ESTIMATION OF THE COMMON OSCILLATION FOR PHASE LOCKED MATRIX FACTORIZATION

Miguel Almeida[1,2], Ricardo Vigário[2] and José Bioucas-Dias[1]
[1]*Institute of Telecommunications, Instituto Superior Técnico, Lisbon, Portugal*
[2]*Adaptive Informatics Research Centre, Aalto University School of Science, Finland*

Keywords: Matrix factorization, Phase synchrony, Phase-locking, Independent component analysis, Blind source separation, Convex optimization.

Abstract: Phase Locked Matrix Factorization (PLMF) is an algorithm to perform separation of synchronous sources. Such a problem cannot be addressed by orthodox methods such as Independent Component Analysis, because synchronous sources are highly mutually dependent. PLMF separates available data into the mixing matrix and the sources; the sources are then decomposed into amplitude and phase components. Previously, PLMF was applicable only if the oscillatory component, common to all synchronized sources, was known, which is clearly a restrictive assumption. The main goal of this paper is to present a version of PLMF where this assumption is no longer needed – the oscillatory component can be estimated alongside all the other variables, thus making PLMF much more applicable to real-world data. Furthermore, the optimization procedures in the original PLMF are improved. Results on simulated data illustrate that this new approach successfully estimates the oscillatory component, together with the remaining variables, showing that the general problem of separation of synchronous sources can now be tackled.

1 INTRODUCTION

Synchrony is an increasingly studied topic in modern science. On one hand, there is an elegant yet deep mathematical framework which is applicable to many domains where synchrony is present, including laser interferometry, the gravitational pull of stellar objects, and the human brain (Pikovsky et al., 2001).

It is believed that synchrony plays an important role in the way different sections of human brain interact. For example, when humans perform a motor task, several brain regions oscillate coherently with the muscle's electromyogram (EMG) (Palva et al., 2005; Schoffelen et al., 2008). Also, processes such as memorization and learning have been associated with synchrony; several pathologies such as autism, Alzheimer's and Parkinson's are associated with a disruption in the synchronization profile of the brain; and epilepsy is associated with an anomalous increase in synchrony (Uhlhaas and Singer, 2006).

To infer knowledge on the synchrony of the networks present in the brain or in other real-world systems, one must have access to the dynamics of the individual oscillators (which we will call "sources"). Usually, in the brain electroencephalogram (EEG) and magnetoencephalogram (MEG), and other real-world situations, individual oscillator signals are not directly measurable; one has only access to a superposition of the sources.[1] In fact, EEG and MEG signals measured in one sensor contain components coming from several brain regions (Nunez et al., 1997). In this case, spurious synchrony occurs, as has been shown both empirically and theoretically in previous works (Almeida et al., 2011a). We briefly review this evidence in Section 2.3.

Undoing this superposition is usually called a blind source separation (BSS) problem. Typically, one assumes that the mixing is linear and instantaneous, which is a valid and common approximation in brain signals (Vigário et al., 2000) and other applications. In this case, if the vector of sources is denoted by $\mathbf{s}(t)$ and the vector of measurements by $\mathbf{x}(t)$, they are related through $\mathbf{x}(t) = \mathbf{M}\mathbf{s}(t)$ where $\mathbf{M}$ is a real matrix called the mixing matrix. Even with this assumption, the BSS problem is ill-posed:

[1]In EEG and MEG, the sources are not individual neurons, whose oscillations are too weak to be detected from outside the scalp even with no superposition. In this case, the sources are populations of closely located neurons oscillating together.

Almeida M., Vigário R. and Bioucas-Dias J. (2012).
ESTIMATION OF THE COMMON OSCILLATION FOR PHASE LOCKED MATRIX FACTORIZATION.
In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, pages 78-85
DOI: 10.5220/0003774300780085
Copyright © SciTePress

there are infinitely many solutions. Thus, one must also make some assumptions on the sources, such as statistical independence in Independent Component Analysis (ICA) (Hyvärinen et al., 2001). However, in the case discussed in this paper, independence of the sources is not a valid assumption, because synchronous sources are highly dependent. In this paper we address the problem of how to separate these dependent sources, a problem we name Separation of Synchronous Sources, or Synchronous Source Separation (SSS). Although many possible formal models for synchrony exist (see, *e.g.*, (Pikovsky et al., 2001) and references therein), in this paper we use a simple yet popular measure of synchrony: the Phase Locking Factor (PLF), or Phase Locking Value (PLV). The PLF between two signals is 1 if they are perfectly synchronized. Thus, in this paper we tackle the problem of source separation where all pairs of sources have a PLF of 1.

A more general problem has also been addressed, where the sources are organized in subspaces, with sources in the same subspace having strong synchrony and sources in different subspaces having weak synchrony. This general problem was tackled with a two-stage algorithm called Independent Phase Analysis (IPA) which performed well in the noiseless case (Almeida et al., 2010) and with moderate levels of added Gaussian white noise (Almeida et al., 2011a). In short, IPA uses TDSEP (Ziehe and Müller, 1998) to separate the subspaces from one another. Then, each subspace is a separate SSS problem; IPA uses an optimization procedure to complete the intra-subspace separation. Although IPA performs well for the noiseless case, and for various types of sources and 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 a number of measurements equal to the number of sources. It may as well return singular solutions, where two or more estimated sources are (almost) identical. On the other hand, IPA can deal with subspaces of phase-locked sources and with sources that are not perfectly phase-locked (Almeida et al., 2011a).

