

PHASE LOCKED MATRIX FACTORIZATION

Miguel Almeida^{1,2}, Ricardo Vigário², José Bioucas Dias¹

¹Institute of Telecommunications, Superior Technical Institute, Technical University of Lisbon, Portugal

²Department of Information and Computer Science, School of Science, Aalto University, Helsinki, Finland

email: malmeida@lx.it.pt, ricardo.vigario@aalto.fi, bioucas@lx.it.pt

web: ¹www.lx.it.pt, ²ics.tkk.fi/en/

ABSTRACT

We present a novel approach to separate linearly mixed dependent sources that are phase-locked. The separation is done through a minimization problem involving three variables (the mixing matrix, the source time-dependent amplitudes, and their relative phases). Results obtained in toy data sets show that this algorithm is very fast, that it estimates the mixing matrix with remarkable precision even with considerable amounts of noise, and that the sources are also correctly estimated. We interpret these results as a “proof-of-concept” that this approach is valid and discuss the necessary improvements to deal with more general situations.

1. INTRODUCTION

Synchrony is an increasingly studied topic in modern science. It is a relevant topic for several reasons, including the availability of an elegant yet deep mathematical framework that is applicable to many domains where synchrony is present, including laser interferometry, the gravitational pull of stellar objects, and the human brain [14].

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 [13, 15]. Also, several pathologies such as autism, Alzheimer and Parkinson are associated with a disruption in the synchronization profile of the brain [18].

To formally model synchrony phenomena one usually uses a special type of dynamical system called a self-sustained oscillator. Oscillating dynamical systems have been used extensively to model the behavior of neurons [9]. A self-sustained oscillator is a dynamical system with an internal energy source, which exhibits periodic motion when isolated from the rest of the universe [14]. In terms of the system’s phase space [17], self-sustained oscillators have a periodic limit cycle¹ which is stable in at least a small neighborhood of that cycle. The position of the system along this limit cycle is called the oscillator’s phase. Perturbations along the limit cycle do not decay and permanently affect the phase, while perturbations in orthogonal directions decay exponentially. In other words, this limit cycle has a zero Lyapunov exponent in the direction tangent to the cycle, and negative exponents in all directions orthogonal to it [17].

The stability of the oscillator’s limit cycle has deep implications when such oscillators are coupled with one another through a weak interaction. A weak interaction will not push

the system out of its limit cycle, but it can push the system forward or pull it backward along the cycle, permanently affecting its phase. If the interactions between N self-sustained oscillators are weak and attractive, the time dynamics of their phases can be described by the Kuramoto model [10]:

$$\dot{\phi}_i(t) = \omega_i + \frac{1}{N} \sum_{j=1}^N \kappa_{ij} \sin [\phi_j(t) - \phi_i(t)], \quad (1)$$

where $t \in \mathbb{R}$, ω_i is the intrinsic frequency of oscillator i , and κ_{ij} is the coupling coefficient between oscillators i and j , which must be positive for attractive interactions. If ϕ_j is slightly larger than ϕ_i , then oscillator i will move slightly faster because of the interaction with oscillator j . Conversely, if ϕ_j is slightly smaller than ϕ_i , then oscillator i will be slowed down by oscillator j . In both cases, this interaction tends to push the phases of the oscillators toward one another. Synchronization will occur if the coupling is strong enough [14, 16].

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 brain electrophysiological signals (EEG and MEG) and other real-world situations, individual oscillator signals are not directly measurable except in very rare situations, and one has only access to a superposition of the sources.² In fact, EEG and MEG signals measured in one sensor contain components coming from several brain regions [12]. In this case, spurious synchrony occurs, as we have shown in previous work [3, 2].

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 approximation in brain signals [19]. 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 problem is ill-posed, thus one must also make some assumptions on the sources, such as statistical independence in Independent Component Analysis (ICA) [8]. However, in our case, independence of the sources is not a valid assumption, because phase-locked sources are highly dependent. In this paper we address the problem of how to separate these dependent sources.

