SIGIBE: Solving Random Bilinear Equations via Gradient Descent with Spectral Initialization

Antonio G. Marques[†], Gonzalo Mateos[‡], and Yonina C. Eldar[*]
[†] King Juan Carlos University, Dept. of Signal Theory and Comms., Madrid, Spain
[‡] University of Rochester, Dept. of Electrical and Computer Eng., Rochester, NY, USA
[*] Technion, Israel Institute of Technology, Dept. of Electrical Engineering, Haifa, Israel

***Abstract*—We investigate the problem of finding the real-valued vectors $\mathbf{h}$, of size L, and $\mathbf{x}$, of size P, from M independent measurements $y_m = \langle \mathbf{a}_m, \mathbf{h}\rangle\langle \mathbf{b}_m, \mathbf{x}\rangle$, where $\mathbf{a}_m$ and $\mathbf{b}_m$ are known random vectors. Recovery of the unknowns entails solving a set of *bilinear* equations, a challenging problem encountered in signal processing tasks such as blind deconvolution for channel equalization or image deblurring. Inspired by the Wirtinger flow approach to the related phase retrieval problem, we propose a solver that proceeds in two steps: (i) first a spectral method is used to obtain an initial guess; which is then (ii) refined using simple and scalable gradient descent iterations to minimize a natural non-convex formulation of the recovery problem. Our method – which we refer to as SIGIBE: Spectral Initialization and Gradient Iterations for Bilinear Equations – can accommodate arbitrary correlations between $\mathbf{a}_m$ and $\mathbf{b}_m$. Different from recent approaches to blind deconvolution using convex relaxation, SIGIBE does not require matrix lifting that could hinder the method's scalability. Numerical tests corroborate SIGIBE's effectiveness in various data settings, and show successful recovery with as few as $M \gtrsim (L+P)$ measurements.**

***Index Terms*—Bilinear equations, blind deconvolution, non-convex optimization, spectral initialization, correlated data.**

I. Introduction

Suppose we are given a collection of M scalar measurements $y_m \in \mathbb{R}$ of the form

$$y_m = \langle \mathbf{a}_m, \mathbf{h}\rangle\langle \mathbf{b}_m, \mathbf{x}\rangle = \mathbf{a}_m^T\mathbf{h} \cdot \mathbf{b}_m^T\mathbf{x}, \quad m = 1, \ldots, M \quad (1)$$

where $\mathbf{h} \in \mathbb{R}^L$ and $\mathbf{x} \in \mathbb{R}^P$ are unknown. The random vectors $\{\mathbf{a}_m\}_{m=1}^M$ and $\{\mathbf{b}_m\}_{m=1}^M$ are given and assumed to be zero-mean i.i.d., with identity covariance matrix. We allow for arbitrary correlation between $\mathbf{a}_m$ and $\mathbf{b}_m$; in particular, we may have $\mathbf{a}_m = \mathbf{b}_m$. Assuming that $M \geq (L+P)$, our goal is to recover $\mathbf{h}$ and $\mathbf{x}$, up to an inherent scaling ambiguity.

Solving a (random) system of bilinear equations as in (1) is a challenging problem typically encountered in signal processing tasks such as blind deconvolution (e.g., in communication channel equalization and for image deblurring [1], [2]), array self-calibration for direction-of-arrival estimation [3], and modeling of network diffusion processes [4], just to name a few applications.

Relation to prior work. While measurements y_m in (1) are bilinear functions of $\mathbf{x}$ and $\mathbf{h}$, they are linear in the entries of the rank-one matrix $\mathbf{x}\mathbf{h}^T$. Exploiting this insight, recent approaches have cast the blind deconvolution problem as one of rank minimization, proposing convex relaxation algorithms with performance guarantees [1]. These *matrix lifting* approaches rely on semidefinite programming (SDP) relaxation which, unlike the algorithms proposed here, do not scale well to large dimensions. Although we assume all vectors are real for simplicity, acquisition in the Fourier domain motivates extensions to the complex case [1], [3]. For complex-valued vectors and the symmetric setup whereby $\mathbf{x} = \mathbf{h}$ and $\mathbf{a}_m = \mathbf{b}_m$, the measurements in (1) take the quadratic form $y_m = |\langle \mathbf{a}_m, \mathbf{x}\rangle|^2$. Finding $\mathbf{x}$ is known as the phase retrieval problem, which has a long history in the physical sciences including astronomy, optics and microscopy [5]–[7]. Recent algorithms for phase retrieval include Phaselift [8] and similar SDP-based methods [9], [10], greedy algorithms such as GESPAR for sparse $\mathbf{x}$ [11], [12], and gradient approaches like Wirtinger flow [13], [14], which has also been extended to low-rank matrix recovery [15], [16]. We note that there is an inherent symmetry to the phase retrieval problem, which is not present in the general bilinear equations (1).

Contributions. Motivated by Wirtinger flow for phase retrieval [13], we propose a two-step algorithm to solve the bilinear equations (1) which we refer to as SIGIBE: Spectral Initialization and Gradient Iterations for Bilinear Equations. In the first step, a spectral method is used to obtain an initial guess. In the second step, the initialization is refined using simple and scalable gradient descent iterations to minimize a natural non-convex formulation of the problem (Section II). SIGIBE accommodates arbitrary correlations between $\mathbf{a}_m, \mathbf{b}_m$ (Section III) and, different from recent approaches to blind deconvolution using convex programming [1], [3], it does not require matrix lifting that could hinder the method's scalability. Numerical tests in Section IV corroborate the effectiveness of SIGIBE for various data settings, and show successful recovery with as few as $M \gtrsim (L+P)$ measurements. Theoretical recovery guarantees are beyond the scope of this algorithmic paper, but subject of ongoing investigation.

[†] The author's work was supported by the Spanish MINECO grant TEC2013-41604-R.
[‡] The authors' work was supported by NSF CCF-1217963.
[*] The author's work was funded by the EU's Horizon 2020 programme under grant agreement ERC-BNYQ, by the ISF under grant no. 335/14, and by ICore: the Israeli Excellence Center Circle of Light.

