, which will prove handy in devising an effective algorithm later.

To handle Problem (3), we follow the rationale of alternating optimization, i.e., we first update $\mathbf{x}$ fixing $\mathbf{u}$, and then we do the same for $\mathbf{u}$.

Assume that after some iterations, the current solution at iteration r is $(\mathbf{x}^{(r)}, \mathbf{u}^{(r)})$. At step $(r+1)$, the subproblem with respect to (w.r.t.) $\mathbf{x}$ is

$$\mathbf{x}^{(r+1)} = \arg\min_{\mathbf{x}} \sum_{m=1}^{M} \left(|y_m u_m^{(r)} - \mathbf{a}_m^H\mathbf{x}|^2 + \epsilon\right)^{p/2} \tag{4}$$

which is still difficult to handle. Particularly, when $p < 1$, the subproblem itself is still non-convex; when $p \geq 1$, the subproblem is convex but has no closed-form solution. Under such circumstances, we propose to employ the following lemma [26]:

Lemma 2.1: Assume $0 < p < 2$, $\epsilon > 0$, and $\phi_p(w) := \frac{2-p}{2}(\frac{2}{p}w)^{\frac{p}{p-2}} + \epsilon w$. Then, we have

$$\left(x^2 + \epsilon\right)^{p/2} = \min_{w \geq 0} \; wx^2 + \phi_p(w) \tag{5}$$

and the unique minimizer is

$$w_{\text{opt}} = \frac{p}{2}\left(x^2 + \epsilon\right)^{\frac{p-2}{2}}. \tag{6}$$

By Lemma 2.1, an upper bound of $\sum_{m=1}^{M}(|y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x}|^2 + \epsilon)^{p/2}$ that is tight at the current solution $\mathbf{x}^{(r)}$ can be easily found:

$$\sum_{m=1}^{M}\left(|y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x}|^2 + \epsilon\right)^{p/2} \leq \sum_{m=1}^{M}\left(w_m^{(r)}\left|y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x}\right|^2 + \phi_p\left(w_m^{(r)}\right)\right) \tag{7}$$

where

$$w_m^{(r)} := \frac{p}{2}\left(\left|y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x}^{(r)}\right|^2 + \epsilon\right)^{\frac{p-2}{2}} \tag{8}$$

and the equality holds if and only if $\mathbf{x} = \mathbf{x}^{(r)}$. Instead of directly dealing with Problem (4), we solve a surrogate problem using the right hand side (RHS) of (7) at each iteration to update $\mathbf{x}$. Notice that the RHS of (7) is convex w.r.t. $\mathbf{x}$ and the corresponding problem can be solved in closed-form:

$$\mathbf{x}^{(r+1)} = \left(\mathbf{W}^{(r)}\mathbf{A}\right)^{\dagger}\mathbf{W}^{(r)}\left(\mathbf{y} \odot \mathbf{u}^{(r)}\right) \tag{9}$$

where

$$\mathbf{W}^{(r)} = \text{diag}\left(\sqrt{w_1^{(r)}} \; \cdots \; \sqrt{w_M^{(r)}}\right). \tag{10}$$

The conditional problem w.r.t. $\mathbf{u}$ is

$$\mathbf{u}^{(r+1)} = \arg\min_{|\mathbf{u}|=\mathbf{1}} \sum_{m=1}^{M}\left(|y_m u_m - \mathbf{a}_m^H \mathbf{x}^{(r+1)}|^2 + \epsilon\right)^{p/2}. \tag{11}$$

Although the problem is non-convex, it can be easily solved to optimality. Specifically, the first observation is that the partial minimization w.r.t. $\mathbf{u}$ is insensitive to the value of p; i.e., given a fixed $\mathbf{x}$, for any $p > 0$, the solutions w.r.t. $\mathbf{u}$ are identical. Second, for all $p > 0$, the solution is simply to align the angle of $y_m u_m$ to that of $\mathbf{a}_m^H \mathbf{x}^{(r+1)}$, which is exactly

$$u_m^{(r+1)} = e^{j\angle\left(\mathbf{a}_m^H \mathbf{x}^{(r+1)}\right)}, \quad m = 1, \ldots, M. \tag{12}$$

We update $\mathbf{x}$ and $\mathbf{u}$ alternately, until some convergence criterion is met. We see that the way that we construct the upper bound of the partial problem w.r.t. $\mathbf{x}$ is in fact the same as the procedure in iteratively reweighted least squares (IRLS) [25], [26]. The difference is that we 'embed' this IRLS step into an alternating optimization algorithm. We therefore call this algorithm alternating IRLS (AltIRLS), which is summarized in Algorithm 1.

Algorithm 1: AltIRLS for phase retrieval.

```
function AltIRLS $(\mathbf{y}, \mathbf{A}, \mathbf{x}^{(0)})$
    Initialize $\mathbf{u}^{(0)} = \exp(\angle(\mathbf{A}\mathbf{x}^{(0)}))$ and $\mathbf{W}^{(0)} = \mathbf{W}^{(0)} = \text{diag}\left(\sqrt{w_1^{(0)}} \cdots \sqrt{w_M^{(0)}}\right)$ with $w_m^{(0)} = \frac{p}{2}\left(|y_m u_m^{(0)} - \mathbf{a}_m^H \mathbf{x}^{(0)}|^2 + \epsilon\right)^{\frac{p-2}{2}}$
    while stopping criterion has not been reached do
        $\mathbf{x}^{(r)} = (\mathbf{W}^{(r-1)}\mathbf{A})^{\dagger}\mathbf{W}^{(r-1)}(\mathbf{y} \odot \mathbf{u}^{(r-1)})$.
        $\mathbf{u}^{(r)} = \exp(j\angle(\mathbf{A}\mathbf{x}^{(r)}))$
        $w_m^{(r)} = \frac{p}{2}\left(\left|y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x}^{(r)}\right|^2 + \epsilon\right)^{\frac{p-2}{2}}, \forall m$
        $\mathbf{W}^{(r)} = \text{diag}\left(\sqrt{w_1^{(r)}} \cdots \sqrt{w_M^{(r)}}\right)$
        $r = r + 1$
    end while
end function
```

A relevant question regarding Algorithm 1 is whether or not this algorithm converges to a meaningful point, e.g., a stationary or KKT point. Note that the block variable $\mathbf{u}$ is constrained to a non-convex set, and we do not compute the optimal solution for the block variable $\mathbf{x}$ at each iteration of Algorithm 1. For such a type of algorithm, there is no existing theoretical framework that establishes convergence. We therefore need careful custom convergence analysis. We have the following result:

Proposition 2.1: Assume that $0 < p < 2$ and $\epsilon > 0$, and that $\mathbf{A}$ has full column rank. Then, the solution sequence produced by Algorithm 1 converges to a set $\mathcal{K}$ that consists of all the KKT points of Problem (3).

Proof: See Appendix A. ■

The result in Proposition 2.1 is interesting: Although the algorithm that we propose to compute the ℓ_p fitting-based estimator solves non-convex subproblems, it ensures convergence to a KKT point. Such a nice property is proven by exploiting the two-block structure of the algorithm. Note that the proof itself does not rely on the specific form of the optimization problem in (3), and thus can be easily extended to other two-block alternating optimization cases, which we believe is of much broader interest.

B. AltGD

Algorithm 1 uses a simple update strategy, but its complexity may still become an issue when the problem size grows. Specifically, the bottleneck of the AltIRLS algorithm lies in solving the subproblem w.r.t. $\mathbf{x}$, i.e.,

$$\min_{\mathbf{x}} \left|\left|\mathbf{W}^{(r)}(\mathbf{y} \odot \mathbf{u}) - \mathbf{W}^{(r)}\mathbf{A}\mathbf{x}\right|\right|_2^2 \tag{13}$$

whose closed-form solution can be computed via (9). It is observed from (9) that the matrix inversion part requires $\mathcal{O}(N^3)$ flops to compute and also $\mathcal{O}(N^2)$ memory to store, both of which are not efficient for high-dimensional $\mathbf{x}$ – e.g., when $\mathbf{x}$ is

Algorithm 2: AltGD for phase retrieval.

```
function AltGD ($\mathbf{y}, \mathbf{A}, \mathbf{x}^{(0)}$)
    Initialize $\mathbf{u}^{(0)} = \exp(\angle(\mathbf{A}\mathbf{x}^{(0)}))$
    while stopping criterion has not been reached do
        Choose $\mu^{(r-1)}$ as the leading eigenvalue of $\mathbf{A}^H(\mathbf{W}^{(r-1)})^2\mathbf{A}$ or the trace of $(\mathbf{W}^{(r-1)})^2$
        $\mathbf{x}^{(r)} = \mathbf{x}^{(r-1)} - 1/\mu^{(r-1)}\nabla f(\mathbf{x}^{(r-1)})$
        $\mathbf{u}^{(r)} = \exp(j\angle(\mathbf{A}\mathbf{x}^{(r)}))$
        $r = r + 1$
    end while
end function
```

a vectorized image, N is usually very large (more specifically, for a 100×100 image, N is 10,000).

To circumvent this difficulty, one way is to employ a certain fast least squares solver such as conjugate gradient [28] to handle Problem (13). This is a viable option, but such solvers still typically require many iterations to obtain a fairly accurate solution of (13). Here, we propose to deal with problem (13) using a simpler approach. Specifically, let us denote

$$f^{(r)}(\mathbf{x}) = \left\|\mathbf{W}^{(r)}(\mathbf{y} \odot \mathbf{u}^{(r)}) - \mathbf{W}^{(r)}\mathbf{A}\mathbf{x}\right\|_2^2 \tag{14}$$

and approximate (13) using the following function

$$g^{(r)}(\mathbf{x}) = f^{(r)}(\mathbf{x}^{(r)}) + \mathrm{Re}\{(\nabla f^{(r)}(\mathbf{x}^{(r)}))^H(\mathbf{x} - \mathbf{x}^{(r)})\} + \frac{\mu^{(r)}}{2}\left\|\mathbf{x} - \mathbf{x}^{(r)}\right\|_2^2. \tag{15}$$

where $\mu^{(r)} \geq 0$ is a pre-specified parameter. Instead of optimizing $f^{(r)}(\mathbf{x})$, we optimize $\mathbf{g}^{(r)}(\mathbf{x})$. By rearranging terms, the subproblem becomes

$$\mathbf{x}^{(r+1)} = \arg\min_{\mathbf{x}} \left\|\mathbf{x} - \left(\mathbf{x}^{(r)} - \frac{1}{\mu^{(r)}}\nabla f(\mathbf{x}^{(r)})\right)\right\|_2^2 \tag{16}$$

and the solution is

$$\mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} - \frac{1}{\mu^{(r)}}\nabla f(\mathbf{x}^{(r)}) \tag{17}$$

i.e., a gradient step with step-size $1/\mu^{(r)}$, where the gradient is

$$\nabla f^{(r)}(\mathbf{x}^{(r)}) = \mathbf{A}^H\left(\mathbf{W}^{(r)}\right)^2\left(\mathbf{A}\mathbf{x}^{(r)} - \mathbf{y} \odot \mathbf{u}^{(r)}\right) \tag{18}$$

which does not require any matrix inversion operation. Also, the $N \times N$ matrix does not need to be stored, if we take the order $\mathbf{A}\mathbf{x}^{(r)} \rightarrow \left(\mathbf{W}^{(r)}\right)^2\mathbf{A}\mathbf{x}^{(r)} \rightarrow \mathbf{A}^H\left(\mathbf{W}^{(r)}\right)^2\mathbf{A}\mathbf{x}^{(r)}$ to compute the update of $\mathbf{x}$. Therefore, using this approach, both memory and complexity requirements are less demanding. We summarize the algorithm in Algorithm 2. We call this algorithm alternating gradient descent (AltGD).