In this paper we address an alternative technique, named Phase Locked Matrix Factorization (PLMF). PLMF was originally introduced in (Almeida et al., 2011b), using a very restrictive assumption, of prior knowledge of the oscillation common to all the sources. The goal of this paper is to remove this restrictive assumption, and to improve the optimization of the problem.

Unlike IPA, PLMF can deal with higher amounts of noise and with non-square mixing matrices (more measurements than sources). Furthermore, it only uses variables directly related with the data model, and is immune to singular solutions. PLMF is inspired on the well-known Non-negative Matrix Factorization (NMF) approach (Lee and Seung, 2001), which is not applicable directly to the SSS problem, because some factors in the factorization are not positive, as will be made clear below. For simplicity, we will restrict ourselves to the case where the sources are perfectly synchronized.

One should not consider PLMF as a replacement for IPA, but rather as a different approach to a similar problem: PLMF is a model-driven algorithm, whereas IPA is data-driven. As we will show, PLMF has advantages and disadvantages relative to IPA.

This paper is organized as follows. In Sec. 2 we introduce the Phase Locking Factor (PLF) quantity which measures the degree of synchronization of two signals, and show that full synchronization between two signals has a very simple mathematical characterization. Sec. 3 describes the PLMF algorithm in detail. In Sec. 4 we explain how the simulated data was generated and show the results obtained by PLMF. Directions for future work are discussed in Sec. 5. Conclusions are drawn in Sec. 6.

2 PHASE SYNCHRONY

2.1 Phase of a Real-valued Signal

In this paper we tackle the problem of Separation of Synchronous Sources (SSS). The sources are assumed to be synchronous, or phase-locked: thus, one must be able to extract the phase of a given signal. In many real-world applications, such as brain EEG or MEG, the set of measurements available is real-valued. In those cases, to obtain the phase of such measurements, it is usually convenient to construct a set of complex-valued data from them. Two approaches have been used in the literature: complex wavelet transforms (Torrence and Compo, 1998) and the Hilbert transform (Gold et al., 1973).

In this paper we present only results on simulated data, which is directly generated as complex-valued, thus curcumventing this issue.

2.2 Phase-locking Factor

Let $\phi_j(t)$ and $\phi_k(t)$, for $t = 1, \ldots, T$, be the time-dependent phases of signals j and k. The real-valued[2]

[2]"Real-valued" is used here to distinguish from other papers, where the absolute value operator is dropped, hence

Phase Locking Factor (PLF) between those two signals is defined as

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

where $\langle\cdot\rangle$ is the time average operator, and $\mathrm{i}=\sqrt{-1}$. Note that $0 \leq \rho_{jk} \leq 1$. The value $\rho_{jk}=1$ corresponds to two signals that are fully synchronized: their phase lag, defined as $\phi_j(t)-\phi_k(t)$, is constant. The value $\rho_{jk}=0$ is attained if the two phases are not correlated, as long as the observation period T is sufficiently long. Values between 0 and 1 represent partial synchrony. Typically, the PLF values are stored in a PLF matrix $\mathbf{Q}$ such that $\mathbf{Q}(j,k) \equiv \rho_{jk}$. Note that a signal's PLF with itself is trivially equal to 1: thus, for all j, $\rho_{jj}=1$.

2.3 Effect of Mixing on the PLF

The effect of a linear mixing operation on a set of sources which have all pairwise PLFs equal to 1 is now discussed. This effect has a simple mathematical characterization: if $\mathbf{s}(t)$ is a set of such sources, and we define $\mathbf{x}(t) \equiv \mathbf{M}\mathbf{s}(t)$, with $\det(\mathbf{M}) \neq 0$, then the only possibility for the observations $\mathbf{x}$ to have all pairwise PLFs equal to 1 is if $\mathbf{M}$ is a permutation of a diagonal matrix (Almeida et al., 2011a). Equivalently, the only possibility for that is if $\mathbf{x}=\mathbf{s}$ up to permutation and scaling, a typical non-restrictive indeterminancy in source separation problems.

This effect is illustrated in Figure 1, which shows a set of three perfectly synchronized sources and their PLFs. That figure also depicts three signals obtained through a linear mixing of the sources, and their PLFs. These mixtures have PLFs lower than 1, in accordance with the result stated in the previous paragraph (even though the PLF between sources 2 and 3 happens to be rather high, but still not 1).

This property illustrates that separation of these sources is necessary to make any type of inference about their synchrony, as measured through the PLF. If they are not properly separated, the synchrony values measured will not be accurate. On the other hand, established BSS methods such as Independent Component Analysis (ICA) are not adequate for this task, since phase-locked sources are not independent (Almeida et al., 2011a). PLMF is a source separation algorithm tailored specifically for this problem, and it is presented in the next section.

making the PLF a complex quantity (Almeida et al., 2011a).

3 ALGORITHM

We begin with a summary of the notation and definitions used in this section; we then formulate the optimization problem for PLMF and present a table of the algorithm at the end.