Notation: Operators $(\cdot)^T$ and $\mathbb{E}[\cdot]$ denote transposition and expectation, respectively; $|\cdot|$ is the magnitude of a scalar and $\|\cdot\|$ the ℓ_2-norm of a vector, $[\cdot]_{i,j}$ the (i,j)-th entry of a matrix and $[\cdot]_{1:n,1:n}$ the submatrix formed by selecting the first n rows and columns. The $n \times n$ identity matrix is represented by $\mathbf{I}_n$, while $\mathbf{0}_n$ stands for the $n \times 1$ vector of all zeros, and $\mathbf{0}_{n\times p} := \mathbf{0}_n\mathbf{0}_p^T$.

II. Problem Formulation and Algorithm

In order to recover $\mathbf{h}$ and $\mathbf{x}$ from the given measurements (1), a natural criterion is to minimize the residual sum of squares

$$\min_{\{\mathbf{x},\mathbf{h}\}} f(\mathbf{x},\mathbf{h}) := \frac{1}{2M}\sum_{m=1}^{M}\left(\mathbf{a}_m^T\mathbf{h}\cdot\mathbf{x}^T\mathbf{b}_m - y_m\right)^2. \quad (2)$$

This is a bilinear, hence non-convex optimization problem. In order to solve it efficiently, we leverage recent ideas in [13] for phase retrieval, to propose judiciously initialized, simple gradient descent iterations that minimize $f(\mathbf{x},\mathbf{h})$. In the sequel, we first present the gradient iterations and then specify a simple procedure to obtain an accurate initial guess of $\{\mathbf{x},\mathbf{h}\}$. These two steps comprise SIGIBE.

A. Gradient iterations

With i denoting an iteration index, the outputs $\{\mathbf{x}_0,\mathbf{h}_0\}$ of the spectral initialization are refined via the following gradient iterations

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mu_{i|x}\nabla_{\mathbf{x}} f(\mathbf{x}_i,\mathbf{h}_i) \quad (3)$$

$$\mathbf{h}_{i+1} = \mathbf{h}_i - \mu_{i|h}\nabla_{\mathbf{h}} f(\mathbf{x}_i,\mathbf{h}_i) \quad (4)$$

where the expressions for the gradients of $f(\mathbf{x},\mathbf{h})$ in (2) are

$$\nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{h}) = \frac{1}{M}\sum_{m=1}^{M}\left(\mathbf{a}_m^T\mathbf{h}\cdot\mathbf{x}^T\mathbf{b}_m - y_m\right)\left(\mathbf{a}_m^T\mathbf{h}\right)\mathbf{b}_m \quad (5)$$

$$\nabla_{\mathbf{h}} f(\mathbf{x},\mathbf{h}) = \frac{1}{M}\sum_{m=1}^{M}\left(\mathbf{a}_m^T\mathbf{h}\cdot\mathbf{x}^T\mathbf{b}_m - y_m\right)\left(\mathbf{b}_m^T\mathbf{x}\right)\mathbf{a}_m. \quad (6)$$

Several alternatives to set the adaptation rules for the stepsizes $\mu_{i|x}$ and $\mu_{i|h}$ arise, each leading to different convergence and recovery performances [17], [18]. The simulations in this paper will be run using rules of the form $\mu_{i|x} = \mu_i/\bar{\mu}_{i|x}$, where $\mu_i = \min\left\{\mu_{\max}, 1 - e^{-i/(-i_{\text{thr}}\ln(1-\mu_{\max}))}\right\}$ [13], and the normalizing constant $\bar{\mu}_{i|x}$ is set to $\|\mathbf{x}\|^2$ – which can be either known, estimated from the spectral initialization, or replaced with $\|\mathbf{x}_i\|^2$. Values for the parameters $\mu_{\max}$ and i_{thr} are chosen in Section IV.

B. Initialization via singular-value decomposition

To build intuition into the general spectral initialization method developed in Section III, here we first introduce a simple but instructive initialization based on the singular-value decomposition (SVD) of the *non-symmetric* $L \times P$ matrix

$$\mathbf{Y}_{NS} := \frac{1}{M}\sum_{m=1}^{M} y_m\mathbf{a}_m\mathbf{b}_m^T. \quad (7)$$

Suppose that $\mathbf{a}_m$ and $\mathbf{b}_m$ are uncorrelated for each $m = 1,\ldots,M$. Plugging in the definition of y_m, and taking expected value while using the moment assumptions on $\{\mathbf{a}_m,\mathbf{b}_m\}_{m=1}^M$ yields

$$\mathbb{E}[\mathbf{Y}_{NS}] = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left[\mathbf{a}_m\mathbf{a}_m^T\right]\mathbf{h}\mathbf{x}^T\mathbb{E}\left[\mathbf{b}_m\mathbf{b}_m^T\right] = \mathbf{h}\mathbf{x}^T \quad (8)$$

which is a rank-one matrix. Since $\mathbf{Y}_{NS} \to \mathbb{E}[\mathbf{Y}_{NS}] = \mathbf{h}\mathbf{x}^T$ as $M \to \infty$ by the Strong Law of Large Numbers (LLN), one would expect that if the number of measurements is large enough then the dominant left-singular vector of $\mathbf{Y}_{NS}$ will be close enough to the direction of $\mathbf{h}$, and likewise the dominant right-singular vector will align with $\mathbf{x}$. This suggests the initialization scheme tabulated under Algorithm 1, whereby the dominant right and left singular vectors of $\mathbf{Y}_{NS}$ are obtained using a power method [19].