We have already addressed a more general problem where the sources are organized in subspaces, with sources

¹A limit cycle is a closed 1-dimensional curve in the phase space of the system. It can be easily shown that such a curve must be simple, i.e., that it cannot intersect itself. This immediately implies that self-sustained oscillators must be dynamical systems of at least dimension 2 [17].

²In EEG and MEG, the sources are not individual neurons, whose oscillations are too weak to be detected from outside the scalp. In this case, the sources are populations of closely located neurons oscillating together.

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 [1], with moderate levels of added Gaussian white noise [3], and with moderate amounts of phase noise [2]. In summary, IPA uses TDSEP [20] to separate the subspaces from one another and then uses an optimization procedure to complete the intra-subspace separation. Although IPA performs well for the noiseless case for various types of sources and subspace structures, and can 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, and it has a regularization term that depends on the estimated mixing matrix and not on the data itself. On the other hand, IPA deals well with subspaces of phase-locked sources and with sources that are not perfectly phase-locked [3].

Our goal in this paper is to develop an alternative technique, named Phase Locked Matrix Factorization (PLMF) for the intra-subspace separation problem that can deal with higher amounts of noise and with non-square mixing matrices (more measurements than sources), and that only uses variables directly related with the data model. Our approach is inspired on the well-known Non-negative Matrix Factorization approach [11], which is not applicable directly to our problem, because some factors in our factorization are not positive, as will be clear below.

For simplicity, we will restrict ourselves to the case where the sources are perfectly synchronized. One should not consider PMLF as a replacement for IPA, but rather as a different approach to a similar problem. As we will show, PMLF has advantages and disadvantages relative to IPA. It should be reinforced that the algorithm presented here assumes nothing specific of brain signals, and should work in any situation where phase-locked sources are mixed linearly.

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 new algorithm in detail. In Sec. 4 we explain how the simulated data was generated and show the results obtained by PMLF. The current limitations of PMLF and directions for future work are discussed in Sec. 5. We draw conclusions in Sec. 6.

2. PHASE SYNCHRONY

2.1 Phase of a real-valued signal

In this paper we tackle a problem of Source Separation (SS) of dependent components. The sources are assumed to be phase-locked, which in particular means that one must have a way to define the phase of a signal. In most real-world applications, such as brain EEG or MEG, the set of measurements available is real-valued. In those cases, to analyse the phase of a signal, it is usually convenient to construct a set of complex-valued data from the original real-valued signals. Two approaches have been used in the literature: complex wavelet transforms and the Hilbert transform.

In this paper we deal only with simulated data, so we generate the complex signals directly and circumvent this issue.

2.2 Phase-Locking Factor

Given two oscillators with phases $\phi_j(t)$ and $\phi_k(t)$ for $t = 1, \dots, T$, the real-valued³ Phase Locking Factor (PLF) between those two oscillators is defined as

$$\rho_{jk} \equiv \left| \frac{1}{T} \sum_{t=1}^T e^{i[\phi_j(t) - \phi_k(t)]} \right| = \left| \langle e^{i(\phi_j - \phi_k)} \rangle \right|, \quad (2)$$

where $\langle \cdot \rangle$ is the time average operator. The PLF obeys $0 \leq \rho_{jk} \leq 1$. The value $\rho_{jk} = 1$ corresponds to two oscillators that are fully synchronized (their phase lag is constant). The value $\rho_{jk} = 0$ is attained if the two oscillators' 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 $\mathbf{Q}(j, k) = \rho_{jk}$.

In Sec. 1 we mentioned that we assumed that the sources are perfectly synchronized. In mathematical terms, we now rephrase it as $\rho_{jk} = 1$ for all j and k . This immediately implies that $\phi_j(t) - \phi_k(t)$ is constant in time.

3. ALGORITHM

This section details the Phase Locked Matrix Factorization algorithm. We start by presenting the notation and definitions used throughout this section. We then formulate the optimization problem used for PMLF and present a summary table of PMLF 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 $j = 1, \dots, N$ and $t = 1, \dots, T$ that are perfectly phase-locked. We also assume that N is known. Let \mathbf{S} be 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 through $\mathbf{S} = \mathbf{A} \odot \mathbf{Z}$, where \odot is the elementwise (or Hadamard) product, \mathbf{A} is a real-valued N by T matrix with $a_j(t) \equiv |s_j(t)|$, and \mathbf{Z} is a N by T complex-valued matrix with $z_j(t) \equiv e^{i(\text{angle}(s_j(t)))} \equiv e^{i\phi_j(t)}$.

