



**UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO**

AALTO UNIVERSITY

Separation of Synchronous Sources

Miguel Sousa Borges de Almeida

Supervisor: Prof. José Bioucas-Dias

Co-Supervisor: Prof. Erkki Oja

Co-Supervisor: Doc. Ricardo Vigário

Thesis specifically prepared to obtain the PhD Degree in Electrotechnical and Computer Engineering (at UL-IST) and Computer Science (at Aalto University)

Draft Version

August 2015

SEPARATION OF SYNCHRONOUS SOURCES

Dissertation to obtain the degrees:

Doctor of Science from Aalto University, Finland, and

Doctor of Electrical and Computer Engineering from Universidade Técnica de
Lisboa, Portugal

AUTHOR: Miguel Almeida

SUPERVISORS

Prof. Erkki Oja, Aalto University

Doc. Ricardo Vigário, Aalto University

Prof. José Bioucas-Dias, Universidade Técnica de Lisboa

ABSTRACT

This thesis studies the Separation of Synchronous Sources (SSS) problem, which deals with the separation of signals resulting from a linear mixing of sources whose phases are synchronous. While this study is made in a form independent of the application, a motivation from a neuroscience perspective is presented. Traditional methods for Blind Source Separation, such as Independent Component Analysis (ICA), cannot address this problem because synchronous sources are highly dependent. We provide sufficient conditions for SSS to be an identifiable problem, and quantify the effect of prewhitening on the difficulty of SSS. We also present two algorithms to solve SSS. Extensive studies on simulated data illustrate that these algorithms yield substantially better results when compared with ICA methods. We conclude that these algorithms can successfully perform SSS in varying configurations (number of sources, number of sensors, level of additive noise, phase lag between sources, among others). Theoretical properties of one of these algorithms are also presented. Future work is discussed extensively, showing that this area of study is far from resolved and still presents interesting challenges.

CONTENTS

1. <i>Introduction</i>	7
1.1 Summary	7
1.2 Contributions	10
1.2.1 Problem Formulation and Characterization	10
1.2.2 Separation Algorithms	11
1.3 Publications	12
1.4 Document Organization	16
2. <i>Blind Source Separation</i>	18
2.1 Inverse Problems	18
2.2 Linear and Instantaneous BSS	20
2.3 Independent Component Analysis	22
2.4 Independent Subspace Analysis	28
3. <i>Phase Synchrony</i>	35
3.1 From Real to Complex Signals: The Analytic Signal	35
3.2 Phase Synchrony	38
3.2.1 Neuroscience Motivation	39
3.2.2 Phase Locking Factor	42
4. <i>Separation of Synchronous Sources</i>	44
4.1 Problem Definition and Identifiability	46
4.2 Algorithm: Independent Phase Analysis	51

4.2.1	Regularization	52
4.2.2	Optimization Strategy	54
4.3	Algorithm: Phase Locked Matrix Factorization	55
4.3.1	General Approach	56
4.3.2	First subproblem	59
4.3.3	Second subproblem	61
4.3.4	Global identifiability	63
4.3.5	Optimization Strategy	65
4.4	The Effect of Whitening in SSS	66
5.	<i>Experimental Results</i>	71
5.1	IPA Results	71
5.2	PLMF Results	83
6.	<i>Future Work</i>	91
6.1	Tests on Real-Life Data	91
6.2	Improvements on PLMF	92
6.3	Generalizing PLMF for subspaces	93
6.4	Partial Synchrony	94
6.5	Partial ISA	94
7.	<i>Conclusions</i>	96

LIST OF PUBLICATIONS

- [P1] M. Almeida and R. Vigário. Source separation of phase-locked signals. In *Proc. of the Independent Component Analysis Conference*, pages 203–210, 2009.
- [P2] M. Almeida, J. Bioucas-Dias, and R. Vigário. Independent phase analysis: Separating phase-locked subspaces. In *Proceedings of the Latent Variable Analysis Conference*, pages 189–196, 2010.
- [P3] M. Almeida, J.-H. Schleimer, J. Bioucas-Dias, and R. Vigário. Source separation and clustering of phase-locked subspaces. *IEEE Transactions on Neural Networks*, 22(9):1419–1434, 2011.
- [P4] M. Almeida, J.-H. Schleimer, J. Bioucas-Dias, and R. Vigario. Source separation and clustering of phase-locked subspaces: Derivations and proofs. In arXiv.org. arxiv:1106.2474, 2011.
- [P5] M. Almeida, J. Bioucas-Dias, and R. Vigário. Separation of phase-locked sources in pseudo-real MEG data. *EURASIP Journal on Advances in Signal Processing*, 32, 2013.
- [P6] M. Almeida, R. Vigario, and J. Bioucas-Dias. Phase locked matrix factorization. In *Proc. of the EUSIPCO conference*, pages 1728–1732, 2011.
- [P7] M. Almeida, R. Vigário, and J. Bioucas-Dias. Estimation of the common oscillation for phase locked matrix factorization. In *Proceedings of the*

-
- International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pages 78–85, 2012.
- [P8] 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*, pages 51–66. Springer, 2013.
- [P9] M. Almeida, R. Vigário, and J. Bioucas-Dias. The role of whitening for separation of synchronous sources. In *Proceedings of the Latent Variable Analysis Conference*, pages 139–146, 2012.
- [P10] M. Almeida, J. Bioucas-Dias, R. Vigário, and E. Oja. A comparison of algorithms for separation of synchronous subspaces. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 60:455–460, 2012.
- [P11] M. Almeida, R. Vigário, and J. Bioucas-Dias. Separation of synchronous sources through phase locked matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*, (in press), 2014.

1. INTRODUCTION

1.1 Summary

In Blind Source Separation (BSS) problems, the goal is to estimate a set of signals, called *sources*, while having access only to combinations of those sources (the *mixtures*). The most widely studied class of BSS problems is Independent Component Analysis (ICA), which assumes that the sources are statistically independent.

Let there be N sources, which are stacked into a vector $\mathbf{s}(t)$, and P mixed signals, stacked into a vector $\mathbf{y}(t)$. Then, if the mixture process is linear, instantaneous, and noiseless, one can write $\mathbf{y}(t) = \mathbf{M}\mathbf{s}(t)$, where \mathbf{M} is called the *mixing matrix*.

This thesis deals with a specific instance of linear and instantaneous BSS called Separation of Synchronous Sources (SSS). Two complex signals are considered (fully) synchronous if the difference of their arguments, or phases, does not change with time.¹ While SSS itself is not specific to a particular domain, the motivation for this problem comes from neuroscience. Many studies in neuroscience have found that synchrony between brain regions is fundamental for normal information processing in the human brain, such as learning and memory. Furthermore, abnormal synchrony patterns have been associated to pathologies such as schizophrenia and Alzheimer's.

¹ This will be defined formally in section 3.2.2, along with definitions of partial synchrony and some mathematical properties of synchrony. We also discuss there how this can also be applied to real signals instead of complex ones.

Studies of synchrony in the neuroscience community have, in some cases, been invasive. Electrodes are placed, usually in mice, and the synchrony between brain regions is measured. Extra-cranial signals, such as the ones obtained through an electroencephalogram (EEG) or magnetoencephalogram (MEG), are an alternative and attractive way to study the synchrony of the brain. The non-invasive character of these signals make them suitable, for example, for diagnostics in human medicine. EEG and MEG signals are the result of a mixing process, and in this thesis we show that this mixing process can destroy synchrony information. Therefore, some researchers have employed BSS techniques, in particular ICA, to extract the original sources from these mixtures, and subsequently analyze the synchrony between the estimated sources. We argue that it is questionable to use ICA algorithms, which assume independent sources, to estimate synchronous sources, because they are highly dependent. Ideally, one would like to use algorithms which directly estimate synchronous sources.

In SSS, a different assumption is made: instead of assuming independence of the sources, as in ICA, it is assumed that the sources are phase-synchronous, which is a particular type of dependency. We shall show two theoretical properties which illustrate a significant parallelism between SSS and ICA:

1. In ICA, if the sources, denoted by the vector \mathbf{s} , are statistically independent, any linear combination $\hat{\mathbf{s}} = \mathbf{G}\mathbf{s}$, where \mathbf{G} denotes a square matrix, such that the components of $\hat{\mathbf{s}}$ are independent, must be such that $\hat{\mathbf{s}} = \mathbf{s}$, up to permutation, scaling and sign change, under some mild assumptions. We will show that a similar property exists for SSS, also under mild conditions: if the sources \mathbf{s} have perfect synchrony, any linear combination $\hat{\mathbf{s}} = \mathbf{G}\mathbf{s}$, with square \mathbf{G} , in which the synchrony between components of $\hat{\mathbf{s}}$ is perfect must be such that $\hat{\mathbf{s}} = \mathbf{s}$, again up to permutation, scaling and sign change, with the added requirement that \mathbf{G} is non-singular. While

this requirement exists in ICA as well, we will see that it introduces an additional difficulty in the design of SSS algorithms which is not present in most ICA algorithms.

2. Whitening has been proven to yield significant advantages as a pre-processing step for ICA; we will show that this is the case for SSS as well, albeit with smaller benefits. In ICA, in the absence of additive noise, whitening reduces the problem to a search for an orthogonal matrix. In SSS, even in the absence of noise, it makes the problem better conditioned, but the search space remains the full space of matrices of appropriate dimension. This is another property of SSS which makes it fundamentally harder than ICA.

In addition to proving these two properties and comparing them to their ICA counterparts, we will characterize and tackle SSS itself, namely in the following fronts:

- We show that sources which have perfect synchrony can be described in a particular mathematical form, and propose two algorithms to solve the SSS problem.
- We propose an algorithm called Independent Phase Analysis (IPA) which uses property 1 above and directly tries to maximize the synchrony of the estimated sources, $\hat{\mathbf{s}} = \mathbf{W}^T \mathbf{y}$, where \mathbf{y} are the mixtures, as a function of the *unmixing matrix* \mathbf{W} . To prevent \mathbf{W} from becoming singular, we use an appropriate term in the objective function that penalizes singular solutions.
- We also propose Phase Locked Matrix Factorization (PLMF), an algorithm which exploits the particular mathematical form that perfectly synchronous sources can be put in. It computes $\hat{\mathbf{M}}$ and $\hat{\mathbf{s}}$, the estimates of the mixing matrix and sources, respectively, which minimize the squared error

between the observed data \mathbf{y} and the product $\hat{\mathbf{M}}\hat{\mathbf{s}}$. Unlike IPA, PLMF has theoretical guarantees which prevent the occurrence of singular solutions. In fact, it will be shown that any global optimum of PLMF's cost function corresponds to correct estimations of \mathbf{M} and \mathbf{s} , up to permutation, scaling and sign change, under mild assumptions.

Experimental comparisons, using simulated data, show that PLMF generally obtains superior results when compared to IPA, and illustrate the limits where PLMF's performance degrades below a reasonable level, thus pointing directions for future improvements. These algorithms are also compared to ICA, showing that existing ICA algorithms fail to solve the SSS problem. Furthermore, an initial exploration towards real applications is performed with IPA, using pseudo-real MEG data, which are constructed from actual MEG data but in such a way that the true sources are known.

To this author's best knowledge, this thesis presents the first consolidated framework for blind source separation of synchronous signals, contributing with the problem formulation, theoretical properties of the problem, two algorithms for its solution and their theoretical properties, and experimental tests of those algorithms. The following chapters present a summary of this work, on which a total of nine papers were published in peer-reviewed journals and conferences.

1.2 Contributions

The contributions of this thesis can be roughly divided into two groups.

1.2.1 Problem Formulation and Characterization

This group of contributions concerns the SSS problem itself. The list of contributions is:

1. Characterizing the SSS problem's solutions. In particular, establishing that the usual BSS indeterminacies of permutation, scaling and sign change

are present, and that they are the only indeterminacies for non-singular solutions under mild conditions,²

2. Establishing that, unlike ICA, SSS can have singular solutions, which are undesirable.
3. Showing that solutions of this problem can be decomposed in a particular way using matrix factorization.
4. Showing that prewhitening results in a bound in the condition number of the equivalent mixing matrix. This can be interpreted as an upper bound on the “difficulty” of the SSS inverse problem.

1.2.2 Separation Algorithms

The second group of contributions regards the proposal of algorithms to solve the problem formulated in the previous subsection. The list of contributions is:

1. Proposing Independent Phase Analysis (IPA), a separation algorithm which directly exploits contribution 2 from the previous subsection to find a non-singular solution. This contribution is a significant extension of work that was initially done by Jan-Hendrik Schleimer.
2. Proposing Phase Locked Matrix Factorization (PLMF), another separation algorithm which exploits contribution 3 of the previous subsection to find a suitable matrix factorization of the data which yields the original sources.
3. Implementing both algorithms in MATLAB.
4. Showing, with simulated data, that both algorithms outperform source separation techniques not tailored for SSS, such as ICA methods.

² We call singular solutions those where the unmixing matrix \mathbf{W} is singular.

5. Creating pseudo-real MEG data and demonstrating the usefulness of IPA on it. To the author's best knowledge, real-world data where these algorithms could be tested can be collected with current technology, but is not publicly available and their acquisition is non-trivial.
6. Showing that, under certain conditions, all global minima of PLMF's cost function correspond to a solution which recovers the original sources.

1.3 Publications

The work in this thesis has resulted in the publication of nine peer-reviewed papers. An extended version of one of these papers was also published, as well as an arXiv supplementary material containing the proofs of some statements. Except where noted, the author of this thesis contributed in the following ways:

- **Algorithms and problem formulation:** In all papers, the design of the algorithms proposed in this thesis and the formulation of the SSS problem were done jointly with the author's supervisors, Ricardo Vigário and José Bioucas-Dias.
- **Theorems:** In all papers, choosing which results should be proven was done jointly with the supervisors. The proofs were all done by the present author, with feedback from his supervisors.
- **Experiments and code:** In almost all cases, implementation of the algorithms (in MATLAB) and performing the experiments leading to the results shown in the papers was done by the present author. An exception is a re-implementation of the code of PLMF, performed by one of the supervisors, which resulted in a significantly faster code.
- **Papers:** In all cases, the present author initially wrote all papers, and then incorporated suggestions from his supervisors at later stages.

These eleven publications are listed below, alongside a brief discription of each. Copies of these papers can be found at the end of the document, in the order listed below. Citations of these publications have the prefix “P”, as in [P1], to distinguish them from publications from other authors, which are not attached to this document, and which appear without that prefix, as in [1]. Note that, therefore, [P1] and [1] refer to different publications.

1. M. Almeida and R. Vigário. Source separation of phase-locked signals. In *Proc. of the Independent Component Analysis Conference*, pages 203-210, 2009 [P1]. This paper presented a novel cost function for an algorithm which would, in subsequent publications, become IPA. It yielded very significant improvements relative to work by Jan-Hendrik Schleimer [71], which can be considered the earliest version of IPA.
2. M. Almeida, J. Bioucas-Dias, and R. Vigário. Independent phase analysis: Separating phase-locked subspaces. In *Proceedings of the Latent Variable Analysis Conference*, pages 189-196, 2010 [P2].³ This paper presents further improvements on the IPA algorithm. The version of the algorithm presented in this paper is the one that is also presented in [P3].
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*, 22(9):1419-1434, 2011 [P3]. This is the main publication about IPA. It contains a description of the algorithm, as well as more extensive experimental results on simulated data. This paper also presented, for the first time, the theorem stating that all non-singular solutions yield the original sources.⁴ Some proofs were skipped due to lack of space; they are available in an arXiv paper.

³ The LVA conference is the same as the ICA one; it simply changed name.

⁴ This paper also presents two other algorithms, called Referenced Phase Analysis (RPA) and Phase Synchronization Cluster Analysis (pSCA), both originally proposed by Jan-Hendrik Schleimer [69, 70]. The author of this thesis reimplemented these two algorithms and per-

-
4. M. Almeida, J.-H. Schleimer, J. Bioucas-Dias, and R. Vigário. Source separation and clustering of phase-locked subspaces: derivations and proofs. *arXiv:1106.2474 [stat.ML]*, available at <http://arxiv.org/abs/1106.2474> [P4]. This arXiv paper contains the proofs from [P3] which were skipped due to lack of space.
 5. M. Almeida, J. Bioucas-Dias, and R. Vigário. Separation of phase-locked sources in pseudo-real MEG data. *EURASIP Journal on Advances in Signal Processing*, 32, 2013 [P5]. In this paper we tested IPA on pseudo-real data from real MEG data and concluded that IPA could separate synchronous sources in such data. In this paper we also presented an optimization strategy for IPA where the regularization to avoid singular solutions is progressively made weaker, so that in the limit one can avoid regularizing and thus be in the conditions of the theorem presented in [P3].
 6. M. Almeida, R. Vigário, and J. Bioucas-Dias. Phase locked matrix factorization. In *Proc. of the EUSIPCO conference*, pages 1728-1732, 2011 [P6]. This paper discussed the earliest form of the PLMF algorithm. It used an unrealistic assumption: it assumed that the phase of one of the sources was known.
 7. M. Almeida, R. Vigário, and J. Bioucas-Dias. Estimation of the common oscillation for phase locked matrix factorization. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, pages 78-85, 2012 [P7]. This paper removed that assumption and presented what ended up being called the “1-stage” PLMF, where all variables are estimated simultaneously. Out of roughly 150 papers presented
-
- formed all the experiments shown in [P3]. The present author also corrected minor errors in the expressions for the gradients in those two algorithms. However, the two algorithms were left mostly unchanged from their original forms, and they are not considered contributions.

at this conference, 12 were selected to have extended versions published in the Springer Proceedings in Mathematics & Statistics. This paper was among the 12 selected.

8. 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*, pages 51-66. Springer, 2013 [P8]. This is the extended version of the previous paper. It was published in the Springer Proceedings in Mathematics & Statistics. It presents significantly more thorough experimental results. The previous paper was peer-reviewed, but there was no further peer-review towards this extended version.
9. M. Almeida, R. Vigário, and J. Bioucas-Dias. The role of whitening for separation of synchronous sources. In *Proceedings of the Latent Variable Analysis Conference*, pages 139-146, 2012 [P9]. This paper presented the upper bound on the condition number of the equivalent mixing matrix if prewhitening is performed.
10. M. Almeida, J. Bioucas-Dias, R. Vigário, and E. Oja. A comparison of algorithms for separation of synchronous subspaces. *Bulleting of the Polish Academy of Sciences: Technical Sciences*, 60:455-460, 2012 [P10]. This paper compared the performance of multiple methods to separate subspaces of synchronous sources.
11. M. Almeida, R. Vigário, and J. Bioucas-Dias. Separation of synchronous sources through phase locked matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*, (in press), 2014 [P11]. This is the main publication about PLMF. It presents a novel form of PLMF with two subproblems, which we call the “2-stage” approach. In the first subproblem, one of the variables is estimated from a relaxed version of the

original problem. In the second subproblem, this variable is kept fixed and the remaining ones are estimated. This paper also presents theorems stating that all solutions of both subproblems are desirable ones. Finally, the algorithm is extensively studied on simulated data, and comparisons are made between ICA, IPA, and the 1-stage and 2-stage versions of PLMF, concluding that the latter is clearly superior in performance.

While all the above papers are pertinent for this thesis, the author considers the three journal papers [P3], [P5] and [P11] as the most important ones. The first and third ones contain the main theoretical results and both algorithms, while the second one presents early work towards application of these algorithms in real situations.

1.4 Document Organization

This dissertation is composed of an introductory part plus a list of publications at the end. In accordance with the rules of both Universities involved in this dissertation, the set of two parts needs to be self-sufficient. The introductory part, composed of chapters 1 to 7, contains most of the contributions: only the proofs of the theorems and some experimental results were omitted from the introduction, but they can be found in the publications in appendix.

This thesis is organized as follows. We begin with a brief introduction to Blind Source Separation in Chapter 2. Special focus is given to Independent Component Analysis, since it is the most widely used BSS problem, and because some SSS theoretical results have ICA counterparts.

We then formally introduce phase synchrony in Chapter 3. We show how the phase of a real signal can be computed through the construction of a complex signal. We present a brief motivation from a neuroscience perspective, and mathematically define synchrony to prepare its use in the algorithms that follow.

Chapter 4 contains this thesis' original contributions. First, the SSS problem

is formalized and discussed, without considering which algorithm will be used to solve it. Two main results are presented: a theorem stating that, like ICA, SSS is a well-posed problem; and another theorem quantifying the effect of whitening on the “difficulty” of SSS. This chapter also presents the two algorithms that are proposed to solve SSS: IPA and PLMF. PLMF has some interesting theoretical properties, which are also discussed in this chapter. Experimental tests with simulated data are presented for both algorithms; also, some results with pseudo-real MEG data, which are considerably more realistic than the simulated data, are shown for IPA.

Chapter 6 discusses future research directions in considerable detail. Conclusions are drawn in chapter 7. Finally, all publications related to this work are presented in Appendix.

2. BLIND SOURCE SEPARATION

2.1 *Inverse Problems*

Consider some physical phenomenon which is taking place, and some sensors that are placed to take measurements about the phenomenon. The *direct problem* is the one of computing what one would measure in the sensors, given the state of the experiment. The *inverse problem* is the problem of computing the state of the experiment given the measurements from the sensors. Blind Source Separation is an inverse problem, as we discuss below.

As an example, we briefly discuss the well-known *cocktail party problem*. In this conceptual problem, several people in a room are talking with one another, and some microphones are scattered throughout the room. The microphones capture sound coming from all the people that are talking, making it difficult to obtain the voice of one person directly from one of the microphones. In this situation, the direct problem would consist of computing the signals measured by each microphone, assuming that we know exactly the sound waves generated by each person, each person's location in the room, the room layout, and so on. While the computations for the direct problem may be non-trivial, conceptually it is a straightforward problem. The inverse problem in this situation involves finding the speech signals produced by each person using the signals measured by the microphones. Conceptually, inverse problems are much harder than their direct counterparts, often being ill-posed without further assumptions about the experimental setup.

Formally, the signals measured at the sensors are known as *mixtures*, *mixed*

signals, or *sensors*; we will use these terms interchangeably. The unknown signals which, when mixed, originate the measurements are usually called *sources*. We adopt this terminology here, and now proceed to define the notation used throughout this work.

In this work, we assume that all sources and sensors are one-dimensional discrete-time signals sampled at times $t = 1, 2, \dots, T$, that sources are numbered from 1 to N , and that sensors are numbered from 1 to P . Let $y_i(t)$ denote the discrete-time signal measured at sensor i , and $s_j(t)$ the discrete-time signal emitted by source j . A rather general BSS problem states that

$$y_i(t) = f(s_1(1), \dots, s_1(t), \dots, s_N(1), \dots, s_N(t)), \quad (2.1)$$

where $f(\cdot)$ is a function which depends on the experimental setup.¹ The measurement at sensor i and time t can depend on the signals of all the sources at all time instants up to the instant considered at the sensor.

Two assumptions can be made which tremendously simplify this problem. The first one is that the information travels instantaneously from the sources to the sensors, making this an *instantaneous* BSS problem. Such problems follow a model of the form

$$y_i(t) = f(s_1(t), \dots, s_N(t)), \quad (2.2)$$

i.e., the signal at sensor i and time t only depends on the signals of the sources at the same time instant.

The second assumption is linearity: if a problem is instantaneous and **linear**, its model is of the form

$$y_i(t) = \sum_{j=1}^N m_{ij} s_j(t), \quad (2.3)$$

where m_{ij} is a mixing coefficient which describes how the signal at sensor i depends on source j .

¹ This is not completely general. BSS problems can deal with signals which are more than one-dimensional, such as images. Also, the function f could depend explicitly on the time t , if the mixing process itself varies with time. We do not consider these two possibilities.

The contributions in this thesis assume that the model in (2.3) holds; this model is further explored in section 2.2. However, it is important to remark that relevant work has been done using other kinds of models. For example, *nonlinear ICA* has been used for image separation using instantaneous but non-linear models (the exact form is not that of equation (2.2), since the sources and sensors are 2D signals) [42, 4]. Furthermore, *convolutive* BSS problems are an important subclass, where the model is of the form

$$y_i(t) = \sum_{j \in \{1, \dots, N\}, \tau \in \{1, \dots, t\}} m_{ij\tau} s_j(t - \tau). \quad (2.4)$$

Convolutive BSS is linear but non-instantaneous, and has been used in applications such as the cocktail party problem and sound source localization, which in fact is similar to the cocktail-party problem, except that the goal is to locate the sound sources (which can be done from the coefficients $m_{ij\tau}$) and not to estimate the source signals $s_j(t)$. A good overview of convolutive BSS methods is available in [65].

2.2 Linear and Instantaneous BSS

Linear and instantaneous BSS is the simplest of all the models presented in the previous section, and is the model used throughout this work. Under this model, the signal measured at time t on sensor i , which we denote by $y_i(t)$, is given by equation (2.3). We have one equation of this form for every sensor i and every time instant t . We thus have $P \times T$ such equations.

It is common to combine these equations using matrix notation. Define $\mathbf{y}(t) \in \mathbb{R}^P$, $\mathbf{s}(t) \in \mathbb{R}^N$ and $\mathbf{M} \in \mathbb{R}^{P \times N}$ as follows:

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_P(t) \end{bmatrix}, \quad \mathbf{s}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{bmatrix} \quad \text{and} \quad \mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1N} \\ m_{21} & m_{22} & \dots & m_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ m_{P1} & m_{P2} & \dots & m_{PN} \end{bmatrix}. \quad (2.5)$$

Matrix \mathbf{M} is called the *mixing matrix*. The P equations of the form (2.3), corresponding to the same time instant t , can be compactly expressed as

$$\mathbf{y}(t) = \mathbf{M}\mathbf{s}(t), \quad (2.6)$$

and this can be done for each time instant t , thus there are T equations.

These T equations can be further compacted into a single equation. Define $\mathbf{Y} \in \mathbb{R}^{P \times T}$ and $\mathbf{S} \in \mathbb{R}^{N \times T}$ as follows:

$$\mathbf{Y} = \begin{bmatrix} y_1(1) & y_1(2) & \dots & y_1(T) \\ y_2(1) & y_2(2) & \dots & y_2(T) \\ \vdots & \vdots & \ddots & \vdots \\ y_P(1) & y_P(2) & \dots & y_P(T) \end{bmatrix} \quad \text{and} \quad \mathbf{S} = \begin{bmatrix} s_1(1) & s_1(2) & \dots & s_1(T) \\ s_2(1) & s_2(2) & \dots & s_2(T) \\ \vdots & \vdots & \ddots & \vdots \\ s_N(1) & s_N(2) & \dots & s_N(T) \end{bmatrix}. \quad (2.7)$$

We shall use a slight abuse of notation and sometimes call the matrix \mathbf{Y} the *sensor data*, or sometimes simply *sensors*. \mathbf{S} will be called *source data* or *sources*. The T equations of the form (2.6) can be compacted into a single equation:

$$\mathbf{Y} = \mathbf{M}\mathbf{S}. \quad (2.8)$$

Usually, the objective of BSS is to find the sources \mathbf{S} , using only the data from the sensors \mathbf{Y} , although in some cases the goal might be to find the mixing matrix \mathbf{M} . In either case, BSS is an ill-posed problem. Equation (2.8) makes the reasons for this clear: in general, there is an infinite number of pairs (\mathbf{M}, \mathbf{S}) which, when multiplied, yield the observed data \mathbf{Y} . It is, therefore, necessary to make some assumptions on the sources \mathbf{S} , on the mixing matrix \mathbf{M} , or on both, to make the problem well-posed. Different BSS problem make different assumptions:

- By far, the most well-known BSS problem is Independent Component Analysis (ICA). Its fundamental assumption is that, at each time instant t , the value of each source $s_j(t)$ is a realization of a random variable S_j , and that the random variables S_1, S_2, \dots, S_N are statistically independent.

ICA will be briefly discussed in Section 2.3; further analysis is deferred to that section.

- A generalization of ICA is Independent Subspace Analysis (ISA). Its fundamental assumption is that there are several sets of sources, which are usually called *subspaces*. Sources in the same subspace can be mutually dependent, but the set of sources in a subspace is independent from the set of all other sources. While ICA can be considered a mature field, ISA is currently being actively researched. It will be briefly discussed in Section 2.4.
- Non-negative Matrix Factorization (NMF) can also be viewed as an instance of BSS, although the literature does not always cast it as such. Its fundamental assumption is that the entries of both the mixing matrix \mathbf{M} and the source data \mathbf{S} are non-negative. Despite being now around ten years old [36, 52], NMF is a very active area of research, with applications in, *e.g.*, acoustic signal processing [44, 45] and hyperspectral unmixing [53]. However, this topic is not central to the work presented in this thesis, and it is not discussed further.
- The topic of this thesis, Separation of Synchronous Sources (SSS), is also an instance of BSS. The fundamental assumption is that the sources have perfect phase synchrony with one another. SSS will be the subject of Section 4.

One can draw several parallelisms between SSS and the well-known case of ICA. For this reason, we now provide a brief overview of ICA.

2.3 Independent Component Analysis

The term “Blind Source Separation” began to be used in the early 1990s (see, *e.g.*, [41]), while the term “Independent Component Analysis” became

widespread a few years later [17]. However, ICA began to take form in the early 1980s, in France, with works by Hérault, Jutten, and Ans [32, 34, 7, 33]. In [43] (an excellent historical overview, including comments from many pioneer researchers in the field), Jutten places the birth of BSS in 1982. According to [38], despite earlier works presenting solutions to the problem which would become known as ICA (such as [33]), a major turning point in the history of ICA was the publication, in 1995, of an approach based on the infomax principle [10, 9], which drew wider attention to the field. ICA can now be considered a mature field of research, with the ICA conference, created in 1989 and occurring every 18 months until 2009, and then the LVA/ICA conference² from 2010 onwards, gathering around 150 researchers from the field.

ICA has seen wide application, even in situations where the independence assumption is not satisfied. Examples of applications are the removal of artifacts from neurophysiological signals [54, 87] and modeling the receptive fields of neurons of the primary visual cortex [11], among many others. Good overviews of ICA include [38, 16, 18].

As was said above, ICA, in its typical form, is a linear and instantaneous BSS problem which assumes that $s_j(t)$, and consequently $y_i(t)$, are realizations of random variables. Specifically, the sources $s_j(t)$ are assumed to be i.i.d. realizations of random variables S_j , with S_1, S_2, \dots, S_N being statistically independent. It turns out that this independence assumption is enough to make the problem “sufficiently well-posed”, in a sense that will be rigorously defined below.

For simplicity, throughout this section the number of sources is assumed to be equal to the number of sensors, *i.e.*, $P = N$, and the mixing matrix \mathbf{M} (which is square for $P = N$) is assumed to be invertible. Most results in this section can be generalized to the case where one has more sensors than sources ($P > N$), which is called the *overdetermined* case. The case where the number of sensors

² LVA stands for the more general designation “Latent Variable Analysis”.

is smaller than the number of sources ($P < N$), which is called *underdetermined*, is considerably harder to tackle.

One of the most important aspects of ICA is that it is still technically ill-posed, in the sense that there are still infinite solutions to equation (2.8), even if the independence assumption is verified. However, the following theorem precisely characterizes the ill-posedness of ICA [17]:

Theorem 2.3.1. *Let $\hat{\mathbf{s}} \equiv \mathbf{G}\mathbf{s}$ be a set of signals resulting from a linear combination of sources \mathbf{s} , with square \mathbf{G} , and let these sources be statistically independent as per the ICA model. Furthermore, assume that at most one of the components of \mathbf{s} is Gaussian, and that none of them has a degenerate point-like distribution. The components of $\hat{\mathbf{s}}$ are statistically independent if and only if $\hat{\mathbf{s}} = \mathbf{D}\mathbf{P}\mathbf{s}$ for some diagonal matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ with nonzero entries in its diagonal, and some permutation matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$.*

This theorem formalizes the previously mentioned statement that ICA is “sufficiently well-posed”. While the problem is ill-posed, in the sense that finding independent estimated sources $\hat{\mathbf{s}}$ does not imply that $\hat{\mathbf{s}} = \mathbf{s}$, all those solutions correspond to situations where each component of $\hat{\mathbf{s}}$ depends only on a single source. Specifically, the following indeterminacies exist:

- The *order* of the sources cannot be determined. This happens because the order of the terms in the sum of Equation (2.3) can be changed without affecting the value of the sum. Equivalently, one can permute the rows of matrix \mathbf{S} and apply the same permutation to the columns of \mathbf{M} without affecting the product \mathbf{MS} . In the matricial notation of Equation (2.8), it is equivalent to considering a new mixing matrix $\tilde{\mathbf{M}} \equiv \mathbf{M}\mathbf{P}^{-1}$ and a new source matrix $\tilde{\mathbf{S}} \equiv \mathbf{P}\mathbf{S}$, where $\mathbf{P} \in \mathbb{R}^{N \times N}$ is some permutation matrix. This is called the *permutation indeterminacy*.
- The *scale* of the sources cannot be determined. This happens because one can, in Equation (2.3), apply scaling factors $\alpha_j \neq 0$ to each source s_j

and apply the inverse scaling $\frac{1}{\alpha_j}$ to all mixing coefficients involving that source, m_{1j}, \dots, m_{Pj} , without affecting the resulting mixture signals. In the matricial notation of Equation (2.8), it is equivalent to considering a new mixing matrix $\tilde{\mathbf{M}} \equiv \mathbf{M}\mathbf{D}^{-1}$ and a new source matrix $\tilde{\mathbf{S}} \equiv \mathbf{D}\mathbf{S}$, where $\mathbf{D} \in \mathbb{R}^{N \times N}$ is some diagonal matrix with non-zero entries in its diagonal. This is called the *scaling indeterminacy*.

- The previous indeterminacy can involve negative scaling factors, which result in changes of sign of the estimated sources. While this is already included in the previous case, this is sometimes referred separately as the *sign indeterminacy*.

These indeterminacies are very common in linear and instantaneous BSS. However, note that they may depend on the specific problem. For example, in Non-Negative Matrix Factorization, where the matrices \mathbf{M} and \mathbf{S} are assumed to have non-negative entries, the sign indeterminacy does not exist. Usually, the way to deal with these indeterminacies is that one aims at finding an *unmixing matrix* \mathbf{W} such that the estimated sources, given by $\hat{\mathbf{s}} \equiv \mathbf{W}^T \mathbf{y} = \mathbf{W}^T \mathbf{M} \mathbf{s}$, are a permutation and scaling of the original sources, since the order and scale are impossible to determine. Equivalently, the *gain matrix* $\mathbf{W}^T \mathbf{M}$, should be a permutation of a diagonal matrix with nonzero elements in the diagonal.

If one could simply “maximize the independence” of the estimated sources $\hat{\mathbf{s}}$ as a function of \mathbf{W} , then the previous theorem would ensure that all global optima, where the estimated sources are independent, would be good enough as long as the order and scale of the sources were not important. It is important to remark, however, that the notion of “maximizing the independence” of the estimated sources glosses over a lot of the research put into ICA. In fact, it is not easy to measure the independence of a set of random variables when one only has access to a set of realizations. ICA algorithms replace independence with other criteria which approximate independence, such as kurtosis, negentropy,

and time-lagged correlations, among others. The exploration of these flavors of ICA is outside the scope of this work; good overviews can be found in [38, 16, 19].

We now describe two very common and very useful preprocessing steps for ICA. Let $\mathbb{E}[\mathbf{y}]$ denote the expected value of the random vector \mathbf{y} , and $\mathbb{E}[\mathbf{y}\mathbf{y}^T]$ denote its covariance. In practice, the mean $\mathbb{E}[\mathbf{y}]$ and covariance $\mathbb{E}[\mathbf{y}\mathbf{y}^T]$ are unknown. If a sufficient number of time samples is available, these can be well approximated by their estimators, $\langle \mathbf{y} \rangle$ and $\frac{T}{T-1} \langle \mathbf{y}\mathbf{y}^T \rangle$, where $\langle . \rangle$ denotes the time averaging operator. ICA is usually preceded by **centering** the data, *i.e.*, removing its mean. This step returns for each time t a new vector given by $\tilde{\mathbf{y}}(t) \equiv \mathbf{y}(t) - \mathbb{E}[\mathbf{y}]$. Equivalently, one can subtract $\mathbb{E}[\mathbf{y}]$ from every column of matrix \mathbf{Y} . For simplicity, we shall assume that centering has been applied to the data.

Also, it is normally useful to perform **prewhitening** of the data: this step applies a linear transformation to the data such that their covariance matrix, after the transformation, is the identity matrix. Let

$$\mathbf{C}_{\mathbf{Y}} \equiv \mathbb{E}[\mathbf{y}\mathbf{y}^T] \quad (2.9)$$

denote the covariance matrix of the data.³ Consider the eigendecomposition of $\mathbf{C}_{\mathbf{Y}}$,

$$\mathbf{C}_{\mathbf{Y}} = \mathbf{V}\mathbf{D}\mathbf{V}^T, \quad (2.10)$$

where \mathbf{D} is a diagonal matrix containing the eigenvalues of $\mathbf{C}_{\mathbf{Y}}$ in some order, and \mathbf{V} is an orthogonal matrix ($\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I}$) containing the eigenvectors of $\mathbf{C}_{\mathbf{Y}}$ in the corresponding order.

Then, prewhitening can be performed by multiplying the data \mathbf{Y} on the left by the matrix

$$\mathbf{B} \equiv \mathbf{D}^{-\frac{1}{2}}\mathbf{V}^T. \quad (2.11)$$

³ Recall that the data have been centered, and therefore their mean is zero. Therefore, we can use the terms “correlation” and “covariance” interchangeably.

It is easy to see that the covariance of the whitened data \mathbf{z} , given by $\mathbf{z} \equiv \mathbf{B}\mathbf{y}$, is the identity matrix:

$$\begin{aligned}
 \mathbb{E}[\mathbf{z}\mathbf{z}^T] &= \mathbb{E}[\mathbf{B}\mathbf{y}\mathbf{y}^T\mathbf{B}^T] \\
 &= \mathbf{B}\mathbb{E}[\mathbf{y}\mathbf{y}^T]\mathbf{B}^T \\
 &= \mathbf{D}^{-\frac{1}{2}} \underbrace{\mathbf{V}^T\mathbf{V}}_{\equiv \mathbf{I}} \mathbf{D} \underbrace{\mathbf{V}^T\mathbf{V}}_{\equiv \mathbf{I}} \mathbf{D}^{-\frac{1}{2}} \\
 &= \mathbf{D}^{-\frac{1}{2}} \mathbf{D} \mathbf{D}^{-\frac{1}{2}} \\
 &= \mathbf{I}.
 \end{aligned} \tag{2.12}$$

An important consequence of prewhitening is the following result [38].

Theorem 2.3.2. *Let $\mathbf{y} \equiv \mathbf{M}\mathbf{s}$ be a set of measurements resulting from a linear combination of sources \mathbf{s} , with invertible \mathbf{M} , and let these sources be statistically independent as per the ICA model. Furthermore, assume that at most one of the components of \mathbf{s} is Gaussian. Let $\mathbf{z} \equiv \mathbf{B}\mathbf{y} = \mathbf{B}\mathbf{M}\mathbf{s}$ denote the result of prewhitening the sensor data. Then, there exists an orthogonal matrix $\mathbf{W} \in \mathbb{O}^{N \times N}$ such that the components of $\hat{\mathbf{s}} \equiv \mathbf{W}^T \mathbf{z}$ are statistically independent.*

Theorem 2.3.2 states that, if prewhitening is performed, we can find a set of independent sources $\hat{\mathbf{s}}$ by searching for an orthogonal matrix. Note that $\hat{\mathbf{s}}$ is a linear combination of the original sources \mathbf{s} , because $\hat{\mathbf{s}} \equiv \mathbf{W}^T \mathbf{z} = \mathbf{W}^T \mathbf{B}\mathbf{y} = \mathbf{W}^T \mathbf{B}\mathbf{M}\mathbf{s}$. Therefore, Theorem 2.3.1 ensures that after prewhitening one can solve the ICA problem (in the absence of noise) by searching for an orthogonal matrix. While the literature on ICA usually discusses prewhitening as well [38, 16, 18], it should be emphasized that while prewhitening is a useful preprocessing step for ICA, it is not part of it, nor is it indispensable.

Let $\sigma_{\max}(\mathbf{M})$ and $\sigma_{\min}(\mathbf{M})$ denote the largest and smallest singular values

of matrix \mathbf{M} .⁴ The **condition number** of \mathbf{M} is defined as

$$\rho(\mathbf{M}) = \frac{\sigma_{max}(\mathbf{M})}{\sigma_{min}(\mathbf{M})}. \quad (2.13)$$

The condition number of a matrix with at least one nonzero element always belongs to the set $[1, +\infty) \cup \{+\infty\}$. In particular, orthogonal matrices have $\rho = 1$, and singular matrices have $\rho = +\infty$.

The condition number of a mixing matrix can be considered an indicator of the difficulty of the corresponding inverse problem.⁵ Problems where the mixing matrix has a high condition number are usually harder to solve than problems with a smaller condition number. Theorem 2.3.2 ensures that, if prewhitening is performed, ICA is reduced to a search for an orthogonal matrix, which has $\rho = 1$. This will not be the case for SSS. Thus, in a certain sense, SSS can be considered a “harder” inverse problem when compared to ICA. We shall return to this interpretation later, when we discuss the effect of whitening on SSS.

2.4 Independent Subspace Analysis

The goal of this thesis is to study the problem of separating sources with a particular kind of dependency: they are assumed to be perfectly synchronous. In reality, however, this assumption limits the applicability of the theory and of the methods developed here. In fact, in real-world situations it is unlikely that one only has sources which are perfectly synchronous with one another. It is more realistic to consider situations where (approximately) synchronous sources of interest may be considered independent of all other sources present

⁴ For square Hermitian matrices, $\sigma_{max}(\mathbf{M})$ and $\sigma_{min}(\mathbf{M})$ are equal to $|\lambda_{max}(\mathbf{M})|$ and $|\lambda_{min}(\mathbf{M})|$, respectively, where $\lambda_{max}(\mathbf{M})$ and $\lambda_{min}(\mathbf{M})$ are the eigenvalues of \mathbf{M} with the largest and smallest absolute values, respectively.

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.

in the mixing process. A brief motivation from a neuroscience perspective will be presented in section 3.2.1.

In this section we explore a generalization of ICA where, instead of assuming independence between individual sources, one assumes independence between groups of sources, and sources within each group may be dependent. This generalization is called *Independent Subspace Analysis* (ISA). If some of these groups' dependency is strong phase synchrony, as may be the case in the human brain, a two-step procedure where ISA is employed to separate the groups, and SSS is employed to extract synchronous sources from their respective groups, may be adequate.

In ISA one assumes that there are sets of sources, called *subspaces*⁶, where each set is independent from the set of sources not belonging to the subspace. Sources within each such set are not necessarily independent. In mathematical notation, the vector of sources, \mathbf{s} , is assumed to have the form

$$\mathbf{s} \equiv \begin{bmatrix} \mathbf{s}^1 \\ \mathbf{s}^2 \\ \vdots \\ \mathbf{s}^K \end{bmatrix}, \text{ with } \mathbf{s}^k \equiv \begin{bmatrix} s_1^k \\ \vdots \\ s_{N_k}^k \end{bmatrix}, \quad (2.14)$$

where $\mathbf{s}^1, \mathbf{s}^2, \dots, \mathbf{s}^K$ are called the K *subspaces* of \mathbf{s} . We use N_k to denote the dimension of each subspace; note that they must obey $N_1 + \dots + N_K = N$, where N is the number of sources, *i.e.*, the dimension of \mathbf{s} . The critical assumption in ISA is that the vectors $\mathbf{s}^1, \mathbf{s}^2, \dots, \mathbf{s}^K$ are statistically independent. As mentioned above, the components of each subspace vector \mathbf{s}^k need not be independent.

If $N_1 = N_2 = \dots = N_K = 1$ and $K = N$, all subspaces have dimension 1 and we recover the ICA case. In the other extreme, where $K = 1$, only one subspace is present, and no independence assumption exists. Source separation,

⁶ Technically, “subspaces” is the term for all possible linear combinations of sources in one of these sets. In the literature it is common to also call a “subspaces” to each set of sources. We will employ this slightly abusive terminology.

in this case, is sometimes called Dependent Component Analysis (DCA), a field with its own active research community (see, *e.g.*, [47] and references therein). SSS is a form of DCA.

Assume that it is known that a certain set of observations $\mathbf{y}(t)$ is the result of a mixture of sources $\mathbf{s}(t)$, of dimension N , using a mixing matrix \mathbf{M} : $\mathbf{y}(t) = \mathbf{M}\mathbf{s}(t)$. Assume also that \mathbf{s} follows the ISA assumption: the N sources can be partitioned into a number of sets $K \in \{1, \dots, N\}$ and the different sets are independent. ICA can be motivated as the minimization of the *mutual information* between the scalar sources [38, chapter 10]:

$$\min MI(\hat{s}_1, \dots, \hat{s}_N), \quad (2.15)$$

where \hat{s}_i is the estimate of the i -th source. In the presence of subspaces, one natural generalization would be to minimize the mutual information between the various subspaces:

$$\min MI(\hat{\mathbf{s}}^1, \dots, \hat{\mathbf{s}}^K)$$

This approach has seen some use. However, it presents two problems:

- In general, this approach is a combinatorial optimization problem [13], since one does not know which of the estimated sources should be grouped together [3, 64] when defining the subspaces. One can then test all possible groupings, but they grow very quickly with N : the problem rapidly becomes intractable.⁷ Alternatively, one could solve a discrete optimization problem by following, *e.g.*, a greedy approach to cluster the estimated sources, an approach that is not guaranteed to yield the optimal solution.
- This approach involves the computation of the entropy of random vectors of dimension N_k . Such computation is non-trivial for $N_k \geq 2$ [8], further

⁷ The number of partitions of a set with n elements is called the n -th *Bell number*, B_n . One has $B_0 = B_1 = 1$ and $B_{n+1} = \sum_{i=0}^n \frac{n!}{i!(n-i)!} B_i$. The first few elements are 1, 1, 2, 5, 15, 52, 203, 877 [2].

increasing the complexity of this approach. Nevertheless, this approach has been tackled, *e.g.*, by estimating the entropy of multi-dimensional components using minimum spanning trees [64], or using variational Bayes approaches [3].

We now divide the general ISA problem, of recovering the original sources when subspaces are present, into three successive parts. The first part is called *inter-subspace separation*. The goal of this first part is to obtain a demixing matrix \mathbf{W}_{inter} , such that the *gain matrix*, $\mathbf{G} = \mathbf{W}_{inter}^T \mathbf{M}$, is a permutation of a block diagonal matrix with blocks corresponding to the subspaces. For example, suppose that there are three subspaces ($K = 3$), the first of which has three components ($N_1 = 3$), while the second and third subspaces have two components each ($N_2 = N_3 = 2$). In this case, the goal is to find a matrix \mathbf{W}_{inter} of the form $\mathbf{W}_{inter}^T \equiv \mathbf{P} \mathbf{B}_{inter}$, where \mathbf{P} is a permutation matrix and \mathbf{B}_{inter} is such that

$$\mathbf{B}_{inter} \mathbf{M} = \begin{bmatrix} \mathbf{U}^1 & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 2} \\ \mathbf{0}_{2 \times 3} & \mathbf{U}^2 & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 2} & \mathbf{U}^3 \end{bmatrix}.$$

Here, $\mathbf{0}_{m \times n}$ is the m -by- n zero matrix, \mathbf{U}^1 is a 3-by-3 invertible matrix, and \mathbf{U}^2 and \mathbf{U}^3 are 2-by-2 invertible matrices. After this step, each entry of the random vector $\mathbf{x}_{inter} \equiv \mathbf{W}_{inter}^T \mathbf{y}$ is a linear combination of sources from one subspace only.

The second step is called *subspace detection*. The goal is to permute the entries of the random vector \mathbf{x}_{inter} so that the first N_1 entries of \mathbf{x}_{inter} are linear combinations of sources from the first subspace, the next N_2 entries are linear combinations of sources from the second subspace, and so on. Formally, we multiply \mathbf{x}_{inter} by a suitable permutation matrix, \mathbf{Q} . Finding \mathbf{Q} is, in general, a combinatorial problem. In the case of SSS, we used a simple heuristic [P3] to perform this step, with reasonable results.

After the subspace detection is completed, one can define

$$\mathbf{y}^1 \equiv \mathbf{Q}_{(1:N_1, :)} \mathbf{x}_{inter} \quad (2.16)$$

$$\mathbf{y}^2 \equiv \mathbf{Q}_{(N_1+1:N_1+N_2, :)} \mathbf{x}_{inter} \quad (2.17)$$

$$\vdots \quad (2.18)$$

$$\mathbf{y}^K \equiv \mathbf{Q}_{(N-N_K+1:N, :)} \mathbf{x}_{inter}, \quad (2.19)$$

where $\mathbf{Q}_{(a:b, :)}$ is a matrix composed of the rows a to b of matrix \mathbf{Q} . Thus, \mathbf{y}^k is a N_k -dimensional vector containing linear combinations of the sources of the k -th subspace.

The third and last step is called *intra-subspace separation*. It involves finding square matrices $\mathbf{W}^k \in \mathbb{R}^{N_k \times N_k}$ such that $\mathbf{s}^k = (\mathbf{W}^k)^T \mathbf{y}^k$, up to permutation, scale and sign change. There are K such matrices to be found, and each can be estimated separately once inter-subspace separation and subspace detection have been performed. This step requires some knowledge about the sources under study or about their interdependency within the subspace. If we know that the interdependency is strong synchrony, SSS algorithms could be employed to perform this step.

Sadly, there is no consensus in the available literature on what the term “Independent Subspace Analysis” means. Some authors (*cf.*, [62, 78]) define ISA as the task of performing all three steps, while others (*cf.*, [39, 73]) define the same term as solving only the first step or the first two steps. To prevent confusion, we shall define *Full ISA* as the task of performing all three steps, and *Partial ISA* as the task of performing only the first step.

Both full and partial ISA have seen increasing interest from the scientific community in recent years. While these problems are usually called “Independent Subspace Analysis” [29, 39, 64, 83], other names have been used, such as “Subspace Independent Component Analysis” [73], “Independent Vector Analysis” [1], and “Multidimensional Independent Component Analysis” [15], among

others. ISA was first proposed for fetal electrocardiogram extraction [21]; another important early work is [15]. It has also been applied to capturing inter-subject variability in functional magnetic resonance imaging (fMRI) [1], natural image analysis [37] and analysis of cosmic microwave background radiation [50], among other fields.

Relevant theoretical results have been published about this topic, such as sufficient conditions on the distribution of the sources for full ISA to be achievable through maximization of kurtosis [78] or through minimization of mutual information [62]. A general discussion of contrast functions can be found in [63]. Under the (quite restrictive) conditions stated in these works, then, simple ICA algorithms which maximize kurtosis (such as some variants of FastICA) or minimize mutual information (such as Infomax) can be safely used to perform the first and third steps of full ISA, even though the assumption of independence of the sources is violated. In other words, one can recover the original sources by applying methods which do not consider subspaces at all; one can, if desired, group the recovered sources into subspaces *a posteriori*.

Dedicated algorithms for partial ISA have also been proposed; see, *e.g.*, [39] and [73]. Techniques for subspace detection have also been recently presented [29]. We performed a comparison of ICA and ISA algorithms for partial ISA where the interdependency in each subspace is perfect synchrony [P10]. The best-performing algorithm was FastICA, which outperformed two other ICA algorithms as well as three ISA ones.

Some researchers have focused on specific types of sources. For example, in [51], second-order statistics are used to perform ISA, and a model is derived for multidimensional Gaussian sources.

The conjecture that full ISA can be solved by using simple ICA and afterwards grouping the sources into subspaces is called *ISA separation principle* [79]. This conjecture has been proven for certain source types (see [79] for an overview), and recent works such as [49] suggest that it may be true for a broad

class of sources.

3. PHASE SYNCHRONY

Unlike ICA, in which we assume that the sources are statistically independent, in Separation of Synchronous Sources (SSS) the underlying assumption is that the sources have a very particular kind of dependence: they are perfectly synchronous. In this section, we provide a brief overview of the concepts of phase and phase synchrony. We begin by discussing how to obtain the phase of a real signal, and then provide precise definitions of synchrony which will be used in the following chapters.

3.1 *From Real to Complex Signals: The Analytic Signal*

In many real-world applications, including the analysis of EEG and MEG signals from the brain, the measurements are real-valued. However, all the methods proposed in this thesis will require us to compute the phase of the signals presented as input. In this section we discuss an important question: how can we define the phase of a real signal $s(t)$?

Typically, this step is performed by obtaining a complex signal $x(t)$ from the given real signal, and defining the phase of the real signal $s(t)$ as the argument, or angle, of the complex signal $x(t)$. This reduces the question of the previous paragraph to a new question: how should we define this complex signal?

Let the original signal be decomposed as $s(t) = a(t) \cos \varphi(t)$, where $a(t)$ is the *amplitude* of the signal and $\varphi(t)$ is its *phase*. Our goal is to know under which conditions we can create a complex signal $x(t)$ equal to $x(t) = a(t)e^{i\varphi(t)}$; we will say that this is a “meaningful” complex signal, and that its phase is

“meaningful” as well. Furthermore, the function mapping $s(t)$ to $x(t)$ should be a linear function, such that a linear combination of real signals yields the same linear combination of the corresponding complex signals.

Let $S(\omega)$ denote the Fourier transform (FT) of $s(t)$, given by

$$S(\omega) \equiv \int_{-\infty}^{+\infty} s(t)e^{-i\omega t} dt, \text{ for } \omega \in \mathcal{R}, \quad (3.1)$$

and let $X(\omega)$ be the FT of $x(t)$. Note that $s(t) = \frac{1}{2}a(t)(e^{i\varphi(t)} + e^{-i\varphi(t)}) = \frac{1}{2}(x(t) + x^*(t))$, where $*$ denotes complex conjugation. Using basic FT properties [61, section 4.6], the FTs of $s(t)$ and $x(t)$ are related through $S(\omega) = \frac{1}{2}(X(\omega) + X^*(-\omega))$.