Algorithm 1: Spectral initialization for uncorrelated data

INPUTS: $\{y_m\}_{m=1}^M$,$\{\mathbf{a}_m\}_{m=1}^M$, $\{\mathbf{b}_m\}_{m=1}^M$, and $I_{\max}^P$
OUTPUTS: initial estimates $\mathbf{h}_0$ and $\mathbf{x}_0$

Compute $\mathbf{Y}_{NS}$. Resorting to a power method, initialize $\mathbf{v}_0$ as a unit-norm random vector and iterate $\mathbf{u}_i = \mathbf{Y}_{NS}\mathbf{v}_i/\|\mathbf{Y}_{NS}\mathbf{v}_i\|$ and $\mathbf{v}_{i+1} = \mathbf{Y}_{NS}^T\mathbf{u}_i/\|\mathbf{Y}_{NS}^T\mathbf{u}_i\|$, for $i = 0,1,\ldots,I_{\max}^P$. When done, compute $\sigma^2 = \|\mathbf{Y}_{NS}\mathbf{v}_{I_{\max}^P}\|\|\mathbf{Y}_{NS}^T\mathbf{u}_{I_{\max}^P}\|$ and return $\mathbf{x}_0 = \sigma\mathbf{v}_{I_{\max}^P}$ and $\mathbf{h}_0 = \sigma\mathbf{u}_{I_{\max}^P}$.

C. Computational complexity

One of the advantages of SIGIBE is that its complexity scales better than that of SDP-based solvers when L, P and M grow large. To quantify the incurred computational cost we need to analyze both steps of our solver. The gradient updates in (3)-(4) require $\mathcal{O}\big(M(L+P)^2\big)$ operations per iteration. For Algorithm 1, forming matrix $\mathbf{Y}_{NS}$ requires $\mathcal{O}\big(MLP\big)$ operations, while the power method requires $\mathcal{O}\big(I_{\max}^P LP\big)$. In practice, setting $I_{\max}^P = 50$ yields good results. Hence, when M is in the order of hundreds or more, the overall complexity is dominated by the gradient iterations. All in all, the cost is on the order of $\mathcal{O}\big(I_{\max}^G M(L+P)^2\big)$, where $I_{\max}^G$ denotes the number of iterations of the gradient descent method.

The initialization proposed thus far requires the vectors $\mathbf{a}_m$ and $\mathbf{b}_m$ to be uncorrelated [cf. 8]. Furthermore, it is based on a non-symmetric matrix which may complicate the theoretical analysis. In order to accommodate arbitrary correlation between $\mathbf{a}_m$ and $\mathbf{b}_m$ (including $\mathbf{a}_m = \mathbf{b}_m$), in the next section we endow SIGIBE with a more general initialization method that relies on an augmented data matrix.

III. Spectral Initialization for Correlated Data

To overcome the aforementioned limitations, one can introduce the augmented vectors $\boldsymbol{\gamma}_m := [\mathbf{a}_m^T,\ \mathbf{b}_m^T]^T \in \mathbb{R}^{L+P}$ and

form the *symmetric* data matrix

$$\mathbf{Y}_S = \frac{1}{M}\sum_{m=1}^{M} y_m \boldsymbol{\gamma}_m \boldsymbol{\gamma}_m^T. \tag{9}$$

It follows that $y_m = (1/2)\boldsymbol{\gamma}_m^T \mathbf{A} \boldsymbol{\gamma}_m$, where $\mathbf{A}$ is the $(L+P)\times(L+P)$ symmetric, rank-two matrix given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{h} \\ \mathbf{0}_P \end{bmatrix} \begin{bmatrix} \mathbf{0}_L^T & \mathbf{x}^T \end{bmatrix} + \begin{bmatrix} \mathbf{0}_L \\ \mathbf{x} \end{bmatrix} \begin{bmatrix} \mathbf{h}^T & \mathbf{0}_P^T \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{L\times L} & \mathbf{h}\mathbf{x}^T \\ \mathbf{x}\mathbf{h}^T & \mathbf{0}_{P\times P} \end{bmatrix}.$$

Plugging y_m back in (9) and taking expectations yields

$$\mathbb{E}\left[\mathbf{Y}_S\right] = \frac{1}{2}\mathbb{E}\left[\boldsymbol{\gamma}_1 \boldsymbol{\gamma}_1^T \mathbf{A} \boldsymbol{\gamma}_1 \boldsymbol{\gamma}_1^T\right]. \tag{10}$$

The expectation of the quartic form in the right-hand-side of (10) can be evaluated if one assumes that for each $m = 1,\ldots,M$, vectors $\mathbf{a}_m \sim \mathcal{N}(\mathbf{0}_L, \mathbf{I}_L)$ and $\mathbf{b}_m \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$ are i.i.d. (with standard multivariate Gaussian distributions), and have known cross-correlation matrix $\mathbf{C} := \mathbb{E}\left[\mathbf{a}_1 \mathbf{b}_1^T\right] \in \mathbb{R}^{L\times P}$. Thus, the augmented measurement vectors are also Gaussian i.i.d., i.e., $\boldsymbol{\gamma}_m \sim \mathcal{N}(\mathbf{0}_{L+P}, \mathbf{S})$, where

$$\mathbf{S} = \begin{bmatrix} \mathbf{I}_L & \mathbf{C} \\ \mathbf{C}^T & \mathbf{I}_P \end{bmatrix}. \tag{11}$$