Since the sources are phase-locked, $\Delta\phi_{jk}(t) = \phi_j(t) - \phi_k(t)$ is constant for all t , for any j and k . Thus, one can extract the time-dependent phase oscillation $\phi(t)$ that is common to all the sources, and represent the sources as

$$\mathbf{S} \equiv \mathbf{A} \odot (\Phi \odot \mathbf{Z}_0) = \Phi \odot \mathbf{A} \odot \mathbf{Z}_0, \quad (3)$$

where Φ is a N by T matrix whose (j, t) -th entry is given by $\Phi_j(t) = e^{i\phi(t)}$ (it does not depend on the source j , thus all its rows are equal to each other) containing the common time-dependence of the oscillations, and \mathbf{Z}_0 is constructed as

$$\mathbf{Z}_0 \equiv \underbrace{[\mathbf{z}_0 \ \mathbf{z}_0 \ \dots \ \mathbf{z}_0]}_{T \text{ copies of } \mathbf{z}_0}, \quad (4)$$

where $\mathbf{z}_0 = [e^{i\phi_1}, \dots, e^{i\phi_N}]^T$ is a complex vector containing the relative phases of each source.

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

$$\mathbf{X} \equiv \mathbf{M}\mathbf{S} + \mathbf{N} = \mathbf{M}(\mathbf{A} \odot \mathbf{Z}) + \mathbf{N} = \Phi \odot \mathbf{M}(\mathbf{A} \odot \mathbf{Z}_0) + \mathbf{N}$$

³The term "real-valued" is used here to distinguish from other phase-based algorithms where we drop the absolute value operator, hence making the PLF a complex quantity [3].

where \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 case $\mathbf{N} = 0$. In this case, one can also remove from \mathbf{X} the common phase oscillation: $\mathbf{X} \equiv \Phi \odot \mathbf{X}_0$. However, \mathbf{X}_0 does not have all columns equal to each other, because one of the factors in \mathbf{X}_0 is the time-dependent amplitudes \mathbf{A} .

Our goal is to minimize the cost function

$$\frac{1}{2} \|\mathbf{X} - \mathbf{M}(\mathbf{A} \odot \mathbf{Z})\|_F^2,$$

which is the optimal Maximum Likelihood Estimator (MLE) of the matrices \mathbf{M} , \mathbf{A} , and \mathbf{Z} for i.i.d. Gaussian additive noise. By factoring out the matrix Φ which has all elements with unit absolute value, the minimization problem can be written as

$$\min_{\mathbf{M}, \mathbf{A}, \mathbf{Z}_0} \frac{1}{2} \|\mathbf{X}_0 - \mathbf{M}(\mathbf{A} \odot \mathbf{Z}_0)\|_F^2, \quad (5)$$

- s.t.: 1) Each row of \mathbf{M} must have unit L_1 norm.
 2) All elements of \mathbf{A} must be non-negative.
 3) All elements of \mathbf{Z}_0 must have unit absolute value.
 4) All columns of \mathbf{Z}_0 must be equal (as in Eq. (4)).

Constraints 2 and 3 ensure that matrices \mathbf{A} and \mathbf{Z}_0 represent amplitudes and phases, and constraint 4 ensures that all the sources are phase-locked. Constraint 1 prevents the mixing matrix \mathbf{M} from diverging while \mathbf{A} goes to zero and vice-versa.⁴ Note that this is still an MLE for \mathbf{M} , \mathbf{A} , and \mathbf{Z}_0 : the complex Gaussian noise \mathbf{N} has rotational symmetry at every time point, and factoring out the matrix Φ is simply performing a rotation at each time point.

