Separation of Synchronous Sources Through Phase Locked Matrix Factorization

Miguel S. B. Almeida, Ricardo Vigário, and José Bioucas-Dias, *Member, IEEE*

***Abstract*—In this paper, we study the separation of synchronous sources (SSS) problem, which deals with the separation of sources whose phases are synchronous. This problem cannot be addressed through independent component analysis methods because synchronous sources are statistically dependent. We present a two-step algorithm, called phase locked matrix factorization (PLMF), to perform SSS. We also show that SSS is identifiable under some assumptions and that any global minimum of PLMFs cost function is a desirable solution for SSS. We extensively study the algorithm on simulated data and conclude that it can perform SSS with various numbers of sources and sensors and with various phase lags between the sources, both in the ideal (i.e., perfectly synchronous and nonnoisy) case, and with various levels of additive noise in the observed signals and of phase jitter in the sources.**

***Index Terms*—Independent component analysis (ICA), matrix factorization, phase-locking, source separation, synchrony.**

I. Introduction

Blind source separation (BSS) is an important class of signal processing problems, which arise in several domains, such as speech processing, image processing, telecommunications, and biomedical applications. In BSS, one has access to a set of measurements $\mathbf{y}(t) \in \mathbb{R}^P$ or $\mathbb{C}^P$, where $\mathbb{R}$ and $\mathbb{C}$ denote the real and complex fields, respectively. These measurements result from a superposition of another set of signals $\mathbf{s}(t) \in \mathbb{R}^N$ or $\mathbb{C}^N$, called sources, which are not directly observable. The goal of BSS is to reconstruct the sources using only the measurements.

Within the broad BSS class there are many types of problems. Linear and instantaneous BSS problems are a relevant subclass, where each entry of $\mathbf{y}(t)$ is a linear combination of the components of $\mathbf{s}(t)$. In this case, the problem is described by the equation $\mathbf{y}(t) = \mathbf{M}\mathbf{s}(t)$, where the mixing matrix $\mathbf{M}$ contains the coefficients of the linear combinations. Generally, both the sources $\mathbf{s}(t)$ and the mixing matrix $\mathbf{M}$ are unknown, and must be estimated using only the observed data $\mathbf{y}(t)$.

Linear BSS problems can also have a convolutive nature, in which case they are known as convolutive BSS (see [1] for an extensive overview). Nonlinear BSS has also been addressed, even though it is considerably harder than its linear counterpart [2], [3].

Although linear and instantaneous BSS is the simplest class of BSS problems, it is far from being trivial. Generally, the BSS problem is ill-posed, i.e., it has an infinity of solutions, because there are infinitely many choices for $\mathbf{M}$ and $\mathbf{s}(t)$ yielding a given $\mathbf{y}(t)$. To solve this problem, one must assume extra conditions on the sources $\mathbf{s}(t)$, on the mixing matrix $\mathbf{M}$, or on both. A popular choice is to assume that, for each time point t, the vector $\mathbf{s}(t)$ is a realization of a random vector whose components are statistically independent. This is the fundamental assumption of independent component analysis (ICA) [4], [5]. While in ICA there still are infinitely many solutions, it can be shown that all of them are equivalent, up to certain nonessential indeterminations, which are typical of BSS problems (see Section II-E). Good overviews of ICA can be found in [6]–[8].

A generalization of ICA is independent subspace analysis (ISA) [9]–[11], also known as subspace ICA [12] and multidimensional ICA [13]. In ISA, there are several groups of sources; within each such group (or subspace) the sources may have dependencies, but groups are mutually independent. ICA is a particular case of ISA with only one source per subspace (see [11] and references therein).

A BSS problem, which has seen increasing interest in recent years, assumes that the sources and the mixing matrix are nonnegative. This problem is known as nonnegative matrix factorization (NMF) [14], [15]. NMF has been extended to the complex domain, and is used, e.g., in the separation of audio signals [16], [17]. This recent field is now known as complex matrix factorization.

This paper deals with yet another BSS problem. We assume that the sources have perfect phase synchrony [18], as measured by their pairwise phase locking factors (PLFs) [19], [20]; we define the PLF in Section II-B. Our motivation for this choice comes from neuroscience: phase synchrony and similar concepts have been used to study electroencephalographic (EEG) and magnetoencephalographic (MEG) signals for more than a decade (see [19]–[21]). As an example, it has been shown that muscle activity measured with an electromyogram and motor cortex EEG or MEG have coherent oscillations when a person engages in a motor task [22], [23]. It was also found that, again during a motor task, several brain regions oscillate coherently with one another [21], [23]. In addition, there are multiple indications that several pathologies, including Alzheimer, Parkinson, autism, and epilepsy, are related to a disruption in the synchronization profile of the brain (see [24] and references therein for a review). Despite this

Manuscript received May 26, 2013; revised October 19, 2013; accepted December 21, 2013. Date of publication January 14, 2014; date of current version September 15, 2014. This work was supported by the Project DECA-Bio of Instituto de Telecomunicacoes under Grant PEst-OE/EEI/LA0008/2013.

M. S. B. Almeida and J. Bioucas-Dias are with the Instituto de Telecomunicações, Lisbon 1049-001, Portugal (e-mail: malmeida@lx.it.pt; bioucas@lx.it.pt).

R. Vigário is with the School of Science, Aalto University, Espoo 02150, Finland (e-mail: ricardo.vigario@aalto.fi).

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/TNNLS.2013.2297791

motivation, none of the work presented in this paper is specific to neuroscience; in particular, the proposed algorithm can be applied to any field where synchronous sources are mixed and these mixing needs to be undone.

In this paper, then, the goal is to solve the BSS problem assuming that all the sources are fully synchronous. This is a linear and instantaneous BSS problem which we call synchronous source separation (SSS).

Previous work on the SSS problem includes a generalization of it, where the sources are organized in subspaces, with sources in the same subspace having strong synchrony and sources in different subspaces having weak synchrony [25]. This problem was tackled with a method related to the one proposed here: independent phase analysis (IPA) is a two-stage algorithm, which performs well in the noiseless case and with moderate levels of added Gaussian white noise [25], [26]. In short, IPA uses TDSEP[1] [27] to separate the subspaces from one another. Then, the separation within each subspace is an SSS problem, which is completely independent from the separations within all other subspaces; IPA uses an optimization procedure to perform these intrasubspace separations. Although it performs well for the noiseless case, with various types of sources and various subspace structures and can even tolerate moderate amounts of noise, its performance for higher noise levels is unsatisfactory. Also, in its current form, IPA is limited to square mixing matrices, i.e., to cases where **y** and **s** are vectors of the same size. It is also susceptible to returning singular solutions, which are local optima where two or more estimated sources are identical [25].

In this paper, we present an alternative technique to solve the SSS problem, named PLMF [28], [29]. Compared with IPA, PLMF has no singular solutions and can deal with higher amounts of noise and with nonsquare mixing matrices (more measurements than sources). Perhaps more importantly, PLMF has a stronger theoretical foundation, which yields some important identifiability properties, discussed ahead in Section III. Although the model used in PLMF assumes perfect synchronization between the sources, we will present results studying PLMFs robustness to deviations of this model, induced, e.g., by additive noise or by imperfect synchronization between the sources.

This paper is organized as follows. In Section II, we provide an overview of some theoretical concepts that will be needed throughout this paper. Section III describes the PLMF algorithm and presents theorems that support the approach it takes. Section IV shows the results of an extensive set of experiments on simulated data. These results are discussed in Section V, and conclusion is drawn in Section VI.

II. Background

A. Phase of Real-Valued Signals

In complex signals, the phase is uniquely defined (up to a multiple of 2π). However, in many real-world applications, such as brain EEG or MEG, the available measurements are real valued. In the case of brain signals, it is very common to preprocess them with bandpass filters with relatively narrow bands. In such cases, a meaningful phase can be obtained by first transforming those real signals into the corresponding complex-valued analytic signals, obtained through the Hilbert transform [30]), and then extracting the phase of the analytic signals [31].

The transformation of a real signal into its corresponding analytic signal is a linear operation [31]. Thus, linear combinations of real signals result in the same linear combinations of the corresponding analytic signals. Since this paper deals with linear mixing processes, we consider the sources directly as complex-valued with no loss of generality, and present results on simulated data, which are directly generated as complex valued.

B. Phase-Locking Factor

Let $\phi_j(t)$ and $\phi_k(t)$, for $t = 1, \dots, T$, denote the time-dependent phases of signals j and k. The real-valued[2] PLF between these two signals is defined as

$$\varrho_{jk} \equiv \left| \frac{1}{T} \sum_{t=1}^{T} e^{\mathrm{i}[\phi_j(t)-\phi_k(t)]} \right| = \left| \left\langle e^{\mathrm{i}(\phi_j-\phi_k)} \right\rangle \right| \tag{1}$$

where $|\cdot|$ and $\langle\cdot\rangle$ are the absolute value and time average operators, and $\mathrm{i} = \sqrt{-1}$. Note that $0 \leq \varrho_{jk} \leq 1$. The value $\varrho_{jk} = 1$ corresponds to two signals that are perfectly synchronized: their phase lag, defined as $\Delta\phi_{jk}(t) \equiv \phi_j(t)-\phi_k(t)$, is constant. For an infinite observation period T, the value $\varrho_{jk} = 0$ is attained, e.g., if the two phases are uniformly distributed in $[0, 2\pi)$ and are statistically independent. For finite T, even that situation may yield nonzero values of ϱ_{jk}, which will tend to become smaller as T grows. Values between 0 and 1 represent partial synchrony. Note that a signal's PLF with itself is trivially equal to 1; thus, for all j, $\varrho_{jj} = 1$.

C. Effect of Mixing on the PLF

We now discuss and illustrate the effect of linearly mixing sources which have all pairwise PLFs equal to 1. The effect of such an operation has a simple, yet powerful, and mathematical characterization: it was shown in [25] that if $\mathbf{s}(t)$ is a set of such sources and if we define $\mathbf{y}(t) \equiv \mathbf{Ms}(t)$ with $\det(\mathbf{M}) \neq 0$, then the only possibility for the observations $\mathbf{y}$ to have all pairwise PLFs equal to 1 is if $\mathbf{M}$ is a permutation of a diagonal matrix (as long as mild assumptions on the sources are met). In other words, the only possibility for all pairwise PLFs to be 1 is for **y** to be equal to **s** up to permutation and scaling, a typical, nonrestrictive indeterminacy in source separation problems (indeterminacies are further discussed in Section II-E).

This effect is illustrated in Fig. 1, which shows four perfectly synchronized sources and their PLFs. This figure also shows four signals obtained through a linear mixing of the sources, and their PLFs. These mixtures have PLFs below 1, in accordance with the result stated in the previous paragraph.

[1] TDSEP is an algorithm which uses time-lagged correlations to separate sources.

[2] Real-valued is used here to distinguish from other works, where the absolute value operator is dropped, hence making the PLF a complex quantity [25].

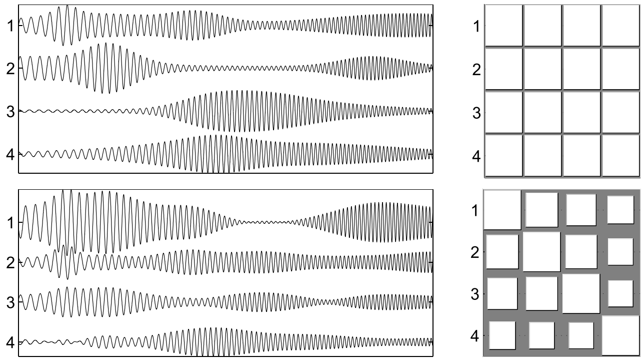

Fig. 1. Top row: three sources (left) and PLFs between them (right). On the right column, the area of the square in position (i, j) is proportional to the PLF between the signals i and j. Therefore, large squares represent PLFs close to 1, while small squares represent values close to zero. The smallest off-diagonal element is 0.9944, instead of 1, due to the finite value of T. Bottom row: three mixed signals (left) and PLFs between them (right). The largest off-diagonal element is 0.7444.

This property shows that separation of sources is necessary to make any type of inference about their synchrony, as measured through the PLF. If the sources are not properly separated, the synchrony values will not be accurate. Established BSS methods such as ICA are not adequate for this separation task, since phase-locked sources are not mutually independent. In fact, ICA-based methods have been shown to fail in this type of problem [25]. PLMF, presented ahead, is a source separation algorithm tailored specifically to this problem.

D. Model

Let us assume that we have a set of N complex-valued sources $s_j(t)$ for $j = 1, \dots, N$ and $t = 1, \dots, T$. We assume that N is known. Let $\mathbf{S}$ denote a $N \times T$ complex-valued matrix whose (j, t)th entry is $s_j(t)$. One can separately represent the amplitude and phase components of the sources through the decomposition $\mathbf{S} = \mathbf{A} \odot \Phi$, where $\odot$ is the element wise (or Hadamard) product, $\mathbf{A}$ is a real-valued $N \times T$ matrix with its (j, t)th element defined as $a_j(t) \equiv |s_j(t)|$, and Φ is a $N \times T$ complex-valued matrix with its (j, t)th element defined as $\Phi_j(t) \equiv e^{\mathrm{i} \arg[s_j(t)]} \equiv e^{\mathrm{i}\phi_j(t)}$. The elements of $\mathbf{A}$ are nonnegative, whereas those of Φ have unit absolute value.