Now, for a Gaussian random vector $\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ and a deterministic matrix $\mathbf{M}$, the fourth-order moment $\mathbb{E}\left[\mathbf{a}\mathbf{a}^T\mathbf{M}\mathbf{a}\mathbf{a}^T\right] = \boldsymbol{\Sigma}\left(\mathbf{M}+\mathbf{M}^T\right)\boldsymbol{\Sigma} + \mathrm{tr}[\mathbf{M}\boldsymbol{\Sigma}]\boldsymbol{\Sigma}$ [20]. The expectation in (10) is thus

$$\mathbb{E}\left[\mathbf{Y}_S\right] = \mathbf{S}\mathbf{A}\mathbf{S} + (1/2)\mathrm{tr}\left[\mathbf{A}\mathbf{S}\right]\mathbf{S} \tag{12}$$

and our goal is to determine those eigenvectors of $\mathbb{E}\left[\mathbf{Y}_S\right]$ from which one could hopefully recover $\mathbf{x}$ and $\mathbf{h}$. To this end, define $\tilde{\mathbf{Y}} := \mathbf{S}^{-1}\mathbf{Y}_S$ (recall $\mathbf{S}$ is given) and consider $\mathbb{E}\left[\tilde{\mathbf{Y}}\right]$ which from the structure of $\mathbf{A}$, $\mathbf{S}$ and (12) can be written as

$$\begin{aligned}\mathbb{E}\left[\tilde{\mathbf{Y}}\right] &= \mathbf{A}\mathbf{S} + (1/2)\mathrm{tr}\left[\mathbf{A}\mathbf{S}\right]\mathbf{I}_{L+P} \\ &= \begin{bmatrix} \mathbf{h}\mathbf{x}^T\mathbf{C}^T & \mathbf{h}\mathbf{x}^T \\ \mathbf{x}\mathbf{h}^T & \mathbf{x}\mathbf{h}^T\mathbf{C} \end{bmatrix} + (\mathbf{x}^T\mathbf{C}^T\mathbf{h})\mathbf{I}_{L+P}. \end{aligned} \tag{13}$$

It follows that $\mathbb{E}\left[\tilde{\mathbf{Y}}\right]$ has: (i) an eigenvector $\mathbf{v}_1 = \frac{1}{\sqrt{2}}\left[\mathbf{h}^T/\|\mathbf{h}\|,\ \mathbf{x}^T/\|\mathbf{x}\|\right]^T$ with eigenvalue $\lambda_1 = 2\mathbf{x}^T\mathbf{C}^T\mathbf{h} + \|\mathbf{x}\|\|\mathbf{h}\|$; (ii) an eigenvector $\mathbf{v}_2 = \frac{1}{\sqrt{2}}\left[-\mathbf{h}^T/\|\mathbf{h}\|,\ \mathbf{x}^T/\|\mathbf{x}\|\right]^T$ with eigenvalue $\lambda_2 = 2\mathbf{x}^T\mathbf{C}^T\mathbf{h} - \|\mathbf{x}\|\|\mathbf{h}\|$; and (iii) all other eigenvalues are $\mathbf{x}^T\mathbf{C}^T\mathbf{h}$.

A. Power method

Leveraging $\mathbb{E}\left[\tilde{\mathbf{Y}}\right]$'s favorable eigenstructure [cf. (i)-(iii)], we initialize $\{\mathbf{x}_0, \mathbf{h}_0\}$ based on $\tilde{\mathbf{Y}}$ using the following three-step algorithm. Note that one can avoid the matrix-matrix multiplication in forming $\tilde{\mathbf{Y}}$, by precomputing $\tilde{\boldsymbol{\gamma}}_m := \mathbf{S}^{-1}\boldsymbol{\gamma}_m$ and then evaluating $\tilde{\mathbf{Y}} = M^{-1}\sum_{m=1}^{M} y_m \tilde{\boldsymbol{\gamma}}_m \boldsymbol{\gamma}_m^T$. Inverting $\mathbf{S}$ is still required, though.

The following lemma states two useful asymptotic properties of the initialization $\{\mathbf{x}_0, \mathbf{h}_0\}$ generated by Algorithm 2.

Algorithm 2: Spectral initialization for correlated data

INPUTS: $\{y_m\}_{m=1}^M$,$\{\mathbf{a}_m\}_{m=1}^M$, $\{\mathbf{b}_m\}_{m=1}^M$, $\mathbf{C}$, and $I_{\max}^P$
OUTPUTS: initial estimates $\mathbf{h}_0$ and $\mathbf{x}_0$

Step 1: *Finding* $\mathbf{z}^*$. Compute $\tilde{\mathbf{Y}}$. Resorting to a power method, initialize $\mathbf{z}_0$ as a unit-norm random vector and then iterate $\mathbf{z}_i = \tilde{\mathbf{Y}}\mathbf{z}_{i-1}/\|\tilde{\mathbf{Y}}\mathbf{z}_{i-1}\|$ for $i = 1,\ldots,I_{\max}^P$. Return $\mathbf{z}^* = \mathbf{z}_{I_{\max}^P}$.

Step 2: *Finding the initializations* $\tilde{\mathbf{h}}_0$ *and* $\tilde{\mathbf{x}}_0$ *using* $\mathbf{z}^*$. Extract $\bar{\mathbf{z}}^{top} := [z_1^*,...,z_L^*]^T$, $\bar{\mathbf{z}}^{bot} := [z_{L+1}^*,...,z_{L+P}^*]^T$ from $\mathbf{z}^*$, and normalize $\bar{\mathbf{z}}_h := \bar{\mathbf{z}}^{top}/\|\bar{\mathbf{z}}^{top}\|$, $\bar{\mathbf{z}}_x := \bar{\mathbf{z}}^{bot}/\|\bar{\mathbf{z}}^{bot}\|$. Stack $\bar{\mathbf{z}}_h$ and $\bar{\mathbf{z}}_x$ in $\mathbf{v}_A := \frac{1}{\sqrt{2}}[\bar{\mathbf{z}}_h^T, \bar{\mathbf{z}}_x^T]^T$, $\mathbf{v}_B := \frac{1}{\sqrt{2}}[-\bar{\mathbf{z}}_h^T, \bar{\mathbf{z}}_x^T]^T$, then compute $\lambda_A = \|\tilde{\mathbf{Y}}\mathbf{v}_A\|$, $\lambda_B = \|\tilde{\mathbf{Y}}\mathbf{v}_B\|$ and $\lambda_{xh} = (\lambda_A + \lambda_B)/2$. Finally, set $\tilde{\mathbf{h}}_0 = \sqrt{\lambda_{xh}}\bar{\mathbf{z}}_h$ and $\tilde{\mathbf{x}}_0 = \sqrt{\lambda_{xh}}\bar{\mathbf{z}}_x$.