Suppose now that the support of $X(\omega)$ is contained in \mathbb{R}^+ . In that case, the support of $X^*(-\omega)$ is contained in \mathbb{R}^- , and $X(\omega)$ can be obtained from $S(\omega)$ through

$$X(\omega) = 2U(\omega)S(\omega), \quad (3.2)$$

where $U(\omega)$ is the *step* (or *Heaviside*) function:

$$U(\omega) = \begin{cases} 1 & \text{if } \omega \geq 0 \\ 0 & \text{if } \omega < 0. \end{cases}$$

Therefore, we can construct $X(\omega)$ by computing the FT of $s(t)$ and multiplying it by the step function $U(\omega)$. To obtain $x(t)$, which is called the *analytic signal*, we merely need to compute the inverse Fourier transform (IFT) of $X(\omega)$, given by

$$x(t) \equiv \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega)e^{i\omega t} d\omega, \text{ for } t \in \mathcal{R}. \quad (3.3)$$

An important question remains to be discussed: for which real signals do the corresponding analytic signals $x(t)$ equal $x(t) = a(t)e^{i\varphi(t)}$? This question is fundamental for this thesis, since apart from the few real-world cases that are intrinsically well represented by complex-valued signals, those will be the signals that our methods can be applied to.

The crucial step in obtaining the analytic signal is the assumption that the support of $X(\omega)$ only contains positive frequencies. For example, consider a signal defined as

$$s(t) = a(t) \cos(\omega_0 t) \quad (3.4)$$

where $a(t)$ is some non-negative signal, and ω_0 is a positive real number. By the shift property of the FT, the FT of $s(t)$ is given by

$$S(\omega) = \frac{A(\omega - \omega_0) + A(\omega + \omega_0)}{2}, \quad (3.5)$$

where $A(\omega)$ is the FT of $a(t)$.

Suppose that the following condition holds:

$$A(\omega) = 0 \text{ for } |\omega| \geq \omega_0. \quad (3.6)$$

In this case, $A(\omega + \omega_0) = 0$ for $\omega > 0$, and the FT of the analytic signal of $s(t)$ (equation (3.2)) yields

$$X(\omega) = \begin{cases} 0 & \text{for } \omega < 0 \\ A(\omega - \omega_0) & \text{for } \omega > 0, \end{cases}$$

and the corresponding analytic signal is

$$x(t) = a(t)e^{i\omega_0 t}.$$

The phase of $x(t)$ is its argument, and since $a(t)$ is real and non-negative, its argument is simply equal to $\omega_0 t$, the argument of the cosine function in the definition of $s(t)$ (equation (3.4)). Therefore, in this case the analytic signal preserves the phase of the real signal in a meaningful way.

In the general case, where $\varphi(t)$ is not constant, the signals for which the analytic signal contains a meaningful phase are those for which the support of $X(\omega)$ is contained in \mathbb{R}^+ . Note that the analytic signal is given by the product of the amplitude, $a(t)$, and the complex exponential of the phase, $e^{i\varphi(t)}$. In

particular, if the support of the FT of $a(t)$ is the interval $[A_{min}, A_{max}]$ and the support of the FT of $e^{i\varphi(t)}$ is the interval $[P_{min}, P_{max}]$, it is sufficient to have

$$A_{min} + P_{min} > 0 \quad (3.7)$$

for the phase of the analytic signal to be meaningful.

An important remark must be made about the extraction of phases of linear combinations of signals. Since this thesis deals with measurements which result from linear mixtures of sources, one must show that the analytic signal of a mixture of those sources is equal to the corresponding mixture of the analytic signals of the sources. In mathematical terms, we must show that, if

$$y(t) = \sum_j s_j(t), \quad (3.8)$$

then the analytic signal of $y(t)$, denoted by $\tilde{y}(t)$, will obey

$$\tilde{y}(t) = \sum_j \tilde{s}_j(t), \quad (3.9)$$

where $\tilde{s}_j(t)$ is the analytic signal of the j -th source. The proof is straightforward: one must merely note that the analytic signal $\tilde{x}(t)$ is a linear function of its input signal $x(t)$, since the operations in equations (3.1), (3.2) and (3.3) are all linear.

We can therefore conclude that in any situation where one deals with signals which are the result of linear mixtures of sources, all of which have non-negative amplitudes and obey condition (3.7), the analytic signal can be employed as a procedure which allows us to extract meaningful phases from real signals.

3.2 Phase Synchrony

The original motivation of this work was to develop source separation algorithms appropriate to the study of synchrony in brain electrophysiological signals, such as the electroencephalogram (EEG) and the magnetoencephalogram (MEG). An in-depth overview of neuroscience is completely out of the

scope of this thesis. The following subsection intends to provide a starting set of references for a reader unfamiliar with this field, motivate why synchrony is relevant for neuroscience, and why it is an interesting topic in itself. A reader interested in neuroscience fundamentals may find [30] to be a good starting point.

Before discussing synchrony in the neuroscience domain, it is important to remark that synchrony is prevalent in many other physical systems, such as organ pipes, electrical circuits, laser beams, astrophysical objects, some types of fireflies, and even among humans and members of other mammal species. More examples, as well as very good overviews of the mathematical formulation of synchrony, can be found in [46, 66, 77].

3.2.1 Neuroscience Motivation

The number of neurons in the human brain was originally grossly estimated to be around 10^{11} [89, 6]. Recent, precise estimates place this number around 8.6×10^{10} [35]. The firing of a neuron involves the travel of an action potential along its axon. These action potentials trigger the release of neurotransmitters in the connections between neurons (called *synapses*), and these in turn cause electrical potentials in the post-synaptic neurons. While the electrical or magnetic activity of a single post-synaptic potential is not measurable through the skull, if multiple neurons fire simultaneously, and if their post-synaptic receptors (the *dendrites*) are aligned, this activity may be measurable as an EEG or MEG signal. Good overviews of the physical phenomena behind EEG and MEG signals can be found in [31, 59, 72].

The neuroscience community has shown a great deal of interest in synchrony. While researchers had measured synchrony in mammal brains in the late 1980s [27], to the author's knowledge, widespread interest from neuroscientists began in the mid 1990s [20], and the second half of that decade saw several experimental verifications that hinted at the role of synchrony in the brain. Two of

Disorder	Anomalies in neural synchrony
Schizophrenia	Reduction of local- and long-range synchronization
Epilepsy	Increase in local synchrony; evidence for a reduction in long-range synchronization
Autism	Reduced functional connectivity; preliminary evidence for impaired neural synchrony
Alzheimer's disease	Reduced neural synchrony during resting state; evidence for reduced functional connectivity
Parkinson's disease	Increase in neural synchrony in the basal ganglia, but also between subcortical-cortical structures

Tab. 3.1: Neurological pathologies associated with anomalous synchrony patterns of the human brain. Adapted from [84, Table 1].

the most impactful works are, perhaps, [82] and [48]. Of particular relevance to this thesis, as motivating factors, are the verification that the scalp conducts electrical activity and that, therefore, the EEG signals can be considered to be a mixture of sources from inside the scalp [60], and the finding of correlations between several pathologies and anomalous synchrony patterns in the brain [84]. Table 3.1 presents a list of some pathologies and of the corresponding synchrony anomalies.

Today, synchrony continues to be an active area of research in the neuroscience community. Many influential researchers consider phase synchrony a fundamental mechanism for understanding the human brain [74, 23, 24, 90, 25], and it has been found to be involved in many brain functions. Findings include:

- An involvement of synchrony in the processing of learning from mistakes, which appears to be similar between rodents and humans [56].
- Preliminary evidence that the hippocampus, one of the central areas in

the brain, “tunes” into different frequencies depending on the need to use path memorization (≈ 35 Hz) or visual landmarks (≈ 60 Hz) in orientation [14]. There is now speculation that synchrony plays a role in prioritizing different sources of information.

- Indications that the memorization of smells, at least in rodents, is connected to synchronized activity between the entorhinal cortex (part of the medial temporal lobe) and the hippocampus, with a frequency around 20 Hz [40].

Also, despite the fact that the association between synchrony and some pathologies has been known for quite a while [84], recent findings keep challenging old knowledge. For example, recent data suggest not only a reduction in local and long-range connectivity in schizophrenia, but also an increase in connectivity in the brain’s default mode network [67, 88].¹

Apart from synchrony, researchers have used other criteria to measure connectivity in the brain. Among these, coherence is a popular choice. For example, [86] applied ICA to MEG recordings and then measured coherence between the resulting sources (in subsequent work [55], a similar approach was presented replacing coherence with synchrony). [57] proposed using the imaginary part of coherence to detect interaction between brain regions, although this was not used to perform source separation. In subsequent work, the same group proposed a BSS technique based on diagonalization of anti-symmetrized cross-correlation matrices [58]. BSS, in particular ICA, has also seen widespread use as a tool for artifact removal in EEG [54] and MEG [87].

A popular measure of synchrony in neuroscience studies is the Phase Locking Factor (PLF), sometimes also called Phase Locking Value (PLV) (see, *e.g.*, [81, 80, 68, 22, 76]). It is introduced in the next section.

¹ The default mode network is part of the rest-state network, a network of regions in the brain that have increased activity when humans are not performing any specific task. The activity in these regions is measurably reduced when humans engage in tasks.

3.2.2 Phase Locking Factor

This section provides a formal way to measure synchrony between two signals. Unlike in section 3.1, in this section and the remainder of the document, all signals are discrete-time signals.

Given a complex discrete-time signal $s(t)$, we define its *phase* as its argument: $\phi(t) \equiv \arg[s(t)]$. Consider now two signals, $s_j(t)$ and $s_k(t)$, with phases $\phi_j(t)$ and $\phi_k(t)$, respectively, and define their *phase lag* as $\Delta\phi_{jk}(t) \equiv \phi_j(t) - \phi_k(t)$. The *Phase Locking Factor* (PLF) between these two signals is defined as

$$\varrho_{jk} \equiv \left| \frac{1}{T} \sum_{t=1}^T e^{i\Delta\phi_{jk}(t)} \right| = \left| \left\langle e^{i\Delta\phi_{jk}(t)} \right\rangle \right|, \quad (3.10)$$

where $\langle \cdot \rangle$ is the time average operator. The PLF has the following properties:

1. $0 \leq \varrho_{jk} \leq 1$;
2. $\varrho_{jk} = 1$ if and only if $e^{i\Delta\phi(t)}$ does not depend on t . This is equivalent to $\Delta\phi(t)$ being constant modulo 2π . A third equivalent statement is that the phase of each source is equal to a source-dependent and time-independent term plus a source-independent and time-varying term: $\phi_j(t) = \phi_j + \phi(t)$.
3. In particular, the PLF of a signal with itself is 1: $\varrho_{jj} = 1$.
4. ϱ_{jk} is invariant to the scale of the signals: if s_j is multiplied by α_j and s_k is multiplied by α_k , where α_j and α_k are two nonzero complex scalars, then the PLF between $\alpha_j s_j$ and $\alpha_k s_k$ is the same as the PLF between s_j and s_k .

$\varrho_{jk} = 0$ occurs, by definition, if and only if $\sum_{t=1}^T e^{i\Delta\phi_{jk}(t)} = 0$. Unlike the $\varrho_{jk} = 1$ case which is equivalent to a precise characterization of the two signals, the case $\varrho_{jk} = 0$ can correspond to vastly different situations. In the limit where the observation period T tends to $+\infty$, and under ergodicity conditions, the following cases, among many others, yield zero PLF:

-
- $\phi_j(t) = 0$, and $\phi_k(t)$ uniformly distributed in $[0, 2\pi)$;
 - $\phi_j(t)$ and $\phi_k(t)$ are both uniformly distributed in $[0, 2\pi)$ and are statistically independent;
 - $\Delta\phi_{jk}(t) = \alpha t$ where α is not a multiple of 2π .

For a finite observation period T , even these cases may yield non-zero values of ϱ_{jk} , which will tend to become smaller as the observation period grows larger.

We can now provide a precise definition of the term “synchrony”: if $\varrho_{jk} = 1$ we say that signals $s_j(t)$ and $s_k(t)$ are *perfectly synchronized*; if $\varrho_{jk} = 0$ we say that they are *perfectly unsynchronized*; if $0 < \varrho_{jk} < 1$ we say that they are *partially synchronized*.

4. SEPARATION OF SYNCHRONOUS SOURCES

We now have all the elements to formally state the Separation of Synchronous Sources problem: a linear and instantaneous BSS problem, following model (2.3), is called a *Separation of Synchronous Sources (SSS)* problem if all pairs of sources are assumed to have pairwise PLFs of 1. We will later consider perturbed versions of this problem and still call it SSS.

One of the main motivations behind this work is the empirical verification that mixing destroys synchrony information, a fact which we verified empirically in multiple works. Consider, for example, the sources depicted in the top-left panel of figure 4.1, adapted from [P5]. These sources have pairwise PLFs of 1 with one another, as depicted in the top-right panel. If these sources are mixed using a 3×3 matrix whose entries are random and drawn from a $\text{Uniform}(-1, 1)$ distribution, a typical result is shown in the bottom-left panel of the figure. As shown on the bottom-right panel, the PLFs of these mixtures are no longer equal to 1, although signals 2 and 3 still exhibit a rather high mutual PLF. One of the main results in this work is a proof that, in the presence of perfectly synchronous sources, mixing will always result in imperfectly synchronous sources (theorem 4.1.1), showing that the example in figure 4.1 is not a coincidence. Another main result of this work is the design, implementation and analysis of two algorithms to solve the SSS problem. The two algorithms are very different in their philosophy:

- In Independent Phase Analysis (IPA), the basic rationale is as follows: Theorem 4.1.1 states that, under certain conditions, maximizing the pair-

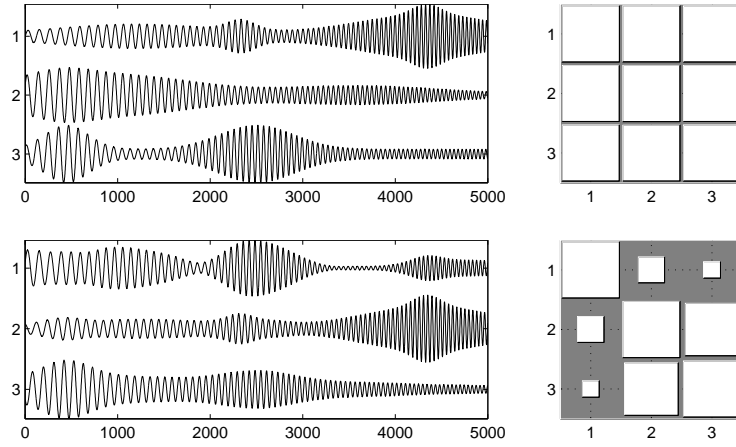


Fig. 4.1: Top row: The real part of the three original sources (left) and PLFs between them (right). Bottom row: The real part of the three mixed signals (left) and PLFs between them (right). On the right column, the area of the square in position (i, j) is proportional to the PLF between the signals i and j . Therefore, large squares represent PLFs close to 1, while small squares represent values close to zero. In this example, the second and third sources have phase lags of $\frac{\pi}{6}$ and $\frac{\pi}{3}$ radians relative to the first source, respectively.

wise PLFs will yield a solution equal to the original sources, apart from the usual indeterminacies. IPA explicitly maximizes an objective function which is the sum of the squares of all pairwise PLFs, plus a regularization term which penalizes singular solutions.

- In Phase Locked Matrix Factorization (PLMF), the basic rationale is as follows: if the sources are synchronous, then they can be expressed as a product of matrices with specific properties, which are presented below. PLMF explicitly models the sources as such a product and tries to estimate each factor separately. In doing this, singular solutions are automatically avoided.

These algorithms are explained in more detail in the next subsections, along with their theoretical properties and experimental results. Throughout this section, all signals are assumed to be complex-valued.

4.1 Problem Definition and Identifiability

A crucial aspect of SSS is that it is “sufficiently well-defined” in a sense similar to the one in which ICA was, but with different assumptions. The following theorem and corollary (both derived in [P3]¹) establish that fact.

Theorem 4.1.1. *Consider a set of N sources which follow the SSS model, such that $s_k(t) \equiv A_k(t)e^{i(\phi(t)+\phi_k)}$. Also, consider a linear combination of those sources, given by $\hat{\mathbf{s}} = \mathbf{G}\mathbf{s}$ with square \mathbf{G} . Furthermore, assume that:*

1. *None of the sources and none of the linear combinations are identically zero.*

2. *ϕ_1, ϕ_2, \dots are all distinct modulo π .*

¹ The proof of the corollary was somewhat unclear in [P3]. For that reason, we present it here again in a clearer form.

3. The amplitudes $A_i(t)$ are linearly independent (i.e., the matrix \mathbf{A} , with entry (i, t) given by $A_i(t)$, has maximum row rank) and positive.

Then, if the components of $\hat{\mathbf{s}}$ have the form

$$\hat{s}_j(t) = C_j(t)e^{i(\phi(t)+\alpha_j)}, \quad (4.1)$$

with positive amplitudes C_j one necessarily has, for every j , that $\hat{s}_j = Ls_k$ for some k , where L is a non-zero real number. Equivalently, each row of \mathbf{G} has exactly one non-zero element.

Corollary 4.1.2. *In the conditions of theorem 4.1.1, the following two statements are equivalent:*

1. For all $j \neq k$, \hat{s}_j and \hat{s}_k are linearly independent.
2. \mathbf{G} is non-singular and is thus a permutation of a diagonal matrix with nonzero diagonal elements.

In particular, to successfully extract all the original sources up to permutation, scale and sign change, \mathbf{G} must be non-singular.

Proof. Proving $1 \implies 2$ is trivial: suppose \mathbf{G} is singular. Since each of its rows has at most one non-zero element, then two of its rows (say, rows j and k) have non-zero elements in the same column (say, column ℓ). This immediately implies that $\hat{s}_j = g_{j\ell}s_\ell$ and $\hat{s}_k = g_{k\ell}s_\ell$, and thus \hat{s}_j and \hat{s}_k are linearly dependent.

We now prove that $2 \implies 1$. This is also straightforward: if for a certain pair j, k , we have $\hat{s}_j = L\hat{s}_k$, then by theorem 4.1.1 these two mixtures must both be equal to the same source up to scale and sign: $\hat{s}_j = L\hat{s}_k = Ks_\ell$. This means that rows j and k of \mathbf{G} both have exactly one nonzero element in the ℓ -th column, thus making \mathbf{G} singular. \square

This theorem and corollary state that, if we find a linear combination of the original sources with the form (4.1), and if the assumptions are met, then we

have found the original sources up to the typical BSS indeterminacies. Note, however, the specific form of (4.1): the linear combinations $\hat{\mathbf{s}}$ have *exactly the same common oscillation* $\phi(t)$ as the original sources \mathbf{s} . We will use a theorem presented later (theorem 4.3.3, in section 4.3.4) to show that, under mild assumptions, all linear combinations with pairwise PLFs of 1, *i.e.*, combinations of the form

$$\hat{s}_j(t) = C_j(t)e^{i(\psi(t)+\alpha_j)}, \quad (4.2)$$

must have $\psi(t) = \phi(t) + \beta$, where β is some real number. Since β can be absorbed by the α_j phase offsets, one can actually state that one can only find linear combinations of the sources which have pairwise PLFs of 1 if those linear combinations have the form (4.1).

Therefore, this theorem and corollary assert that, similarly to the ICA case, in SSS the sources can be recovered through maximization of the PLF, with the permutation, scale and sign indeterminacies. The requirements on the data are quite different from those of ICA, though. Let us now take a closer look at those requirements:

1. ICA is identifiable only if one has at most one Gaussian source – no such constraint is needed for SSS.
2. On the other hand, SSS's identifiability requires that no pair of sources be either in-phase or in anti-phase.
3. SSS's identifiability also requires that the amplitudes of the sources be linearly independent.
4. SSS's identifiability also requires that the gain matrix \mathbf{G} is non-singular. Since the mixing matrix \mathbf{M} is assumed non-singular in BSS problems, this requires that the demixing matrix be non-singular. While this is also a requirement for ICA identifiability, we shall see later that fulfilling this requirement is considerably harder in SSS algorithms.

The reader is referred to the proof of theorem 4.1.1 [P3] for the mathematical reasons for requirements 2–4. We now present simple counterexamples illustrating why each of these requirements is needed.

Example 4.1.3. *Let s_1 and s_2 be two signals exactly in-phase, with a phase lag of zero:*

$$\begin{aligned} s_1(t) &= a_1(t)e^{i\phi(t)} \\ s_2(t) &= a_2(t)e^{i\phi(t)}, \end{aligned} \quad (4.3)$$

with $a_1(t) = 1$ and $a_2(t) = 2 + t$, with $t = 1, 2, \dots, T$. Let

$$\mathbf{G} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad (4.4)$$

and thus $\hat{s}_1(t) = b_1(t)e^{i\phi(t)}$ and $\hat{s}_2(t) = b_2(t)e^{i\phi(t)}$ with $b_1(t) = 3 + t$ and $b_2(t) = 1 + t$. Clearly, all requirements are satisfied except that the phase lag is not different from 0 modulo π .

Since $b_1(t), b_2(t) > 0$ for all t , the PLF between \hat{s}_1 and \hat{s}_2 is 1. However, \hat{s}_1 and \hat{s}_2 are not equal to the original sources s_1 and s_2 , even considering permutation, scaling and sign indeterminacies.

Example 4.1.4. *Let s_1 and s_2 be two signals with linearly dependent amplitudes and with a phase lag $\Delta\phi \notin \{0, \pi\}$:*

$$\begin{aligned} s_1(t) &= a(t)e^{i\phi(t)} \\ s_2(t) &= 2a(t)e^{i\phi(t)+\Delta\phi}, \end{aligned} \quad (4.5)$$

with $t = 1, 2, \dots, T$. Let

$$\mathbf{G} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}. \quad (4.6)$$

Clearly, all requirements are satisfied except that the amplitudes of the sources are linearly dependent.

Since

$$\hat{s}_1(t) = a(t) \left[e^{i\phi(t)} + 2e^{i\phi(t)+\Delta\phi} \right] = a(t)e^{i\phi(t)} \left[2e^{i\Delta\phi} + 1 \right] \quad (4.7)$$

$$\hat{s}_2(t) = a(t) \left[-e^{i\phi(t)} + 2e^{i\phi(t)+\Delta\phi} \right] = a(t)e^{i\phi(t)} \left[2e^{i\Delta\phi} - 1 \right], \quad (4.8)$$

the PLF between \hat{s}_1 and \hat{s}_2 is 1. However, \hat{s}_1 and \hat{s}_2 are not equal to the original sources s_1 and s_2 , even considering permutation, scaling and sign indeterminacies.

Example 4.1.5. Let s_1 and s_2 be any two signals, and let

$$\mathbf{G} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad (4.9)$$

Clearly, all requirements can be made to be satisfied except that the mixing matrix is singular.

We have $\hat{s}_1(t) = \hat{s}_2(t) = s_1(t) + s_2(t)$. Since the PLF of a signal with itself is 1 (see equation (3.10)), the PLF of signals \hat{s}_1 and \hat{s}_2 is 1. However, \hat{s}_1 and \hat{s}_2 are not equal to the original sources s_1 and s_2 , even considering permutation, scaling and sign indeterminacies.

Requirements 2 and 3, that the amplitudes be linearly independent and that phase lags be different from 0 modulo π , can be considered mild requirements: they will be met in the vast majority of situations. Note, however, that in practice we will always be dealing with a finite observation period T . In that case, as will be empirically shown in section 4.3, the performance of SSS algorithms is stable for phase lags that are far from 0 and π , but degrade as they approach those values.

Requirement 4 is much more profound. It corresponds to a fundamental difference between ICA and SSS:

- In ICA, if the mixing matrix is non-singular, “maximizing independence” poses no risk of leading the algorithm towards singular unmixing matrices,

since a signal is not independent of itself.²

- In SSS, even with a non-singular mixing matrix, maximizing the PLF poses the risk of leading the algorithm towards singular matrices, since a solution with $\hat{s}_1 = \hat{s}_2 = \dots = \hat{s}_P$ trivially makes all PLFs equal to 1. Therefore, some means must be employed to prevent this from happening.

We shall see in section 4.2 that this requirement is, to some extent, a drawback of IPA, one of the SSS algorithms proposed in this thesis.

4.2 Algorithm: Independent Phase Analysis

Independent Phase Analysis (IPA) explicitly maximizes the PLF of the estimated sources as a function of the demixing matrix \mathbf{W} . Let $\hat{\mathbf{s}}(t) \equiv \mathbf{W}^T \mathbf{y}(t)$ denote the vector of estimated sources, let $\hat{s}_j(t)$ denote the j -th estimated source, and let $\Delta\hat{\phi}_{jk}(t)$ denote the phase lag between estimated sources j and k . These estimated quantities are, naturally, functions of the demixing matrix \mathbf{W} – this dependency will sometimes be omitted for clarity in the following. The PLF between estimated sources j and k , denoted by $\hat{\varrho}_{jk}$, is given by:

$$\hat{\varrho}_{jk}(\mathbf{W}) \equiv \left| \frac{1}{T} \sum_{t=1}^T e^{i\Delta\hat{\phi}_{jk}(t)} \right| = \left| \left\langle e^{i\Delta\hat{\phi}_{jk}(t)} \right\rangle \right|. \quad (4.10)$$

This estimated PLF is also a function of \mathbf{W} , since $\Delta\hat{\phi}_{jk}$ is a function of it.

IPA maximizes the following objective function:

$$J(\mathbf{W}) \equiv (1 - \lambda) \sum_{j \neq k; j, k=1, \dots, N} \hat{\varrho}_{jk}^2(\mathbf{W}) + \lambda \log |\det \mathbf{W}|. \quad (4.11)$$

The first term of $J(\mathbf{W})$ is the sum of the squares of all PLFs between pairs

² In fact, ICA algorithms do not maximize independence, but rather optimize some surrogate of it. Even so, the most frequently used ICA algorithms do avoid singular solutions. In some of them, the search is reduced to the space of orthogonal matrices, which immediately avoids singular matrices.

of different sources.³ The second term is a regularization term which we will discuss further below.

IPA constrains the rows of \mathbf{W} to have unit Euclidean norm. The set of $N \times N$ matrices whose rows have unit norm will be denoted as \mathcal{S} . Thus, the optimization problem to be solved in IPA is

$$\max_{\mathbf{W} \in \mathcal{S}} J(\mathbf{W}) \quad (4.12)$$

We proved [P4] that the gradient of J relative to a column \mathbf{w}_j of the matrix \mathbf{W} is given by

$$\nabla_{\mathbf{w}_j} J = 4 \frac{1-\lambda}{N^2} \sum_{k=1}^N \hat{\varrho}_{jk} \left\langle \sin \left[\hat{\Psi}_{jk} - \Delta \hat{\phi}_{jk}(t) \right] \frac{\mathbf{\Gamma}(t)}{Y_j(t)^2} \right\rangle \mathbf{w}_j, \quad (4.13)$$

where $S_j \equiv |\hat{s}_j|$ where \hat{s}_j is the j -th estimated source, $\hat{\phi}_j \equiv \text{angle}(\hat{s}_j)$ and $\Delta \hat{\phi}_{jk}(t) \equiv \hat{\phi}_j(t) - \hat{\phi}_k(t)$ is the instantaneous phase difference between two estimated sources, $\hat{\Psi}_{jk}(t) \equiv \langle \Delta \hat{\phi}_{jk}(t) \rangle$ is the average phase difference between two estimated sources, and $\mathbf{\Gamma}(t) = \mathbf{y}_h(t) \mathbf{y}^T(t) - \mathbf{y}(t) \mathbf{y}_h^T(t)$, where $\mathbf{y}_h = \text{Imag}(\hat{\mathbf{y}})$ is the imaginary part of \mathbf{y} . $\mathbf{\Gamma}(t)$ is a matrix that can be pre-computed, before the optimization of $J(\mathbf{W})$, because it depends only on the data.

4.2.1 Regularization

Let us temporarily focus on the case $\lambda = 0$. For this case, we have $0 \leq J(\mathbf{W}) \leq N(N-1)$. The reason for this is that each $\hat{\varrho}_{jk}^2$ term in the summation is between 0 and 1, and there are $N(N-1)$ such terms.

If the assumptions of the SSS problem hold (*i.e.*, if the original sources have PLFs of 1 with one another and the mixing is linear and instantaneous), then one global maximizer of $J(\mathbf{W})$ corresponds to the case where the estimated sources are equal to the original sources. This will yield $\hat{\varrho}_{jk} = \varrho_{jk} = 1$ for all j, k , and

³ Note that, since the PLF of a signal with itself is always 1, we could include the case $j = k$ in the summation without changing the solution.

this is the maximum value each of the $\hat{\varrho}_{jk}$ can take. This case corresponds to making $\mathbf{W}^T = \mathbf{M}^{-1}$, up to permutation, scaling and sign change.

Theorem 4.1.1 ensures that the solution $\mathbf{W}^T = \mathbf{M}^{-1}$ is the only one where $\det(\mathbf{W}) \neq 0$, apart from permutation, scaling and sign change. In more rigorous terms, there are multiple non-singular solutions of the form $\mathbf{W}^T = \mathbf{PDM}^{-1}$, where \mathbf{P} is some permutation matrix, and \mathbf{D} is a diagonal matrix with nonzero entries in its diagonal.⁴ We call all solutions of this form *desirable solutions* of IPA. However, there are many other global maximizers of $J(\mathbf{W})$, which have $\det(\mathbf{W}) = 0$. For example, there are N solutions of the form

$$\hat{s}_1 = \hat{s}_2 = \dots = \hat{s}_N = s_i, \quad (4.14)$$

for some $i \in \{1, \dots, N\}$. This solution corresponds to making all estimated sources equal to the i -th true source. It is equivalent to having a matrix \mathbf{W} such that the gain matrix has all entries in the i -th row equal to 1 and all other entries equal to zero. It is simple to see that this is a global maximizer for $\lambda = 0$: all estimated sources are equal to one another, thus their pairwise PLFs are all equal to 1. Apart from these N solutions, there are many other ones, which are not good solutions, in the sense that they are global maximizers of $J(\mathbf{W})$, for $\lambda = 0$, but do not recover the original sources. We call *undesirable solutions* all global maximizers of $J(\mathbf{W})$ that are not desirable solutions. Theorem 4.1.1 ensures that all undesirable solutions have $\det(\mathbf{W}) = 0$.

The fact that the maximizers of the first term of $J(\mathbf{W})$ can correspond to desirable or undesirable solutions is the motivation for the second term, which explicitly penalizes undesirable solutions: these solutions make the second term equal to $-\infty$, whereas desirable solutions make it finite. The parameter λ controls the tradeoff between the first term and the regularization one.

The previous reasoning motivates the fact that $0 < \lambda < 1$ is a better choice

⁴ The diagonal entries of \mathbf{D} are not free: they must be such that the rows of \mathbf{W} have unit norm.

than $\lambda = 0$. However, there is a significant disadvantage in using regularization: Theorem 4.1.1 characterizes the maximizers of the first term, but the objective function now also contains a regularization term. Theorem 4.1.1 now does not characterize solutions for this case. In general, for $0 < \lambda < 1$, maximizers of $J(\mathbf{W})$ will not exactly correspond to desirable solutions.

The requirement for the columns of \mathbf{W} to be normalized can now be justified: if \mathbf{W} was not constrained, $|\det \mathbf{W}|$ could be made arbitrarily large, without affecting the first term, by scaling the columns of \mathbf{W} .

4.2.2 Optimization Strategy

The existence of multiple solutions for $\lambda = 0$ shows that IPA is not a concave optimization problem [13] and we have verified, in practice, that optimizing IPA is difficult if a fixed value of λ is used. Using small values (or zero) for λ tends to yield singular solutions often. Ideally, one would wish to avoid undesirable solutions, which would force us to use relatively large values of λ , but one would also wish the assurance, through Theorem 4.1.1, that a solution with $\det \mathbf{W} \neq 0$ is a desirable one, which would force us to use $\lambda = 0$. To reconcile these two aspects, we proposed in [P5] that the value of λ should start at a relatively large value, and be decreased throughout the optimization, such that at the end λ is zero. The reasoning is that the initial stages, with large λ , force \mathbf{W} away from undesirable solutions; the later stages, with smaller λ , corresponds to the optimization of functions which are progressively better approximations of the first term, and are therefore expected to yield results that are progressively closer to desirable solutions, yielding a desirable solution in the final step, with $\lambda = 0$. We have no theoretical guarantee that this method will yield a desirable solution. However, we have verified in practice that it yields much better results than using a fixed value of λ , as shown in section 5.1.

In the results that follow, each epoch with a fixed λ is optimized with 200 iterations of gradient ascent, as shown in table 4.1. After this, MATLAB's

INDEPENDENT PHASE ANALYSIS
1: Input $\mathbf{y}(t)$, η , k_{max}
2: $\tilde{\mathbf{y}}(t) \leftarrow$ analytic signal of $\mathbf{y}(t)$ (eqs. (3.1), (3.2), (3.3))
3: $\varphi_j(t) \leftarrow \text{angle}(\tilde{y}_j(t))$, $j = 1, \dots, N$
4: $Y_j(t) \leftarrow \tilde{y}_j(t) $
5: Initialize $\mathbf{W} \sim \mathcal{N}(0, 1)$; set $k = 1$
6: repeat
7: $\hat{\mathbf{s}}(t) \leftarrow \mathbf{W}^T \tilde{\mathbf{y}}(t)$
8: $\Delta \mathbf{W} \leftarrow \text{Eq. (4.13)}$
9: $\mathbf{W} \leftarrow \mathbf{W} + \eta \Delta \mathbf{W}$
10: $\mathbf{w}_j \leftarrow \mathbf{w}_j / \ \mathbf{w}_j\ $, $j = 1, \dots, N$
11: $k \leftarrow k + 1$
12: until ($\ \Delta \mathbf{W}\ < \epsilon$) or ($k > k_{max}$)

Tab. 4.1: Pseudocode for the fixed- λ version of IPA using gradient ascent. The varying λ version merely runs the fixed version until convergence multiple times with decreasing values of λ .

implementation of BFGS is run until convergence. The solution found in this manner is used to initialize the optimization with the next value of λ .

4.3 Algorithm: Phase Locked Matrix Factorization

As discussed in the previous section, non-regularized IPA ($\lambda = 0$) suffers from the drawback of allowing undesirable solutions, and regularized IPA ($\lambda > 0$) suffers from a bias due to the regularization term. These disadvantages motivated us to design, in a more principled way, an SSS algorithm which would guarantee exact separation. Phase Locked Matrix Factorization (PLMF), the algorithm detailed in this section, accomplishes this, and yields substantially

better separation results than IPA.

4.3.1 General Approach

Whereas IPA tries to estimate the unmixing matrix \mathbf{W} directly from the data, PLMF is based on a factorized model of synchronous sources. Let us decompose, elementwise, the complex-valued sources into their absolute values and arguments:

$$\mathbf{S} = \mathbf{A} \odot \mathbf{\Phi}, \quad (4.15)$$

where the entries of \mathbf{A} are real-valued and non-negative, those of $\mathbf{\Phi}$ are complex-valued and have unit absolute value, and the symbol \odot denotes elementwise (or Hadamard) product. We call \mathbf{A} and $\mathbf{\Phi}$ the sources' *amplitudes* and *phases*, respectively.

If the sources \mathbf{S} are perfectly synchronous, then the phases $\mathbf{\Phi}$ can be further decomposed as

$$\mathbf{\Phi} = \mathbf{z}\mathbf{f}^T, \quad (4.16)$$

where \mathbf{z} and \mathbf{f} are complex-valued vectors of sizes N and T , respectively, and whose entries all have unit absolute value. To see why this decomposition is always possible, recall from section 3.2.2 that, if a set of sources are perfectly synchronous, then the phase of the j -th source could be written as $\phi_j(t) = \phi_j + \psi(t)$. Equation (4.16) follows from setting the j -th element of \mathbf{z} as $e^{i\phi_j}$, and the t -th element of \mathbf{f} as $e^{i\psi(t)}$. Notice that there are more assignments that uphold equation (4.16): one can also set the j -th element of \mathbf{z} to $e^{i\phi_j}e^{i\gamma}$ and the t -th element of \mathbf{f} to $e^{i\psi(t)}e^{-i\gamma}$, for any real scalar γ . This is a new indeterminacy, specific to PLMF, which we call the *rotation indeterminacy*. Note that, while these assignments yield different factors \mathbf{z} and \mathbf{f} , they do not alter the sources \mathbf{S} .

Consecutively applying equations (2.8), (4.15) and (4.16) yields, for syn-

chronous sources and no observation noise,

$$\mathbf{Y} = \mathbf{M}[\mathbf{A} \odot (\mathbf{z}\mathbf{f}^T)]. \quad (4.17)$$

If one defines $\mathbf{D}_{\mathbf{z}}$ as a $N \times N$ diagonal matrix with diagonal entries equal to the entries of \mathbf{z} , and analogously defines $\mathbf{D}_{\mathbf{f}}$ as a $T \times T$ matrix with diagonal entries equal to the entries of \mathbf{f} , then equation (4.17) can also be written as

$$\mathbf{Y} = \mathbf{M}\mathbf{D}_{\mathbf{z}}\mathbf{A}\mathbf{D}_{\mathbf{f}}. \quad (4.18)$$

The basic idea behind PLMF is to minimize the squared error of this model relative to the observed data:

$$\min_{\mathbf{M}, \mathbf{A}, \mathbf{z}, \mathbf{f}} \frac{1}{2} \|\mathbf{Y} - \mathbf{M}\mathbf{D}_{\mathbf{z}}\mathbf{A}\mathbf{D}_{\mathbf{f}}\|_F^2, \quad (4.19)$$

$$\begin{aligned} \text{s.t.: } & 1) \max_{i,j} |m_{ij}| = 1 \\ & 2) |z_j| = 1 \text{ for all } j \\ & 3) |f_t| = 1 \text{ for all } t \end{aligned}$$

where $\|\cdot\|_F$ is the Frobenius norm. The first constraint forces the largest absolute value among all elements of \mathbf{M} to be 1, a constraint that we shall discuss further below. 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, while \mathbf{z} and \mathbf{f} are complex ones.⁵

Let the term *desirable solutions* denote all solutions such that the sources $\mathbf{S} = \mathbf{D}_{\mathbf{z}}\mathbf{A}\mathbf{D}_{\mathbf{f}}$ are equal to the original sources up to permutation, scaling and sign change. Suppose that the data \mathbf{Y} are indeed generated according to the PLMF model (4.17). Then, the true solution (*i.e.*, the values of $\mathbf{M}, \mathbf{A}, \mathbf{z}, \mathbf{f}$ used

⁵ We also experimented adding the constraint $\mathbf{A} \geq 0$, since our experiments will use amplitudes which obey this constraint. However, adding this constraint makes the algorithm sometimes become “stuck” with many entries of \mathbf{A} equal to zero and unable to depart from that situation, even when the true matrix has only positive entries.

to generate the data) clearly makes the cost function in equation (4.19) attain its minimum possible value of zero. That solution also obeys constraints 2 and 3, by construction. The true solution will probably not obey constraint 1; however, it is easy to obtain a new solution that obeys it, by multiplying \mathbf{M} by an appropriate scalar and dividing \mathbf{A} by the same scalar. Since the sources are only affected by a scaling factor, this new solution is a desirable one.

There is, therefore, a global minimizer of the objective function of (4.19) which yields a desirable solution. Constraint 1 ensures that there is no scale indeterminacy, but due to the permutation and rotation indeterminacies, there is in fact an infinite number of desirable solutions, all of which minimize the cost function in (4.19).

The PLMF problem is non-convex [13]. There are two reasons for this:

- The cost function involves the product of several variables.
- The feasible sets, *i.e.*, the sets of values of the variables which obey constraints 1, 2, and 3, are not convex sets.

One of the results in this thesis is that, under certain conditions, all global optima of this problem correspond to desirable solutions, as will be shown further below.

The basic PLMF algorithm iteratively solves the minimization problem in (4.19). Each iteration is a sequence of four steps. At each step, three of the four variables are kept fixed, and the minimization problem is solved with respect to the remaining variable. This is known in the literature as a Block Nonlinear Gauss-Seidel (BNGS) method [28], or Block Coordinate Descent, or Alternating Optimization, and it is not guaranteed to converge to the optimal solution in general. It does converge under some assumptions [28], which are not met in our case. We discuss this aspect further in section 6.2.

The first version of PLMF tackled the minimization problem in (4.19) directly, *i.e.*, optimizing on all four variables using the BNGS method. However,

we have since shown that PLMF can also be solved by finding a correct value for \mathbf{f} by solving a relaxed version of (4.19).⁶ This value of \mathbf{f} is then kept fixed, and the BNGS method is then applied to optimize on the three remaining variables $\mathbf{M}, \mathbf{A}, \mathbf{z}$. To distinguish these two approaches, when necessary, we will refer to the first one as the “1-stage” PLMF and to the second one as the “2-stage” PLMF. Whenever we do not make a distinction, we will be referring to the second version.

Experimental comparisons have shown that the 2-stage PLMF outperforms the 1-stage version. The reason for this, which we verified experimentally, is that 1-stage PLMF yields local minima frequently; the 2-stage version is not immune to this, but returns local minima less often. Hence, we now proceed to discuss only the 2-stage version; the 1-stage version was presented in [P6].

4.3.2 First subproblem

In the first stage of 2-stage PLMF, the goal is to estimate the common oscillation \mathbf{f} up to the rotation indeterminacy. This estimation is performed by solving the subproblem

$$\min_{\mathbf{H}, \mathbf{A}, \mathbf{f}} \frac{1}{2} \|\mathbf{Y} - \mathbf{H}\mathbf{A}\mathbf{D}\mathbf{f}\|_F^2, \quad (4.20)$$

$$\begin{aligned} \text{s.t.: } & 1) \max_{i,j} |h_{ij}| = 1 \\ & 2) |f_t| = 1 \text{ for all } t, \end{aligned}$$

where \mathbf{H} can be any complex matrix with the same dimensions as \mathbf{M} , as long as the largest absolute value among its entries is 1, to fulfill constraint 1, and \mathbf{f} is complex with entries having unit absolute value, as before. This formulation collapses the product $\mathbf{M}\mathbf{D}\mathbf{z}$ from (4.19) into the matrix \mathbf{H} , which is now allowed to be any complex matrix. Note that, despite the fact that we minimize relative

⁶ Recall that, due to the rotation indeterminacy, there are infinite correct values for \mathbf{f} . The method finds one of them.

to \mathbf{H} , \mathbf{A} , and \mathbf{f} , the purpose is only to estimate \mathbf{f} . The values found for \mathbf{H} and \mathbf{A} are discarded. Since the product of a real matrix \mathbf{M} and a complex diagonal matrix \mathbf{D}_z does not span the space of all complex matrices, the minimization problem (4.20) is a relaxation of (4.19).

If the sources exactly follow the model in equation (4.18), a factorization of the form $\mathbf{Y} = \mathbf{H}\mathbf{A}\mathbf{D}_f$ always exists, since the true factorization is a special case of it. The following theorem, proven in [P11], shows that if $\mathbf{Y} = \mathbf{H}\mathbf{A}\mathbf{D}_f$ (in particular, if the sources follow the model in equation (4.18)), then finding a solution of (4.20) yields a correctly estimated \mathbf{f} , apart from a sign indeterminacy which can be easily compensated, and from the rotation indeterminacy.

Theorem 4.3.1. *Let $\mathbf{Y} = \mathbf{H}_1\mathbf{A}_1\mathbf{D}_{f1}$ with $\mathbf{H}_1 \in \mathbb{C}^{P \times N}$, $\mathbf{A}_1 \in \mathbb{R}^{N \times T}$, $\mathbf{D}_{f1} \in \mathbb{D}_1^T$, where \mathbb{D}_1^T is the set of T -by- T diagonal complex 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, $\mathbf{Y} = \mathbf{H}_2\mathbf{A}_2\mathbf{D}_{f2}$, then one necessarily has $\mathbf{D}_{f2} = \mathbf{E}\mathbf{D}_{f1}$ 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 some real number.*

This theorem only ensures a “quasi-identifiability” of \mathbf{f} , since \mathbf{D}_f is determined up to multiplication by matrix \mathbf{E} . This means that we may not obtain the true \mathbf{D}_f , for two reasons (which may occur simultaneously):

1. The first possibility is that all entries of \mathbf{D}_f are multiplied by $e^{i\gamma}$, i.e., all its entries are rotated by an angle γ . This ambiguity corresponds to the rotation indeterminacy.
2. The second possibility is that some entries of the estimated \mathbf{D}_f are multiplied by $+1$ and some by -1 . This means that some entries of \mathbf{D}_f are estimated with the wrong sign.

The first issue does not need to be solved at this point, since the rotation indeterminacy does not affect the estimated sources. It will be compensated

when we estimate \mathbf{z} , in the second subproblem (section 4.3.3).

The second issue can easily be solved if the common oscillation \mathbf{f} is *smooth*, *i.e.*, if it varies slowly with time. In that case, it is natural to expect that f_{t+1} is not very different from f_t . Therefore, to correct this sign estimation, we compute, for $t = 1, \dots, T - 1$, the quantity

$$|f_R(t) - f_R(t+1)| + |f_I(t) - f_I(t+1)|, \quad (4.21)$$

where $f_R(t)$ is the real part of the t -th 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. For a smoothly varying \mathbf{f} , we expect the values of (4.21) to be small if there is no change of sign from time t to time $t + 1$, and to be $\gtrsim \sqrt{2}$ if such a sign change occurs from t to $(t + 1)$. In our simulations we determine that there is a change in sign when

$$|f_R(t) - f_R(t+1)| + |f_I(t) - f_I(t+1)| > 1. \quad (4.22)$$

In the tests presented in section 5.2, this simple procedure successfully captures all sign changes. However, if \mathbf{f} is not sufficiently smooth, better phase unwrapping techniques must be employed [12].

4.3.3 Second subproblem

The first subproblem yields an estimate of \mathbf{f} . To motivate the second subproblem, let us take the source model (4.18) and multiply both sides of the equation by the inverse of $\mathbf{D}_\mathbf{f}$, on the right. Note that the inverse of $\mathbf{D}_\mathbf{f}$ is its complex conjugate: $\mathbf{D}_\mathbf{f}^* = \mathbf{D}_\mathbf{f}^{-1}$. This yields

$$\mathbf{Y}\mathbf{D}_\mathbf{f}^* = \mathbf{M}\mathbf{D}_\mathbf{z}\mathbf{A}. \quad (4.23)$$

The second subproblem attempts to minimize the squared difference between both sides of this equation:

$$\min_{\mathbf{M}, \mathbf{A}, \mathbf{z}} \frac{1}{2} \|\mathbf{Y}\mathbf{D}_\mathbf{f}^* - \mathbf{M}\mathbf{D}_\mathbf{z}\mathbf{A}\|_F^2, \quad (4.24)$$

$$\begin{aligned} \text{s.t.: } & 1) \max_{i,j} |m_{ij}| = 1 \\ & 2) |z_j| = 1 \text{ for all } j. \end{aligned}$$

One again has identifiability in this second subproblem, as shown by the following theorem, which was derived in [P11].

Theorem 4.3.2 (Identifiability of $\mathbf{M}, \mathbf{A}, \mathbf{z}$). *Let $\mathbf{YD}_f^* = \mathbf{MD}_{z1}\mathbf{A}_1$ with $\mathbf{M}_1 \in \mathbb{R}^{P \times N}$, $\mathbf{D}_{z1} \in \mathbb{D}_1^N$, $\mathbf{A}_1 \in \mathbb{R}^{N \times T}$, where $\mathbb{R}^{N \times T}$ denotes the set of N -by- T matrices with real entries. Further assume that the phases of all sources are different from one another modulo π (in other words, that two entries $e^{i\alpha}$ and $e^{i\beta}$ of the diagonal of \mathbf{D}_{z1} 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, $\mathbf{YD}_f^* = \mathbf{M}_2\mathbf{D}_{z2}\mathbf{A}_2$, then one necessarily has $\mathbf{M}_1 = \mathbf{M}_2$, $\mathbf{D}_{z1} = \mathbf{D}_{z2}$, and $\mathbf{A}_1 = \mathbf{A}_2$, up to permutation, scaling, and sign change.*

Importantly, note that this theorem does not state that \mathbf{D}_z is estimated up to rotation. This ensures that the rotation indeterminacy, which was a potential issue from the first subproblem, is no longer an issue. At the end of the second subproblem, if a global minimum of problem (4.24) is found, the theorem ensures that one will have a solution $\mathbf{M}, \mathbf{A}, \mathbf{z}, \mathbf{f}$ such that $\mathbf{YD}_f^* = \mathbf{MD}_z\mathbf{A}$, or equivalently, such that $\mathbf{Y} = \mathbf{MD}_z\mathbf{A}\mathbf{D}_f$ up to permutation, scaling and sign change. While two different solutions may have vectors \mathbf{z} and \mathbf{f} which differ by arbitrary rotations, the two theorems ensure that both pairs (\mathbf{z}, \mathbf{f}) yield the same sources.

This theorem assumes that all the arguments of the entries in the diagonal of \mathbf{D}_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}_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. A sketch of this proof was presented in [P11, footnote 9].

4.3.4 Global identifiability

Given that both subproblems of PLMF are identifiable (or quasi-identifiable for the first one), one may naturally ask the question: is the original problem in (4.19) identifiable? In other words, if we find a solution of that problem (not necessarily with PLMF), can we be sure that we found the original sources up to the usual indeterminacies? By combining the two Theorems 4.3.1 and 4.3.2, the answer turns out to be affirmative, with one additional assumption. Along with the assumptions of both theorems, one also needs that the amplitudes \mathbf{A} are non-negative, as shown by the following theorem, which was also derived in [P11].

Theorem 4.3.3. *Let \mathbf{Y} be data generated according to the model in equation (4.18) with all the elements of \mathbf{A} non-negative, 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 non-negative, the constraints of problem (4.19) 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.*

This theorem also shows that there is no loss of generality in the assumption, in theorem 4.1.1, that the common oscillation of the mixed signals is equal to the common oscillation of the sources, if the sources' amplitudes are positive. Recall that we proved that, for the case where the number of sources is equal to the number of mixed signals, if the sources were given by

$$s_k(t) \equiv A_k(t) e^{i(\phi(t) + \phi_k)}, \quad (4.25)$$

then, under the assumptions of theorem 4.1.1, the only way to construct a linear combination $\mathbf{y} = \mathbf{M}\mathbf{s}$ such that

$$y_j(t) = C_j(t)e^{i(\phi(t)+\alpha_j)} \quad (4.26)$$

was to have $y_i = Ks_j$ for some i and j , where K is a non-zero real number. At the time, we noted that (4.26) enforces a very particular form for the mixtures \mathbf{y} , since their common oscillation, $\phi(t)$, must be equal to that of the sources \mathbf{s} .

Using Theorem 4.3.3, it is straightforward to show that this must be the case, and therefore that no generality was lost in 4.1.1. One merely needs to decompose the sources according to 4.18:

$$\mathbf{S} = \mathbf{D}_{\mathbf{z}_s} \mathbf{A}_s \mathbf{D}_{\mathbf{f}_s}, \quad (4.27)$$

and similarly decompose the data as

$$\mathbf{Y} = \mathbf{D}_{\mathbf{z}_y} \mathbf{A}_y \mathbf{D}_{\mathbf{f}_y}. \quad (4.28)$$

Note that Theorem 4.1.1 assumes that the mixtures \mathbf{y} have a PLF of 1, therefore the factorization (4.28) must exist.

We then use the equality $\mathbf{Y} = \mathbf{M}\mathbf{S}$ and plug in equation (4.27) to obtain

$$\mathbf{Y} = \mathbf{M} \mathbf{D}_{\mathbf{z}_s} \mathbf{A}_s \mathbf{D}_{\mathbf{f}_s}. \quad (4.29)$$

Finally, we use the fact that the left-hand sides of equations (4.28) and (4.29) are the same, to obtain

$$\mathbf{D}_{\mathbf{z}_y} \mathbf{A}_y \mathbf{D}_{\mathbf{f}_y} = \mathbf{M} \mathbf{D}_{\mathbf{z}_s} \mathbf{A}_s \mathbf{D}_{\mathbf{f}_s}. \quad (4.30)$$

One can now use theorem 4.3.3 for the two factorizations in (4.30) to show that the common oscillation for the linear mixtures, $\mathbf{D}_{\mathbf{f}_y}$, must be the same as that of the sources, $\mathbf{D}_{\mathbf{f}_s}$. This implies that no generality is lost in assuming that any linear combinations of the sources that have a PLF of 1 must be of the form (4.26).

4.3.5 Optimization Strategy

The PLMF algorithm is presented in Table 4.2. We now explain in further detail how each of the two subproblems is tackled. We employ the BNGS method in both optimizations; in the first subproblem, we randomly initialize the variables $\hat{\mathbf{H}}$, $\hat{\mathbf{A}}$ and $\hat{\mathbf{f}}$, and iteratively optimize relative to each of them, while keeping all others fixed (lines 4-8 of Table 4.2). Similarly, for the second subproblem, we randomly initialize $\hat{\mathbf{M}}$, $\hat{\mathbf{A}}$ and $\hat{\mathbf{z}}$ and optimize each of them while keeping all others fixed (lines 15-19 of Table 4.2). The use of BNGS has a great advantage: problems (4.20) and (4.24), which are hard to directly solve, in particular due to the presence of products of variables, are solved through an iteration of constrained least-squares problems, which we optimize in a simple, albeit suboptimal, procedure. There is a downside: BNGS is not guaranteed to converge. We discuss this aspect further in section 6.2.