3.1 Assumptions and General Formulation

We assume that we have a set of N complex-valued sources $s_j(t)$ for $j=1,\ldots,N$ and $t=1,\ldots,T$. We assume also that N is known. Denote by $\mathbf{S}$ a N by T complex-valued matrix whose (j,t)-th entry is $s_j(t)$. One can easily separate the amplitude and phase components of the sources through $\mathbf{S}=\mathbf{A}\odot\mathbf{\Phi}$, where $\odot$ is the elementwise (or Hadamard) product, $\mathbf{A}$ is a real-valued N by T matrix with its (j,t) element defined as $a_j(t) \equiv |s_j(t)|$, and $\mathbf{\Phi}$ is a N by T complex-valued matrix with its (j,t) element defined as $\Phi_j(t) \equiv e^{\mathrm{i}(\mathrm{angle}(s_j(t)))} \equiv e^{\mathrm{i}\phi_j(t)}$.

The representation of $\mathbf{S}$ in amplitude and phase is, thus far, completely general: it merely represents $\mathbf{S}$ in polar coordinates. We place no constraints on $\mathbf{A}$ other than non-negativity, since it its elements are absolute values of complex numbers. This is consistent with the use of the PLF as a measure of synchrony: the PLF uses no information from the signal amplitudes.

We assume that the sources are perfectly synchronized; as discussed in Section 2.2, in this situation, $\Delta\phi_{jk}(t)=\phi_j(t)-\phi_k(t)$ is constant for all t, for any j and k. Thus, $\mathbf{\Phi}$ can be decomposed as

$$\mathbf{\Phi} \equiv \mathbf{z}\mathbf{f}^{\mathsf{T}}, \quad (2)$$

where $\mathbf{z}$ is a complex-valued column vector of size N containing the relative phase lags of each source, and $\mathbf{f}$ is a complex-valued column vector of size T containing the common oscillation. In simpler terms, if the sources are phase-locked, then $\mathrm{rank}(\mathbf{\Phi})=1$, and the above decomposition is always possible, even though it is not unique. Then, the time evolution of each source's phase is given by $\phi_j(t)=\mathrm{angle}(z_j)+\mathrm{angle}(f_t)$, where z_j and f_t are the j-th entry of $\mathbf{z}$ and the t-th element of $\mathbf{f}$, respectively.

Although one can conceive complex-valued sources where the rows of $\mathbf{A}$ and the vector $\mathbf{f}$ vary rapidly with time, in real-world systems we expect them to vary smoothly; for this reason, as will be seen below, we chose to softly enforce the smoothness of these two variables in PLMF.

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

$$\mathbf{X} \equiv \mathbf{M}\mathbf{S}+\mathbf{N},$$

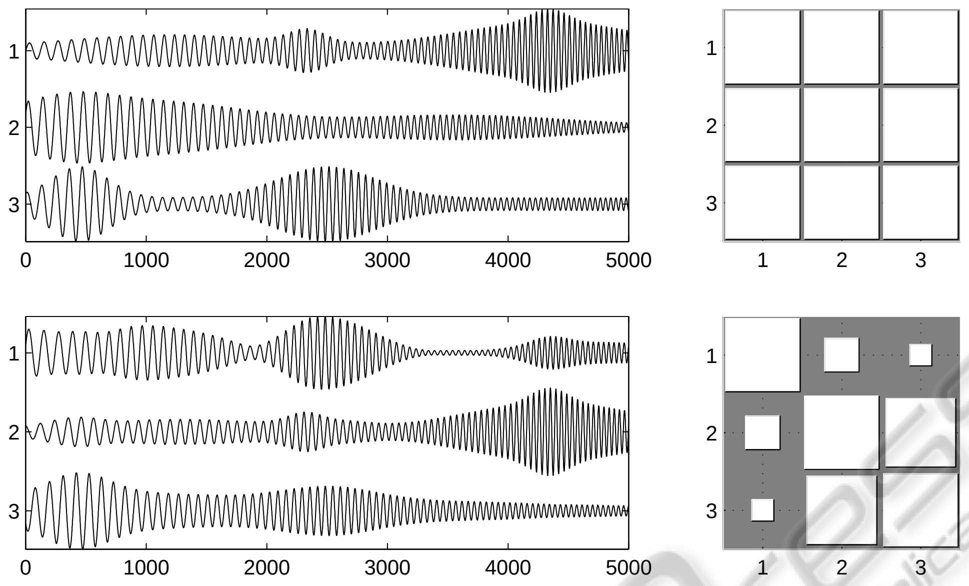

Figure 1: *(Top row)* Three sources *(left)* and PLFs between them *(right)*. *(Bottom row)* Three mixed signals *(left)* and PLFs between them *(right)*. On the right column, the area of the square in position (i, j) is proportional to the PLF between the signals i and j. Therefore, large squares represent PLFs close to 1, while small squares represent values close to zero.

where $\mathbf{X}$ is a P by T matrix containing the measurements, $\mathbf{M}$ is a P by N real-valued mixing matrix and $\mathbf{N}$ is a P by T complex-valued matrix of noise. Our assumption of a real mixing matrix is appropriate in the case of linear and instantaneous mixing, as motivated earlier. We will deal only with the noiseless model, where $\mathbf{N} = 0$, although we then also test how it copes with noisy data.