Step 3: *Fixing the sign of the initializations.* If the sign of the entries $[\tilde{\mathbf{Y}}]_{1,L+1}$ and that of $[\tilde{\mathbf{h}}_0\tilde{\mathbf{x}}_0^T]_{1,1}$ coincide, return $\mathbf{h}_0 = \tilde{\mathbf{h}}_0$ and $\mathbf{x}_0 = \tilde{\mathbf{x}}_0$. Otherwise, return $\mathbf{h}_0 = -\tilde{\mathbf{h}}_0$ and $\mathbf{x}_0 = \tilde{\mathbf{x}}_0$.

Lemma 1 *As* $M \to \infty$, *then: (P1)* $\|\mathbf{h}_0\| = \|\mathbf{x}_0\| = \sqrt{\|\mathbf{h}\|\|\mathbf{x}\|}$, *and (P2)* $\mathbf{h}_0\mathbf{x}_0^T = \mathbf{h}\mathbf{x}^T$.

Proof: First note that as $M \to \infty$, the LLN guarantees that $\tilde{\mathbf{Y}} = \mathbf{S}^{-1}\mathbb{E}\left[\tilde{\mathbf{Y}}_S\right]$. Second, if $I_{\max}^P$ is large enough, Step 1's output $\mathbf{z}^*$ is the dominant eigenvector of $\tilde{\mathbf{Y}}$. Asymptotically, from (i)-(iii) the dominant (unit-norm) eigenvector can either be $\mathbf{v}_1 = \frac{1}{\sqrt{2}}\left[\mathbf{h}^T/\|\mathbf{h}\|,\ \mathbf{x}^T/\|\mathbf{x}\|\right]^T$ or $\mathbf{v}_2 = \frac{1}{\sqrt{2}}\left[-\mathbf{h}^T/\|\mathbf{h}\|,\ \mathbf{x}^T/\|\mathbf{x}\|\right]^T$, with associated eigenvalues $\lambda_1 = 2\mathbf{x}^T\mathbf{C}^T\mathbf{h} + \|\mathbf{x}\|\|\mathbf{h}\|$ and $\lambda_2 = 2\mathbf{x}^T\mathbf{C}^T\mathbf{h} - \|\mathbf{x}\|\|\mathbf{h}\|$. If $\mathbf{x}^T\mathbf{C}^T\mathbf{h} > 0$ the main eigenvector is either: (a.1) $\mathbf{z}^* = \mathbf{v}_1$, or (a.2) $\mathbf{z}^* = -\mathbf{v}_1$. On the other hand, if $\mathbf{x}^T\mathbf{C}^T\mathbf{h} < 0$, we have that either: (b.1) $\mathbf{z}^* = \mathbf{v}_2$, or (b.2) $\mathbf{z}^* = -\mathbf{v}_2$.

Suppose that (a.1) holds true and $\mathbf{z}^* = \mathbf{v}_1$. Then, we have that $\bar{\mathbf{z}}_h = \mathbf{h}/\|\mathbf{h}\|$, $\bar{\mathbf{z}}_x = \mathbf{x}/\|\mathbf{x}\|$, $\mathbf{v}_A = \mathbf{v}_1$, $\mathbf{v}_B = \mathbf{v}_2$, $\lambda_A = \lambda_1$, $\lambda_B = -\lambda_2$ and $\lambda_{hx} = \|\mathbf{h}\|\|\mathbf{x}\|$. This implies that $\tilde{\mathbf{h}}_0 = \sqrt{\lambda_{xh}}\bar{\mathbf{z}}_h = \sqrt{\|\mathbf{x}\|/\|\mathbf{h}\|}\mathbf{h}$ and $\tilde{\mathbf{x}}_0 = \sqrt{\lambda_{xh}}\bar{\mathbf{z}}_x = \sqrt{\|\mathbf{h}\|/\|\mathbf{x}\|}\mathbf{x}$. Then, it follows that $\|\mathbf{h}_0\| = \|\tilde{\mathbf{h}}_0\| = \sqrt{\|\mathbf{x}\|\|\mathbf{h}\|}$ and $\|\mathbf{x}_0\| = \|\tilde{\mathbf{x}}_0\| = \sqrt{\|\mathbf{h}\|\|\mathbf{x}\|}$, so that the claim in (P1) follows. If (a.2) holds true, then one has $\bar{\mathbf{z}}_h = -\mathbf{h}/\|\mathbf{h}\|$ and $\bar{\mathbf{z}}_x = -\mathbf{x}/\|\mathbf{x}\|$. This leads to $\lambda_{xh} = \|\mathbf{h}\|\|\mathbf{x}\|$, $\tilde{\mathbf{h}}_0 = -\sqrt{\|\mathbf{x}\|/\|\mathbf{h}\|}\mathbf{h}$ and $\tilde{\mathbf{x}}_0 = -\sqrt{\|\mathbf{h}\|/\|\mathbf{x}\|}\mathbf{x}$, so that (P1) is again true. Likewise, one can show that when (b.1) holds, then $\bar{\mathbf{z}}_h = -\mathbf{h}/\|\mathbf{h}\|$ and $\bar{\mathbf{z}}_x = \mathbf{x}/\|\mathbf{x}\|$, which leads to $\lambda_{xh} = \|\mathbf{h}\|\|\mathbf{x}\|$, $\tilde{\mathbf{h}}_0 = -\sqrt{\|\mathbf{x}\|/\|\mathbf{h}\|}\mathbf{h}$ and $\tilde{\mathbf{x}}_0 = \sqrt{\|\mathbf{h}\|/\|\mathbf{x}\|}\mathbf{x}$. Finally, for (b.2) one has $\bar{\mathbf{z}}_h = \mathbf{h}/\|\mathbf{h}\|$ and $\bar{\mathbf{z}}_x = -\mathbf{x}/\|\mathbf{x}\|$, which leads to $\lambda_{xh} = \|\mathbf{h}\|\|\mathbf{x}\|$, $\tilde{\mathbf{h}}_0 = \sqrt{\|\mathbf{x}\|/\|\mathbf{h}\|}\mathbf{h}$ and $\tilde{\mathbf{x}}_0 = -\sqrt{\|\mathbf{h}\|/\|\mathbf{x}\|}\mathbf{x}$. Both for (b.1) and (b.2), property (P1) is true.