3.2 Optimization

The minimization problem presented in Eq. (5) depends on the three variables \mathbf{M} , \mathbf{A} , \mathbf{Z}_0 . Although the minimization problem is not globally convex, it is convex in \mathbf{A} while keeping the other variables fixed. For this reason, we chose to optimize it in each variable at a time, by first optimizing on \mathbf{M} while keeping \mathbf{A} and \mathbf{Z}_0 constant, then doing the same for \mathbf{A} , then for \mathbf{Z}_0 , and repeating the cycle until convergence. From our experience with the method, the particular order in which the variables are optimized is not important. Although this algorithm is not guaranteed to converge to a global minimum, we have experienced very good results in practice.

In the following we show that the minimization problem above can be translated into well-known forms (least squares problems and linear systems) for each variable.

3.2.1 Optimization on \mathbf{M}

If we define $\mathbf{m} \equiv \text{vec}(\mathbf{M})$ and $\mathbf{x}_0 \equiv \text{vec}(\mathbf{X}_0)$ ⁵, then the minimization on \mathbf{M} with no constraints is equivalent to the following least-squares problem:

$$\min_{\mathbf{m}} \frac{1}{2} \left\| \begin{bmatrix} \mathcal{R}(\mathbf{x}_0) \\ \mathcal{I}(\mathbf{x}_0) \end{bmatrix} - \begin{bmatrix} \mathcal{R}(\mathbf{R}) \\ \mathcal{I}(\mathbf{R}) \end{bmatrix} \mathbf{m} \right\|_2^2 \quad (6)$$

⁴A unit L_2 norm constraint could have been used instead. Both versions yield non-convex problems, so there is little reason to choose one over the other.

⁵The $\text{vec}(\cdot)$ operator stacks the columns of a matrix into a column vector.

where $\mathcal{R}(\cdot)$ and $\mathcal{I}(\cdot)$ are the real and imaginary parts, \mathbf{I}_P is the P by P identity matrix, and $\mathbf{R} \equiv [(\mathbf{A}^T \odot \mathbf{Z}_0^T) \otimes \mathbf{I}_P]$, with \otimes denoting the Kronecker product. For convenience, we used the least-squares solver implemented in the MATLAB Optimization Toolbox to solve this unconstrained problem, although many other solvers exist.

The unit L_1 norm constraint is not a convex constraint. We instead solve the unconstrained problem above and, after a solution is found, convert \mathbf{m} back to \mathbf{M} and divide the i -th row of \mathbf{M} by its L_1 norm.

3.2.2 Optimization on \mathbf{A}

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

$$\min_{\mathbf{a}} \frac{1}{2} \left\| \begin{bmatrix} \mathcal{R}(\mathbf{x}_0) \\ \mathcal{I}(\mathbf{x}_0) \end{bmatrix} - \begin{bmatrix} \mathcal{R}(\mathbf{N}) \\ \mathcal{I}(\mathbf{N}) \end{bmatrix} \mathbf{a} \right\|_2^2 \quad \text{s.t.} \quad \mathbf{a} \geq 0 \quad (7)$$

where $\mathbf{N} \equiv [(\mathbf{I}_P \otimes \mathbf{M}) \text{Diag}(\mathbf{z}_0)]$, and $\text{Diag}(\cdot)$ is a square diagonal matrix of appropriate dimension having the input vector on the main diagonal. We use SUNSAL [6] to perform this optimization; in our experiments, SUNSAL was considerably faster than the MATLAB built-in functions.

3.2.3 Optimization on \mathbf{z}_0

The minimization problem in \mathbf{Z}_0 can be shown to be equivalent to solving the linear system

$$\mathbf{O} \mathbf{z}_0 = \mathbf{x}_0 \quad \text{with} \quad \mathbf{O} = \begin{bmatrix} \mathbf{M} \text{Diag}(\mathbf{a}(1)) \\ \mathbf{M} \text{Diag}(\mathbf{a}(2)) \\ \vdots \\ \mathbf{M} \text{Diag}(\mathbf{a}(T)) \end{bmatrix}, \quad (8)$$

with the same constraints, where $\mathbf{a}(t)$ 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 the unconstrained linear system instead, and afterwards we normalize \mathbf{z}_0 for all sources i and time instants t ,

$$\mathbf{a}_i(t) \leftarrow |\mathbf{z}_{0,i}| \mathbf{a}_i(t) \quad \text{and} \quad \mathbf{z}_{0,i} \leftarrow \mathbf{z}_{0,i} / |\mathbf{z}_{0,i}|.$$

3.2.4 Phase Locked Matrix Factorization

While solving unconstrained variants of a problem and then enforcing a normalization is a suboptimal procedure, it is guaranteed to lower the cost function, and we have found that it works well in practice. The consecutive cycling of optimizations on \mathbf{M} , \mathbf{A} and \mathbf{z}_0 constitutes the Phase Locked Matrix Factorization (PLMF). A summary of this algorithm is presented below.