Note that when μ is chosen as the Lipschitz constant, (15) is an upper bound of the cost function in (13). In other words, we have

$$f^{(r)}(\mathbf{x}) \leq g^{(r)}(\mathbf{x})$$

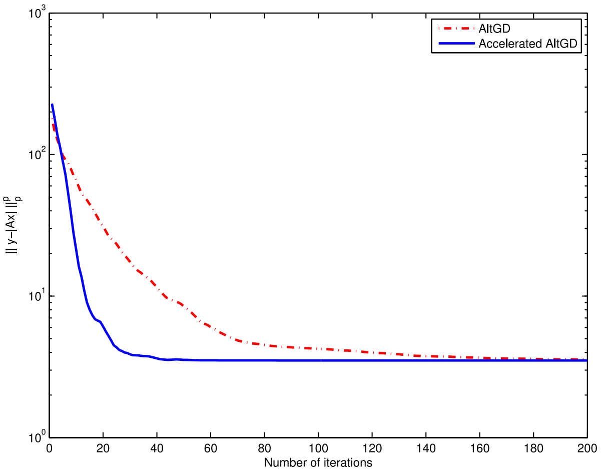


Fig. 1. Cost function value versus number of iterations.

if $\mu^{(r)} \geq \lambda_{\max}(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A})$, and the equality holds if and only if $\mathbf{x} = \mathbf{x}^{(r)}$. Therefore, the algorithmic structure of Algorithm 2 is the same as that of Algorithm 1, except that the majorizing functions of the $\mathbf{x}$-block are different – which means that the proof of convergence of Algorithm 1 also applies here:

Corollary 2.1: If $\mu^{(r)} \geq \lambda_{\max}(\mathbf{A}^T(\mathbf{W}^{(r)})^2\mathbf{A})$ for all r, then the whole sequence produced by Algorithm 1 converges to $\mathcal{K}$.

Remark 2.1: Exactly computing $\lambda_{\max}(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A})$ may be time consuming in practice. Many practically easier ways can be employed, e.g., the Armijo rule-based line search [27]. In our simulations, we use a simple heuristic: we let $\mu^{(r)} = \mathrm{trace}((\mathbf{W}^{(r)})^2)$ instead of $\lambda_{\max}(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A})$. The rationale behind is that we observe that the energy of $\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A}$ is usually dominated by $\mathbf{W}^{(r)}$, and using $\mu^{(r)} = \mathrm{trace}((\mathbf{W}^{(r)})^2)$ is a good approximation of $\mathrm{trace}(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A})$ that is an upper bound of $\lambda_{\max}(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A})$. We should mention that this step-size choice is a heuristic, but works well in practice, as will be shown in the simulations.

C. Further Reducing Complexity

1) Extrapolation: Algorithm 2 is easier to compute than Algorithm 1 in terms of per-iteration complexity. However, first-order methods tend to require more iterations in total. One way to alleviate this effect is to incorporate Nesterov's "optimal first-order" method [29]–[31], i.e., for each update, we set

$$\mathbf{z}^{(r)} = \mathbf{x}^{(r)} + \frac{t^{(r-1)} - 1}{t^{(r)}}\left(\mathbf{x}^{(r)} - \mathbf{x}^{(r-1)}\right) \tag{19}$$

$$t^{(r)} = \frac{1 + \sqrt{1 + 4(t^{(r-1)})^2}}{2} \tag{20}$$

$$\mathbf{x}^{(r+1)} = \mathbf{z}^{(r)} - \frac{1}{\mu^{(r)}}\nabla f^{(r)}(\mathbf{z}^{(r)}) \tag{21}$$

In practice, the above 'extrapolation' technique greatly expedites the whole process in various applications [23], [32]. Some numerical evidence can be seen in Fig. 1, where a simple comparison between the plain Algorithm 2 and the extrapolated version is presented. We choose SNR $= 20$ dB, $N = 16$ and $M = 128$. Each element in the signal and measurement vectors is independently drawn from the complex circularly symmetric Gaussian distribution with mean zero and variance one. The noise is generated from a symmetric α stable (SαS distribution) which will be described in detail in Section IV. Fig. 1 shows the convergence of AltGD and accelerated AltGD using extrapolation when $p = 1.3$. As we can see, the accelerated AltGD converges after 40 iterations which is much faster than AltGD that converges after 200 iterations.

2) Block Incremental / Stochastic Gradient: When N is very large (thus $M > N$ is larger), even gradient computation is too much of a burden. In such cases, a pragmatic way is to "break down" the problem to pieces and do (block) incremental or stochastic gradient. We divide the measurement matrix into L blocks Γ_l. Then it is straightforward to get the gradient for the lth block as

$$\nabla f_{\Gamma_l}(\mathbf{x}^{(r)}) = \mathbf{A}_{\Gamma_l}^H \left(\mathbf{W}_{\Gamma_l}^{(r)}\right)^2 \left(\mathbf{A}_{\Gamma_l}\mathbf{x}^{(r)} - \mathbf{y}_{\Gamma_l} \odot \mathbf{u}_{\Gamma_l}^{(r)}\right). \tag{22}$$

The estimate of $\mathbf{x}$ is updated according to

$$\mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} - \frac{1}{\mu_{\Gamma_l}^{(r)}} \nabla f_{\Gamma_l}(\mathbf{x}^{(r)}) \tag{23}$$

where $\mu_{\Gamma_l}^{(r)}$ is chosen as the leading eigenvalue of $\mathbf{A}_{\Gamma_l}^H \mathbf{A}_{\Gamma_l}$ or $\mathrm{trace}((\mathbf{W}_{\Gamma_l}^{(r)})^2)$. The algorithm can proceed block by block with revisits, or by randomly picking blocks, resulting in block incremental gradient and stochastic gradient versions, respectively.

A subtle point here is that, to maintain robustness, one should choose a block size larger than one. The reason is that the robustness of the algorithm is brought by treating different y_m with different weights (more specifically, by downweighting the outliers). When using only one y_m for updating, this ability vanishes.

III. Cramér-Rao Bound Analysis

In this section, the CRB on the accuracy of retrieving $\mathbf{x}$ in (1) is presented. The CRB provides a lower bound on the MSE of unbiased estimators. In many signal processing applications, one cannot guarantee that an estimator is unbiased, or even that an unbiased estimator exists, yet the CRB is still surprisingly predictive of estimator performance [35], [38]. For phase retrieval under Gaussian noise, over the past few years, several CRBs have been derived for different models, e.g., 2-D Fourier-based measurements [19], noise added prior to taking the magnitude [20] and noise added after taking the magnitude square [11], [21]. To the best of our knowledge, there is no available CRB formula for the signal model in (1). Here, we present the CRBs for two particular types of noise: Laplacian and Gaussian noise. Although our main interest here lies in evaluating performance of robust algorithms and the Laplacian CRB serves this purpose, the Gaussian CRB is of interest in other application contexts. Note that we use subscripts *r, c, L* and *G* to stand for real, complex, Laplacian and Gaussian, respectively.

To get started, we should note that our derivations are based on the assumption that $\mathbf{a}_m^H\mathbf{x}$ is nonzero. That is because the term $|\mathbf{Ax}|$ is non-differentiable at $\mathbf{a}_m^H\mathbf{x} = 0$, the Fisher information only exists at $\mathbf{a}_m^H\mathbf{x} \neq 0$. With this caveat, we have the following proposition:

Proposition 3.1: In Laplacian noise, the CRB is

$$\mathrm{CRB}_{L,c} = \mathrm{trace}\left(\mathbf{F}_{L,c}^{\dagger}\right) \tag{24}$$

where

$$\mathbf{F}_{L,c} = \frac{2}{\sigma_n^2}\mathbf{G}_{L,c}\,\mathrm{diag}(|\mathbf{Ax}|^{-2})\,\mathbf{G}_{L,c}^T \tag{25}$$

with

$$\mathbf{G}_{L,c} = \begin{bmatrix} \mathrm{Re}\{\mathbf{A}^H \mathrm{diag}(\mathbf{Ax})\} \\ \mathrm{Im}\{\mathbf{A}^H \mathrm{diag}(\mathbf{Ax})\} \end{bmatrix}. \tag{26}$$

Proof: See Appendix B. ■

The CRB for real $\mathbf{x}$ is a special case of the complex case, which can be easily derived from Proposition 3.1:

Proposition 3.2: In Laplacian noise, the CRB of real-valued $\mathbf{x}$ is

$$\mathrm{CRB}_{L,r} = \mathrm{trace}\left(\mathbf{F}_{L,r}^{-1}\right) \tag{27}$$

where

$$\mathbf{F}_{L,r} = \frac{2}{\sigma_n^2}\mathbf{G}_{L,r}\,\mathrm{diag}(|\mathbf{Ax}|^{-2})\,\mathbf{G}_{L,r}^T \tag{28}$$

with

$$\mathbf{G}_{L,r} = \mathrm{Re}\{\mathbf{A}^H \mathrm{diag}(\mathbf{Ax})\}. \tag{29}$$

In Appendix C, we show that when $\mathbf{A}$ has full column rank, $\mathbf{F}_{L,c}$ is singular with rank $(2N-1)$ while $\mathbf{F}_{L,r}$ is always nonsingular. Thus, for complex $\mathbf{x}$, we adopt its pseudo-inverse to compute an optimistic (looser) CRB, which is still a valid lower bound that can be used to benchmark the efficiency of any biased estimator [33]–[35]. If $\mathbf{x}$ is close to zero, then the CRB is not tight at low SNRs, and more measurements should be used to approach the CRB.

Proposition 3.3: In the Gaussian noise case, the CRB is two times larger than the CRB in Proposition 3.1, i.e.,

$$\mathrm{CRB}_G = 2\mathrm{CRB}_L. \tag{30}$$

Proof: The proof is straightforward by calculating the Gaussian FIM and comparing it with the Laplacian FIM in (28). We omit it here. ■

In certain applications of phase retrieval, we may be more interested in the performance bound for retrieving the phase of $\mathbf{x}$. The following proposition provides the CRB on both the phase and amplitude of $\mathbf{x}$ under Laplacian noise. Note that in the Gaussian noise case, it is straightforward to apply Proposition 3.3 to compute the lower bound. Similar results can also be found in [36].