In the previous analysis we did not include the effect of changing the sign of $\tilde{\mathbf{h}}_0$ (Step 3 of Algorithm 2). The reason was that the sign is irrelevant for the claim (P1), which pertains to the norms of $\{\mathbf{x}_0, \mathbf{h}_0\}$. However, the signs matter for the claim (P2). In the four cases analyzed before [(a.1), (a.2), (b.1) and (b.2)] we showed that $\tilde{\mathbf{h}}_0$ is either $\pm\sqrt{\|\mathbf{x}\|/\|\mathbf{h}\|}\mathbf{h}$ and

that $\tilde{\mathbf{x}}_0$ is $\pm\sqrt{\|\mathbf{h}\|/\|\mathbf{x}\|}\mathbf{x}$. This implies that the outer product $\tilde{\mathbf{h}}_0\tilde{\mathbf{x}}_0^T$ will be either $\mathbf{h}\mathbf{x}^T$ (if the signs coincide) or $-\mathbf{h}\mathbf{x}^T$ (if the signs are different). The purpose of Step 3 is to change the sign of one of the initializations when $\tilde{\mathbf{h}}_0\tilde{\mathbf{x}}_0^T = -\mathbf{h}\mathbf{x}^T$. After that change $\mathbf{h}_0\mathbf{x}_0^T = \mathbf{h}\mathbf{x}^T$ holds under the four cases, as claimed in (P2). ■

Lemma 1 states that when $M \to \infty$, Algorithm 2 identifies $\{\mathbf{x}, \mathbf{h}\}$ up to an inherent scaling ambiguity. Although the generated initializations have the same norm, if available information about the value of $\|\mathbf{x}\|$ and $\|\mathbf{h}\|$ can be easily incorporated into the algorithm. In closing, a quick remark on the algorithm's complexity is in order.

Remark 1 The computational cost incurred by Algorithm 2 is higher than that for computing $\mathbf{Y}_{NS}$ and running Algorithm 1. The reason is twofold: a) the power method used in Algorithm 2 is more involved and operates on a larger matrix, and b) computing matrix $\tilde{\mathbf{Y}}$ requires inverting the block matrix $\mathbf{S}$. To be specific, the complexity associated with Algorithm 2 is dominated by Step 1, which requires: a) $\mathcal{O}\big(I_{\max}(L+P)^2\big)$ iterations for the power method, and b) $\mathcal{O}\big((L+P)^3\big)$ operations to compute $\mathbf{S}^{-1}$ and then $\mathcal{O}\big(M(L+P)^2\big)$ more to evaluate $\tilde{\mathbf{Y}} = M^{-1}\sum_{m=1}^{M} y_m(\mathbf{S}^{-1}\boldsymbol{\gamma}_m\boldsymbol{\gamma}_m^T)$. Note that in online setups where $\{\mathbf{x}, \mathbf{h}\}$, hence y_m, can change frequently and the value of $(\mathbf{S}^{-1}\boldsymbol{\gamma}_m\boldsymbol{\gamma}_m^T)$ remains the same, the complexity of the matrix inversion can be neglected. In any case, SIGIBE's overall complexity is dominated by the gradient iterations with cost $\mathcal{O}\big(I_{\max}^G M(L+P)^2\big)$.

B. Special cases

We now consider two important special cases subsumed by our general formulation. In the first, $\mathbf{a}_m$ and $\mathbf{b}_m$ are uncorrelated for each $m = 1, \ldots, M$ as in Section II-B. In this case $\mathbf{C} = \mathbf{0}_{L\times P}$ and $\mathbf{S} = \mathbf{I}_{L+P}$. Hence, simplifying (12) yields $\mathbb{E}[\mathbf{Y}_S] = \mathbf{A}$ (notice that $\mathrm{tr}[\mathbf{A}] = 0$). Then it follows that $\mathbb{E}[\mathbf{Y}_S] = \mathbf{A}$ has: (i) the eigenvector $\mathbf{v}_1 = \frac{1}{\sqrt{2}}\big[\mathbf{h}^T/\|\mathbf{h}\|,\, \mathbf{x}^T/\|\mathbf{x}\|\big]^T$ with eigenvalue $\lambda_1 = \|\mathbf{x}\|\|\mathbf{h}\|$; (ii) the eigenvector $\mathbf{v}_2 = \frac{1}{\sqrt{2}}\big[-\mathbf{h}^T/\|\mathbf{h}\|,\, \mathbf{x}^T/\|\mathbf{x}\|\big]^T$ with eigenvalue $\lambda_2 = -\|\mathbf{x}\|\|\mathbf{h}\|$; and (iii) all other eigenvalues are zero.