PHASE LOCKED MATRIX FACTORIZATION	
1:	Input data with common oscillation removed $\hat{\mathbf{X}}_0$
2:	Input random initializations $\hat{\mathbf{M}}, \hat{\mathbf{A}}, \hat{\mathbf{z}}_0$
3:	for iter $\in \{1, 2, \dots, \text{MaxIter}\}$, do
4:	Solve the unconstrained least squares problem in Eq. (6)
5:	Normalize the rows of \mathbf{M} to have unit L_1 norm
5:	Solve the constrained least squares problem in Eq. (7)
6:	Solve the unconstrained linear system in Eq. (8)
7:	$\mathbf{a}_i(t) \leftarrow \mathbf{z}_{0,i} \mathbf{a}_i(t)$ and $\mathbf{z}_{0,i} \leftarrow \mathbf{z}_{0,i} / \mathbf{z}_{0,i} $ for $i = 1, \dots, N$
8:	end for

4. SIMULATION AND RESULTS

In this section we will show results on small simulated datasets demonstrating that PLMF can correctly factor the data \mathbf{X}_0 into a mixing matrix \mathbf{M} , amplitudes \mathbf{A} , and phases \mathbf{z}_0 . Despite deriving PLMF for the noiseless case, we will also test its robustness to a small noisy perturbation. We will measure the quality of the result by directly comparing the estimated variables $\hat{\mathbf{M}}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{z}}_0$ with their true counterparts.

4.1 Data generation

We generate the data directly from the model. \mathbf{M} is taken as a random matrix with entries between 0 and 1. We then normalize it so that each row of \mathbf{M} has unit norm. Each row of \mathbf{A} (i.e. each source’s amplitude) is generated as a sum of 4 sinusoids, each with random frequencies between zero and $4/T$ and random initial phase. \mathbf{z}_0 is generated by uniformly spacing the N sources in the interval $[0, \frac{\pi}{2}]$. \mathbf{X}_0 is then generated according to the data model: $\mathbf{X}_0 = \mathbf{M}(\mathbf{A} \odot \mathbf{Z}_0)$ where \mathbf{Z}_0 is given by Eq. (4). Note that we generate \mathbf{X}_0 directly, so we skip the factorization of Φ entirely.

The initial values for the three 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 the same way as \mathbf{M}), while the elements of $\hat{\mathbf{z}}_0$ are of the form $e^{i\alpha}$ with α taken from the Uniform ($[0, \frac{\pi}{2}]$) distribution.

4.2 Quality measures

$\hat{\mathbf{M}}$ can be compared with \mathbf{M} through the gain matrix $\mathbf{G} \equiv \hat{\mathbf{M}}^+ \mathbf{M}$, where $\hat{\mathbf{M}}^+$ is the Moore-Penrose pseudo-inverse of $\hat{\mathbf{M}}$ [5]. 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. We use the Amari Performance Index (API) [4] to compute how close \mathbf{G} is to such a permutation. It is a non-negative number that is zero only if the gain matrix is exactly a permutation of the identity matrix. $\hat{\mathbf{A}}$ will be compared to \mathbf{A} through visual inspection. It is similar to compare \mathbf{Z}_0 or \mathbf{z}_0 to their estimated counterparts, because they contain exactly the same information. We chose the latter since it is much easier to represent. We compare them by computing for each source i the angular deviation from the true value: $\text{angle}(\mathbf{z}_{0,i}) - \text{angle}(\hat{\mathbf{z}}_{0,i})$.

We generate two datasets with the following features:

- Dataset 1: 5 sources, 10 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).