Proposition 3.4: In Laplacian noise, the CRB of the amplitude of $\mathbf{x}$ is

$$\mathrm{CRB}_{L,|\mathbf{x}|} = \sum_{i=1}^{N} d_i \tag{31}$$

and the variance of any biased estimate of the phase of $\mathbf{x}$ is bounded below by

$$\mathrm{CRB}_{L,\angle(\mathbf{x})} = \sum_{i=N+1}^{2N} d_i \tag{32}$$

where $d = [d_1 \ \cdots \ d_{2N}]$ contains the main diagonal elements of $\mathbf{F}_L^{\dagger}$ which in this case is defined as

$$\mathbf{F}_L = \frac{2}{\sigma_n^2} \mathbf{G}_L \operatorname{diag}(|\mathbf{Ax}|^{-2})\, \mathbf{G}_L^T \tag{33}$$

with

$$\mathbf{G}_L = \begin{bmatrix} \operatorname{diag}(|\mathbf{x}|)^{-1} & \\ & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \operatorname{Re}\left\{\operatorname{diag}\left(\mathbf{x}^*\right) \mathbf{A}^H \operatorname{diag}(\mathbf{Ax})\right\} \\ \operatorname{Im}\{\operatorname{diag}(\mathbf{x}^*) \mathbf{A}^H \operatorname{diag}(\mathbf{Ax})\} \end{bmatrix}. \tag{34}$$

Proof: See [11] and Appendix B. ■

Remark 3.1: In deriving the CRB in Proposition 3.4, we use no additional assumptions on $\mathbf{x}$. Therefore, Proposition 3.4 works for both real and complex $\mathbf{x}$. Specifically, in the real case, the phase is actually the sign of $\mathbf{x}$. Here, $\mathbf{F}_L$ is also singular with rank deficit equal to one, so we adopt its pseudo-inverse to compute the CRB. We omit the proof of rank-1 deficiency of $\mathbf{F}_L$, since it follows the line of argument in Appendix C.

IV. Simulation Results

In this section, we evaluate the performance of the proposed methods and compare them with some existing algorithms, namely, WF [7], TWF [8], TAF [9], AltMinPhase [4], MTWF [18] and GS [2] in terms of MSE performance, where the MSE is computed after removing the global phase ambiguity between the true and estimated $\mathbf{x}$. In the simulations, we consider an exponential signal $\mathbf{x} = \exp(j0.16\pi t)$ comprising 16 samples, i.e., $t = 1, \cdots, 16$. We test the algorithms using different types of $\mathbf{A}$, i.e., random matrix that is usually employed in [4], [7]–[9], [18], and the 2D Fourier matrix that is widely used in real devices [2], [3] (cf. subsection IV-E). For the random measuring matrix case (used in subsections A-D), the measurement vectors are generated from a masked Fourier transformation, which takes the form of

$$\mathbf{A} = \left[\, (\mathbf{D}\boldsymbol{\Lambda}_1)^T \ \cdots \ (\mathbf{D}\boldsymbol{\Lambda}_K)^T \,\right]^T$$

where $K = M/N$ is the number of masks, $\mathbf{D}$ is a $N \times N$ discrete Fourier transform (DFT) matrix with $\mathbf{D}\mathbf{D}^H = N\mathbf{I}_N$ and $\boldsymbol{\Lambda}_k$ is a $N \times N$ diagonal masking matrix with its diagonal entries generated by $b_1 b_2$, where b_1 and b_2 are independent and distributed as [5]

$$b_1 = \begin{cases} 1 & \text{w. prob. } 0.25 \\ -1 & \text{w. prob. } 0.25 \\ -j & \text{w. prob. } 0.25 \\ j & \text{w. prob. } 0.25 \end{cases} \quad \text{and } b_2 = \begin{cases} \sqrt{2}/2 & \text{w. prob. } 0.8 \\ \sqrt{3} & \text{w. prob. } 0.2. \end{cases}$$

All results are obtained using a computer with 3.1 GHz Intel Xeon E31225 and 16 GB RAM. For the random measurement case, the algorithms are all initialized from the same starting point that is computed by picking the principal eigenvector of $\sum_{i=1}^{M} y_i^2 \mathbf{a}_i \mathbf{a}_i^H$, and the stopping criterion is

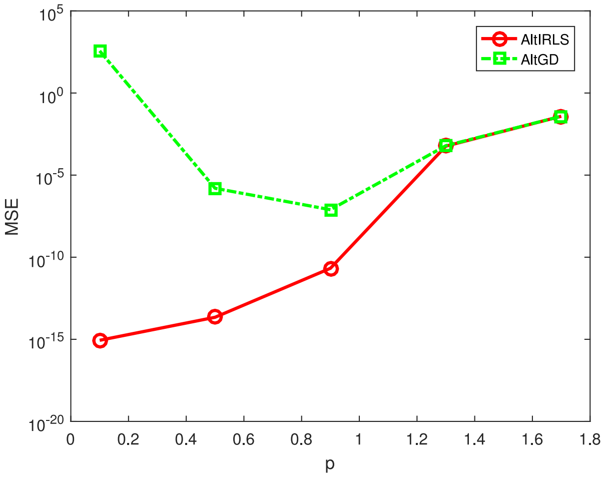


Fig. 2. MSE versus p.

$$\frac{\left| \, ||y - |\mathbf{Ax}^{(r)}|||_2^2 - ||y - |\mathbf{Ax}^{(r-1)}|||_2^2 \, \right|}{||y - |\mathbf{Ax}^{(r-1)}|||_2^2} \leq 10^{-7}$$

or the number of iterations reaching 1000. Furthermore, we use AltGD with extrapolation for the simulations.

A. Selection of p

Before we do the performance comparison, let us study how p affects the performance of the proposed methods. In this example, SNR is fixed at 20 dB, where the SNR is computed via

$$\mathrm{SNR} = 10 \log_{10} \left(\frac{||\mathbf{Ax}||^2}{||\mathbf{n}||^2} \right).$$

We assume that 10% of the data are corrupted by outliers that are generated from the Gaussian distribution with mean zero and variance 100, and the remaining elements in $\mathbf{n}$ are zero. $K = 8$ masks are employed to generate the measurements. Fig. 2 shows the MSE as a function of p. Note that, when p is smaller than 1, in order to achieve convergence in a non-convex setting, we initialize our methods in a two-step way. Specifically, for each of the proposed methods, when $0.6 < p < 1$, we use the spectrum output to initialize our methods with $p = 1.3$ and 100 iterations, and then choose the corresponding output to do another 100 iterations with $p = 1$ and use the output as a final starting point. When $p \leq 0.6$, we take an additional intermediate step with $p = 0.7$ and 100 iterations to gradually stage the initialization process for our approaches, since in this p-regime the subproblem is strictly non-convex. It is observed that generally, $p \leq 1$ provides better performance than $p > 1$, especially for AltIRLS. When $p < 0.5$, AltGD suffers severe performance degradation while AltIRLS still performs well. Our understanding is that first-order algorithms are in general more sensitive to problem conditioning, and a small p can lead to badly-conditioned $\mathbf{W}^{(r)}$. Working with $p < 1$ usually requires much more careful

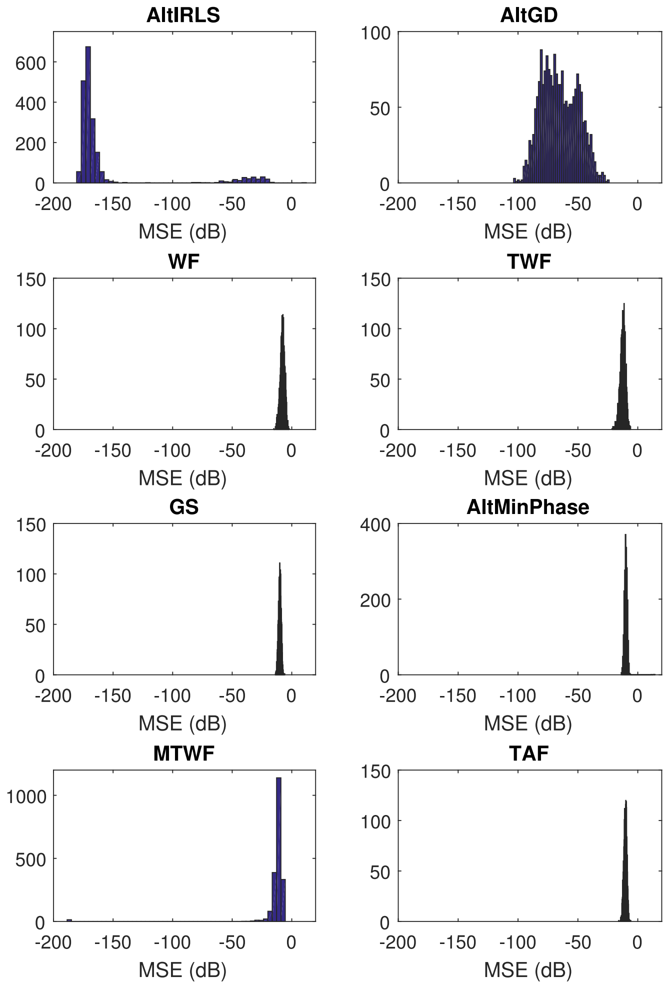


Fig. 3. Signal recovery performance comparison.

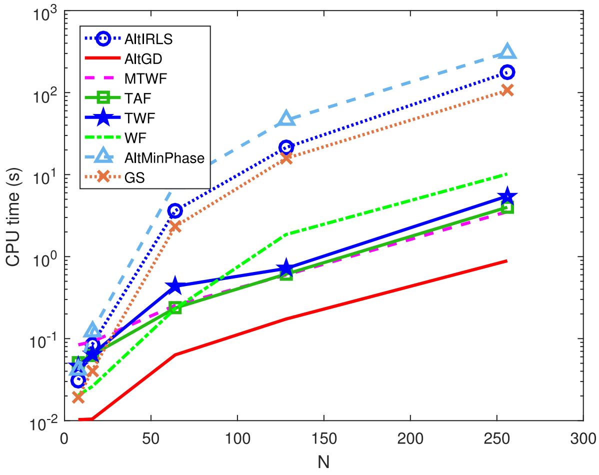


Fig. 4. CPU time versus number of variables N in GMM noise.

initialization. In our experience, $p \approx 1.3$ strikes a good compromise between numerical stability and estimation accuracy.

To illustrate the performance provided by our proposed methods with $p < 1$ over the state-of-art algorithms including TWF, WF, TAF, AltMinPhase, MTWF, and GS, we present the following example, where the parameters are the same as for Fig. 2, except that 20% of the data are corrupted by outliers. Note that among these competitors, TAF, AltMinPhase and GS share the same magnitude measurement model in (1) as ours, but WF is designed under the energy measurement model (i.e., $y_m = |\mathbf{a}_m^H \mathbf{x}|^2 + n_m$). In addition, TWF and MTWF adopt a Poisson model with $\lambda = |\mathbf{a}_i^H \mathbf{x}|^2$. Hence, $\mathbf{y}^2$ is fed to WF, TWF and MTWF, where the squaring is element-wise. We set $p = 0.4$ for the proposed methods, and perform Monte-Carlo trials to estimate $\text{MSE} = 10\log_{10}(||\hat{\mathbf{x}} - \mathbf{x}||_2^2)$. We show the results of 1,000 random trials in Fig. 3. One can see that for most trials the benchmark algorithms fail to give reasonable results. However, our methods produce much lower MSEs, which indicates that our proposed methods are efficient in suppressing outliers.

B. Statistical Performance Comparison

We now compare the MSE performance as a function of SNR. The results are averaged over 500 Monte-Carlo trials. To simulate outliers, we use the Laplacian distribution, the α-stable distribution, and a Gaussian mixture model (GMM) to generate heavy-tailed noise $\mathbf{n}$, respectively. Specifically, the outlier generating process is summarized below:

1) Laplacian: The probability density function (PDF) of the Laplacian distribution is given in (52).