The other special case is when $\mathbf{a}_m = \mathbf{b}_m$ (fully correlated), so that $\mathbf{C} = \mathbf{I}_P$ and

$$\mathbf{S} = \begin{bmatrix} \mathbf{I}_P & \mathbf{I}_P \\ \mathbf{I}_P & \mathbf{I}_P \end{bmatrix}.$$

Simplifying (12) for this particular $\mathbf{S}$, one observes that each of the four $P \times P$ blocks of $\mathbb{E}[\mathbf{Y}_S]$ are identical and equal to e.g., the top-left one $[\mathbb{E}[\mathbf{Y}_S]]_{1:P,1:P} = \mathbf{h}\mathbf{x}^T + \mathbf{x}\mathbf{h}^T + (\mathbf{x}^T\mathbf{h})\mathbf{I}_P$. Then it follows that $[\mathbb{E}[\mathbf{Y}_S]]_{1:P,1:P}$ has: (i) the eigenvector $\mathbf{v}_1 = \mathbf{x}/\|\mathbf{x}\| + \mathbf{h}/\|\mathbf{h}\|$ with eigenvalue $\lambda_1 = 2\mathbf{x}^T\mathbf{h} + \|\mathbf{x}\|\|\mathbf{h}\|$; (ii) the $\mathbf{v}_2 = \mathbf{x}/\|\mathbf{x}\| - \mathbf{h}/\|\mathbf{h}\|$ with eigenvalue $\lambda_2 = 2\mathbf{x}^T\mathbf{h} - \|\mathbf{x}\|\|\mathbf{h}\|$; and (iii) all other eigenvalues are $\mathbf{x}^T\mathbf{h}$. This can be used to simplify Algorithm 2, that now operates over a matrix with smaller size and does not require pre-whitening with $\mathbf{S}^{-1}$.

IV. Numerical tests

Here we present preliminary simulation results to asses SIGIBE's performance. Three test cases are considered: uncorrelated, loosely correlated and highly correlated measurement vectors $\mathbf{a}_m$ and $\mathbf{b}_m$, for $m = 1, \ldots, M$. The default setup is as follows: $P = 64$ and $L = 128$, while vectors $\mathbf{x}$ and $\mathbf{h}$ are generated randomly according to a multivariate, zero-mean Gaussian distribution with variances $\sigma_x^2 = 4^2$ and $\sigma_h^2 = 1^2$, respectively. The number of the gradient iterations is $I_{\max}^G = 500$ and the stepsize adaptation rule uses $\mu_{\max} = 0.4$, $i_{\text{thr}} = 75$, $\bar{\mu}_{i|x} = \|\mathbf{x}_i\|^2$ and $\bar{\mu}_{i|h} = \|\mathbf{h}_i\|^2$. Results are averaged across 100 realizations of $\{\mathbf{x}, \mathbf{h}\}$.

Uncorrelated measurement vectors. Five algorithms are adopted to carry out the comparisons: A1) SIGIBE using Algorithm 1; A2) SIGIBE using Algorithm 2 for $\mathbf{C} = \mathbf{0}$; A3) random initializations with $K_1 = 5$ seeds; A4) random initializations with $K_2 = 15$ seeds; and A5) SDP relaxation based on matrix lifting [1]. The relative error $\text{err} = \|\mathbf{x}\mathbf{h}^T - \hat{\mathbf{x}}\hat{\mathbf{h}}^T\|_F/\|\mathbf{x}\mathbf{h}^T\|_F$ is adopted as figure of merit, where $\|\cdot\|_F$ denotes the Frobenius norm. The results are plotted in the first (left) column of Fig 1. The top panel reports the median error and the bottom one shows the probability of successful recovery. The first observation is that when the number of observations is small so that $M \leq 1.5L$, none of the algorithms is able to find the solution. Note that $M = 1.5L = L + S$, so that $M = 1.5L$ is the minimum number that yields as many equations as unknowns. On the other hand, when $M \geq 8L$ all algorithms find the optimal vectors (up to scaling). Within the intermediate range, we observe that the less computationally demanding SIGIBE variants outperform their competitors. Moreover, the algorithms based on random initializations perform surprisingly well, especially when M is large. Finally, although both SIGIBE variants perform very similarly, the slight advantage of using Algorithm 1 can be attributed to the fact that $\mathbf{Y}_{NS}$ is smaller, so that the approximation to $\mathbb{E}[\mathbf{Y}_{NS}]$ is more accurate for each M. Regarding the running times, if we set as reference the running time of A1 (which is the fastest), the speed of A2 is similar, A3 is 5.0 times slower, A4 14.9 times slower, and A5 is on average 20.4 times slower than A1.

Correlated measurement vectors. For simplicity, a setup with $P = L = 128$ and $\mathbf{C} = \rho\mathbf{I}_P$ is considered. Due to space limitations, Fig. 1 presents results for only two values of ρ. The two panels in the second (center) column correspond to $\rho = 0.25$ and the two in the third (right) column correspond to $\rho = 0.75$. Once more, algorithms A1-A5 are compared here (with the actual $\mathbf{C}$ used in A2). The main observation is that configurations with higher correlation are more challenging. For $\rho = 0.25$, M must be in the order of $5.5L$, while for $\rho = 0.75$, $M = 6.5L$ is required. The results confirm the previous findings: SIGIBE outperforms the other tested alternatives and for high values of M all algorithms are able to find the optimal solution. Interestingly, A1 which implicitly assumes uncorrelated $\mathbf{a}_m$ and $\mathbf{b}_m$, works satisfactorily for the