4.3 Results

The algorithm shown above is extremely fast – for example, for the two toy datasets above it takes less than a second to estimate all the three variables. Furthermore, we did not implement a convergence criterion and simply do 100 cycles of the optimization on $\hat{\mathbf{M}}$, $\hat{\mathbf{A}}$ and $\hat{\mathbf{z}}_0$. Implementation of a proper convergence criterion will likely make the algorithm significantly faster, as the first 20 to 30 iterations are usually enough to obtain convergence, as shown in Figure 1.

Figure 2 shows the results of the estimation of the source amplitudes, showing that $\hat{\mathbf{A}}$ is virtually equal to the real \mathbf{A}

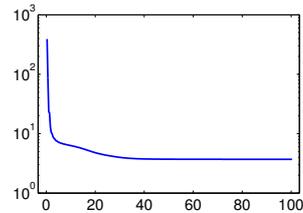


Figure 1: Evolution of the cost function as a function of the iteration for dataset 2.

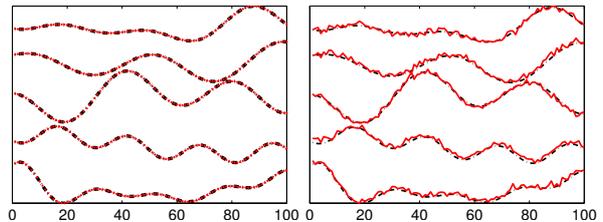


Figure 2: Visual comparison of the estimated amplitudes $\hat{\mathbf{A}}$ (red dots on the left side, red lines on the right side) with the true amplitudes \mathbf{A} (black lines). (Left) Results for dataset 1: the three estimated and true amplitudes coincide perfectly. (Right) Results for dataset 2: due to the presence of noise, it is impossible for the five estimated amplitudes to coincide perfectly with the true ones, but nevertheless the estimated amplitudes follow the real ones very well.

for dataset 1 and that it is approximately equal to \mathbf{A} for dataset 2. Note that if noise is present, it is impossible to recreate the original amplitudes as they are only present in the data corrupted by the noise. One can only estimate the corrupted amplitudes. If desired, a simple low-pass filtering procedure can closely recreate the original amplitudes.

Table 1 shows the API of the gain matrix \mathbf{G} and the worst-case angular distance in \mathbf{z}_0 . These values show that the result obtained on dataset 1 is near-perfect, and that it is still very good for dataset 2. An important remark is that our previous algorithm, Independent Phase Analysis [3], yields a performance for similar datasets that is worse, especially under noise. IPA also takes considerably longer to run.

5. DISCUSSION

The above results show that this approach has a high potential, although some limitations must be addressed to turn this

Data	API(\mathbf{G})	Angular distance (rad)
Dataset 1	0.0019	0.0020
Dataset 2	0.0520	0.0271

Table 1: Comparison of the estimated mixing matrix $\hat{\mathbf{M}}$ with the true mixing matrix \mathbf{M} through the Amari Performance Index of the gain matrix $\mathbf{G} \equiv \hat{\mathbf{M}}^+ \mathbf{M}$, and comparison of the estimated and true phases $\hat{\mathbf{z}}_0$ and \mathbf{z}_0 through the worst-case angular deviation. For zero noise (dataset 1), the estimation is near-perfect. The presence of noise understandably deteriorates the results, which are nevertheless still very good.

algorithm practical for real-world applications.

Throughout this paper we assumed that the matrix Φ , containing the oscillations common to all the sources, is known. In real applications this is often not the case. However, if the data is noiseless and well-sampled enough that the phase of two consecutive points never jumps by a value greater than π , it is a very simple procedure to obtain Φ from the data \mathbf{X} : since Φ is also a factor in \mathbf{X} , one can simply unwrap the phase [7] of one of the measured signals and use its phase time-series as the common oscillation.