The two subproblems (4.20) and (4.24) are convex in some variables and non-convex in other variables. Instead of trying to find the global minimum for a certain variable 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 three reasons: simplicity, because like this all variables are optimized in a similar way; speed, which allowed us to run the extensive experiments shown in section 5.2; and the quality of the results in those experiments. Note that, while this is a sub-optimal procedure, the fact that the two subproblems are non-convex in some variables would prevent us from having a guaranteed optimal solution.

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 5) is done without taking the first constraint of (4.20) into account. After the unconstrained solution is found, \mathbf{H} is multiplied by a scalar so that the largest absolute value of its elements becomes exactly 1. All variables, in both subproblems, are handled in a similar manner.

We use the values of the cost functions of problems (4.20) and (4.24) 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 10 and 21 of table 4.2), to partially cope with the possible existence of non-absolute minima.

4.4 The Effect of Whitening in SSS

Theorem 4.1.1 can be considered the SSS version of Theorem 2.3.1, which establishes identifiability conditions for ICA. In this section we present an SSS version of Theorem 2.3.2, showing that prewhitening yields some advantages when performing SSS. This result was originally presented in [P9].

Let $\text{Cov}[\mathbf{y}\mathbf{y}^T]$ denote the covariance matrix of the mixtures. Prewhitening [38] involves multiplying the data \mathbf{Y} on the left by the matrix

$$\mathbf{B} \equiv \mathbf{D}^{-\frac{1}{2}} \mathbf{V}^H, \quad (4.31)$$

where \mathbf{D} is a $N \times N$ diagonal matrix containing only the nonzero eigenvalues of $\text{Cov}[\mathbf{y}\mathbf{y}^T]$ 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. Then, the equation $\mathbf{B}\mathbf{Y} = \mathbf{B}\mathbf{M}\mathbf{S}$ defines a new BSS problem, with new data $\mathbf{B}\mathbf{Y}$ which now has N rows.⁷ The new mixing matrix $\mathbf{B}\mathbf{M}$ is called the *equivalent mixing matrix*, and is now square.

⁷ In the presence of additive noise, all eigenvalues of $\text{Cov}[\mathbf{y}\mathbf{y}^T]$ will be nonzero. Even in that case, this reasoning remains valid if noise levels are low and only the N largest eigenvalues are used to construct \mathbf{D} .

Phase Locked Matrix Factorization	
1:	Given: data \mathbf{Y} , MaxRuns_f , MaxIter_f , $\text{MaxRuns}_{\mathbf{M}, \mathbf{A}, \mathbf{z}}$, $\text{MaxIter}_{\mathbf{M}, \mathbf{A}, \mathbf{z}}$
I:	ESTIMATION OF \mathbf{f}
2:	for $\text{run} \in \{1, 2, \dots, \text{MaxRuns}_f\}$, do
3:	Randomly initialize $\hat{\mathbf{H}}, \hat{\mathbf{A}}, \hat{\mathbf{f}}$
4:	for $\text{iter} \in \{1, 2, \dots, \text{MaxIter}_f\}$, do
5:	Solve minimization (4.20) for \mathbf{H}
6:	Solve minimization (4.20) for \mathbf{A}
7:	Solve minimization (4.20) for \mathbf{f}
8:	end for
9:	end for
10:	From the MaxRuns_f solutions, choose the one which yields the lowest value of the function being minimized in (4.20)
11:	Store \mathbf{f} and discard \mathbf{H} and \mathbf{A}
12:	Correct sign of \mathbf{f} by detecting values of (4.21) greater than 1
II:	ESTIMATION OF $\mathbf{M}, \mathbf{A}, \mathbf{z}$
13:	for $\text{run} \in \{1, 2, \dots, \text{MaxRuns}_{\mathbf{M}, \mathbf{A}, \mathbf{z}}\}$, do
14:	Randomly initialize $\hat{\mathbf{M}}, \hat{\mathbf{A}}, \hat{\mathbf{z}}$
15:	for $\text{iter} \in \{1, 2, \dots, \text{MaxIter}_{\mathbf{M}, \mathbf{A}, \mathbf{z}}\}$, do
16:	Solve problem (4.24) for \mathbf{M}
17:	Solve problem (4.24) for \mathbf{A}
18:	Solve problem (4.24) for \mathbf{z}
19:	end for
20:	end for
21:	From the $\text{MaxRuns}_{\mathbf{M}, \mathbf{A}, \mathbf{z}}$ solutions, choose the one which yields the lowest value of the function being minimized in eq. (4.24)
22:	return $\mathbf{M}, \mathbf{A}, \mathbf{z}, \mathbf{f}$

Tab. 4.2: The Phase Locked Matrix Factorization algorithm.

In SSS, the mixing matrix \mathbf{M} is real but the data \mathbf{Y} are complex. Therefore, if \mathbf{B} is defined as in (4.31), the equivalent mixing matrix $\mathbf{B}\mathbf{M}$ is, in general, complex. Thus, without whitening, one is searching for a real $P \times N$ mixing matrix (or equivalently, a real $N \times P$ unmixing matrix); with whitening one has to search for a complex $N \times N$ mixing matrix (or a complex $N \times N$ unmixing matrix). We now show how one can transform this into a search for a real $N \times N$ mixing (or unmixing) matrix.

We split the data matrix \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 similar way for the source matrix \mathbf{S} . Since \mathbf{M} is real, the initial complex problem $\mathbf{Y} = \mathbf{M}\mathbf{S}$ can be turned into an equivalent real problem in two different ways:

$$\begin{bmatrix} \mathbf{Y}_R \\ \mathbf{Y}_I \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{S}_R \\ \mathbf{S}_I \end{bmatrix} \quad \text{or} \quad [\mathbf{Y}_R \ \mathbf{Y}_I] = \mathbf{M} [\mathbf{S}_R \ \mathbf{S}_I]. \quad (4.32)$$

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 transformable into a solution for the original problem.

Recall from section 2.3 that the condition number can be used as an indicator of the difficulty of an inverse problem. The following theorem states that, in SSS, this difficulty is bounded above if prewhitening is performed. One can apply the whitening procedure to the left-hand side of either the VS form or the HS form, both of which are real. Both of these methods would yield the same upper bound for the condition number of the equivalent mixing matrix in the theorem that follows. We have empirically found, however, that the condition number of the equivalent mixing matrix is, on average, farther from the upper bound presented ahead (and thus, that the matrix is 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, derived in [P9].

Theorem 4.4.1. *Let $\mathbf{S}_{RI} \equiv [\mathbf{S}_R \ \mathbf{S}_I]$ and $\mathbf{Y}_{RI} \equiv [\mathbf{Y}_R \ \mathbf{Y}_I]$. Let \mathbf{B} be the result of applying the procedure from equation (4.31) to \mathbf{Y}_{RI} . Let $a_j(t) = |s_j(t)|$ and $\phi_j(t) = \text{angle}(s_j(t))$. Furthermore, suppose that the following assumptions hold:*

- \mathbf{M} and \mathbf{S} both have maximum rank.
- There is no additive noise; thus, $\mathbf{Y} = \mathbf{MS}$ holds.
- $a_j(t)$, are i.i.d. realizations of a random variable which we denote by A_j ;
- A_j is independent of A_k for $j \neq k$;
- $\phi_j(t)$, are i.i.d. realizations of a random variable which we denote by Φ_j ;
- A_j is independent of Φ_k for any j and k , including $j = k$;
- All A_j have the same distribution (we denote by A a generic random variable with that distribution);
- s_j and s_k have maximum PLF, i.e., they have a constant phase lag; this implies that there exists $\phi(t)$, independent of j , such that $\phi_j(t) = \phi_j + \phi(t)$ for all j and t ;
- For each value of t , $\phi(t)$ is random, and uniformly distributed in $[0, 2\pi)$; note, however, that $\phi(t)$ does not need to be i.i.d..

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

$$\rho(\mathbf{BM}) \leq \sqrt{1 + N \frac{\mathbb{E}[A]^2}{\text{Var}[A]}}, \quad (4.33)$$

where N is the number of sources, $\mathbb{E}[\cdot]$ is the expected value operator and $\text{Var}[\cdot]$ is the variance operator. Furthermore, this upper bound is tight, meaning that in some cases equation (4.33) holds with equality.

In the ICA case, prewhitening ensures that we can restrict the search to orthogonal unmixing matrices. Equivalently, the equivalent mixing matrix after prewhitening is guaranteed to have a condition number of 1. In SSS, the

condition number of the equivalent mixing matrix can be larger than 1, but it is bounded above by a value which depends on properties of the amplitudes of the sources.

In [P9], we presented experimental results confirming the validity of this bound, by randomly generating mixing matrices and sources which obeyed the assumptions of the theorem, for a few types of sub- and super-gaussian distributions. We then applied prewhitening and computed the condition number of the equivalent mixing matrix, verifying that the upper bound is correct and that it is tight.

The assumptions of theorem 4.4.1 are quite restrictive, and will probably not be obeyed in most practical situations. Nevertheless, we have empirically found that even in such situations, prewhitening improves the quality of the results of SSS, and improves the convergence time of the separation methods. This was shown for PLMF in [P11]. In the results that follow, prewhitening was always applied to the data before any separation algorithm was used.

5. EXPERIMENTAL RESULTS

5.1 IPA Results

Results obtained with various versions of IPA have been presented in four papers [P1, P2, P3, P5]. For conciseness, we present only the two most significant results, and omit some details; the reader can consult the above references for more information.

Comparison of IPA with other BSS techniques on simulated data

In [P3] we tackled the problem of full ISA where the dependency within each subspace was perfect synchrony. Our goal was to show that IPA could successfully be used as the third step (intra-subspace separation). For this, we generated data that obeys the ISA model. We then used TDSEP¹ [91] for the first step (inter-subspace separation) and a simple heuristic for subspace detection. Since perfectly synchronous sources are strongly dependent, we did not expect TDSEP to be appropriate for the third step. Our goal was to study whether applying TDSEP to separate subspaces, and then using IPA to separate within each subspace, was effective.²

¹ TDSEP is an algorithm which separates sources based on the principle that independent sources should have $\mathbb{E}[s_i(t)s_j(t+\tau)] = \mu_\tau \delta_{ij}$, where μ_τ is a real number that depends on the time lag τ , and δ_{ij} is Kronecker's delta. TDSEP constructs multiple time-lagged correlation matrices \mathbf{C}_τ whose (i, j) element is $\mathbb{E}[s_i(t)s_j(t+\tau)]$ and performs joint diagonalization on them to estimate the original sources.

² These results were obtained with a constant λ optimization strategy. The varying λ strategy detailed in section 4.2.2 was only proposed later, in [P5].

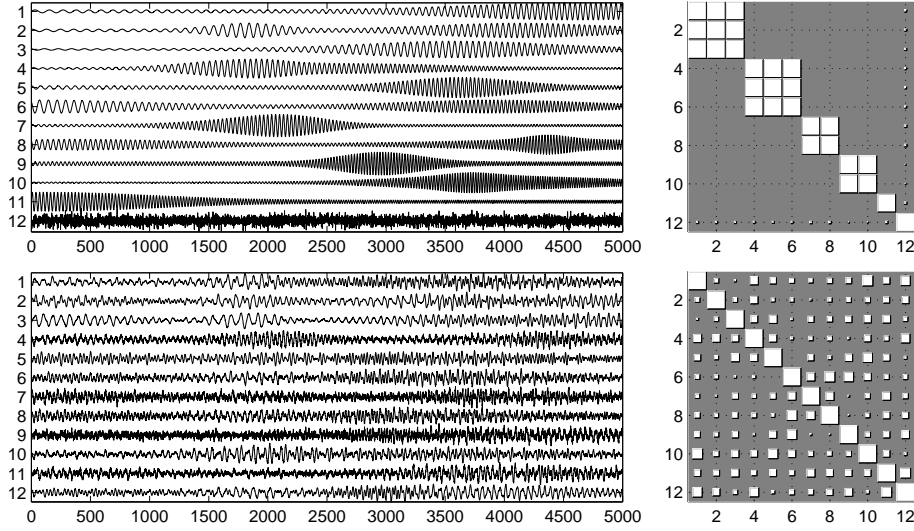


Fig. 5.1: Example dataset with subspaces of dimensions 3,3,2,2,1,1. *First row*: Original sources (*left*) and PLFs between them (*right*). *Second row*: Mixed signals and PLFs between them. In the second column, the numbers denote the indexes of the sources, and the area of each square is proportional to their pairwise PLF.

In order to study this, we randomly generated 300 sets of 12 sources, grouped in 6 subspaces. The subspaces have sizes 3, 3, 2, 2, 1, 1, and the sources were such that different subspaces could not simply be separated through a bandpass filter (see the right panel of figure 5 of the paper). We also generated corresponding mixing matrices randomly. An example of a set of sources generated this way is depicted in Figure 5.1.³

As an illustration of the need for specific techniques for separating synchronous sources, figure 5.2 shows the results of applying FastICA to the set of signals from figure 5.1. The results are quite poor, since the sources are very strongly dependent. This kind of result was consistently obtained throughout our experiments with synchronous sources.

³ The specific way in which the sources and mixing matrices were generated can be found in [P3].

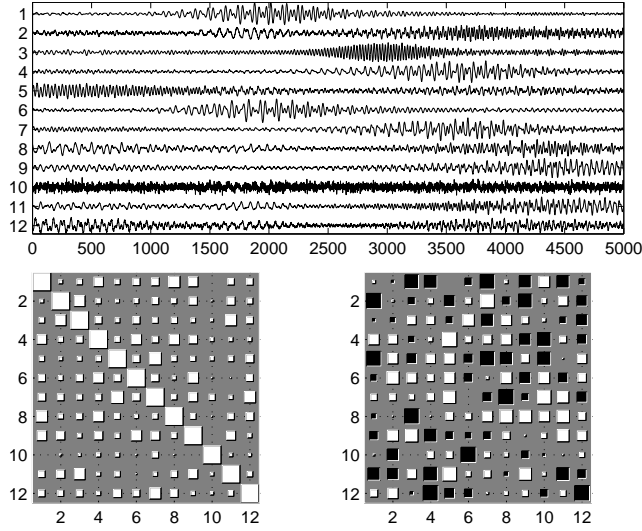


Fig. 5.2: Result of FastICA applied to the dataset of figure 5.1. *Top*: Sources estimated by FastICA. *Bottom left*: PLFs between the estimated sources. *Bottom right*: Estimated gain matrix. It is clear that FastICA is not adequate for the problem.

We applied TDSEP to each of the 300 sets of mixed signals. Then, a simple heuristic procedure was applied to estimate the subspaces present in the data.⁴ If subspaces could be detected, IPA was applied on each of these estimated subspaces. Otherwise, the result of TDSEP was returned with no further processing.

The above procedure, which we denote TDSEP+IPA, was compared with simply applying TDSEP to the sets of mixed signals with no further processing (*i.e.* no subspace detection and no application of IPA to the detected subspaces). To measure the quality of the output, we measured the signal-to-noise ratio (SNR) of the estimated sources relative to the corresponding true sources. We

⁴ We used a hard threshold on the matrix containing the pairwise PLFs. Several values of the threshold are swept; if any of them returns a block-diagonal structure, the subspace structure corresponding to those blocks is considered correct. Otherwise, we consider that the subspaces cannot be detected.

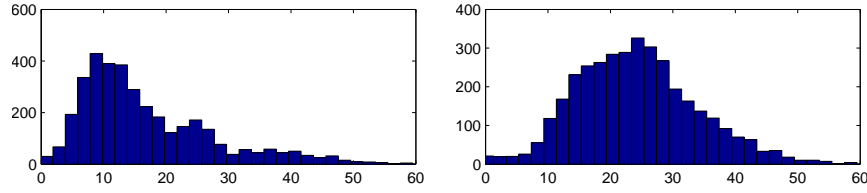


Fig. 5.3: (Left) Histogram of the signal-to-noise ratio (SNR) between the sources found by TDSEP and the original sources. (Right) Similar histogram for the sources found by IPA.

compensate the permutation indeterminacy using a simple heuristic, and then define the SNR of source i as $10 \log_{10} \frac{E[(\alpha \hat{s}_i)^2]}{E[(\alpha \hat{s}_i \pm s_i^2)]}$, where the real scalar α and the \pm sign are chosen to maximize the SNR value. This ensures that these SNR values are independent of permutation, scaling, and sign, as is common in source separation contexts. We then compute the average SNR over all estimated sources. Histograms of the average SNRs of the 300 runs are shown in figure 5.3. The average output SNR is 16.78 dB for TDSEP and 24.18 dB for TDSEP+IPA. These results show that IPA was able to perform an effective separation within the subspaces separated by TDSEP.

The result of these two procedures (TDSEP+IPA and TDSEP only) for the set of sources and mixtures in figure 5.1 is shown in figure 5.4. These two results correspond to the modes of the histograms from figure 5.3, and thus they can be considered typical results. It can be seen that TDSEP is quite successful in separating the subspaces, but that it does not correctly estimate the original sources – this observation is what led us to use TDSEP as a subspace-identification method. It can also be concluded that IPA performs a correct separation within the subspaces.

We also tested the sensitivity of both approaches to additive noise. To that effect, we added random Gaussian white noise to each of the 300 sources with SNRs of 60, 50, 40, 30, and 20 dB (to avoid confusion with the SNR as a quality measure, we use the term “output SNR” for that measure and “input SNR”

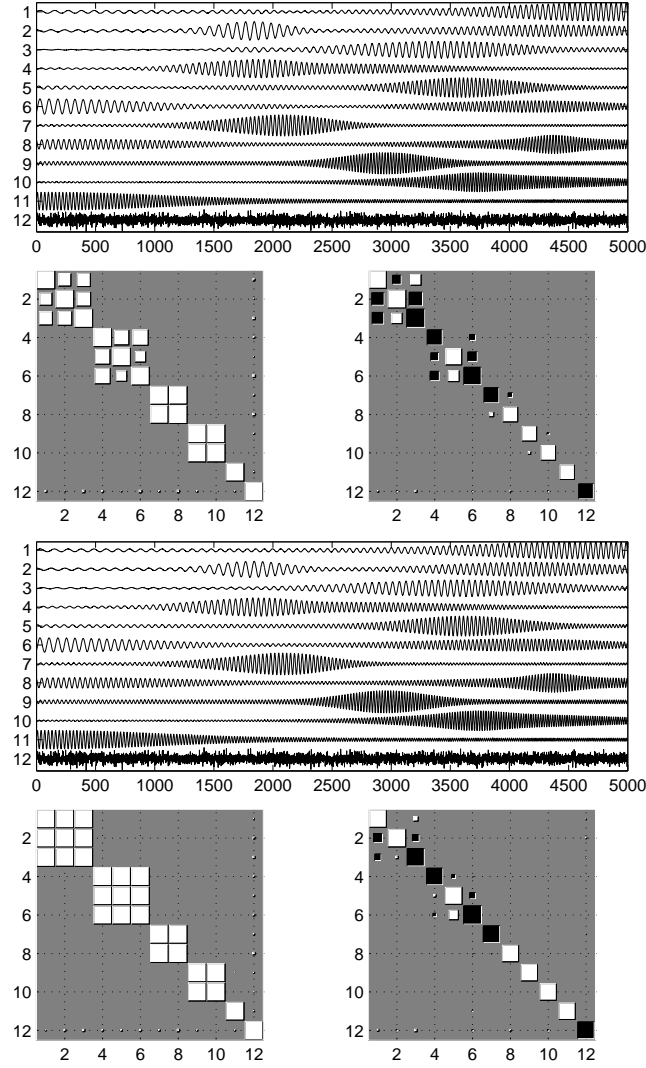


Fig. 5.4: *First row*: Sources resulting from TDSEP. Note that the inter-subspace PLFs (*second row, left*) are very close to zero, but the intra-subspace PLFs are not all close to 1. Furthermore, the intra-subspace separation is poor, as can be seen from inspection of the gain matrix estimated by TDSEP (*second row, right*). *Third row*: Results found after applying IPA to each subspace. The estimated sources are very similar to the original ones. This is corroborated by the PLFs between the estimated sources (*fourth row, left*) and the final gain matrix (*fourth row, right*). The permutation of the sources was manually corrected. White squares represent positive values and black squares represent negative values.

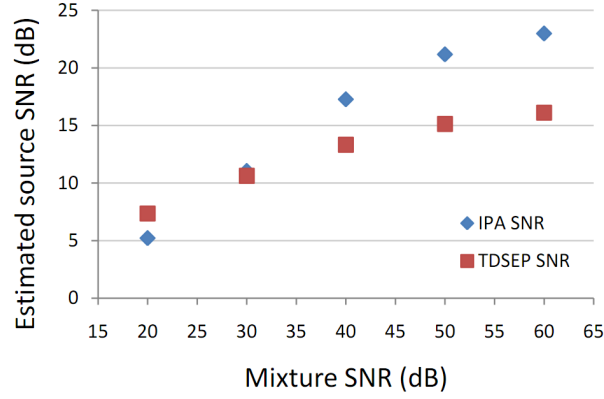


Fig. 5.5: Effect of noise on separation quality, for TDSEP+IPA and for TDSEP alone.

for the measure of the noise added to the mixture). We then repeated the two procedures above for each of these sets of 300 signals. Figure 5.5 shows the average output SNR of the 300 runs, for each input SNR level. It can be seen that both TDSEP+IPA and TDSEP alone yielded similar results, and rather poor ones (output SNR around 11 dB), when the input SNR was 30 dB. For SNR values below 30 dB, TDSEP+IPA actually yields worse results than TDSEP by itself, whereas for an input SNR above 30 dB, TDSEP+IPA performed better than TDSEP alone.

This set of results shows that TDSEP+IPA is capable of separating synchronous sources with significantly better results than other BSS methods in low-noise situations. They also show that even moderate levels of noise hinder TDSEP+IPA's ability to successfully separate this type of sources.

Application of IPA to pseudo-real MEG data

The second set of results illustrates a more realistic test of IPA; it was presented in [P5]. Ideally, one should validate an SSS algorithm on real-life data. For this, one would need a set of real-world signals which were the result

of a mixture of synchronous sources; one would also need direct measurements of those sources, or knowledge of the mixing matrix (or both), to be able to assess the quality of the separation results. For example, for EEG or MEG, we would need not only the EEG/MEG recordings from outside the scalp, but also simultaneous acquisitions of the sources from inside the scalp. We are not aware of any dataset where these acquisitions were simultaneously performed. On the other hand, simulated data, such as the data used in the previous set of results, can only go a certain length in showing the usefulness of an algorithm in real situations.

In an attempt to obtain the best of both worlds, we generated a set of pseudo-real data from actual MEG recordings. By doing this, we were able to generate a set of sources on which we knew the true sources and the true mixing matrix, while still using sources that were of a nature similar to that of the signals one observes in real-world MEG. We begin by summarily describing the process that we used to generate a data set with perfectly synchronous sources. We then explain how we modified these data to analyze non-perfect cases as well.

We began by obtaining a realistic mixing matrix, using the EEGIFT software package and a real-world EEG dataset⁵ it includes to obtain a 64×20 mixing matrix. In each run, we then selected N random rows and N random columns, and formed the $N \times N$ mixing matrix by taking the corresponding submatrix.

The second step involved obtaining a set of physiologically plausible sources which obey the SSS model. For this, we used the MEG dataset previously studied in [87], and selected N sources at random from its 122 channels. These N sources were bandpass filtered using a filter with zero phase. The passband was 18-24 Hz, which is of a width similar to filters used in typical MEG studies [85]. We then computed the Hilbert transform of each of these bandpass-filtered

⁵ This is not a typo. We did use an EEG dataset to obtain the mixing matrix and an MEG dataset to obtain the sources. This was intended only as a proof-of-concept with more realistic data, therefore we do not believe this to be a serious issue.

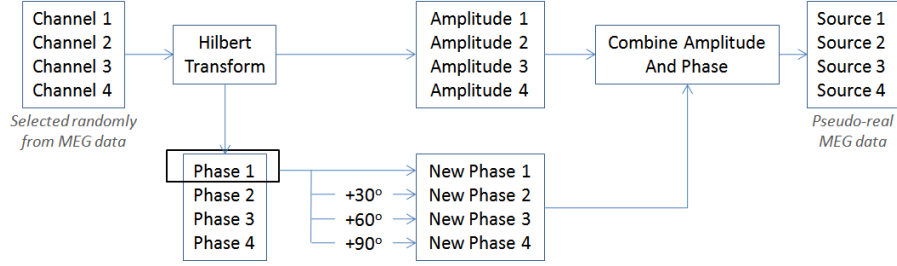


Fig. 5.6: The process used to generate the pseudo-real MEG sources.

sources and extracted their amplitudes and phases.

To generate sources which were phase-locked, we generated new *pseudo-real* sources which used the amplitudes of the original sources, but whose phases were phase-lagged versions of the first source's phase. As an example, suppose that $N = 4$. The first pseudo-real source was equal to the first original channel. We replaced the phase of the second of these channels with the phase of the first channel with a constant phase lag of $\frac{\pi}{6}$ radians. The phase of the third channel was replaced with the phase of the first channel with a constant phase lag of $\frac{\pi}{3}$ radians, and that of the fourth channel with the phase of the first channel with a lag of $\frac{\pi}{2}$ radians. The amplitudes of the four sources were kept as the original amplitudes of the four random channels themselves. The process is illustrated in figure 5.6.

The above procedure yielded sources which exactly obeyed the SSS model. To study situations which deviated from the model, we multiplied each sample t of each source j by $e^{i\delta_j(t)}$, where the phase jitter $\delta_j(t)$ was drawn from a random Gaussian distribution with zero mean and standard deviation σ . We tested IPA for σ from 0 to 20 degrees, in 5 degrees steps. One example with $\sigma = 5$ degrees is shown in figure 5.7, and one with $\sigma = 20$ degrees is shown in figure 5.8. 100 different datasets were generated for each of these phase jitter values, by selecting at random different rows and different columns from the

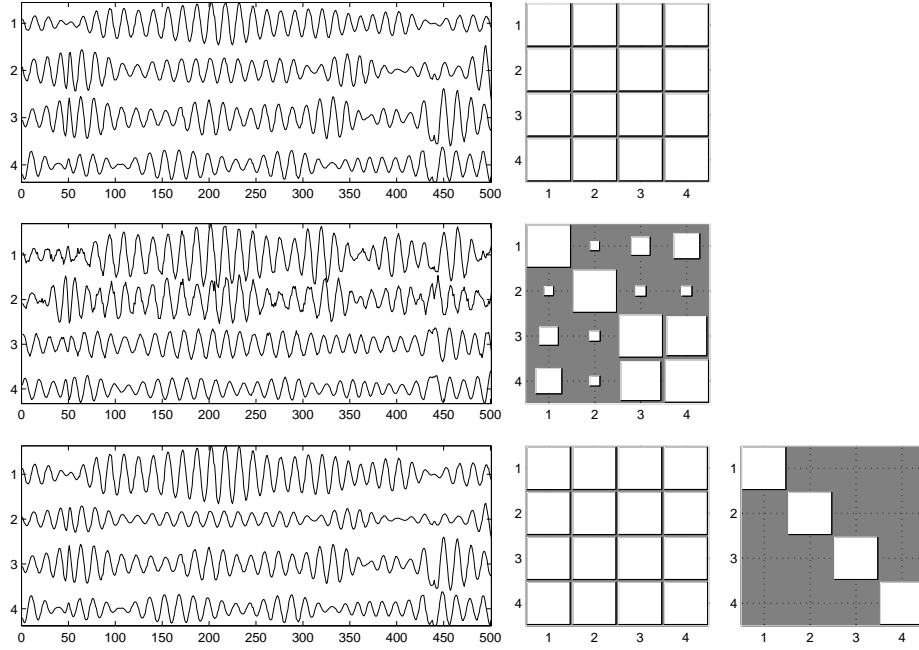


Fig. 5.7: Example of a dataset where $\sigma = 5$ degrees. Only a short segment of the signals is shown, for clarity. *Top row*: original sources (*left*) and PLFs between them (*right*). *Middle row*: mixed signals (*left*) and PLFs between them (*right*). *Bottom row*: estimated sources, after manual compensation of permutation, scaling, and sign (*left*); PLFs between them (*middle*); and the gain matrix $\mathbf{W}^T \mathbf{A}$ (*right*). The gain matrix is virtually equal to the identity matrix, indicating a correct separation.

original 64×20 mixing matrix, and by selecting at random different channels from the original 122 channels of the MEG data.

Figure 5.9 shows the output SNR and Amari Performance Index⁶ as a function of the phase jitter for $N = 4$. It can be seen that for phase jitter values from 0 (jitterless case) to 5 degrees (mild jitter) there was virtually no loss of

⁶ The Amari Performance Index is another quality measure for source separation. It measures how different the gain matrix $\mathbf{W}^T \mathbf{M}$ is from a diagonal matrix. Please see [5] for a formal definition.

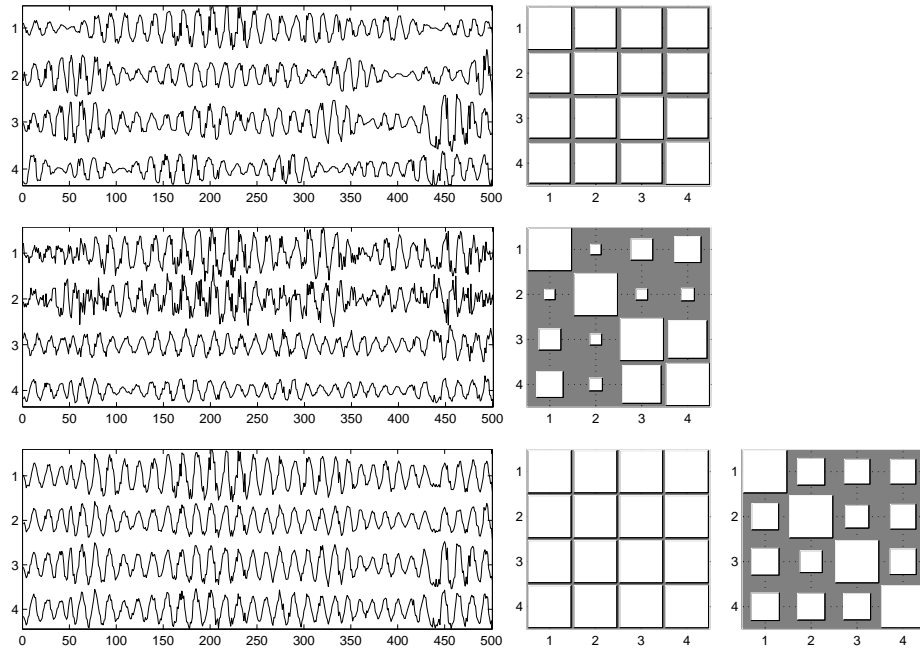


Fig. 5.8: Similar to figure 5.7, but with $\sigma = 20$ degrees. The gain matrix has significant values outside the diagonal, indicating that a complete separation was not achieved. Nevertheless, the largest values are in the diagonal, corresponding to a partial separation.

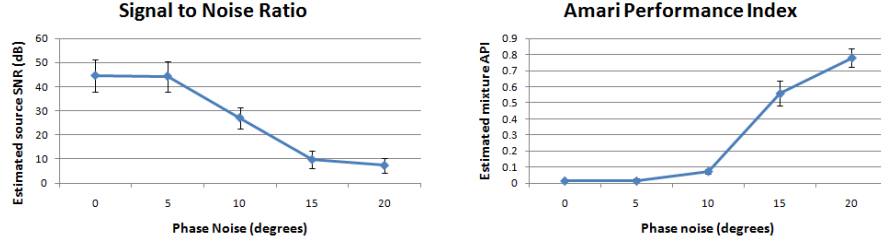


Fig. 5.9: Result of applying IPA to pseudo-real MEG data with $N = 4$, with varying phase jitter: Signal to Noise Ratio (left) and Amari Performance Index (right).

performance. The performance at a jitter of 10 degrees was already deteriorated but still acceptable with an average output SNR of 27 dB. The quality of the separation then gradually deteriorated until 15 degrees (strong jitter), after which performance remained at a low level. We added the noise after the bandpass filtering; we expect that results would be better if the noise had been added before that filtering step.

We also studied the effect of the phase lag between the sources. For this, we used $N = 2$, and generated 100 data sets at random as above, with phase lags of $\frac{\pi}{12}$, $\frac{2\pi}{12}$, $\frac{3\pi}{12}$, $\frac{4\pi}{12}$ (*i.e.*, multiples of 15 degrees). The results are shown in figure 5.10. The results show that phase lags of $\frac{\pi}{6}$ and below yield significant variability in performance, with the error bars including SNR values from around 60 dB to under 20 dB. In contrast, values of $\frac{\pi}{4}$ and above yield consistently good results.

We also studied how results varied with the choice of N , by generating 100 datasets for each of $N = 2, 3, 4, 5$. Phase lags were multiples of $\frac{\pi}{6}$.⁷ The results, shown in figure 5.11 are quite surprising: while performance was consistently good for $N = 3, 4, 5$, with only slight deterioration as N increased, there was a significant decrease in performance for $N = 2$. We do not have a solid explana-

⁷ In other words, for $N = 2$, sources had phase lags of 0 and $\frac{\pi}{6}$. For $N = 3$ they had phase lags of 0, $\frac{\pi}{6}$ and $\frac{\pi}{3}$, and so on.

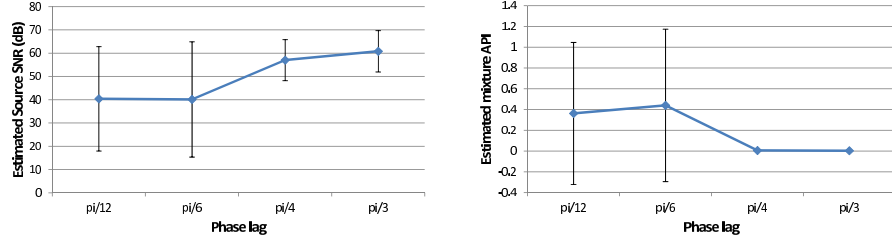


Fig. 5.10: Effect of applying IPA to pseudo-real MEG data with varying phase lags between the sources, with $N = 2$: Signal to Noise Ratio (left) and Amari Performance Index (right).

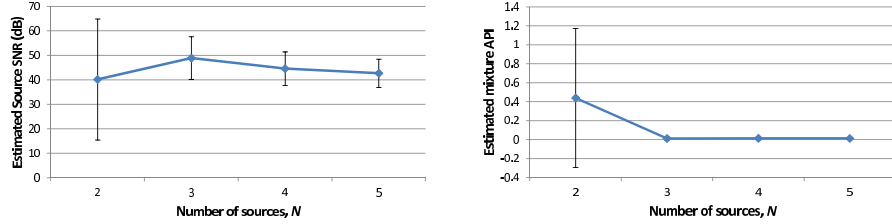


Fig. 5.11: Effect of applying IPA to pseudo-real MEG data with varying values of N : Signal to Noise Ratio (left) and Amari Performance Index (right).

tion for this fact. Our conjecture, which remains open, is that the presence of some pairs of sources with larger phase lags (for example, for $N = 4$, the first and third sources had a phase lag of $\frac{\pi}{3}$ and the first and fourth sources had a phase lag of $\frac{\pi}{2}$) aids in the separation of all the sources, including the ones with small phase lags.

Finally, we compared the varying- λ strategy (which was used for all the above results) with a fixed- λ strategy. Results are shown in table 5.1 for $N = 3$. While the best fixed λ yielded a decent separation quality, with an average output SNR close to 35 dB, using a varying λ yielded much better results, with an average output SNR of almost 50 dB.

Globally, these results illustrate that IPA can handle realistically simulated

	λ	0.025	0.05	0.1	0.2	0.4
SNR	fixed	17.5 ± 21.2	27.5 ± 18.0	34.4 ± 4.3	27.2 ± 3.6	13.5 ± 5.5
	varying	48.9 ± 8.7				
API	fixed	0.795 ± 0.570	0.369 ± 0.465	0.048 ± 0.057	0.079 ± 0.027	0.327 ± 0.097
	varying	0.013 ± 0.015				

Tab. 5.1: Values of Signal to Noise Ratio (SNR) and Amari Performance Index (API) for jitterless data with $N = 3$, for various fixed values of λ , as well as for the varying-lambda strategy detailed in the text. While the best fixed value, $\lambda = 0.1$, yields decent results, the results using a varying value of λ are consistently better, with a large margin.

signals and that it is robust to a variation in the number of sources up to, at least, $N = 5$. They also illustrate that it can handle phase lags of $\frac{\pi}{4}$ and above. Results also show that IPA can handle mild deviations from the true SSS model, by exhibiting some robustness to phase jitter. Even in cases with strong phase jitter, the imperfectly separated sources were normally closer to the true sources than the original mixed signals. Finally, the results show that the varying λ strategy yielded big improvements in separation quality, when compared to any strategy with a fixed λ .

5.2 PLMF Results

Results obtained using PLMF with simulated data were reported in [P6, P7, P8, P11]. PLMF was most extensively studied in [P11]. In that paper, we studied the effect of several variables (number of sources, number of sensors, amount of additive noise, amount of phase jitter, number of time samples, and more) by starting from a “central case” where PLMF yielded good results, and changing one of these variables at a time, to find how the algorithm’s performance varied with each of them. We report the most important experimental results from that paper in what follows.

We used a noisy variant of the source model in expression (4.18) to generate the data. This variant accomodated two deviations from the noiseless case: the

presence of additive noise and of phase jitter. The model used to generate the data was

$$\mathbf{Y} \equiv \mathbf{M}(\mathbf{A} \odot (\mathbf{z}\mathbf{f}^T) \odot \mathbf{J}) + \mathbf{N}, \quad (5.1)$$

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 sensor 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 and jitterless model of Equation (4.18).

We generated 1000 datasets for each set of parameters studied. For each dataset, a mixing matrix \mathbf{M} was randomly generated, with each entry uniformly distributed between -1 and 1, the vector of phase lags \mathbf{z} was generated as $[0, \Delta\phi, \dots, (N-1)\Delta\phi]^T$ ($\Delta\phi$ is determined below), and the common oscillation \mathbf{f} was generated as a sinusoid: $\mathbf{f} = [0, \exp(i\Delta t), \exp(i2\Delta t), \dots, \exp(i(T-1)\Delta t)]^T$, with $T = 100$ and $\Delta t = 0.1$. While this was 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 that paper. We have empirically verified that PLMF worked well with other choices for \mathbf{f} as long as they were smoothly-varying (otherwise, the correction of phase jumps, mentioned at the end of section 4.3.2, became unreliable).

The amplitude \mathbf{A} was generated as the result of lowpass filtering a Gaussian white noise signal. Specifically, we began by generating random Gaussian white noise of length T . We then took the discrete cosine transform (DCT) of that signal, kept only the 10% of coefficients corresponding to the lowest frequencies, and took the inverse DCT of the result. We then added a constant to this filtered signal to ensure that it was non-negative⁸, and the result became $\mathbf{a}_1(t)$, the first row of \mathbf{A} . The process was repeated, with different random initializations, for each of the remaining rows of \mathbf{A} .

One example of a set of signals generated in this manner is depicted in figure

⁸ While the algorithm presented in this work does not require positive amplitudes, we compared it to other algorithms which do require this assumption.

5.12, where we present an extended time period ($T = 500$) to better illustrate the structure of the signals.

The paper [P11] studied the effect of the following variables:

- Additive noise \mathbf{N} , as measured by the signal-to-noise ratio (SNR) of each mixture. The energy of the noise in each channel was selected so that all channels had the same SNR, which is called the input SNR. We studied the cases of an SNR of 80, 60, 40, 20, and 0 dB.
- Phase jitter \mathbf{J} . We studied two types of jitter:
 - The first case was jitter where each entry of \mathbf{J} was of the form $e^{i\delta}$, where δ was independently drawn from a Gaussian distribution with zero mean and standard deviation σ_{iid} . We studied the cases of $\sigma_{iid} = 0, 0.02, 0.04, \dots, 0.1$. We name this *i.i.d. jitter*, since the jitter for time t and for source k was independent from the jitter in any other entry of \mathbf{J} .
 - The second case is called *correlated jitter*. We generate a matrix \mathbf{Q} in a similar manner to the way in which we generated the amplitude \mathbf{A} , except that positivity was not enforced, and that we kept the lowest 2% of coefficients of the DCT, instead of the lowest 10%. This yielded a very slowly varying signal. We then generated the jitter \mathbf{J} as $e^{i\mathbf{Q}}$, where the exponential is taken elementwise. This resulted in a jitter which was slow-varying. While in a statistical sense the sources are uncorrelated, due to the finite observation time T , this jitter is correlated from one source to another. In the context of this 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.
- Phase lag $\Delta\phi$. We studied the cases of $\Delta\phi = \pi/50, 2\pi/50, \dots, 12\pi/50$.

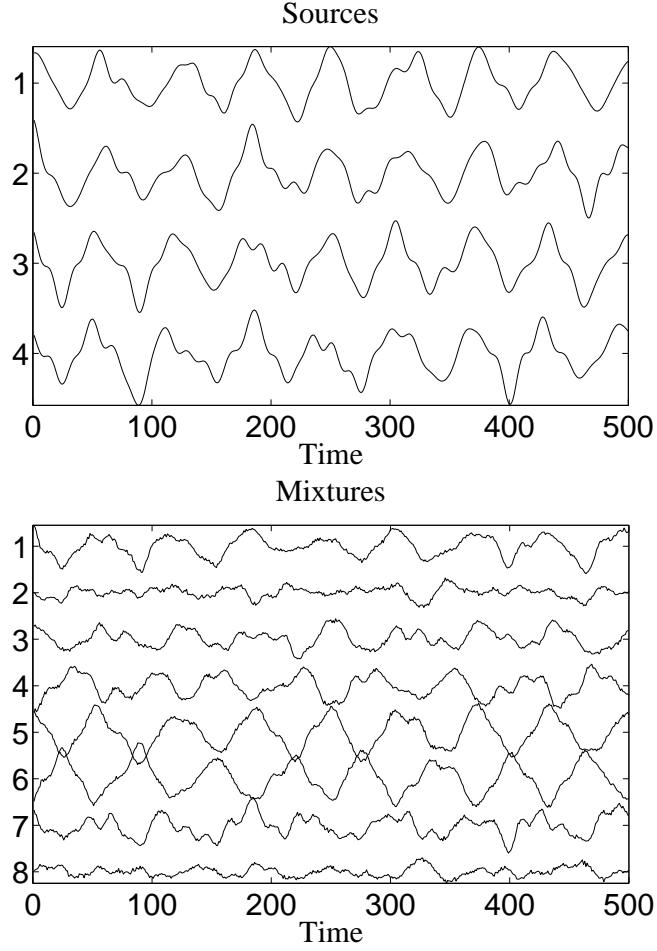


Fig. 5.12: *Top*: the real part of a typical set of four sources generated as described in the text, with no phase jitter. *Bottom*: the real part of a corresponding set of eight mixtures, with an input signal-to-noise ratio (SNR) of 20 dB. The horizontal axis measures time in samples. Note that in most of the following experiments, only 100 points were used.

- Number of sources N and number of sensors P . We studied the cases $N = 2, 4, \dots, 10$, with $P = N$ and with $P = 2N$.
- Number of time samples T . We studied the values $T = 100, 200, 400, 800$.

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

We first applied both IPA and PLMF to 1000 datasets in each of four situations, all of which had $N = P = 2$ 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, $\Delta\phi = \pi/10$). We also compared with the first variant of PLMF [P7], in which all four variables are estimated simultaneously, and with FastICA [38].

The results are shown in figure 5.13. Apart from one situation where both versions of PLMF are tied, these results show a clear superiority of 2-stage PLMF when compared to the other two SSS algorithms. FastICA performed poorly, as expected, given that the sources were strongly inter-dependent.⁹

In other experiments [P11], we showed how PLMF performed when each of the above variables was varied. We concluded that, unlike IPA, the performance of PLMF decreased gracefully with the input SNR (see figure 5.14), since the separation quality was only 2-3 dB below the input SNR, except for

⁹ 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.

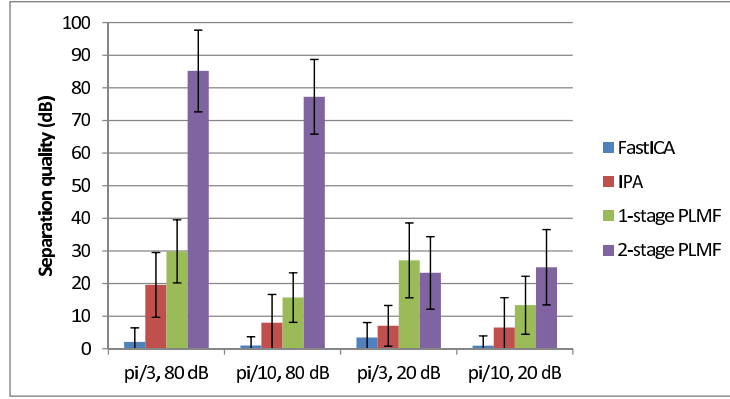


Fig. 5.13: Comparison of FastICA, IPA, one-stage PLMF, and two-stage PLMF. Error bars correspond to plus or minus one standard deviation. The two-stage PLMF algorithm clearly dominates the other two algorithms, except for one situation ($\Delta\phi = \pi/3$, input SNR of 20 dB) where it is essentially tied for first place with one-stage PLMF.

low-noise cases (input SNR of 80 and 60 dB), in which the separation quality was nevertheless very good (65 and 55 dB, respectively).

PLMF can handle very small phase lags. Figure 5.15 shows how the separation quality varied with the phase lag $\Delta\phi$. For most values of this parameter, the separation quality was very high. However, it became progressively lower when $\Delta\phi$ approached zero, where the hypothesis of Theorem 4.3.2 fails to hold. Nevertheless, this deterioration in performance was gradual, and was only relevant for very small phase lags (smaller than $\frac{2\pi}{50}$, or 7.2 degrees, which yielded a separation quality of 23.7 dB).

We also concluded that, in low-noise situations, having as many sensors as sources ($P = N$) was enough to obtain a good separation: little benefit is brought by having $P > N$. However, that benefit became significant in the presence of noise. Figure 5.16 shows the effect of varying the number of sources N and the number of sensors P . Generally, the quality of the results decreased with increasing N , which is expected since the size of the problem variables \mathbf{M} ,

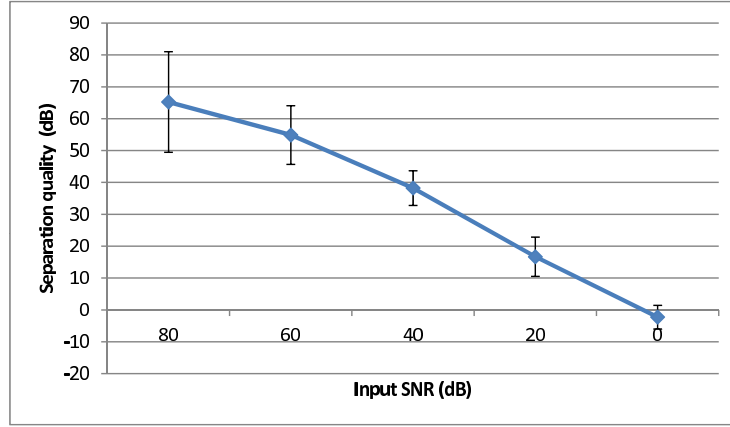


Fig. 5.14: Separation quality versus input SNR. Under heavy noise, PLMF can recover the sources with about as much noise as they had in the input.

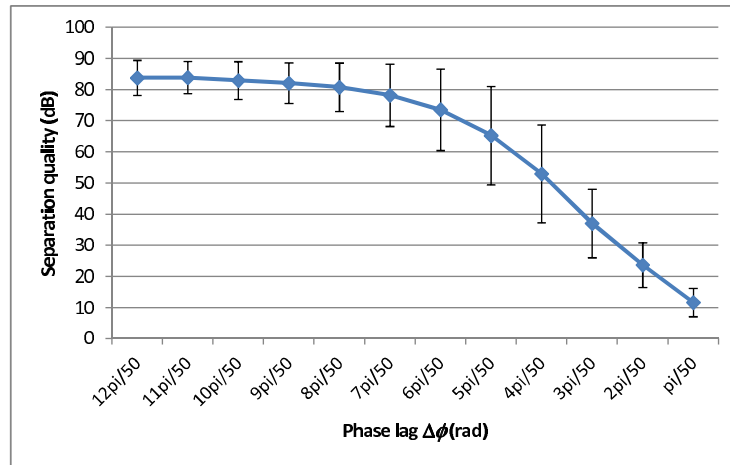


Fig. 5.15: Separation quality versus phase lag. PLMF's results are, in general, good, but they deteriorate progressively as one approaches the case where $\Delta\phi = 0$, where theorem 4.3.2 fails to hold.

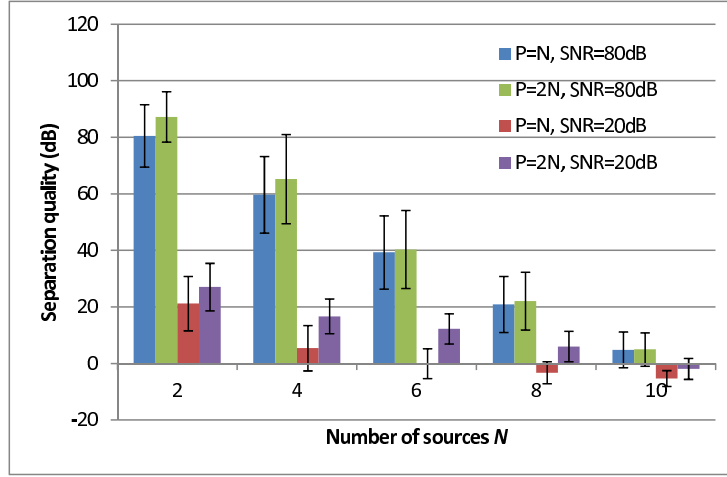


Fig. 5.16: Separation quality versus number of sources (N), number of sensors (P), and input SNR.

\mathbf{A} and \mathbf{z} increases. When there was very little noise (input SNR of 80 dB), there was little benefit in doubling the number of sensors from $P = N$ to $P = 2N$. However, when there was considerable noise (input SNR of 20 dB), that benefit became significant, especially for $P = 4, 6, 8$ where the improvement exceeded 10 dB.

Finally, we observed that PLMF did not handle the presence of phase jitter well. Even small amounts of jitter brought the performance from around 65 dB to 30 dB, and larger amounts lowered it further. The paper presents a more involved discussion explaining that the effect of i.i.d. jitter can probably be mitigated using lowpass filtering of \mathbf{f} as post-processing, after \mathbf{f} is estimated, whereas correlated jitter cannot be mitigated that way.

6. FUTURE WORK

In this chapter we present a few directions for future work based on the research presented above. Some of these directions were already studied in a limited depth, and we present suggestions to deepen them.

6.1 *Tests on Real-Life Data*

We begin this section by briefly discussing an important aspect which was not dealt with in this thesis: validation using real-world data. Indeed, perhaps the most important future development would be the acquisition of non-simulated data, and the testing of the algorithms proposed here on those data. The study with pseudo-real data, presented in [P5], can be considered a step in this direction; however, those pseudo-real data were generated in such a way that they exactly followed the SSS model, or only deviated from it in controlled ways. Real-life data may deviate from the model in other ways, and only tests on those data can tell how the proposed algorithms perform in such situations.

Acquisition of real-life data in an EEG or MEG context, in a form that allows the assessment of the performance of the proposed algorithms, is not easy, for several reasons:

- Expensive equipment is required.
- Mixture data can be acquired from the outside of the skull, but the sources, which are needed to assess the algorithms' performance, must be acquired from inside the skull. Invasive procedures are, therefore, necessary.

- Neuroscience experts are required, to ensure that data are acquired from areas which are expected to exhibit synchrony, and to help in evaluating the results.

It was not possible to acquire such data in this thesis' work. Nevertheless, it is certainly important to test SSS algorithms in real-world data. A non-invasive, but also less conclusive approach is to collect only data from the outside of the skull, apply SSS to those data, and see whether the resulting sources “make sense”. This eliminates the invasiveness requirement, but makes neuroscientists' input even more important. This approach has been followed in [86], which applied ICA to MEG recordings and then measured coherence between the resulting sources. A similar approach was followed in [55], which measured synchrony instead of coherence. Other works, such as [58], which used BSS with non-synchrony criteria to extract sources, also follow this approach.

Another possibility is to validate these algorithms on another domain where SSS's assumptions are (approximately) valid. One such domain might be music. Musical instruments which are playing the same tone will have the same fundamental frequency, and should therefore have perfect phase synchrony. It is possible that the algorithms presented here can be applied to the separation of instruments playing the same tone, and subspace versions of these algorithms may be used to separate sets of instruments playing different tones. If the music being played is known, this knowledge can be incorporated in a variant of the PLF measure known as $n : m$ synchrony [82]. Finally, the domain of multipath communication systems [26] may find the techniques proposed here, or adapted versions of them, useful.

6.2 Improvements on PLMF

One future direction that was briefly discussed in [P11] consists in turning each of PLMF's subproblems into a sequence of convex problems. More specif-

ically, the goal is to have an equivalent, or near-equivalent, formulation, such that the optimization in each variable is a convex problem. In its present state, the first subproblem in PLMF (problem (4.20)) is not solved through a sequence of convex problems: 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 (problem (4.24)) is also not solved through a sequence of convex problems for a similar reason involving \mathbf{M} and \mathbf{z} .

Let us discuss why one should be interested in doing this. There is considerable theoretical work on BNGS methods. In particular, [28] 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. Turning each of PLMF's subproblems into a sequence of convex problems (or more specifically, ensuring that the feasible set for each variable is a closed, non-empty, convex set) would allow direct application of this theorem, ensuring that if the algorithms for each subproblem converged, we would have found critical points of the subproblems.