The representation of $\mathbf{S}$ in terms of amplitude and phase is, thus far, completely general: it merely represents $\mathbf{S}$ in polar coordinates. Let us now assume that the sources are perfectly synchronized; as discussed in Section II-B, in this situation $\Delta\phi_{jk}(t) = \phi_j(t) - \phi_k(t)$ is constant (independent of t), for any j and k. For this reason, from now on we drop the dependency on t from $\Delta\phi_{jk}$. Φ can always be decomposed as

$$\Phi \equiv \mathbf{z}\mathbf{f}^T \tag{2}$$

where $\mathbf{z} \equiv [z_1, \dots, z_N]^T$ is a complex-valued column vector of size N, with all entries having unit absolute value, containing the relative phase lags of the sources. Similarly, $\mathbf{f} \equiv [f_1, \dots, f_T]^T$ is a complex-valued column vector of size T, with entries with unit absolute value, containing the common phase variation along time. If the sources are phase-locked, then $\mathrm{rank}(\Phi) = 1$, and the above decomposition is always possible. Note, however, that this decomposition is not unique.[3] The time evolution of each source's phase is given by $\phi_j(t) = \arg(z_j) + \arg(f_t)$, where z_j and f_t are the jth element of $\mathbf{z}$ and the tth element of $\mathbf{f}$, respectively.

We assume that we only have access to P measurements ($P \geq N$) with time duration T each, that result from a linear mixing of the sources, as is customary in source separation problems

$$\mathbf{Y} \equiv \mathbf{MS} + \mathbf{N} \tag{3}$$

where $\mathbf{Y}$ is a $P \times T$ matrix containing the measurements, $\mathbf{M}$ is a $P \times N$ real-valued mixing matrix and $\mathbf{N}$ is a $P \times T$ complex-valued noise matrix. In our analysis, we will start by dealing with the noiseless model, where $\mathbf{N} = 0$. Later, we will also test how PLMF copes with noisy data. In the noiseless case, then, the model of the observed data is

$$\mathbf{Y} \equiv \mathbf{MS} = \mathbf{M}[\mathbf{A} \odot (\mathbf{z}\mathbf{f}^T)] = \mathbf{M}\mathbf{D_z}\mathbf{A}\mathbf{D_f} \tag{4}$$

where $\mathbf{D_z} \equiv \mathrm{diag}(\mathbf{z})$ is a $N \times N$ diagonal matrix with the elements of $\mathbf{z}$ on its diagonal, and $\mathbf{D_f} \equiv \mathrm{diag}(\mathbf{f})$ is a $T \times T$ diagonal matrix with the elements of $\mathbf{f}$ on its diagonal.

E. Indeterminacies

BSS problems usually do not have unique solutions. For example, in ICA it is well known that the order of the estimated sources might not be the same as their original order. This happens because if $\mathbf{Y} = \mathbf{MS}$, then we also have $\mathbf{Y} = \mathbf{M}_0\mathbf{S}_0$ where $\mathbf{M}_0$ is any permutation of the columns of $\mathbf{M}$, and $\mathbf{S}_0$ is the same permutation applied to the rows of $\mathbf{S}$. Also, the amplitude scale of the sources cannot be determined: from any given solution, a new solution can be generated by multiplying a column of $\mathbf{M}$ by some nonzero scalar and the corresponding row of $\mathbf{S}$ by the inverse of that scalar. These two indeterminacies are common in BSS problems, and are known as the permutation and scaling indeterminacies. They are also present in the SSS problem.

In this paper, it is also necessary to deal with a third indeterminacy; unlike the former two, this one does not affect the result of source separation, but rather our model of the sources. If $\mathbf{z}$ and $\mathbf{f}$ are a factorization of Φ, then so are $e^{\mathrm{i}\gamma}\mathbf{z}$ and $e^{-\mathrm{i}\gamma}\mathbf{f}$, as noted earlier. We call this the rotation indeterminacy. Note that all values for γ yield exactly the same Φ, and thus the same sources $\mathbf{S}$; nevertheless, since we explicitly model the sources using $\mathbf{z}$ and $\mathbf{f}$, we still have to consider this indeterminacy.

III. Algorithm

We now present PLMF, an algorithm to solve the SSS problem. The method is presented in three parts: preprocessing, which is covered in Section III-A, and two successive estimation procedures (which we call subproblems), which are covered in Sections III-B and III-C. These two sections also present identifiability theorems for each subproblem, which

[3] Multiplying $\mathbf{z}$ by a complex number of the form $e^{\mathrm{i}\gamma}$ (where γ is any real number), and multiplying $\mathbf{f}$ by $e^{-\mathrm{i}\gamma}$ yields the same matrix Φ. See also Section II-E.

are then merged into a global theorem in III-D. A summary is presented in Section III-E.

We assume that the number of sources, N, is known. PLMF tackles the SSS problem by solving

$$\begin{aligned} \min_{\mathbf{M},\mathbf{A},\mathbf{z},\mathbf{f}} \ & \frac{1}{2}\|\mathbf{Y}-\mathbf{M}\mathbf{D}_{\mathbf{z}}\mathbf{A}\mathbf{D}_{\mathbf{f}}\|_F^2 \qquad (5)\\ \text{s.t.: } & 1)\max_{i,j}|m_{ij}|=1\\ & 2)|\mathbf{z}_j|=1 \text{ for all } j\\ & 3)|\mathbf{f}_t|=1 \text{ for all } t \end{aligned}$$

where $\|\cdot\|_F$ is the Frobenius norm and s.t., which literally means subject to, indicates the list of constraints on the variables. The first constraint forces the largest absolute value among all elements of $\mathbf{M}$ to be 1. The second and third constraints force $\mathbf{z}$ and $\mathbf{f}$ to have entries with unit absolute value. $\mathbf{M}$ and $\mathbf{A}$ are real variables, while $\mathbf{z}$ and $\mathbf{f}$ are complex ones. This choice of variables and constraints follows from our model of the sources, as explained in Section II-D. This is a nonconvex problem [32] due to the presence of a product of several variables in the cost function and of nonconvex constraints. Nonconvex problems can have several optima and, in fact, this problem has multiple global optima because of the indeterminacies mentioned in Section II-E. One of the results in this paper is that, under certain conditions, all global optima of this problem correspond to desirable solutions.

Our assumption of real $\mathbf{M}$ is a consequence of assuming an instantaneous mixture of the sources, as motivated in Section I: if the mixture is instantaneous, the sources are present in each mixed signal without any phase change due to delay, thus the mixing matrix is real. Also, note that while our model for the sources from Section II-D results in a matrix $\mathbf{A}$ with nonnegative entries, PLMF does not explicitly force that assumption. As we shall observe, even if the sources follow the model from Section II-D with nonnegative amplitudes, it is not necessary to impose this constraint on the optimization problem.

To solve the minimization problem in (6), we use a two-stage approach. In the first stage, or subproblem, (Section III-B) we find $\mathbf{f}$, by solving a relaxed version of problem (6). In the second subproblem (Section III-C), we use the estimated $\mathbf{f}$, keep it fixed, and solve (6) relative to $\mathbf{M}$, $\mathbf{A}$ and $\mathbf{z}$.[4] This approach is motivated by theoretical results, presented ahead, which show that both subproblems are identifiable; in other words, all their global minima correspond to correct solutions. Each of these subproblems is tackled with the block nonlinear Gauss–Seidel (BNGS) method [33], as we discuss below.

Prior to that, however, we will discuss prewhitening, an important preprocessing step, which is useful in several BSS problems, including SSS.

A. Prewhitening

It is well known that the difficulty of solving inverse problems such as ICA and SSS can be approximately characterized by the condition number of the mixing matrix [34].[5] The condition number of a matrix $\mathbf{M}$ is defined as the ratio $\rho = \sigma_{\max}/\sigma_{\min}$, where $\sigma_{\max}$ is the largest singular value of $\mathbf{M}$ and $\sigma_{\min}$ is the smallest singular value.[6] The condition number obeys $\rho \geq 1$ for any matrix. Problems with a lower ρ are, in general, easier than problems with a higher ρ, even though this number does not fully characterize the difficulty of these problems [34].

The condition number of a BSS problem depends on the unknown matrix $\mathbf{M}$. In ICA without additive noise, after prewhitening, the mixing matrix has $\rho = 1$ [6]; therefore, whitening is often used there as a preprocessing step. In SSS, prewhitening does not guarantee that the mixing matrix will have a condition number of 1. However, we will prove that, under certain assumptions, an upper bound for this condition number exists if prewhitening is performed; we will also show empirical evidence of the benefits of prewhitening.

We begin by noting that the usual way to perform whitening involves computing the (empirical) covariance matrix of the data,[7] given by the square $P \times P$ matrix $\mathbf{C}_{\mathbf{Y}} \equiv 1/T\mathbf{Y}\mathbf{Y}^H$. Usually, prewhitening involves multiplying the data $\mathbf{Y}$ on the left by a matrix, which we define as

$$\mathbf{B} \equiv \mathbf{D}^{-\frac{1}{2}}\mathbf{V}^H \qquad (6)$$

where $\mathbf{D}$ is a $N \times N$ diagonal matrix containing the nonzero eigenvalues of $\mathbf{C}_{\mathbf{Y}}$ in its diagonal, $\mathbf{V}$ is a $P \times N$ matrix with the corresponding eigenvectors in its columns, and $(\cdot)^H$ denotes the conjugate transpose of a matrix.

Multiplying both sides of the equation $\mathbf{Y} = \mathbf{MS}$, on the left, by this matrix transforms the original source separation problem $\mathbf{Y} = \mathbf{MS}$ into a new problem $\mathbf{BY} = \mathbf{BMS}$, where $\mathbf{BY}$ can be interpreted as new data, and $\mathbf{BM}$ as a new mixing matrix. If this new problem is solved, we obtain an estimate of the original sources and an estimate of the product $\mathbf{BM}$. While it would be possible to subsequently estimate $\mathbf{M}$ itself, in this paper, we are concerned only with recovering the original sources, and thus a good estimation of the product $\mathbf{BM}$ will suffice.

In the problem considered in this paper, the mixing matrix $\mathbf{M}$ is real but the data $\mathbf{Y}$ are complex. Therefore, the equivalent mixing matrix $\mathbf{BM}$ is, in general, complex. Thus, without whitening, one is searching for a real $P \times N$ mixing matrix; with whitening one has to search for a complex $N \times N$ mixing matrix. We now show how one can transform this into a search for a real $N \times N$ mixing matrix.

We split the data $\mathbf{Y}$ into its real part $\mathbf{Y}_R \equiv \text{real}(\mathbf{Y})$ and its imaginary part $\mathbf{Y}_I \equiv \text{imag}(\mathbf{Y})$, and define $\mathbf{S}_R$ and $\mathbf{S}_I$ in a

[4] This contrasts with our original PLMF approach presented in [29]. In that work, we optimized the cost function (6) on all four variables simultaneously. We refer to the approach from [29] as one-stage PLMF. Section IV shows a comparison between that approach and the one presented in this paper.

[5] While essentially all linear inverse problems involve a matrix whose function is similar to the function of our mixing matrix, in many cases it does not correspond to a mixing of signals and therefore is not called mixing matrix. We still call it mixing matrix for brevity.

[6] Other definitions of condition number exist. The one presented here is quite common, and will be used throughout the paper.

[7] In BSS problems it is common to start by removing the mean from the data. Our data has a mean value of zero, by construction; therefore, we will assume that the data has zero mean from now on.

similar way for the sources $\mathbf{S}$ [35]. Since $\mathbf{M}$ is real, the initial complex problem $\mathbf{Y} = \mathbf{MS}$ can be turned into an equivalent real problem in two different ways

$$\begin{bmatrix}\mathbf{Y}_R\\ \mathbf{Y}_I\end{bmatrix} = \begin{bmatrix}\mathbf{M}\ \mathbf{0}\\ \mathbf{0}\ \mathbf{M}\end{bmatrix}\begin{bmatrix}\mathbf{S}_R\\ \mathbf{S}_I\end{bmatrix} \quad \text{or} \quad [\mathbf{Y}_R\ \mathbf{Y}_I] = \mathbf{M}\,[\mathbf{S}_R\ \mathbf{S}_I]\,. \tag{7}$$

We call the first formulation the vertically stacked form (VS form) and the second one the horizontally stacked form (HS form). Clearly, any of these two formulations is equivalent to the original one, in the sense that a solution for either of them is immediately transformable into a solution for the original problem.

One can apply the whitening procedure to the left-hand side of either the VS form or the HS form, which are now real. Both of these would yield the same upper bound for the condition number of the equivalent mixing matrix in the Theorem that follows. We have empirically found that the condition number of the equivalent mixing matrix is, on average, farther from the upper bound presented ahead (and thus, better conditioned) if the HS form is used. Therefore, we focus on that formulation only.

The upper bound for the condition number of the mixing matrix after whitening is given by the following theorem.

Theorem 1: Let $\mathbf{S}_{\text{RI}} \equiv [\mathbf{S}_R\ \mathbf{S}_I]$ and $\mathbf{Y}_{\text{RI}} \equiv [\mathbf{Y}_R\ \mathbf{Y}_I]$. Let $\mathbf{B}$ be the result of applying the procedure from (6) to $\mathbf{Y}_{\text{RI}}$. Furthermore, suppose that the following assumptions hold.

1) $\mathbf{M}$ and $\mathbf{S}$ both have maximum rank.
2) There is no additive noise; thus, $\mathbf{Y} = \mathbf{MS}$ holds.
3) The amplitudes of each source, $a_j(t)$, are independent identically distributed (i.i.d.) realizations of a random variable, which we denote by A_j.
4) A_j is independent of A_k for $j \neq k$.
5) A_j is independent of ϕ_k for any j and k, including $j = k$.
6) All A_j have the same distribution (we denote by A a random variable with that distribution).
7) ϕ_j and ϕ_k have maximum PLF, i.e., they have a constant phase lag; this implies that there exists $\phi(t)$, independent of j, such that $\phi_j(t) = \phi_j(1) + \phi(t)$ for all j and t.
8) The random sequence $\phi(t)$ is uniformly distributed in $[0, 2\pi)$; note, however, that this sequence does not need to be i.i.d. on that interval.

Then, the condition number of the equivalent mixing matrix, denoted by $\rho(\mathbf{BM})$, obeys

$$\rho(\mathbf{BM}) \leq \sqrt{1 + N\frac{\mathbb{E}[A]^2}{\text{Var}[A]}} \tag{8}$$

where N is the number of sources, $\mathbb{E}[\cdot]$ is the expected value operator and Var$[\cdot]$ is the variance operator. Furthermore, this upper bound is tight, meaning that in some cases (8) holds with equality.