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 \mathbf{z}_0 . We are investigating a way to estimate the “most common” phase oscillation Φ from the data \mathbf{X} , after which PLMF can be used to initialize a more general algorithm that estimates the full $\mathbf{Z} \equiv \Phi \odot \mathbf{Z}_0$. We are currently testing this more general algorithm, which optimizes \mathbf{Z} with a gradient descent algorithm. Yet, it is somewhat prone to local minima. A good initialization is likely to circumvent 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 [3]. In fact, using sources with $\Delta\phi_{ij} < \frac{\pi}{10}$ starts to deteriorate the results of PLMF, even with zero noise.

We did not tackle in this paper the case of sources organized in subspaces, i.e., where several clusters of sources are present in the data such that the intra-cluster phase synchrony is high but the inter-cluster phase-synchrony is low. We have previously shown that TDSEP can properly separate the subspaces from each other but fails to separate the sources within each subspace [1, 3, 2]. After running TDSEP, an algorithm such as PLMF can properly yield a full separation.

6. CONCLUSION

We presented 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 three variables: the mixing matrix, the source amplitudes, and their phases.

The results show that the proposed algorithm is fast, accurate, and can deal with low noise under the assumption that the sources are fully phase-locked. This approach opens a new research front in blind source-separation of phase-locked signals using concepts from matrix factorization.

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

- [1] M. Almeida, J. Bioucas-Dias, and R. Vigário. Independent phase analysis: Separating phase-locked subspaces. In *Proceedings of the Latent Variable Analysis Conference*, 2010.
- [2] M. Almeida, J. Bioucas-Dias, and R. Vigário. Detection and separation of phase-locked subspaces with phase noise. *Signal Processing*, (submitted), 2011.
- [3] M. Almeida, J.-H. Schleimer, J. Bioucas-Dias, and R. Vigário. Source separation and clustering of phase-locked subspaces. *IEEE Transactions on Neural Networks*, (accepted), 2011.
- [4] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems*, 8:757–763, 1996.
- [5] A. Ben-Israel and T. Greville. *Generalized inverses: theory and applications*. Springer-Verlag, 2003.
- [6] J. Bioucas-Dias and M. Figueiredo. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In *2nd workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2010.
- [7] J. Bioucas-Dias and G. Valadão. Phase unwrapping via graph cuts. *IEEE Transactions on Image Processing*, 16:698–709, 2007.
- [8] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.
- [9] E. Izhikevich. *Dynamic Systems in Neuroscience*. MIT Press, 2007.
- [10] Y. Kuramoto. *Chemical Oscillations, Waves and Turbulences*. Springer Berlin, 1984.
- [11] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13, pages 556–562, 2001.
- [12] P. L. Nunez, R. Srinivasan, A. F. Westdorp, R. S. Wijesinghe, D. M. Tucker, R. B. Silberstein, and P. J. Cadusch. EEG coherency I: statistics, reference electrode, volume conduction, laplacians, cortical imaging, and interpretation at multiple scales. *Electroencephalography and clinical Neurophysiology*, 103:499–515, 1997.
- [13] J. M. Palva, S. Palva, and K. Kaila. Phase synchrony among neuronal oscillations in the human cortex. *Journal of Neuroscience*, 25(15):3962–3972, April 2005.
- [14] A. Pikovsky, M. Rosenblum, and J. Kurths. *Synchronization: A universal concept in nonlinear sciences*. Cambridge Nonlinear Science Series. Cambridge University Press, 2001.
- [15] J.-M. Schoffelen, R. Oostenveld, and P. Fries. Imaging the human motor system’s beta-band synchronization during isometric contraction. *NeuroImage*, 41:437–447, 2008.
- [16] S. Strogatz. From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D*, 143:1–20, 2000.
- [17] S. Strogatz. *Nonlinear Dynamics and Chaos*. Westview Press, 2000.
- [18] P. J. Uhlhaas and W. Singer. Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology. *Neuron*, 52:155–168, Oct 2006.
- [19] R. Vigário, J. Särelä, V. Jousmäki, M. Hämäläinen, and E. Oja. Independent component approach to the analysis of EEG and MEG recordings. *IEEE Trans. On Biomedical Engineering*, 47(5):589–593, May 2000.
- [20] A. Ziehe and K.-R. Müller. TDSEP - an efficient algorithm for blind separation using time structure. In *International Conference on Artificial Neural Networks (1998)*, pages 675–680, 1998.