6.3 Generalizing PLMF for subspaces

The algorithms presented in this thesis could be adapted to work with subspaces. One possibility would be to incorporate them into a full ISA framework, using a first step which is agnostic to synchrony (*i.e.*, it does not use synchrony information at all) to separate subspaces, followed by a second step which uses synchrony to separate within a subspace. In [P3], we used an approach of this kind, where the synchrony-agnostic first step was TDSEP, and the second step was IPA. The references presented at the end of section 2.4 illustrate that in some situations this is possible using ICA methods as the first step.

However, if it is known that the dependency within each subspace is synchrony, and that sources from different subspaces are not synchronous, it makes

sense to develop algorithms which exploit this information. One possible advantage is that these algorithms may not require full independence between sources in different subspaces.

6.4 Partial Synchrony

While sources with partial synchrony were tackled with the algorithms presented in this work, the model assumes that the sources have full synchrony. Yet another improvement would involve devising a model which does not make this assumption. It is likely that this would lead to techniques quite different from the ones presented here.

One possibility is to consider a Bayesian framework under which the phase of source j , at time t , is no longer given by the product of z_j and $f(t)$, but rather by the product of z_j and $f_j(t)$, where $f_j(t)$ is, for example, a Gaussian random variable centered at $f(t)$, with some variance. As the variance tends to zero we recover the perfectly-synchronous model of SSS.

6.5 Partial ISA

Some work was developed in the study of the ISA separation principle (defined at the end of section 2.4) for a particular type of sources. Since some details of the proof are not finished, it is omitted, and the result is here presented as a conjecture.

The conjecture is the following: by minimizing the sum of the entropies of the individual estimated sources, one can solve Partial ISA, for sources which are a sum of Gaussian components with zero mean. Formally, consider the usual BSS model, $\mathbf{y} = \mathbf{M}\mathbf{s}$, where sets of sources are independent as in (2.14),

repeated here for convenience:

$$\mathbf{s} \equiv \begin{bmatrix} \mathbf{s}^1 \\ \mathbf{s}^2 \\ \vdots \\ \mathbf{s}^K \end{bmatrix}, \text{ where } \mathbf{s}^k \equiv \begin{bmatrix} s_1^k \\ \vdots \\ s_{N_k}^k \end{bmatrix}. \quad (6.1)$$

We consider only sources of a particular type: Gaussian mixture densities where each Gaussian component has zero mean. Concretely, the probability density functions (PDFs) of the subspaces are assumed to be given by

$$p(\mathbf{s}^k) = \sum_i \alpha_{ik} \mathcal{N}(\mathbf{s}^k | 0, \mathbf{A}_{ik}). \quad (6.2)$$

Furthermore, let $\hat{\mathbf{s}} = \mathbf{W}\mathbf{y}$ denote the estimated sources, and assume that the data \mathbf{y} have been pre-whitened. Our conjecture is that any solution of

$$\begin{aligned} \min_{\mathbf{W}} \sum_{i=1}^N H(\hat{\mathbf{s}}_i) \\ \text{s.t. } \mathbf{W}^T \mathbf{W} = \mathbf{I} \end{aligned} \quad (6.3)$$

yields a gain matrix $\mathbf{W}^T \mathbf{M}$ which is a permutation of a block diagonal matrix, with blocks corresponding to each of the subspaces, as exemplified in section 2.4. Consequently, each estimated source will be a linear combination of true sources from a single subspace.

If this conjecture turns out to be true, its applicability is considerable. If each Gaussian component could have an arbitrary mean, sources of the form (6.2) could approximate any density in a broad class of densities arbitrarily well as the number of Gaussian components increases, a result which has been known for over forty years [75]. Future work in this direction would involve determining the veracity of this conjecture, by presenting a fully correct proof or a counter-example. Furthermore, it would be relevant to characterize which densities can be well approximated by densities of the form (6.2), and investigating what is the intersection between those densities and densities which describe sources with perfect synchrony.

7. CONCLUSIONS

In this thesis we have studied the problem of Separation of Synchronous Sources, an instance of blind source separation where the sources exhibit perfect synchrony. Since synchrony is present in many topics in neuroscience, and EEG/MEG signals can be considered mixtures of underlying sources, SSS is a relevant problem for this field. SSS had not been studied before; this thesis presents the first formalization of the problem. Previously, to separate synchronous sources, researchers were forced to use algorithms which make inadequate assumptions about the sources (such as ICA, which assumes independence of the sources).

It was shown that SSS is sufficiently well-defined, like in the case of ICA. Unlike ICA, singular solutions are an issue in SSS, and they should be taken into consideration when designing algorithms. We showed that pre-whitening results in a bound on the condition number of the equivalent mixing matrix, ensuring a relatively well-conditioned numerical problem. We presented two algorithms to solve SSS problems: IPA, which penalizes singular solutions through regularization, and PLMF, which factorizes the sources using a matrix factorization model which automatically avoids singular solutions.

Experimental tests with simulated data showed that both approaches yield very significant improvements in separation quality compared to ICA algorithms. Within the SSS algorithms, PLMF yielded significantly better results than IPA in experimental comparisons. PLMF also possesses better theoretical properties.

The contributions in this thesis allow, for the first time, the separation of synchronous sources within a grounded theoretical framework, with appropriate algorithms which yield good separation quality and avoid singular solutions, and in the case of PLMF, with identifiability guarantees. Some directions for further work were presented, which illustrate that the topic is far from being solved and present interesting research topics for the future.

BIBLIOGRAPHY

- [1] T. Adali, M. Anderson, and G.-S. Fu. Diversity in independent component and vector analyses. *IEEE Signal Processing Magazine*, pages 18–33, 2014.
- [2] M. Aigner. A characterization of the Bell numbers. *Discrete Mathematics*, 205:207–210, 1999.
- [3] E. Alhoniemi, A. Honkela, K. Lagus, and J. Seppä. Compact modeling of data using independent variable group analysis. *IEEE Transactions on Neural Networks*, 18:1762–1776, 2007.
- [4] L. B. Almeida. Synthesis lectures on signal processing. In *Nonlinear Source Separation*. Morgan & Claypool, 2006.
- [5] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In *Advances in NIPS*, volume 8, pages 757–763, 1996.
- [6] B. B. Andersen, L. Korbo, and B. Pakkenberg. A quantitative study of the human cerebellum with unbiased stereological techniques. *Journal of Computational Neurology*, 326:549–560, 1992.
- [7] B. Ans, J. Herault, and C. Jutten. Adaptive neural architectures: detection of primitives. In *Proceedings of COGNITIVA '85*, 1985.
- [8] J. Beirlant, E. Dudewicz, L. Györfi, and E. van der Meulen. Nonparametric entropy estimation: an overview. *International Journal of Mathematical and Statistical Sciences*, 6:17–39, 1997.

-
- [9] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.
 - [10] A. J. Bell and T. J. Sejnowski. A non-linear information maximization algorithm that performs blind separation. In *Advances in Neural Information Processing Systems*, 1995.
 - [11] A. J. Bell and T. J. Sejnowski. Edges are the” independent components” of natural scenes. In *Advances in Neural Information Processing Systems*, pages 831–837, 1996.
 - [12] J. Bioucas-Dias and G. Valadão. Phase unwrapping via graph cuts. *IEEE Transactions on Image Processing*, 16:698–709, 2007.
 - [13] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
 - [14] H. O. Cabral, M. Vinck, C. Fouquet, C. M. A. Pennartz, L. Rondi-Reig, and F. P. Battaglia. Oscillatory dynamics and place field maps reflect sequence and place memory processing in hippocampal ensembles under nmda receptor control. *Neuron*, 81:402–415, 2014.
 - [15] J.-F. Cardoso. Multidimensional independent component analysis. In *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP ’98)*, 1998.
 - [16] A. Cichocki and S. Amari. *Adaptive blind signal and image processing - Learning algorithms and applications*. John Wiley & Sons, 2002.
 - [17] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.
 - [18] P. Comon and C. Jutten. *Handbook of Blind Source Separation*. Academic Press, 2010.

-
- [19] P. Comon and C. Jutten, editors. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press, 2010.
 - [20] B. A. Conway, D. M. Halliday, S. F. Farmer, U. Shahani, P. Maas, A. I. Weir, and J. R. Rosenberg. Synchronization between motor cortex and spinal motoneuronal pool during the performance of a maintained motor task in man. *Journal of Physiology*, 489:917–924, 1995.
 - [21] L. De Lathauwer, B. De Moor, and J. Vandewalle. Fetal electrocardiogram extraction by source subspace separation. In *Proc. IEEE SP/ATHOS Workshop HOS*, 1995.
 - [22] S. M. Doesburg, A. B. Roggeveen, K. Kitajo, and L. M. Ward. Large-scale gamma-band phase synchronization and selective attention. *Cerebral Cortex*, 18:386–396, 2008.
 - [23] A. K. Engel, P. Fries, and W. Singer. Dynamic predictions: oscillations and synchrony in top-down processing. *Nature Reviews Neuroscience*, 2:704–716, 2001.
 - [24] P. Fries. A mechanism for cognitive dynamics: neuronal communication through neuronal coherence. *Trends in Cognitive Sciences*, 9:474–480, 2005.
 - [25] P. Fries. Neuronal gamma-band synchronization as a fundamental process in cortical computation. *Annual Review of Neuroscience*, 32:209–224, 2009.
 - [26] A. Goldsmith. *Wireless Communicatnions*. Cambri, 2005.
 - [27] C. Gray, P. König, A. Engel, and W. Singer. Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, 338:334–337, 1989.
 - [28] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Operations Research Letters*, 26:127–136, 2000.

-
- [29] H. W. Gutch, J. Krumsiek, and F. J. Theis. An ISA algorithm with unknown group sizes identifies meaningful clusters in metabolomics data. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2011.
- [30] A. Guyton and J. Hall. *Textbook of Medical Physiology*. Saunders, 12th edition, 2011.
- [31] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa. Magnetoencephalography theory, instrumentation, and applications to noninvasive studies of the working human brain. *Rev. Mod. Phys.*, 65:413–497, 1993.
- [32] J. Herault and B. Ans. Circuits neuronaux à synapses modifiables: décodage de messages composites par apprentissage non supervisé. In *Comptes rendus de l'Academie des sciences. Serie III, Sciences de la vie*, volume 299, pages 525–528, 1983.
- [33] J. Herault and C. Jutten. Space or time adaptive signal processing by neural network models. *Neural networks for computing*, 151:206–211, 1986.
- [34] J. Herault, C. Jutten, and B. Ans. Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé. In *10^e Colloque sur le traitement du signal et des images*, pages 1017–1022, 1985.
- [35] S. Herculano-Houzel. The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost. *Proceedings of the National Academy of Sciences*, 109:10661–10668, 2012.
- [36] P. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

-
- [37] A. Hyvärinen and P. Hoyer. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12:1705–1720, 2000.
 - [38] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.
 - [39] A. Hyvärinen and U. Köster. FastISA: A fast fixed-point algorithm for independent subspace analysis. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, 2006.
 - [40] K. M. Igarashi, L. Lu, L. L. Colgin, M.-B. Moser, and E. I. Moser. Coordination of entorhinal-hippocampal ensemble activity during associative learning. *Nature*, 2014.
 - [41] C. Jutten and J. Herault. Blind separation of sources: an adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.
 - [42] C. Jutten and J. Karhunen. Advances in nonlinear blind source separation. In *Proc. of the 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003)*, 2003.
 - [43] C. Jutten and A. Taleb. Source separation: From dusk till dawn. In *Proceedings of the 2nd International Workshop on Independent Component Analysis and Blind Source Separation*, pages 15–26, 2000.
 - [44] H. Kameoka, N. Ono, K. Kashino, and S. Sagayama. Complex NMF: a new sparse representation for acoustic signals. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009.
 - [45] B. J. King and L. Atlas. Single-channel source separation using complex

-
- matrix factorization. *IEEE Transactions on Audio, Speech, and Language Processing*, 19:2591–2597, 2011.
- [46] Y. Kuramoto. *Chemical Oscillations, Waves and Turbulences*. Springer Berlin, 1984.
- [47] E. Kuruoglu and F. Theis. Dependent component analysis. *EURASIP Journal on Advances in Signal Processing*, 2013:185, 2013.
- [48] J.-P. Lachaux, E. Rodriguez, J. Martinerie, and F. J. Varela. Measuring phase synchrony in brain signals. *Human Brain Mapping*, 8:194–208, 1999.
- [49] D. Lahat, J. Cardoso, and H. Messer. Second-order multidimensional ICA: Performance analysis. *IEEE Transactions on Signal Processing*, 60:4598–4610, 2012.
- [50] D. Lahat, J.-F. Cardoso, M. Le Jeune, and H. Messer. Multidimensional ICA and its performance analysis applied to CMB observations. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pages 3724–3727, 2011.
- [51] D. Lahat, J.-F. Cardoso, and H. Messer. Second-order multidimensional ICA: Performance analysis. *IEEE Transactions on Signal Processing*, 60(9):4598–4610, 2012.
- [52] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13, pages 556–562, 2001.
- [53] W. Ma, J. Bioucas-Dias, T. Chan, N. Gillis, P. Gader, A. Plaza, A. Ambikapathi, and C. Chi. A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. *IEEE Signal Processing Magazine*, 31:67–81, 2014.

-
- [54] S. Makeig, A. J. Bell, T. P. Jung, and T. J. Sejnowski. Independent component analysis of electroencephalographic data. In *Advances in Neural Information Processing Systems*, pages 145–151, 1996.
 - [55] C. Meinecke, A. Ziehe, J. Kurths, and K.-R. Müller. Measuring phase synchronization of superimposed signals. *Physical Review Letters*, 94, Mar 2005.
 - [56] M. J. Frank M. Laubach N. S. Narayanan, J. F. Cavanagh. Common medial frontal mechanisms of adaptive control in humans and rodents. *Nature Neuroscience*, 16:1888–1895, 2013.
 - [57] G. Nolte, O. Bai, L. Wheaton, Z. Mari, S. Vorbach, and M. Hallett. Identifying true brain interaction from EEG data using the imaginary part of coherency. *Clin. Neuroph.*, 115:2292–2307, 2004.
 - [58] G. Nolte, A. Ziehe, F. Meinecke, and K.-R. Müller. Analyzing coupled brain sources: Distinguishing true from spurious interaction. In *Advances in Neural Information Processing Systems (NIPS)*, 18, pages 1027–1034, 2006.
 - [59] P. L. Nunez and R. Srinivasan. *Electric Fields of the Brain: the neurophysics of EEG*. Oxford University Press, 2006.
 - [60] 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.
 - [61] A. V. Oppenheim, A. S. Willsky, and S. Hamid. *Signals and Systems*. Signal Processing Series. Prentice-Hall, 1996.

-
- [62] J. A. Palmer and S. Makeig. Blind separation of dependent sources and subspaces by minimum mutual information. Technical report, University of California, San Diego, 2010.
- [63] J. A. Palmer and S. Makeig. Contrast functions for independent subspace analysis. In *Latent Variable Analysis and Signal Separation*, pages 115–122, 2012.
- [64] B. Póczos and A. Lörincz. Independent subspace analysis using geodesic spanning trees. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 673–680, 2005.
- [65] M. S. Pedersen, J. Larsen, U. Kjems, and L. C. Parra. *Springer Handbook of Speech Processing*, chapter A Survey of Convolutional Blind Source Separation methods, pages 1065–1084. Springer Press, 2008.
- [66] A. Pikovsky, M. Rosenblum, and J. Kurths. *Synchronization: A universal concept in nonlinear sciences*. Cambridge Nonlinear Science Series. Cambridge University Press, 2001.
- [67] E. Pomarol-Clotet, R. Salvador, S. Sarró, J. Gomar, F. Vila, Á Martínez, A. Guerrero, J. Ortiz-Gil, B. Sans-Sansa, A. Capdevila, J. M. Cebamano, and P. J. McKenna. Failure to deactivate in the prefrontal cortex in schizophrenia: dysfunction of the default mode network? *Psychological Medicine*, 38:1185–1194, 2008.
- [68] B. Roach and D. Mathalon. Event-related eeg time-frequency analysis: An overview of measures and an analysis of early gamma band phase locking in schizophrenia. *Schizophrenia Bulletin*, 34:907–926, 2008.
- [69] J.-H. Schleimer and R. Vigário. Reference-based extraction of phase synchronous components. In *Proceedings of the International Conference on Artificial Neural Networks*, 2006.

-
- [70] J.-H. Schleimer and R. Vigário. Clustering limit cycle oscillators by spectral analysis of the synchronisation matrix with an additional phase sensitive rotation. In *Proc. of the Int. Conf. on Artificial Neural Networks*, 2007.
 - [71] J.-H. Schleimer and R. Vigário. Order in complex systems of nonlinear oscillators: Phase locked subspaces. In *Proc. of the Eur. Symp. on Neural Networks*, 2007.
 - [72] D. L. Schomer and F. L. da Silva. *Niedermeyer's Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Wolters Kluwer Health, 2012.
 - [73] A. Sharma and K. K. Paliwal. Subspace independent component analysis using vector kurtosis. *Pattern Recognition*, 39:2227–2232, 2006.
 - [74] W. Singer. Neuronal synchrony: A versatile code for the definition of relations? *Neuron*, 24:49–65, Sep 1999.
 - [75] H. W. Sorenson and D. L. Alspach. Recursive bayesian estimation using gaussian sums. *Automatica*, 7:465–479, 1971.
 - [76] K. Spencer, M. Niznikiewicz, M. Shenton, and R. McCarley. Sensory-evoked gamma oscillations in chronic schizophrenia. *Biological Psychiatry*, 63:744–747, 2008.
 - [77] S. Strogatz. From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D*, 143:1–20, 2000.
 - [78] Z. Szabó, B. Póczos, and A. Lorincz. Undercomplete blind subspace deconvolution. *Journal of Machine Learning Research*, 8:1063–1095, 2007.
 - [79] Z. Szabó, B. Póczos, and A. Lorincz. Separation theorem for independent subspace analysis and its consequences. *Pattern Recognition*, 45:1782–1791, 2012.

-
- [80] C. Tallon-Baudry, O. Bertrand, C. Delpuech, and J. Pernier. Oscillatory γ -band (30-70 hz) activity induced by a visual search task in humans. *The Journal of Neuroscience*, 17:722–734, 1997.
- [81] C. Tallon-Baudry, O. Bertrand, C. Delpuech, and J. Pernier. Stimulus specificity of phase-locked and non-phase-locked 40 hz visual responses in human. *The Journal of Neuroscience*, 16:4240–4249, 1996.
- [82] P. Tass, M. G. Rosenblum, J. Weule, J. Kurths, A. Pikovsky, J. Volkmann, A. Schnitzler, and H.-J. Freund. Detection of n:m phase locking from noisy data: Application to magnetoencephalography. *Physical Review Letters*, 81:3291–3294, 1998.
- [83] F. J. Theis. Towards a general independent subspace analysis. In *Advances in Neural Information Processing Systems*, pages 1361–1368, 2007.
- [84] P. J. Uhlhaas and W. Singer. Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology. *Neuron*, 52:155–168, Oct 2006.
- [85] F. Varela, J.-P. Lachaux, E. Rodriguez, and J. Martinerie. The brainweb: phase synchronization and large-scale integration. *Nature Reviews Neuroscience*, 2:229–239, 2001.
- [86] R. Vigário and O. Jensen. Identifying cortical sources of corticomuscle coherence during bimanual muscle contraction by temporal decorrelation. In *Proceedings of IEEE International Symposium on Signal Processing and Its Applications*, 2003.
- [87] R. Vigário, V. Jousmäki, M. Hämäläinen, R. Hari, and E. Oja. Independent component analysis for identification of artifacts in magnetoencephalographic recordings. In *Advances in NIPS*, 1997.

-
- [88] S. Whitfield-Gabrieli, H. W. Thermenos, S. Milanovic, M. T. Tsuang, S. V. Faraone, R. W. McCarley, M. E. Shenton, A. I. Green, A. Nieto-Castanon, P. LaViolette, J. Wojcik, J. D. E. Gabrieli, and L. J. Seidman. Hyperactivity and hyperconnectivity of the default network in schizophrenia and in first-degree relatives of persons with schizophrenia. *Proceedings of the National Academy of Sciences*, 4:1279–1284, 2009.
- [89] R. W. Williams and K. Herrup. The control of the neuron number. *Annual Review of Neuroscience*, 11:423–453, 1988.
- [90] T. Womelsdorf, J.-M. Schoffelen, R. Oostenveld, W. Singer, R. Desimone, A. K. Engel, and P. Fries. Modulation of neuronal interactions through neuronal synchronization. *Science*, 316:1609–1612, 2007.
- [91] A. Ziehe and K.-R. Müller. TDSEP - an efficient algorithm for blind separation using time structure. In *International Conference on Artificial Neural Networks*, pages 675–680, 1998.

Source Separation of Phase-Locked Subspaces

Miguel Almeida^{*} and Ricardo Vigário

Adaptive Informatics Research Centre
Department of Information and Computer Science
Helsinki University of Technology, Finland
{miguel.almeida,ricardo.vigario}@hut.fi
<http://www.hut.fi/en>

Abstract. ICA can be interpreted as the minimization of a disorder parameter, such as the sum of the mutual informations between the estimated sources. Following this interpretation, we present a disorder parameter minimization algorithm for the separation of sources organized in subspaces when the phase synchrony within a subspace is high and the phase synchrony across subspaces is low. We demonstrate that a previously reported algorithm for this type of situation has poor performance and present a modified version, called Independent Phase Analysis (IPA), which drastically improves the quality of results. We study the performance of IPA for different numbers of sources and discuss further improvements that are necessary for its application to real data.

Key words: phase-locking, synchrony, blind source separation (BSS), independent component analysis (ICA), subspaces, multiple runs, cost smoothness

1 Introduction

The interest of the scientific community in synchrony phenomena has risen in recent years. Synchrony has been observed in a multitude of oscillating physical processes, including electrical circuits, laser beams and human neurons [1]. This behaviour is usually not due to a strong interaction forcing in-phase oscillations, but rather a consequence of a weak interaction that slowly drifts the relative phase values of the oscillators toward one another.

Our particular motivation for studying synchrony phenomena comes from the human brain. It has been discovered that, during a motor task, several brain regions oscillate coherently with one another [2, 3]. There are multiple indications that several pathologies, including Alzheimer, Parkinson and autism, are associated with a disruption in the synchronization profile of the brain (see [4] for a review).

When trying to detect and quantify synchrony in real applications, it is important to have access to the time evolution of the individual oscillators. Otherwise, synchrony measures will not be accurate (midrange values of synchrony

^{*} Corresponding author.

become more likely than high or low values). In many real applications such as EEG and MEG, the signals from individual oscillators, henceforth denominated sources, are not directly measurable. Instead, the analyst only has access to a superposition of the sources. For example, in brain electrophysiological signals (EEG and MEG), the signals measured in one sensor contain components coming from several brain regions [5].

Blind source separation (BSS) addresses these issues. The typical instantaneous linear mixing assumption is valid in this situation, because most of the energy is in frequencies below 1 KHz, and the quasi-static approximation of Maxwell's equations holds [6]. However, for example, independence of the sources is not valid, because phase-locked sources are not independent. We now address how to correctly separate the sources in this context to avoid the erroneous detection of spurious synchronization. The separation algorithm we propose uses solely the phase information of the signals, since signals may exhibit synchrony even when their amplitudes are uncorrelated [8].

It should be emphasized that the algorithm presented here assumes nothing specific of brain signals, and should be applicable to any situation where phase-locked sources are mixed approximately linearly and where noise levels are low.

2 Background, Motivation and Algorithm

Given two oscillators with phases $\phi_j(t)$ and $\phi_k(t)$ obtained through the Hilbert transform [9] for $t = 1, \dots, T$, the Phase Locking Factor (PLF) is defined as

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

where $\langle \cdot \rangle$ denotes a time average operation. It is easy to see that $0 \leq \varrho_{jk} \leq 1$. $\varrho_{jk} = 1$ corresponds to two oscillators that are perfectly synchronized, i.e., that have a constant phase lag. Note that this lag may be non-zero, which allows for a non-instantaneous interaction. $\varrho_{jk} = 0$ is attained if the two oscillators are not synchronized as long as the observation period T is sufficiently long.

It is important to understand the effect of a linear mixture on the phase-locking between signals. Such effect can be intuitively described as “tending toward partial synchrony”: If some sources have very low synchrony (PLF ≈ 0), their mixtures have a higher PLF, since each source is present in all mixed signals. If some sources have very high synchrony (PLF ≈ 1), their mixtures have a lower PLF, because each mixed signal has components from sources that were not phase-locked. These statements are illustrated in Fig. 1. Note that significant partial synchrony is present in all pairs of mixture signals.

We now discuss how linearly mixed signals \mathbf{y} can be separated. Denote the extracted sources by $\mathbf{z} = \mathbf{W}^T \mathbf{y}$. One classic formulation of ICA [7] is the minimization of the mutual information I of the extracted sources,

$$I(\mathbf{z}) = \sum_j H[z_j] - H[\mathbf{y}] - \log |\det \mathbf{W}|, \quad (2)$$

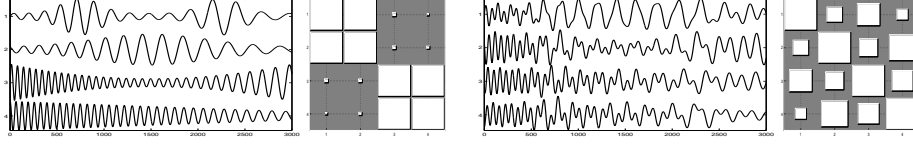


Fig. 1. Simulated data set used for comparing IPA with the logarithm and cosine cost functions: original sources (*far left*) and PLFs between them (*middle left*); mixed signals (*middle right*) and PLFs between them (*far right*). The area of each white square is proportional to the corresponding pairwise PLF. Sources 1 and 2 have a mutual constant phase difference, as do sources 3 and 4.

where \mathbf{W} is a matrix to be determined such that $\mathbf{W}^T \mathbf{A}$ (where \mathbf{A} is the mixing matrix) is close to a permutation of a diagonal matrix. The entropy of the observations $H[\mathbf{y}]$ can be dropped since it does not depend on \mathbf{W} . The sum of the estimated source entropies $\sum_j H[z_j]$ can be considered a disorder or complexity measure, and $-\log |\det \mathbf{W}|$ can be viewed as a penalty term preventing degenerate solutions with \mathbf{W} close to singular. This motivated the original formulation of Independent Phase Analysis (IPA) [10]. The disorder parameter P_{log} was given by

$$P_{log} = - \sum_{j,k} \varrho_{jk} \log \varrho_{jk}, \quad (3)$$

where ϱ_{jk} is the PLF between estimated sources j and k . P_{log} is non-negative and attains its minimum value of 0 only if all the ϱ_{jk} are either zero or one.¹ In other words, P_{log} is low when the estimated sources are either non-synchronized or fully synchronized. This suggests that minimization of P_{log} can separate sources that have PLFs close to one or zero. Thus the cost function to be minimized was

$$C_{log} = - \sum_{j,k} \varrho_{jk} \log \varrho_{jk} - \log |\det \mathbf{W}|, \quad (4)$$

where the penalty term $-\log |\det \mathbf{W}|$ prevents degenerate solutions which trivially have $\varrho_{jk} = 1$. Each column of \mathbf{W} is constrained to have unit norm, to prevent trivial decreases of the penalty term. This cost function was used in [10].

We present results showing that this formulation of IPA gives poor results. Using the data set in Fig. 1, a typical solution given by this formulation of IPA is presented in the left half of Fig. 2. The key to understanding these poor results is the presence of a PLF value of zero (between estimated sources 2 and 4), when the real sources do not have any zero values of PLF. Since the disorder parameter P_{log} falls sharply close to $\varrho_{jk} = 0$, this situation is a sharp local minimum of C_{log} . Intuitively, once the algorithm finds a PLF of zero, it “stays there”, preventing the optimization of the remaining PLF values.

Given the above considerations we propose that the disorder parameter P should still have minima for $\varrho_{jk} = 0, 1$ but these minima should not be sharp.

¹ Note that P_{log} cannot be interpreted as an entropy because $\sum_{j,k} \varrho_{jk} \neq 1$ in general.

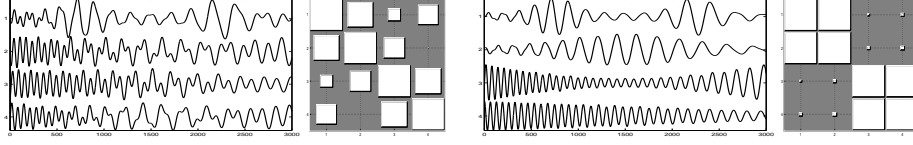


Fig. 2. Results of the IPA algorithm. Estimated sources for the logarithm cost function (*far left*) and PLFs between them (*middle left*). Estimated sources for the cosine cost function (*middle right*) and PLFs between them (*far right*). For the results of the cosine cost function, the permutation, scaling and sign of the extracted sources were manually manually. Results with $\lambda = 0.2$. The Amari Performance Index was 0.87 for the logarithm and 0.10 for the cosine cost functions, corresponding to the highest bins of Fig. 3.

Also, we propose that the disorder parameter be normalized regarding the total number of pairs of oscillators N^2 to make it scale better for large N . We thus propose a new disorder parameter, and a corresponding cost function, given by

$$P_{cos} = \frac{1}{N^2} \sum_{j,k} (1 - \cos(2\pi \varrho_{jk})) \quad (5)$$

$$C_{cos} = (1 - \lambda) \frac{1}{N^2} \sum_{j,k} (1 - \cos(2\pi \varrho_{jk})) - \lambda \log |\det \mathbf{W}|, \quad (6)$$

where λ controls the relative weight of the penalty term versus the disorder parameter. P_{cos} still has minima for $\varrho_{jk} = 0, 1$ but the derivative at those points is zero, allowing for a smoother optimization of the PLF values.

The gradient of C_{cos} relative to an entry w_{ij} of the weight matrix is given by

$$\frac{\partial C_{cos}}{\partial w_{ij}} = 4\pi \sum_{k=1}^N [\sin(2\pi \varrho_{jk})] \left\langle \sin(\Psi_{jk} - \Delta\phi_{jk}) \frac{Y_i}{Z_j} \sin(\psi_i - \phi_j) \right\rangle - [\mathbf{W}^{-\top}]_{ij} \quad (7)$$

where ϱ_{jk} is the PLF between estimated sources j and k , $Y_i = |\tilde{y}_i|$ where \tilde{y}_i is the analytic signal of the i -th measurement (obtained from the Hilbert transform [9]), $Z_j = |\tilde{z}_j|$ where \tilde{z}_j is the analytic signal of the j -th estimated source, $\psi_i = \text{angle}(\tilde{y}_i)$ is the phase of the i -th measurement, $\phi_j = \text{angle}(\tilde{z}_j)$ is the phase of the j -th estimated source, $\Delta\phi_{jk} = \phi_j - \phi_k$ is the phase difference of estimated sources j and k , $\Psi_{jk} = \text{angle}(\langle e^{i\Delta\phi_{jk}} \rangle)$ is the average phase difference between estimated sources j and k , and $[\mathbf{W}^{-\top}]_{ij}$ is the (i, j) element of the inverse of \mathbf{W}^\top . Each column of \mathbf{W} must be constrained to have unit norm to prevent trivial decreases of the penalty term.

3 Results

We present results that demonstrate that this smoother cost function drastically improves the quality of the separation. Gradient descent with adaptive step

sizes was used to perform the optimization, using MATLAB. This takes about 10 minutes on a 1.6 GHz machine. The data used are depicted in Fig. 1. We artificially simulate a noiseless mixture of four sources in two clusters of size 2 each. The mixing matrix was chosen randomly but constrained to be orthogonal (this is not strictly necessary; see Sec. 4), producing the measurements in the bottom of Fig. 1. We then run the IPA algorithm 1000 times for each of the two cost functions with random initializations of \mathbf{W} .

Typical results are shown, in Fig. 2, for the logarithm and the cosine cost function. To assess the quality of the source extraction, we use the Amari performance index (API) [11], which measures how close $\mathbf{W}^T \mathbf{A}$ is to a permutation of a diagonal matrix. The API is non-negative and the lower its value, the better the separation quality. We show histograms of the API for the logarithm and cosine cost functions in Fig. 3. The histograms show that the use of C_{cos} yields a drastic improvement in the quality of the results, relative to C_{log} . If we consider that an API below 0.3 indicates a good separation, in 71% of the tests using C_{cos} yielded good results, compared to just 2% for C_{log} .

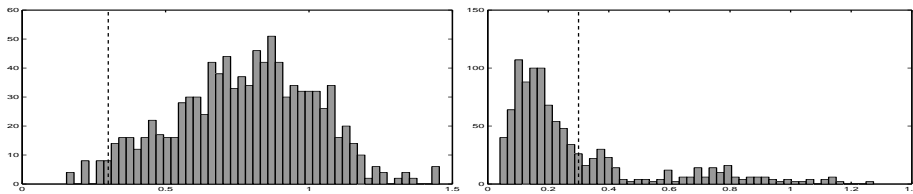


Fig. 3. Histogram of the Amari Performance Index (API) for the logarithm cost function (*left*) and for the cosine cost function (*right*), and threshold for good separations (Amari Performance Index = 0.3, *dashed vertical line*). The logarithm cost function achieves only 2% of good separations, versus 71% for the cosine cost function. Results for $\lambda = 0.2$.

A second experiment involved a larger number of sources with a more complex cluster structure. The data used and the corresponding results are shown in Fig. 4. It can be seen that the source separation was very good (API = 0.06). Naturally, since the number of parameters to optimize is N^2 , the optimization procedure took considerably longer in this case, but on a 1.6 GHz machine it still took less than an hour. The majority of the runs yielded good separations, although the percentage of good separations is slightly less than for the small data set.

4 Discussion

These results show that IPA with the cosine cost function can successfully extract mixed sources based on their phase synchronization properties. The results of the multiple runs show also that most of the runs yield good separations, and

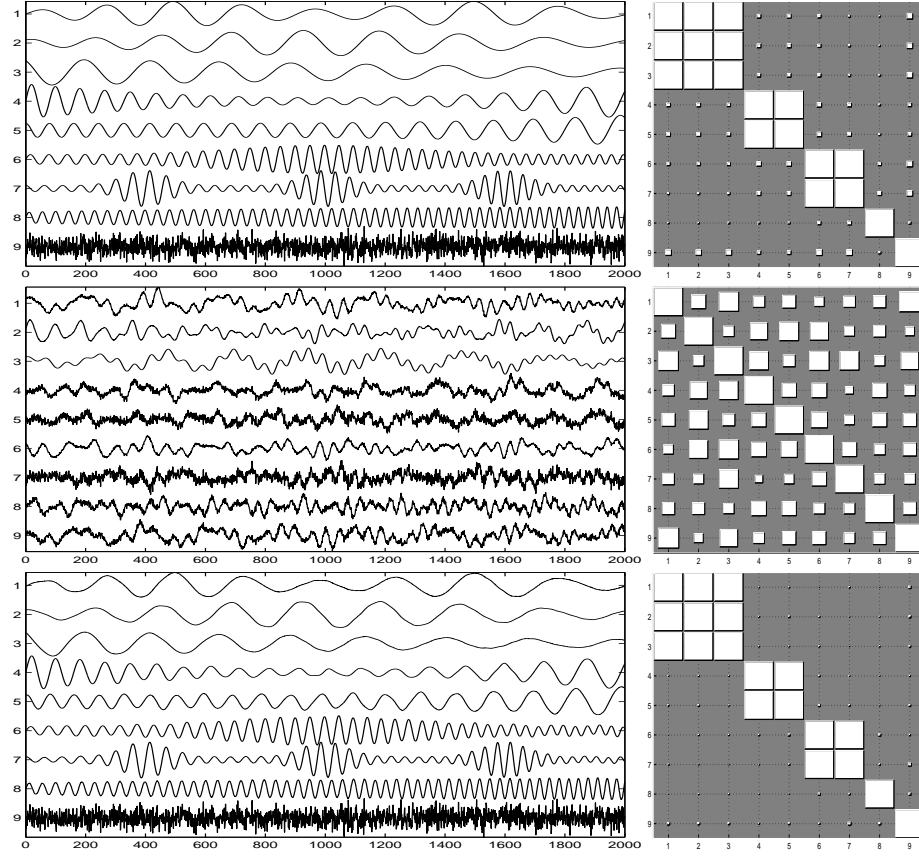


Fig. 4. Results of one run of IPA: original sources (*top left*) and PLFs between them (*top right*); mixed signals (*middle left*) and PLFs between them (*middle right*); extracted sources (*bottom left*) and PLFs between them (*bottom right*). Results obtained for $\lambda = 0.2$, after manually compensating for permutation, scaling and sign of the extracted sources. The Amari Performance Index was 0.06.

the few times that IPA yields bad results can be circumvented by running it a few times and keeping the most consistent solutions.

It could be argued that since the used sources have distinct frequencies, a simple frequency filtering algorithm could separate the signals. Such a procedure would not be able to disambiguate the signals within each cluster. Also, in real applications, IPA should prove useful for signals which are not synchronized but have overlapping frequency spectra. It could also be argued that other source separation techniques are able to separate this kind of signals. We investigated this for FastICA [7] and TDSEP [12] on the small data set (results are not shown due to lack of space). FastICA fails to separate the sources because they are not independent. TDSEP, however, gives results as good as IPA. Since IPA and TDSEP are based on different principles, further research is warranted to investigate their relative merits.

The results presented in Fig. 4 suggest that IPA will not stop when the correct PLF values are found, but will actually overtrain and yield more extreme values, although the quality of the separation is unharmed. In fact, preliminary results (not shown) suggest that IPA's performance is quite good if the sources' PLFs are above approximately 0.9 or below approximately 0.1. The decrease of performance of IPA for sources with PLF values far from 1 or 0 suggests that further improvements are required before IPA is applicable to noisy mixtures in real situations, where a) noise corrupts our estimate of the PLF values and b) even the true, noiseless underlying sources are probably not fully synchronized or fully desynchronized.

IPA has only one parameter, λ , which controls the relative weight of the disorder parameter and the penalty term. We found the algorithm to be very robust to changes in this parameter, with values between $\lambda = 0.1$ and $\lambda = 0.7$ yielding similar results. For example, using $\lambda = 0.5$ yields 65% of good separations instead of 71%. Note that if $\lambda = 0$, and if the sources have PLFs of 1 or 0, the trivial solutions with \mathbf{W} singular have the same cost value as the correct solution. Therefore, even small values of λ are enough to differentiate these two situations.

Interestingly, the optimization procedure has two different time scales: usually, the first tens of iterations of gradient descent are enough to separate the subspaces from one another (up to around 20 iterations for $N = 4$, and around 100 for $N = 9$). Even for $N = 9$ this usually takes no more than a few minutes. After this initial phase, several hundred iterations of gradient descent are necessary to separate the sources within each subspace, and convergence is slower. This suggests that using more advanced optimization algorithms might prove very useful in speeding up IPA and refining the extracted sources.

Currently, IPA works well for mixing matrices not far from orthogonal, in particular for matrices with a low value of the eigenvalue ratio $\frac{|\text{eig}_{max}|}{|\text{eig}_{min}|}$, but its performance decays for matrices with higher ratios. We believe this limitation can be avoided by making the penalty term depend on the extracted sources alone instead of the demixing matrix \mathbf{W} .

5 Conclusion

We have presented an algorithm to separate phase-locked sources from linear mixtures. We have shown that using a cosine cost function in IPA yields drastically better separation results than those of the previous version, making IPA a valid choice for separation of sources that are either synchronized or desynchronized, in noise-free situations. Further improvements are necessary to improve convergence speed and to deal with more complex real-world applications.

Acknowledgments. MA is funded by scholarship SFRH/BD/28834/2006 of the Portuguese Foundation for Science and Technology. This study was partially funded by the Academy of Finland through its Centres of Excellence Program 2006-2011.

References

1. Pikovsky, A., Rosenblum, M., Kurths, J.: Synchronization: A Universal Concept in Nonlinear Sciences. Cambridge University Press (2001).
2. Palva, J.M., Palva, S., Kaila, K.: Phase Synchrony Among Neuronal Oscillations in the Human Cortex. *Journal of Neuroscience*. 25, 3962–3972 (2005).
3. Schoffelen, J.M., Oostenveld, R., Fries, P.: Imaging the Human Motor System's Beta-Band Synchronization During Isometric Contraction. *NeuroImage*. 41, 437–447 (2008).
4. Uhlhaas, P.J., Singer, W.: Neural Synchrony in Brain Disorders: Relevance for Cognitive Dysfunctions and Pathophysiology. *Neuron*. 52, 155–168 (2006).
5. Nunez, P.L., Srinivasan, R., Westdorp, A.F., Wijesinghe, R.S., Tucker, D.M., Silberstein, R.B., and Cadusch, P.J.: EEG Coherency I: Statistics, Reference Electrode, Volume Conduction, Laplacians, Cortical Imaging, and Interpretation at Multiple Scales. *Electroencephalography and clinical Neurophysiology*. 103, 499–515 (1997).
6. Vigário, R., Särelä, J., Jousmäki, V., Hämmäläinen, M., Oja, E.: Independent Component Approach to the Analysis of EEG and MEG Recordings. *IEEE Transactions On Biomedical Engineering*. 47, 589–593 (2000).
7. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons (2001).
8. Rosenblum, M.G., Pikovsky, A.S., Kurths, J.: Phase Synchronization of Chaotic Oscillators. *Physical Review Letters*. 76, 1804–1807 (1996).
9. Oppenheim, A.V., Schaffer, R.W., Buck, J.R.: Discrete-Time Signal Processing. Prentice-Hall International Editions (1999).
10. Schleimer, J.H., Vigário, R.: Order in Complex Systems of Nonlinear Oscillators: Phase Locked Subspaces. In: *Proceedings of the European Symposium on Neural Networks* (2007).
11. Amari, S., Cichocki, A., Yang, H.H.: A New Learning Algorithm for Blind Signal Separation. *Advances in Neural Information Processing Systems*. 8, 757–763 (1996).
12. Ziehe, A., Müller, K.-R.: TDSEP - an Efficient Algorithm for Blind Separation Using Time Structure. In: *Proceedings of the International Conference on Artificial Neural Networks* (1998).

Independent Phase Analysis: Separating Phase-Locked Subspaces

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

miguel.almeida@lx.it.pt, jose.bioucas@lx.it.pt, ricardo.vigario@hut.fi

¹Institute of Telecommunications

Instituto Superior Técnico, Lisbon, Portugal

<http://www.ist.utl.pt/en/>

²Adaptive Informatics Research Centre

Aalto University School of Science and Technology, Finland

<http://www.aalto.fi/en/>

Abstract. We present a two-stage algorithm to perform blind source separation of sources organized in subspaces, where sources in different subspaces have zero phase synchrony and sources in the same subspace have full phase synchrony. Typical separation techniques such as ICA are not adequate for such signals, because phase-locked signals are not independent. We demonstrate the usefulness of this algorithm on a simulated dataset. The results show that the algorithm works very well in low-noise situations. We also discuss the necessary improvements to be made before the algorithm is able to deal with real-world signals.

Key words: phase-locking factor (PLF), synchrony, blind source separation (BSS), independent component analysis (ICA), subspaces, temporal decorrelation separation (TDSEP)

1 Introduction

Recently, synchrony phenomena have been studied with increasing frequency by the scientific community. Such phenomena have been observed in many different physical systems, including electric circuits, laser beams and human neurons [1].

Synchrony is believed to play a relevant role in the way different parts of human brain interact. For example, it is known that when humans engage in a motor task, several brain regions oscillate coherently [2, 3]. Also, several pathologies such as autism, Alzheimer and Parkinson are thought to be associated with a disruption in the synchronization profile of the brain (see [4] for a review).

To perform inference on the networks present in the brain or in other real-world systems, it is important to have access to the dynamics of the individual oscillators (which we will call “sources”). Unfortunately, in many real applications including brain electrophysiological signals (EEG and MEG), the signals from individual oscillators are not directly measurable, and one only has access

* Corresponding author.

to a superposition of the sources. For example, in EEG and MEG the signals measured in one sensor contain components coming from several brain regions [5]. In this case, spurious synchrony occurs, as we will show in this paper.

Undoing this superposition is typically called a blind source separation (BSS) problem. One usually assumes that the mixing is linear and instantaneous, which is a valid approximation in brain signals [6]. However, 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 phase-locked sources. We have previously addressed this problem with relative success [7], and in this paper we propose an improved version of that approach which yields better results, is faster and has fewer limitations. The separation algorithm we propose uses TDSEP [8] as an initialization and then uses only the phase information of the signals. The amplitude information is discarded, because signals may exhibit synchrony even when their amplitudes are uncorrelated [9].

The algorithm presented here assumes nothing specific of brain signals, and should work in any situation where phase-locked sources are mixed approximately linearly and noise levels are low.

2 Background, Notation and Algorithm

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

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

where $\langle \cdot \rangle$ is the time average operator. The PLF obeys $0 \leq \varrho_{jk} \leq 1$. The value $\varrho_{jk} = 1$ corresponds to two oscillators that are perfectly synchronized, i.e., that have a constant phase lag. The value $\varrho_{jk} = 0$ is attained if the two oscillators' phases are not correlated, as long as the observation period T is sufficiently long. The oscillators can be represented by real signals, since techniques such as the Hilbert Transform [10] can be used to extract the phase of real signals.¹ Typically, the PLF values are stored in a PLF matrix \mathbf{Q} such that $\mathbf{Q}(j, k) = \varrho_{jk}$.

We assume that we have a number of signals (called “sources”) that are organized in subspaces, such that the PLF between sources belonging to the same subspace is high, and the PLF between sources in different subspaces is low. Let $\mathbf{s}(t)$, for $t = 1, \dots, T$, denote the vector of true sources and $\mathbf{y}(t) = \mathbf{A}\mathbf{s}(t)$ denote the mixed signals, where \mathbf{A} is the mixing matrix, which is assumed to be square and non-singular. Our goal is to find an unmixing matrix \mathbf{W} such that the estimated sources $\mathbf{x}(t) = \mathbf{W}^T \mathbf{A}\mathbf{s}(t)$ are as close to the true sources as possible, up to permutation, scaling, and sign.

We now illustrate the effect of a linear mixture on PLF values. If the sources have very low or very high synchrony (PLF ≈ 0 or 1), their mixtures will have intermediate values of synchrony. This is illustrated in the top two rows of Fig. 1. Note that significant partial synchrony is present in all pairs of mixture signals.

¹ We assume that the real signals are bandpass.

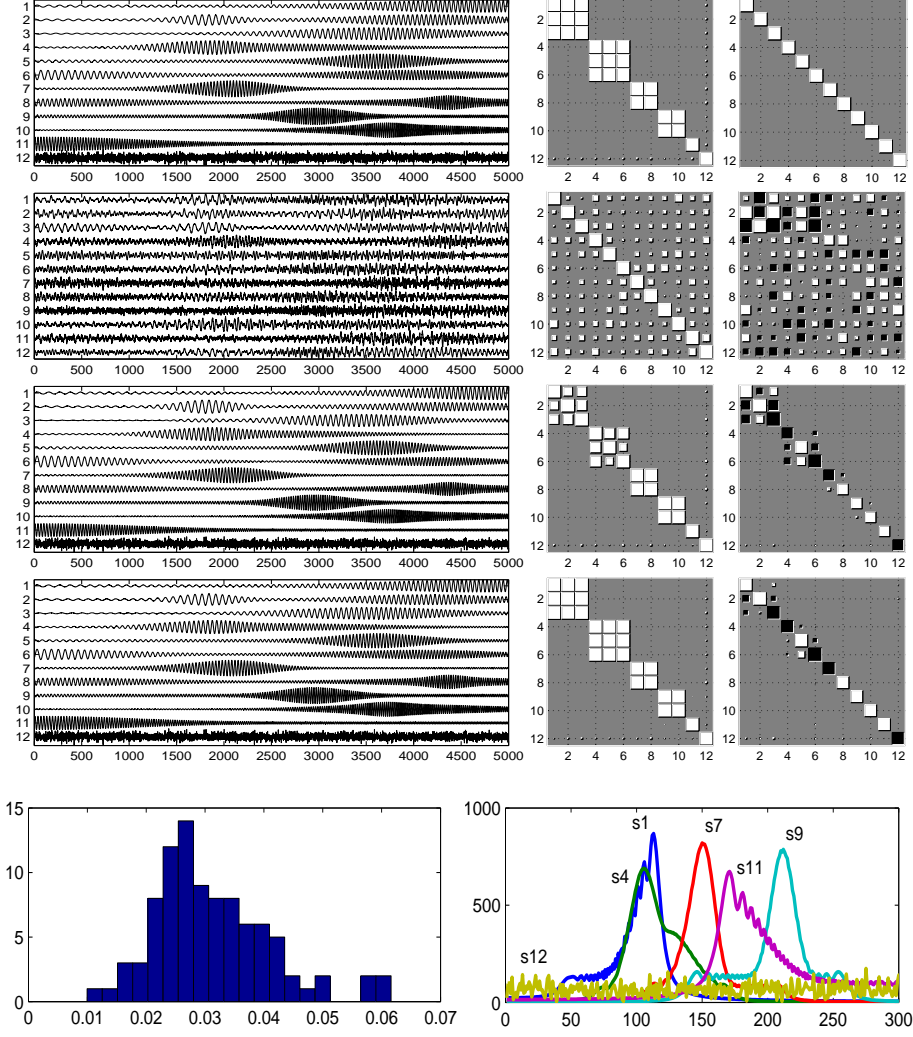


Fig. 1. The dataset used throughout this paper. (*First row*) Original sources (*left*), PLFs between them (*middle*) and the identity matrix (*right*), symbolizing that these are the true sources. (*Second row*) Mixed signals, PLFs between them and the mixing matrix \mathbf{A} . (*Third row*) Sources resulting from TDSEP (*left*). Note that the inter-subspace PLFs (*middle*) are very close to zero, but the intra-subspace PLFs are not all close to 1. Further, the intra-space separation is poor, as can be seen from inspection of the product $\mathbf{W}_{tdsep}^T \mathbf{A}$ (*right*). (*Fourth row*) Results found after the second stage of the algorithm. The estimated sources (*left*) are very similar to the original ones. This is corroborated by the PLFs between the estimated sources (*middle*) and the final unmixing matrix (*right*). In the third and fourth rows, the permutation was corrected manually. (*Bottom row*) Histogram of the Amari Performance Index for 100 runs, corresponding to 100 random mixing matrices for these sources (*left*), and zoom-in of the Discrete Fourier Transform of the first, fourth, seventh, ninth, eleventh and twelfth sources, one from each subspace (*right*).

Since linearly mixing sources which have PLFs close to 1 or 0 yields signals which have partial synchrony, one can reason that finding an unmixing matrix such that the estimated sources have PLFs of 1 or 0 can separate such sources. We have previously used such an approach, motivated by ICA, to separate this kind of sources [7]. That approach, which we called Independent Phase Analysis (IPA), showed decent results, but was limited to near-orthogonal mixing matrices. A non-near-orthogonal mixing matrix yielded, however, poor results. This poor performance is closely related to the fact that, given two phase-locked signals, it is always possible to construct a linear combination of them which has a PLF of zero with those signals.² This implies that our problem is ill-posed, when the goal is to find an unmixing matrix such that the PLF matrix has only zeroes and ones. Some zero PLFs may, in fact, correspond to phase-locked signals.

To avoid the above referred ill-posedness, we introduce in this paper a different unmixing criterion which consists in dividing the problem into two subproblems: first, separate the subspaces from one another, even if within the subspaces some mixing remains. Second, unmix the sources within each subspace. We now discuss each of these subproblems in detail and explain why this new problem is no longer ill-posed. Since this method is an improvement of the previous approach [7], we will continue to name it Independent Phase Analysis.

2.1 Inter-subspace separation and subspace detection

The objective of the first stage is to find an unmixing matrix \mathbf{W} such that the estimated subspaces are correct, even if sources within each subspace are still mixed. We assume that signals in different subspaces have little interaction with each other, which should usually correspond to a distinct time structure. Therefore, techniques that use temporal structure to perform separation should be adequate to this first stage. We chose to use Ziehe *et. al.*'s implementation of TDSEP [8] for this first subproblem, but SOBI [11] can be used instead. Although we don't know any theoretical results that support TDSEP's adequacy to this task, we have repeatedly observed that it separates subspaces quite well.

A non-trivial step is the detection of subspaces from the result of TDSEP. From our experience, TDSEP can perform the inter-subspace separation very well but cannot adequately do the intra-subspace separation. This means that PLF values within each subspace will be underestimated. Naively, one can arbitrarily define a hard PLF threshold, below which signals are considered not synchronized and above which signals are considered in full phase synchrony.

The matrix resulting from this hard thresholding should be block-diagonal, with each block having all elements equal to 1. If this is the case, we can group the found signals into subspaces, and we further unmix each of these subspaces at a time (see section 2.2). By inverting and transposing the mixing matrix estimated by TDSEP, we have a first estimate of the unmixing matrix \mathbf{W}_{tdsep} .

² Unfortunately, the proof of this claim is too lengthy to show here.

If the matrix resulting from the thresholding is not block-diagonal with blocks with all elements equal to 1, our algorithm considers that the subspaces were wrongly detected and fails. See section 4 for possible improvements on this.

2.2 Intra-subspace separation

In the second stage of IPA, we begin by selecting the subset of columns of the unmixing matrix \mathbf{W}_{tdsep} found by TDSEP that form the l -th subspace, which we denote by S_l . We construct a rectangular matrix $\mathbf{W}_{tdsep,l}$ from those columns. Denote the number of signals in the l -th subspace by N_l , and let $\mathbf{z}_l(t) = \mathbf{W}_{tdsep,l}^T \mathbf{y}(t)$ be the vector of the sources of subspace S_l estimated by TDSEP.