2) α-stable: The PDF of an α-stable distribution is generally not available, but its characteristic function can be written in closed-form as

$$\phi(t; \alpha, \beta, c, \mu) = \exp\left(jt\mu - \gamma^{\alpha} |t|^{\alpha} \left(1 - j\beta \text{sgn}(t)\Phi(\alpha)\right)\right)$$

where $\Phi(\alpha) = \tan(\alpha\pi/2)$, $0 < \alpha \leq 2$ is the stability parameter, $-1 \leq \beta \leq 1$ is a measure of asymmetry, $\gamma > 0$ is a scale factor which measures the width of the distribution and μ is a shift parameter. There are two special cases that admit closed-form PDF expression, that is, $\alpha = 1$ and $\alpha = 2$ which respectively yield the Cauchy and Gaussian distributions. For $\alpha < 2$, $\phi(t; \alpha, \beta, \gamma, \mu)$ possesses heavy tails, and thus is considered suitable for simulating impulsive noise. The parameter α controls the density of impulses - smaller α's correspond to more outliers. In the following, we set $\alpha = 0.8$, and the other parameters are $\beta = 0$, $\gamma = 2$ and $\mu = 0$, resulting in a symmetric α-stable (SαS) distribution with zero-shift.

3) GMM: In our simulations, we also use a two-component GMM to generate impulsive noise, whose PDF is as follows:

$$p(n) = \sum_{i=1}^{2} \frac{c_i}{\sqrt{2\pi\sigma_i^2}} \exp\left\{-\frac{|n|^2}{2\sigma_i^2}\right\} \tag{35}$$

where σ_i^2 is the variance of the ith term, and $0 \leq c_i \leq 1$ is the probability of occurrence of the ith component. Note that we have $c_1 + c_2 = 1$, and we set $c_2 < c_1$ and $\sigma_2^2 > \sigma_1^2$ – therefore, the second component corresponds to outliers. In this example, we choose $c_1 = 0.9$, $c_2 = 0.1$, $\sigma_1^2 = 0.1$ and $\sigma_2^2 = 100$, which corresponds to a situation where strong outliers are present.

Figs. 5, 6 and 7 show the MSEs of $\hat{\mathbf{x}}$ under Laplacian, SαS and GMM noise, respectively. In the simulations, the number of measurements is set to be $M = 8N$. Note that for SαS and

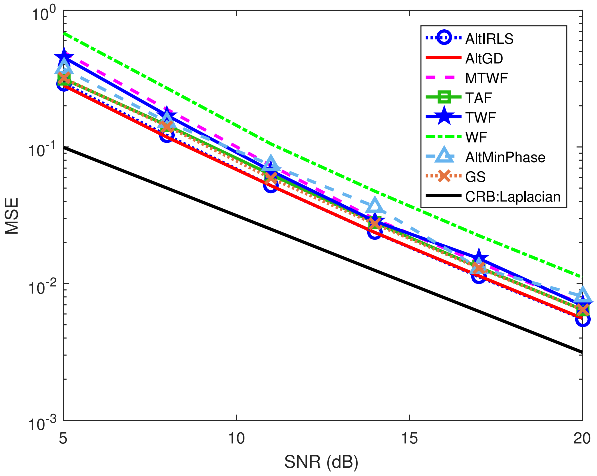


Fig. 5. MSE versus SNR in Laplacian noise.

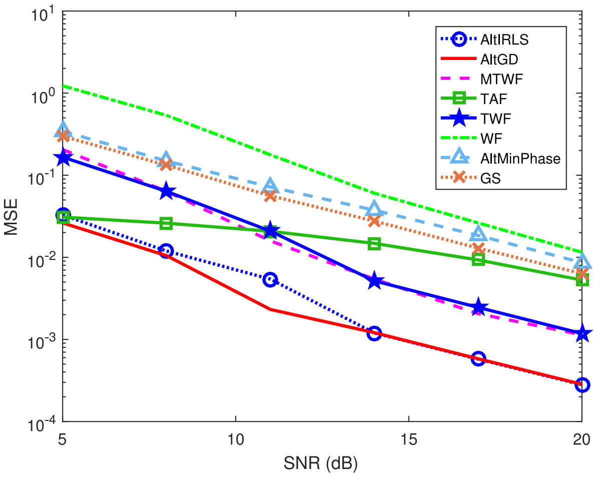


Fig. 7. MSE versus SNR in GMM noise.

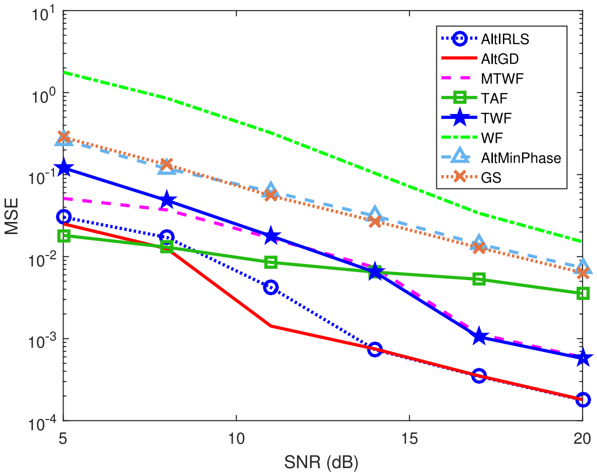


Fig. 6. MSE versus SNR in SαS noise.

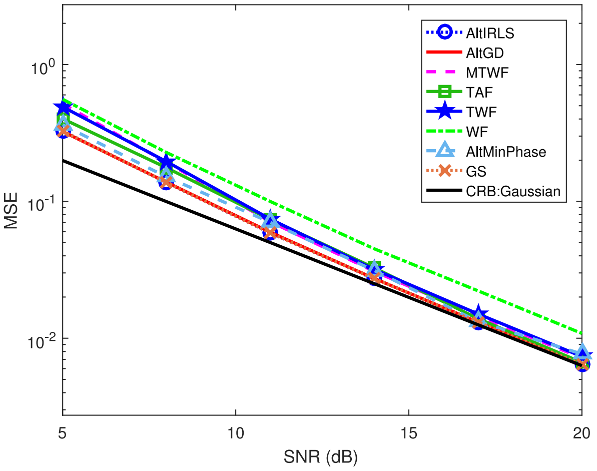


Fig. 8. MSE versus SNR in Gaussian noise.

GMM noise, we set $p = 1.3$ for AltIRLS and AltGD, while for Laplacian noise we let $p = 1$ which corresponds to the maximum likelihood estimator. In the Laplacian noise case, we include the CRB derived in Proposition 3.2 as a performance benchmark. It is observed from Fig. 5 that our approaches outperform the GS algorithm and yield the MSEs which are closest to the CRB. The performance gap between the proposed methods and their competitors becomes larger in the SαS and GMM noise cases since the outliers become stronger; see Fig. 6 and 7. This indicates that the proposed algorithms work better in more critical situations, i.e., when more severe outliers exist. Note that the MTWF performs slightly better than TWF in GMM noise and its performance is inferior to our methods in the three noise scenarios.

As a reference, we also plot the MSE performance versus SNR in Gaussian noise in Fig. 8, where we set $p = 2$ for having a maximum likelihood estimator. We compare the performance of the algorithms with the CRB in Proposition 3.4. One can see that the algorithms perform similarly when the noise follows an i.i.d. Gaussian distribution. This suggests that the Gaussian noise case may be an easier case - where all the algorithms under test perform reasonably.

Fig. 4 compares the CPU time as a function of N, where SNR is 10 dB, N increases from 8 to 128 and the other parameters are the same as Fig. 7. It is seen that AltGD is the fastest[1].

C. Success Rate Performance Comparison

Fig. 9 shows the success rate versus the proportion of outliers in the measurements, where the GMM is employed to control the outlier fraction. We vary c_2 from 0 to 0.5 and fix $\sigma_1^2 = 0$ and $\sigma_2^2 = 100$. SNR is 10 dB. For the proposed methods, we choose $p = 0.4$ for comparison. We declare *success* when the squared Euclidean distance of the estimate $\hat{\mathbf{x}}$ from the ground-truth $\mathbf{x}$ is smaller than or equal to 10^{-4}; i.e., the success rate is

[1] Note that in the simulation we use the heuristic step size as stated in Remark 2.1; if the step size $\mu = \lambda_{\max}(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A})$ is used, AltGD may consume more time.

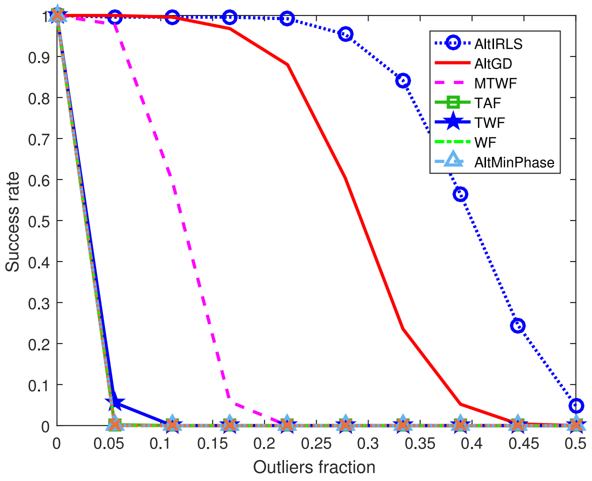


Fig. 9. Success rate of exact recovery versus outlier fraction in GMM noise.

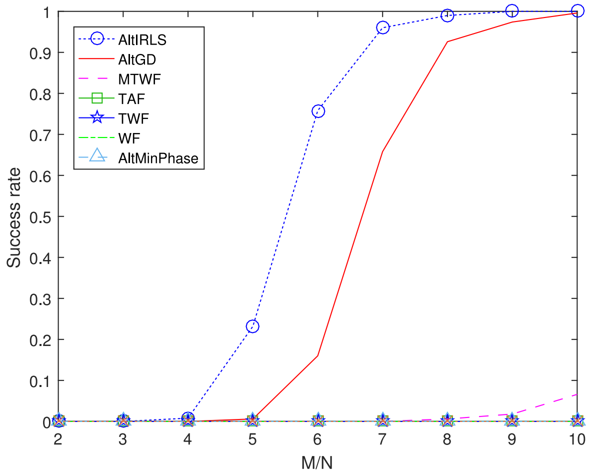


Fig. 10. Success rate of exact recovery versus M/N under GMM noise.

computed as

$$\text{success rate} = \frac{1}{500}\sum_{i=1}^{500}\gamma_i \tag{36}$$

where

$$\gamma_i = \begin{cases} 1, & \text{if } \|\hat{\mathbf{x}}_i - \mathbf{x}\|_2^2 \le 10^{-4} \\ 0, & \text{otherwise.} \end{cases} \tag{37}$$

Fig. 9 shows that our methods outperform the benchmark methods by a large margin in terms of success rate. When the outlier fraction is greater than 0.25, our methods can still estimate $\mathbf{x}$ quite accurately, while the other six competitors all fail. Note that the performance of AltIRLS is better than AltGD – but the complexity of AltGD is much lower.

Fig. 10 plots the success rate versus sample complexity (measured in terms of M/N with $N = 16$), where SNR is 10 dB and 20% of the data are corrupted by outliers. It is obvious that our two algorithms have the highest success rates when $M/N \ge 5$, while the other algorithms have very low success rates even when $M/N \ge 8$.