Proof: See Appendix A. ■

Intuitively, what this theorem says is that, in the absence of noise and after prewhitening, the difficulty of the problem is bounded above by (8).

In practice, the assumptions of this theorem are very restrictive. However, we have empirically found that even with MEG-like data, where the assumptions of the theorem are far from being true (in particular, due to the presence of noise), whitening strongly improves the conditioning of the SSS problem, and consequently improves the performance of separation algorithms [36]. In the beginning of Section IV-B, we present empirical results, which corroborate and illustrate this theorem.

To keep the notation simple, from now on, except where noted, we assume that prewhitening has been performed; thus, we take the prewhitened data $\mathbf{BY}$ as new data, and we designate these new data simply as $\mathbf{Y}$ and the equivalent mixing matrix $\mathbf{BM}$ simply as $\mathbf{M}$.

B. Estimation of f

The first stage of PLMF estimates the common oscillation $\mathbf{f}$. We perform this estimation by solving the subproblem

$$\begin{aligned} &\min_{\mathbf{H},\mathbf{A},\mathbf{f}} \frac{1}{2}\,\|\mathbf{Y} - \mathbf{HAD_f}\|_F^2 \\ &\text{s.t.: } 1)\max_{i,j}|h_{ij}| = 1 \\ &\qquad 2)|\mathbf{f}_t| = 1 \text{ for all } t. \end{aligned} \tag{9}$$

$\mathbf{H}$ is any complex matrix with the same dimensions as $\mathbf{M}$, and the largest absolute value among its entries must be 1; $\mathbf{f}$ is complex with entries having unit absolute value, as before. This formulation collapses the product $\mathbf{MD_z}$ into the matrix $\mathbf{H}$, which is now allowed to be any complex matrix. This relaxation[8] means that the subproblem in (10) is easier to tackle than the original problem in (6). If the sources exactly follow the model in (4), a factorization of the form $\mathbf{Y} = \mathbf{HAD_f}$ always exists, since the true factorization is a special case of it.

It is important to remark that the goal of this first subproblem is to estimate $\mathbf{f}$; even though a solution of (10) will also yield estimates for $\mathbf{H}$ and $\mathbf{A}$, these are discarded at the end of this first stage.

We now show that if $\mathbf{Y} = \mathbf{HAD_f}$, then $\mathbf{f}$ is correctly estimated through the minimization in (10), apart from a sign indeterminacy (which can be easily compensated, as discussed below), and from the rotation indeterminacy, which was already discussed in Section II-E.

Theorem 2 (Quasi-identifiability of $\mathbf{f}$*):* Let $\mathbf{Y} = \mathbf{H}_1\mathbf{A}_1\mathbf{D}_{\mathbf{f}1}$ with $\mathbf{H}_1 \in \mathbb{C}^{P\times N}$, $\mathbf{A}_1 \in \mathbb{R}^{N\times T}$, $\mathbf{D}_{\mathbf{f}1} \in \mathbb{D}_1^T$, where $\mathbb{D}_1^T$ is the set of T-by-T diagonal matrices whose diagonal entries have unit absolute value, and $\mathbf{H}_1$ has full column rank. If there is another factorization of the same form, $\mathbf{Y} = \mathbf{H}_2\mathbf{A}_2\mathbf{D}_{\mathbf{f}2}$, then necessarily one has $\mathbf{D}_{\mathbf{f}2} = \mathbf{E}\mathbf{D}_{\mathbf{f}1}$ where $\mathbf{E} \in \mathbb{D}_1^T$ is a diagonal matrix whose diagonal elements belong to the two-element set $\{-e^{i\gamma}, +e^{i\gamma}\}$, where γ is a real number.

Proof: See Appendix B. ■

The previous theorem only ensures a quasi-identifiability of $\mathbf{f}$, since $\mathbf{D_f}$ is determined up to multiplication by matrix $\mathbf{E}$. Note that it is not necessary to determine the value of γ, which corresponds to the rotation indeterminacy, since we will subsequently estimate $\mathbf{z}$ which will compensate this indeterminacy.

[8] Recall that $\mathbf{M}$ can be any real matrix and $\mathbf{D_z}$ is a complex diagonal matrix whose diagonal elements have unit absolute value. It is easy to verify that the product $\mathbf{MD_z}$ does not span the space of all possible complex matrices $\mathbf{H}$. Therefore, this is a relaxed version of the original problem (6).

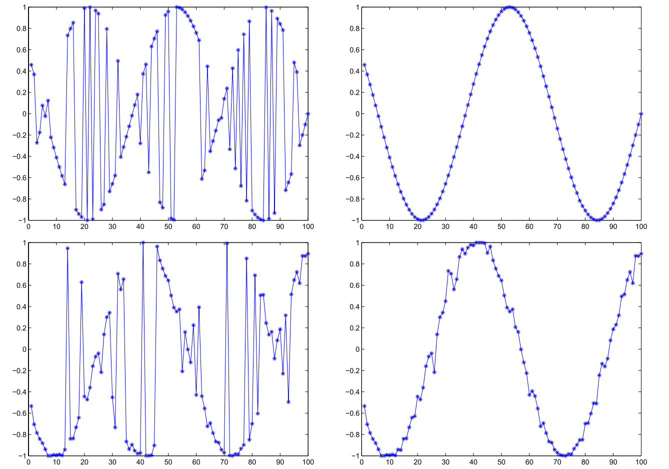

Fig. 2. Typical result of the phase correction procedure described at the end of Section III-B. Top row: results for noiseless data. We show the real part of **f** before the correction is made (left) and the real part of **f** after that correction (right). Bottom row: similar to the top row but using data with additive noise. In both cases, the phase jumps are all corrected.

Due to the phase indeterminacy, for some $\mathbf{A}$, there are multiple $(\mathbf{z}, \mathbf{f})$ pairs which yield the same sources. This theorem, and the one in the following section, ensure that even though we estimate $\mathbf{f}$ first and $\mathbf{z}$ later, we end up with a correct pair.

It is thus only necessary to estimate which diagonal elements of $\mathbf{E}$ are equal to $-e^{i\gamma}$ and which are equal to $+e^{i\gamma}$. This sign estimation is easy to perform if $\mathbf{f}$ varies smoothly with time; in our case, we simply compute, for $t = 1, \ldots, T-1$, the quantity

$$|f_R(t) - f_R(t+1)| + |f_I(t) - f_I(t+1)| \qquad (10)$$

where $f_R(t)$ is the real part of the tth entry of $\mathbf{f}$, and $f_I(t)$ is the imaginary part of that entry. It is easy to show that, if $f(t+1) = -f(t)$, then this quantity lies between $\sqrt{2}$ and 2. If $\mathbf{f}$ varies smoothly, we expect the values of (10) to be small if there is no change of sign from time t to time $(t+1)$, and to be $\gtrapprox\sqrt{2}$ if such a sign change occurs from t to $(t+1)$. In our algorithm, we determine that there is a change in sign when

$$|f_R(t) - f_R(t+1)| + |f_I(t) - f_I(t+1)| > 1. \qquad (11)$$

We have empirically verified that this simple procedure captures all sign changes in the data used in this paper. However, for other types of data, one may need to use better phase unwrapping techniques [37]. Typical results of this procedure are shown in Fig. 2 for noiseless and noisy data; note that all discontinuities have been detected and corrected.

C. Estimation of $\mathbf{M}$, $\mathbf{A}$, *and* $\mathbf{z}$

After estimating $\mathbf{f}$ (up to rotation), the original problem in (6) reduces to a second subproblem

$$\min_{\mathbf{M},\mathbf{A},\mathbf{z}} \frac{1}{2}\left\|\mathbf{Y}\mathbf{D}_{\mathbf{f}}^{*} - \mathbf{M}\mathbf{D}_{\mathbf{z}}\mathbf{A}\right\|_F^2 \qquad (12)$$
$$\text{s.t.: 1) } \max_{i,j} |m_{ij}| = 1$$
$$\text{2)} |\mathbf{z}_j| = 1 \text{ for all } j$$

(note that $\mathbf{D}_{\mathbf{f}}^{*} = \mathbf{D}_{\mathbf{f}}^{-1}$). Constraints should be interpreted as in the original formulation (6).

This problem is again easier than the original one, since one of the variables $(\mathbf{f})$ is already estimated. Importantly, one again has identifiability in this second subproblem, as we now show.

Theorem 3 (Identifiability of $\mathbf{M}$,$\mathbf{A}$,$\mathbf{z}$*):* Let $\mathbf{Y}\mathbf{D}_{\mathbf{f}}^{*} = \mathbf{M}_1\mathbf{D}_{\mathbf{z}1}\mathbf{A}_1$ with $\mathbf{M}_1 \in \mathbb{R}^{P\times N}$, $\mathbf{D}_{\mathbf{z}1} \in \mathbb{D}_1^N$, $\mathbf{A}_1 \in \mathbb{R}^{N\times T}$, where $\mathbb{R}^{N\times T}$ denotes the set of N-by-T matrices with real entries. Furthermore, assume that the phases of all sources are different from one another modulo π (in other words, two entries $e^{i\alpha}$ and $e^{i\beta}$ of the diagonal of $\mathbf{D}_{\mathbf{z}1}$ never satisfy $e^{i\alpha} = e^{i\beta}$ nor $e^{i\alpha} = -e^{i\beta}$), and that $\mathbf{A}_1$ has maximum row rank. If there is another factorization of the same form, $\mathbf{Y}\mathbf{D}_{\mathbf{f}}^{*} = \mathbf{M}_2\mathbf{D}_{\mathbf{z}2}\mathbf{A}_2$, then one necessarily has $\mathbf{M}_1 = \mathbf{M}_2$, $\mathbf{D}_{\mathbf{z}1} = \mathbf{D}_{\mathbf{z}2}$, and $\mathbf{A}_1 = \mathbf{A}_2$ (up to permutation, scaling, and sign).

Proof: See Appendix C. ■

The previous theorem assumes that all the arguments of the entries in the diagonal of $\mathbf{D}_{\mathbf{z}}$ are different modulo π. A similar theorem can be proven for a more general case where k diagonal elements violate this assumption, whereas the remaining $(N-k)$ obey it. In that case, $\mathbf{D}_{\mathbf{z}}$ is still identifiable. However, only $(N-k)$ rows of $\mathbf{A}$ and the corresponding $(N-k)$-by-$(N-k)$ block of $\mathbf{M}$ are identifiable. In other words, only the $(N-k)$ sources with distinct phase values (modulo π) are identifiable; the remaining sources will, in general, be mixed with one another in the estimated sources. Due to lack of space, we do not present a proof of this generalization.[9]

D. Global Identifiability

If the data are generated according to (4) with nonnegative amplitudes (i.e. $\mathbf{A}$ has nonnegative entries), Theorems 2 and 3 can be combined to produce an interesting result: the original PLMF problem, as stated in (6), has a unique solution with nonnegative amplitudes. This is stated in the next theorem.

Theorem 4: Let $\mathbf{Y}$ be data generated according to the model in (4) with nonnegative amplitudes, and let $\mathbf{Y} = \mathbf{M}_1\mathbf{D}_{\mathbf{z}_1}\mathbf{A}_1\mathbf{D}_{\mathbf{f}_1}$ be a factorization of the data such that the entries of $\mathbf{A}_1$ are nonnegative, the constraints of problem (6) are satisfied, $\mathbf{M}_1$ has full column rank, the phases of the entries of $\mathbf{z}_1$ are different modulo π, and $\mathbf{A}_1$ has maximum row rank. Let $\mathbf{Y} = \mathbf{M}_2\mathbf{D}_{\mathbf{z}_2}\mathbf{A}_2\mathbf{D}_{\mathbf{f}_2}$ be another such factorization. Then, the two factorizations are equal up to permutation, scaling, and rotation, as defined in Section II-E.

Proof: See Appendix D. ■

Given this theorem, and since in our experiments we will indeed use data generated using nonnegative amplitudes, it is relevant to clarify why we chose to split PLMF into two subproblems, even though the whole problem is identifiable. There are two reasons for this choice. The first one is that we empirically found that simultaneously estimating all four variables, as in [29], is more prone to getting trapped in local minima than the two-stage procedure presented in this paper. The second reason is that the proof of theorem 4

[9] A sketch of the proof is as follows: the previous derivation remains valid until (42) in the appendix; however, now only $(N-k)$ eigenvalues have multiplicities of 1, and identifiability holds for the sources corresponding to those eigenvalues only.

TABLE I

PLMF ALGORITHM

```
PHASE LOCKED MATRIX FACTORIZATION
1:  Given: data $\mathbf{Y}$
0:  WHITENING
2:  Whiten data $\mathbf{Y}$
I:  ESTIMATION OF $\mathbf{f}$
3:  for run $\in$ {1,2,...,MaxRuns$_{\mathbf{f}}$}, do
4:    Randomly initialize $\hat{\mathbf{H}}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{f}}$
5:    for iter $\in$ {1,2,...,MaxIter$_{\mathbf{f}}$}, do
6:      Solve minimization (9) for $\mathbf{H}$
7:      Solve minimization (9) for $\mathbf{A}$
8:      Solve minimization (9) for $\mathbf{f}$
9:    end for
10: end for
11: From the MaxRuns$_{\mathbf{f}}$ solutions, choose the one which yields
    the lowest value of the function being minimized in (9)
12: Store $\mathbf{f}$ and discard $\mathbf{H}$ and $\mathbf{A}$
13: Correct sign of $\mathbf{f}$ by detecting values of (10) greater than 1
II: ESTIMATION OF $\mathbf{M}$, $\mathbf{A}$, $\mathbf{z}$
14: for run $\in$ {1,2,...,MaxRuns$_{\mathbf{M},\mathbf{A},\mathbf{z}}$}, do
15:   Randomly initialize $\hat{\mathbf{M}}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{z}}$
16:   for iter $\in$ {1,2,...,MaxIter$_{\mathbf{M},\mathbf{A},\mathbf{z}}$}, do
17:     Solve problem (12) for $\mathbf{M}$
18:     Solve problem (12) for $\mathbf{A}$
19:     Solve problem (12) for $\mathbf{z}$
20:   end for
21: end for
22: From the MaxRuns$_{\mathbf{M},\mathbf{A},\mathbf{z}}$ solutions, choose the one
    which yields the lowest value of the function
    being minimized in eq. (12)
23: return $\mathbf{M}$,$\mathbf{A}$, $\mathbf{z}$, $\mathbf{f}$
```

itself suggests that one should split the problem into two subproblems.