In this second stage, our goal is to separate each of these subsets of sources. As explained above, the true sources should have a PLF of 1 with the sources in the same subspace. We should therefore unmix the N_l sources found by TDSEP such that their PLFs are as high as possible. Mathematically, this corresponds to finding a N_l by N_l matrix \mathbf{W}_l such that the estimated sources in the l -th subspace, $\mathbf{x}_l(t) = \mathbf{W}_l^T \mathbf{W}_{tdsep,l}^T \mathbf{y}(t) = \mathbf{W}_l^T \mathbf{z}_l(t)$, have the highest possible PLFs.

In this second stage, for each subspace l , the objective function to be maximized is

$$J_l = (1 - \lambda) \sum_{j,k \in S_l} \varrho_{jk}^2 + \lambda \log |\det \mathbf{W}_l|, \quad (2)$$

where $\sum_{j,k \in S_l} \varrho_{jk}^2$ is a sum over all pairs of sources in subspace S_l of the PLF between those sources squared. We use the Hilbert Transform [10] to obtain the phase values of the estimated sources. The second term, similarly to ICA [12], penalizes unmixing matrices that are close to singular, and λ is a parameter controlling the relative weight of the two terms. The second term, already present in the previous version of IPA [7], serves the purpose of preventing the algorithm from finding solutions which trivially have $\varrho_{jk} = 1$. Each column of \mathbf{W} must be constrained to have unit norm to prevent trivial decreases of the penalty term.

With this formulation, the problem is no longer ill-posed as described above. Furthermore, we now need only optimize a subset of parameters at a time. This can drastically reduce the time needed to separate a set of sources.

The gradient of J_l relative to an entry w_{ij} of the weight matrix \mathbf{W}_l is given by (we omit many dependences on l for clarity)

$$\frac{\partial J_l}{\partial w_{ij}} = (1 - \lambda) 4\pi \sum_{k=1}^N [2\varrho_{jk}] \left\langle \sin(\Psi_{jk} - \Delta\phi_{jk}) \frac{Z_i}{X_j} \sin(\psi_i - \phi_j) \right\rangle - \lambda [\mathbf{W}_l^{-T}]_{ij} \quad (3)$$

where ϱ_{jk} is the PLF between estimated sources j and k , $Z_i = |\tilde{z}_i|$ where \tilde{z}_i is the analytic signal of the i -th source estimated by TDSEP, $X_j = |\tilde{x}_j|$ where \tilde{x}_j is the analytic signal of the j -th estimated source, $\psi_i = \text{angle}(\tilde{z}_i)$ is the phase of the i -th source found by TDSEP, $\phi_j = \text{angle}(\tilde{x}_j)$ is the phase of the j -th estimated source, $\Delta\phi_{jk} = \phi_j - \phi_k$ is the phase difference of estimated sources j and k , $\Psi_{jk} = \text{angle}(\langle e^{i\Delta\phi_{jk}} \rangle)$ is the average phase difference between estimated sources j and k , and $[\mathbf{W}_l^{-T}]_{ij}$ is the (i, j) element of the inverse of \mathbf{W}_l^T .

3 Results

We present results showing that this new approach provides drastic improvements on the separation quality. The optimization was done using a gradient algorithm with adaptive step sizes, running up to 600 gradient iterations until the average PLF within the subspace is greater than 0.9999. λ was hand tuned for optimal performance. However, the algorithm yields similar results for λ within a factor of 2 of the optimal one, which is $\lambda = 0.1$ in this case.

We simulate the noiseless instantaneous linear mixture of 12 sources depicted in the first row of Fig. 1. These sources belong to 6 clusters of sizes 3, 2 and 1. We generate 100 datasets from these sources, by generating 100 mixing matrices, each of which with elements i.i.d. from the Uniform(-1,1) distribution. We then run the algorithm once for each mixing matrix, for a total of 100 runs. Each run took about 1 minute on a modern laptop computer.

The second row of Fig. 1 shows the mixed signals which are the input to our algorithm. The third row shows the sources estimated by TDSEP. Inspection of the PLFs between these sources (shown on the subfigure in the second column, third row) shows that some of the estimated sources do not have high PLFs, and an inspection of the product $\mathbf{W}_{tdsep}^T \mathbf{A}$ reveals that the inter-subspace separation was very good, but the intra-subspace separation was poor.

The fourth row of Fig. 1 shows that by maximizing the intra-subspace PLFs we can significantly improve the separation within each subspace. This is clearly visible in the product $\mathbf{W}^T \mathbf{A}$, depicted on the third column.

We measure the performance of IPA using the Amari Performance Index (API) [13], which measures the average relative contamination in each estimated source from all other sources. The API is non-negative and decreases to zero as the separation quality increases. A histogram of the API for the 100 runs of IPA is shown in Fig. 1. The mode of this histogram corresponds to an API of 0.0267. This is very similar to the example in Fig. 1, which has an API of 0.0265.

We used a threshold of 0.1 on the PLF matrix for the detection of subspaces. In 7% of the runs, the resulting matrix is not block-diagonal with each block full of ones, and therefore the algorithm stops.

4 Discussion

The above results demonstrate that IPA can successfully extract mixed sources based on their phase synchrony values and that it performs considerably better than the previous version. The previous version had 71% of separations with API below 0.3 [7], while this new version has all runs below 0.06, which is a remarkable improvement. Furthermore, the new version of IPA works well with any mixing matrix, even if it is close to singular. The previous version worked well only if the mixing matrix was close to orthogonal [7].

It should be noted that a PLF value of zero does not imply that the signals have distinct frequency spectra. In fact, the subspaces used here have overlapping

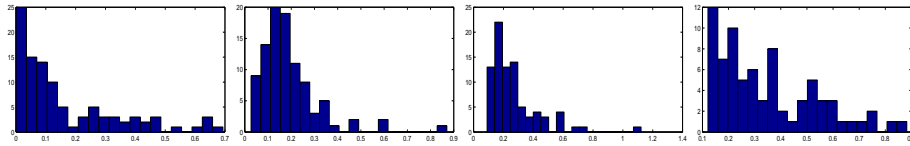


Fig. 2. Histograms of API for 100 runs of separate datasets containing a single cluster of 2 (*far left*), 3 (*middle left*), 4 (*middle right*) and 5 (*far right*) signals.

frequency spectra (shown in the bottom right corner of Fig. 1), and an attempt at separating the subspaces based on Fourier transforms alone would fail.

Our selection of the value 0.1 for the PLF threshold was empirical. Although all the successful separations were very good, 7% of the runs yielded cases where our (admittedly crude) subspace detection procedure failed. This suggests that an improved subspace detection procedure is the most important improvement to be made, and it is one we are actively working on. Also, although the applicability of IPA is rather general, further research is warranted to search for TDSEP’s “blind spots”: signals that have similar temporal structure but low PLF values, and signals that have different temporal structure and high PLF values. The possible existence of such signals might make TDSEP identify the wrong subspaces, and the optimization stage will produce erroneous results.

Although this algorithm works quite well for this simulated data, several improvements must be made for it to be usable in real data. First of all, we have noted that the performance of IPA degrades when it is used with large subspaces. Subspaces of 1, 2 or 3 signals can be separated quite well, but on subspaces of 4 signals the performance begins to decrease. For 5 or more signals in a subspace, performance becomes considerably worse. To illustrate this limitation, we present in Fig. 2 histograms of 100 runs of the algorithm on datasets with 2, 3, 4 and 5 signals, all belonging to a single subspace.

Another unclear aspect is how this algorithm will perform if the true sources do not have extreme values of PLF but more intermediate values. It is possible that the second stage of the algorithm will overfit in such situations.

The major hindrance to be overcome before IPA can be applied to real signals is to make it work under noisy conditions. IPA performs quite well for noiseless mixtures, but real signals will always have some amount of noise. Preliminary results show that the algorithm can tolerate small amounts of noise (up to 1% of noise amplitude relative to the signal amplitude), but this still needs to be improved prior to its application to real world signals.

5 Conclusion

We have presented a two-stage algorithm, called Independent Phase Analysis (IPA), to separate phase-locked subspaces from linear mixtures. We have shown that this approach yields much better results than the previous version of IPA, and that it is no longer limited to near-orthogonal mixing matrices. Our results

show that although TDSEP alone is not enough to adequately separate phase-locked sources, its conjunction with a subsequent intra-subspace separation gives very good separation quality in low-noise situations. Nevertheless, improvements are necessary before this algorithm can be applied to real-world signals.

Acknowledgments. MA is funded by scholarship SFRH/BD/28834/2006 of the Portuguese Foundation for Science and Technology. This study was partially funded by the Academy of Finland through its Centres of Excellence Program 2006-2011.

References

1. Pikovsky, A., Rosenblum, M., Kurths, J.: *Synchronization: A Universal Concept in Nonlinear Sciences*. Cambridge University Press (2001).
2. Palva, J.M., Palva, S., Kaila, K.: Phase Synchrony Among Neuronal Oscillations in the Human Cortex. *Journal of Neuroscience*. 25, 3962–3972 (2005).
3. Schoffelen, J.M., Oostenveld, R., Fries, P.: Imaging the Human Motor System's Beta-Band Synchronization During Isometric Contraction. *NeuroImage*. 41, 437–447 (2008).
4. Uhlhaas, P.J., Singer, W.: Neural Synchrony in Brain Disorders: Relevance for Cognitive Dysfunctions and Pathophysiology. *Neuron*. 52, 155–168 (2006).
5. Nunez, P.L., Srinivasan, R., Westdorp, A.F., Wijesinghe, R.S., Tucker, D.M., Silberstein, R.B., and Cadusch, P.J.: EEG Coherency I: Statistics, Reference Electrode, Volume Conduction, Laplacians, Cortical Imaging, and Interpretation at Multiple Scales. *Electroencephalography and clinical Neurophysiology*. 103, 499–515 (1997).
6. Vigário, R., Särelä, J., Jousmäki, V., Hämäläinen, M., Oja, E.: Independent Component Approach to the Analysis of EEG and MEG Recordings. *IEEE Transactions On Biomedical Engineering*. 47, 589–593 (2000).
7. Almeida, M., Vigário, R.: Source-Separation of Phase-Locked Subspaces, *Proceedings of the Independent Component Analysis Conference*, 2009.
8. Ziehe, A., Müller, K.-R.: TDSEP - an Efficient Algorithm for Blind Separation Using Time Structure. In: *Proceedings of the International Conference on Artificial Neural Networks* (1998).
9. Rosenblum, M.G., Pikovsky, A.S., Kurths, J.: Phase Synchronization of Chaotic Oscillators. *Physical Review Letters*. 76, 1804–1807 (1996).
10. Oppenheim, A.V., Schafer, R.W., Buck, J.R.: *Discrete-Time Signal Processing*. Prentice-Hall International Editions (1999).
11. Belouchrani, A., Abed-Meraim, K., Cardoso, J.-F., Moulines, E.: A Blind Source Separation Technique Using Second Order Statistics. *IEEE Transactions on Signal Processing*. 45, 434–444 (1997).
12. Hyvärinen, A., Karhunen, J., Oja, E.: *Independent Component Analysis*. John Wiley & Sons (2001).
13. Amari, S., Cichocki, A., Yang, H.H.: A New Learning Algorithm for Blind Signal Separation. *Advances in Neural Information Processing Systems*. 8, 757–763 (1996).

Source Separation and Clustering of Phase-Locked Subspaces

Miguel Almeida, Jan-Hendrik Schleimer, José Bioucas-Dias, Ricardo Vigário

Abstract—It has been proven that there are synchrony (or phase-locking) phenomena present in multiple oscillating systems such as electrical circuits, lasers, chemical reactions and human neurons. If the measurements of these systems cannot detect the individual oscillators but rather a superposition of them, as in brain electrophysiological signals (EEG and MEG), spurious phase-locking will be detected. Current source-extraction techniques attempt to undo this superposition by assuming properties on the data which are not valid when underlying sources are phase-locked. Statistical independence of the sources is one such invalid assumption, as phase-locked sources are dependent. In this article we introduce methods for source separation and clustering which make adequate assumptions for data where synchrony is present, and show with simulated data that they perform well even in cases where ICA and other well-known source-separation methods fail. The results in this paper provide a “proof-of-concept” that synchrony-based techniques are useful for low-noise applications.

Index Terms—phase-locking, synchrony, source separation, clustering, subspaces

I. INTRODUCTION

In recent years there has been an increased scientific interest in the study of synchrony-related phenomena as well as in the amount of relevant results in this field, both of theoretical and empirical nature. These phenomena are present in a multitude of physical substrata. The first detection of synchrony was made by Huygens in the XVII century, when he observed that two pendulum clocks interacting through a common supporting beam would always synchronize after a brief transient period [1]. Since then, other systems were found to exhibit similar behaviour, such as organ-pipes, electrical circuits, laser beams, astrophysical objects, some types of fireflies, and human neurons, among others. This behaviour is often not caused by a strong interaction which forces the oscillators to oscillate in phase, but by weak interactions between the individual oscillators which, in time, drift their individual phases towards one another [1].

Our particular motivation for studying synchrony phenomena comes from the human brain. As an example, it has been shown that muscle activity measured with an electromyogram (EMG) and the activity of the motor cortex measured with an electroencephalogram (EEG) or magnetoencephalogram

(MEG) have coherent oscillations when a person engages in a motor task. Because these coherent oscillations occur mostly in the beta-band¹ [2], [3], [4], [5], the cortico-muscular coherence (CMC) phenomena are also denominated “beta-band synchronization”, although some studies have showed that they also occur in other frequency ranges [6], [7]. It was also found that, again during a motor task, several brain regions oscillate coherently with one another [8], [5]. In addition, there are multiple indications that several pathologies, including Alzheimer, Parkinson and autism, are correlated with a disruption in the synchronization profile of the brain [9].

The typical formulation of a synchronization behavior is based on self-sustained oscillators as individual units. A self-sustained oscillator is a dynamical system, such as a pendulum clock, which has an intrinsic energy source and exhibits a periodic motion when isolated [1]. More rigorously, these oscillators have a limit cycle, which can be defined as a periodic trajectory in the oscillator’s phase space. The position along the limit cycle is the oscillator’s phase. Also, at least in a small neighborhood of this cycle, a self-sustained oscillator is stable, which means that after being slightly perturbed, the trajectory always returns to the limit cycle after some time.

This return to the limit cycle has two distinct aspects. Perturbations along the limit cycle do not decay, and will result in a permanent change to the oscillator’s phase. On the other hand, perturbations orthogonal to this cycle decay exponentially.² For these reasons, a weak interaction can have a long-lasting effect on an oscillator’s phase, but the influence on its amplitude can be disregarded.

Assuming a weak, attractive and time-invariant coupling, the interaction of N self-sustained oscillators affects their phases only, and can be described by the Kuramoto model [11],

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

where $\phi_i(t)$ and $\omega_i(t)$ are the instantaneous phase and intrinsic (natural) frequency of oscillator i , and $\kappa_{ij} > 0$ is called the coupling coefficient between oscillators i and j . Note that in this model, the coupling coefficients are constant in time, which implicitly assumes that the coupling between the oscillators is stationary in time. We do not address here the case of time-dependent coupling. The influence on ϕ_i is fully determined by the phase differences between oscillator i and each of the other oscillators. If oscillator j is slightly ahead

¹The beta-band range of brain signals is loosely defined as 15-30 Hz.

²The direction tangent to the limit cycle has a zero Lyapunov exponent, while the orthogonal directions have negative Lyapunov exponents [10].

M. Almeida is with Instituto de Telecomunicações, Instituto Superior Técnico, Technical University of Lisbon, Portugal. Email: malmeida@lx.it.pt

J.-H. Schleimer is with Bernstein Center for Computational Neuroscience, Humboldt University, Berlin. Email: jan-hendrik.schleimer@bccn-berlin.de

J. Bioucas-Dias is with Instituto de Telecomunicações, Instituto Superior Técnico, Technical University of Lisbon, Portugal. Email: bioucas@lx.it.pt

R. Vigário is with Adaptive Informatics Research Centre, Aalto University, Finland. Email: ricardo.vigario@hut.fi

of oscillator i (ϕ_j is slightly larger than ϕ_i), this will make oscillator i go slightly faster. Conversely, if oscillator j is slightly behind oscillator i , the latter will slow down. In both cases this interaction tends to make the pairs of oscillators approach each other in phase, and synchrony can occur if this interaction is enough to compensate the difference in intrinsic frequencies $\omega_i - \omega_j$, called detuning. Naturally, if the detuning is large, a weak interaction cannot compensate it and the oscillators will never synchronize. These ideas are thoroughly explained in [1] and references therein.

Synchrony has been studied from a theoretical point of view since the original formulation by Kuramoto. Golomb *et al.* [12] have shown that in a large and sparse neural network with weak coupling there is a threshold value of the average number of synapses a neuron receives, above which synchrony is established and below which it is not, in a way reminiscent of Kuramoto's critical coupling coefficient [11]. The value of this threshold can be analytically found from the parameters of an Integrate-and-Fire model. Integrate-and-Fire oscillators, also known as relaxation oscillators, are described in more detail in the context of synchrony in [1], [13], [14].

In real applications, it is often the case that one does not have direct access to the activity of the individual oscillators. Instead, one only has access to measurements of superpositions of the individual oscillators. The underlying oscillators are usually called sources in this context. When this is the case, synchrony between measurements is mostly due not to a true phase-locking of the underlying sources, but instead to the mixture itself. We will discuss and illustrate this in Sec. II.

In this paper, we tackle the general problem of analyzing synchrony of the sources in the special case where the superposition is an instantaneous linear mixing. Such problem is widely called "blind source separation" (BSS), because one generally only assumes rather generic properties of the sources, such as statistical independence or temporal structure. One important example of this problem is in brain electrophysiological signals (EEG and MEG), where instantaneous linear mixing is a valid assumption. This comes from the fact that most of the energy is in frequencies below 1 KHz, allowing the use of the quasistatic approximation in Maxwell's equations [15].

Independent Component Analysis (ICA) [16] is one of the most widely used blind source-separation techniques. Traditional ICA approaches only use probability density function information (such as kurtosis) and therefore disregard the time dynamics of the sources. Thus, in applications where phase is relevant, ICA is not appropriate. A related approach, Temporal Decorrelation Separation (TDSEP) [17], which is similar to Second-Order Blind Identification (SOBI) [18], is better suited for source separation where dynamics are relevant. We will refer to these two approaches as "TD methods". TD methods extract sources using information on their autocorrelation function for several time-lags simultaneously. Apart from the independence and temporal decorrelation criteria, approaches based on Non-Negative Matrix Factorization (NMF) have also been used (*c.f.* [19], [20] for recent examples).

The main novelty of our work is that we focus exclusively on the phase information of the sources. By finding the sources in an appropriate way, and analyzing the synchrony between

those, we avoid the problem of spurious synchronization mentioned above, allowing for more contrast between synchronized and non-synchronized pairs of signals, thus permitting sharper detection. Two of the methods we propose are source-separation algorithms based on phase synchrony,³ applicable when one has a reference signal (Referenced Phase Analysis, RPA) or in a blind manner (Independent Phase Analysis, IPA). They are described in Secs. III-A and III-B respectively.

We now review previous approaches on synchrony and coherence (which is a different but related concept), and their applications on brain signals. Vigário *et al.* and Meinecke *et al.* use TD methods to separate sources and observe that some of them are coherent [22] or synchronized [23]. This is a valid approach, but coherence and synchrony come as an epiphenomenon, rather than constituting the main searching criterion. We wish to focus on approaches that tackle synchrony directly, so that it is clear why the extracted sources are synchronous. We will show empirically in this paper that for certain types of sources TD methods are not adequate for source separation.

Nolte *et al.* [24] use the imaginary part of the Fourier coherence (IPC), noting that instantaneous mixing does not affect this measure. IPC does not work on the source space, but only with the observed mixtures. Nolte *et al.* argue that this method should not replace other methods, but should instead be regarded as a "safe" method due to its low rate of false positives. In at least one experimental study, the IPC did not find any consistent results [25]. A drawback of this approach is that it has no time resolution. Therefore, it is only mostly applicable to stationary signals, and it will not be able to resolve non-stationarities. IPC should not be regarded as a BSS approach but rather as an approach that circumvents the problems that come from the superposition of the sources.

Allefeld and Kurths [26], [27] have proposed the Synchronization Cluster Analysis (SCA) method, which detects synchrony through the eigendecomposition of the real-valued Phase Locking Factor (PLF) matrix thus finding dominant phase-locked populations in the measured signals. This approach has the drawback of clustering together populations that are not interacting but that have similar frequencies. One of the methods we propose (phase SCA) is a generalization of such approach, which circumvents this drawback by using a complex-valued version of the PLF matrix (see Sec. III-C). Both SCA and pSCA are clustering methods and should not be considered source-separation techniques.

The methods mentioned above vary on how the obtained phase is used. Apart from the methods we propose, only SCA focuses exclusively on the phase information. It has been shown that coupled oscillators can exhibit regimes with uncorrelated amplitudes but with bounded phase lag [28]. We therefore argue that there is a need for methods which focus only on the phase information. The techniques we propose attempt to fulfill this need: our methods focus on the phase information, unlike the traditional coherence-based approaches used in CMC studies. We believe the proposed methods can contribute to a more precise characterization of synchrony

³Phase synchrony is a natural measure of dependence between signals. Other non-independent source separation approaches have seen a recent growing interest, *c.f.*, [21].

in brain-brain or brain-muscle interactions. However, it is important to keep in mind that the algorithms we introduce here are applicable to any context where synchrony occurs, and do not assume anything specific of brain or muscle signals.

This paper is organized as follows. In Sec. II we provide some background on synchrony and related concepts. The new algorithms we propose are detailed in Sec. III, along with results for simulated data. We discuss the algorithms' limitations and results in Sec. IV, and present concluding remarks in Sec. V. This paper summarizes and extends results previously reported in [29], [30], [31], [32], [33].

II. SYNCHRONY

In this section we provide some background on phase synchrony. We begin by noting the difference between synchrony and coherence. We then review the Hilbert transform and related concepts, introduce the Phase-Locking Factor quantity which is central in the methods we propose, and define the synchronization matrix. We conclude with an illustration of the effect of linear mixing in synchrony.

a) Synchrony is not Coherence: Terms such as “synchrony” and “coherence” are usually used to describe quantities related to frequency and phase. Here we make a formal distinction, to prevent confusion. Given two real signals $x(t)$ and $y(t)$, with power spectra $P_{xx}(\omega)$ and $P_{yy}(\omega)$, their spectral (or Fourier) coherence, also called coherence or coherency, is

$$C(\omega) \equiv \frac{|P_{xy}(\omega)|^2}{P_{xx}(\omega)P_{yy}(\omega)},$$

where $P_{xy}(\omega)$ is the cross-spectrum between the two signals. This quantity measures the similarity of two Fourier spectra. It can be understood as a correlation factor in frequency.

Since the Fourier transform forfeits time resolution, coherence measures are hard to interpret when computed from non-stationary signals, such as brain signals. However, coherence measures exhibiting time resolution have been proposed based on the Morlet wavelet transform. This “wavelet transform coherence” is quite popular in geophysics [34], [35] but has also been used in the biomedical field [36].

b) Phase of a real-valued signal: Typically, the signals under analysis are real-valued discrete signals. To obtain the phase of a real signal, popular choices include using a complex Morlet (or Gabor) wavelet, which can be seen as a bank of bandpass filters [37], and using the Hilbert transform, which has to be applied to a locally narrowband signal or be preceded by appropriate filtering [38]. The Hilbert transform is then applied to multiple frequency bands of the original signal. The two approaches have been shown to be equivalent for the study of brain signals [39], but they may differ for other kinds of signals. In real applications, appropriate filtering must be used to ensure that the signals under analysis are narrow-band. In this work, the analyzed signals are narrow-band by construction, thus we use the Hilbert transform.

c) Hilbert transform, analytic signal and phase: The discrete Hilbert transform $x_h(t)$ of a band-limited discrete-time signal $x(t)$, $t \in \mathbb{Z}$, is given by a convolution [40]:

$$x_h(t) \equiv x(t) * h(t), \text{ where } h(t) \equiv \begin{cases} 0, & \text{for } t = 0 \\ \frac{1-e^{i\pi t}}{\pi t}, & \text{for } t \neq 0. \end{cases}$$

The Hilbert filter $h(t)$ is not causal and has infinite duration, which makes direct implementation of the above formula impossible. In practice, the Hilbert transform is usually computed in the frequency domain, where the above convolution becomes a product of the discrete Fourier transforms of $x(t)$ and $h(t)$. A more thorough mathematical explanation of this transform is given in [38] and [40]. We used the Hilbert transform as implemented in MATLAB.

The analytic signal of $x(t)$, denoted by $\tilde{x}(t)$, is given by $\tilde{x}(t) \equiv x(t) + i x_h(t)$, where $i = \sqrt{-1}$ is the imaginary unit. The phase of $x(t)$ is defined as the angle of its analytic signal.

d) Phase Locking Factor: The common feature of our proposed methods is their use of the Phase Locking Factor (PLF), which plays a central role in the study of synchrony phenomena [41]. For two oscillators with phases $\phi_1(t)$ and $\phi_2(t)$ for $t = 1, \dots, T$, the PLF is defined as⁴

$$\varrho \equiv \frac{1}{T} \sum_{t=1}^T e^{i[\phi_1(t) - \phi_2(t)]} = \left\langle e^{i(\phi_1 - \phi_2)} \right\rangle,$$

where $\langle \cdot \rangle$ denotes a time average operation. The PLF obeys the constraints $0 \leq |\varrho| \leq 1$. The value of 1 is attained when the two oscillators are perfectly synchronized, *i.e.*, they have a constant phase lag. The value of 0 is attained, *e.g.*, if the phase difference $\phi_1 - \phi_2$ is uniformly distributed in $[0, 2\pi]$. Values between 0 and 1 indicate partial synchrony. Just as the coherence can be seen as a correlation factor in frequency, the PLF can be seen as a correlation factor in phase. Some authors (see, *e.g.*, [27] and [41]), use the absolute value of ϱ as PLF. However, this complex version of the PLF can be used to obtain better results, as we will show in Sec. III-C.

e) Synchronization matrix: When one has multiple signals with phases $\phi_1(t), \dots, \phi_N(t)$, it is common to compute all the pairwise PLFs and store them in a complex synchronization matrix (or PLF matrix) \mathbf{Q} :

$$\mathbf{Q}(j, k) \equiv \left\langle e^{i(\phi_j - \phi_k)} \right\rangle = \langle \mathbf{a} \mathbf{a}^H \rangle, \quad (2)$$

where $\mathbf{a} \equiv [e^{i\phi_1(t)} \dots e^{i\phi_N(t)}]^T$ and $(\cdot)^H$ denotes the Hermitian. This matrix has all its diagonal values equal to 1, and it is Hermitian, which means that all its eigenvalues are real. Also, for any vector \mathbf{x} , $\mathbf{x}^H \mathbf{Q} \mathbf{x} = \mathbf{x}^H \langle \mathbf{a} \mathbf{a}^H \rangle \mathbf{x} = \langle \mathbf{x}^H \mathbf{a} \mathbf{a}^H \mathbf{x} \rangle = \langle \|\mathbf{x}^H \mathbf{a}\|^2 \rangle \geq 0$. Therefore \mathbf{Q} is positive semi-definite. Unlike the definition used in [27], in our case this matrix is complex.

f) Effect of mixing in synchrony: Two of the methods described in this article aim at unmixing linear and instantaneous mixtures of sources, using the Phase-Locking Factor as the main criterion. As motivated above, in EEG and MEG it is plausible to assume that the observed measurements are the result of a linear and instantaneous mixture of the underlying source signals. Thus, it is important to understand the effect of such mixing on the PLF. Such effect can be summarized as “tending towards partial synchrony”: if some sources have low synchrony (PLF ≈ 0), the mixed signals have a higher PLF, since each source is now present in both mixed signals. If some sources have high synchrony (PLF ≈ 1), the mixed

⁴Recall that in Eq. 1 we assumed stationarity of the coupling coefficient, hence the use of the whole observation period in computing the PLF.

signals have a lower PLF, because now each mixed signal has components from sources that were not phase-locked.

These statements are illustrated in Fig. 1. The top left subfigure shows a set of 12 sources. The pairwise PLFs of these sources are shown in the top middle subfigure, where larger squares represent higher PLF values. A random linear mixture of these 12 sources is depicted in the bottom left subfigure, and the pairwise PLFs of the mixed signals are shown in the bottom middle subfigure. Note that partial synchrony is now present in all pairs of measurements. This experiment illustrates the need for source separation methods that tackle synchrony problems directly. These methods should not be regarded as replacements or improvements of techniques such as ICA or TDSEP, but rather as related techniques that are applicable for different kinds of data.

III. PHASE-LOCKING FACTOR ALGORITHMS

In this section, we introduce three methods to analyze synchrony. We show their usefulness with simulated examples using instantaneous linear mixing.

- Referenced Phase Analysis (RPA) performs non-blind source-separation when a reference signal is available, extracting the projection which has maximum PLF with the reference.
- Independent Phase Analysis (IPA) performs blind unmixing of sources that are organized in subspaces, i.e., sources that have PLFs of one or zero with one another. IPA works in two stages: it first unmixes the subspaces, and then unmixes the sources within each subspace.
- Phase SCA (pSCA) is a non-trivial generalization of Synchronization Cluster Analysis (SCA) [27]. In the original formulation of SCA, the essential procedure is to perform an eigendecomposition of the real-valued synchronization matrix, i.e., the elementwise absolute value of the synchronization matrix defined in Sec. II. We will show that instead of forfeiting the phase information, when taking the absolute value of the PLF, it is more useful to include it, to prevent multiple populations from ending in the same cluster, a problem which was present in the original SCA and which will be illustrated here.

A. Referenced Phase Analysis

In electrophysiological recording signals from the brain, such as EEG and MEG, we often have not only the scalp measurements, but also a reference signal which provides a hint of what to look for in the brain. This reference signal could be, for example, an electromyogram (EMG), collecting information of a muscle's activity. The objective of Referenced Phase Analysis is to extract a source which is phase-locked to a reference signal [29]. In the context of brain signals it would allow, for example, the identification of which cortical areas are phase-locked to the muscle activity, evidencing neuronal control of the muscle or sensory feedback from it.

As usual in linear separation techniques, we assume that the observations $\mathbf{x}(t)$ result from a linear and instantaneous superposition of the sources $\mathbf{s}(t)$, as in

$$\mathbf{x}(t) \equiv \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), \quad (3)$$

where $\mathbf{n}(t)$ is noise. Throughout this paper, we will only consider the case where the noise $\mathbf{n}(t)$ is negligible.

Also usual in linear separation is a pre-processing step called whitening, or sphering [16], which usually results in a numerically better conditioned problem. We assume, with no loss of generality, that the data have zero mean. The whitening procedure starts with the computation of the empirical covariance matrix $\mathbf{C} = \langle \mathbf{x}\mathbf{x}^T \rangle$. Then one computes the eigenvector decomposition of \mathbf{C} . One can store the eigenvalues of \mathbf{C} in a diagonal matrix \mathbf{D} in descending order, and the corresponding eigenvectors as the columns of a matrix \mathbf{V} . Whitened data \mathbf{x}_w can be obtained through

$$\mathbf{x}_w(t) \equiv \mathbf{D}^{-1/2} \mathbf{V}^T \mathbf{x}(t). \quad (4)$$

We assume that we have a reference signal, $u(t)$, with its phase denoted by $\psi(t)$. We define the estimated source as $y(t) \equiv \mathbf{w}^T \mathbf{x}(t)$, denoting its phase by $\phi(t)$, with the dephasing defined as $\Delta\phi(t) \equiv \phi(t) - \psi(t)$. We can now compute the absolute value of the PLF between the estimated source and the reference signal,

$$|\varrho| = \left| \frac{1}{T} \sum_{t=1}^T e^{i\Delta\phi(t)} \right| = \left| \frac{1}{T} \sum_{t=1}^T \frac{\tilde{y}(t)\tilde{u}^*(t)}{|\tilde{y}(t)\tilde{u}(t)|} \right| = \left| \left\langle \frac{\tilde{y}(t)\tilde{u}^*(t)}{|\tilde{y}(t)\tilde{u}(t)|} \right\rangle \right|, \quad (5)$$

where \tilde{y} and \tilde{u} are the analytic signals of y and u , obtained through the Hilbert transform.

The idea of this algorithm is to maximize $|\varrho|$, or equivalently $|\varrho|^2$, with respect to \mathbf{w} . The global maximizer, denoted by \mathbf{w}_{opt} , will be the linear combination maximizing the synchrony between the estimated source $y(t) = \mathbf{w}_{\text{opt}}^T \mathbf{x}(t)$ and the reference signal. We want to find $y(t)$ only up to scale and sign, and therefore we constrain \mathbf{w} to have unit norm.

The gradient of $|\varrho|^2$ w.r.t. \mathbf{w} is given by [42]

$$\nabla |\varrho|^2 = 2|\varrho| \left\langle \frac{\sin[\Phi - \Delta\phi(t)]}{Y^2(t)} \mathbf{\Gamma}_x(t) \right\rangle \mathbf{w}, \quad (6)$$

where $Y(t) \equiv |\tilde{y}(t)|$ is the amplitude of the estimated source, $\Phi \equiv \text{angle}(\varrho)$ is the phase of the PLF, and $\mathbf{\Gamma}_x(t) = \mathbf{x}_h(t)\mathbf{x}^T(t) - \mathbf{x}(t)\mathbf{x}_h^T(t)$ (where $\mathbf{x}_h(t)$ is the Hilbert transform of $\mathbf{x}(t)$) is a matrix that can be pre-computed, because it depends only on the data. Since $\mathbf{x}(t) = \text{Re}(\tilde{\mathbf{x}}(t))$ and $\mathbf{x}_h(t) = \text{Im}(\tilde{\mathbf{x}}(t))$, it can easily be seen that $\mathbf{\Gamma}_{x_{ij}}(t) = X_i(t)X_j(t)\sin(\varphi_i(t) - \varphi_j(t))$, where $X_i(t)$ and $\varphi_i(t)$ are the amplitude and phase of $x_i(t)$.

There are many procedures to find the maximizer of a given objective function. We chose to use a gradient ascent algorithm with adaptive step sizes due to its simplicity. A step-by-step description of RPA is presented in Table I.

If the global maximizer is found, RPA outputs the linear combination of the data that is maximally phase-locked to the reference. If two sources are maximally phase-locked to the reference, there are two correct solutions for the problem, and the algorithm will output one or the other, depending on the initialization of the weight vector \mathbf{w} . However, the algorithm never outputs linear combinations of the two solutions, because such mixtures have a lower PLF with the reference.⁵

⁵An exceptional case is when the two sources have a phase lag of exactly 0 or π between themselves, in which case linear combinations of those two sources have the same PLF with the reference.

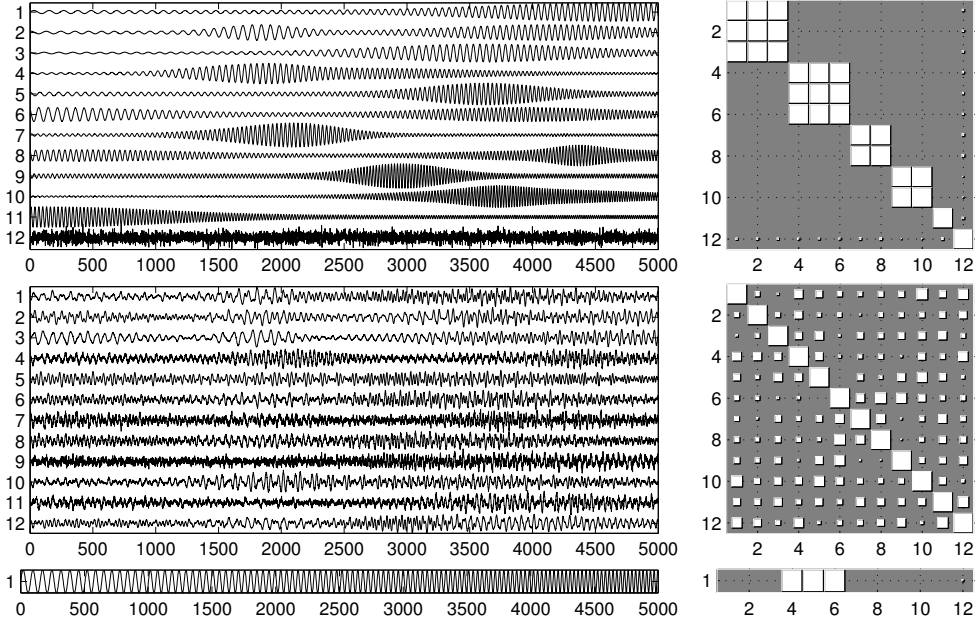


Fig. 1. The dataset used throughout this paper. (First row) Original sources (left) and PLFs between them (right). (Second row) Mixed signals and PLFs between them. (Third row) Reference signal (left), which is a sinusoid with varying frequency, and its PLF with the 12 sources (right). The reference is phase-locked with sources 4, 5 and 6. In the second column, the numbers denote the index of the sources, and the area of each square is proportional to the absolute value of their pairwise PLF.

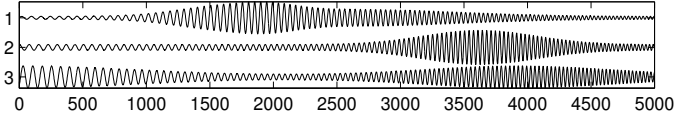


Fig. 2. Results of the RPA algorithm. The algorithm correctly finds the three sources phase-locked to the reference signal. PLF between the found sources and reference: 0.999 (chosen as stopping criterion). These three sources were found on three separate runs of the algorithm.

1) *Application to simulated data:* We applied the above described algorithm to a set of noiseless simulated data. The data set's sources have unit variance and varying instantaneous frequency (see first row of Fig. 1). The reference signal, also shown in Fig. 1, is a sinusoid with the same instantaneous frequency as the fourth, fifth and sixth source signals, with exactly zero phase-lag with the fourth source.

These sources were mixed without noise, using a random mixing matrix, with entries uniformly distributed between -1 and 1 . The algorithm was then run on the mixed signals, which are shown on the second row of Fig. 1. Depending on the initial conditions, RPA will return one of the three solutions shown on Fig. 2 each time it is run. The results in Fig. 2 show that despite having three sources phase-locked to the reference (sources 4, 5 and 6), the algorithm returns one of them separately without mixing them.

These results show that RPA works as expected in the ideal, noiseless condition.

B. Independent Phase Analysis

In this section, we introduce the Independent Phase Analysis algorithm. This algorithm separates sources that are organized in subspaces, such that the intra-subspace PLFs are one and

the inter-subspace PLFs are zero. The general idea is similar to Independent Subspace Analysis [43]. IPA is a true blind source-separation method, in contrast to RPA which is not blind. The original formulation of IPA was heavily motivated by ICA [30], but it has since been improved [32], [33].

We assume that a set of sources undergoes linear mixing as in Eq. (3), and also assume that the data \mathbf{x} has been whitened as in Eq. (4). Let the estimated sources be denoted by $\mathbf{y}(t) \equiv \mathbf{W}^T \mathbf{x}(t) = \mathbf{W}^T \mathbf{A} \mathbf{s}(t)$. Our goal is to find \mathbf{W} such that $\mathbf{W}^T \mathbf{A}$ is a permutation of a diagonal matrix, in which case the estimated sources $\mathbf{y}(t)$ are equal to the original sources $\mathbf{s}(t)$ up to permutation, scaling and sign. We will limit ourselves to the case where the number of sources is equal to the number of measured signals. Therefore, \mathbf{W} is a square matrix.

Since the objective of this algorithm is to extract sources that have pairwise PLFs of one or zero, one might initially consider a cost function that is minimized at those values of PLF. That was the motivation behind our first attempts to solve this problem, which met some success [30], [32]. Unfortunately, that approach gives poor results when the mixing matrix \mathbf{A} is far from orthogonal. This limitation is due to the following fact: given three phase-locked signals s_1, s_2 and s_3 , it is sometimes possible to construct a linear combination $s_4(t) = as_1 + bs_2 + cs_3$ such that the PLFs between s_4 and s_1, s_2 and s_3 are zero. A simple example, illustrated in Fig. 3, is:

$$\begin{aligned}\tilde{s}_1(t) &= (2 + \cos(\omega_2 t))e^{i\omega_1(t)t} \\ \tilde{s}_2(t) &= (2 + \sin(\omega_2 t))e^{i\omega_1(t)t + \pi/2} \\ \tilde{s}_3(t) &= 2\sqrt{2}e^{i\omega_1(t)t - 3\pi/4} \\ \tilde{s}_4(t) &= \tilde{s}_1(t) + \tilde{s}_2(t) + \tilde{s}_3(t) = e^{i(\omega_1(t)t + \omega_2 t)}.\end{aligned}$$

The above mentioned hypothetical cost function would be

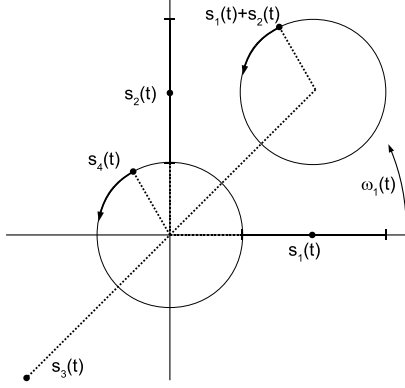


Fig. 3. Three signals, s_1 , s_2 , and s_3 , that are fully phase-locked (PLF = 1), and a linear combination of them, $s_4 = s_1 + s_2 + s_3$, that has zero PLF with all of them. The picture shows the four signals in a frame rotating with angular speed $\omega_1(t)$. The PLF between s_4 and any of the other three sources is zero since the average angle between s_4 and any other source is zero.

unable to distinguish between the correct subspace $\{s_1, s_2, s_3\}$ and an incorrect subspace such as $\{s_1, s_2, s_4\}$. In more formal terms, this problem formulation is ill-posed. To prevent this ill-posedness, we have recently proposed that the full problem be divided into two subproblems [33]: first, separate the subspaces from one another, even if some mixing remains within each subspace. Second, unmix the sources within each subspace, one subspace at a time. The key idea is that if the first subproblem is successful, the second part needs only separate sources that have PLFs equal to one, therefore avoiding the above mentioned ill-posedness. We now discuss each of these subproblems in detail.

1) *Inter-subspace separation and subspace detection*: This first subproblem aims to find an unmixing matrix \mathbf{W} such that the estimated subspaces are correct. We assume that signals that are in distinct subspaces will have little interaction with each other, which should usually correspond to different dynamics in time. Techniques using temporal structure to perform separation should be adequate to this first subproblem. Therefore, we chose to use Ziehe *et al.*'s implementation of TDSEP [17] for this first subproblem, but SOBI [18] can be used instead. Although we have no theoretical results that support TDSEP's adequacy to this task, we have repeatedly observed that it separates subspaces of various sizes quite well.

Detecting the subspace structure (which signals belong to which subspace) from the results of TDSEP is not trivial and warrants some discussion. From our experience, TDSEP can perform the inter-separation very well, but cannot adequately do the intra-subspace separation. This means that PLF values within each subspace will be underestimated. One (admittedly crude) solution for this step is to define a hard PLF threshold, above which signals are considered synchronized and form part of the same subspace, and below which they do not.

The matrix resulting from this hard thresholding should be block-diagonal, with each block having all elements equal to 1. If this is the case, no inconsistencies were found (i.e. no signal belongs to two subspaces), and we can move on to the second subproblem which separates the sources within each subspace (see Section III-B2). Let the unmixing matrix estimated by

TDSEP be denoted by \mathbf{W}_{tdsep} .

If the matrix resulting from the thresholding is not block-diagonal with all blocks filled with ones, our algorithm considers that the subspaces were wrongly detected and returns the results of TDSEP. See Sec. IV for possible improvements.

2) *Intra-subspace separation*: In the second stage of IPA, we select the subset of columns of \mathbf{W}_{tdsep} that form the l -th subspace, which is denoted by S_l , and concatenate them into a rectangular matrix $\mathbf{W}_{tdsep,l}$. Let N_l denote the number of signals in S_l and let $\mathbf{z}_l(t) = \mathbf{W}_{tdsep,l}^T \mathbf{x}(t)$ be the vector of sources in S_l estimated by TDSEP.

In this second stage, the goal is to separate the sources in S_l . This procedure is then repeated for all subspaces found by TDSEP. As explained above, each source in S_l should have a PLF of one with all other sources in S_l . Generally, TDSEP will severely underestimate this value (see top row of Fig 4). We should therefore unmix the N_l sources found by TDSEP such that their PLFs are as high as possible. Mathematically, this corresponds to finding a N_l by N_l matrix \mathbf{W}_l such that the estimated sources in the l -th subspace, $\mathbf{y}_l(t) = \mathbf{W}_l^T \mathbf{z}_l(t) = \mathbf{W}_l^T \mathbf{W}_{tdsep,l}^T \mathbf{x}(t)$, have the highest possible PLFs.

It turns out that if all the sources are in the same subspace and they are all fully synchronized with each other (PLF = 1), then the only vector of the form $\mathbf{y} = \mathbf{B}\mathbf{s}$ such that the estimated sources \mathbf{y} have a PLF of 1 with each other is $\mathbf{y} = \mathbf{s}$ up to permutation, scaling, and sign (see Appendix A). An immediate corollary is that if the inter-subspace separation can fully separate the subspaces from one another, the intra-subspace separation has a unique solution.

For each subspace l , the objective function to maximize is

$$J_l = \frac{1 - \lambda}{N_l} \sum_{j,k \in S_l} |\varrho_{jk}|^2 + \lambda \log |\det \mathbf{W}_l|, \quad (7)$$

where $\sum_{j,k \in S_l} |\varrho_{jk}|^2$ is the sum over all pairs of sources in subspace S_l of the square of the absolute PLF between those sources. We use the Hilbert Transform [38] to obtain the phase values of the estimated sources (see Sec. II). The second term prevents the algorithm from finding solutions which trivially have $|\varrho_{jk}| = 1$, i.e., "finding the same source twice", and $\lambda \in [0, 1]$ is a parameter controlling the relative weight of the two terms. Each column of \mathbf{W}_l is constrained to have unit norm to prevent trivial decreases of the penalty term.

Using this formulation, the problem is no longer ill-posed as above, since we no longer consider a PLF of zero as a valid solution. Furthermore, only a subset of parameters need be optimized at a time, which can drastically reduce the time needed to separate a set of sources.

The gradient of J_l relative to a column \mathbf{w}_j of the matrix \mathbf{W}_l is given by [42] (we omit dependences on l for clarity)

$$\begin{aligned} \nabla_{\mathbf{w}_j} J_l &= \\ &= 4 \frac{1 - \lambda}{N^2} \sum_{k=1}^N |\varrho_{jk}| \left\langle \sin [\Psi_{jk} - \Delta\phi_{jk}(t)] \frac{\mathbf{r}_z(t)}{Y_j(t)^2} \right\rangle \mathbf{w}_j, \quad (8) \end{aligned}$$

where ϱ_{jk} is the PLF between two estimated sources j and k , $Y_j \equiv |\tilde{y}_j|$ where \tilde{y}_j is the analytic signal of the j -th estimated source, $\phi_j \equiv \text{angle}(\tilde{y}_j)$ and $\Delta\phi_{jk}(t) \equiv \phi_j(t) - \phi_k(t)$ is the instantaneous phase difference between two estimated sources,

$\Psi_{jk}(t) \equiv \langle \Delta\phi_{jk}(t) \rangle$ is the average phase difference between two estimated sources, and $\mathbf{\Gamma}_z(t) = \mathbf{z}_{hl}(t)\mathbf{z}_l^T(t) - \mathbf{z}_l(t)\mathbf{z}_{hl}^T(t)$, as in RPA, is a matrix that can be pre-computed, because it depends only on the data resulting from TDSEP.

3) *Application to simulated data:* We present results that show that IPA can successfully separate sets of sources with non-trivial subspace structure. After running TDSEP, we optimize the objective function in Eq. (7) using a gradient ascent algorithm with adaptive step sizes, which runs until a convergence criterion is met. The parameter λ was hand tuned for optimal performance. However, similar results are obtained for λ within a factor of 2 of the optimal value, which in this case is $\lambda = 0.1$. We simulate the noiseless instantaneous linear mixture of the 12 sources depicted in the first row of Fig. 1. These sources are organized in 6 clusters of sizes 3, 2 and 1. We generate 300 mixing matrices with random entries taken from the Uniform(-1,1) distribution, and use those to generate 300 different datasets. We then run the algorithm once for each mixing matrix, for a total of 300 runs. Each run takes about 1 minute on a modern laptop computer.

The second row of Fig. 1 shows the mixed signals which are the input to our algorithm. The first row of Fig. 4 shows the sources estimated by TDSEP. Visual inspection of the PLFs between these sources (second column) shows that some of the estimated sources do not have high PLFs. Inspection of the product $\mathbf{W}_{tdsep}^T \mathbf{A}$ (third column) reveals a very good inter-subspace separation, but a poor intra-subspace separation. Finally, the second row of Fig. 4 reveals that by maximizing the intra-subspace PLFs we can significantly improve the separation within each subspace. This is best noted through the product $\mathbf{W}^T \mathbf{A}$ depicted on the third column.

We analyze the results of IPA using the Amari Performance Index (API) [44], which measures the average relative contamination in each estimated source from all other sources. The API is non-negative and decreases to zero as the separation quality improves. A histogram of the API for the 300 runs of IPA is shown on Fig. 5. The mode of this histogram is 0.03, which is also the API of the example in Fig. 5.

We also measured the performance of TDSEP and IPA using the well-known signal-to-noise (SNR) criterion. After matching the estimated sources to the true sources, we compute the SNR of the sources (in energy) estimated by TDSEP and by IPA.⁶ The average SNR is 16.78 dB for TDSEP and 24.18 dB for IPA, which shows that IPA yields an average increase of 7.4 dB of the estimated sources relative to TDSEP alone. Furthermore, the histograms in Fig. 6 show that the SNR distribution of TDSEP is skewed toward low values, while the values from IPA have much less skewness. In other words, IPA has a much lower probability of yielding very low SNR values (below, say, 10 dB).

We used a threshold of 0.1 on the squared PLF matrix for the detection of subspaces. In 7% of the runs, the matrix resulting from TDSEP and the thresholding is not block-diagonal with blocks full of ones, and therefore the algorithm stops. See Sec. IV for a discussion of this limitation.

⁶See Sec. IV for more details on the SNR calculation.

C. Phase Synchronization Cluster Analysis

In Phase Synchronization Cluster Analysis (pSCA) we are interested in clustering a population of oscillators into synchronous clusters. Contrary to RPA and IPA, here we assume that the sources are not mixed. It is important to remark that in SCA and pSCA no whitening is performed.

We begin by reviewing the original Synchronization Cluster Analysis (SCA) recently proposed by Allefeld *et al.* [27] to automatically cluster a number of oscillators into subpopulations. We empirically show that, for signals that follow the Kuramoto model, SCA can sometimes cluster oscillators incorrectly. We will show that under this model the absolute value of the PLF is not enough to determine whether two signals are coupled and introduce pSCA as an improved method that can perform a correct clustering in some of those situations.

1) *Synchronization Cluster Analysis:* SCA can be seen as a clustering method, which attempts to cluster the original set of oscillators into sub-populations such that the intra-population PLFs are high but the inter-population PLFs are low.

Synchronization Cluster Analysis begins with the construction of the real-valued synchronization matrix \mathbf{R} :

$$\mathbf{R}_{ij} \equiv |\mathbf{Q}_{ij}| = \left| \left\langle e^{i(\phi_i - \phi_j)} \right\rangle \right|.$$

Note that this real-valued matrix is the elementwise absolute value of the complex synchronization matrix \mathbf{Q} defined in Eq. (2). The matrix \mathbf{R} is symmetric and its elements can only take real values between 0 and 1.

The main step in SCA consists of an eigendecomposition of \mathbf{R} . As shown in [27], the eigenvalues of this matrix are real and non-negative, and their sum obeys $\text{tr}(\mathbf{R}) = N$, where N is the total number of oscillators. The largest eigenvalue of \mathbf{R} , λ_1 , is an estimate of the strength of the largest cluster, and the associated eigenvector \mathbf{v}_1 has in its j -th component the relative participation of oscillator j in cluster 1. Similar properties hold for the remaining eigenvalues and eigenvectors. The participation index of oscillator j in cluster k is $\lambda_k v_{jk}^2$, where v_{jk} is the j -th entry of the k -th eigenvector \mathbf{v}_k [27]. Each oscillator is attributed to the cluster with which it has the highest participation index.