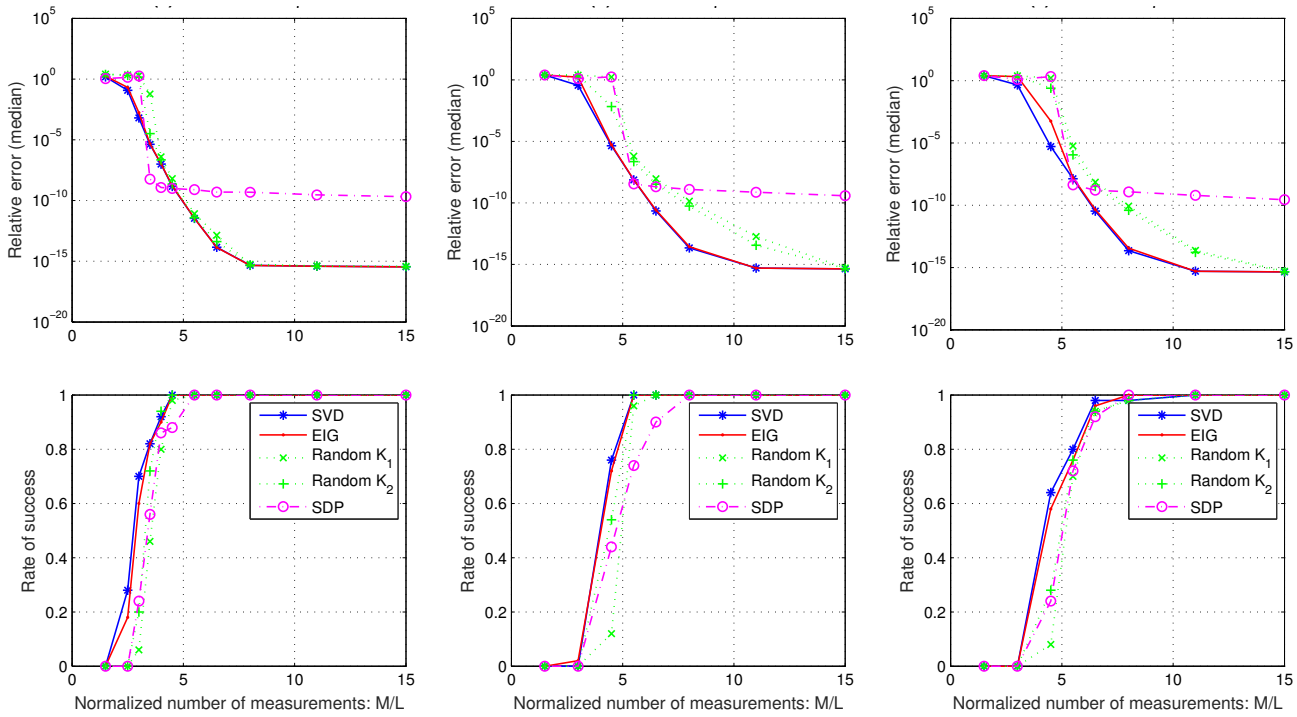

Fig. 1. Recovery performance of the joint identification of $\mathbf{x}$ and $\mathbf{h}$ using 5 different algorithms. The top row reports the median error, and the bottom one the percentage of tests able to recover the exact solution (up to a relative error of less than 10^{-3}). Each column represents a different test case.

values of ρ tested.

V. Conclusions

We developed SIGIBE – a carefully initialized, simple gradient descent algorithm to solve for $\{\mathbf{x}, \mathbf{h}\}$ in a system of M bilinear equations $y_m = \langle \mathbf{a}_m, \mathbf{h}\rangle\langle \mathbf{b}_m, \mathbf{x}\rangle$. SIGIBE can accommodate correlations between $\mathbf{a}_m$ and $\mathbf{b}_m$, and scales well to high-dimensional problems. Our current research seeks to substantiate the encouraging performance observed in simulated tests through theoretical recovery guarantees, and extend SIGIBE to the complex case.

References

[1] A. Ahmed, B. Recht, and J. Romberg, "Blind deconvolution using convex programming," *IEEE Trans. Inf. Theory*, vol. 60, no. 3, pp. 1711–1732, 2014.

[2] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman, "Understanding blind deconvolution algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2354–2367, 2011.

[3] T. Strohmer S. Ling, "Self-calibration and biconvex compressive sensing," *arXiv preprint arXiv:1501.06864 [cs.IT]*, 2015.

[4] S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, "Blind identification of graph filters with sparse inputs," in *6th Int. Workshop on Comp. Advances in Multi-Sensor Adaptive Process.*, Cancun, Mexico, Dec. 13-16 2015.

[5] J. R. Fienup, "Phase retrieval algorithms: a comparison," *Appl. Opt.*, vol. 21, pp. 2758–2769, 1982.

[6] J. R. Fienup, "Phase retrieval algorithms: a personal tour," *Appl. Opt.*, vol. 52, pp. 45–56, 2013.

[7] Yoav Shechtman, Yonina C Eldar, Oren Cohen, Henry Nicholas Chapman, Jianwei Miao, and Mordechai Segev, "Phase retrieval with application to optical imaging: a contemporary overview," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 87–109, 2015.

[8] E. J. Candès, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM J. Imaging Sci.*, vol. 6, no. 1, pp. 199–225, 2013.

[9] Y. Shechtman, Y.C. Eldar, A. Szameit, and M. Segev, "Sparsity based sub-wavelength imaging with partially incoherent light via quadratic compressed sensing," *Optics Express*, vol. 19, no. 16, pp. 14807–14822, 2011.

[10] I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, maxcut and complex semidefinite programming," *arXiv preprint arXiv:1206.0102*, 2012.

[11] Y. Shechtman, A. Beck, and Y.C. Eldar, "GESPAR: Efficient phase retrieval of sparse signals," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 928–938, 2014.

[12] A. Beck and Y. C. Eldar, "Sparsity constrained nonlinear optimization: Optimality conditions and algorithms," *SIAM Optimization*, vol. 23, no. 3, pp. 1480–1509, Oct. 2013.

[13] 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, 2015.

[14] E. J. Candès Y. Chen, "Solving random quadratic systems of equations is nearly as easy as solving linear systems," *arXiv preprint arXiv:1505.05114 [cs.IT]*, 2015.

[15] Q. Zheng and J. Lafferty, "A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements," *arXiv preprint arXiv:1506.06081 [stat.ML]*, 2015.

[16] S. Tu, R. Boczar, M. Soltanolkotabi, and B. Recht, "Low-rank solutions of linear matrix equations via procrustes flow," *arXiv preprint arXiv:1507.03566 [math.OC]*, 2015.

[17] D. P. Bertsekas, *Nonlinear programming*, Athena Scientific, 1999.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge Univ. Press, New York, NY, USA, 2004.

[19] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*, Johns Hopkins Univ. Press, Baltimore, MD, USA, 1996.

[20] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," http://www2.imm.dtu.dk/pubdb/p.php?3274, Nov. 2012.