D. Performance Comparison on Caffeine Molecule Image Data

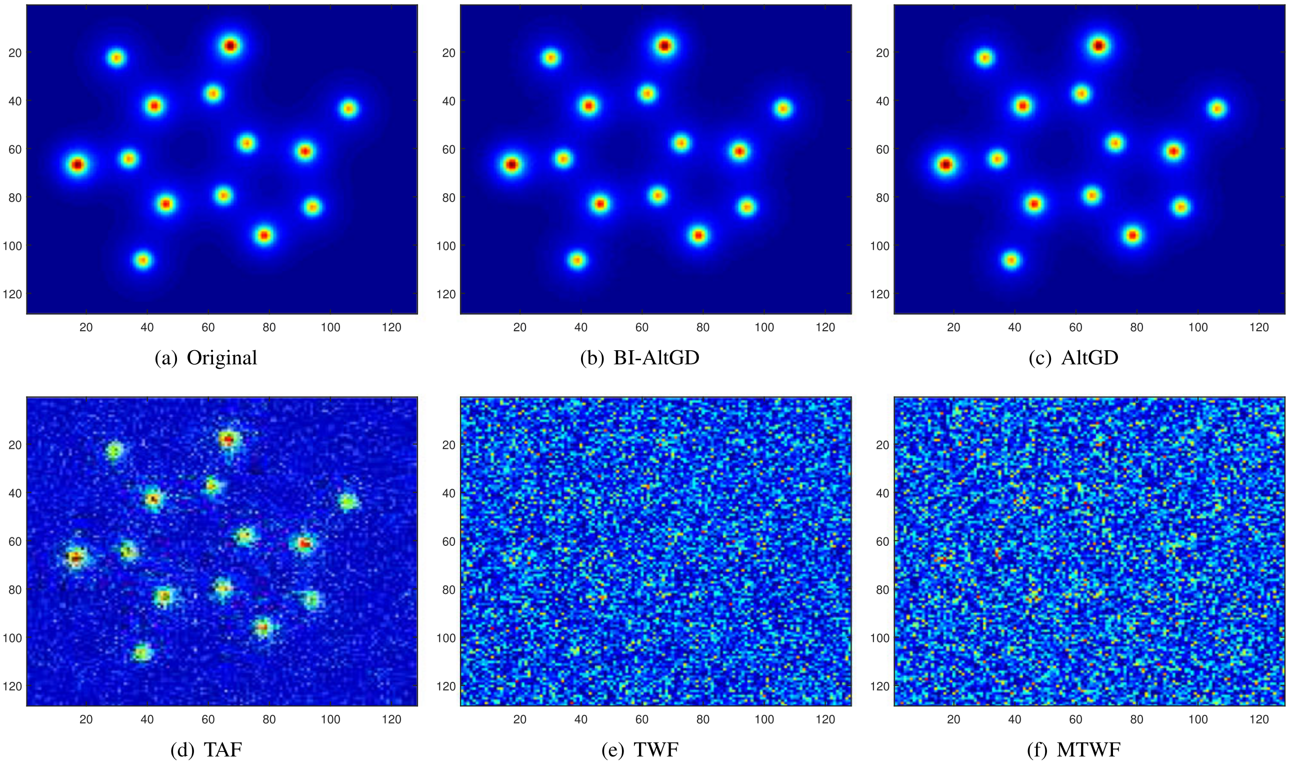
In this subsection, we showcase the performance using real caffeine molecule data that is often used to test phase retrieval algorithms [7]. The caffeine molecule data with size 128 × 128 is the projection of electron density of a 3-D Caffeine molecule's density map onto the xy-plane. The objective is to reconstruct the data from the magnitude of its masked Fourier transform. Here, we use the same measuring process as in the previous simulations and $K = 8$ masks are used to generate measurements. We compare the performance of AltGD, TAF, MTWF, and TWF. We also include the block incremental (BI) implementation of AltGD (cf. Section II.C2), which is referred to as *BI-AltGD*. The outliers are generated from GMM with $c_2 = 0.3$, $\sigma_1^2 = 0$ and $\sigma_2^2 = 100$, which indicates that there are 30% of the data are corrupted by outliers. SNR is 0 dB and the data is normalized with unit norm. Fig. 11 plots the retrieved molecule's density map using AltGD, BI-AltGD with $p = 1.3$, TAF, MTWF and TWF. It is seen in Fig. 11 that our schemes still work well, while the other competitors yield blurred images.

E. Performance Comparison with 2D Fourier Measurements

In the above simulations, the measurement vectors are generated from coded diffraction pattern (CDP) with random masks. The success of the existing algorithms such as the well-known PhaseLift, WF and its variants are all based on the randomness of measurements. However, to the best of our knowledge, the CDP model is still considered to be impractical, as there has been no device supporting the masking technique yet. In practice, most devices record the intensity of the Fourier transform of the object, where the intensity is usually proportional to the magnitude of the oversampled 2D Fourier measurements with 2× oversampling. In this case, the measurement matrix $\mathbf{A}$ is a Kronecker product of two oversampled Fourier matrices.

With 2D Fourier measurements, the most successful method is Fienup's hybrid-input-output (HIO) algorithm [3], which has been observed to give empirically reliable phase retrieval performance, especially in the noiseless case. In the noisy case, HIO can also provide good initialization for other algorithms, e.g., GS. Therefore, HIO and HIO-initialized GS are employed as baselines in this subsection.

In this simulation, we compare the performance of AltGD, HIO and GS under 2D Fourier measurements in the presence of outliers, where GMM-type outliers are generated with $c_2 = 0.1$, $\sigma_1^2 = 0$ and $\sigma_2^2 = 100$. Each entry of the signal matrix $\mathbf{X} \in \mathbb{R}^{16\times16}$ is generated from the normal distribution. After zero-padding around $\mathbf{X}$, we have $\tilde{\mathbf{X}} \in \mathbb{R}^{32\times32}$. Then the measurements are obtained by first calculating 2D fast Fourier transformation (FFT) on $\tilde{\mathbf{X}}$ and then taking the magnitude of 2D FFT$(\tilde{\mathbf{X}})$, where the sizes of left and right Fourier matrices are both 32 × 32. Note that SNR is 10 dB and is computed as $10\log_{10}(\|\tilde{\mathbf{X}}\|_F^2/\|\mathbf{N}\|_F^2)$ where $\mathbf{N} \in \mathbb{R}^{32\times32}$ is the noise matrix. For AltGD and GS, HIO with 5000 iterations is employed to provide an initial estimate of $\mathbf{X}$, and then an additional 5000 iterations are used in AltGD and GS, respectively. Moreover, HIO with 10000 iterations is included as a baseline. For AltGD, we



(a) Original (b) BI-AltGD (c) AltGD

(d) TAF (e) TWF (f) MTWF

Fig. 11. Retrieving molecule image in SαS noise.

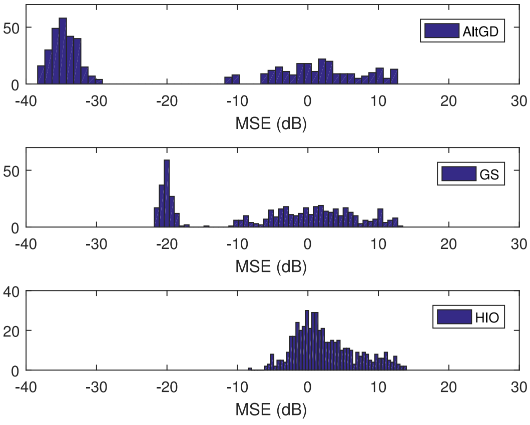
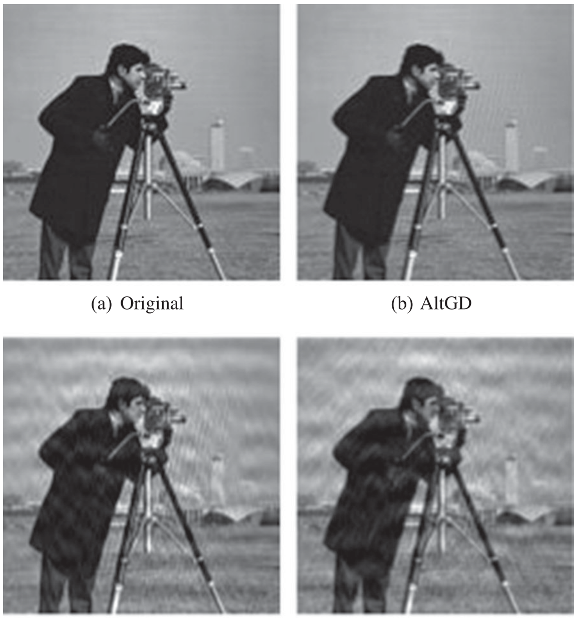


Fig. 12. Signal recovery performance comparison with 2D Fourier measurements in GMM noise.

choose $p = 1.3$. The MSEs obtained from 1000 Monte-Carlo tests are plotted in Fig. 12. It is seen that HIO does not perform very well, and is inferior to AltGD and GS. AltGD has better performance than GS, since most of its MSEs are concentrated around -35 dB, while those obtained by GS are located around -20 dB.

Finally, Fig. 13 shows the recovered image of 128×128 Cameraman from 2D Fourier measurements under GMM noise, where SNR is 12 dB and the noise is generated in the same way as for Fig. 12. The number of iterations used for initialization and refinement are all 5000. For AltGD, we set $p = 0.6$. It is seen that the image recovered from AltGD is much more clear than those recovered by HIO and GS.



(a) Original (b) AltGD

(c) GS (d) HIO

Fig. 13. Retrieving Cameraman image in GMM noise with 2D Fourier measurements.

V. Conclusion

In this paper, we considered phase retrieval in the presence of grossly corrupted data – i.e., outliers. We formulated this problem as an ℓ_p fitting problem, where $0 < \ell_p < 2$, and provided an algorithmic framework that is based on two-block inexact alternating optimization. Two algorithms, namely, AltIRLS and

AltGD, were proposed under this framework. Although the algorithms cannot be analyzed using standard convergence results for alternating optimization due to a nonconvex constraint, we managed to show that the algorithms converge to a KKT point. The tools used for convergence analysis can also be used for analyzing convergence of other types of algorithms that involve non-convex constraints and inexact alternating optimization. Pertinent CRBs were derived for the noisy measurement models considered. Simulations showed that the proposed algorithms are promising in dealing with outliers in the context of phase retrieval using both random measurements and 2D Fourier measurements, for simulated and real image data.