Although for many situations SCA works as expected [27], we now present an example showing that it can produce incorrect results. The dataset is depicted in Fig. 7. We simulate the phase of the oscillators using Kuramoto's model (Eq. (1)). The intrinsic frequencies were drawn randomly from a Gaussian with average value 0.003 and standard deviation 0.0005 (negative values are discarded and sampled again). The initial phase values are taken from a uniform distribution in $[0, 2\pi]$. Oscillators 1-5 are coupled in an "all-with-all" fashion, as are oscillators 6-8. There is zero coupling between these two sub-populations. The coupling strength is strong enough to ensure that, in each cluster, all the oscillators are phase-locked to their respective clusters. The time series of all oscillators is shown in Fig. 7, along with their pairwise PLFs, and their relative phase values. When applied to this dataset, SCA clusters all eight oscillators into one cluster, despite finding two eigenvalues larger than 1 (6.35 and 1.64). Clearly, in this situation the real-valued PLF matrix \mathbf{R} does

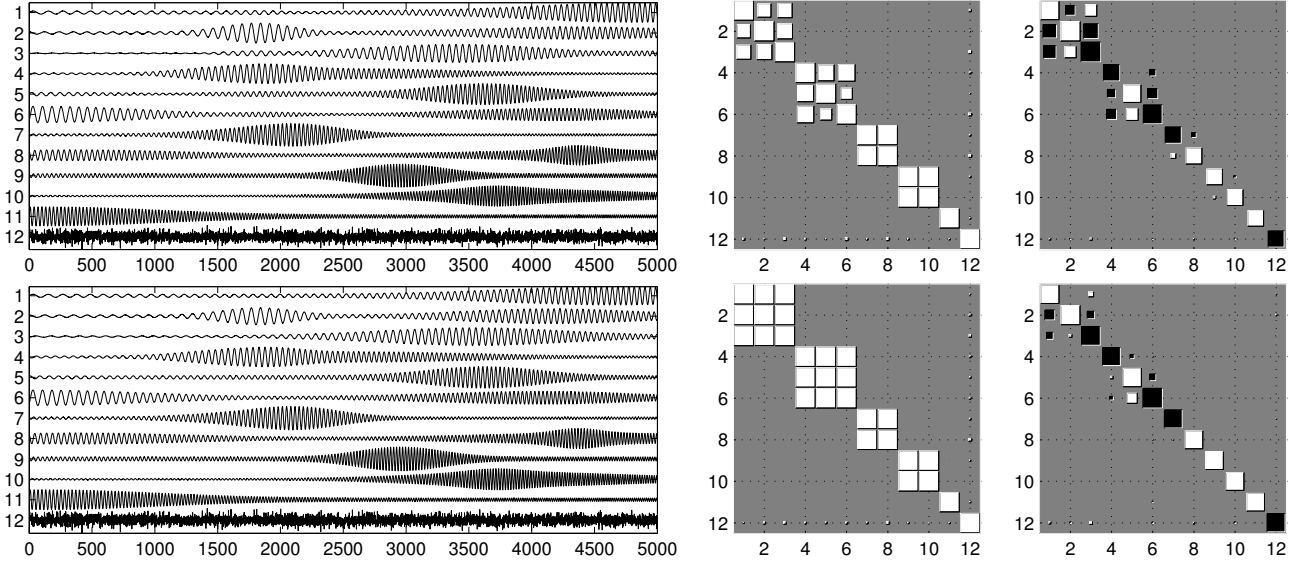


Fig. 4. (First row) Sources resulting from TDSEP (left). Note that the inter-subspace PLFs (middle) are very close to zero, but the intra-subspace PLFs are not all close to 1. Further, the intra-space separation is poor, as can be seen from inspection of the product $\mathbf{W}_{tdsep}^T \mathbf{A}$ (right). (Second row) Results found after the second stage of the algorithm. The estimated sources (left) are very similar to the original ones. This is corroborated by the PLFs between the estimated sources (middle) and the final unmixing matrix (right). The permutation of the sources was corrected manually. White squares represent positive values, while black squares represent negative values.

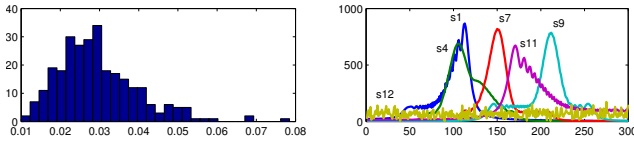


Fig. 5. Histogram of the Amari Performance Index for 300 runs, corresponding to 300 random mixing matrices for these sources (left), and Discrete Fourier Transform of the first, fourth, seventh, ninth, eleventh and twelfth sources, one from each subspace (right). The latter illustrates that simple bandpass filters cannot separate the subspaces.

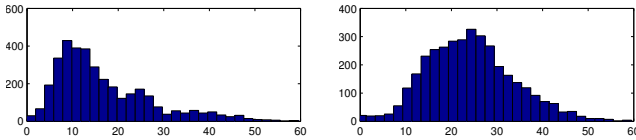


Fig. 6. (Left) Histogram of the signal-to-noise ratio (SNR) between the sources found by TDSEP and the original sources. (Right) Similar histogram for the sources found by IPA.

not contain enough information to perform the clustering. One simple way to aid in this distinction is to look at the relative phase values of the eight oscillators, which are shown in the right subfigure of Fig. 7. In that figure, the relative phase values are clearly clustered into two groups. This immediately suggests that phase values can be used to improve these results.

2) *Limitations of SCA*: Although SCA was not developed with the Kuramoto model in mind, there is nothing specific of this model preventing the application of SCA to oscillators that follow that model. We now demonstrate that high PLF values, which are independent of the relative phase difference between two oscillators, are a necessary but not sufficient condition for two oscillators to be coupled under the Kuramoto model.

If the intra-cluster coupling within cluster c_j is strong enough and the inter-cluster couplings are weak enough to be disregarded, and if all intra-cluster interactions have the

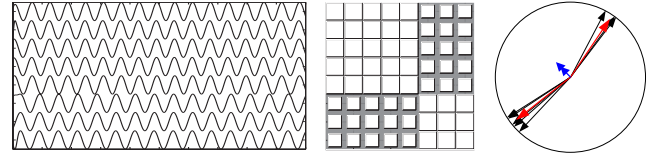


Fig. 7. (Left) Small data set used to illustrate how pSCA works. Oscillators 1-5 are strongly coupled in an “all-with-all” fashion, as are oscillators 6-8. Note the transient behaviour at the beginning of the observation period. (Center) Pairwise PLFs between the oscillators. Note that there is partial synchrony because of the limited time interval. (Right, black thin arrows) Phase values of the eight oscillators at the end of the observation period. (Thick red arrows) Mean-field of each cluster. (Thick blue double arrow) Mean-field of the whole population.

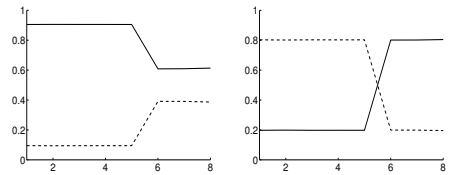


Fig. 8. (Left) Participation indices for each oscillator for SCA. (Right) Participation indices for each oscillator for pSCA. The horizontal axis is the oscillator number. Each oscillator is assigned to the cluster with which it has the highest participation index. SCA fails to find the two clusters.

same coupling strength κ , a mean field can be defined for that cluster as

$$\varrho_{c_j} e^{i\Phi_{c_j}} \equiv \frac{1}{N_j} \sum_{k \in c_j} e^{i\phi_k},$$

where N_j is the number of oscillators in cluster c_j . The mean fields for the above example are shown as thick red arrows in Fig. 7. The meanfield of the whole population of 8 oscillators is represented by the thick blue double arrow. In this case the

original Kuramoto model (1) can be written as

$$\dot{\phi}_i(t) = \omega_i + 2N_j \kappa \varrho_{c_j} \sin[\Phi_{c_j}(t) - \phi_i(t)], \quad (9)$$

for oscillators $i \in c_j$ [42]. Such formulation allows for an interpretation of this case as an interaction between the oscillator and the cluster to which it belongs, instead of the pairwise interactions. This idea was introduced in [27].

In the equilibrium state we have

$$\sin(\phi_i - \Phi_{c_j}) = \left(\frac{\omega_i}{2N_j \kappa \varrho_{c_j}} \right).$$

Equation (9) thus has two equilibrium solutions, one which has $|\phi_i - \Phi_{c_j}| < \frac{\pi}{2}$ and one with $|\phi_i - \Phi_{c_j}| > \frac{\pi}{2}$. The latter is an unstable equilibrium point, because if $|\phi_i - \Phi_{c_j}|$ increases then $|\sin(\phi_i - \Phi_{c_j})|$ will decrease. Therefore, the interaction term in Eq. (9) will be smaller and the oscillator's phase will move further away from the mean field.

A similar reasoning shows that the first solution, with $|\phi_i - \Phi_{c_j}| < \frac{\pi}{2}$, is stable. We can write, for this solution,

$$\phi_i - \Phi_{c_j} = \arcsin\left(\frac{\omega_i}{2N_j \kappa \varrho_{c_j}}\right).$$

This result shows that oscillators for which $|\phi_i - \Phi_{c_j}| > \frac{\pi}{2}$ cannot physically belong to cluster c_j under the Kuramoto model. Graphically, all the oscillators should be contained in a half circle around the meanfield direction. Therefore, a common frequency is a necessary but not sufficient condition for two oscillators to be considered phase-synchronous.

To surpass this limitation, we have developed a more complete approach. Instead of keeping only the real-valued PLF matrix \mathbf{R} , we use the complex PLF matrix \mathbf{Q} to cluster the oscillators. We now describe the procedure in detail.

3) *Phase Synchronization Cluster Analysis*: We present a small theoretical motivation to pSCA, motivated by two small examples. After this we present the general algorithm and show results in simulated datasets.

Example 1: Find the maximum of the average energy $\langle |y|^2 \rangle = \langle yy^* \rangle$ of a signal given by the superposition of two sources with the same frequency: $y = \alpha e^{i\omega t} + \beta e^{i(\omega t + \phi)}$, where $\alpha, \beta \in \mathbb{C}$ are the problem variables. To prevent unboundedness, we add the constraint $|\alpha|^2 + |\beta|^2 = 1$.

The Lagrangian for this problem is $L = \langle |y|^2 \rangle - \lambda(\alpha e^{i\omega t} + \beta e^{i(\omega t + \phi)} - 1)$; the conditions $\frac{\partial L}{\partial \alpha^*} = \frac{\partial L}{\partial \beta^*} = \frac{\partial L}{\partial \lambda^*} = 0$ yield⁷

$$(\lambda - 1)\alpha = \beta e^{i\phi}, \quad (\lambda - 1)\beta = \alpha e^{-i\phi}, \quad |\alpha|^2 + |\beta|^2 = 1.$$

The solutions of these equations are of the form $|\alpha| = |\beta| = 1/\sqrt{2}$ and $\beta = \alpha e^{-i\phi}$, such that $y = \alpha e^{i\omega t} + \beta e^{i(\omega t + \phi)}$ becomes the sum of two signals exactly in phase. Intuitively, we can say that this maximization problem finds the best coefficients to compensate the dephasing of the sources.

Example 2: We now consider three oscillators: the first two with a frequency ω and the third with a frequency $\omega + \Delta\omega$, where $\Delta\omega > 0$, all with the same amplitude and initial phase. Suppose the signals are observed from $t = 0$ to $t = T$.

We again want to maximize the average energy of $y = \alpha e^{i\omega t} + \beta e^{i\omega t} + \gamma e^{i(\omega + \Delta\omega)t}$ subject to the constraint $|\alpha|^2 + |\beta|^2 + |\gamma|^2 = 1$. We can obtain the following two relations:

$$|\alpha| = |\beta|, \quad |\gamma| \leq \frac{4}{\Delta\omega T} |\alpha|$$

(see Appendix B). This shows that, for non-zero α and β , the magnitude of γ will be close to zero if either the observation time T or the frequency detuning $\Delta\omega$ (or both) are large. In particular, it shows that if we have a large enough observation time this maximization problem “detects” that there are two phase-locked sources and a third non-locked source.

Example 1 shows that this maximization procedure can compensate relative phase offsets between oscillators, while example 2 shows that it can select synchronized oscillators from a population which also includes non-synchronized oscillators. We now formulate a general algorithm based on the intuitive ideas provided by these examples. This algorithm has two steps: an eigendecomposition and an additional rotation.

We assume that we have N measurements. We extract their phases $\phi_i(t)$ using the Hilbert transform and construct the normalized signals $a_i(t) = e^{i\phi_i(t)}$, which we use to form the vector $\mathbf{a}(t)$. Then we search for a complex superposition of the form $y(t) = \mathbf{c}^H \mathbf{a}(t)$, which maximizes the average energy $E = \langle |y|^2 \rangle$. To prevent unboundedness we constrain the projection vector \mathbf{c} to have unit norm. This maximization problem is the general case of Examples 1 and 2.

We can rewrite the energy as $E = \langle |y|^2 \rangle = \langle yy^* \rangle = \mathbf{c}^H \langle \mathbf{a} \mathbf{a}^H \rangle \mathbf{c} = \mathbf{c}^H \mathbf{Q} \mathbf{c}$, where $\mathbf{Q} \equiv \langle \mathbf{a} \mathbf{a}^H \rangle$. Note that \mathbf{Q} is exactly the synchronization matrix defined in Eq. (2). The Lagrangian for the maximization of E is thus $L = \mathbf{c}^H \mathbf{Q} \mathbf{c} - \lambda(\mathbf{c}^H \mathbf{c} - 1)$, and the solutions of the problem must obey $\nabla_{\mathbf{c}} L = 0$. If \mathbf{v} is a solution of this problem, it must obey the eigenvalue condition $\mathbf{Q} \mathbf{v} = \lambda \mathbf{v}$.

This reasoning presents additional motivation to the use of the eigendecomposition originally proposed in [27]. We now assume, without loss of generality, that the eigenvalues of \mathbf{Q} are ordered in descending order: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$. Let \mathbf{v}_k denote the eigenvector associated with λ_k , and let v_{jk} denote the j -th coordinate of \mathbf{v}_k . The largest eigenvalue of \mathbf{Q} is the maximum of L , and its corresponding eigenvector \mathbf{v}_1 is thus the solution to the maximization problem above. Since this is a global maximum, the Hessian of L , $\nabla \nabla L = \mathbf{Q} - \lambda \mathbf{I}$, is negative (semi)definite for $\lambda = \lambda_1$.

The value of λ_i is, for noiseless data and infinite observation time, the number of oscillators in the i -th largest population (this is easy to show from Eq. (2)). Thus, $\text{tr}(\mathbf{Q}) = N$, the total number of oscillators. As in [27], we use the number of eigenvalues of \mathbf{Q} larger than 1 as the number of clusters.

Note that the main difference between the eigendecomposition in pSCA and the one done in SCA is that in pSCA we retain the value of the phase lag between any oscillator i and cluster j , which is the angle of the i -th component of \mathbf{v}_j . We now describe how these phase values can be used to overcome the limitations of SCA described in Sec. III-C2.

Assume that the population has N oscillators in d clusters. As mentioned before, we choose d as the number of eigenvalues of \mathbf{Q} larger than 1. Our goal is to find a new

⁷We use the complex gradient as defined in [45].

set of complex-valued coefficients u_{ij} , which contain the participation of oscillator i in cluster j , such that the phase compactness values of all clusters j , defined as $|\sum_{i=1}^N u_{ij}| = |\sum_{i=1}^N |u_{ij}| e^{i\phi_i}|$, are as high as possible. The phase compactness measures the compactness of the phase angles ϕ_i weighted by their participation in cluster j . Note that the sum is over all oscillators, regardless of which cluster they belong to (if oscillator i does not belong to cluster j , then $u_{ij} = 0$ and it will not influence the sums). We define the vectors \mathbf{u}_j , $j = 1, \dots, d$, as column vectors containing the u_{ij} coefficients, similarly to the definition of \mathbf{v}_j . Our objective will be to maximize the sum of the phase compactnesses of all the d clusters:

$$J = \sum_{j=1}^d \left| \sum_{i=1}^N u_{ij} \right|.$$

This criterion can be expressed as “assign these N oscillators to d clusters in such a way that the phase compactness of the clusters is as high as possible”. Regarding the right subfigure of Fig. 7, the goal is to assign the eight oscillators into two clusters such that the red arrows are as big as possible.

To obtain the u_{ij} coefficients we perform a linear transformation of the v_{ij} coefficients obtained from the eigendecomposition, as motivated by Examples 1 and 2. Thus, the new coefficients are given by $|u_{ij}| \equiv |\sum_k v_{ik} w_{kj}|$ where w_{kj} are real coefficients to be optimized through the maximization of the objective function J . If we concatenate the w_{kj} in a matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$, the v_{ij} in $\mathbf{V} \in \mathbb{C}^{N \times d}$ and the u_{ij} in $\mathbf{U} \in \mathbb{C}^{N \times d}$ this linear transformation can be written as $\mathbf{U} = \mathbf{V}\mathbf{W}$.

There are some restrictions on \mathbf{W} . Since we want to measure the compactness of the phase values in the clusters, the coefficients in \mathbf{W} should not compensate for the different phase values (as in Example 1) and should therefore be real. Also, we force \mathbf{W} to be orthonormal so that the total number of oscillators remains equal to N . Mathematically, if $\mathbf{W}^T \mathbf{W} = \mathbf{I}$, then $\text{tr}(\mathbf{U}^H \mathbf{Q} \mathbf{U}) = \text{tr}(\mathbf{W}^T \mathbf{V}^H \mathbf{D} \mathbf{V} \mathbf{W}) = \text{tr}(\mathbf{W}^T \mathbf{Q} \mathbf{W}) = N$, where \mathbf{D} is a diagonal $d \times d$ matrix with only the d largest eigenvectors of \mathbf{Q} in descending order.

The gradient of the objective J with respect to w_{ij} is [42]

$$\mathbf{G}_{ij} = \frac{\partial J}{\partial w_{ij}} = \frac{1}{|\bar{u}_j|} \left[\text{Re}(\bar{v}_i) \times \text{Re}(\bar{u}_j) + \text{Im}(\bar{v}_i) \times \text{Im}(\bar{u}_j) \right] \quad (10)$$

where $\text{Re}(\cdot)$ and $\text{Im}(\cdot)$ are real and imaginary parts, and $\bar{u}_j \equiv \sum_i u_{ij}$ and $\bar{v}_i \equiv \sum_k v_{ki}$ are column sums of \mathbf{U} and \mathbf{V} .

Since \mathbf{W} is constrained to be orthonormal, a simple gradient method is not appropriate to maximize J . The manifold of orthonormal \mathbf{W} matrices is called the Stiefel manifold \mathcal{S} [46]. To maximize J subject to the constraint $\mathbf{W} \in \mathcal{S}$, one possibility is to project the gradient in Eq. (10) onto the tangent space of \mathcal{S} at \mathbf{W} by $\mathbf{G}_{\text{tang}} = \mathbf{W} \text{skew}(\mathbf{W}^T \mathbf{G}) + (\mathbf{I} - \mathbf{W} \mathbf{W}^T) \mathbf{G}$ [46], where \mathbf{G} is a matrix whose (i, j) element is given by Eq. (10), and $\text{skew}(\cdot)$ is the anti-symmetric part of a matrix. Another possibility is to take directly into account the geometry of the Stiefel manifold when computing the gradient. This yields $\mathbf{G}_{\text{geom}} = \mathbf{G} - \mathbf{W} \mathbf{G}^T \mathbf{W}$ [46]. We used \mathbf{G}_{tang} in all the results presented here, due to its simpler interpretation.

The final result of pSCA is a clustering of the oscillators into clusters with tightly packed phase angles. We define the

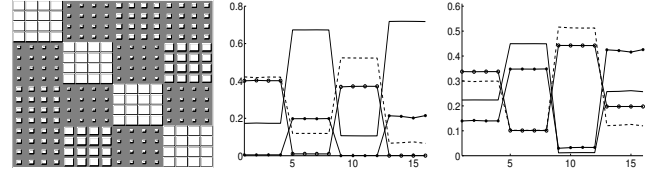


Fig. 9. (Left) Pairwise PLFs between 16 oscillators in a second pSCA dataset. (Center) Results of the original SCA algorithm. SCA detects only two clusters, with 8 oscillators each. (Right) Results of the pSCA algorithm. All the clusters are correctly detected, and with a larger discriminating power.

unnormalized participation index of oscillator i in cluster c_j as $|u_{ij}|$. Each oscillator is assigned to the cluster with which it has the largest unnormalized participation index:

$$i \in c_j \Leftrightarrow j = \arg \max_k \{|u_{ik}|\}. \quad (11)$$

Because of the rotation mentioned above, the eigenvalues of \mathbf{Q} are not necessarily the correct cluster sizes after the optimization of J . Therefore, in pSCA, the unnormalized participation indices for each oscillator do not necessarily sum to one as in SCA. Thus we force this normalization by scaling the participation indices. For each oscillator i , we define the normalized participation indices as

$$p_{ij} = \frac{u_{ij}}{\sum_k |u_{ik}|}. \quad (12)$$

Note that for each oscillator, this normalization is a scaling of its participation in all clusters such that $\sum_j |p_{ij}| = 1$. This scaling does not alter the assignment rule in Eq. (11), i.e., the same rule applied to the normalized participation indices in Eq. (12) yields the same clustering.

The complete pSCA procedure is summarized in Table I.

4) *Application to simulated data:* We now show the result of some simulated experiments. We begin by considering the small example with eight oscillators already mentioned above. We applied SCA and pSCA to this toy dataset for comparison. The only two eigenvalues larger than one are 6.35 and 1.64, indicating the presence of two clusters. Note that the eigenvalues are equal for the eigendecomposition of the real PLF matrix and the complex PLF matrix. The participation indices are shown in Fig. 8 and show that in this example pSCA is capable of distinguishing the two clusters, while SCA is not. One interesting remark is that, although there are two eigenvalues larger than one, SCA assigns all eight oscillators to one cluster. This example shows that pSCA yields better results than SCA for this small dataset.

We also considered a larger example, with 16 oscillators organized into 4 clusters. The oscillators are simulated in the same way: we use Euler’s method (Eq. (1)) to simulate the dynamics of the population, and all oscillators start with a random phase value between 0 and 2π . The only four eigenvalues larger than 1 are 6.55, 4.78, 3.07 and 1.59. The pairwise PLFs and participation indices are shown in Fig. 9.

These results show that pSCA can use the additional phase information to improve the results of the original SCA algorithm. Despite finding 4 eigenvalues larger than 1 (the criterion mentioned in [27] for presence of a cluster), the original SCA

fails to find four distinct clusters, instead finding two clusters with double the correct number of oscillators.

IV. DISCUSSION

As mentioned earlier, observed signals are often mixtures of underlying sources. This mixing process has a serious influence on phase-locking relations, and should be inverted as much as possible when one analyzes the synchrony of such signals. In the absence of prior knowledge of the sources, often one uses blind source separation (BSS) approaches. One of the most widely used BSS techniques is ICA, which works by finding linear combinations of the measurements that are as independent as possible. ICA uses information from the probability density function of the data, and therefore discards all temporal dynamics information, rendering it useless for synchrony studies (see Fig. 10).

Other BSS approaches such as TDSEP are based on the temporal structure of the sources, and therefore take into account their temporal dynamics. The first row of Fig. 4 already shows that TDSEP can separate subspaces from each other but fails to separate sources within each subspace.

One important aspect to study is the robustness of the proposed algorithms to small levels of noise, especially since the influence of e.g. Gaussian noise on the phase values is quite complex.⁸ We present in Fig. 11 results that show that all the algorithms are robust to small levels of noise. We define the SNR of the i -th mixture signal $x_i = (\mathbf{A}\mathbf{s})_i + n_i$ as $10 \log_{10} \frac{E[(\mathbf{A}\mathbf{s})_i^2]}{E[n_i^2]}$. To produce the noisy mixture, we first generate the noiseless mixture and then add noise to each mixture signal to ensure that all mixture signals have the same SNR. We generated 300 data sets for RPA and IPA similar to the one shown in Fig. 1, and 300 data sets for pSCA similar to the one in Fig. 9, but with amplitude bursts of random position and width as in RPA and IPA.⁹

After matching the estimated sources with the original sources, the SNR of the estimated source y_i is defined as $10 \log_{10} \frac{E[(\alpha y_i)^2]}{E[(\alpha y_i \pm s)^2]}$, where the real scalar α and the \pm sign are chosen to maximize the SNR value. This ensures that these SNR values are independent of permutation, scaling, and sign, as is common in source separation contexts.

Much in line with other methods dealing with analysis of phase-locking relations, one limitation of methods presented in this paper is the use of the Hilbert transform, which only has a clear interpretation when applied to locally narrow-band signals. Note that since a wavelet decomposition can be interpreted as a narrowband filter bank [37], using wavelets does not solve this problem. All the signals generated for our simulated examples are narrow-band, therefore the use of the Hilbert transform is justified.

One could consider using a deflationary version of RPA to extract multiple sources with a single run of the algorithm. This cannot be done simply by removing the orthogonal projection of the first estimated source from the data and

re-running the algorithm. Since phase-locked sources may be correlated, removing the orthogonal projection of one estimated source from the data will also remove parts of other sources. Therefore, a deflationary approach is currently not possible, and if multiple sources are desired RPA must be run multiple times with different initial conditions.

IPA is quite robust to the choice of λ : values within a factor of two from the optimal value yield similar results.

We observed that pSCA does not always find the correct solution, especially in cases with multiple populations with equal sizes, where the numerical computation of the eigenvalues is less accurate. Yet, we found that not only does pSCA consistently yield better clustering results than SCA, it usually has a larger distance between the two largest participation factors (see, for example, oscillators 1 to 4 in Fig. 9). This suggests that pSCA has a larger “discriminant power” than SCA, which explains pSCA’s higher tolerance to noise.

Another aspect to be analyzed further is the performance of these algorithms in situations of partial synchrony. Preliminary tests suggest that the algorithms behave differently in this situation. RPA performs quite well when the reference is only partially phase-locked to some of the sources, as long as the stopping criteria are well chosen. IPA performs well for phase-locking values above approximately 0.9 or below approximately 0.1. pSCA seems to work well for intermediate values of the PLF, as suggested by the results obtained on the small dataset in Fig. 8. A related aspect to be investigated in the future is how to tackle a time-dependent PLF, which is relevant in some real situations. In this case, using time windows allows the algorithms to look only at parts of the data, but further work would be needed to implement a change detection technique to track these time variations.

Apart from the results with simulated data, we have previously applied RPA to real data from cortico-muscle control using EMG as a reference for MEG recordings [29]. Those results agreed with established knowledge on relations between cortical and muscle activity, suggesting that the use of the methods we propose here in real applications is possible.

V. CONCLUSION

Synchrony phenomena are present in many scientific domains, including astrophysics, biology, chemistry, and neuroscience. Even though many different systems exhibit synchrony, a common framework, such as Kuramoto’s model, can be used for all of them. Whichever the underlying physical system may be, in any synchrony phenomena it is important to have direct access to the underlying oscillators. If that is not possible or feasible, as in brain electrophysiological recordings, it is crucial that one looks at the sources and not at the indirect measurements as these will have spurious synchrony. Unfortunately, few researchers in neuroscience perform synchrony (or coherence) analysis on the underlying sources instead of on the measurements.

There is a need for source separation methods that tackle this type of problems. We showed empirically that current state-of-the-art techniques such as ICA and TD methods are not adequate. Since these methods make assumptions that are

⁸This noise would be projected into a phase component and an amplitude component, each of which has different impacts. See, e.g., [47].

⁹The position and width of the amplitude bursts, as well as the mixing matrices (for RPA and IPA), are chosen randomly for each of the 300 runs.

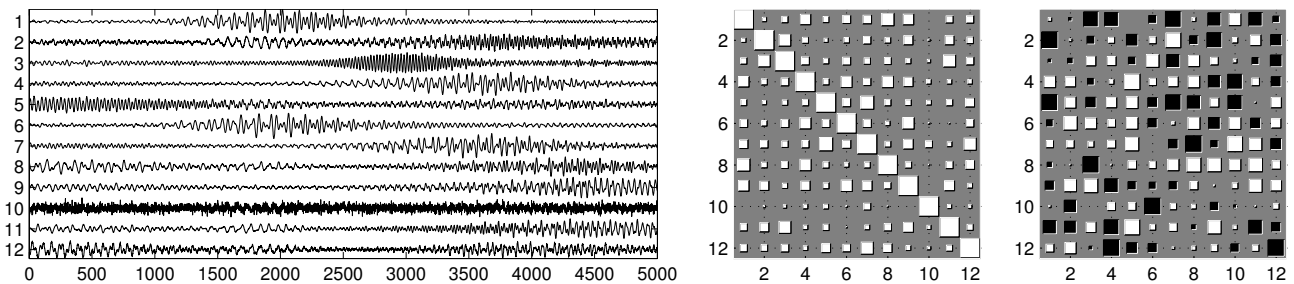


Fig. 10. Result of FastICA applied to the dataset of Fig. 1. (Left) Sources estimated by FastICA. (Center) PLFs between the estimated sources. (Right) Estimated unmixing matrix. It is clear that FastICA is not adequate to the problem discussed in this paper.

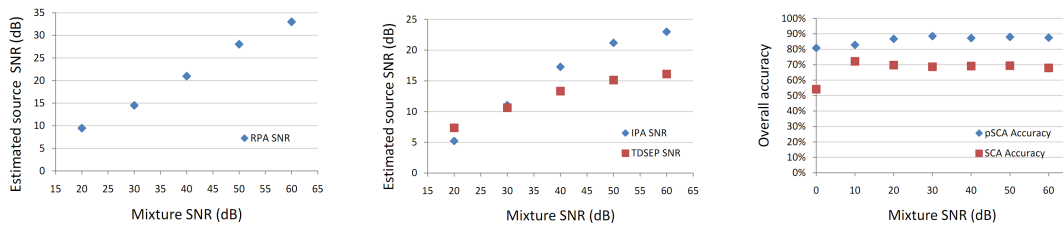


Fig. 11. Results with noise for RPA (left), IPA and TDSEP (middle) and pSCA and SCA (right).

not valid when the underlying sources are phase-locked, synchrony between sources extracted with these methods should be regarded as an epiphenomenon.

We have presented three methods deeply rooted in the concept of phase synchrony which can be used in different circumstances. The methods are inspired by concepts from blind source separation and attempt to circumvent the disruption of synchrony values in the event of a linear mixing. RPA is used to extract an underlying source phase-locked to a reference signal. IPA performs blind source-separation of phase-locked sources. pSCA clusters a population of oscillators into synchronous subpopulations. The three methods work for simulated data and are robust to small levels of noise. We believe these algorithms provide the community with methods applicable in situations which were not addressable previously.

ACKNOWLEDGMENTS

M. Almeida was partially funded by the Portuguese Science and Technology Foundation, grant SFRH/BD/28834/2006. This work was partially funded by project P01075 of the Institute of Telecommunications, and by the Academy of Finland through its Centres of Excellence Program 2006-2011.

REFERENCES

- [1] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A universal concept in nonlinear sciences*, ser. Cambridge Nonlinear Science Series. Cambridge University Press, 2001.
- [2] B. A. Conway, D. M. Halliday, S. F. Farmer, U. Shahani, P. Maas, A. I. Weir, and J. R. Rosenberg, "Synchronization between motor cortex and spinal motoneuronal pool during the performance of a maintained motor task in man," *Journal of Physiology*, vol. 489, pp. 917–924, 1995.
- [3] S. Salenius, K. Portin, M. Kajola, R. Salmelin, and R. Hari, "Cortical control of human motoneuron firing during isometric contraction," *Journal of Neurophysiology*, vol. 77, pp. 3401–3405, 1997.
- [4] J. M. Kilner, S. N. Baker, S. Salenius, V. Jousmäki, R. Hari, and R. N. Lemon, "Task-dependent modulation of 15–30 Hz coherence between rectified EMGs from human hand and forearm muscles," *Journal of Physiology*, vol. 516, pp. 559–570, 1999.
- [5] J.-M. Schoffelen, R. Oostenveld, and P. Fries, "Imaging the human motor system's beta-band synchronization during isometric contraction," *NeuroImage*, vol. 41, pp. 437–447, 2008.
- [6] W. Omlor, L. Patino, M. Hepp-Raymond, and R. Kristeva, "Gamma-range corticomuscular coherence during dynamic force output," *NeuroImage*, vol. 34, pp. 1191–1198, 2007.
- [7] A. Andrykiewicz, L. Patino, J. R. Naranjo, M. Witte, M. Hepp-Raymond, and R. Kristeva, "Corticomuscular synchronization with small and large dynamic force output," *BMC Neuroscience*, vol. 8, 2007.
- [8] J. M. Palva, S. Palva, and K. Kaila, "Phase synchrony among neuronal oscillations in the human cortex," *Journal of Neuroscience*, vol. 25, no. 15, pp. 3962–3972, April 2005.
- [9] P. J. Uhlhaas and W. Singer, "Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology," *Neuron*, vol. 52, pp. 155–168, Oct 2006.
- [10] S. Strogatz, *Nonlinear Dynamics and Chaos*. Westview Press, 2000.
- [11] Y. Kuramoto, *Chemical Oscillations, Waves and Turbulences*. Springer Berlin, 1984.
- [12] D. Golomb and D. Hansel, "The number of synaptic inputs and the synchrony of large sparse neuronal networks," *Neural Computation*, vol. 12, pp. 1095–1139, 2000.
- [13] A. N. Burkitt, "A review of the integrate-and-fire neuron model: I. homogeneous synaptic input," *Biological Cybernetics*, vol. 95, pp. 1–19, 2006.
- [14] —, "A review of the integrate-and-fire neuron model: II. inhomogeneous synaptic input and network properties," *Biological Cybernetics*, vol. 95, pp. 97–112, 2006.
- [15] 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 Biom. Eng.*, vol. 47, no. 5, pp. 589–593, May 2000.
- [16] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley & Sons, 2001.
- [17] A. Ziehe and K.-R. Müller, "TDSEP - an efficient algorithm for blind separation using time structure," in *International Conference on Artificial Neural Networks*, 1998, pp. 675–680.
- [18] A. Belouchrani, K. A. Meraim, J.-F. Cardoso, and E. Moulines, "Second-order blind separation of temporally correlated sources," in *Proc. of the Int. Conf. on Digital Signal Processing*, 1993, pp. 346–351.
- [19] G. Zhou, Z. Yang, and S. Xie, "Online blind source separation using incremental nonnegative matrix factorization with volume constraint," *IEEE Trans. on Neural Networks*, vol. 22, pp. 550–560, 2011.
- [20] M. Almeida, R. Vigario, and J. Bioucas-Dias, "Phase locked matrix factorization," in *Proc. of the EUSIPCO conference*, 2011.
- [21] Y. Xiang, S. K. Ng, and V. K. Nguyen, "Blind separation of mutually correlated sources using precoders," *IEEE Trans. on Neural Networks*, vol. 21, pp. 82–90, 2010.

- [22] R. Vigiário and O. Jensen, "Identifying cortical sources of corticomuscle coherence during bimanual muscle contraction by temporal decorrelation," in *Proceedings of IEEE International Symposium on Signal Processing and Its Applications*, 2003.
- [23] C. Meinecke, A. Ziehe, J. Kurths, and K.-R. Müller, "Measuring phase synchronization of superimposed signals," *Physical Review Letters*, vol. 94, Mar 2005.
- [24] G. Nolte, O. Bai, L. Wheaton, Z. Mari, S. Vorbach, and M. Hallett, "Identifying true brain interaction from EEG data using the imaginary part of coherency," *Clin. Neurophys.*, vol. 115, pp. 2292–2307, 2004.
- [25] L. A. Wheaton, G. Nolte, S. Bohlhalter, E. Fridman, and M. Hallett, "Synchronization of parietal and premotor areas during preparation and execution of praxis hand movements," *Clinical Neurophysiology*, vol. 116, pp. 1382–1390, 2005.
- [26] C. Allefeld and J. Kurths, "An approach to multivariate phase synchronization analysis and its application to event-related potentials," *Int. J. of Bifurcation and Chaos*, vol. 14, no. 2, pp. 417–426, 2004.
- [27] C. Allefeld, M. Müller, and J. Kurths, "Eigenvalue decomposition as a generalized synchronization cluster analysis," *International Journal of Bifurcation and Chaos*, vol. 17, no. 10, pp. 3493–3497, 2007.
- [28] M. G. Rosenblum, A. S. Pikovsky, and J. Kurths, "Phase synchronization of chaotic oscillators," *Physical Review Letters*, vol. 76, no. 11, pp. 1804–1807, Mar 1996.
- [29] J.-H. Schleimer and R. Vigiário, "Reference-based extraction of phase synchronous components," in *Proceedings of the International Conference on Artificial Neural Networks*, 2006.
- [30] —, "Order in complex systems of nonlinear oscillators: Phase locked subspaces," in *Proc. of the Eur. Symp. on Neural Networks*, 2007.
- [31] —, "Clustering limit cycle oscillators by spectral analysis of the synchronisation matrix with an additional phase sensitive rotation," in *Proc. of the Int. Conf. on Artificial Neural Networks*, 2007.
- [32] M. Almeida and R. Vigiário, "Source separation of phase-locked signals," in *Proc. of the Independent Component Analysis Conference*, 2009.
- [33] M. Almeida, J. Bioucas-Dias, and R. Vigiário, "Independent phase analysis: Separating phase-locked subspaces," in *Proceedings of the Latent Variable Analysis Conference*, 2010.
- [34] A. Grinsted, J. C. Moore, and S. Jevrejeva, "Application of the cross wavelet transform and wavelet coherence to geophysical time series," *Nonlinear Processes in Geophysics*, vol. 11, pp. 561–566, 2004.
- [35] S. Bloomfield, J. McAteer, B. W. Lites, P. G. Judge, M. Mathioudakis, and F. P. Keenan, "Wavelet phase coherence analysis: Application to a quiet-sun magnetic element," *The Astrophysical Journal*, vol. 617, pp. 623–632, 2004.
- [36] A. Klein, T. Sauer, A. Jedynak, and W. Skrandies, "Conventional and wavelet coherence applied to sensory-evoked electrical brain activity," *IEEE Transactions on Biomedical Engineering*, 2006.
- [37] C. Torrence and G. P. Compo, "A practical guide to wavelet analysis," *Bull. of the Am. Meteorological Society*, vol. 79, pp. 61–78, 1998.
- [38] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*. Prentice-Hall International Editions, 1999.
- [39] M. L. V. Quyen, J. Foucher, J.-P. Lachaux, E. Rodriguez, A. Lutz, J. Martinerie, and F. J. Varela, "Comparison of hilbert transform and wavelet methods for the analysis of neuronal synchrony," *Journal of Neuroscience Methods*, vol. 111, pp. 83–98, 2001.
- [40] 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., 1973.
- [41] J.-P. Lachaux, E. Rodriguez, J. Martinerie, and F. J. Varela, "Measuring phase synchrony in brain signals," *Human Brain Mapping*, vol. 8, pp. 194–208, 1999.
- [42] M. Almeida, J.-H. Schleimer, J. Bioucas-Dias, and R. Vigiário, (2011) Source separation and clustering of phase-locked subspaces: Derivations and proofs. In arXiv.org. arxiv:1106.2474. [Online]. Available: <http://arxiv.org/abs/1106.2474>
- [43] A. Hyvärinen and P. Hoyer, "Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces," *Neural Computation*, vol. 12, pp. 1705–1720, 2000.
- [44] S. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," *Advances in Neural Information Processing Systems*, vol. 8, pp. 757–763, 1996.
- [45] D. Brandwood, "A complex gradient operator and its application in adaptive array theory," *IEE Proceedings, Part F - Communications, Radar and Signal Processing*, vol. 130, pp. 11–16, 1983.
- [46] A. Edelman, T. Arias, and S. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, pp. 303–353, 1998.
- [47] P. Celka, "Statistical analysis of the phase-locking value," *IEEE Signal Processing Letters*, vol. 14, pp. 577–580, 2007.

APPENDIX A UNICITY OF SOLUTION IN IPA

A. Assumptions and Definitions

Consider a set of N sources, which are defined as complex time-series: $s_k(t) \equiv A_k(t)e^{i(\omega(t)+\phi_k)}$, with k from 1 to N . Also, consider N measured signals, obtained by linear instantaneous mixture of the sources:

$$y_j(t) \equiv b_{j1}s_1(t) + b_{j2}s_2(t) + \dots + b_{jN}s_N(t) \quad (13)$$

Our goal is to prove that to put y_1, y_2, \dots in the form¹⁰:

$$y_j(t) = C_j(t)e^{i(\omega(t)+\alpha_j)}, \quad (14)$$

with $C_i(t)$ non-negative functions and α_i scalar values, one must always have $y_i = Ks_j$ for some i and j . In other words, for each $i = 1, 2, \dots$, exactly one of the coefficients b_{i1}, b_{i2}, \dots must be nonzero, with all others being zero. One also demands, as is natural in source separation, that $y_i \neq y_j$. The matrix \mathbf{B} , whose (i, j) entry is b_{ij} , must therefore be a permutation of a diagonal matrix with nonzero diagonal.

The proposition we are trying to prove is not generally true. However, it is true under mild assumptions, which we list here:

- 1) For all i , $s_i(t), y_i(t) \neq 0$ for some t .
- 2) For all $i \neq j$, $y_i(t) \neq y_j(t)$.
- 3) ϕ_1, ϕ_2, \dots are all distinct modulo π .
- 4) The amplitudes $A_i(t)$ are linearly independent. This means that if one has two linear combinations that obey $\sum_i c_i A_i(t) = \sum_i d_i A_i(t)$, then necessarily one has $c_i = d_i$ for all i .

Assumption 1 immediately rules out the possibility that for some i , all b_{i1}, b_{i2}, \dots are all zero.

B. Proof

We start by proving that y_1 is equal to one of the sources up to scaling and sign. If we define a matrix \mathbf{A} where the (i, t) -th entry is $A_i(t)$, $\mathbf{C} \equiv [C_1(1) \dots C_1(T)]$, $\Phi \equiv [\phi_1 \dots \phi_N]^T$, $\Phi - \alpha_1 = [(\phi_1 - \alpha_1) \dots (\phi_N - \alpha_1)]^T$, and $\mathbf{Z}(\Phi) = [b_{11}e^{i\phi_1} \dots b_{1N}e^{i\phi_N}]$, then by equating Equations (13) and (14), and by eliminating the common term $e^{i\omega(t)}$, the equation for y_1 yields $\mathbf{C}^T e^{i\alpha_1} = \mathbf{A}^T \mathbf{Z}(\Phi) \Leftrightarrow \mathbf{C}^T = \mathbf{A}^T \mathbf{Z}(\Phi - \alpha_1) \Leftrightarrow \mathbf{C}^T = \mathbf{A}^T \mathbf{Z}(\alpha_1 - \Phi)$, where the last equation was obtained by taking the complex conjugate of both sides (note that \mathbf{C} and \mathbf{A} are real). Because of Assumption 4, \mathbf{A} is a full rank matrix, so we can conclude that $\mathbf{Z}(\Phi - \alpha_1) = \mathbf{Z}(\alpha_1 - \Phi)$, which is equivalent to either $\sin(\phi_j - \alpha_1) = 0$ or $b_{1j} = 0$, for all j . Because of Assumption 1, at least one of the b_{1j} must be non-zero. On the other hand, because of Assumption 3, $\sin(\phi_j - \alpha_1) = 0$ is verified for at most one j . Combining these two statements, one can conclude that exactly one of the b_{1j} coefficients is non-zero.

The demonstration for y_2, \dots, y_N is similar: if we force $y_i \neq y_j$ if $i \neq j$ (Assumption 2), this immediately forces that if $b_{1k} \neq 0$, then $b_{mk} = 0$ for all $m \neq 1$, etc.

¹⁰The y_j 's must be of this form to have a PLF of 1 with the sources: the PLF between two signals is 1 if and only if their phase difference is constant.

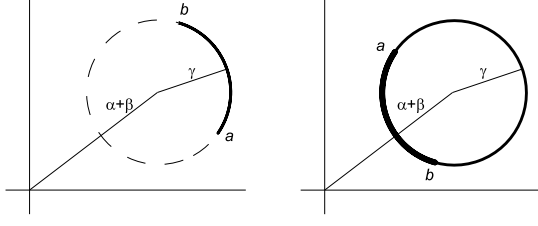


Fig. 12. The integration path in Eq. (15). $a = -T/2$ and $b = +T/2$ represent the end points of the integration. (Left) A case with $\Delta\omega T < \frac{\pi}{2}$. (Right) A case with $\pi < \Delta\omega T < \frac{3\pi}{2}$. Note that the arc between a and b is counted twice in the integration for this second case.

APPENDIX B

THE INEQUALITY IN EXAMPLE 2 OF PSCA

We can begin by choosing the observation period as $[-T/2, T/2]$, with no loss of generality. In this case, the average energy of $y(t)$ is given by the integral

$$E = \langle |y(t)|^2 \rangle = \left\langle |y(t) \cdot e^{-i\omega t}|^2 \right\rangle$$

$$= \frac{1}{T} \int_{-T/2}^{T/2} (\alpha + \beta + \gamma e^{i\Delta\omega t}) (\alpha^* + \beta^* + \gamma^* e^{-i\Delta\omega t}) dt \quad (15)$$

$$= |\alpha + \beta|^2 + |\gamma|^2 + (\Delta\omega T)^{-1} 2 \sin \frac{\Delta\omega T}{2} (\alpha^* + \beta^*) \gamma$$

$$+ (\Delta\omega T)^{-1} 2 \sin \frac{\Delta\omega T}{2} (\alpha + \beta) \gamma^*. \quad (16)$$

This integral is depicted geometrically in Fig. 12.

From Fig. 12 it is clear that the maximum of E will occur when α and β are parallel and γ is either parallel or anti-parallel to α and β . With no loss of generality we can choose α , β and γ to be real numbers. In that case, we can rewrite the constraint $|\alpha|^2 + |\beta|^2 + |\gamma|^2 = 1$ as $\alpha^2 + \beta^2 + \gamma^2 = 1$, and the energy of y as

$$(\alpha + \beta)^2 + \gamma^2 + (\Delta\omega T)^{-1} 4 \sin \frac{\Delta\omega T}{2} (\alpha + \beta) \gamma$$

$$= 1 + 2\alpha\beta + (\Delta\omega T)^{-1} 4 \sin \frac{\Delta\omega T}{2} (\alpha + \beta) \gamma.$$

We can now write the Lagrangean of this problem as

$$L = 1 + 2\alpha\beta + \frac{4 \sin \frac{\Delta\omega T}{2}}{\Delta\omega T} (\alpha + \beta) \gamma - \lambda (\alpha^2 + \beta^2 + \gamma^2 - 1),$$

and force the derivatives with respect to α , β , γ and λ to be zero:

$$\begin{cases} \frac{\partial L}{\partial \alpha} = 2\beta + \frac{4 \sin \frac{\Delta\omega T}{2}}{\Delta\omega T} \gamma - 2\lambda\alpha = 0 \\ \frac{\partial L}{\partial \beta} = 2\alpha + \frac{4 \sin \frac{\Delta\omega T}{2}}{\Delta\omega T} \gamma - 2\lambda\beta = 0 \\ \frac{\partial L}{\partial \gamma} = \frac{4 \sin \frac{\Delta\omega T}{2}}{\Delta\omega T} (\alpha + \beta) - 2\lambda\gamma = 0 \\ \frac{\partial L}{\partial \lambda} = \alpha^2 + \beta^2 + \gamma^2 - 1 = 0. \end{cases} \quad (17)$$

The two first equations in Eq. 17 yield $\lambda = -1 \vee \alpha = \beta$. Let us start with the choice $\lambda = -1$. Substitution in the first or second equation in Eq. (17) shows that $\gamma = 0$ or $\Delta\omega T = 0$. The latter is impossible because we assumed positive detuning and observation time. On the other hand, substituting $\gamma = 0$ in the third equation gives $\alpha = -\beta$. Substitution in Eq. (16) gives $E = 0$, which is clearly a minimum as E is non-negative.

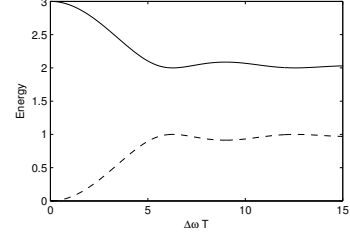


Fig. 13. The two solutions for E given by Eq. (21) plotted against $\Delta\omega T$. (Solid line) The maximum energy, obtained by choosing the plus sign in Eq. (21). (Dashed line) The minimum energy, obtained by choosing the minus sign in Eq. (21).

This leaves us with $\alpha = \beta$. In this case, the system in Eq. (17) can be simplified to

$$\begin{cases} \frac{\partial L}{\partial \alpha} = (1 - \lambda)\alpha + \frac{2 \sin \frac{\Delta\omega T}{2}}{\Delta\omega T} \gamma = 0 \\ \frac{\partial L}{\partial \gamma} = \frac{4 \sin \frac{\Delta\omega T}{2}}{\Delta\omega T} \alpha - \lambda\gamma = 0 \\ \frac{\partial L}{\partial \lambda} = 2\alpha^2 + \gamma^2 - 1 = 0. \end{cases} \quad (18)$$

Because of the third equation in Eq. (18), $\alpha = \gamma = 0$ is not a valid solution. Therefore, the two first equations must be equivalent. This happens if and only if

$$\frac{1 - \lambda}{2A} = -\frac{A}{\lambda} \Leftrightarrow \lambda = \frac{1 \pm \sqrt{1 + 8A^2}}{2}, \quad (19)$$

where $A = \left(\frac{\Delta\omega T}{2}\right)^{-1} \sin \frac{\Delta\omega T}{2}$. Using Eq. (19) in the second equation of Eq. (18) yields

$$\gamma = \frac{4A}{1 \pm \sqrt{1 + 8A^2}} \alpha. \quad (20)$$

Using this relation between α and γ on the last equation in Eq. (18) yields

$$\alpha^2 = \frac{2 + 8A^2 \pm 2\sqrt{1 + 8A^2}}{4 + 32A^2 \pm 4\sqrt{1 + 8A^2}},$$

and direct substitution into Eq. (16) gives us

$$E = 1 + \frac{1 + 4A^2 \pm \sqrt{1 + 8A^2}}{1 \pm \sqrt{1 + 8A^2}}. \quad (21)$$

Choosing the minus sign in Eq. (21) gives a value not greater than 1, while the plus sign gives a value not smaller than 2 (see Fig. 13). Therefore, the plus sign is clearly the maximum of E that we were looking for.

Let us recover Eq. (20) with the plus sign. By taking the absolute value on both sides we obtain

$$|\gamma| = \left| \frac{4A}{1 + \sqrt{1 + 8A^2}} \right| |\alpha|. \quad (22)$$

Since $|A| < 1$, the numerator in Eq. (22) is not greater than $\frac{8}{\Delta\omega T}$ in absolute value, and the absolute value of the denominator is not smaller than 2. We can conclude that

$$|\gamma| \leq \frac{4}{\Delta\omega T} |\alpha|,$$

which is the inequality in Example 2.