E. Optimization Procedures

The PLMF algorithm is shown in Table I. We now explain in further detail how each of the two subproblems is tackled. We employ the BNGS [33] method in both optimizations; in the first subproblem, we randomly initialize the variables $\mathbf{H}$, $\mathbf{A}$ and $\mathbf{f}$, and iteratively optimize each of them while keeping all others fixed (lines 5–9 of Table I). Similarly, for the second subproblem, we initialize $\mathbf{M}$, $\mathbf{A}$ and $\mathbf{z}$ randomly and optimize each of them while keeping all others fixed (lines 16–20 of Table I). The use of BNGS has a great advantage: problems (10) and (13), which are hard to solve globally (in particular, due to the presence of products of variables), become an iteration of simpler problems (constrained least-squares problems). There is a downside: BNGS is not guaranteed to converge to an optimal solution in the general case. We discuss this aspect further in Section V.

The two subproblems (10) and (13) are convex in some variables and nonconvex in other variables. Instead of trying to find the global minimum at each iteration, we chose to always solve for each variable without enforcing any constraints, then projecting that solution onto the feasible set; this projection is an approximation of the true solution. Our choice is motivated for two reasons: simplicity, because this way all variables are optimized in a similar way; and speed, which allowed us to run the extensive experiments shown in Section IV. Note that, while this is a suboptimal procedure, the fact that the two subproblems are nonconvex in some variables prevents us from having a guaranteed optimal solution anyway.

Each iteration of the Gauss–Seidel method simply involves solving an unconstrained least-squares problem, which we solve using the Moore–Penrose pseudoinverse. After finding the solution of the unconstrained problem that solution is projected into the space of feasible solutions. For example, in the first subproblem, solving for $\mathbf{H}$ (line 6) is done without considering the first constraint of (10). After the unconstrained solution is found, $\mathbf{H}$ is multiplied by a scalar such that the largest absolute value of its elements is exactly 1. All variables, in both subproblems, are handled in a similar manner.

We use the value of the cost functions of problems (10) and (13) as imperfect indicators of the goodness of a solution. For this reason, each subproblem is solved multiple times for given data $\mathbf{Y}$; we then keep only the solution, which yielded the lowest cost value for that subproblem (lines 11 and 22).

IV. EXPERIMENTAL RESULT

A. Data Generation

We use a noisy variant of the source model in (4) to generate the data. This variant accommodates two deviations from the noiseless case: the presence of additive noise and of phase jitter. The model used to generate the data is

$$\mathbf{Y} \equiv \mathbf{M}(\mathbf{A} \odot (\mathbf{z}\mathbf{f}^T) \odot \mathbf{J}) + \mathbf{N} \tag{13}$$

where $\mathbf{J}$ is a $N \times T$ matrix of complex values with unit absolute value, representing phase jitter, and $\mathbf{N}$ is a $P \times T$ matrix of complex values representing additive channel noise. If all entries of $\mathbf{J}$ are equal to 1 and all entries of $\mathbf{N}$ are equal to zero, we recover the noiseless model of (4).

We generate 1000 data sets for each set of parameters that we study. For each data set, the mixing matrix $\mathbf{M}$ is randomly generated, with each entry uniformly distributed between -1 and 1, the vector of phase lags $\mathbf{z}$ is generated as $[0, \Delta\phi, \ldots, (N-1)\Delta\phi]^T$ ($\Delta\phi$ is determined below), and the common oscillation $\mathbf{f}$ is generated as a sinusoid: $\mathbf{f} = [0, \exp(\mathrm{i}\Delta t), \exp(\mathrm{i}2\Delta t), \ldots, \exp(\mathrm{i}(T-1)\Delta t)]$, with $T = 100$ and $\Delta t = 0.1$. While this is a very specific choice (a phase which grows linearly with time), it is representative of the smoothly varying $\mathbf{f}$ case, which is treated in this paper. We have empirically verified that PLMF works well with other choices for $\mathbf{f}$ as long as they are smoothly varying (otherwise, the correction of phase jumps, mentioned at the end of Section III-B, becomes unreliable).

The amplitude $\mathbf{A}$ is generated as the result of low-pass filtering a Gaussian white noise signal. This is appropriate for situations where the amplitudes $\mathbf{A}$ are expected to vary slower than the phase oscillations $\mathbf{f}$. Specifically, we begin by generating random Gaussian white noise of length T. We then take the discrete cosine transform (DCT) of that signal, keep only the 10% of coefficients corresponding to the lowest frequencies, and take the inverse DCT of the result. We then add a constant to this filtered signal to ensure that it is nonnegative,[10] and the result becomes $\mathbf{a}_1(t)$, the first row of $\mathbf{A}$. The process is then repeated, with different random initializations, for each source in succession.

An example of signals generated in this manner is shown in Fig. 3, where we present an extended time period ($T = 500$) to better illustrate the structure of the signals.

[10] While the algorithm presented in this paper does not require positive amplitudes, we will compare it with other algorithms which do require this assumption.

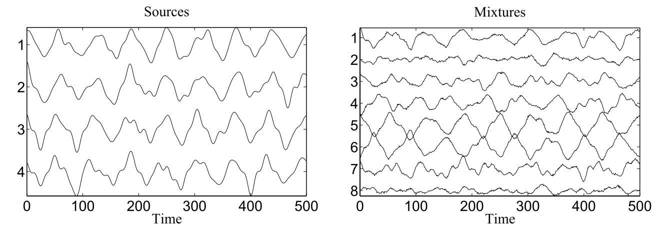

Fig. 3. Left: the real part of a typical set of four sources generated, as described in Section IV-A, with no phase jitter. Right: the real part of a corresponding set of eight mixtures, with an input SNR of 20 dB. Note that in most of the following experiments, only 100 points are used.

We study the effect of the following variables.

1) Additive noise $\mathbf{N}$, as measured by the signal-to-noise ratio (SNR) of each channel. The energy of the noise in each channel is generated such that all channels have the same SNR, which is called the input SNR. We study the cases of an SNR of 80, 60, 40, 20, and 0 dB.
2) Phase jitter $\mathbf{J}$. We study two types of jitter.
    a) The first case is jitter where each entry of $\mathbf{J}$ is of the form $e^{i\delta}$, where δ is independently drawn from a Gaussian distribution with zero mean and standard deviation σ_{iid}. We study the cases of $\sigma_{\text{iid}} = 0, 0.02, 0.04, \ldots, 0.1$. We name this i.i.d. jitter, since the jitter for time t and for source k is independent from the jitter at any other entry of $\mathbf{J}$.
    b) The second case is called correlated jitter. We generate a matrix $\mathbf{Q}$ in a similar manner as the amplitude $\mathbf{A}$, except that positivity is not enforced, and that we keep the lowest 2% of coefficients of the DCT, instead of the lowest 10%. This yields a very slowly varying signal. We then generate the jitter $\mathbf{J}$ as $e^{i\mathbf{Q}}$, where this exponential is taken element wise. This results in a jitter, which is slow-varying. Due to the finite observation time T, this jitter is also correlated from one source to another. In the context of correlated jitter, we will use the symbol σ_{corr} to denote the standard deviation of the Gaussian white noise used in the generation of the jitter.
3) Phase lag $\Delta\phi$. We study the cases of $\Delta\phi = \pi/50$, $2\pi/50, \ldots$, up to $12\pi/50$.
4) Number of sources N and number of sensors P. We study the cases $N = 2, 4, \ldots, 10$, with $P = N$ and with $P = 2N$.
5) Number of time samples T. We study the values $T = 100, 200, 400, 800$.

It would be extremely cumbersome to compute and show results for all possible combinations of the above variables. To avoid this while still studying all variables, we study a central case where PLMF performs very well, and then change the above variables, one at a time. In total, we study 64 different cases. The central case has $N = 4$ sources, $P = 8$ sensors, $T = 100$ time samples, an input SNR of 80 dB, no jitter, and a phase lag of $\Delta\phi = \pi/10$.

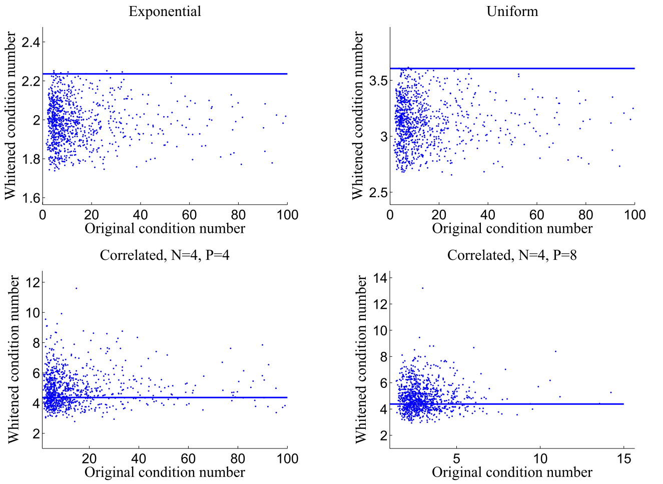

Fig. 4. Comparison of the condition number of the original mixing matrix, $\rho(\mathbf{M})$ (horizontal axis), with that of the equivalent mixing matrix after whitening, $\rho(\mathbf{BM})$ (vertical axis). Top left: results for data sets, which follow the assumptions of theorem 1, and for which each entry of $\mathbf{A}$ is i.i.d. drawn from an exponential distribution. Top right: same, but each entry of $\mathbf{A}$ is i.i.d. drawn from a uniform distribution. Bottom left: results for data sets generated as in Section IV-A, with $N = 4$ and $P = 4$. In this case, the horizontal axis was truncated for clarity, since the maximum value on that axis would be approximately 10^4 without truncating. Bottom right: same, but with $P = 8$. No truncation was necessary in this case.

B. Results

1) Effect of Whitening: We begin by empirically confirming Theorem 1. For this, we compute the condition number of $\mathbf{M}$ before whitening, $\rho(\mathbf{M})$, and after whitening, $\rho(\mathbf{BM})$, for the data described in Section IV-A. We use $N = 4$ and study two situations: $P = 4$ and $P = 8$. Note that these data sets grossly violate the first assumption of Theorem 1, since the different time points in each source's amplitude are not i.i.d. These slow-varying amplitudes are closer to what is observed in brain signals, so we study them nevertheless. For these data sets, we empirically compute $\mathbb{E}[A]^2$ and $\text{Var}[A]$ to compute the value of the bound.

We also generate 1000 data sets, similar to those of the previous paragraph with $P = 4$, but where each entry of $\mathbf{A}$ is drawn independently from an exponential distribution. Finally, we also generate 1000 more data sets where each time sample of $\mathbf{A}$ is drawn independently from a uniform distribution. These data sets have $T = 10000$, to ensure that the sample covariance matrix is very similar to the true one. These data sets verify all the assumptions of theorem 1, and are presented for comparison with those of the previous paragraph. In these two cases, we analytically compute $\mathbb{E}[A]^2$ and $\text{Var}[A]$ for the exponential and uniform distribution.

Fig. 4 shows the results for these four types of data sets. For each figure, each point is plotted in position $(\rho(\mathbf{M}), \rho(\mathbf{BM}))$. Each figure also shows the theoretical value of the upper bound as a horizontal line. The results for the top row show that, when the data follows the theorem assumptions, the upper bound is correct. While it may appear unexpected that a few points are above the upper bound, this is justified by the difference between the ideal case of $T = \infty$, which was used in Theorem 4 to derive the bound, and the simulated case of finite T, where we use the sample covariance matrix instead of the true covariance matrix.

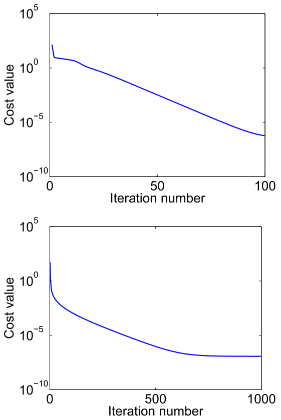

Fig. 5. Influence of whitening on the convergence of PLMF. The top row shows the evolution of the cost function of the first subproblem (10) as a function of the number of iterations, without whitening (left) and with whitening (right). The top-left and top-right subfigures used the exact same data. The bottom row shows the same for the second subproblem [equation (13)]; again, both subfigures used the same data. Clearly, whitening improves the speed of convergence.

The results for the bottom row show that, for the more realistic data studied below, the upper bound is not correct. This does not contradict theorem 1, since these data sets do not obey its hypothesis. These results also show a relevant difference between the cases $P = 4$ and $P = 8$: in the former, whitening typically yields a very large decrease in condition number, whereas in the latter it yields no significant change on average.[11]

2) Results of PLMF: In all the results discussed in this section, $\text{MaxRuns}_{\mathbf{f}} = \text{MaxRuns}_{\mathbf{M},\mathbf{A},\mathbf{z}} = 5$. In other words, we solved each subproblem five times and kept only the solution which yielded the lowest cost value. We used $\text{MaxIter}_{\mathbf{f}} = 100$ on the first subproblem and $\text{MaxIter}_{\mathbf{M},\mathbf{A},\mathbf{z}} = 1000$ on the second one.

We start by assessing the usefulness of prewhitening by comparing how PLMF behaves when it is or is not used. Fig. 5 shows a typical result of this comparison. On both subproblems, the number of iterations required until convergence decreases if whitening is performed.