Appendix A
Proof of Proposition 2.1

Let us denote

$$f(\mathbf{x}, \mathbf{u}) = \sum_{m=1}^{M} \left(|y_m u_m - \mathbf{a}_m^H \mathbf{x}|^2 + \epsilon \right)^{p/2}.$$

To simplify the analysis, let us define

$$\begin{aligned} \tilde{\mathbf{u}}_m &= [\mathrm{Re}\{u_m\}, \mathrm{Im}\{u_m\}]^T, \\ \tilde{\mathbf{x}} &= [\mathrm{Re}\{\mathbf{x}^T\}, \mathrm{Im}\{\mathbf{x}^T\}]^T, \\ \tilde{\mathbf{A}}_m &= \begin{bmatrix} \mathrm{Re}\{\boldsymbol{a}_m^H\}, & -\mathrm{Im}\{\boldsymbol{a}_m^H\} \\ \mathrm{Im}\{\boldsymbol{a}_m^H\}, & \mathrm{Re}\{\boldsymbol{a}_m^H\} \end{bmatrix}. \end{aligned}$$

Therefore, our problem can be written as

$$\begin{aligned} \min_{\{\tilde{\mathbf{u}}_m\}, \tilde{\mathbf{x}}} \; & f(\tilde{\mathbf{x}}, \{\tilde{\mathbf{u}}_m\}) = \sum_{m=1}^{M} \left(\|y_m \tilde{\mathbf{u}}_m - \tilde{\mathbf{A}}_m \tilde{\mathbf{x}}\|_2^2 + \epsilon \right)^{p/2} \\ \text{s.t.} \; & \|\tilde{\mathbf{u}}_m\|_2^2 = 1, \; \forall m. \end{aligned} \tag{38}$$

Note that in the new expression (38), all the variables are real-valued. Accordingly, we may define

$$g(\tilde{\mathbf{x}}, \{\tilde{\mathbf{u}}_m^{(r)}\}) = \sum_{m=1}^{M} \left(w_m^{(r)} \|y_m \tilde{\mathbf{u}}_m^{(r)} - \tilde{\mathbf{A}}_m \tilde{\mathbf{x}}\|_2^2 + \phi(w_m^{(r)}) \right). \tag{39}$$

Now our algorithm can be re-expressed as

$$\tilde{\mathbf{x}}^{(r+1)} = \arg\min_{\tilde{\boldsymbol{x}}} \; g(\tilde{\mathbf{x}}, \{\tilde{\mathbf{u}}_m^{(r)}\}) \tag{40}$$

$$\tilde{\mathbf{u}}_m^{(r+1)} = \arg \min_{\|\tilde{\mathbf{u}}_m\|_2^2 = 1, \forall m} \; f(\tilde{\mathbf{x}}^{(r+1)}, \{\tilde{\mathbf{u}}_m\}) \tag{41}$$

Let us assume the boundedness of $\tilde{\mathbf{x}}^{(r)}$ and $\tilde{\mathbf{u}}^{(r)}$ and $w_m^{(r)}$, which will be shown later. Under this assumption, the gradients of $g(\tilde{\mathbf{x}}, \{\tilde{\mathbf{u}}_m^{(r)}\})$ and $f(\tilde{\mathbf{x}}, \{\tilde{\mathbf{u}}_m\})$ exist. According to Lemma 2.1, it is easily seen that

$$f(\tilde{\mathbf{x}}, \{\tilde{\mathbf{u}}_m^{(r)}\}) \le g(\tilde{\mathbf{x}}, \{\tilde{\mathbf{u}}_m^{(r)}\}) \quad \forall \tilde{\mathbf{x}}, \tag{42a}$$

$$f(\tilde{\mathbf{x}}^{(r)}, \{\tilde{\mathbf{u}}_m^{(r)}\}) = g(\tilde{\mathbf{x}}^{(r)}, \{\tilde{\mathbf{u}}_m^{(r)}\}) \tag{42b}$$

$$\nabla_{\tilde{\mathbf{x}}} f(\tilde{\mathbf{x}}^{(r)}, \{\tilde{\mathbf{u}}_m^{(r)}\}) = \nabla_{\tilde{\mathbf{x}}} g(\tilde{\mathbf{x}}^{(r)}, \{\tilde{\mathbf{u}}_m^{(r)}\}). \tag{42c}$$

Now, we have

$$f(\tilde{\mathbf{x}}^{(r)}, \{\tilde{\mathbf{u}}_m^{(r)}\}) = g(\tilde{\mathbf{x}}^{(r)}, \{\tilde{\mathbf{u}}_m^{(r)}\}) \tag{43a}$$

$$\ge g(\tilde{\mathbf{x}}^{(r+1)}, \{\tilde{\mathbf{u}}_m^{(r)}\}) \tag{43b}$$

$$\ge f(\tilde{\mathbf{x}}^{(r+1)}, \{\tilde{\mathbf{u}}_m^{(r)}\}) \tag{43c}$$

$$\ge f(\tilde{\mathbf{x}}^{(r+1)}, \{\tilde{\mathbf{u}}_m^{(r+1)}\}) \tag{43d}$$

where (43a) follows (42b), (43b) is obtained because of (40), (43c) holds since we have the property in (42a), and (43b) is obtained by the fact that the subproblem w.r.t. $\mathbf{u}$ is optimally solved, i.e., (41).

Assume that $\{r_j\}$ denotes the index set of a convergent subsequence, and that $\{\tilde{\mathbf{x}}^{(r_j)}, \{\tilde{\mathbf{u}}_m^{(r_j)}\}\}$ converges to $(\tilde{\mathbf{x}}^\star, \{\tilde{\mathbf{u}}_m^\star\})$ Then, we have

$$g(\tilde{\mathbf{x}}, \{\tilde{\mathbf{u}}_m^{(r_j)}\}) \ge g(\tilde{\mathbf{x}}^{(r_j+1)}, \{\tilde{\mathbf{u}}_m^{(r_j)}\}) \tag{44a}$$

$$\ge f(\tilde{\mathbf{x}}^{(r_j+1)}, \{\tilde{\mathbf{u}}_m^{(r_j)}\}) \tag{44b}$$

$$\ge f(\tilde{\mathbf{x}}^{(r_j+1)}, \{\tilde{\mathbf{u}}_m^{(r_j+1)}\}) \tag{44c}$$

$$\ge f(\tilde{\mathbf{x}}^{(r_{j+1})}, \{\tilde{\mathbf{u}}_m^{(r_{j+1})}\}) \tag{44d}$$

$$= g(\tilde{\mathbf{x}}^{(r_{j+1})}, \{\tilde{\mathbf{u}}_m^{(r_{j+1})}\}), \tag{44e}$$

where (44d) is obtained by the fact that $r_{j+1} \ge r_j + 1$ since r_j indexes a subsequence. Taking $j \to \infty$, and by the boundedness of $\mathbf{W}^{(r)}$ and continuity of $g(\cdot)$, we see that

$$g(\tilde{\mathbf{x}}, \{\tilde{\mathbf{u}}_m^\star\}) \ge g(\tilde{\mathbf{x}}^\star, \mathbf{u}^\star). \tag{45}$$

The inequality in (45) means that $\tilde{\mathbf{x}}^\star$ is blockwise minimizer of $g(\tilde{\mathbf{x}}, \{\tilde{\mathbf{u}}_m^\star\})$. Therefore, it satisfies the partial KKT condition w.r.t. $\tilde{\mathbf{x}}$, i.e.,

$$\nabla_{\tilde{\mathbf{x}}} g(\tilde{\mathbf{x}}^\star, \{\tilde{\mathbf{u}}_m^\star\}) = \mathbf{0}. \tag{46}$$

By (42c), we immediately have

$$\nabla_{\tilde{\mathbf{x}}} f(\tilde{\mathbf{x}}^\star, \{\tilde{\mathbf{u}}_m^\star\}) = \mathbf{0}. \tag{47}$$

Similarly, by the update rule in (41), we have

$$f(\tilde{\mathbf{x}}^{(r_j)}, \mathbf{u}) \ge f(\tilde{\mathbf{x}}^{(r_j)}, \{\tilde{\mathbf{u}}_m^{(r_j)}\}) \tag{48}$$

and thus

$$f(\tilde{\mathbf{x}}^\star, \{\mathbf{u}_m\}) \ge f(\tilde{\mathbf{x}}^\star, \{\mathbf{u}_m^\star\}). \tag{49}$$

Therefore, $\tilde{\mathbf{u}}_m^\star$ also satisfies the partial KKT condition

$$\nabla_{\tilde{\mathbf{u}}_m} f(\tilde{\mathbf{x}}^\star, \{\tilde{\mathbf{u}}_m^\star\}) + \boldsymbol{\lambda}_m^\star \odot \tilde{\mathbf{u}}_m^\star = \mathbf{0}, \; \forall m \tag{50}$$

where $\boldsymbol{\lambda}_m \in \mathbb{R}^2$ are dual variables. Combining (47) and (50), it follows that every limit point of the solution sequence is a KKT point.

Now, we rigorously show the boundedness of $\tilde{\mathbf{x}}^{(r)}$ and $\tilde{\mathbf{u}}^{(r)}$ and $\mathbf{W}^{(r)}$. Let us first show that $\mathbf{W}^{(r)}$ is bounded given finite $\tilde{\mathbf{x}}^{(r)}$ and $\tilde{\mathbf{u}}^{(r)}$. This is relatively easy, since $\mathbf{y}$ and $\mathbf{A}$ are always bounded and thus $\mathbf{W}^{(r)}$ is also bounded when $0 < p < 2$ and $\epsilon > 0$ by definition (cf. Eq. (8)). Now, we show that $\tilde{\mathbf{x}}^{(r)}$ and $\tilde{\mathbf{u}}^{(r)}$ are bounded. Since the algorithm is essentially majorization minimization, there is no risk for $\tilde{\mathbf{x}}^{(r)}$ being unbounded, if the initialization is bounded, $\mathbf{A}$ has full column rank, and $\mathbf{y}$ is bounded. To explain, one can see that

$$\begin{aligned}\|\mathbf{y}\odot\tilde{\mathbf{u}}^{(r)}-\tilde{\mathbf{A}}\tilde{\mathbf{x}}^{(0)}\|_p^p &\geq \|\mathbf{y}\odot\tilde{\mathbf{u}}^{(r)}-\tilde{\mathbf{A}}\tilde{\mathbf{x}}^{(r)}\|_p^p \\ &\geq \left|\|\mathbf{y}\|_p^p-\|\tilde{\mathbf{A}}\tilde{\mathbf{x}}^{(r)}\|_p^p\right| \\ &\geq \|\tilde{\mathbf{A}}\tilde{\mathbf{x}}^{(r)}\|_p^p-\|\mathbf{y}\|_p^p. \qquad (51)\end{aligned}$$

The first inequality holds because block upper bound minimization algorithm always decreases the objective [40]. The second inequality is due to the triangle inequality for $\|\cdot\|_p^p$ when $0<p<1$ [39]; for $p>1$, one can apply the same argument to $\|\cdot\|_p$. From the above, one can see that

$$\|\tilde{\mathbf{A}}\tilde{\mathbf{x}}^{(r)}\|_p^p \leq \infty.$$

Therefore, if $\mathbf{A}$ does not have a null space, then $\tilde{\mathbf{x}}^{(r)}$ has to be bounded. Consequently, $\mathbf{W}^{(r)}$ is always bounded from above.

The next step is to show that the whole sequence converges to $\mathcal{K}$, the set of all the KKT points. This is relatively straightforward. Let us assume that the whole sequence does not converge to $\mathcal{K}$. This means that there exists a subsequence which does not converge to $\mathcal{K}$. The boundedness of $\tilde{\mathbf{x}}^{(r)},\tilde{\mathbf{u}}^{(r)}$ implies that every subsequence of $\{\tilde{\mathbf{x}}^{(r)},\tilde{\mathbf{u}}^{(r)}\}$ has a limit point. We have just shown that every limit point is a KKT point, and thus this is a contradiction. Therefore, the whole sequence has to converge to $\mathcal{K}$.