APPENDIX C
ALGORITHMS TABLE

REFERENCED PHASE ANALYSIS		INDEPENDENT PHASE ANALYSIS	
1:	Input $\mathbf{x}(t)$, $u(t)$, η , k_{max}	I:	INTER-SUBSPACE SEPARATION
2:	Whiten $\mathbf{x}(t)$ as in Eq. (4)	1:	Input $\mathbf{x}(t)$, ϱ_c
3:	Initialize $\mathbf{w} \sim \mathcal{N}(0, 1)$; $k = 1$	2:	Whiten $\mathbf{x}(t)$ as in Eq. (4)
4:	$\mathbf{\Gamma}(t) \leftarrow \mathbf{x}_h(t)\mathbf{x}^T(t) - \mathbf{x}(t)\mathbf{x}_h^T(t)$	3:	Perform TDSEP on $\mathbf{x}(t)$ as in [17]
5:	repeat	4:	$\mathbf{A}_{tdsep} \leftarrow$ mixing matrix estimated from TDSEP
6:	$y(t) \leftarrow \mathbf{w}^T \mathbf{x}(t)$	5:	$\mathbf{W}_{tdsep} \leftarrow \mathbf{A}_{tdsep}^{-T}$
7:	$\tilde{y} \leftarrow y(t) + i y_h(t)$	6:	$\mathbf{z} \leftarrow \mathbf{W}_{tdsep}^T \mathbf{x}(t)$
8:	$\varrho \leftarrow \frac{1}{T} \sum_t \tilde{y}(t) \tilde{u}^*(t) / \tilde{y}(t) \tilde{u}(t) $	7:	$\tilde{\mathbf{z}} \leftarrow \mathbf{z}(t) + i \mathbf{z}_h(t)$
9:	$\Psi \leftarrow \text{angle}(\varrho)$	8:	$\beta_j(t) \leftarrow \text{angle}(\tilde{z}_j(t))$, $j = 1, \dots, N$
10:	$\Delta \mathbf{w} \leftarrow \text{Eq. (6)}$	9:	$Q_{ij} \leftarrow \left \frac{1}{N} \sum_t e^{i[\beta_i(t) - \beta_j(t)]} \right $ as in Eq. (2)
11:	$\mathbf{w} \leftarrow \mathbf{w} + \eta \Delta \mathbf{w}$	10:	$\mathbf{R} \leftarrow 1$ if $\mathbf{Q} \leq \varrho_c$, 0 otherwise
12:	$\mathbf{w} \leftarrow \mathbf{w} / \ \mathbf{w}\ $	11:	Detect subspaces from block-diagonal structure, as described in Sec. III-B1
13:	$k \leftarrow k + 1$	12:	Construct $\mathbf{W}_{tdsep,l}$ for each subspace l , as described in Sec. III-B1
14:	until ($ \varrho > 1 - \delta$) or ($\ \Delta \mathbf{w}\ < \epsilon$) or ($k > k_{max}$)	13:	$\mathbf{z}_l \leftarrow \mathbf{W}_{tdsep,l} \mathbf{x}(t)$
<hr/>		II:	INTRA-SUBSPACE SEPARATION
PHASE SYNCHRONIZATION CLUSTER ANALYSIS		14:	For each subspace l , do
I: EIGENDECOMPOSITION		15:	Input $\mathbf{z}_l(t)$, η , k_{max}
1:	Input $\mathbf{y}(t)$, η , k_{max}	16:	$\tilde{\mathbf{z}}_l(t) \leftarrow \mathbf{z}_l(t) + i \mathbf{z}_{h_l}(t)$
2:	$\tilde{\mathbf{y}}(t) \leftarrow \mathbf{y}(t) + i \mathbf{y}_h(t)$	17:	$\varphi_j(t) \leftarrow \text{angle}((\tilde{z}_l)_j(t))$, $j = 1, \dots, N$
3:	$\phi_i(t) \leftarrow \text{angle}[\tilde{y}_i(t)]$, $i = 1, \dots, N$	18:	$Z_j(t) \leftarrow ((\tilde{z}_l)_j(t)) $
4:	$\mathbf{Q}_{ij} \leftarrow \sum_t e^{i[\phi_i(t) - \phi_j(t)]}$, $i, j = 1, \dots, N$	19:	Initialize $\mathbf{W} \sim \mathcal{N}(0, 1)$; $k = 1$
5:	$\mathbf{D}, \mathbf{V} \leftarrow \text{trunc. eigendecomp. of } \mathbf{Q}$	20:	repeat
6:	$d \leftarrow \# \lambda > 1$	21:	$\mathbf{y}_l(t) \leftarrow \mathbf{W}^T \mathbf{z}_l(t)$
II: ADDITIONAL ROTATION		22:	$\tilde{\mathbf{y}}_l \leftarrow \mathbf{y}_l(t) + i \mathbf{y}_{h_l}(t)$
7:	Initialize $\mathbf{W} \sim \mathcal{N}(0, 1)$; $k = 1$	23:	$\phi_j(t) \leftarrow \text{angle}((\tilde{y}_l)_j(t))$, $j = 1, \dots, N$
8:	repeat	24:	$Y_j(t) \leftarrow ((\tilde{y}_l)_j(t)) $
8:	$\mathbf{U} = \mathbf{V}\mathbf{W}$	25:	$\Delta \phi_{jk}(t) \leftarrow \phi_j(t) - \phi_k(t)$
10:	$\bar{u}_j \leftarrow \sum_i u_{ij}$, $j = 1, \dots, d$	26:	$\Psi_{jk} \leftarrow \text{angle}(\frac{1}{N} \sum_t e^{i \Delta \phi_{jk}(t)})$
11:	$\bar{v}_i \leftarrow \sum_k v_{ki}$, $i = 1, \dots, d$	27:	$\varrho_{jk} \leftarrow \left \frac{1}{N} \sum_t e^{i \Delta \phi_{jk}(t)} \right $
12:	$\Delta \mathbf{W}_{ij} \leftarrow \text{Eq. (10)}$, $i, j = 1, \dots, N$	28:	$\Delta \mathbf{W} \leftarrow \text{Eq. (8)}$
13:	$\mathbf{W} \leftarrow \mathbf{W} + \eta \Delta \mathbf{W}$	29:	$\mathbf{W} \leftarrow \mathbf{W} + \eta \Delta \mathbf{W}$
14:	$k \leftarrow k + 1$	30:	$\mathbf{w}_j \leftarrow \mathbf{w}_j / \ \mathbf{w}_j\ $, $j = 1, \dots, N$
15:	until ($\ \Delta \mathbf{W}\ < \epsilon$) or ($k > k_{max}$)	31:	$k \leftarrow k + 1$
16:	Assign i to subpop. c_j as in Eq. (11)	32:	until ($\ \Delta \mathbf{W}\ < \epsilon$) or ($k > k_{max}$)
17:	$p_{ij} \leftarrow \text{Eq. (12)}$		

TABLE I

(Top left) $y_h(t)$ is the HILBERT transform of $y(t)$, $\mathbf{x}_h(t)$ is the HILBERT transform of $\mathbf{x}(t)$, $1 - \delta$ is the THRESHOLD for the PLF and ϵ is the THRESHOLD for the GRADIENT of the OBJECTIVE FUNCTION w.r.t. \mathbf{w} . (*) REPRESENTS THE COMPLEX CONJUGATE. (Right) $\mathbf{z}_h(t)$, $\mathbf{z}_{h_l}(t)$ and $\mathbf{y}_{h_l}(t)$ are the HILBERT TRANSFORMS of $\mathbf{z}(t)$, $\mathbf{z}_l(t)$ and $\mathbf{y}_l(t)$, ϱ_c is the PLF THRESHOLD for SUBSPACE DETECTION and ϵ is the THRESHOLD for the GRADIENT of the OBJECTIVE FUNCTION w.r.t. \mathbf{W} . MANY DEPENDENCES ON THE SUBSPACE l WERE OMITTED FOR CLARITY. (Bottom left) $\mathbf{y}_h(t)$ is the HILBERT TRANSFORM of $\mathbf{y}(t)$, $\tilde{y}_i(t)$ is the i -TH ROW OF $\tilde{\mathbf{y}}(t)$ AND ϵ IS THE THRESHOLD for the GRADIENT of the OBJECTIVE w.r.t. \mathbf{W} .

Source Separation and Clustering of Phase-Locked Subspaces: Derivations and Proofs

Miguel Almeida, Jan-Hendrik Schleimer, José Bioucas-Dias, Ricardo Vigário

Abstract—Due to space limitations, our submission “Source Separation and Clustering of Phase-Locked Subspaces”, accepted for publication on the IEEE Transactions on Neural Networks in 2011, presented some results without proof. Those proofs are provided in this paper.

Index Terms—phase-locking, synchrony, source separation, clustering, subspaces

APPENDIX A GRADIENT OF $|\varrho|^2$ IN RPA

In this section we derive that the gradient of $|\varrho|^2$ is given by Eq. 6 of [1], where $|\varrho|$ is defined as in Eq. 5 of [1]. Recall that $\Delta\phi(t) = \phi(t) - \psi(t)$, where $\phi(t)$ is the phase of the estimated source $y(t) = \mathbf{w}^T \mathbf{x}(t)$ and $\psi(t)$ is the phase of the reference $u(t)$. Further, define $\varrho \equiv |\varrho|e^{i\Phi}$.

We begin by noting that $|\varrho|^2 = (|\varrho| \cos(\Phi))^2 + (|\varrho| \sin(\Phi))^2$, so that

$$\begin{aligned} \nabla |\varrho|^2 &= \nabla (|\varrho| \cos(\Phi))^2 + \nabla (|\varrho| \sin(\Phi))^2 \\ &= 2|\varrho| [\cos(\Phi) \nabla (|\varrho| \cos(\Phi)) + \sin(\Phi) \nabla (|\varrho| \sin(\Phi))]. \end{aligned}$$

Note that we have $\frac{1}{T} \sum_{t=1}^T \cos(\Delta\phi(t)) = |\varrho| \cos(\Phi)$ and $\frac{1}{T} \sum_{t=1}^T \sin(\Delta\phi(t)) = |\varrho| \sin(\Phi)$, so we get

$$\begin{aligned} \nabla |\varrho|^2 &= 2|\varrho| \left\{ \cos(\Phi) \nabla \left[\frac{1}{T} \sum_{t=1}^T \cos(\Delta\phi(t)) \right] + \right. \\ &\quad \left. + \sin(\Phi) \nabla \left[\frac{1}{T} \sum_{t=1}^T \sin(\Delta\phi(t)) \right] \right\} \\ &= \frac{2|\varrho|}{T} \sum_{t=1}^T \left[\sin(\Phi) \cos(\Delta\phi(t)) - \cos(\Phi) \sin(\Delta\phi(t)) \right] \times \\ &\quad \times \nabla \Delta\phi(t) \\ &= \frac{2|\varrho|}{T} \sum_{t=1}^T \sin[\Phi - \Delta\phi(t)] \nabla \Delta\phi(t). \end{aligned} \quad (1)$$

Miguel Almeida is with the Institute of Telecommunications, Superior Technical Institute, Portugal. Email: malmeida@lx.it.pt

Jan-Hendrik Schleimer is with the Bernstein Center for Computational Neuroscience, Humboldt University, Berlin. Email: jan-hendrik.schleimer@bccn-berlin.de

José Bioucas-Dias is with the Institute of Telecommunications, Superior Technical Institute, Portugal. Email: bioucas@lx.it.pt

Ricardo Vigário is with the Adaptive Informatics Research Centre, Aalto University, Finland. Email: ricardo.vigario@hut.fi

Let's now take a closer look on $\nabla \Delta\phi(t)$. Note that

$$\begin{aligned} \phi(t) &= \arctan \left(\frac{\mathbf{w}^T \mathbf{x}_h(t)}{\mathbf{w}^T \mathbf{x}(t)} \right) \quad \text{or} \\ \phi(t) &= \arctan \left(\frac{\mathbf{w}^T \mathbf{x}_h(t)}{\mathbf{w}^T \mathbf{x}(t)} \right) + \pi. \end{aligned}$$

Because of this we can say, if $\mathbf{w}^T \mathbf{x}(t) \neq 0$, that $\nabla \phi(t) = \nabla \arctan \left(\frac{\mathbf{w}^T \mathbf{x}_h(t)}{\mathbf{w}^T \mathbf{x}(t)} \right)$. On the other hand, since $\Delta\phi(t) = \phi(t) - \psi(t)$ and $\psi(t)$ does not depend on \mathbf{w} , we have (we will omit the time dependence for the sake of clarity):

$$\begin{aligned} \nabla \Delta\phi &= \nabla \phi - \nabla \psi = \nabla \phi = \nabla \arctan \left(\frac{\mathbf{w}^T \mathbf{x}_h}{\mathbf{w}^T \mathbf{x}} \right) \\ &= \frac{\mathbf{x}_h \cdot \mathbf{w}^T \mathbf{x} - \mathbf{x} \cdot \mathbf{w}^T \mathbf{x}_h}{\left[1 + \left(\frac{\mathbf{w}^T \mathbf{x}_h}{\mathbf{w}^T \mathbf{x}} \right)^2 \right] \cdot (\mathbf{w}^T \mathbf{x})^2} = \frac{\mathbf{I}_x(t) \cdot \mathbf{w}}{Y^2(t)}, \end{aligned}$$

where $Y^2(t) = (\mathbf{w}^T \mathbf{x}(t))^2 + (\mathbf{w}^T \mathbf{x}_h(t))^2$ is the squared magnitude of the estimated source, and $\mathbf{I}_x(t) = \mathbf{x}_h(t) \mathbf{x}^T(t) - \mathbf{x}(t) \mathbf{x}_h^T(t)$, thus $\mathbf{I}_{x_{ij}}(t) = X_i(t) X_j(t) \sin(\phi_i(t) - \phi_j(t))$.

We can now replace $\nabla \Delta\phi(t)$ in (1) to obtain

$$\begin{aligned} \nabla |\varrho|^2 &= \frac{2|\varrho|}{T} \left[\sum_{t=1}^T \frac{\sin[\Phi - \Delta\phi(t)]}{Y^2(t)} \mathbf{I}_x(t) \right] \mathbf{w} \\ &= 2|\varrho| \left\langle \frac{\sin[\Phi - \Delta\phi(t)]}{Y^2(t)} \mathbf{I}_x(t) \right\rangle \mathbf{w}. \end{aligned} \quad (2)$$

APPENDIX B GRADIENT OF J_l IN IPA

In this section we show that the gradient of J_l in Eq. 7 of [1] is given by Eq. 8 of [1]. Throughout this whole section, we will omit the dependence on the subspace l , for the sake of clarity. In other words, we are assuming (with no loss of generality) that only one subspace was found. Whenever we write \mathbf{W} , \mathbf{y} , y_m or \mathbf{z} , we will be referring to \mathbf{W}_l , \mathbf{y}_l , $(y_l)_m$ or \mathbf{z}_l .

The derivative of $\log |\det \mathbf{W}|$ is \mathbf{W}^{-T} . We will therefore focus on the gradient of the first term of Eq. 7 of [1], which we will denote by P :

$$P \equiv \frac{1 - \lambda}{N^2} \sum_{m,n} |\varrho_{mn}|^2.$$

Let's rewrite P as $P = \frac{1 - \lambda}{N^2} \sum_{m,n} p_{mn}$ with $p_{mn} = |\varrho_{mn}|^2$. Define $\Delta\phi_{mn} = \phi_m - \phi_n$. Omitting the time dependency, we

have

$$\begin{aligned}
\nabla_{\mathbf{w}_j} p_{mn} &= 2|\varrho_{mn}| \nabla_{\mathbf{w}_j} \left| \langle e^{i\Delta\phi_{mn}} \rangle \right| \\
&= |\varrho_{mn}| \times \left(\langle \cos(\Delta\phi_{mn}) \rangle^2 + i \langle \sin(\Delta\phi_{mn}) \rangle^2 \right)^{-1/2} \times \\
&\quad \times \left[2 \langle \cos(\Delta\phi_{mn}) \rangle \nabla_{\mathbf{w}_j} \langle \cos(\Delta\phi_{mn}) \rangle + \right. \\
&\quad \left. + 2 \langle \sin(\Delta\phi_{mn}) \rangle \nabla_{\mathbf{w}_j} \langle \sin(\Delta\phi_{mn}) \rangle \right] \\
&= 2|\varrho_{mn}| \left| \langle e^{i\Delta\phi_{mn}} \rangle \right|^{-1} \times \\
&\quad \times \left[- \langle \cos(\Delta\phi_{mn}) \rangle \langle \sin(\Delta\phi_{mn}) \nabla_{\mathbf{w}_j} \Delta\phi_{mn} \rangle + \right. \\
&\quad \left. + \langle \sin(\Delta\phi_{mn}) \rangle \langle \cos(\Delta\phi_{mn}) \nabla_{\mathbf{w}_j} \Delta\phi_{mn} \rangle \right], \quad (3)
\end{aligned}$$

where we have interchanged the partial derivative and the time average operators, and used

$$\left(\langle \cos(\Delta\phi_{mn}) \rangle^2 + i \langle \sin(\Delta\phi_{mn}) \rangle^2 \right)^{1/2} = \left| \langle e^{i\Delta\phi_{mn}} \rangle \right|.$$

Since ϕ_m is the phase of the m -th measurement, its derivative with respect to any \mathbf{w}_j is zero unless $m = j$ or $n = j$. In the former case, a reasoning similar to Appendix A shows that

$$\begin{aligned}
\nabla_{\mathbf{w}_j} \Delta\phi_{jk} &\equiv \nabla_{\mathbf{w}_j} \phi_j - \nabla_{\mathbf{w}_j} \phi_k = \nabla_{\mathbf{w}_j} \phi_j = \\
&= \frac{[\mathbf{z}_h \cdot \mathbf{z} - \mathbf{z} \cdot \mathbf{z}_h] \cdot \mathbf{w}_j}{Y_j^2} = \frac{\mathbf{\Gamma}_z \cdot \mathbf{w}_j}{Y_j^2}, \quad (4)
\end{aligned}$$

where $\mathbf{\Gamma}_z(t) = \mathbf{z}_h(t)\mathbf{z}^T(t) - \mathbf{z}(t)\mathbf{z}_h^T(t)$. It is easy to see that $\nabla_{\mathbf{w}_j} \Delta\phi_{jk} = -\nabla_{\mathbf{w}_j} \Delta\phi_{kj}$. Furthermore, $p_{mm} = 1$ by definition, hence $\nabla_{\mathbf{w}_j} p_{mm} = 0$ for all m and j . From these considerations, the only nonzero terms in the derivative of P are of the form

$$\begin{aligned}
\nabla_{\mathbf{w}_j} p_{jk} &= \nabla_{\mathbf{w}_j} p_{kj} = 2|\varrho_{jk}| \left| \langle e^{i(\phi_j - \phi_k)} \rangle \right|^{-1} \times \\
&\quad \times \left[- \langle \cos(\Delta\phi_{jk}) \rangle \left\langle \sin(\Delta\phi_{jk}) \frac{\mathbf{\Gamma}_z \cdot \mathbf{w}_j}{Y_j^2} \right\rangle + \right. \\
&\quad \left. + \langle \sin(\Delta\phi_{jk}) \rangle \left\langle \cos(\Delta\phi_{jk}) \frac{\mathbf{\Gamma}_z \cdot \mathbf{w}_j}{Y_j^2} \right\rangle \right]. \quad (5)
\end{aligned}$$

We now define $\Psi_{jk} \equiv \langle \phi_j - \phi_k \rangle = \langle \Delta\phi_{jk} \rangle$. Plugging in this definition into Eq. (5) we obtain

$$\begin{aligned}
\nabla_{\mathbf{w}_j} p_{jk} &= 2|\varrho_{jk}| \times \\
&\quad \times \left[- \cos(\Psi_{jk}) \left\langle \sin(\Delta\phi_{jk}) \frac{\mathbf{\Gamma}_z \cdot \mathbf{w}_j}{Y_j^2} \right\rangle + \right. \\
&\quad \left. + \sin(\Psi_{jk}) \left\langle \cos(\Delta\phi_{jk}) \frac{\mathbf{\Gamma}_z \cdot \mathbf{w}_j}{Y_j^2} \right\rangle \right] \\
&= 2|\varrho_{jk}| \left[\left\langle - \cos \Psi_{jk} \sin \Delta\phi_{jk} \frac{\mathbf{\Gamma}_z \cdot \mathbf{w}_j}{Y_j^2} \right\rangle + \right. \\
&\quad \left. + \left\langle \sin \Psi_{jk} \cos \Delta\phi_{jk} \frac{\mathbf{\Gamma}_z \cdot \mathbf{w}_j}{Y_j^2} \right\rangle \right] \\
&= 2|\varrho_{jk}| \left\langle \sin(\Psi_{jk} - \Delta\phi_{jk}) \frac{\mathbf{\Gamma}_z}{Y_j^2} \right\rangle \cdot \mathbf{w}_j,
\end{aligned}$$

where we again used $\sin(a - b) = \sin a \cos b - \cos a \sin b$ in the last step. Finally,

$$\begin{aligned}
\nabla_{\mathbf{w}_j} P &= \frac{1 - \lambda}{N^2} \sum_{m,n} \nabla_{\mathbf{w}_j} p_{mn} = 2 \frac{1 - \lambda}{N^2} \sum_{m < n} \nabla_{\mathbf{w}_j} p_{mn} = \\
&= 4 \frac{1 - \lambda}{N^2} \sum_{k=1}^N |\varrho_{jk}| \left\langle \sin[\Psi_{jk} - \Delta\phi_{jk}(t)] \frac{\mathbf{\Gamma}_z(t)}{Y_j(t)^2} \right\rangle \cdot \mathbf{w}_j.
\end{aligned}$$

which is Eq. 8 of [1].

APPENDIX C GRADIENT OF J IN PSCA

In this section we derive Eq. 10 of [1] for the gradient of J . Recall that J is given by

$$J \equiv \sum_{j=1}^P \left| \sum_{i=1}^N u_{ij} \right| = \sum_{j=1}^P \left| \sum_{i=1}^N \sum_{k=1}^P v_{ik} w_{kj} \right|$$

where the w_{kj} are real coefficients that we want to optimize and the v_{ik} are fixed complex numbers. Also recall that $\text{Re}(\cdot)$ and $\text{Im}(\cdot)$ denote the real and imaginary parts.

We begin by expanding the complex absolute value:

$$\begin{aligned}
\sum_j \left| \sum_{i,k} v_{ik} w_{kj} \right| &= \sum_j \left\{ \left[\text{Re} \left(\sum_{i,k} v_{ik} w_{kj} \right) \right]^2 \right. \\
&\quad \left. + \left[\text{Im} \left(\sum_{i,k} v_{ik} w_{kj} \right) \right]^2 \right\}^{1/2}.
\end{aligned}$$

When computing the derivative in order to w_{kj} , only one term in the leftmost sum matters. Thus,

$$\begin{aligned}
\frac{\partial J}{\partial w_{kj}} &= \\
&= 2 \left(\left[\text{Re} \left(\sum_{i,k} v_{ik} w_{kj} \right) \right]^2 + \left[\text{Im} \left(\sum_{i,k} v_{ik} w_{kj} \right) \right]^2 \right)^{-1/2} \times \\
&\quad \times \left[\frac{\partial \left[\text{Re} \left(\sum_{i,k} v_{ik} w_{kj} \right) \right]^2}{\partial w_{kj}} + \frac{\partial \left[\text{Im} \left(\sum_{i,k} v_{ik} w_{kj} \right) \right]^2}{\partial w_{kj}} \right] \\
&= \frac{1}{|\sum_i u_{ij}|} \left[\text{Re} \left(\sum_{i,k} v_{ik} w_{kj} \right) \frac{\partial \text{Re} \left(\sum_{i,k} v_{ik} w_{kj} \right)}{\partial w_{kj}} + \right. \\
&\quad \left. + \text{Im} \left(\sum_{i,k} v_{ik} w_{kj} \right) \frac{\partial \text{Im} \left(\sum_{i,k} v_{ik} w_{kj} \right)}{\partial w_{kj}} \right]. \quad (6)
\end{aligned}$$

In the sums inside the derivatives, the sum on k can be dropped as only one of those terms will be nonzero. Therefore,

$$\begin{aligned}
\frac{\partial \text{Re} \left(\sum_{i,k} v_{ik} w_{kj} \right)}{\partial w_{kj}} &= \frac{\partial \text{Re} \left(\sum_i v_{ik} w_{kj} \right)}{\partial w_{kj}} \\
&= \frac{\partial \text{Re} \left(\sum_i v_{ik} \right) w_{kj}}{\partial w_{kj}} = \text{Re} \left(\sum_i v_{ik} \right) = \text{Re}(\bar{v}_k),
\end{aligned}$$

where we used $\bar{v}_i \equiv \sum_k v_{ki}$ to denote the sum of the i -th column of \mathbf{V} . Similarly,

$$\frac{\partial \operatorname{Im}\left(\sum_{i,k} v_{ik} w_{kj}\right)}{\partial w_{kj}} = \operatorname{Im}(\bar{v}_k).$$

These results, with the notation $\bar{u}_j \equiv \sum_k v_{jk}$ as the sum of the j -th column of \mathbf{U} , can be plugged into Eq. (6) to yield

$$\mathbf{G}_{kj} = \frac{\partial J}{\partial w_{kj}} = \frac{1}{|\bar{u}_j|} \left[\operatorname{Re}(\bar{v}_k) \times \operatorname{Re}(\bar{u}_j) + \operatorname{Im}(\bar{v}_k) \times \operatorname{Im}(\bar{u}_j) \right].$$

APPENDIX D MEAN FIELD

In this section we derive Eq. 9 of [1] for the interaction of an oscillator with the cluster it is part of. We will assume that there are N_j oscillators in this cluster, coupled all-to-all with the same coupling coefficient κ , and that all inter-cluster interactions are weak enough to be disregarded. We begin with Kuramoto's model (Eq. 1 of [1]) omitting the time dependency:

$$\begin{aligned} \dot{\phi}_i &= \omega_i + \sum_{k \in c_j} \kappa_{ik} \sin(\phi_k - \phi_i) + \sum_{k \notin c_j} \kappa_{ik} \sin(\phi_k - \phi_i) \\ \dot{\phi}_i &= \omega_i + \sum_{k \in c_j} \kappa_{ik} \sin(\phi_k - \phi_i) \\ &= \omega_i + \sum_{k \in c_j} \kappa_{ik} \frac{e^{i(\phi_j - \phi_i)} - e^{-i(\phi_j - \phi_i)}}{2i} \\ &= \omega_i + \frac{e^{-i\phi_i}}{2i} \sum_{k \in c_j} \kappa_{ik} e^{i\phi_k} - \frac{e^{i\phi_i}}{2i} \sum_{k \in c_j} \kappa_{ik} e^{-i\phi_k}. \end{aligned}$$

We now plug in the definition of mean field $\varrho_{c_j} e^{i\Phi_{c_j}} = \frac{1}{N_j} \sum_{k \in c_j} e^{i\phi_k}$ to obtain

$$\begin{aligned} \dot{\phi}_i &= \omega_i + N_j \frac{e^{-i\phi_i}}{2i} \kappa \varrho_{c_j} e^{i\Phi_{c_j}} - N_j \frac{e^{i\phi_i}}{2i} \kappa \varrho_{c_j} e^{-i\Phi_{c_j}} \\ &= \omega_i + N_j \kappa \varrho_{c_j} [\sin(\Phi_{c_j} - \phi_i) - \sin(\phi_i - \Phi_{c_j})] \\ &= \omega_i + 2N_j \kappa \varrho_{c_j} \sin(\Phi_{c_j} - \phi_i). \end{aligned}$$

REFERENCES

- [1] 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.

Separation of phase-locked sources in pseudo-real MEG data

Miguel Almeida^{*12} and José Bioucas-Dias¹ and Ricardo Vigário²

¹Institute of Telecommunications, Superior Technical Institute, Portugal.

²Department of Information and Computer Science, Aalto University, Finland.

Email: Miguel Almeida^{*} - malmeida@lx.it.pt; José Bioucas-Dias - bioucas@lx.it.pt; Ricardo Vigário - ricardo.vigario@aalto.fi;

^{*}Corresponding author

Abstract

This paper addresses the blind separation of linear mixtures of synchronous signals (*i.e.*, signals with locked phases), which is a relevant problem, *e.g.*, in the analysis of electrophysiological signals of the brain such as the electroencephalogram (EEG) and the magnetoencephalogram (MEG). Popular separation techniques such as Independent Component Analysis (ICA) are not adequate for phase-locked signals, because such signals have strong mutual dependency. Aiming at unmixing this class of signals, we have recently introduced the Independent Phase Analysis (IPA) algorithm, which can be used to separate synchronous sources. Here, we apply IPA to pseudo-real MEG data. The results show that this algorithm is able to separate phase-locked MEG sources in situations where the phase jitter (*i.e.*, the deviation from the perfectly-synchronized case) is moderate. This represents a significant step towards performing phase-based source separation on real data.

1 Introduction

In recent years the interest of the scientific community in synchrony has risen. This interest is both in its physical manifestations and in the development of a theory unifying and describing those manifestations in various systems such as laser beams, astrophysical objects and brain neurons [1].

It is believed that synchrony plays a relevant role in the way different parts of the human brain interact. For example, when humans engage in a motor task, several brain regions oscillate coherently [2, 3]. Also, several pathologies such as autism, Alzheimer and Parkinson are associated with a disruption in the syn-

chronization profile of the brain, whereas epilepsy is associated with an anomalous increase in synchrony (see [4] for a review).

To perform inference on the synchrony of networks present in the brain or in other real-world systems, one must have access to the phase dynamics of the individual oscillators (which we will call “sources”). Unfortunately, in brain electrophysiological signals such as encephalograms (EEG) and magnetoencephalograms (MEG), and in other real-world situations, individual oscillator signals are not directly measurable, and 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 [5]. In this case, spurious synchrony may occur, as we will illustrate later.

The problem of undoing this superposition is called blind source separation (BSS). Typically, one assumes that the mixing is linear and instantaneous, which is a valid approximation in brain signals [6]. One must also make some assumptions on the sources, such as in Independent Component Analysis – ICA – where the assumption is mutual statistical independence of the sources [7]. ICA has seen multiple applications in EEG and MEG processing (for recent applications see, *e.g.*, [8, 9]). Different BSS approaches use criteria other than statistical independence, such as non-negativity of sources [10, 11] or time-dependent frequency spectrum criteria [12, 13]. In our case, independence of the sources is not a valid assumption, because phase-locked sources are highly mutually dependent. Also, phase-locking is not equivalent to frequency coherence: in fact, two signals may have a severe overlap between their frequency spectra but still exhibit low or no phase synchrony at all [14]. In this paper we address the problem of how to separate such phase-locked sources using a phase-specific criterion.

Recently, we have presented a two-stage algorithm called Independent Phase Analysis (IPA) which performed very well in noiseless simulated data [15] and with moderate levels of added Gaussian white noise [14]. The separation algorithm we then proposed uses Temporal Decorrelation Separation (TDSEP) [16] as a first step, followed by the maximization of an objective function involving the phases of the estimated sources. In [14] we presented a “proof-of-concept” of IPA, laying down the theoretical foundations of the algorithm and applying it to a toy dataset of manually generated data. However, in that paper we were not concerned with the application of IPA to real-world data. In this paper we study the applicability of IPA to pseudo-real MEG data. These data are not yet meant to allow inference about the human brain; however, they are generated in such a way that both the sources and the mixing process mimic what actually happens in the

¹In EEG and MEG, the sources are not individual neurons, whose oscillations are too weak to be detected from outside the scalp. In these cases, the sources are populations of closely located neurons oscillating together.

human brain. The advantage of using such pseudo-real data is that the true solution is known, thus allowing a quantitative assessment of the performance of the algorithm. We also study the robustness of IPA to the case where the sources are not perfectly phase-locked. It should however be reinforced that the algorithm presented here makes no assumptions that are specific of brain signals, and should work in any situation where phase-locked sources are mixed approximately linearly and noise levels are low.

This paper is organized as follows. In Sec. 2 we introduce the Hilbert Transform. We also introduce there the Phase Locking Factor (PLF), a measurement of synchrony which is central to the algorithm; finally, we show that synchrony is disrupted when the sources undergo a linear mixing. Sec. 3 describes the IPA algorithm in detail, including illustrations using a toy data set. In Sec. 4 we explain how the pseudo-real MEG data are generated and show the results obtained by IPA on those data. These results are discussed in Sec. 5 and conclusions are drawn in Sec. 6.

2 Background

2.1 Hilbert Transform: Phase of a real-valued signal

Usually, the signals under study are real-valued discrete signals. To obtain the phase of a real signal, one can use a complex Morlet (or Gabor) wavelet transform, which can be seen as a bank of bandpass filters [17]. Alternatively, one can use the Hilbert transform, which should be applied to a locally narrowband signal or be preceded by appropriate filtering [18] for the meaning of the phase extracted by the Hilbert Transform to be clear. The two transforms have been shown to be equivalent for the study of brain signals [19], but they may differ for other kinds of signals. In this paper we chose to use the Hilbert Transform. To ensure that this transform yields meaningful results, we will precede its use by band-pass filtering the pseudo-real MEG sources used in this paper (see Sec. 4.1). Note that this is a very common preprocessing step in the analysis of real MEG signals (*cf.*, [20–22]).

The discrete Hilbert transform $x_h(t)$ of a band-limited discrete-time signal $x(t)$, $t \in \mathbb{Z}$, is given by a convolution [18]:

$$x_h(t) \equiv x(t) * h(t), \text{ where } h(t) \equiv \begin{cases} 0, & \text{for even } t \\ \frac{2}{\pi t}, & \text{for odd } t. \end{cases}$$

Note that the Hilbert Transform is a linear operator. The Hilbert filter $h(t)$ is not causal and has infinite duration, which makes direct implementation of the above formula impossible. In practice, the Hilbert transform is usually computed in the frequency domain, where the above convolution becomes a product of the discrete Fourier transforms of $x(t)$ and $h(t)$. A more thorough mathematical explanation of this transform is given in [18] and [23]. We used the Hilbert transform as implemented by MATLAB.

The analytic signal of $x(t)$, denoted by $\tilde{x}(t)$, is given by $\tilde{x}(t) \equiv x(t) + i x_h(t)$, where $i = \sqrt{-1}$ is the imaginary unit. The phase of $x(t)$ is defined as the angle of its analytic signal. In the remainder of the paper, we drop the tilde notation; it should be clear from the context whether the signals under consideration are the real signals or the corresponding analytic signals.

2.2 Phase-Locked Sources

Throughout this paper we assume that the sought sources, in number of N and denoted by s_j , $j = 1 \dots, N$, are phase-locked. In other words, s_j , $j = 1 \dots, N$ are complex valued signals with nonnegative amplitudes and equal phase up to a constant plus small perturbations. Formally,

$$s_j(t) = a_j(t) e^{i(\alpha_j + \phi(t) + \delta_j(t))}, \quad (1)$$

where $a_j(t)$ are the amplitudes of the sources, which are by definition non-negative and real-valued. α_j is the constant dephasing (or phase lag) between the sources (it does not depend on the time t), $\phi(t)$ represents an oscillation common to all the sources (it does not depend on the source j), and $\delta_j(t)$ is the phase jitter, which represents the deviation of the j -th source from its nominal phase $\alpha_j + \phi(t)$. Throughout this paper we will assume that the phase jitter is Gaussian with zero mean and a standard deviation σ .

One situation where signals follow the model in (1) is the one described by the (time-dependent) Kuramoto model, under some circumstances. This simple model has been extensively used in the context of, *e.g.*, modeling neuronal excitation and inhibition interactions, as well as large-scale experimental neuroscience data [20,24]. Under this model, the interactions between oscillators are weak relative to the stability of their limit cycles, and thus affect the oscillators' phases only, not their amplitudes. The phase of oscillator j is governed by [1,25,26]

$$\dot{\phi}_j(t) = \omega_j(t) + \frac{1}{N} \sum_{k=1}^N \kappa_{jk} \sin [\phi_k(t) - \phi_j(t)], \quad (2)$$

where $\phi_j(t)$ is the phase of oscillator j (it is unrelated to $\phi(t)$ in equation (1)), $\omega_j(t)$ is its natural frequency, and κ_{jk} measures the strength of the interaction between oscillators j and k . If the κ_{jk} coefficients are large enough and $\omega_j(t) = \omega_k(t)$ for all j, k , then the solutions of the Kuramoto model are of the form (1) with small $\delta_j(t)$.

2.3 Phase Locking Factor

Given two oscillators with phases $\phi_j(t)$ and $\phi_k(t)$ for $t = 1, \dots, T$, the real-valued² Phase Locking Factor (PLF), or Phase Locking Value (PLV), between those two oscillators is defined as

$$\varrho_{jk} \equiv \left| \frac{1}{T} \sum_{t=1}^T e^{i[\phi_j(t) - \phi_k(t)]} \right| = \left| \left\langle e^{i(\phi_j - \phi_k)} \right\rangle \right|, \quad (3)$$

where $\langle \cdot \rangle$ is the time average operator. The PLF satisfies $0 \leq \varrho_{jk} \leq 1$. The value $\varrho_{jk} = 1$ corresponds to two oscillators that are fully synchronized (*i.e.*, their phase lag is constant). In terms of Eq. (1), a PLF of 1 is obtained only if the phase jitter $\delta_j(t)$ is zero. The value $\varrho_{jk} = 0$ is attained, for example, if the phase difference $\phi_j(t) - \phi_k(t)$ modulo 2π is uniformly distributed in $[-\pi, \pi[$. Values between 0 and 1 represent partial synchrony; in general, higher values of the standard deviation of the phase jitter $\delta_j(t)$ yield lower PLF values.

Note that a PLF of 1 is obtained if and only if $\phi_j(t) - \phi_k(t)$ is constant.³ Thus, studying the separation of sources with constant phase lags can equivalently become the study of separation of sources with pairwise PLFs of 1.

Throughout this paper, phase synchrony is measured using the PLF; two signals are perfectly synchronous if and only if they a PLF of 1. Other approaches exist, *e.g.*, for chaotic systems or specific types of oscillators [27]. Studying separation algorithms based on such other definitions is outside of the scope of this paper. The definition used here has the advantages of being tractable from an algorithmic point of view, and of being applicable to any situation where $\phi_j(t) - \phi_k(t)$ is constant,⁴ regardless of the type of oscillator.

2.4 Effect of linear mixing on synchrony

Assume that we have N sources which have PLFs of 1 with each other. Let $\mathbf{s}(t)$, for $t = 1, \dots, T$, denote the vector of sources and $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$ denote the mixed signals, where \mathbf{A} is the mixing matrix, which is assumed to be square and non-singular.⁵ Our goal is to find a square unmixing matrix \mathbf{W} such that the estimated sources $\mathbf{y}(t) = \mathbf{W}^T \mathbf{x}(t) = \mathbf{W}^T \mathbf{A}\mathbf{s}(t)$ are as close to the true sources as possible, up to permutation, scaling, and sign change.

The effect of linear mixing on the PLF matrix is illustrated in Fig. 1 for a set of simulated sources. This set has three sources, with PLFs of 1 with each other. These sources are of the form (1) with negligible

²The term “real-valued” is used here to distinguish from other phase-based algorithms where a complex quantity is used [14].

³Technically, this condition could be violated in a set with zero measure. Since we will deal with a discrete and finite number of time points, no such sets exist and this technicality is not important.

⁴We will also show results where this phase difference is not exactly constant; see Fig. 6.

⁵These assumptions are not as restrictive as they may sound; see Section 3.1.

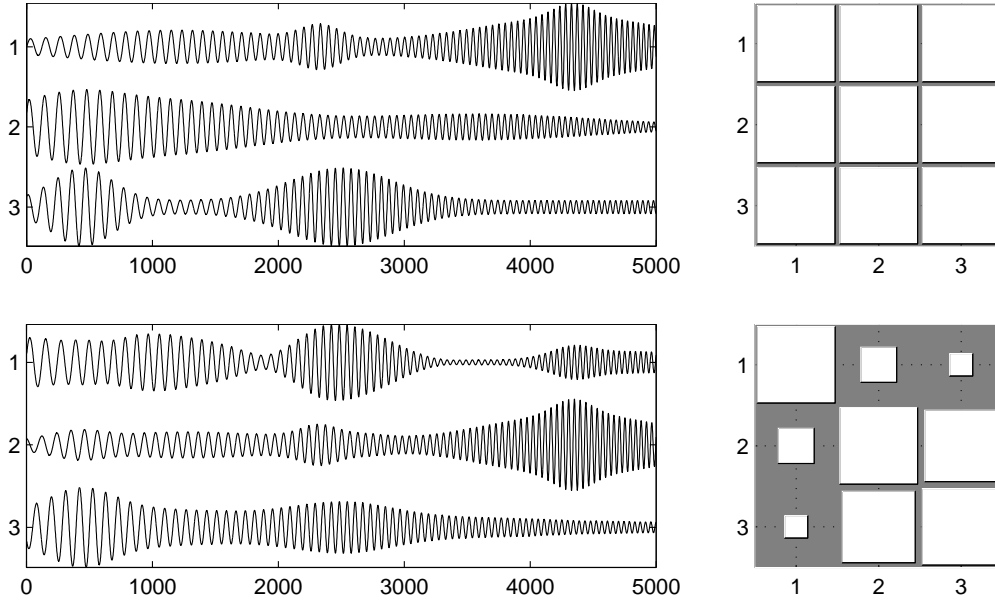


Figure 1: Top row: The three original sources (left) and PLFs between them (right). Bottom row: The three mixed signals (left) and PLFs between them (right). On the right column, the area of the square in position (i, j) is proportional to the PLF between the signals i and j . Therefore, large squares represent PLFs close to 1, while small squares represent values close to zero. In this example, the second and third sources have phase lags of $\frac{\pi}{6}$ and $\frac{\pi}{3}$ radians relative to the first source, respectively.

phase jitter, and the phase lags α_j are 0, $\frac{\pi}{6}$, and $\frac{\pi}{3}$ radians, respectively. The common oscillation is a time-dependent sinusoid. The amplitudes are generated by adding a small constant baseline to a random number of “bursts” with Gaussian shape. Each “burst” has a random center and a random width, and each source amplitude has 1 to 5 such “bursts”.

The first row of Fig. 1 shows on the left the original sources and on the right their PLF matrix. The second row depicts the mixed signals $\mathbf{x}(t)$ on the left and their PLFs on the right; the mixing matrix has random entries uniformly distributed between -1 and 1. It is clear that the mixed signals have lower pairwise PLFs than the sources, although signals 2 and 3 still exhibit a rather high mutual PLF. This example suggests that linear mixing of synchronous sources reduces their synchrony, a fact that will be proved in Section 3.3, ahead; this fact will be used to extract the sources from the mixtures by trying to maximize the PLF of the estimated sources.

3 Algorithm

In this section we describe the Independent Phase Analysis (IPA) algorithm. As mentioned in Sec. 1, this algorithm first performs subspace separation, and then performs separation within each subspace. In this

paper we only study the performance of IPA in the case where all the sources are phase-locked; in this situation, the inter-subspace separation can be entirely skipped, since there is only one subspace of locked sources. Therefore, we will not discuss here the part of IPA relating to subspace separation; the reader is referred to [14] for a discussion on that subject.

3.1 Preprocessing

3.1.1 Whitening

As happens in ICA and other source separation techniques, whitening is a useful preprocessing step for IPA. Whitening, or sphering, is a procedure that linearly transforms the data so that the transformed data have the identity as its covariance matrix; in particular, the whitened data are uncorrelated [7]. In ICA, there are clear reasons to pursue uncorrelatedness: independent data are also uncorrelated, and therefore whitening the data already fulfills one of the required conditions to find independent sources. If \mathbf{D} denotes the diagonal matrix containing the eigenvalues of the covariance matrix of the data and \mathbf{V} denotes an orthonormal matrix which has, in its columns, the corresponding eigenvectors, then whitening can be performed in a PCA-like manner by multiplying the data $\mathbf{x}(t)$ by a matrix \mathbf{B} , where [7]

$$\mathbf{B} = \mathbf{D}^{-1/2} \mathbf{V}^T. \quad (4)$$

The whitened data are given by $\mathbf{z}(t) = \mathbf{B}\mathbf{A}\mathbf{s}(t)$. Therefore, whitening merely transforms the original source separation problem with mixing matrix \mathbf{A} into a new problem with mixing matrix $\mathbf{B}\mathbf{A}$. The advantage is that $\mathbf{B}\mathbf{A}$ is an orthogonal mixing matrix, and its estimation becomes easier [7].

The above reasoning is not valid for the separation of phase-locked sources. However, under rather general assumptions, satisfied by the data studied here, it can be shown that whitening places a relatively low upper bound on the condition number of the equivalent mixing matrix (see [28] and references therein). Therefore, we always whiten the mixture data before applying the procedures described in Sec. 3.2.

3.1.2 Number of sources

As will be seen below, IPA assumes knowledge of the number of sources, and also assumes that the mixing matrix is square: if this is not the case, a simple procedure can be used to detect the number of sources and to transform the data to obey these constraints. If the mixing process is noiseless and is given by $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$, where \mathbf{A} has more rows than columns and has maximum rank⁶, the number of nonzero eigenvalues of the

⁶This is usually called the over-determined case. The under-determined case, where \mathbf{A} has fewer rows than columns, is more difficult and is not addressed here.

covariance matrix of \mathbf{x} is N , where N is the number of sources (or equivalently, the number of columns of \mathbf{A}). If the mixture is noisy with a low level of i.i.d. Gaussian additive noise, the former zero-valued eigenvalues now have small non-zero values, but detection of N is still easy to do by detecting how many eigenvalues are large relative to the plateau level of the small eigenvalues [7].⁷ After N is known, the data need only be multiplied by a matrix $\mathbf{B}' = \mathbf{D}'^{-1/2}\mathbf{V}'^T$ in a similar fashion to Eq. (4), where \mathbf{D}' is a smaller $N \times N$ diagonal matrix containing only the N largest eigenvalues in \mathbf{D} and \mathbf{V}' is a rectangular matrix containing only the N columns of \mathbf{V} corresponding to those eigenvalues. The mixture to be separated now becomes

$$\mathbf{x}'(t) = \mathbf{B}'\mathbf{x}(t) = \mathbf{B}'\mathbf{A}\mathbf{s}(t). \quad (5)$$

Since $\mathbf{B}'\mathbf{A}$ is a square matrix and the number of sources is now given simply by the number of components of \mathbf{x}' , the problem now has a known number of sources and a square mixing matrix.

A remark should be made about complex-valued data. The above procedure is appropriate when both the mixing matrix and the sources are real-valued. If both the mixing matrix and the sources are complex-valued, Eq. (4) still applies (\mathbf{V} will now have complex values). However, in our case the sources and measurements are complex-valued (due of the Hilbert Transform), but the mixing matrix is real. When this is the case, Eq. (4) is not directly applicable. The above procedure must instead be applied not to the original data $\mathbf{x}(t)$, but to new data \mathbf{x}_0 with twice as many time samples, given by $\mathbf{x}_0(t) = \mathcal{R}(\mathbf{x}(t))$ for $t = 1, \dots, T$ and $\mathbf{x}_0(t) = \mathcal{I}(\mathbf{x}(t - T))$ for $t = T + 1, \dots, 2T$, where \mathcal{R} and \mathcal{I} denote the real and imaginary parts of a complex number, respectively. The matrix \mathbf{B} which results from applying Eq. (4) to \mathbf{x}_0 (or \mathbf{B}' if appropriate) is then applied to the original data \mathbf{x} as before, and the remainder of the procedure is similar [28].

3.2 Separation of phase-locked sources

The goal of the IPA algorithm is to separate a set of N fully phase-locked sources which have been linearly mixed. Since these sources have a maximal PLF with each other and the mixture components do not (as motivated in Section 2.4 above and proved in Section 3.3 below), we can unmix them by searching for projections that maximise the resulting PLFs. Specifically, this corresponds to finding a N by N matrix \mathbf{W} such that the estimated sources, $\mathbf{y}(t) = \mathbf{W}^T\mathbf{x}(t) = \mathbf{W}^T\mathbf{A}\mathbf{s}(t)$, have the highest possible PLFs.

⁷There are more rigorous criteria that can be used to choose N . Two very popular methods are the Akaike Information Criterion (AIC) and the Minimum Description Length (MDL). It is out of the scope of this paper to discuss these two criteria; the reader is referred to [7] and references therein for more information.

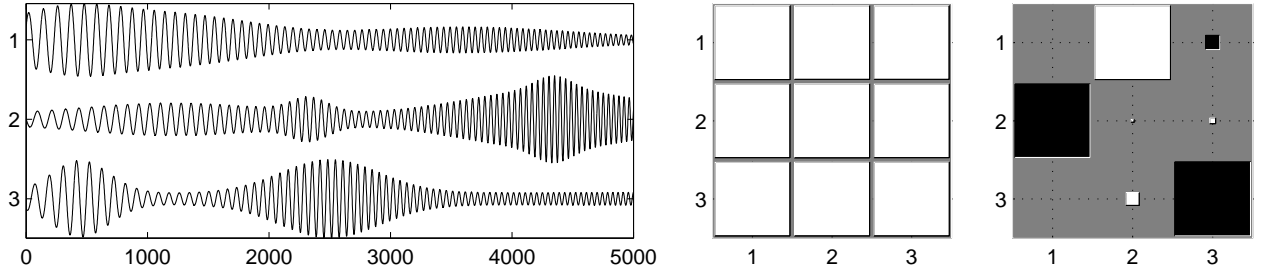


Figure 2: The three sources estimated by IPA (left), PLFs between them (middle), and the gain matrix $\mathbf{W}^T \mathbf{A}$ (right). Black squares represent negative values of the gain matrix, while white squares represent positive values. Since the gain matrix is very close to a permuted diagonal matrix, we can conclude that IPA successfully recovered the sources, up to permutation, scaling, and sign change.

The optimization problem that we shall solve is

$$\begin{aligned} \max_{\mathbf{W}} \quad & (1 - \lambda) \sum_{\substack{j,k=1 \\ j>k}}^N \varrho_{jk}^2 + \lambda \log |\det \mathbf{W}| \\ \text{s.t.} \quad & \|\mathbf{w}_j\| = 1, \text{ for } j = 1, \dots, N \end{aligned} \quad (6)$$

where \mathbf{w}_j is the j -th column of \mathbf{W} . In the first term we sum the squared PLFs between all pairs of sources. The second term penalizes unmixing matrices that are close to singular, and λ is a parameter controlling the relative weights of the two terms. This second term serves the purpose of preventing the algorithm from finding, *e.g.*, solutions where two columns j and k of \mathbf{W} are colinear, which trivially yields $\varrho_{jk} = 1$ (a similar term is used in some ICA algorithms [7]). Each column of \mathbf{W} is constrained to have unit norm to prevent trivial decreases of that term.

The optimization problem in Eq. (6) is highly non-convex: the objective function is a sum of two terms, each of which is non-convex in the variable \mathbf{W} . Furthermore, the unit norm constraint is also non-convex. Despite this, as we show below in Section 3.3, it is possible to characterize all the global maxima of this problem for the case $\lambda = 0$ and to devise an optimization strategy taking advantage of that result.

The above optimization problem can be tackled through various maximization algorithms. Our choice was to use a gradient ascent algorithm with momentum and adaptive step sizes; after this gradient algorithm has run for 200 iterations, we use the BFGS algorithm implemented in MATLAB to improve the solution. The result of this optimization for the sources shown in Fig. 1 is shown in Fig. 2 for $\lambda = 0.1$, illustrating that IPA successfully recovers the original sources for this dataset.

3.3 Unicity of solution

In [14] we proved that a few mild assumptions on the sources, which are satisfied in the vast majority of real-world situations, suffice for a useful characterization of the global maxima of Problem (6): it turns out that there are infinitely many such maxima, and that they correspond either to correct solutions (*i.e.*, the original sources up to permutation, scaling, and sign changes) or to singular matrices \mathbf{W} . More specifically, we proved the following: Assume that we have a set of complex-valued and linearly independent sources denoted by $\mathbf{s}(t)$, which have a PLF of 1 with one another. Consider also linear combinations of the sources of the form $\mathbf{y}(t) = \mathbf{C}\mathbf{s}(t)$ where \mathbf{C} is a square matrix of appropriate dimensions. Further assume that the following conditions hold:

1. Neither $s_j(t)$ nor $y_j(t)$ can be identically zero, for all j .
2. \mathbf{C} is non-singular.
3. The phase lag between any two sources is different from 0 or π .
4. The amplitudes of the sources, $a_j(t) = |s_j(t)|$, are linearly independent.

Then, the only linear combination $\mathbf{y}(t) = \mathbf{C}\mathbf{s}(t)$ of the sources $\mathbf{s}(t)$ in which the PLF between any two components of \mathbf{y} is 1 is $\mathbf{y}(t) = \mathbf{s}(t)$, up to permutation, scaling, and sign changes [14].

3.4 Comparison to ICA

The above result is simple, but some relevant remarks should be made. If the optimum is found using $\lambda = 0$ and the second assumption is not violated (or equivalently, $\det(\mathbf{C}) = \det(\mathbf{W})\det(\mathbf{A}) \neq 0$, which is equivalent to $\det(\mathbf{W}) \neq 0$ if \mathbf{A} is non-singular), then we can be certain that the correct solution has been found. However, if the optimization is made using $\lambda = 0$, there is a possibility that the algorithm will estimate a bad solution where, for example, some of the estimated sources are all equal to one another (in which case the PLFs between those estimated sources is trivially equal to 1). On the other hand, if we use $\lambda \neq 0$ to guarantee that \mathbf{W} is nonsingular, the unicity result stated above cannot be applied to the complete objective function. We call “non-singular solutions” and “singular solutions” those in which $\det(\mathbf{W}) \neq 0$ and $\det(\mathbf{W}) = 0$, respectively. The result expressed in Sec. 3.3 is thus equivalent to stating that “all non-singular global optima of Eq. (6) with $\lambda = 0$ correspond to correct solutions”.

This contrasts strongly with ICA, where singular solutions are not an issue, because ICA algorithms attempt to find independent sources and one signal is never independent from itself [7]. In other words, sin-

gular solutions always yield poor values of the objective function of ICA algorithms. Here we are attempting to estimate phase-locked sources, and any signal is perfectly phase-locked with itself. Thus, one must always use $\lambda \neq 0$ in the objective function of Eq. (6) when attempting to separate phase-locked sources.

We use a simple strategy to deal with this problem. We start by optimizing Eq. (6) for a relatively large value of λ ($\lambda = 0.4$), and once convergence has been obtained, we use the result as the starting point for a new optimization, this time with $\lambda = 0.2$. The same process is repeated with the value of λ halved each time, until five such epochs have been run. The early optimization steps move the algorithm away from the singular solutions discussed above, whereas the final steps are done with a very low value of λ , where the above unicity conditions are approximately valid. As the following experimental results show, this strategy can successfully prevent singular solutions from being found, while making the influence of the second term of Eq. (6) on the final result negligible.

4 Experimental Results

4.1 Data generation

As mentioned earlier, the main goal of this work is to study the applicability of IPA to real-world electrophysiological data from human brain EEG and MEG. The choice of the data for this study was not trivial, since we need to know the true sources in order to quantitatively measure the quality of the results. On the one hand, to know the actual sources in the brain would require simultaneous data from outside the scalp (EEG or MEG, which would be the mixed signals) and from inside the scalp (intra-cranial recordings, corresponding to the sources). If intra-cranial recordings are not available, results cannot be quantitatively assessed; they can only be qualitatively assessed by experts who can tell whether the extracted sources are meaningful or not. On the other hand, due to their extreme simplicity, synthetic data such as those used so far to illustrate IPA, shown in Figure 1, cannot be used to assess the usefulness of the method in real-world situations.

In an attempt to obtain “the best of both worlds”, we have generated a pseudo-real data set from actual MEG recordings. By doing this, we know the true sources and the true mixing matrix, while still using sources that are of a nature similar to what one observes in real-world MEG. We begin by describing the process that we used to generate a perfectly phase-locked data set; we then explain how we modified these data to analyze non-perfect cases as well. It is important to stress that the generation process described below has no relation to the one used to generate the data of Figure 1, even though both processes generate sources with maximum PLF.

Our first step was to obtain a realistic mixing matrix. To do so, we used the well-known EEGIFT software package [29]. This package includes a real-world sample EEG dataset with 64 channels. Using all the default options of the software package, we extracted 20 independent components from the data of Subject 1 in that dataset. The result that was important for us, in this process, were not the independent components themselves (which were discarded), but rather the 64×20 mixing matrix. As discussed in Section 3.1, we have opted for using a square mixing matrix, with little loss of generality. Therefore, we selected N random rows and N random columns of that mixing matrix (without repetition), and formed a $N \times N$ mixing matrix from the corresponding values of the original 64×20 matrix. We will later show results for data sets ranging from $N = 2$ to $N = 5$ sources; in the following, assume, for the sake of concreteness, that $N = 4$.

Having generated a physiologically plausible mixing matrix, the next step was to generate a set of four sources. For this, we used the MEG dataset studied previously in [30],⁸ which has 122 channels with 17730 samples per channel. The sampling frequency is 297 Hz, and the data have already been subjected to low-pass filtering with cutoff at 90 Hz. Since band-pass filtering is a very common preprocessing step in the analysis of MEG data [20–22] and is useful for the use of the Hilbert Transform, we performed a further band-pass filtering with no phase distortion, keeping only the 18-24 Hz band.⁹ The resulting filtered data were used to generate a complex signal through the Hilbert Transform; these data were whitened as described in Sec. 3.1, and from the whitened data we extracted the time-dependent amplitudes and phases.