We measure the separation quality using the estimated equivalent mixing matrix, which in this section is denoted as $\mathbf{B}\hat{\mathbf{M}}$ to distinguish it from the true equivalent mixing matrix $\mathbf{BM}$.[12] We begin by computing the gain matrix, defined as $\mathbf{G} \equiv (\mathbf{B}\hat{\mathbf{M}})^{\dagger}\mathbf{BM}$, where † denotes the Moore–Penrose pseudoinverse of a matrix. The gain matrix is always square, of size N by N. If the separation was perfect, $\mathbf{G}$ should be a permutation of a diagonal matrix; we undo this permutation using the knowledge of the true mixing matrix before computing the following measure. Let g_{ij} denote the (i, j) element of $\mathbf{G}$. Our quality measure, which we term simply separation quality, is defined as

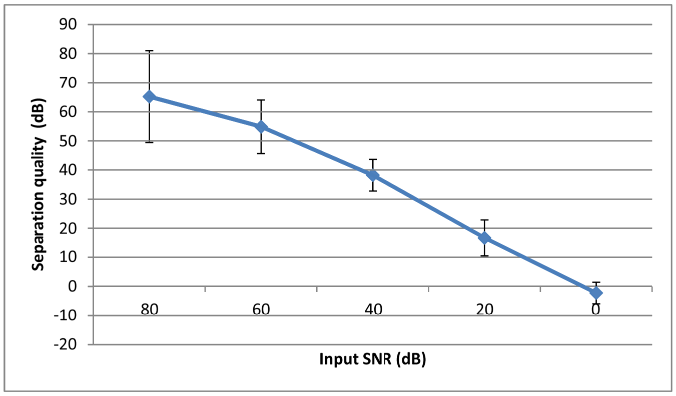

Fig. 6. Separation quality versus input SNR. Under heavy noise, PLMF can recover the sources with about as much noise as they had in the input. Error bars correspond to one standard deviation.

$$Q \equiv 10 \log \left(\frac{\sum_j g_{jj}^2}{\sum_{i \neq j} g_{ij}^2} \right). \tag{14}$$

In words, we sum the squares of the diagonal elements of $\mathbf{G}$ and divide that value by the sum of the squares of its nondiagonal elements. The $10 \log(.)$ function allows us to express that quotient in dB, a unit typically used for the SNR.

If the sources were orthogonal and all had the same energy, Q would measure the SNR. However, this is not the case: while our sources have the same energy, they are not uncorrelated. In fact, for very small phase lag, the correlation factor between pairs of sources is usually quite high (as shown in Fig. 3). Despite this, we present values of Q in dB to allow an easier interpretation.

We also compute the quality measure for each source separately, given by

$$Q_j \equiv 10 \log \left(\frac{g_{jj}^2}{\sum_{i \neq j} g_{ij}^2} \right). \tag{15}$$

Here, the numerator is the square of the (j, j) entry of $\mathbf{G}$, while the denominator is the sum of the squares of all other elements in the jth column. In every figure conveying results of our experiments, each point in the figure corresponds to the average value of Q (or Q_j) among the 1000 runs for those experimental conditions, and the error bars represent the value of that average plus/minus one standard deviation.

As a final remark, note that we could also compute a quality measure that depends on the sources, i.e., on the variables $\mathbf{A}$, $\mathbf{z}$ and $\mathbf{f}$ and their estimations. The quantity Q defined above has the advantage of having a simpler interpretation, because it depends only on $\mathbf{BM}$ and its estimation. Finally, note that if $\mathbf{B}\hat{\mathbf{M}}$ is a good estimate of $\mathbf{BM}$, that allows us to recover a good estimate of the original sources, by directly computing $\hat{\mathbf{S}} \equiv (\mathbf{B}\hat{\mathbf{M}})^{\dagger}\mathbf{BY}$. If estimates of $\mathbf{A}$, $\mathbf{z}$ and $\mathbf{f}$ are desired, those can easily be obtained from $\hat{\mathbf{S}}$.

Fig. 6 shows how the average separation quality varies when the input SNR changes from 80 dB, which is a case of virtually no noise, to 0 dB, which corresponds to very strong noise. It can be seen that PLMF performs very well: it yields results

[11] There is a simple explanation for this behavior. $\mathbf{M}$ has size (P, N), and its entries are mutually independent and are drawn from centered variables. Due to this, its columns tend to become orthogonal as $P \to \infty$, and therefore increasing P tends to make the singular values of $\mathbf{M}$ very similar. This is why doubling P makes the typical values of $\rho(\mathbf{M})$ much lower in the bottom row of Fig. 4.

[12] As stated in Section III-A, estimating the equivalent mixing matrix is enough if we are interested in recovering the original sources, and not the mixing matrix.

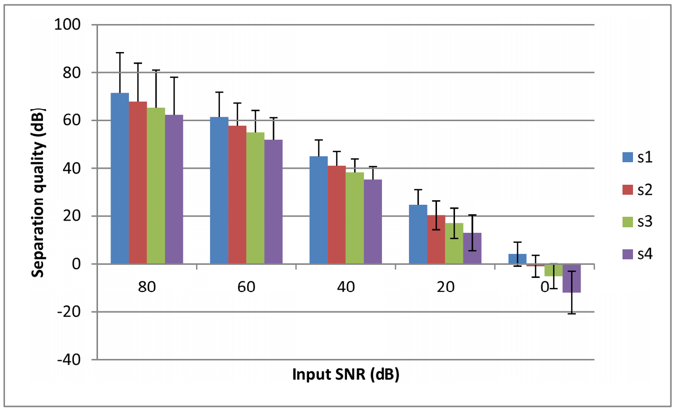

Fig. 7. Separation quality per source versus input SNR. In each group of bars, s_1 represents mean and standard deviation of the SNR of the best-estimated source for the 1000 runs of PLMF. s_2 represents the same for the second-best-estimated source, and so on. This graph illustrates that all sources are estimated with a roughly similar level of quality.

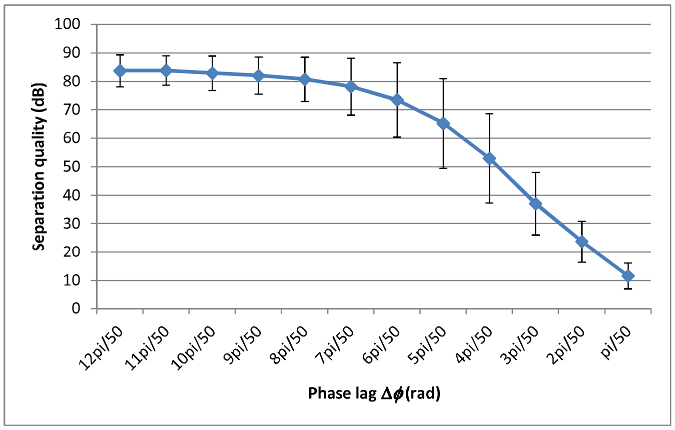

Fig. 8. Separation quality versus phase lag. PLMFs results are, in general, good, but they deteriorate progressively as one approaches the case where $\Delta\phi = 0$, where theorem 3 fails to hold.

with a separation quality, which is only 2–3 dB below the input SNR, except for low-noise cases (input SNR of 80 and 60 dB), in which the separation quality is nevertheless very good (65 and 55 dB, respectively).

The separation quality per source is shown in Fig. 7. It can be observed that the average difference between the best estimated source and the worst estimated one is around 7–9 dB. This behavior is consistent through all the simulations in this paper. For this reason, and due to lack of space, we shall not present any further per-source results.

Fig. 8 shows how the separation quality varies with the phase lag $\Delta\phi$. For most values of this parameter, the separation quality is very high. However, the separation quality becomes progressively lower when $\Delta\phi$ approaches zero, where the hypothesis of Theorem 3 fails to hold. Nevertheless, this deterioration is gradual and is only relevant for very small phase lags (smaller than $2\pi/50$, or 7.2 degrees, which yields a separation quality of 23.7 dB).

Fig. 9 shows the effect of varying the number of sources N and the number of sensors P. Generally, the quality of the results decreases with increasing N, which is expected since the size of the problem variables **M**, **A** and **z** increases. When there is very little noise (input SNR of 80 dB), there is a little benefit in doubling the number of sensors from $P = N$ to $P = 2N$. However, when there is considerable noise (input SNR of 20 dB), that benefit becomes significant, especially for $P = 4, 6, 8$ where the improvement exceeds 10 dB.

Figs. 10 and 11 show the results with various levels of i.i.d. and correlated phase jitter, respectively. At first glance, it appears that either type of jitter deteriorates the results. However, there are fundamental differences between these two types of noises, which render the i.i.d. version less damaging for source estimation than the correlated version, as will be discussed in Section V.

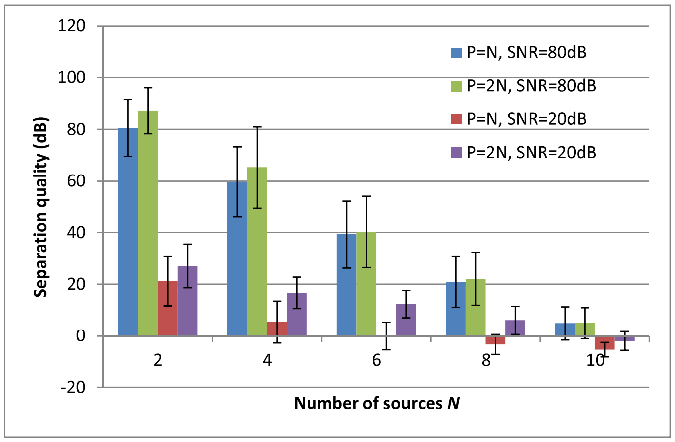

Fig. 9. Separation quality versus number of sources (N), number of sensors (P), and input SNR. For the low noise case, having twice as many sensors only brings a negligible benefit. However, when there is considerable noise, having more sensors improves the results considerably, especially for $N = 4, 6, 8$ where the improvement is larger than 10 dB.

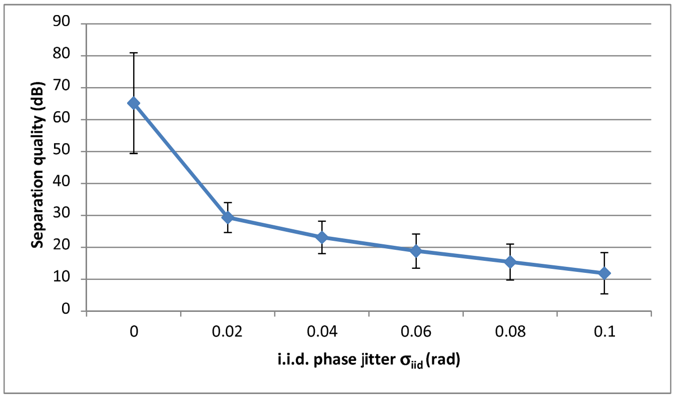

Fig. 10. Separation quality versus i.i.d. phase jitter. The horizontal axis represents the standard deviation of the phase of each entry of **J**.

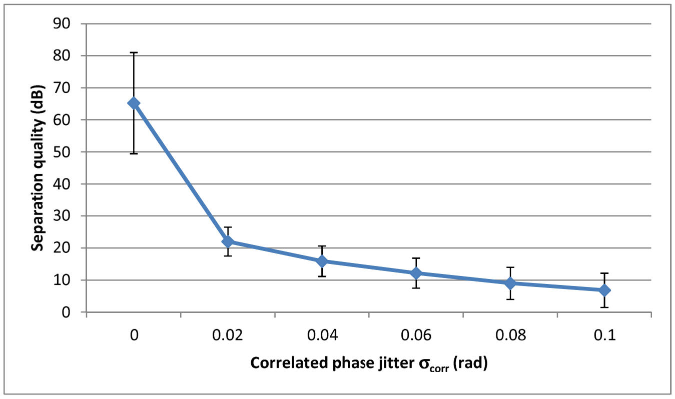

Fig. 11. Separation quality versus correlated phase jitter. The horizontal axis represents the amplitude of the correlated phase jitter.

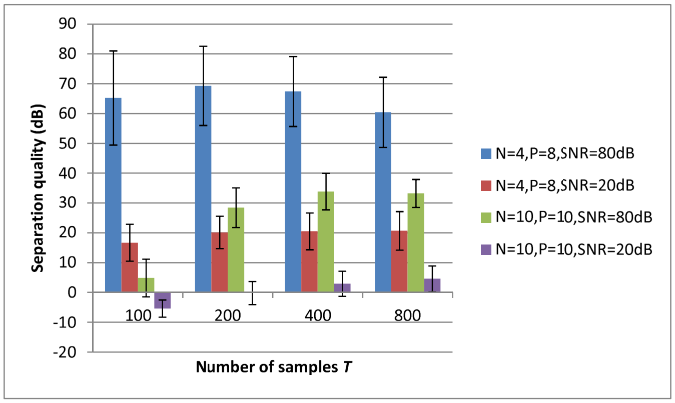

Fig. 12. Separation quality versus number of time samples T. We study four cases: $N = 4$, $P = 8$ and $N = P = 10$, each with input SNR of 80 and 20 dB.

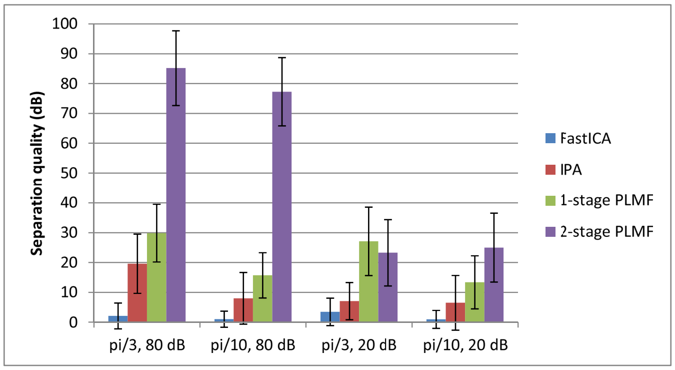

Fig. 13. Comparison of FastICA [6], IPA [25], one-stage PLMF [29], and two-stage PLMF (this paper). The two-stage PLMF algorithm clearly dominates the other two algorithms, except for one situation ($\Delta\phi = \pi/3$, input SNR of 20 dB) where it is essentially tied for first place.