The goal of PLMF is to recover $\mathbf{S}$ and $\mathbf{M}$ using only $\mathbf{X}$. A simple way to do this is to find $\mathbf{M}$ and $\mathbf{S}$ such that the data misfit, defined as $\frac{1}{2}\left\|\mathbf{X} - \mathbf{M}\big(\mathbf{A} \odot (\mathbf{z}\mathbf{f}^\mathsf{T})\big)\right\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm, is as small as possible. As mentioned above, we also want the estimates of $\mathbf{A}$ and $\mathbf{f}$ to be smooth. Thus, the minimization problem to be solved is given by

$$\min_{\mathbf{M},\mathbf{A},\mathbf{z},\mathbf{f}} \frac{1}{2}\left\|\mathbf{X} - \mathbf{M}\big(\mathbf{A} \odot (\mathbf{z}\mathbf{f}^\mathsf{T})\big)\right\|_F^2 + \\ + \lambda_\mathbf{A}\left\|\mathbf{A}\mathbf{D}_\mathbf{A}\right\|_F^2 + \lambda_\mathbf{f}\left\|\mathbf{D}_\mathbf{f}\mathbf{f}\right\|_2^2, \qquad (3)$$

s.t.: 1) All elements of $\mathbf{M}$ must lie between -1 and +1.
2) All elements of $\mathbf{A}$ must be non-negative.
3) All elements of $\mathbf{z}$ and $\mathbf{f}$ must have unit absolute value.

where $\mathbf{D}_\mathbf{A}$ and $\mathbf{D}_\mathbf{f}$ are the first-order difference operators of appropriate size, such that the entry (j,t) of $\mathbf{A}\mathbf{D}_\mathbf{A}$ is given by $a_j(t) - a_j(t+1)$, and the k-th entry of $\mathbf{D}_\mathbf{f}\mathbf{f}$ is given by $f_k - f_{(k+1)}$. The first term directly measures the misfit between the real data and the product of the estimated mixing matrix and the estimated sources. The second and third terms enforce smoothness of the rows of $\mathbf{A}$ and of the vector $\mathbf{f}$, respectively. These two terms allow for better estimates for $\mathbf{A}$ and $\mathbf{f}$ under additive white noise, since enforcing smoothness is likely to filter the high-frequency components of that noise.

Constraint 2 ensures that $\mathbf{A}$ represents amplitudes, whereas Constraint 3 ensures that $\mathbf{z}$ and $\mathbf{f}$ represent phases. Constraint 1 prevents the mixing matrix $\mathbf{M}$ from exploding to infinity while $\mathbf{A}$ goes to zero. Note that we also penalize indirectly the opposite indeterminancy, where $\mathbf{M}$ goes to zero while $\mathbf{A}$ goes to infinity: that would increase the value of the second term while keeping the other terms constant, as long as the rows of $\mathbf{A}$ do not have all elements equal to each other. Thus, the solution for $\mathbf{M}$ lies on the boundary of the feasible set for $\mathbf{M}$; using this constraint instead of forcing the L_1 norm of each row to be exactly 1, as was done in (Almeida et al., 2011b), makes the subproblem for $\mathbf{M}$ convex, with all the known advantages that this brings (Boyd and Vandenberghe, 2004).

3.2 Optimization

The minimization problem presented in Eq. (3) depends on the four variables $\mathbf{M}$, $\mathbf{A}$, $\mathbf{z}$, and $\mathbf{f}$. Although the minimization problem is not globally convex, it is convex in $\mathbf{A}$ and $\mathbf{M}$ individually, while keeping the other variables fixed. For simplicity, we chose to optimize Eq. (3) in each variable at a time, by first optimizing on $\mathbf{M}$ while keeping $\mathbf{A}$, $\mathbf{z}$ and $\mathbf{f}$ constant; then doing the same for $\mathbf{A}$, followed by $\mathbf{z}$, and then $\mathbf{f}$. This cycle is repeated until convergence. From our experience with the method, the particular order in which the variables are optimized is not critical. Although this algorithm is not guaranteed to converge to a global minimum, we have experienced very few cases of local optima.

In the following, we show that the minimization problem above can be translated into well-known forms (constrained least squares problems) for each of the four variables. We also detail the optimization procedure for each of the four sub-problems. For brevity, we do not distinguish the real variables such as $\mathbf{M}$ from their estimates $\hat{\mathbf{M}}$ throughout this section: in each sub-problem, only one variable is being estimated, while all the others are kept fixed and equal to their current estimates.

3.2.1 Optimization on M

If we define $\mathbf{m} \equiv \mathrm{vec}(\mathbf{M})$ and $\mathbf{x} \equiv \mathrm{vec}(\mathbf{X})$[3], then the minimization sub-problem for $\mathbf{M}$, while keeping all other variables fixed, is equivalent to the following constrained least-squares problem:

$$\min_{\mathbf{m}} \frac{1}{2} \left\| \begin{bmatrix} \mathcal{R}(\mathbf{x}) \\ \mathcal{I}(\mathbf{x}) \end{bmatrix} - \begin{bmatrix} \mathcal{R}(\mathbf{R}) \\ \mathcal{I}(\mathbf{R}) \end{bmatrix} \mathbf{m} \right\|_2^2 \qquad (4)$$
$$\text{s.t.:} \ -1 \le \mathbf{m} \le +1,$$

where $\mathcal{R}(.)$ and $\mathcal{I}(.)$ are the real and imaginary parts, $\mathbf{I}_P$ is the P by P identity matrix, and $\mathbf{R} \equiv [\mathbf{S}^\top \otimes \mathbf{I}_P]$, with $\otimes$ denoting the Kronecker product and $\|\cdot\|_2$ denoting the Euclidean norm. Here, and throughout this paper, all inequalities should be understood in the componentwise sense, *i.e.*, every entry of $\mathbf{M}$ is constrained to be between -1 and +1. For convenience, we used the least-squares solver implemented in the MATLAB Optimization Toolbox to solve this problem, although many other solvers exist.