Appendix B
CRB for Laplacian Noise

The likelihood function for Laplacian noise is given by [36], [37]

$$p(\mathbf{y};\mathbf{x})=\prod_{i=1}^{M}\frac{1}{\sqrt{2}\sigma_n}\exp\left\{-\frac{\sqrt{2}}{\sigma_n}\left|y_i-|\mathbf{a}_i^H\mathbf{x}|\right|\right\} \qquad (52)$$

where the noise variance is σ_n^2, and its log-likelihood function is

$$\ln p(\mathbf{y};\mathbf{x})=-M\ln(\sqrt{2}\sigma_n)-\frac{\sqrt{2}}{\sigma_n}\sum_{i=1}^{M}\left|y_i-|\mathbf{a}_i^H\mathbf{x}|\right|. \qquad (53)$$

The vector of unknown parameters for complex-valued $\mathbf{x}$ is

$$\boldsymbol{\beta}=[\mathrm{Re}\{x_1\}\ \cdots\ \mathrm{Re}\{x_N\},\ \mathrm{Im}\{x_1\}\ \cdots\ \mathrm{Im}\{x_N\}]^T. \qquad (54)$$

It is worth noting that σ_n^2 is actually an unknown parameter which should be considered as a part of $\boldsymbol{\beta}$. However, since σ_n^2 is uncorrelated with the real and imaginary parts of x_i, their mutual Fisher information is zero. It will not impact the final CRB formula for $\mathbf{x}$. For this reason, we do not include σ_n^2 in $\boldsymbol{\beta}$. Thus, the FIM can be partitioned into four parts, i.e.,

$$\mathbf{F}_{L,c}=\begin{bmatrix}\mathbf{F}_{L,rr} & \mathbf{F}_{L,ri}\\ \mathbf{F}_{L,ir} & \mathbf{F}_{L,ii}\end{bmatrix} \qquad (55)$$

where

$$[\mathbf{F}_{L,c}]_{m,n}=\mathbb{E}\left[\frac{\partial\ln p(\mathbf{y};\mathbf{x})}{\partial\boldsymbol{\beta}_m}\frac{\partial\ln p(\mathbf{y};\mathbf{x})}{\partial\boldsymbol{\beta}_n}\right]. \qquad (56)$$

The partial derivative of $\ln p(\mathbf{y};\mathbf{x})$ with respective to $\boldsymbol{\beta}_m$ is

$$\begin{aligned}\frac{\partial\ln p(\mathbf{y};\mathbf{x})}{\partial\boldsymbol{\beta}_m} &= -\frac{\sqrt{2}}{\sigma_n}\sum_{i=1}^{M}\frac{\partial\left|y_i-|\mathbf{a}_i^H\mathbf{x}|\right|}{\partial\boldsymbol{\beta}_m}\\ &= \frac{\sqrt{2}}{\sigma_n}\sum_{i=1}^{M}\frac{y_i-|\mathbf{a}_i^H\mathbf{x}|}{\left|y_i-|\mathbf{a}_i^H\mathbf{x}|\right|}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_m}\\ &= \frac{\sqrt{2}}{\sigma_n}\sum_{i=1}^{M}\mathrm{sgn}(y_i-|\mathbf{a}_i^H\mathbf{x}|)\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_m} \qquad (57)\end{aligned}$$

where

$$\mathrm{sgn}(a)=\begin{cases}1, & a>0\\ -1\,, & a<0\end{cases} \qquad (58)$$

and

$$\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_m}=\begin{cases}\frac{[\mathrm{Re}\{\mathbf{a}_i\mathbf{a}_i^H\mathbf{x}\}]_m}{|\mathbf{a}_i^H\mathbf{x}|}, & \text{for } \boldsymbol{\beta}_m=\mathrm{Re}\{x_m\}\\ \frac{[\mathrm{Im}\{\mathbf{a}_i\mathbf{a}_i^H\mathbf{x}\}]_m}{|\mathbf{a}_i^H\mathbf{x}|}, & \text{for } \boldsymbol{\beta}_m=\mathrm{Im}\{x_m\}.\end{cases} \qquad (59)$$

Substituting (57) into (56), we have

$$\begin{aligned}[\mathbf{F}_{L,c}]_{m,n} &= \frac{2}{\sigma_n^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_m}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_n}\\ &\quad\times\mathbb{E}\left[\mathrm{sgn}(y_i-|\mathbf{a}_i^H\mathbf{x}|)\cdot\mathrm{sgn}(y_j-|\mathbf{a}_j^H\mathbf{x}|)\right]. \qquad (60)\end{aligned}$$

Next we compute the value of $\mathbb{E}[\mathrm{sgn}(y_i-|\mathbf{a}_i^H\mathbf{x}|)\mathrm{sgn}(y_j-|\mathbf{a}_j^H\mathbf{x}|)]$. For notational simplicity, let $s_i=\mathrm{sgn}(y_i-|\mathbf{a}_i^H\mathbf{x}|)$. It is obvious that when $i=j$, we have

$$\mathbb{E}\left[s_is_j\right]=1. \qquad (61)$$

For $i\neq j$, we first write

$$\begin{aligned}\mathbb{E}\left[s_is_j\right] &= \Pr(s_i=s_j)\times(+1)+\Pr(s_i\neq s_j)\times(-1)\\ &= 2\Pr(s_i=s_j)-1 \qquad (62)\end{aligned}$$

where $\Pr$ standards for the probability. Then the value of $\Pr(s_i=s_j)$ is computed as

$$\begin{aligned}\Pr(s_i=s_j) &= \Pr(s_i=1|s_j=1)\Pr(s_j=1)\\ &\quad+\Pr(s_i=-1|s_j=-1)\Pr(s_j=-1)\\ &= \Pr(s_i=1)\Pr(s_j=1)\\ &\quad+\Pr(s_i=-1)\Pr(s_j=-1)\\ &= 0.5 \qquad (63)\end{aligned}$$

where $\Pr(s_i=1)=\Pr(s_i=-1)=0.5$ and the second equation follows by independence of s_i and s_j when $i\neq j$. Substituting (63) into (62) and using (61) yields

$$\mathbb{E}\left[s_is_j\right]=\begin{cases}1, & i=j\\ 0, & i\neq j.\end{cases} \qquad (64)$$

Substituting (64) into (60), we obtain

$$[\mathbf{F}_{L,c}]_{m,n}=\frac{2}{\sigma_n^2}\sum_{i=1}^{M}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_m}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_n}. \qquad (65)$$

Now using (59) and (65), the four sub-FIMs can be easily derived as

$$\mathbf{F}_{L,rr} = \frac{2}{\sigma_n^2}\mathrm{Re}\{\mathbf{A}^H\mathrm{diag}(\mathbf{Ax})\}\cdot\mathrm{diag}(|\mathbf{Ax}|^{-2}) \times \mathrm{Re}\{\mathbf{A}^H\mathrm{diag}(\mathbf{Ax})\}^T \tag{66}$$

$$\mathbf{F}_{L,ii} = \frac{2}{\sigma_n^2}\mathrm{Im}\{\mathbf{A}^H\mathrm{diag}(\mathbf{Ax})\}\cdot\mathrm{diag}(|\mathbf{Ax}|^{-2}) \times \mathrm{Im}\{\mathbf{A}^H\mathrm{diag}(\mathbf{Ax})\}^T \tag{67}$$

$$\mathbf{F}_{L,ri} = \frac{2}{\sigma_n^2}\mathrm{Re}\{\mathbf{A}^H\mathrm{diag}(\mathbf{Ax})\}\cdot\mathrm{diag}(|\mathbf{Ax}|^{-2}) \times \mathrm{Im}\{\mathbf{A}^H\mathrm{diag}(\mathbf{Ax})\}^T \tag{68}$$

$$\mathbf{F}_{L,ir} = \mathbf{F}_{L,ri}^T. \tag{69}$$

Combining (66)–(69), we obtain

$$\mathbf{F}_{L,c} = \frac{2}{\sigma_n^2}\mathbf{G}_{L,c}\,\mathrm{diag}(|\mathbf{Ax}|^{-2})\,\mathbf{G}_{L,c}^T \tag{70}$$

where

$$\mathbf{G}_{L,c} = \begin{bmatrix}\mathrm{Re}\{\mathbf{A}^H\mathrm{diag}(\mathbf{Ax})\}\\ \mathrm{Im}\{\mathbf{A}^H\mathrm{diag}(\mathbf{Ax})\}\end{bmatrix}. \tag{71}$$

This completes the proof of Proposition 3.1.

Appendix C
Proof of Rank Property of $\mathbf{F}_{L,c}$ and $\mathbf{F}_{L,r}$

We note here that $\mathbf{F}_{L,c}$ is derived under the assumption of nonzero $\mathbf{a}_m^H\mathbf{x}$. Therefore, $\mathrm{diag}(|\mathbf{Ax}|^{-2})$ is full rank. As a result, computing the rank of $\mathbf{F}_{L,c}$ is equivalent to computing the rank of $\mathbf{G}_{L,c}$. To this end, define a nonzero vector $\mathbf{v} = [\,\mathbf{v}_1^T\ \mathbf{v}_2^T\,]^T \in \mathbf{R}^{2N}$, which leads to

$$\mathbf{G}_{L,c}^T\mathbf{v} = \mathrm{Re}\{\mathbf{A}^H\mathrm{diag}(\mathbf{Ax})\}^T\mathbf{v}_1 + \mathrm{Im}\{\mathbf{A}^H\mathrm{diag}(\mathbf{Ax})\}^T\mathbf{v}_2. \tag{72}$$

Now let $\mathbf{u} = \mathbf{v}_1 + j\mathbf{v}_2$, then

$$\mathbf{G}_{L,c}^T\mathbf{v} = \mathrm{Re}\left\{\left(\mathbf{A}^H\mathrm{diag}(\mathbf{Ax})\right)^H\mathbf{u}\right\} = \mathrm{Re}\left\{(\mathbf{Ax})^*\odot(\mathbf{Au})\right\} \tag{73}$$

which equals to zero if and only if

$$\mathbf{u} = j\mathbf{x} \tag{74}$$

i.e.,

$$\mathbf{G}_{L,c}^T\mathbf{v} = \mathrm{Re}\left\{j\,|\mathbf{Ax}|^2\right\} = \mathbf{0}. \tag{75}$$

This means that there is only one direction $\mathbf{v} = [-\mathrm{Im}\{\mathbf{x}\}^T\ \mathrm{Re}\{\mathbf{x}\}^T]^T$, which is non-zero, lies in the null space of $\mathbf{G}_{L,c}$, thus also in the null space of $\mathbf{F}_{L,c}$.

In the real $\mathbf{x}$ case, similar to the proof of $\mathbf{F}_{L,c}$, it suffices to show that there is no nonzero vector $\mathbf{v} \in \mathbb{R}^N$ making $\mathbf{F}_{L,r}\mathbf{v} = \mathbf{0}$. It is easy to see that

$$\mathbf{G}_{L,r}^T\mathbf{v} = \mathrm{Re}\left\{(\mathbf{Ax})^*\odot(\mathbf{Av})\right\}. \tag{76}$$

Since $\mathbf{v}$ is real-valued, it cannot be set to $\mathbf{v} = j\mathbf{x}$ to make (76) zero. Given a nontrivial $\mathbf{A}$, it is impossible to find a $\mathbf{v}$ such that $\mathbf{Av} = \mathbf{0}$ except $\mathbf{v} = \mathbf{0}$. Therefore, $\mathbf{F}_{L,r}$ is full rank. This completes the proof.

References

[1] C. Qian, X. Fu, N. D. Sidiropoulos, and L. Huang, "Inexact alternating optimization for phase retrieval with outliers," in *Proc. 24th Eur. Signal Process. Conf.*, Budapest, Hungary, pp. 1538–1542, 2016.