Fig. 12 shows that, in some situations, increasing the number of time samples T is beneficial. The results indicate that in the ($N = P = 10$, input SNR $= 80$ dB) case the performance can be significantly improved by doubling the number of samples from $T = 100$ to $T = 200$, whereas in the other cases the improvement is small. This suggests that poor performance may not always be due to an insufficient number of time samples. It is also interesting to verify that in some cases, such as the ($N = 4$, $P = 8$, input SNR $= 80$ dB) case shown here, increasing the number of points actually yielded a slight decrease in performance. This is probably due to the larger size of the variables $\mathbf{A}$ and $\mathbf{f}$, which become harder to estimate.

3) Comparison With ICA and Other SSS Algorithms: To finalize, we compare the two-stage PLMF algorithm presented in this paper with IPA [25] and the one-stage PLMF algorithm which estimates all four variables simultaneously [29]. To illustrate, we also compare with FastICA [6], a popular ICA method. We use the same data generation procedure as in the previous results, with one exception: IPA needs a large number of samples T to perform well, whereas PLMF does not and actually works better with smaller T, since the problem is easier to solve with smaller matrices. For these reasons, we use $T = 1000$, $\Delta t = 0.01$ for IPA and $T = 100$, $\Delta t = 0.1$ for both versions of PLMF, keeping everything else equal. In practice, this corresponds to having data with a sampling frequency 10 times higher for IPA than for the two versions of PLMF.

We compare the algorithms in four situations, all of which have $N = P = 2$ sources and sensors and no phase jitter: low noise and large phase lag (input SNR of 80 dB, $\Delta\phi = \pi/3$), low noise and small phase lag (input SNR of 80 dB, $\Delta\phi = \pi/10$), moderate noise and large phase lag (input SNR of 20 dB, $\Delta\phi = \pi/3$), and moderate noise and small phase lag (input SNR of 20 dB, $\Delta\phi = \pi/10$).

The results are shown in Fig. 13. FastICA performs poorly[13] compared with all SSS algorithms, a consequence of the strong interdependence of the sources used here. Apart from one situation where both versions of PLMF are tied, these results show a clear superiority of the two-stage PLMF when compared with the other two SSS algorithms.

Fig. 13 only studies the influence of two variables: additive noise and phase lag. We did not study the influence of other variables due to space limitations and because IPA is considerably slower than PLMF.

V. Discussion

We begin our discussion by briefly mentioning the runtime of PLMF: it takes about 3 s in total, on a typical modern desktop computer, to run PLMF in MATLAB for a data set with $N = 4$ sources, $P = 8$ sensors, and $T = 100$ time samples. The time grows to about 11 s for $N = 10$, $P = 20$ and $T = 100$ (more sources and sensors, same number of samples), and to about 6 s for $N = 4$, $P = 8$ and $T = 400$ (same number of sources and sensors, more samples).

Although one can conceive of sources where the rows of $\mathbf{A}$ and the vector $\mathbf{f}$ vary rapidly with time, in many real-world systems we expect them to vary slowly. This smoothness can be enforced explicitly, by adding regularizer terms to the cost function in (6), penalizing large fluctuations in the values of $\mathbf{A}$ and $\mathbf{f}$. In that case, the problem becomes

$$\min_{\mathbf{M},\mathbf{A},\mathbf{z},\mathbf{f}} \frac{1}{2} \|\mathbf{Y} - \mathbf{M}\mathbf{D}_{\mathbf{z}}\mathbf{A}\mathbf{D}_{\mathbf{f}}\|_F^2 + \lambda_{\mathbf{A}} \|\mathbf{A}\mathbf{L}_{\mathbf{A}}\|_F^2 + \lambda_{\mathbf{f}} \|\mathbf{L}_{\mathbf{f}}\mathbf{f}\|_2^2 \quad (16)$$

with the same constraints as before, where $\mathbf{L}_{\mathbf{A}}$ and $\mathbf{L}_{\mathbf{f}}$ are the first-order difference operators of appropriate size, such that the entry (j, t) of $\mathbf{A}\mathbf{L}_{\mathbf{A}}$ is given by $a_j(t+1) - a_j(t)$, and the kth entry of $\mathbf{L}_{\mathbf{f}}\mathbf{f}$ is given by $f_{(k+1)} - f_k$. The two parameters $\lambda_{\mathbf{A}}$ and $\lambda_{\mathbf{f}}$ control the strength of the two regularizer terms. These two extra terms are especially useful in noisy situations, where they can filter out the high-frequency components of additive noise [38].

We now discuss the difference between the two types of jitter whose effects were shown in Figs. 10 and 11. The results from Figs. 10 and 11 might suggest, at first, that both types of phase jitter cause PLMFs performance to deteriorate. However, a detailed inspection of the estimated variables reveals a significant difference in behavior. Figs. 14 and 15 present a typical result of the first subproblem for two cases,

[13] We used the MATLAB FastICA implementation available from http://research.ics.aalto.fi/ica/fastica/code/dlcode.shtml. All parameters were left at their default values, except for the nonlinearity option where we tried all possibilities. All such options yield very similar results; the results reported here use the default option.

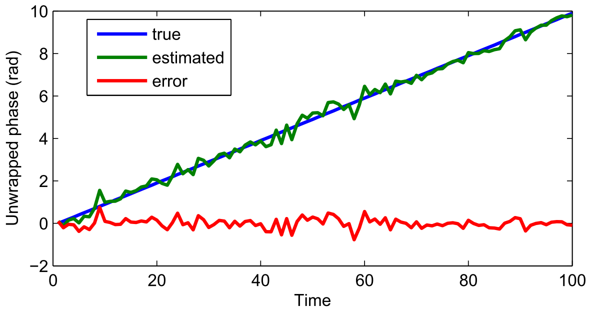

Fig. 14. Typical results of the first subproblem (estimation of $\mathbf{f}$) in the presence of strong i.i.d. phase jitter. While there is considerable error in the estimation of $\mathbf{f}$, this error could be significantly reduced using a simple low-pass filter.

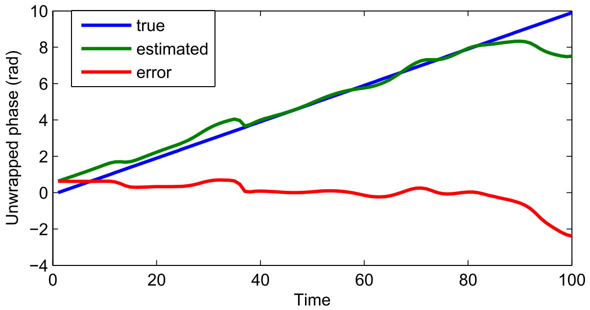

Fig. 15. Typical results of the first subproblem (estimation of $\mathbf{f}$) in the presence of strong correlated phase jitter. There is considerable error in the estimation of $\mathbf{f}$, and this error could not be significantly reduced using a low-pass filter.

which are similar in all aspects, except that the first one has i.i.d. phase jitter with a very high standard deviation ($\sigma_{\text{iid}} = 0.2$ rad), whereas the second one has very strong correlated jitter (with an amplitude of $\sigma_{\text{corr}} = 0.2$).[14] Let $\psi(t)$ denote the phase of the tth entry of $\mathbf{f}$. The plots show the true value of $\psi(t)$ used to generate the data, the estimated value of the same variable (taken as the unwrapped angle [37] of the estimated $\mathbf{f}$), and the difference between the two. Both cases have significant estimation errors. However, the errors in Fig. 14 are i.i.d. and could easily be corrected, at least partially, by low-pass filtering of $\psi(t)$, applied between the first and second subproblems of PLMF. The errors shown in Fig. 15 are not easy to correct, unless one knows *a priori* the type of correlated noise present in the system.

As mentioned in Section III-E, the optimization problems in PLMF are solved by optimizing each variable while keeping the other variables fixed; this is known as a BNGS method. There is considerable theoretical work on BNGS methods. In particular, [33] gives sufficient conditions for the following property: if a BNGS method converges to some limit solution, then that limit solution is a critical point of the problem. In other words, it is a point where the gradient of the cost function is zero. Unfortunately, the first subproblem of PLMF does not obey the conditions of that theorem; it would be necessary that $\mathbf{H}$, $\mathbf{A}$ and $\mathbf{f}$ each lie in convex sets, and that is not true for $\mathbf{H}$ and $\mathbf{f}$. The second subproblem does not obey those conditions for a similar reason involving $\mathbf{M}$ and $\mathbf{z}$.

It is possible to adapt the two subproblems to be convex in each variable. For example, one could replace the constraint on $\mathbf{M}$ with a new one, where $\max_{i,j} |m_{ij}| \leq 1$. The problem in doing this is that we introduce a new indeterminacy, where $\mathbf{M}$'s elements can tend to zero while those of $\mathbf{A}$ tend to infinity. Similar adaptations could be done to make both subproblems convex in all variables. This would yield a theoretical guarantee that if the algorithm converges, it does so to a critical point; however, it is unclear whether these new indeterminancies would deteriorate the results. This is a research direction we will pursue in the future.

It would be desirable to test PLMF using real data. To do so for EEG or MEG data, one would need data where one simultaneously knows both the EEG/MEG recordings from outside the scalp and the corresponding electrical activity within the brain. Such data are not easily accessible. In [36], we addressed this issue by constructing pseudoreal data. These data start from an actual MEG recording (which contains the mixed signals but not the actual sources, which are unknown). We then extract amplitude and phase from those recordings and use those to artificially construct data which are synchronous, and whose actual sources we know. Application of PLMF to such pseudoreal data is a direction we intend to address in future work.

VI. Conclusion

We have presented PLMF, an algorithm to perform SSS. We have shown that, under reasonable assumptions, the SSS problem has a single solution up to natural indeterminacies. In the PLMF algorithm, we have split the SSS problem into two subproblems, which are both identifiable under mild assumptions.

We have presented extensive results, using simulated data, showing how the quality of the separation varies as a function of several variables (number of sources, number of sensors, level of noise, and so on). These results show that PLMF has good robustness against additive noise and can handle small phase lags between sources; furthermore, PLMF can handle numbers of sources at least up to eight, in low noise conditions. In its present form, PLMF is unable to cope with moderate or strong phase jitter; however, for specific situations, such as i.i.d. jitter, simple post-processing of certain variables can mitigate that limitation. Results also show that splitting the problem into two subproblems yields large performance benefits when compared with previous algorithms for the SSS problem.

Appendix A

Proof of Theorem 1

Let $\mathbf{C}_{\mathbf{Z}_{\text{RI}}}$ denote the correlation matrix of $\mathbf{Z}_{\text{RI}} \equiv \mathbf{B}\mathbf{Y}_{\text{RI}} = \mathbf{B}\mathbf{M}\mathbf{S}_{\text{RI}}$. First of all, we confirm that $\mathbf{C}_{\mathbf{Z}_{\text{RI}}} = \mathbf{I}$

$$\mathbf{C}_{\mathbf{Z}_{\text{RI}}} \equiv \mathbf{Z}_{\text{RI}}\mathbf{Z}_{\text{RI}}^T \tag{17}$$
$$= \mathbf{B}\mathbf{Y}_{\text{RI}}\mathbf{Y}_{\text{RI}}^T\mathbf{B}^T \tag{18}$$
$$= \mathbf{B}\mathbf{C}_{\mathbf{Y}_{\text{RI}}}\mathbf{B}^T \tag{19}$$
$$= \mathbf{D}^{-1/2}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{V}\mathbf{D}^{-1/2} \tag{20}$$
$$= \mathbf{I} \tag{21}$$

[14] These very high values were chosen for illustration purposes, and do not correspond to any of the results in Figs. 10 and 11.

because $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ by construction. Therefore, one has

$$\mathbf{I} = \mathbf{C}_{\mathbf{Z}_{\mathrm{RI}}} = \mathbf{B}\mathbf{M}\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}\mathbf{M}^T\mathbf{B}^T = (\mathbf{B}\mathbf{M}\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}^{1/2})(\mathbf{B}\mathbf{M}\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}^{1/2})^T \quad (22)$$

and one can conclude that $\mathbf{B}\mathbf{M}\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}^{1/2}$ is an orthogonal matrix, which we denote by $\mathbf{R}$.

We now study the singular values of the equivalent mixing matrix $\mathbf{BM}$. From the definition of $\mathbf{R}$, it holds that $\mathbf{BM} = \mathbf{R}\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}^{-1/2}$, and that the singular values of $\mathbf{BM}$ are the same as those of $\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}^{-1/2}$, since $\mathbf{R}$ is orthogonal. Therefore, the conditioning of the equivalent source separation problem can be studied by studying the singular values of $\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}^{-1/2}$.

We first note that $\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}} = (\mathbf{C}_{\mathbf{S}_R} + \mathbf{C}_{\mathbf{S}_I})/2$. We start by computing the (j,k) element of $\mathbf{C}_{\mathbf{S}_R}$

$$\begin{aligned}[\mathbf{C}_{\mathbf{S}_R}]_{jk} &\equiv \mathbb{E}[\mathrm{Re}[s_j(t)]\mathrm{Re}[s_k(t)]] &(23)\\ &= \mathbb{E}[A_j(t)\cos[\phi_j(t)]A_k(t)\cos[\phi_k(t)]]. &(24)\end{aligned}$$

If $j \neq k$, the phases are independent of the amplitudes and the two amplitudes are independent of each other. Thus, for $j \neq k$

$$\begin{aligned}[\mathbf{C}_{\mathbf{S}_R}]_{jk} &= \mathbb{E}[A_j(t)]\mathbb{E}[A_k(t)]\mathbb{E}[\cos[\phi_j(t)]\cos[\phi_k(t)]] &(25)\\ &= \mathbb{E}[A]^2\mathbb{E}\Big[\frac{1}{2}\cos[\phi_j(t)+\phi_k(t)]\\ &\qquad + \frac{1}{2}\cos[\phi_j(t)-\phi_k(t)]\Big] &(26)\\ &= \mathbb{E}[A]^2\mathbb{E}\Big[\frac{1}{2}\cos[\phi_j(1)+\phi_k(1)+2\phi(t)]\\ &\qquad + \frac{1}{2}\cos[\phi_j(1)-\phi_k(1)]\Big] &(27)\\ &= \frac{1}{2}\mathbb{E}[A]^2\mathbb{E}[\cos[\phi_j(1)-\phi_k(1)]] &(28)\end{aligned}$$

where in the second-to-last equality we used the assumption that all sources are perfectly phase-locked, and thus for any j, $\phi_j(t) = \phi_j(1) + \phi(t)$; and in the last equality we used the assumption that $\phi(t)$ is uniformly distributed in $[0, 2\pi)$, and thus $\mathbb{E}[\cos[\phi_j(1)+\phi_k(1)+2\phi(t)]] = 0$.