The main advantage of using the constraint $-1 \le \mathbf{M} \le +1$ is now clear: it is very simply translated into $-1 \le \mathbf{m} \le +1$ after applying the vec(.) operator, remaining a convex constraint, whereas other constraints would be harder to apply.

3.2.2 Optimization on A

The optimization in $\mathbf{A}$ can also be reformulated as a least-squares problem. If $\mathbf{a} \equiv \mathrm{vec}(\mathbf{A})$, the minimization on $\mathbf{A}$ is equivalent to

$$\min_{\mathbf{a}} \frac{1}{2} \left\| \begin{bmatrix} \mathcal{R}(\mathbf{x}) \\ \mathcal{I}(\mathbf{x}) \\ \mathbf{0}_{(N^2-N)} \end{bmatrix} - \begin{bmatrix} \mathcal{R}(\mathbf{K}) \\ \mathcal{I}(\mathbf{K}) \\ \sqrt{\lambda_{\mathbf{A}}}\mathbf{D}_{\mathbf{A}} \otimes \mathbf{I}_N \end{bmatrix} \mathbf{a} \right\|_2^2 \qquad (5)$$
$$\text{s.t.:} \quad \mathbf{a} \ge 0,$$

where $\mathbf{K} \equiv [(\mathrm{Diag}(\mathbf{f}) \otimes \mathbf{M})\mathrm{Diag}(\mathbf{z}_0)]$, $\mathbf{0}_{(N^2-N)}$ is a column vector of size (N^2-N), filled with zeros, and Diag(.) is a square diagonal matrix of appropriate dimension having the input vector on the main diagonal. We again use the built-in MATLAB solver to solve this sub-problem.

3.2.3 Optimization on z

The minimization problem in $\mathbf{z}$ with no constraints is equivalent to:

$$\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{O}\mathbf{z} - \mathbf{x}\|_2^2 \quad \text{with} \quad \mathbf{O} = \begin{bmatrix} f_1\mathbf{M}\ \mathrm{Diag}(\mathbf{a}(1)) \\ f_2\mathbf{M}\ \mathrm{Diag}(\mathbf{a}(2))) \\ \vdots \\ f_T\mathbf{M}\ \mathrm{Diag}(\mathbf{a}(T)) \end{bmatrix}, \qquad (6)$$

where f_t is the t-th entry of $\mathbf{f}$, and $\mathbf{a}(t)$ is the t-th column of $\mathbf{A}$. Usually, the solution of this system will not obey the unit absolute value constraint. To circumvent this, we solve this unconstrained linear system and afterwards normalize $\mathbf{z}$ for all sources j and time instants t, by transferring its absolute value onto variable $\mathbf{A}$:

$$a_j(t) \leftarrow |z_j| a_j(t) \quad \text{and} \quad z_j \leftarrow z_j/|z_j|.$$

It is easy to see that the new $\mathbf{z}$ obtained after this normalization is still a global minimizer of (6) (where the new value of $\mathbf{A}$ should be used).

3.2.4 Optimization on f

Let $\tilde{\mathbf{x}} \equiv \mathrm{vec}(\mathbf{X}^\top)$. The minimization problem in $\mathbf{f}$ with no constraints can be shown to be equivalent to

$$\min_{\mathbf{f}} \frac{1}{2} \left\| \begin{bmatrix} \mathbf{P} \\ \sqrt{\lambda_{\mathbf{f}}}\mathbf{D}_{\mathbf{f}} \end{bmatrix} \mathbf{f} - \begin{bmatrix} \tilde{\mathbf{x}} \\ \mathbf{0}_{(N-1)} \end{bmatrix} \right\|_2^2 \qquad (7)$$
$$\text{with} \quad \mathbf{P} = \begin{bmatrix} \sum_j m_{1j} z_j \mathrm{Diag}(\mathbf{a}_j) \\ \sum_j m_{2j} z_j \mathrm{Diag}(\mathbf{a}_j) \\ \vdots \\ \sum_j m_{Pj} z_j \mathrm{Diag}(\mathbf{a}_j) \end{bmatrix},$$

where f_t is the t-th entry of $\mathbf{f}$, m_{ij} is the (i,j) entry of $\mathbf{M}$, z_j is the j-th entry of $\mathbf{z}$, and $\mathbf{a}_j$ is the j-th row of $\mathbf{A}$.

As in the sub-problem for $\mathbf{z}$, in general the solution of this system will not obey the unit absolute value constraint. Thus, we perform a similar normalization, given by

$$a_j(t) \leftarrow |f_t| a_j(t) \quad \text{and} \quad f_t \leftarrow f_t/|f_t|. \qquad (8)$$

Note that unlike the previous case of the optimization for $\mathbf{z}$, this normalization changes the cost function, in

[3]The vec(.) operator stacks the columns of a matrix into a column vector.

particular the term $\lambda_\mathbf{f} \|\mathbf{D_f f}\|_2^2$. Therefore, there is no guarantee that after this normalization we have found a global minimum for **f**.

For this reason, we construct a vector of angles $\beta \equiv \text{angle}(\mathbf{f})$ and minimize the cost function (3) as a function of β, using 20 iterations of Newton's algorithm. Although infinitely many values of β correspond to a given **f**, any of those values is suitable. The advantage of using this new variable is that there are no constraints in β, so the Newton algorithm can be used freely. Thus, the normalized solution of the linear system in (7) can be considered simply as an initialization for the Newton algorithm on β, which in most conditions can find a local minimum.