[2] R. Gerchberg and W. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, pp. 237–246, 1972.

[3] J. R. Fienup, "Phase retrieval algorithms: A comparison," *Appl. Opt.*, vol. 21, no. 15, pp. 2758–2769, 1982.

[4] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4814–4826, Sep. 2015.

[5] E. J. Candès, T. Strohmer, and V. Voroninski, "PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming," *Commun. Pure Appl. Math.*, vol. 66, no. 8, pp. 1241–1274, 2013.

[6] P. Hand, "PhaseLift is robust to a constant fraction of arbitrary errors," *Appl. Comput. Harmonic Anal.*, vol. 42, no. 3, pp. 550–562, 2017.

[7] E. J. Candès, X. Li, and M. Soltanolkotabi, "Phase retrieval via Wirtinger flow: Theory and algorithms," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1985–2007, Apr. 2015.

[8] Y. Chen and E. Candès, "Solving random quadratic systems of equations is nearly as easy as solving linear systems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 739–747.

[9] G. Wang and G. Giannakis, "Solving random systems of quadratic equations via truncated generalized gradient flow," in *Advances in Neural Inf. Process. Syst.*, pp. 568–576, 2016.

[10] G. Wang, G. B. Giannakis, and J. Chen, "Scalable solvers of random quadratic equations via stochastic truncated amplitude flow," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1961–1974, Apr. 2017.

[11] C. Qian, N. D. Sidiropoulos, K. Huang, L. Huang, and H. C. So, "Phase retrieval using feasible point pursuit: Algorithms and Cramér–Rao bound," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5282–5296, Oct. 2016.

[12] C. Qian, N. D. Sidiropoulos, K. Huang, L. Huang, and H. C. So, "Least squares phase retrieval using feasible point pursuit," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, pp. 4288–4292, 2016.

[13] E. J. Candès, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM Rev.*, vol. 57, no. 2 pp. 225–251, 2015.

[14] J. Sigl, "Nonlinear residual minimization by iteratively reweighted least squares," *Comput. Optim. Appl.*, vol. 64, no. 3, pp. 755–792, 2016.

[15] D. S. Weller, A. Pnueli, G. Divon, O. Radzyner, Y. C. Eldar, and J. A. Fessler, "Undersampled phase retrieval with outliers," *IEEE Trans. Comput. Imag.*, vol. 1, no. 4, pp. 247–258, Dec. 2015.

[16] S. Chen and H. Lu, "Noise characteristic and its removal in digital radiographic system," in *Proc. 15th World Conf. Nondestruct. Test.*, Roma 2000.

[17] I. Frosio and N. A. Borghese, "Statistical based impulsive noise removal in digital radiography," *IEEE Trans. Med. Imag.*, vol. 28, no. 1, pp. 3–16, Jan. 2009.

[18] H. Zhang, Y. Chi, and Y. Liang, "Provable non-convex phase retrieval with outliers: Median truncated wirtinger flow," in *Proc. Int. Conf. Mach. Learn.*, New York, NY, USA, pp. 1022–1031, 2016.

[19] J. N. Cederquist and C. C. Wackerman, "Phase-retrieval error: A lower bound," *J. Opt. Soc. Amer. A*, vol. 4, no. 9 pp. 1788–1792, 1987.

[20] R. Balan, "The Fisher information matrix and the CRLB in a non-AWGN model for the phase retrieval problem," in *Proc. Int. Conf. Sampling Theory Appl.*, Washington, DC, USA, pp. 178–182, 2015.

[21] R. Balan, "Reconstruction of signals from magnitudes of redundant representations: The complex case," *Found. of Comput. Math.*, vol. 16, no. 3, pp. 677–721, 2016.

[22] A. S. Bandeira, J. Cahill, D. G. Mixon, and A. A. Nelson, "Saving phase: Injectivity and stability for phase retrieval," *Appl. Comput. Harmonic Anal.*, vol. 37, no. 1, pp. 106–125, 2014.

[23] X. Fu, K. Huang, B. Yang, W.-K. Ma, and N. D. Sidiropoulos, "Robust volume minimization-based matrix factorization for remote sensing and document clustering," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6254–6268, Dec. 2016.

[24] Y. Liu, Y. Dai, and S. Ma, "Joint power and admission control: Non-convex ℓ_q approximation and an effective polynomial time deflation approach," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3641–3656, Jul. 2015.

[25] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Las Vegas, NV, USA, pp. 3869–3872, 2008.

[26] X. Fu, K. Huang, W.-K. Ma, N. D. Sidiropoulos, and R. Bro, "Joint tensor factorization and outlying slab suppression with applications," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6315–6328, Dec. 2015.

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

[28] Y. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM J. Optim.*, vol. 10, no. 1, pp. 177–182, 1999.

[29] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence O $(1/k^2)$," *Doklady an SSSR*, vol. 269, no. 3, pp. 543–547, 1983.

[30] O. Fercoq and P. Richtàrik, "Accelerated, parallel, and proximal coordinate descent," *SIAM J. Optim.*, vol. 25, no. 4, pp. 1997–2023, 2015.

[31] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM J. Optim.*, vol. 22, no. 2 pp. 341–362, 2012.

[32] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," *Inverse Probl. Imag.*, vol. 9, no. 2, pp. 601–624, 2015.

[33] P. Stoica and T. L. Marzetta, "Parameter estimation problems with singular information matrices," *IEEE Trans. Signal Process.*, vol. 49, no. 1, pp. 87–90, Jan. 2001.

[34] A. O. Hero, III, J. A. Fessler, and M. Usman, "Exploring estimator bias-variance tradeoffs using the uniform CR bound," *IEEE Trans. Signal Process.*, vol. 44, no. 8, pp. 2026–2041, Aug. 1996.

[35] K. Huang and N. D. Sidiropoulos, "Putting nonnegative matrix factorization to the test: A tutorial derivation of pertinent Cramér–Rao bounds and performance benchmarking," *IEEE Signal Process. Mag., Special Issue Source Separation Appl.*, vol. 31, no. 3, pp. 76–86, May 2014.

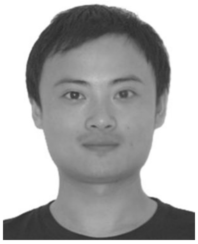
[36] S. A. Vorobyov, Y. Rong, N. D. Sidiropoulos, and A. B. Gershman, "Robust iterative fitting of multilinear models," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2678–2689, Aug. 2005.

[37] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, 1998.

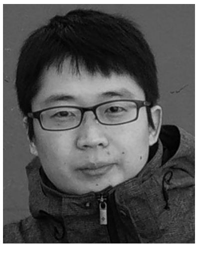
[38] S. Basu and Y. Bresler, "The stability of nonlinear least squares problems and the Cramér–Rao bound," *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3426–3436, Dec. 2000.

[39] R. Chartrand and V. Staneva, "Restricted isometry properties and nonconvex compressive sensing," *Inverse Probl.*, vol. 24, no. 3, 035020, 2008.

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



Cheng Qian was born in Deqing, Zhejiang, China. He received the B.E. degree in communication engineering from Hangzhou Dianzi University, Hangzhou, China, in 2011, the M.S. and the Ph.D. degrees in information and communication engineering from Harbin Institute of Technology, Harbin, China, in 2013 and 2017, respectively. During 2014–2016, he was a visiting Ph.D. student at the Signal and Tensor Analytics Research group, University of Minnesota, Minneapolis, MN, USA, where he is currently a Postdoctoral Associate. His research interests include statistical signal processing, optimization, and wireless communications.



Xiao Fu (S'12–M'15) received the B.Eng. and M.Eng. degrees in communications and information engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2005 and 2010, respectively. In 2014, he received the Ph.D. degree in electronic engineering from The Chinese University of Hong Kong, Shatin, Hong Kong.

From 2005 to 2006, he was an Assistant Engineer at China Telecom Co. Ltd., Shenzhen, China. He is currently a Postdoctoral Associate at the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. His research interests include signal processing and machine learning, with a recent emphasis on factor analysis and its applications. He received a Best Student Paper Award at IEEE ICASSP 2014, and was a finalist of the Best Student Paper Competition at IEEE SAM 2014. He also co-authored a paper that received a Best Student Paper Award at IEEE CAMSAP 2015.

Nicholas D. Sidiropoulos (F'09) received the Diploma in electrical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, and the M.S. and the Ph.D. degrees in electrical engineering from the University of Maryland-College Park, College Park, MD, USA, in 1988, 1990, and 1992, respectively.

He served as an Assistant Professor at the University of Virginia (1997–1999); an Associate Professor at the University of Minnesota-Minneapolis (2000–2002); a Professor at the Technical University of Crete, Greece (2002–2011); and a Professor at the University of Minnesota, Minneapolis, MN, USA (2011-), where he currently holds an ADC Chair in digital technology. His current research focuses primarily on signal and tensor analytics, with applications in cognitive radio, big data, and preference measurement. He received the NSF/CAREER award (1998), the IEEE Signal Processing Society (SPS) Best Paper Award (2001, 2007, 2011), and the IEEE SPS Meritorious Service Award (2010). He has served as IEEE SPS Distinguished Lecturer (2008–2009), and the Chair of the IEEE Signal Processing for Communications and Networking Technical Committee (2007–2008). He received the Distinguished Alumni Award of the Department of Electrical and Computer Engineering, University of Maryland, College Park (2013). He was elected Fellow of EURASIP in 2014.

Lei Huang (M'07–SM'14) was born in Guangdong, China. He received the B.Sc., M.Sc., and Ph.D. degrees in electronic engineering from Xidian University, Xian, China, in 2000, 2003, and 2005, respectively. From 2005 to 2006, he was a Research Associate with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. From 2009 to 2010, he was a Research Fellow with the Department of Electronic Engineering, City University of Hong Kong, and a Research Associate with the Department of Electronic Engineering, The Chinese University of Hong Kong. From 2011 to 2014, he was a Professor with the Department of Electronic and Information Engineering, Harbin Institute of Technology Shenzhen Graduate School. Since 2014, he has been with the College of Information Engineering, Shenzhen University, Shenzhen, China, where he is currently a Distinguished Professor and the Director of the Shenzhen Key Lab of Advanced Navigation Technology. His research interests include array signal processing, statistical signal processing, sparse signal processing, and their applications in radar, navigation, and wireless communications. He is currently serving as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and Digital Signal Processing. Additionally, he is an elected member of the Sensor Array and Multichannel Technical Committee of the IEEE Signal Processing Society.

Junhao Xie (M'02–SM'15) received the B.S. degree in electronic engineering from Harbin Institute of Shipbuilding Engineering, Harbin, China, the M.S. degree in signal and information processing from Harbin Engineering University, Harbin, China, and the Ph.D. degree in communication and information system from Harbin Institute of Technology, Harbin, China, in 1992, 1995, and 2001, respectively.

From 2004 to 2006, he was a visiting scholar at Curtin University of Technology, Perth, WA, Australia, where he dealt with radar remote sensing signal processing. He is a Professor in the Department of Electronic Engineering, Harbin Institute of Technology. His past and present research interests include radar system analysis and modeling, array signal processing, target detection and estimation, and ocean remote sensing for shipborne high-frequency surface wave radar, and inverse synthetic aperture radar.