On the other hand, if $j = k$, from (24) we get

$$[\mathbf{C}_{\mathbf{S}_R}]_{jj} = \mathbb{E}[A_j(t)A_j(t)]\mathbb{E}[\cos[\phi_j(t)]^2] = \frac{1}{2}\mathbb{E}[A^2]. \quad (29)$$

By replacing co-sines with sines in (24), a very similar reasoning yields the exact same expressions for the (j,k) element of $\mathbf{C}_{\mathbf{S}_I}$. Since $\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}} = (\mathbf{C}_{\mathbf{S}_R} + \mathbf{C}_{\mathbf{S}_I})/2$, the expressions for the (j,k) element of $\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}$ are similar to those in (28) and (29) with the factor $1/2$ omitted.

Since $\mathbb{E}[A^2] = \mathrm{Var}[A] + \mathbb{E}[A]^2$, we can merge the cases $j = k$ and $j \neq k$ into the following expression:

$$\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}} = \frac{\mathrm{Var}[A]\mathbf{I} + \mathbb{E}[A]^2\mathbf{F}}{2} \quad (30)$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{F}_{jk} \equiv \cos[\phi_j(1) - \phi_k(1)]$ for all j, k, including $j = k$.

We now study the eigenvalues of matrix $\mathbf{F}$ (which are equal to its singular values, since $\mathbf{F}$ is symmetric and positive semidefinite, as shown below). It is easy to see that $\mathbf{F} = \mathrm{Re}(\mathbf{G})$, with $\mathbf{G} \equiv \mathbf{x}\mathbf{x}^H$, where the vector $\mathbf{x}$ has components $x_j \equiv e^{\mathrm{i}\phi_j(1)}$. $\mathbf{G}$ has a simple eigenvalue with value N (the number of sources), and an eigenvalue with value 0 with multiplicity $N - 1$.

Since the eigenvalues of $\mathbf{G}$ are 0 and N, the eigenvalues of $\mathbf{F}$ necessarily obey $0 \leq \lambda(\mathbf{F}) \leq N$. To see this, let $\mathbf{v}$ be any real vector with unit norm. Note that since $\mathbf{v}$ is real, we have $\mathbf{v}^H = \mathbf{v}^T$. Note further that $\mathbf{v}^T\mathrm{Im}(\mathbf{G})\mathbf{v} = 0$ because $\mathbf{G}$ is Hermitian, and therefore its imaginary part is skew-symmetric. Then

$$\mathbf{v}^T\mathbf{F}\mathbf{v} = \mathbf{v}^T\mathbf{F}\mathbf{v} + \mathbf{v}^T\mathrm{Im}(\mathbf{G})\mathbf{v} = \mathbf{v}^T\mathbf{G}\mathbf{v} = \mathbf{v}^H\mathbf{G}\mathbf{v}. \quad (31)$$

The rightmost expression's value is between 0 and N, since those are the smallest and largest eigenvalues of $\mathbf{G}$. Thus the leftmost expression must also be between those values. Therefore, the eigenvalues of $\mathbf{F}$ obey $0 \leq \lambda(\mathbf{F}) \leq N$.

We now use simple properties of eigenvalues to get bounds for the eigenvalues of $\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}$, using the result from (30)

$$\begin{aligned}0 \leq \lambda(\mathbf{F}) &\leq N &(32)\\ 0 \leq \lambda(\frac{\mathbb{E}[A]^2}{2}\mathbf{F}) &\leq N\frac{\mathbb{E}[A]^2}{2} &(33)\\ \frac{\mathrm{Var}[A]}{2} \leq \lambda\left(\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}\right) &\leq \frac{\mathrm{Var}[A] + N\mathbb{E}[A]^2}{2}. &(34)\end{aligned}$$

Thus, the condition number of $\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}$ is bounded above by the quotient of these two bounds: $\rho(\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}) \leq 1 + N\mathbb{E}[A]^2/\mathrm{Var}[A]$. Also, from simple properties of singular values, one can conclude that

$$\rho(\mathbf{BM}) = \rho(\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}^{-1/2}) = \sqrt{\rho\left(\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}\right)} \leq \sqrt{1 + N\frac{\mathbb{E}[A]^2}{\mathrm{Var}[A]}}. \quad (35)$$

The proof that this upper bound is tight is very simple. It is sufficient to consider the case $\phi_j(1) = \phi_k(1)$ for all j, k, i.e., the situation where all sources have zero phase lag with one another. In that case, $\mathbf{F}$ is a matrix of ones, and its eigenvalues are exactly 0 and N. It is very simple to see that in that case, $\rho(\mathbf{C}_{\mathbf{S}_{\mathrm{RI}}}^{-1/2}) = (1 + N\mathbb{E}[A]^2/\mathrm{Var}[A])^{1/2}$.

Appendix B
Proof of Theorem 2

Our starting point is $\mathbf{H}_1\mathbf{A}_1\mathbf{D}_{\mathbf{f}1} = \mathbf{H}_2\mathbf{A}_2\mathbf{D}_{\mathbf{f}2}$. We multiply both sides on the left by $\mathbf{H}_1^\dagger$, where the symbol † denotes the Moore–Penrose pseudoinverse [39]. We also multiply on the right by $\mathbf{D}_{\mathbf{f}1}^{-1} \equiv \mathbf{D}_{\mathbf{f}1}^*$, where * represents the entrywise complex conjugate. We thus obtain a new equation $\mathbf{A}_1 = \mathbf{H}_0\mathbf{A}_2\mathbf{D}_{\mathbf{f}0}$ where $\mathbf{H}_0 \equiv \mathbf{H}_1^\dagger\mathbf{H}_2 \in \mathbb{C}^{N\times N}$ has full rank and $\mathbf{D}_{\mathbf{f}0} \equiv \mathbf{D}_{\mathbf{f}2}\mathbf{D}_{\mathbf{f}1}^* \in \mathbb{D}_1^T$.

We now write the equation for the tth columns of $\mathbf{A}_1$ and $\mathbf{A}_2$ (which we denote by $\mathbf{a}_1(t)$ and $\mathbf{a}_2(t)$, respectively)

$$\mathbf{a}_1(t) = \mathbf{H}_0\mathbf{a}_2(t)e^{\mathrm{i}\psi(t)} \quad (36)$$

where $e^{\mathrm{i}\psi(t)}$ is the tth diagonal element of $\mathbf{D}_{\mathbf{f}0}$. If we write this equation for two time instants t_1 and t_2 and linearly combine them with real coefficients α_1 and α_2, we get

$$\begin{aligned}&\alpha_1\mathbf{a}_1(t_1) + \alpha_2\mathbf{a}_1(t_2)\\ &\qquad = \mathbf{H}_0\left[\alpha_1\mathbf{a}_2(t_1)e^{\mathrm{i}\psi(t_1)} + \alpha_2\mathbf{a}_2(t_2)e^{\mathrm{i}\psi(t_2)}\right]. \quad (37)\end{aligned}$$

The left-hand side of (37) is real for all $\alpha_1, \alpha_2 \in \mathbb{R}$. The right-hand side can be real for all $\alpha_1, \alpha_2 \in \mathbb{R}$ only if $e^{\mathrm{i}\psi(t_1)}$ and

$e^{\mathrm{i}\psi(t_2)}$ are parallel vectors in the complex plane, which yields $\psi(t_1) = \psi(t_2)$ or $\psi(t_1) = \psi(t_2) + \pi$.

By using the above reasoning for all pairs (t_1, t_2) we can conclude that $\mathbf{D_{f0}} = \mathbf{E}$ where $\mathbf{E}$ is diagonal with elements in the set $\{-e^{\mathrm{i}\gamma}, +e^{\mathrm{i}\gamma}\}$, where γ is a real number. Multiplying both sides of this last equation on the right by $\mathbf{D_{f1}} \equiv \mathbf{D}_{\mathbf{f}1}^{*-1}$ yields $\mathbf{D_{f2}} = \mathbf{ED_{f1}}$, as desired.

Appendix C

Proof of Theorem 3

Let $\mathbf{E}_1$ and $\mathbf{E}_2$ be real diagonal matrices such that $\mathbf{E}_1\mathbf{D_{z1}}$ and $\mathbf{E}_2\mathbf{D_{z2}}$ are diagonal matrices whose diagonal elements have a real part of 1. We assume, with no loss of generality, that these matrices exist (see the end of this proof for an explanation of why no generality is lost with this assumption). It is easy to see that, if they exist, $\mathbf{E}_1$ and $\mathbf{E}_2$ are invertible. We thus have

$$\mathbf{M}_1\mathbf{D_{z1}}\mathbf{A}_1 = \mathbf{M}_2\mathbf{D_{z2}}\mathbf{A}_2 \tag{38}$$

$$\mathbf{M}_1\mathbf{E}_1^{-1}\mathbf{E}_1\mathbf{D_{z1}}\mathbf{A}_1 = \mathbf{M}_2\mathbf{E}_2^{-1}\mathbf{E}_2\mathbf{D_{z2}}\mathbf{A}_2 \tag{39}$$

$$\mathbf{M}'_1\mathbf{D}'_{\mathbf{z}1}\mathbf{A}_1 = \mathbf{M}'_2\mathbf{D}'_{\mathbf{z}2}\mathbf{A}_1 \tag{40}$$

where $\mathbf{M}'_{1,2} \equiv \mathbf{M}_{1,2}\mathbf{E}_{1,2}^{-1}$ and $\mathbf{D}'_{\mathbf{z}1,2} \equiv \mathbf{E}_{1,2}\mathbf{D}_{\mathbf{z}1,2}$. Note that the assumption that all diagonal entries of $\mathbf{D_{z1}}$ are different modulo π ensures that all diagonal entries of $\mathbf{D}'_{\mathbf{z}1}$ are different, and the same for $\mathbf{D}'_{\mathbf{z}2}$. Let us split (40) into its real and imaginary parts. Since by definition the real part of $\mathbf{D}'_{\mathbf{z}1,2}$ is the identity matrix, we obtain

$$\begin{aligned}\mathbf{M}'_1\mathbf{A}_1 &= \mathbf{M}'_2\mathbf{A}_2\\ \mathbf{M}'_1\mathrm{Im}(\mathbf{D}'_{\mathbf{z}1})\mathbf{A}_1 &= \mathbf{M}'_2\mathrm{Im}(\mathbf{D}'_{\mathbf{z}2})\mathbf{A}_2.\end{aligned} \tag{41}$$

Solving the first equation for $\mathbf{A}_2$ and plugging the result into the second equation yields, after simplification

$$\mathbf{M}'_1\mathrm{Im}(\mathbf{D}'_{\mathbf{z}1})\mathbf{M}_1'^{-1} = \mathbf{M}'_2\mathrm{Im}(\mathbf{D}'_{\mathbf{z}2})\mathbf{M}_2'^{-1}. \tag{42}$$

Since $\mathrm{Im}(\mathbf{D}'_{\mathbf{z}1})$ is a diagonal matrix, the left-hand side can be interpreted as an eigenvalue decomposition of some matrix, where $\mathrm{Im}(\mathbf{D}'_{\mathbf{z}1})$ contains the eigenvalues in its diagonal and $\mathbf{M}'_1$ contains the corresponding eigenvectors in its columns.[15] A similar interpretation can be given to the right-hand side. Furthermore, since all diagonal elements of $\mathbf{D}'_{\mathbf{z}1,2}$ are different from one another, all eigenvalues have algebric and geometric multiplicities of 1 on the left-hand side and on the right-hand side. Therefore, $\mathrm{Im}(\mathbf{D}'_{\mathbf{z}1}) = \mathrm{Im}(\mathbf{D}'_{\mathbf{z}2})$ up to an arbitrary permutation of its diagonal elements, and $\mathbf{M}'_1 = \mathbf{M}'_2$ up to the same permutation and up to arbitrary scaling of each of their columns. Consequently, from (40), one concludes that $\mathbf{A}_1 = \mathbf{A}_2$ up to the same permutation and to a scaling of its rows which is the inverse of the scaling of the columns of $\mathbf{M}'_1$ and $\mathbf{M}'_2$.

Since $\mathbf{A}_1 = \mathbf{A}_2$ up to permutation and scaling of rows, we can instead write that fact as $\mathbf{A}_1 = \mathbf{PA}_2$, where $\mathbf{P}$ is a permutation of a diagonal matrix with nonzero entries on the diagonal.

We now have, from (38)

$$\mathbf{M}_1\mathbf{D_{z1}}\mathbf{P}\mathbf{A}_2 = \mathbf{M}_2\mathbf{D_{z2}}\mathbf{A}_2 \tag{43}$$

and, since $\mathbf{A}_1$ and $\mathbf{A}_2$ have maximum row rank by assumption

$$\mathbf{M}_1\mathbf{D_{z1}}\mathbf{P} = \mathbf{M}_2\mathbf{D_{z2}}. \tag{44}$$

Let us consider the mth column of both sides of this equation. Suppose that the (m, n)th entry of $\mathbf{P}$ is nonzero. We get

$$e^{\mathrm{i}\alpha_{1m}}\mathbf{m}_{1m} = e^{\mathrm{i}\alpha_{2n}}\mathbf{m}_{2n} \tag{45}$$

where $e_m^{\mathrm{i}\alpha_1}$ is the (m, m)th entry of $\mathbf{D_{z1}}$, $e_n^{\mathrm{i}\alpha_2}$ is the (n, n)th entry of $\mathbf{D_{z2}}$, $\mathbf{m}_{1m}$ is the mth column of $\mathbf{M}_1$, and $\mathbf{m}_{2n}$ is the nth column of $\mathbf{M}_2$. We can thus conclude that $\mathbf{M}_1 = \mathbf{M}_2$ up to permutation and positive scaling of columns, and $\mathbf{D_{z1}} = \mathbf{D_{z2}}$ up to the same permutation of columns.