On the first time that the Newton algorithm is run, it is initialized using the unconstrained problem (7) and the ensuing normalization (8). On the second and following times that it is run, the result of the previous minimization on **f** is used as initial value.

3.2.5 Phase Locked Matrix Factorization

The consecutive cycling of optimizations on **M**, **A**, **z** and **f** constitutes the Phase Locked Matrix Factorization (PLMF) algorithm. A summary of this algorithm is presented below.

```
PHASE LOCKED MATRIX FACTORIZATION
 1:  Input data $\mathbf{X}$
 2:  Input random initializations $\hat{\mathbf{M}}, \hat{\mathbf{A}}, \hat{\mathbf{z}}, \hat{\mathbf{f}}$
 3:  for iter $\in \{1,2,\ldots,\text{MaxIter}\}$, do
 4:    Solve the constrained problem in Eq. (4)
 5:    Solve the constrained problem in Eq. (5)
 6:    Solve the unconstrained system in Eq. (6)
 7:    $a_j(t) \leftarrow |z_j| a_j(t)$ and $z_j \leftarrow z_j/|z_j|$, $j = 1,\ldots,N$
 8:    if iter = 1
 9:      Solve the unconstrained system in Eq. (7)
10:      $a_j(t) \leftarrow |f_t| a_j(t)$ and $f_t \leftarrow f_t/|f_t|$, $t = 1,\ldots,T$
11:      Optimize $\beta \equiv \text{angle}(\mathbf{f})$ with Newton algorithm
         (use result of step 10 as initialization)
12:    else
13:      Optimize $\beta \equiv \text{angle}(\mathbf{f})$ with Newton algorithm
         (use Newton algorithm from (iter-1) as init.)
14:  end for
```

4 SIMULATION AND RESULTS

In this section we show results on small simulated datasets, demonstrating that PLMF can correctly factor the data **X** into a mixing matrix **M**, amplitudes **A**, and phases **z** and **f**. Despite deriving PLMF for the noiseless case, we will also test its robustness to a small noisy perturbation.

4.1 Data Generation

We generate the data directly from the model $\mathbf{X} = \mathbf{MS}$, with $\mathbf{S} = \mathbf{A} \odot \Phi = \mathbf{A} \odot (\mathbf{z}\mathbf{f}^\mathsf{T})$, taking $N = 2$ and $P = 4$. The number of time samples is $T = 100$. **M** is taken as a random matrix with entries uniformly distributed between -1 and +1. We then normalize **M** so that the entry with the largest absolute value is ± 1. Each row of **A** (*i.e.* each source's amplitude) is generated as a sum of a constant baseline and 2 to 5 Gaussians with random mean and random variance. **z** is generated by uniformly spacing the N sources in the interval $\left[0, \frac{2\pi}{3}\right]$.[4] **f** is generated as a complex sinusoid with angular frequency 0.06 in the first half of the observation period, and angular frequency 0.04 in its second half, in a way that **f** has no discontinuities. **X** is then generated according to the data model: $\mathbf{X} = \mathbf{M}(\mathbf{A} \odot (\mathbf{z}\mathbf{f}^\mathsf{T}))$.

The initial values for the estimated variables are all random: elements of $\hat{\mathbf{M}}$ and $\hat{\mathbf{A}}$ are drawn from the Uniform([0,1]) distribution ($\hat{\mathbf{M}}$ is then normalized in the same way as **M**), while the elements of $\hat{\mathbf{z}}$ are of the form $e^{i\alpha}$ with α taken from the Uniform $\left(\left[0, \frac{\pi}{2}\right]\right)$ distribution. The elements of **f** are also of the form $e^{i\beta}$, with β uniformly distributed between 0 and 2π.

We generate 100 datasets of each of two types of datasets with the following features:

- Dataset 1: 2 sources, 4 sensors, 100 time points, no noise.
- Dataset 2: exactly the same data as dataset 1, plus complex Gaussian noise with standard deviation 0.1, added after the mixture (additive noise).

4.2 Quality Measures

$\hat{\mathbf{M}}$ can be compared with **M** through the gain matrix $\mathbf{G} \equiv \hat{\mathbf{M}}^{+}\mathbf{M}$, where $\hat{\mathbf{M}}^{+}$ is the Moore-Penrose pseudo-inverse of $\hat{\mathbf{M}}$ (Ben-Israel and Greville, 2003). This is the same as $\hat{\mathbf{M}}^{-1}\mathbf{M}$ if the number of sensors is equal to the number of sources. If the estimation is well done, the gain matrix should be close to a permutation of the identity matrix. After manually compensating a possible permutation of the estimated sources, we compare the sum of the squares of the diagonal elements of **G** with the sum of the squares of its off-diagonal elements. This is a criterion not unlike the well-known Signal-to-Interference Ratio (SIR), but our criterion depends only on the true and estimated mixing matrices. Since this criterion is inspired on the SIR, we call this criterion pseudo-SIR, or pSIR.

[4] This choice of **z** is done to ensure that the sources never have phase lags close to 0 or π, which violate the mild assumptions mentioned in Section 2.3 (Almeida et al., 2011a).