We now show that we can, with no loss of generality, assume that $\mathbf{E}_1$ and $\mathbf{E}_2$ exist. Note that they exist unless some elements in $\mathbf{z}_1$ and/or $\mathbf{z}_2$ have a real part of zero. In that case, let θ denote a real number such that the elements of $e^{\mathrm{i}\theta}\mathbf{z}_1$ and $e^{\mathrm{i}\theta}\mathbf{z}_2$ all have nonzero real parts. Since the number of elements in $\mathbf{z}_1$ and $\mathbf{z}_2$ is finite, such a number always exists. All the steps of this proof remain valid if one replaces $\mathbf{z}_1$ and $\mathbf{z}_2$ with $e^{\mathrm{i}\theta}\mathbf{z}_1$ and $e^{\mathrm{i}\theta}\mathbf{z}_2$ everywhere.

Appendix D

Proof of Theorem 4

Define $\mathbf{H}_1 \equiv \mathbf{M}_1\mathbf{D}_{\mathbf{z}_1}$ and $\mathbf{H}_2 \equiv \mathbf{M}_2\mathbf{D}_{\mathbf{z}_2}$. Theorem 2 can be applied to the factorizations $\mathbf{H}_1\mathbf{A}_1\mathbf{D}_{\mathbf{z}_1}$ and $\mathbf{H}_2\mathbf{A}_2\mathbf{D}_{\mathbf{z}_2}$. Therefore, $\mathbf{D}_{\mathbf{f}_1} = \mathbf{E}\mathbf{D}_{\mathbf{f}_2}$, where $\mathbf{E} = e^{\mathrm{i}\gamma}\mathbf{I}_\pm$, where $\mathbf{I}_\pm$ is a diagonal matrix with diagonal elements equal to -1 or $+1$ and γ is a real number.

Substituting for $\mathbf{D}_{\mathbf{f}_1}$, we get the two factorizations

$$\mathbf{Y} = \mathbf{M}_1\mathbf{D}_{\mathbf{z}_1}e^{\mathrm{i}\gamma}\mathbf{A}_1\mathbf{I}_\pm\mathbf{D}_{\mathbf{f}_2} \qquad \text{and} \qquad \mathbf{Y} = \mathbf{M}_2\mathbf{D}_{\mathbf{z}_2}\mathbf{A}_2\mathbf{D}_{\mathbf{f}_2}.$$

Define $\mathbf{D}'_{\mathbf{z}_1} \equiv \mathbf{D}_{\mathbf{z}_1}e^{\mathrm{i}\gamma}$ and $\mathbf{A}'_1 \equiv \mathbf{A}_1\mathbf{I}_\pm$. Then, theorem 3 is applicable to the two factorizations $\mathbf{YD}^*_{\mathbf{f}_2} = \mathbf{M}_1\mathbf{A}'_1\mathbf{D}'_{\mathbf{z}_1}$ and $\mathbf{YD}^*_{\mathbf{f}_2} = \mathbf{M}_2\mathbf{A}_2\mathbf{D}_{\mathbf{z}_2}$.

Therefore, we get $\mathbf{M}_1 = \mathbf{M}_2$, $\mathbf{z}'_1 = \mathbf{z}_2$ (i.e., $\mathbf{z}_1 = e^{\mathrm{i}\gamma}\mathbf{z}_2$, thus $\mathbf{z}_1 = \mathbf{z}_2$ up to rotation), and $\mathbf{A}'_1 = \mathbf{A}_1\mathbf{I}_\pm = \mathbf{A}_2$, up to permutation and scaling. Since $\mathbf{A}_1$ and $\mathbf{A}_2$ are negative by assumption, the last of these equalities implies that $\mathbf{A}_1 = \mathbf{A}_2$ up to permutation and positive scaling.

References

[1] M. S. Pedersen, J. Larsen, U. Kjems, and L. C. Parra, "A survey of convolutive blind source separation methods," *Springer Handbook of Speech Processing*. New York, NY, USA: Springer-Verlag, 2008, pp. 1065–1084.

[2] C. Jutten and J. Karhunen, "Advances in nonlinear blind source separation," in *Proc. 4th Int. Symp. Independ. Compon. Anal. Blind Signal Separat.*, 2003, pp. 245–256.

[3] L. B. Almeida, "Synthesis lectures on signal processing," in *Nonlinear Source Separation*. San Rafael, CA, USA: Morgan & Claypool, 2006.

[4] C. Jutten and J. Herault, "Blind separation of sources part I: An adaptive algorithm based on neuromimetic architecture," *Signal Process.*, vol. 24, no. 1, pp. 1–10, Jul. 1991.

[5] P. Comon, "Independent component analysis, a new concept?" *Signal Process.*, vol. 36, no. 3, pp. 287–314, 1994.

[6] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York, NY, USA: Wiley, 2001.

[7] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing—Learning Algorithms and Applications*. New York, NY, USA: Wiley, 2002.

[8] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. New York, NY, USA: Academic, 2010.

[15] Contrary to common convention, in this case the eigenvectors do not have unit norm.

[9] B. Póczos and A. Lörincz, "Independent subspace analysis using geodesic spanning trees," in *Proc. 22nd ICML*, 2005, pp. 673–680.

[10] F. Theis, "Towards a general independent subspace analysis," in *Adv. Neural Inf. Process. Syst.*, 2007, pp. 1361–1368.

[11] Z. Szabó, B. Póczos, and A. Lorincz, "Undercomplete blind subspace deconvolution," *J. Mach. Learn. Res.*, vol. 8, no. 5, pp. 1063–1095, 2007.

[12] A. Sharma and K. Paliwal, "Subspace independent component analysis using vector kurtosis," *Pattern Recognit.*, vol. 39, no. 11, pp. 2227–2232, 2006.

[13] J.-F. Cardoso, "Multidimensional independent component analysis," in *Proc. ICASSP*, May 1998, pp. 1941–1944.

[14] P. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Nov. 2004.

[15] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," *Adv. Neural Inf. Process. Syst.*, vol. 13, pp. 556–562, Apr. 2001.

[16] H. Kameoka, N. Ono, K. Kashino, and S. Sagayama, "Complex NMF: A new sparse representation for acoustic signals," in *Proc. IEEE ICASSP*, Apr. 2009, pp. 3437–3440.

[17] B. J. King and L. Atlas, "Single-channel source separation using complex matrix factorization," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2591–2597, Nov. 2011.

[18] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Sciences*, Cambridge, U.K.: Cambridge Univ. Press, 2001.

[19] J.-P. Lachaux, E. Rodriguez, J. Martinerie, and F. J. Varela, "Measuring phase synchrony in brain signals," *Human Brain Mapping*, vol. 8, no. 4, pp. 194–208, 1999.

[20] P. Tass, M. G. Rosenblum, J. Weule, J. Kurths, A. Pikovsky, J. Volkmann, *et al.*, "Detection of n:m phase locking from noisy data: Application to magnetoencephalography," *Phys. Rev. Lett.*, vol. 81, no. 15, pp. 3291–3294, 1998.

[21] J. M. Palva, S. Palva, and K. Kaila, "Phase synchrony among neuronal oscillations in the human cortex," *J. Neurosci.*, vol. 25, no. 15, pp. 3962–3972, Apr. 2005.

[22] B. A. Conway, D. M. Halliday, S. F. Farmer, U. Shahani, P. Maas, A. I. Weir, *et al.*, "Synchronization between motor cortex and spinal motoneuronal pool during the performance of a maintained motor task in man," *J. Physiol.*, vol. 489, pp. 917–924, Dec. 1995.

[23] J.-M. Schoffelen, R. Oostenveld, and P. Fries, "Imaging the human motor system's beta-band synchronization during isometric contraction," *Neuroimage*, vol. 41, no. 2, pp. 437–447, Jun. 2008.

[24] P. J. Uhlhaas and W. Singer, "Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology," *Neuron*, vol. 52, no. 1, pp. 155–168, Oct. 2006.

[25] M. Almeida, J.-H. Schleimer, J. Bioucas-Dias, and R. Vigário, "Source separation and clustering of phase-locked subspaces," *IEEE Trans. Neural Netw.*, vol. 22, no. 9, pp. 1419–1434, Sep. 2011.

[26] M. Almeida, J. Bioucas-Dias, and R. Vigário, "Independent phase analysis: Separating phase-locked subspaces," in *Proc. Latent Variable Anal. Conf.*, 2010, pp. 189–196.

[27] A. Ziehe and K.-R. Müller, "TDSEP—An efficient algorithm for blind separation using time structure," in *Proc. ICANN*, Sep. 1998, pp. 675–680.

[28] M. Almeida, R. Vigario, and J. Bioucas-Dias, "Phase locked matrix factorization," in *Proc. EUSIPCO Conf.*, 2011, pp. 1728–1732.

[29] M. Almeida, R. Vigário, and J. Bioucas-Dias, "Estimation of the common oscillation for phase locked matrix factorization," in *Proc. ICPRAM*, 2012, pp. 78–85.

[30] B. Gold, A. V. Oppenheim, and C. M. Rader, "Theory and implementation of the discrete Hilbert transform," in *Discrete Signal Processing*, L. R. Rabiner and C. M. Rader, Eds. Upper Saddle River, NJ, USA: Prentice-Hall, 1973.

[31] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal—Part I: Fundamentals," *Proc. IEEE*, vol. 80, no. 4, pp. 519–538, Apr. 1992.

[32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[33] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints," *Oper. Res. Lett.*, vol. 26, no. 3, pp. 127–136, 2000.

[34] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. New York, NY, USA: Taylor & Francis, 1998.

[35] M. Almeida, R. Vigário, and J. Bioucas-Dias, "The role of whitening for separation of synchronous sources," in *Proc. Latent Variable Anal. Conf.*, 2012, pp. 139–146.

[36] M. Almeida, J. Bioucas-Dias, and R. Vigário, "Separation of phase-locked sources in pseudo-real MEG data," *EUSIPCO J. Adv. Signal Process.*, vol. 32, pp. 2–12, Feb. 2013.

[37] J. Bioucas-Dias and G. Valadão, "Phase unwrapping via graph cuts," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 698–709, Mar. 2007.

[38] M. Almeida, R. Vigário, and J. Bioucas-Dias, "Phase-locked matrix factorization with estimation of the common oscillation," in *Mathematical Methodologies in Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2013, pp. 51–66.

[39] A. Ben-Israel and T. Greville, *Generalized Inverses: Theory and Applications*. New York, NY, USA: Springer-Verlag, 2003.

Miguel S. B. Almeida received the M.Sc. degree in physics and technology engineering from Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, in 2006, and the Advanced Training degree in biophysics from Instituto de Biofísica e Engenharia Biomédica, University of Lisbon, in 2007. He is currently pursuing the Doctoral degree in electrical and computer engineering with the Technical University of Lisbon.

He joined the Neuroinformatics Group, Helsinki University of Technology (currently Aalto University), Aalto, Finland, in 2008, and the Pattern and Image Analysis Group, Instituto de Telecomunicações, Technical University of Lisbon, in 2010. His current research interests include statistical machine learning, optimization, and pattern recognition; and application of these topics to brain research using EEG, MEG, or fMRI data, in particular to study sleep and synchrony between distant brain regions.

Ricardo Vigário received the Licenciatura and Mestrado degrees in applied physics and biomedical engineering from the University of Lisbon, Lisbon, Portugal, in 1992 and 1994, respectively, and the D.Sc. degree from the Helsinki University of Technology (currently Aalto University, since 2010), Aalto, Finland, in 1999.

He became a Docent of statistical machine learning in biomedical systems with the Helsinki University of Technology in 2004. He has been a Visiting Lecturer with the Technical University of Graz, Graz, Austria, in 2000 and 2001, and Zaragoza University, Zaragoza, Spain, in 2003. He has been a Visiting Associate Professor with Institut National Polytechnique de Grenoble, Grenoble, France, in 2007. He is a Senior Researcher with the Department of Information and Computer Science, School of Science, Aalto University, where he leads a research team in neuroinformatics. His current research interests include adaptive informatics and statistical machine learning, as well as their application to neuroinformatics, functional human brain mapping, and biomedical signal processing.

José Bioucas-Dias (S'87–M'95) received the E.E., M.Sc., Ph.D., and Agregado degrees in electrical and computer engineering from Instituto Superior Técnico (IST), the Engineering School, Technical University of Lisbon, Lisbon, Portugal, in 1985, 1991, 1995, and 2007, respectively.

He has been with the Department of Electrical and Computer Engineering, IST, since 1995, where he was an Assistant Professor from 1995 to 2007 and an Associate Professor since 2007. Since 1993, he has been a Senior Researcher with the Pattern and Image Analysis Group, Instituto de Telecomunicações, which is a private nonprofit research institution. His current research interests include inverse problems, signal and image processing, pattern recognition, optimization, and remote sensing.

Dr. Bioucas-Dias was an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1997 to 2000 and he is an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He was a Guest Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING for the *Special Issue on Spectral Unmixing of Remotely Sensed Data* and the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING for the *Special Issue on Hyperspectral Image and Signal Processing*, and he is a Guest Editor of the IEEE SIGNAL PROCESSING MAGAZINE for the *Special Issue on Signal and Image Processing in Hyperspectral Remote Sensing*, He was the General Co-Chair of the Third IEEE GRSS Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote sensing in 2011, and has been a member of program/technical committees of several international conferences.