Table 1: Comparison of the estimated mixing matrix $\hat{\mathbf{M}}$ with the true mixing matrix $\mathbf{M}$ through the pseudo-SIR of the gain matrix $\mathbf{G} \equiv \hat{\mathbf{M}}^{+}\mathbf{M}$. For zero noise (dataset 1), the estimation is quite good, and the performance hit due to the presence of noise (dataset 2) is minimal.

Data	pSIR (dB)
Dataset 1	21.2 ± 11.3
Dataset 2	20.7 ± 11.7

Also, $\hat{\mathbf{A}}$ will be compared to $\mathbf{A}$ through visual inspection for one dataset with a pSIR close to the average pSIR of the 100 datasets.

4.3 Results

We did not implement a convergence criterion; we simply do 400 cycles of the optimization on $\mathbf{M}$, $\mathbf{A}$, $\mathbf{z}$ and $\mathbf{f}$ using $\lambda_{\mathbf{A}} = 3$ and $\lambda_{\mathbf{f}} = 1$. The mean and standard deviation of the pSIR criterion are presented in Table 1. Figure 2 shows the results of the estimation of the source amplitudes for one representative dataset, showing that $\hat{\mathbf{A}}$ is quite close to the real $\mathbf{A}$ for both the noiseless and the noisy datasets. Note that if noise is present, it is impossible to recreate the original amplitudes as they are only present in the data corrupted by noise: one can thus only estimate the corrupted amplitudes. If desired, a simple low-pass filtering procedure can closely recreate the original amplitudes.

These results illustrate that PLMF can separate phase-locked sources in both the noiseless and the noisy condition. Furthermore, they show that the performance hit due to the presence of a small amount of noise is minimal, suggesting that PLMF has good robustness against small perturbations.

5 DISCUSSION

The above results show that this approach has a high potential, although some limitations must be addressed to turn this algorithm practical for real-world applications.

The values of the parameters that we chose were somewhat *ad hoc*. Nevertheless, it is likely that knowledge of the covariance of the additive noise allows one to choose the values of these parameters optimally.

One incomplete aspect of PLMF is its lack of a stopping criterion; in fact, the results shown in Table 1 could be considerably improved if the number of iterations is increased to, say, 1000, although that is not the case for all of the 100 datasets. We did not tackle this aspect due to lack of time; however, the data misfit (first term of the cost function) can probably be used to design a decent criterion.

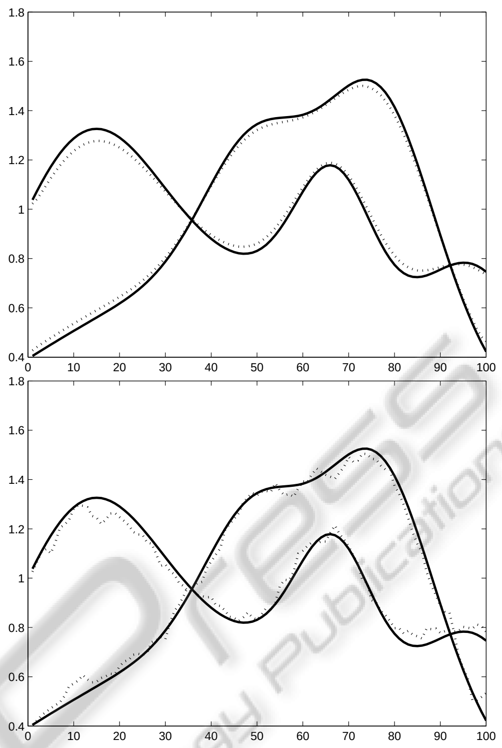

Figure 2: Visual comparison of the estimated amplitudes $\hat{\mathbf{A}}$ (dashed lines) with the true amplitudes $\mathbf{A}$ (solid lines), for a representative dataset, after both are normalized so that they have unit means over the observation period. (Top) Results for dataset 1: the two estimated amplitudes are close to the true values. (Bottom) Results for dataset 2: due to the presence of noise, it is impossible for the two estimated amplitudes to coincide perfectly with the true ones, but nevertheless the estimated amplitudes follow the real ones very closely.

If the sources are not perfectly phase-locked, their pairwise phase differences $\Delta\phi_{ij}$ are not constant in time and therefore one cannot represent the source phases by a single vector of phase lags $\mathbf{z}$ and a single vector $\mathbf{f}$ with a common oscillation. In other words, $\mathbf{\Phi}$ will have a rank higher than 1 (in most cases, it will have the maximum possible rank, which is N), which makes a representation $\mathbf{\Phi} = \mathbf{z}\mathbf{f}^{\mathsf{T}}$ impossible. We are investigating a way to estimate the "most common" phase oscillation $\mathbf{f}$ from the data $\mathbf{X}$, after which PLMF can be used to initialize a more general algorithm that estimates the full $\mathbf{\Phi}$. We are currently testing also a more general algorithm, which optimizes $\mathbf{\Phi}$ with a gradient descent algorithm. Yet, it is somewhat prone to local minima, as one would expect for optimizing variables of size $NT = 200$. A good initialization is likely to alleviate this problem.

Another limitation of PLMF is the indetermination that arises if two sources have $\Delta\phi_{ij} = 0$ or π. In that case, the problem becomes ill-posed, as was already the case in IPA (Almeida et al., 2011a). In fact, using sources with $\Delta\phi_{ij} < \frac{\pi}{10}$ starts to deteriorate the results of PLMF, even with zero noise.

One further aspect which warrants discussion is PLMF's identifiability. If we find two factorizations such that $\mathbf{X} = \mathbf{M}_1\big(\mathbf{A}_1 \odot (\mathbf{z}_1\mathbf{f}_1^\mathsf{T})\big) = \mathbf{M}_2\big(\mathbf{A}_2 \odot (\mathbf{z}_2\mathbf{f}_2^\mathsf{T})\big)$ (*i.e.*, two factorizations which perfectly describe the same data $\mathbf{X}$), does that imply that $\mathbf{M}_1 = \mathbf{M}_2$, and similar equalities for the other variables? It is quite clear that the answer is negative: the usual indeterminancies of BSS apply to PLMF as well, namely the indeterminancies of permutation, scaling, and sign of the estimated sources. There is at least one further indeterminancy: starting from a given solution $\mathbf{X} = \mathbf{M}_1\big(\mathbf{A}_1 \odot (\mathbf{z}_1\mathbf{f}_1^\mathsf{T})\big)$, one can always construct a new one by defining $\mathbf{z}_2 \equiv e^{i\psi}\mathbf{z}_1$ and $\mathbf{f}_2 \equiv e^{-i\psi}\mathbf{f}_1$, while keeping $\mathbf{M}_2 \equiv \mathbf{M}_1$ and $\mathbf{A}_2 \equiv \mathbf{A}_1$. Note that $\mathbf{S}_1 = \mathbf{A}_1 \odot (\mathbf{z}_1\mathbf{f}_1^\mathsf{T}) = \mathbf{A}_2 \odot (\mathbf{z}_2\mathbf{f}_2^\mathsf{T}) = \mathbf{S}_2$, thus the estimated sources are exactly the same.

6 CONCLUSIONS

We presented an improved version of Phase Locked Matrix Factorization (PLMF), an algorithm that directly tries to reconstruct a set of measured signals as a linear mixing of phase-locked sources, by factorizing the data into a product of four variables: the mixing matrix, the source amplitudes, their phase lags, and a common oscillation.

PLMF is now able to estimate the sources even when their common oscillation is unknown – an advantage which greatly increases the applicability of the algorithm. Furthermore, the sub-problem for $\mathbf{M}$ is now convex, and the sub-problems for $\mathbf{z}$ and $\mathbf{f}$ are tackled in a more appropriate manner which should find local minima. The results show good performance for the noiseless case and good robustness to small amounts of noise. The results show as well that the proposed algorithm is accurate and can deal with low amounts of noise, under the assumption that the sources are fully phase-locked, even if the common oscillation is unknown. This generalization brings us considerably closer to being able to solve the Separation of Synchronous Sources (SSS) problem in real-world data.

ACKNOWLEDGEMENTS

This work was partially funded by the DECA-Bio project of the Institute of Telecommunications, and by the Academy of Finland through its Centres of Excellence Program 2006-2011.

REFERENCES

Almeida, M., Bioucas-Dias, J., and Vigário, R. (2010). Independent phase analysis: Separating phase-locked subspaces. In *Proceedings of the Latent Variable Analysis Conference*.

Almeida, M., Schleimer, J.-H., Bioucas-Dias, J., and Vigário, R. (2011a). Source separation and clustering of phase-locked subspaces. *IEEE Transactions on Neural Networks*, 22(9):1419–1434.

Almeida, M., Vigario, R., and Bioucas-Dias, J. (2011b). Phase locked matrix factorization. In *Proc. of the EUSIPCO conference*.

Ben-Israel, A. and Greville, T. (2003). *Generalized inverses: theory and applications*. Springer-Verlag.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Gold, B., Oppenheim, A. V., and Rader, C. M. (1973). Theory and implementation of the discrete hilbert transform. *Discrete Signal Processing*.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons.

Lee, D. and Seung, H. (2001). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13, pages 556–562.

Nunez, P. L., Srinivasan, R., Westdorp, A. F., Wijesinghe, R. S., Tucker, D. M., Silberstein, R. B., and Cadusch, P. J. (1997). EEG coherency I: statistics, reference electrode, volume conduction, laplacians, cortical imaging, and interpretation at multiple scales. *Electroencephalography and clinical Neurophysiology*, 103:499–515.

Palva, J. M., Palva, S., and Kaila, K. (2005). Phase synchrony among neuronal oscillations in the human cortex. *Journal of Neuroscience*, 25(15):3962–3972.

Pikovsky, A., Rosenblum, M., and Kurths, J. (2001). *Synchronization: A universal concept in nonlinear sciences*. Cambridge Nonlinear Science Series. Cambridge University Press.

Schoffelen, J.-M., Oostenveld, R., and Fries, P. (2008). Imaging the human motor system's beta-band synchronization during isometric contraction. *NeuroImage*, 41:437–447.

Torrence, C. and Compo, G. P. (1998). A practical guide to wavelet analysis. *Bull. of the Am. Meteorological Society*, 79:61–78.

Uhlhaas, P. J. and Singer, W. (2006). Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology. *Neuron*, 52:155–168.

Vigário, R., Särelä, J., Jousmäki, V., Hämäläinen, M., and Oja, E. (2000). Independent component approach to the analysis of EEG and MEG recordings. *IEEE Trans. On Biom. Eng.*, 47(5):589–593.

Ziehe, A. and Müller, K.-R. (1998). TDSEP - an efficient algorithm for blind separation using time structure. In *International Conference on Artificial Neural Networks*, pages 675–680.