We then selected four random channels of these filtered MEG data. Since none of these MEG recordings were actually phase-locked (recall that they were themselves the result of a mixing process) and we wanted to study the performance on fully phase-locked sources (possibly corrupted by jitter, as explained below), we replaced the phase of the second of these channels with the phase of the first channel with a constant phase lag of $\frac{\pi}{6}$ radians. The phase of the third channel was replaced with the phase of the first channel with a constant phase lag of $\frac{\pi}{3}$ radians, and that of the fourth channel with the phase of the first channel with a lag of $\frac{\pi}{2}$ radians. The amplitudes of the four sources were kept as the original amplitudes of the four random channels themselves. The process is illustrated in Fig. 3. The above process, including the choice of the 4×4 submatrix, was repeated 100 times, with different initializations of the random number generator. This way of constructing the data ensured that the sources were fully phase-locked.

We also constructed datasets in which the sources were not perfectly phase locked. For this, we used the

⁸Freely available from http://research.ics.tkk.fi/ica/eegmeg/MEG_data.html

⁹The choice of this specific band is rather arbitrary. The band is narrow enough that the Hilbert Transform will allow correct estimation of instantaneous amplitude and phase, but wide enough that the instantaneous frequency of the signals retains some variability. The passband is also of a similar width as in typical studies using MEG [20].

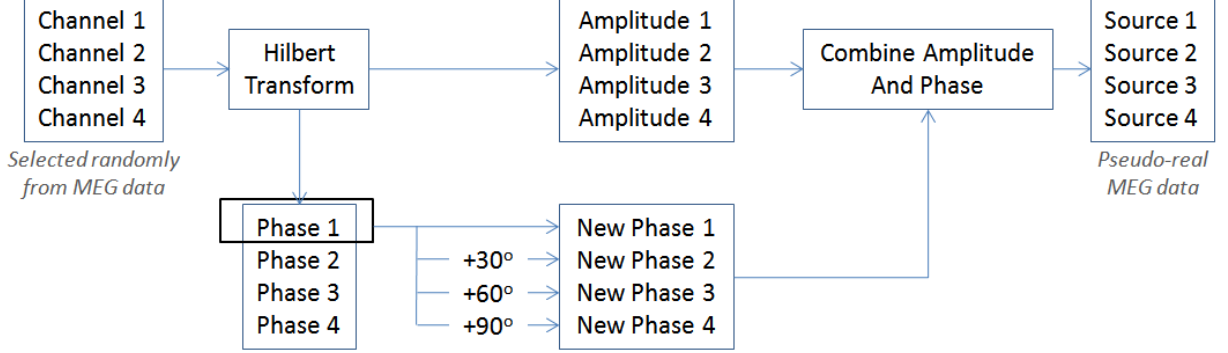


Figure 3: The process used to generate the pseudo-real MEG sources.

same 100 sets of sources, but with those sources now corrupted by phase jitter: each sample t of each source j was multiplied by $e^{i\delta_j(t)}$, where the phase jitter $\delta_j(t)$ was drawn from a random Gaussian distribution with zero mean and standard deviation σ . We tested IPA for σ from 0 to 20 degrees, in 5 degrees steps. One example with $\sigma = 5$ degrees is shown in Fig. 4, and one with $\sigma = 20$ degrees is shown in Fig. 5.

Finally, we studied the effect of N on the results of the proposed algorithm. We created 100 datasets similar to the jitterless datasets mentioned earlier, using $N = 2, 3$ and 5 . In all of these, and similarly to the data with $N = 4$, we used sources with phase lags multiple of $\frac{\pi}{3}$.

4.2 Results

We measured the separation quality using two measures: the Amari Performance Index (API) [31] and the well-known Signal to Noise Ratio (SNR). The API measures how far the gain matrix $\mathbf{W}^T \mathbf{A}$ is from a permuted diagonal matrix; the SNR measures how far the estimated sources are from the true sources. In summary, the API measures the quality of the estimation of the mixing matrix, while the SNR measures the quality of the estimation of the sources themselves.

Fig. 6 presents the means and standard deviations of these measures for the 100 runs mentioned in Sec. 4.1, for each of the jitter levels. The results indicate that IPA has virtually perfect performance on the jitterless case, in data of this kind, and that this level of performance is approximately maintained even in the presence of low levels of phase jitter, up to 5 degrees of standard deviation. Some deterioration in performance occurs from 5 to 10 degrees of phase jitter standard deviation, but with a SNR of 27 dB and an API below 0.1 the sources can still be considered to be well estimated.

The results for high jitter levels (*sigma* equal to 15 or 20 degrees) show that there is a limit to IPA's

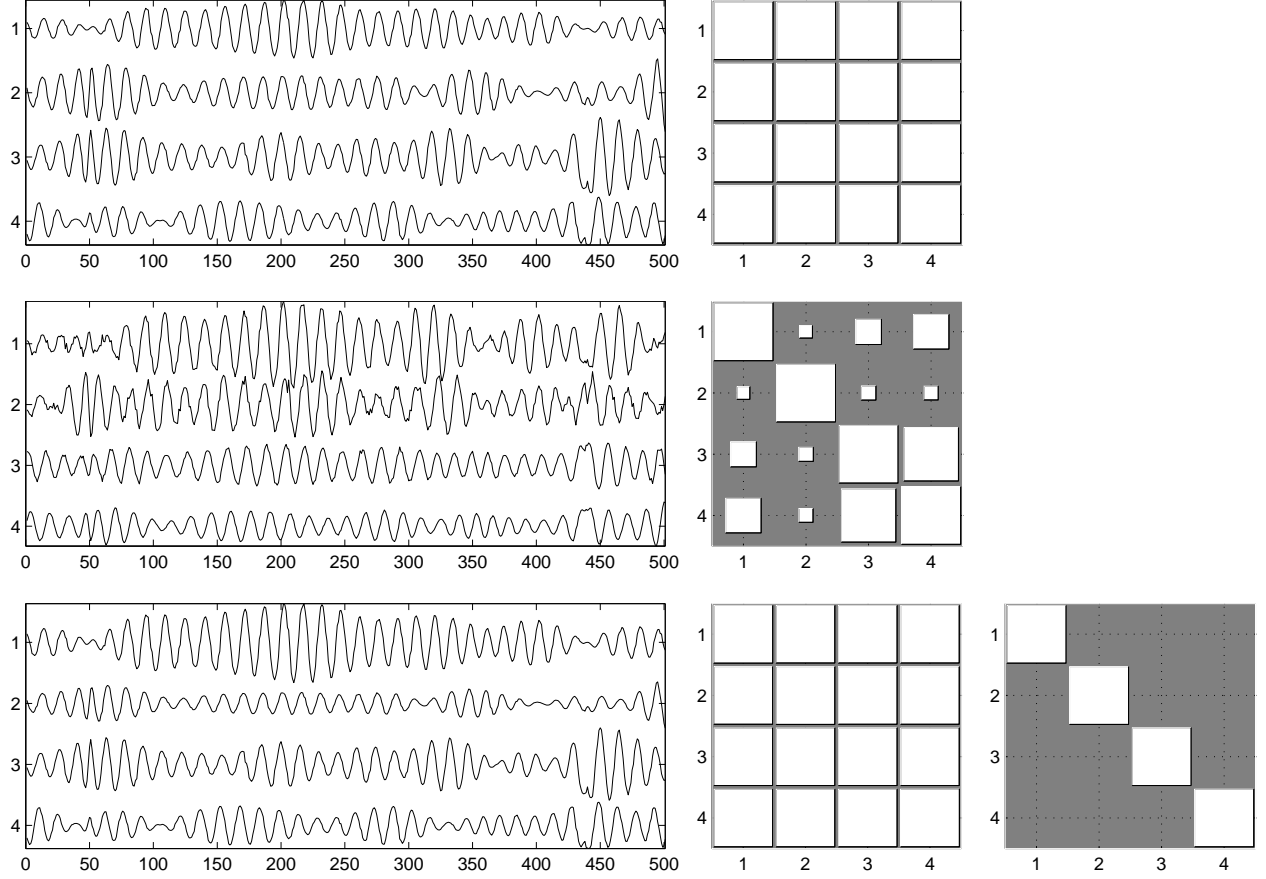


Figure 4: Example of a dataset where $\sigma = 5$ degrees. Only a short segment of the signals is shown, for clarity. Top row: original sources (left) and PLFs between them (right). Middle row: mixed signals (left) and PLFs between them (right). Bottom row: estimated sources, after manual compensation of permutation, scaling, and sign (left); PLFs between them (middle); and the gain matrix $\mathbf{W}^T \mathbf{A}$ (right). The gain matrix is virtually equal to the identity matrix, indicating a correct separation.

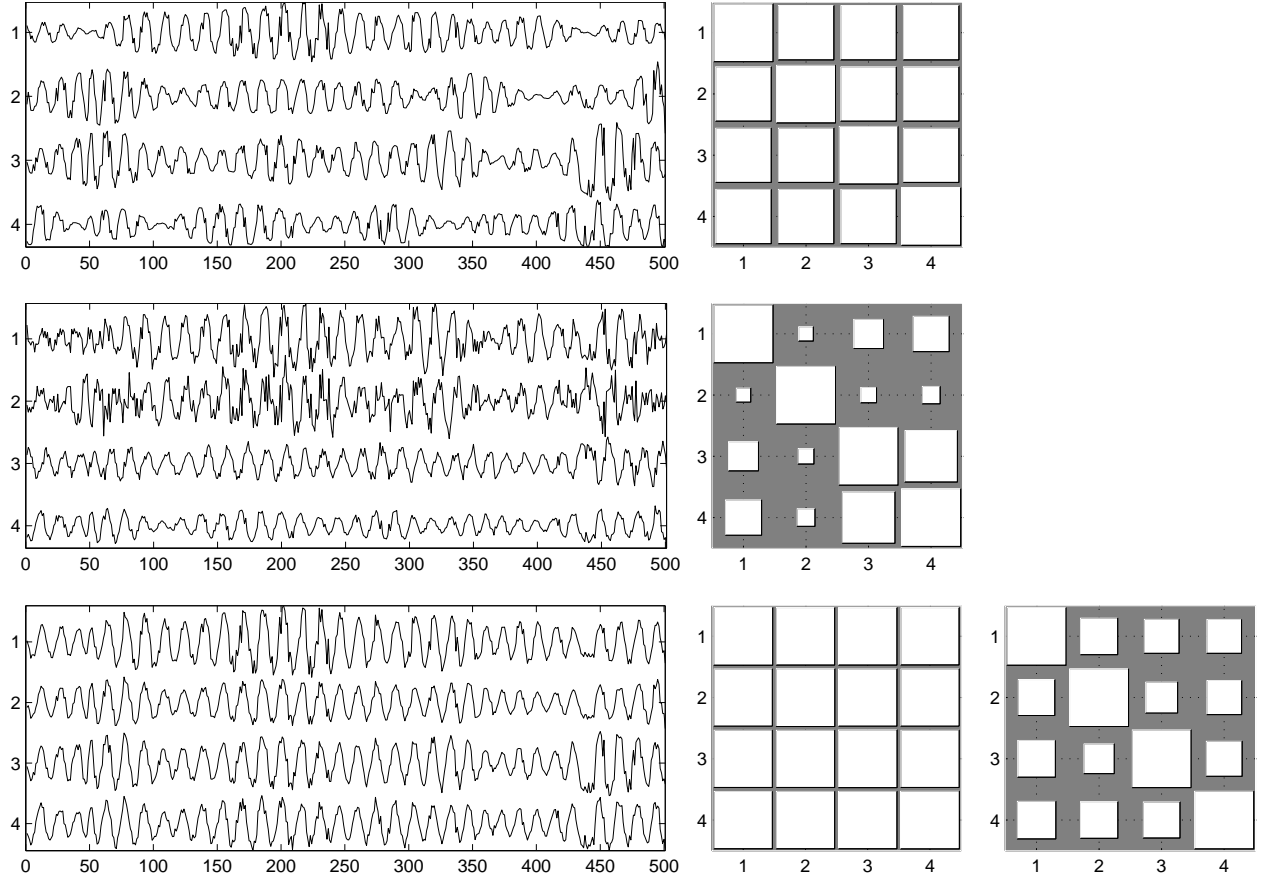


Figure 5: Example of a dataset where $\sigma = 20$ degrees. Only a short segment of the signals is shown, for clarity. Top row: original sources (left) and PLFs between them (right). Middle row: mixed signals (left) and PLFs between them (right). Bottom row: estimated sources, after manual compensation of permutation, scaling, and sign (left); PLFs between them (middle); and the gain matrix $\mathbf{W}^T \mathbf{A}$ (right). The gain matrix has significant values outside the diagonal, indicating that a complete separation was not achieved. Nevertheless, the largest values are in the diagonal, corresponding to a partial separation.

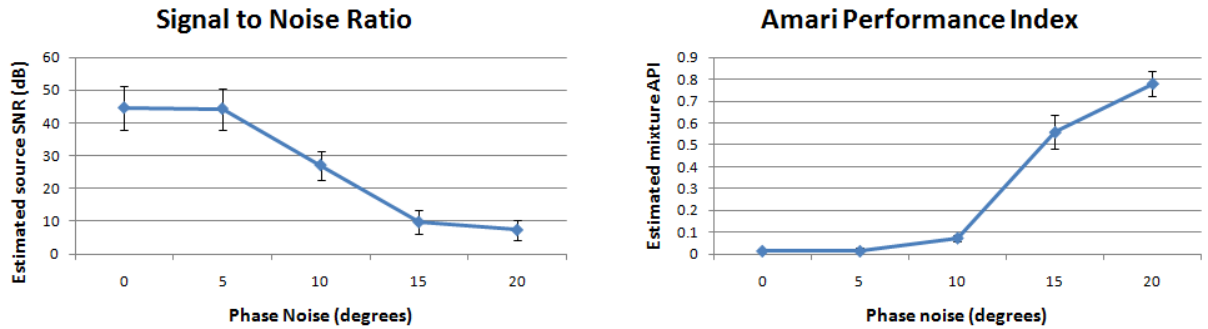


Figure 6: Result of applying IPA to pseudo-real MEG data with $N = 4$, with varying phase jitter: Signal to Noise Ratio (left) and Amari Performance Index (right).

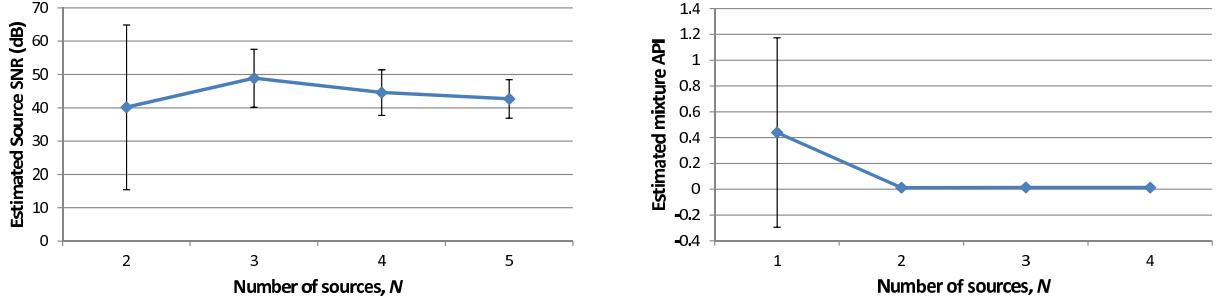


Figure 7: Effect of applying IPA to pseudo-real MEG data with varying values of N : Signal to Noise Ratio (left) and Amari Performance Index (right).

robustness; this limit lies somewhere between 10 and 15 degrees. Equivalently, in terms of the PLF, the algorithm shows good robustness to PLF values smaller than 1 as long as they are above 0.95, but below that value its performance deteriorates progressively up to a PLF of approximately 0.9, at which point only partial separations are obtained.

We show the effect of varying the number of sources N in figure 7. The figure shows that IPA can handle values of N up to $N = 5$ with only a slight decrease in performance.

Figure 7 also shows something rather peculiar: for $N = 2$, the results are mediocre (with an average API around 0.4).¹⁰ This is not an effect of lowering the number of sources N , but rather an indirect effect of the phase lag between the sources. We generated datasets of jitterless data with $N = 2$, using phase lags of $\frac{\pi}{12}$, $\frac{2\pi}{12}$ (the value used in figure 7), $\frac{3\pi}{12}$ and $\frac{4\pi}{12}$ (100 datasets for each of these values). Figure 8 shows that a phase lag of $\frac{2\pi}{12}$ yields poor API values, as we already knew, but $\frac{3\pi}{12}$ yields very good values. Naively, one could conclude that when the sources have a phase lag of $\frac{\pi}{6}$, the separation cannot be accurately undone.

The effect is, however, not so clear-cut. The results for $N = 3, 4, 5$ also involve sources with phase lags of $\frac{\pi}{6}$, but the API values for those experiments are very good. We do not have a solid explanation for this fact; we conjecture that the presence of some pairs of sources with larger phase lags (for example, for $N = 4$, the first and third sources have a phase lag of $\frac{\pi}{3}$ and the first and fourth sources have a phase lag of $\frac{\pi}{2}$) aids in the separation of all the sources.

¹⁰It might appear contradictory that the average SNR has a good value, 40 dB, when the average API has a mediocre score. In reality, when the standard deviation of the SNR is very high, it is usually an indication that the separation is poor. As an example, consider a case where one source is very well estimated, with an SNR of 80 dB, and one is poorly estimated, with an SNR of 0 dB. The average SNR would be 40, but with a very high standard-deviation. Good values of the average SNR are indicators of a good separation only when the standard-deviation of the SNR is small.

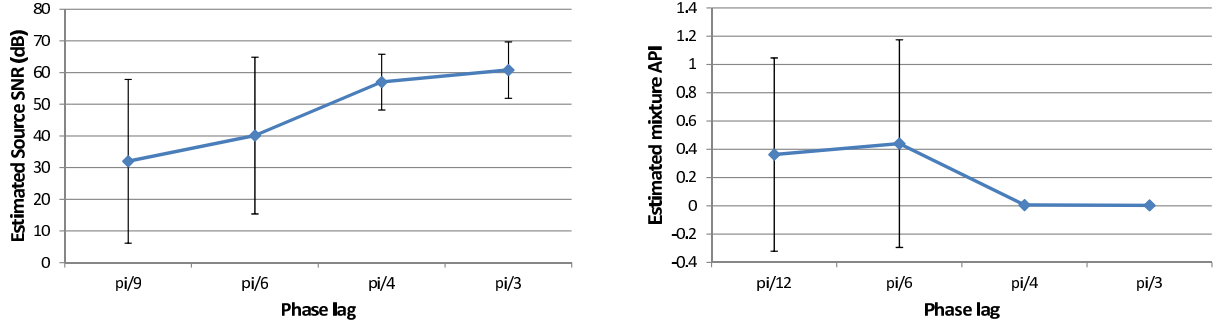


Figure 8: Effect of applying IPA to pseudo-real MEG data with varying phase lags between the sources, with $N = 2$: Signal to Noise Ratio (left) and Amari Performance Index (right).

	λ	0.025	0.05	0.1	0.2	0.4
SNR	fixed	17.5 ± 21.2	27.5 ± 18.0	34.4 ± 4.3	27.2 ± 3.6	13.5 ± 5.5
	varying	48.9 ± 8.7				
API	fixed	0.795 ± 0.570	0.369 ± 0.465	0.048 ± 0.057	0.079 ± 0.027	0.327 ± 0.097
	varying	0.013 ± 0.015				

Table 1: Values of Signal to Noise Ratio (SNR) and Amari Performance Index (API) for jitterless data with $N = 3$, for various fixed values of λ , as well as for the varying-lambda strategy detailed in the text. While the best fixed value, $\lambda = 0.1$, yields decent results, the results using a varying value of λ are consistently better, with a large margin.

5 Discussion

IPA has a parameter, λ , which controls the relative weights given to the optimization of the PLF matrix and to the penalization of close-to-singular solutions. Our optimization procedure starts with a high value of λ , which is lowered as the optimization progresses. We confirmed that this variation of the parameter’s value is necessary: the quality of the results is noticeably degraded if λ is kept at a constant value, no matter how high or low it is. Table 1 confirms this: while $\lambda = 0.1$, the best fixed value, yields decent results, the results with a varying value of λ are considerably better.

Furthermore, although the final epoch in the optimization is not done with $\lambda = 0$, we have verified that the results are virtually the same as if we had used $\lambda = 0$ at the last epoch.

The above paragraph illustrates something already mentioned in Sec. 3.4: separation of phase-locked sources is a non-trivial change from ICA because there are wrong, singular solutions that yield exactly the same values of the PLF matrix as the correct non-singular solutions. Our approach to distinguish these two types of solutions consists in adding a term depending on the determinant of the matrix \mathbf{W} . This approach works correctly, as our results show. However, it is perhaps inelegant to do this through matrix \mathbf{W} , instead of doing it directly through the estimated sources. It would be preferable to replace this term with one

depending directly on the estimated sources.

The size of the optimization variable, \mathbf{W} , is N^2 ; there are N constraints on this variable, yielding $N(N-1)$ independent parameters. This means that the IPA algorithm is quadratic in the number of sources N . This is the main reason why we do not present results for $N > 5$; while running IPA on 100 datasets with $N = 2$ takes a few hours, doing so for $N = 5$ takes several days.

The results that we obtained show that IPA can separate perfectly locked MEG-like sources. However, while the phase locking in the jitterless pseudo-real MEG data is perfect, in real MEG data it will probably be less than perfect. This is the reason why we also studied data with phase jitter, which have pairwise PLFs smaller than 1. The results indicate that IPA has some robustness to PLFs smaller than 1, but the sources still need to exhibit considerable phase locking for the separation to be accurate; weaker synchrony results only in partial separation. Note, however, that the partially separated data are, usually, still closer to the true sources than the original mixtures.

The comments made in the previous paragraph raise an additional optimization challenge: if the true sources have PLFs smaller than 1, optimization of the objective function in Eq. (6) can lead to overfitting. The results presented here show that IPA has some robustness to sources which have a PLF smaller than 1, while being stationary (since the phase jitter is stationary, the distribution of the PLF does not vary with time). In real-world cases, it is likely that the PLF is non-stationary: for example, some sources may be phase-locked at the start of the observation period and not phase-locked at its end. While simple techniques such as windowing can be devised to tackle smaller time intervals where stationarity is (almost) verified, one would still need to find a way to integrate the information from different intervals. Such integration is out of the scope of this paper.

One interesting extension of this work would be the separation of specific types of systems, such as van der Pol oscillators [27]. For those, fully entrained oscillators may even present a $\text{PLF} < 1$, and a different measure of synchrony, tailored to those oscillators, may need to be used. Such a study would fall out of the scope of this paper. Nevertheless, it is expected that additional knowledge of the oscillator type can be exploited to improve the algorithm's performance or its robustness to deviations from the ideal case.

One can derive a relationship between additive Gaussian noise (*e.g.* from the sensors) and the phase jitter used throughout this paper. Fig. 9 depicts, in the complex plane, a sample of a noiseless signal $x(t) \equiv a(t)e^{i\phi(t)}$, to which complex noise $n(t)$ is added to form the noisy signal $x_n \equiv a(t)e^{i\phi(t)} + n(t)$.¹¹

¹¹In most real applications, one will be dealing with models consisting of real signals to which real-valued noise is added. However, the linearity of the Hilbert Transform allows the same type of analysis for that case as for the case of complex signals with complex additive noise which is considered here.

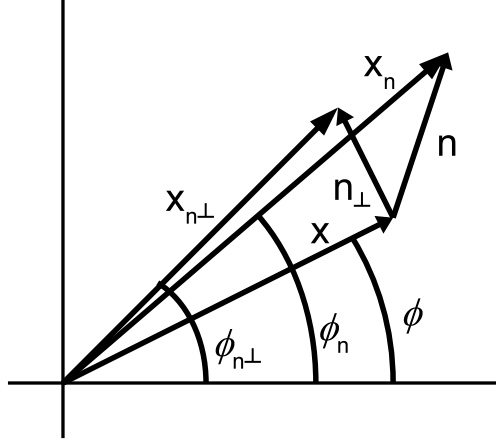


Figure 9: Diagram illustrating the relationship between phase jitter and additive noise. A single time sample is shown, and the time argument has been dropped for simplicity.

That figure also shows $n_{\perp}(t)$, which is the projection of $n(t)$ on the direction orthogonal to $x(t)$, and $x_{n\perp}(t) \equiv x(t) + n_{\perp}(t)$. Also depicted are $\phi(t)$, $\phi_n(t)$ and $\phi_{n\perp}(t)$, which are defined as the phases of $x(t)$, $x_n(t)$ and $x_{n\perp}(t)$, respectively.

It can be easily shown that, if $|n(t)| \ll |x(t)| = a(t)$, then $\phi_n(t) \approx \phi_{n\perp}(t) \approx \phi(t) + \frac{n_{\perp}(t)}{a(t)}$ [32]. This is an important relationship, because it shows that, under additive noise, portions of the signal with a large amplitude will have a better phase estimate than portions with a small amplitude, in which even small amounts of additive noise can severely disrupt the phase estimation. We thus believe that the PLF quantity, while attractive and elegant in theory, and despite working well with low amounts of additive noise [14], will probably need to be changed to factor in the amplitude in an appropriate way to deal with applications where considerable amounts of additive noise are present.

6 Conclusion

We have shown that Independent Phase Analysis (IPA) can successfully separate phase-locked sources from linear mixtures in pseudo-real MEG data. We showed that IPA tolerates deviations from the ideal case, yielding excellent results for low amounts of phase jitter, and that it exhibits some robustness to moderate amounts of phase jitter. We also showed that it can handle numbers of sources up to $N = 5$. We believe that these results bring us closer to the goal of successfully separating phase-locked sources in real-world signals.

References

1. Pikovsky A, Rosenblum M, Kurths J: *Synchronization: A universal concept in nonlinear sciences*. Cambridge Nonlinear Science Series, Cambridge University Press 2001.
2. Palva JM, Palva S, Kaila K: **Phase Synchrony among Neuronal Oscillations in the Human Cortex**. *Journal of Neuroscience* 2005, **25**(15):3962–3972.
3. Schoffelen JM, Oostenveld R, Fries P: **Imaging the human motor system’s beta-band synchronization during isometric contraction**. *NeuroImage* 2008, **41**:437–447.
4. Uhlhaas PJ, Singer W: **Neural Synchrony in Brain Disorders: Relevance for Cognitive Dysfunctions and Pathophysiology**. *Neuron* 2006, **52**:155–168.
5. Nunez PL, Srinivasan R, Westdorp AF, Wijesinghe RS, Tucker DM, Silberstein RB, Cadusch PJ: **EEG Coherency I: statistics, reference electrode, volume conduction, Laplacians, cortical imaging, and interpretation at multiple scales**. *Electroencephalography and clinical Neurophysiology* 1997, **103**:499–515.
6. Vigário R, Särelä J, Jousmäki V, Härmäläinen M, Oja E: **Independent Component Approach to the Analysis of EEG and MEG Recordings**. *IEEE Trans. On Biom. Eng.* 2000, **47**(5):589–593.
7. Hyvärinen A, Karhunen J, Oja E: *Independent Component Analysis*. John Wiley & Sons 2001.
8. Akhtar M, Mitsuhashi W, James C: **Employing spatially constrained ICA and wavelet denoising for automatic removal of artifacts from multichannel EEG data**. *Signal Processing* 2012, **92**:401–416.
9. de Vos M, de Lathauwer L, van Huffel S: **Spatially constrained ICA algorithm with an application in EEG processing**. *Signal Processing* 2011, **91**:1963–1972.
10. Lee D, Seung H: **Algorithms for Non-negative Matrix Factorization**. In *Advances in Neural Information Processing Systems, Volume 13* 2001:556–562.
11. Chan TH, Ma WK, Chi CY, Wang Y: **A Convex Analysis Framework for Blind Separation of Non-Negative Sources**. *IEEE Transactions on Signal Processing* 2008, **56**:5120–5134.
12. de Frein R, Rickard S: **The Synchronized Short-Time-Fourier-Transform: Properties and Definitions for Multichannel Source Separation**. *IEEE Transactions on Signal Processing* 2011, **59**:91–103.
13. Hosseini S, Deville Y, Saylani H: **Blind separation of linear instantaneous mixtures of non-stationary signals in the frequency domain**. *Signal Processing* 2009, **89**:819–830.
14. Almeida M, Schleimer JH, Bioucas-Dias J, Vigário R: **Source Separation and Clustering of Phase-Locked Subspaces**. *IEEE Transactions on Neural Networks* 2011, **22**(9):1419–1434.
15. Almeida M, Bioucas-Dias J, Vigário R: **Independent Phase Analysis: Separating Phase-Locked Subspaces**. In *Proceedings of the Latent Variable Analysis Conference* 2010.
16. Ziehe A, Müller KR: **TDSEP - an efficient algorithm for blind separation using time structure**. In *International Conference on Artificial Neural Networks* 1998:675–680.
17. Torrence C, Compo GP: **A Practical Guide to Wavelet Analysis**. *Bull. of the Am. Meteorological Society* 1998, **79**:61–78.
18. Oppenheim AV, Schafer RW, Buck JR: *Discrete-Time Signal Processing*. Prentice-Hall International Editions 1999.
19. Quyen MLV, Foucher J, Lachaux JP, Rodriguez E, Lutz A, Martinerie J, Varela FJ: **Comparison of Hilbert transform and wavelet methods for the analysis of neuronal synchrony**. *Journal of Neuroscience Methods* 2001, **111**:83–98.
20. Varela F, Lachaux JP, Rodriguez E, Martinerie J: **The Brainweb: phase synchronization and large-scale integration**. *Nature Reviews Neuroscience* 2001, **2**:229–239.
21. Niedermeyer E, da Silva FHL: *Electroencephalography: basic principles, clinical applications, and related fields*. Lippicott Williams and Wilkins 2005.
22. Nunez P, Srinivasan R: *Electric Fields of the Brain: the neurophysics of EEG*. Oxford University Press 2006.
23. Gold B, Oppenheim AV, Rader CM: **Theory and Implementation of the Discrete Hilbert Transform**. In *Discrete Signal Processing*. Edited by Rabiner LR, Rader CM 1973.

24. Breakspear M, Heitmann S, Daffertshofer A: **Generative models of cortical oscillations: neurobiological implications of the Kuramoto model.** *Frontiers in Human Neuroscience* 2010, **4**(190).
25. Kuramoto Y: *Chemical Oscillations, Waves and Turbulences.* Springer Berlin 1984.
26. Strogatz S: *Nonlinear Dynamics and Chaos.* Westview Press 2000.
27. Izhikevich E: *Dynamic Systems in Neuroscience.* MIT Press 2007.
28. Almeida M, Vigário R, Bioucas-Dias J: **The role of whitening for separation of synchronous sources.** In *Proceedings of the Latent Variable Analysis Conference* 2012 (accepted).
29. Eichele T, Rachakonda S, Brakedal B, Eikeland R, Calhoun VD: **EEGIFT: Group Independent Component Analysis for Event-Related EEG Data.** *Computational Intelligence and Neuroscience* 2011, **2011**:1–9.
30. Vigário R, Jousmäki V, Hämäläinen M, Hari R, Oja E: **Independent Component Analysis for identification of artifacts in Magnetoencephalographic recordings.** In *Advances in NIPS* 1997.
31. Amari S, Cichocki A, Yang HH: **A new learning algorithm for blind signal separation.** In *Advances in NIPS, Volume 8* 1996:757–763.
32. Carlson A, Crilly P, Rutledge J: *Communication Systems: An Introduction to Signals and Noise in Electrical Communication.* McGraw-Hill 2001.

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 PLMF as a replacement for IPA, but rather as a different approach to a similar problem. As we will show, PLMF 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 PLMF. The current limitations of PLMF 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

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

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 PLMF and present a summary table of PLMF 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}

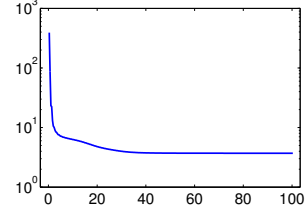


Figure 1: Evolution of the cost function as a function of the iteration for dataset 2.

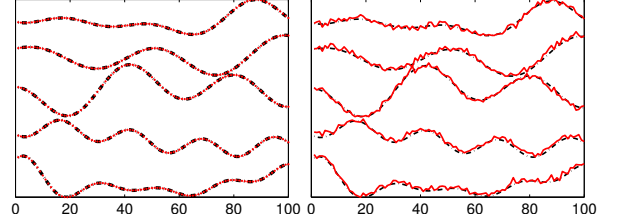


Figure 2: Visual comparison of the estimated amplitudes $\hat{\mathbf{A}}$ (red dots on the left side, red lines on the right side) with the true amplitudes \mathbf{A} (black lines). (Left) Results for dataset 1: the three estimated and true amplitudes coincide perfectly. (Right) Results for dataset 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.

ESTIMATION OF THE COMMON OSCILLATION FOR PHASE LOCKED MATRIX FACTORIZATION

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

¹*Institute of Telecommunications, Instituto Superior Técnico, Lisbon, Portugal*

²*Adaptive Informatics Research Centre, Aalto University School of Science, Finland*
{malmeida,bioucas}@lx.it.pt, ricardo.vigario@aalto.fi

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. PMLF separates available data into the mixing matrix and the sources; the sources are then decomposed into amplitude and phase components. Previously, PMLF 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 PMLF where this assumption is no longer needed – the oscillatory component can be estimated alongside all the other variables, thus making PMLF much more applicable to real-world data. Furthermore, the optimization procedures in the original PMLF 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:

¹In 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.

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 circumventing this issue.

2.2 Phase-Locking Factor

Let $\phi_j(t)$ and $\phi_k(t)$, for $t = 1, \dots, T$, be the time-dependent phases of signals j and k . The real-valued²

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

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

where $\langle \cdot \rangle$ is the time average operator, and $i = \sqrt{-1}$. 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 indeterminacy 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 $j = 1, \dots, N$ and $t = 1, \dots, T$. 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 $\mathbf{S} = \mathbf{A} \odot \Phi$, 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 Φ is a N by T complex-valued matrix with its (j, t) element defined as $\Phi_j(t) \equiv e^{i(\text{angle}(s_j(t)))} \equiv e^{i\phi_j(t)}$.

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 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, Φ can be decomposed as

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

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 $\text{rank}(\Phi) = 1$, 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 $\phi_j(t) = \text{angle}(z_j) + \text{angle}(f_t)$, 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:

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

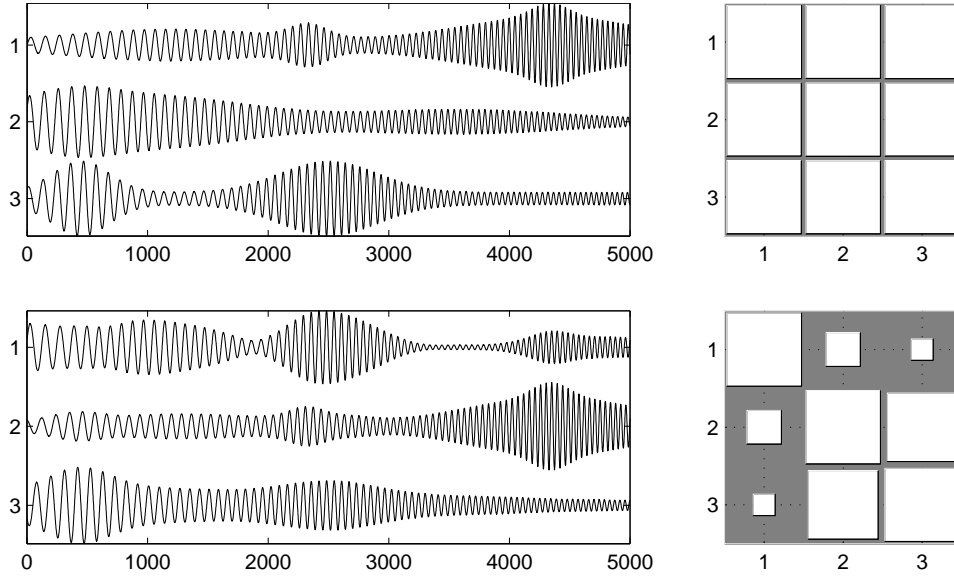


Figure 1: (Top row) Three sources (left) and PLFs between them (right). (Bottom row) Three mixed signals (left) and PLFs between them (right). On the right column, the area of the square in position (i, j) is proportional to the PLF between the signals i and j . Therefore, large squares represent PLFs 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 $\frac{1}{2} \|\mathbf{X} - \mathbf{M}(\mathbf{A} \odot (\mathbf{z}\mathbf{f}^T))\|_F^2$, 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

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

- 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 indeterminacy, 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 op-

timizing 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 \mathbf{M}

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

$$\min_{\mathbf{m}} \frac{1}{2} \left\| \begin{bmatrix} \mathcal{R}(\mathbf{x}) \\ I(\mathbf{x}) \end{bmatrix} - \begin{bmatrix} \mathcal{R}(\mathbf{R}) \\ I(\mathbf{R}) \end{bmatrix} \mathbf{m} \right\|_2^2 \quad (4)$$

s.t.: $-1 \leq \mathbf{m} \leq +1$,

where $\mathcal{R}(\cdot)$ and $I(\cdot)$ are the real and imaginary parts, \mathbf{I}_P is the P by P identity matrix, and $\mathbf{R} \equiv [\mathbf{S}^T \otimes \mathbf{I}_P]$, 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 $-1 \leq \mathbf{M} \leq +1$ is now clear: it is very simply translated into $-1 \leq \mathbf{m} \leq +1$ after applying the $\text{vec}(\cdot)$ operator, remaining a convex constraint, whereas other constraints would be harder to apply.

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

$$\min_{\mathbf{a}} \frac{1}{2} \left\| \begin{bmatrix} \mathcal{R}(\mathbf{x}) \\ I(\mathbf{x}) \end{bmatrix} - \begin{bmatrix} \mathcal{R}(\mathbf{K}) \\ I(\mathbf{K}) \end{bmatrix} \mathbf{a} \right\|_2^2 \quad (5)$$

³The $\text{vec}(\cdot)$ operator stacks the columns of a matrix into a column vector.

$$\text{s.t.: } \mathbf{a} \geq 0,$$

where $\mathbf{K} \equiv [(\text{Diag}(\mathbf{f}) \otimes \mathbf{M}) \text{Diag}(\mathbf{z}_0)]$, $\mathbf{0}_{(N^2-N)}$ is a column vector of size $(N^2 - N)$, filled with zeros, and $\text{Diag}(\cdot)$ 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 \mathbf{z}

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

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

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 \mathbf{f}

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

$$\min_{\mathbf{f}} \frac{1}{2} \left\| \begin{bmatrix} \mathbf{P} \\ \sqrt{\lambda_t} \mathbf{D}_f \end{bmatrix} \mathbf{f} - \begin{bmatrix} \tilde{\mathbf{x}} \\ \mathbf{0}_{(N-1)} \end{bmatrix} \right\|_2^2 \quad (7)$$

with $\mathbf{P} = \begin{bmatrix} \sum_j m_{1j} z_j \text{Diag}(\mathbf{a}_j) \\ \sum_j m_{2j} z_j \text{Diag}(\mathbf{a}_j) \\ \vdots \\ \sum_j m_{Pj} z_j \text{Diag}(\mathbf{a}_j) \end{bmatrix},$

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

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

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

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

3.2.5 Phase Locked Matrix Factorization

The consecutive cycling of optimizations on \mathbf{M} , \mathbf{A} , \mathbf{z} and \mathbf{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 $\in \{1, 2, \dots, \text{MaxIter}\}$, 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 $, $j = 1, \dots, N$
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 $, $t = 1, \dots, T$
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 \mathbf{X} into a mixing matrix \mathbf{M} , amplitudes \mathbf{A} , and phases \mathbf{z} and \mathbf{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 $\mathbf{X} = \mathbf{M}\mathbf{S}$, with $\mathbf{S} = \mathbf{A} \odot \Phi = \mathbf{A} \odot (\mathbf{z}\mathbf{f}^T)$, taking $N = 2$ and $P = 4$. The number of time samples is $T = 100$. \mathbf{M} is taken as a random matrix with entries uniformly distributed between -1 and +1. We then normalize \mathbf{M} so that the entry with the largest absolute value is ± 1 . Each row of \mathbf{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. \mathbf{z} is generated by uniformly spacing the N sources in the interval $[0, \frac{2\pi}{3}]$.⁴ \mathbf{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 \mathbf{f} has no discontinuities. \mathbf{X} is then generated according to the data model: $\mathbf{X} = \mathbf{M}(\mathbf{A} \odot (\mathbf{z}\mathbf{f}^T))$.

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 \mathbf{M}), while the elements of $\hat{\mathbf{z}}$ are of the form $e^{i\alpha}$ with α taken from the Uniform $([0, \frac{\pi}{2}])$ distribution. The elements of $\hat{\mathbf{f}}$ 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 \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 $\hat{\mathbf{M}}$ (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 \mathbf{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.

⁴This choice of \mathbf{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).

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

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.

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

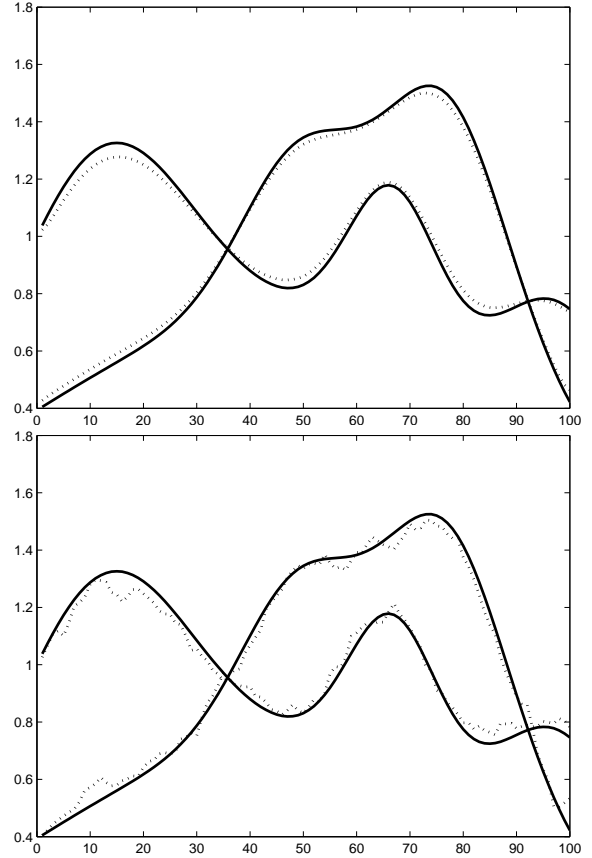


Figure 2: Visual comparison of the estimated amplitudes $\hat{\mathbf{A}}$ (dashed lines) with the true amplitudes \mathbf{A} (solid lines), for a representative dataset, after both are normalized so that they have unit means over the observation period. (Top) Results for dataset 1: the two estimated 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.

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.

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, Φ will have a rank higher than 1 (in most cases, it will have the maximum possible rank, which is N), which makes a representation $\Phi = \mathbf{z}\mathbf{f}^T$ 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 Φ . We are currently testing also

a more general algorithm, which optimizes Φ 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 $\mathbf{X} = \mathbf{M}_1(\mathbf{A}_1 \odot (\mathbf{z}_1 \mathbf{f}_1^T)) = \mathbf{M}_2(\mathbf{A}_2 \odot (\mathbf{z}_2 \mathbf{f}_2^T))$ (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 indeterminacies of BSS apply to PLMF as well, namely the indeterminacies of permutation, scaling, and sign of the estimated sources. There is at least one further indeterminacy: starting from a given solution $\mathbf{X} = \mathbf{M}_1(\mathbf{A}_1 \odot (\mathbf{z}_1 \mathbf{f}_1^T))$, 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 $\mathbf{S}_1 = \mathbf{A}_1 \odot (\mathbf{z}_1 \mathbf{f}_1^T) = \mathbf{A}_2 \odot (\mathbf{z}_2 \mathbf{f}_2^T) = \mathbf{S}_2$, thus the estimated sources are exactly the same.

6 CONCLUSION

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. In Rabiner, L. R. and Rader, C. M., editors, *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.

Phase Locked Matrix Factorization with estimation of the common oscillation

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

¹ Institute of Telecommunications, Instituto Superior Técnico, Lisbon, Portugal

² Department of Information and Computer Science, Aalto University, Finland
{malmeida,bioucas}@lx.it.pt, ricardo.vigario@aalto.fi

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.

Keywords: Matrix Factorization, Phase Synchrony, Phase-Locking, Independent Component Analysis, Blind Source Separation, Convex Optimization

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.³ 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: 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

³ 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.

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 circumventing this issue.

2.2 Phase-Locking Factor

Let $\phi_j(t)$ and $\phi_k(t)$, for $t = 1, \dots, T$, be the time-dependent phases of signals j and k . The real-valued⁴ Phase Locking Factor (PLF) between those two signals is defined as

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

⁴ “Real-valued” is used here to distinguish from other papers, where the absolute value operator is dropped, hence making the PLF a complex quantity (Almeida et al., 2011a).

where $\langle \cdot \rangle$ is the time average operator, and $i = \sqrt{-1}$. 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 indeterminacy 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).

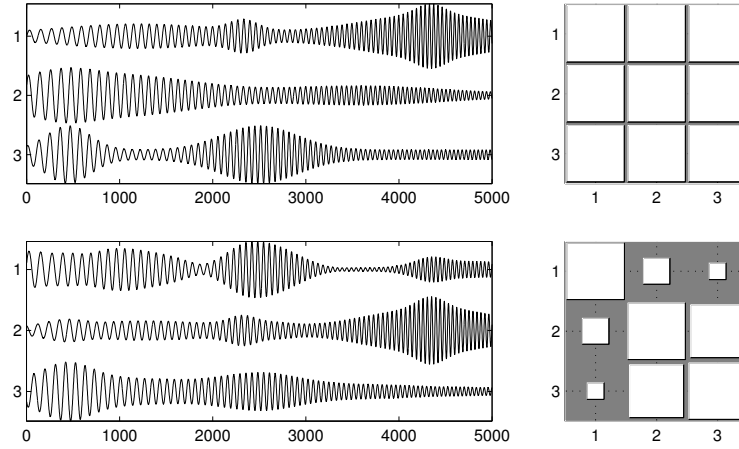


Fig. 1. (Top row) Three sources (left) and PLFs between them (right). (Bottom row) Three mixed signals (left) and PLFs between them (right). On the right column, the area of the square in position (i, j) is proportional to the PLF between the signals i and j . Therefore, large squares represent PLFs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

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.

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 $j = 1, \dots, N$ and $t = 1, \dots, T$. 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 $\mathbf{S} = \mathbf{A} \odot \Phi$, 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 Φ is a N by T complex-valued matrix with its (j, t) element defined as $\Phi_j(t) \equiv e^{i(\text{angle}(s_j(t)))} \equiv e^{i\phi_j(t)}$.

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 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, Φ can be decomposed as

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

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 $\text{rank}(\Phi) = 1$, 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 $\phi_j(t) = \text{angle}(z_j) + \text{angle}(f_t)$, 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:

$$\mathbf{X} \equiv \mathbf{M}\mathbf{S} + \mathbf{N}, \quad (3)$$

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

$$\begin{aligned} \min_{\mathbf{M}, \mathbf{A}, \mathbf{z}, \mathbf{f}} & \frac{1}{2} \|\mathbf{X} - \mathbf{M}(\mathbf{A} \odot (\mathbf{z}\mathbf{f}^T))\|_F^2 + \\ & + \lambda_A \|\mathbf{A}\mathbf{D}_A\|_F^2 + \lambda_f \|\mathbf{D}_f\mathbf{f}\|_2^2, \end{aligned} \quad (4)$$

- 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}_A and \mathbf{D}_f are the first-order difference operators of appropriate size, such that the entry (j, t) of $\mathbf{A}\mathbf{D}_A$ is given by $a_j(t) - a_j(t+1)$, and the k -th entry of $\mathbf{D}_f\mathbf{f}$ 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 indeterminacy, 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. (4) 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. (4) 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.

Optimization on \mathbf{M} If we define $\mathbf{m} \equiv \text{vec}(\mathbf{M})$ and $\mathbf{x} \equiv \text{vec}(\mathbf{X})$ ⁵, then the minimization sub-problem for \mathbf{M} , while keeping all other variables fixed, is equivalent to the following constrained least-squares problem:

$$\begin{aligned} \min_{\mathbf{m}} \quad & \frac{1}{2} \left\| \begin{bmatrix} \mathcal{R}(\mathbf{x}) \\ I(\mathbf{x}) \end{bmatrix} - \begin{bmatrix} \mathcal{R}(\mathbf{R}) \\ I(\mathbf{R}) \end{bmatrix} \mathbf{m} \right\|_2^2 \\ \text{s.t.:} \quad & -1 \leq \mathbf{m} \leq +1, \end{aligned} \quad (5)$$

where $\mathcal{R}(\cdot)$ and $I(\cdot)$ are the real and imaginary parts, \mathbf{I}_P is the P by P identity matrix, and $\mathbf{R} \equiv [\mathbf{S}^T \otimes \mathbf{I}_P]$, 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 $-1 \leq \mathbf{M} \leq +1$ is now clear: it is very simply translated into $-1 \leq \mathbf{m} \leq +1$ after applying the $\text{vec}(\cdot)$ operator, remaining a convex constraint, whereas other constraints would be harder to apply.

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

$$\begin{aligned} \min_{\mathbf{a}} \quad & \frac{1}{2} \left\| \begin{bmatrix} \mathcal{R}(\mathbf{x}) \\ I(\mathbf{x}) \\ \mathbf{0}_{(N^2-N)} \end{bmatrix} - \begin{bmatrix} \mathcal{R}(\mathbf{K}) \\ I(\mathbf{K}) \\ \sqrt{\lambda_{\mathbf{A}}} \mathbf{D}_{\mathbf{A}} \otimes \mathbf{I}_N \end{bmatrix} \mathbf{a} \right\|_2^2 \\ \text{s.t.:} \quad & \mathbf{a} \geq 0, \end{aligned} \quad (6)$$

where $\mathbf{K} \equiv [(\text{Diag}(\mathbf{f}) \otimes \mathbf{M}) \text{Diag}(\mathbf{z}_0)]$, $\mathbf{0}_{(N^2-N)}$ is a column vector of size $(N^2 - N)$, filled with zeros, and $\text{Diag}(\cdot)$ 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.

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

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

$$\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{O}\mathbf{z} - \mathbf{x}\|_2^2 \quad \text{with} \quad \mathbf{O} = \begin{bmatrix} f_1 \mathbf{M} \text{Diag}(\mathbf{a}(1)) \\ f_2 \mathbf{M} \text{Diag}(\mathbf{a}(2)) \\ \vdots \\ f_T \mathbf{M} \text{Diag}(\mathbf{a}(T)) \end{bmatrix}, \quad (7)$$

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 (7) (where the new value of \mathbf{A} should be used).

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

$$\min_{\mathbf{f}} \frac{1}{2} \left\| \begin{bmatrix} \mathbf{P} \\ \sqrt{\lambda_{\mathbf{f}}} \mathbf{D}_{\mathbf{f}} \end{bmatrix} \mathbf{f} - \begin{bmatrix} \tilde{\mathbf{x}} \\ \mathbf{0}_{(N-1)} \end{bmatrix} \right\|_2^2 \quad (8)$$

$$\text{with} \quad \mathbf{P} = \begin{bmatrix} \sum_j m_{1j} z_j \text{Diag}(\mathbf{a}_j) \\ \sum_j m_{2j} z_j \text{Diag}(\mathbf{a}_j) \\ \vdots \\ \sum_j m_{Pj} z_j \text{Diag}(\mathbf{a}_j) \end{bmatrix},$$

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

$$a_j(t) \leftarrow |f_t| a_j(t) \quad \text{and} \quad f_t \leftarrow f_t / |f_t|. \quad (9)$$

Note that unlike the previous case of the optimization for \mathbf{z} , this normalization changes the cost function, in particular the term $\lambda_{\mathbf{f}} \|\mathbf{D}_{\mathbf{f}} \mathbf{f}\|_2^2$. Therefore, there is no guarantee that after this normalization we have found a global minimum for \mathbf{f} .

For this reason, we construct a vector of angles $\beta \equiv \text{angle}(\mathbf{f})$ and minimize the cost function (4) as a function of β , using 20 iterations of Newton's algorithm. Although infinitely many values of β correspond to a given \mathbf{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 (8) 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 (8) and the ensuing normalization (9). On the second and following times that it is run, the result of the previous minimization on \mathbf{f} is used as initial value.

Phase Locked Matrix Factorization The consecutive cycling of optimizations on \mathbf{M} , \mathbf{A} , \mathbf{z} and \mathbf{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  $\in \{1, 2, \dots, \text{MaxIter}\}$ , do
4:   Solve the constrained problem in Eq. (5)
5:   Solve the constrained problem in Eq. (6)
6:   Solve the unconstrained system in Eq. (7)
7:    $a_j(t) \leftarrow |z_j|a_j(t)$  and  $z_j \leftarrow z_j/|z_j|$ ,  $j = 1, \dots, N$ 
8:   if iter = 1
9:     Solve the unconstrained system in Eq. (8)
10:     $a_j(t) \leftarrow |f_t|a_j(t)$  and  $f_t \leftarrow f_t/|f_t|$ ,  $t = 1, \dots, T$ 
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 \mathbf{X} into a mixing matrix \mathbf{M} , amplitudes \mathbf{A} , and phases \mathbf{z} and \mathbf{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 $\mathbf{S} = \mathbf{A} \odot \Phi = \mathbf{A} \odot (\mathbf{zf}^T)$, taking $N = 2$ and $P = 4$. The number of time samples is $T = 100$. \mathbf{M} is taken as a random matrix with entries uniformly distributed between -1 and +1. We then normalize \mathbf{M} so that the entry with the largest absolute value is ± 1 . Each row of \mathbf{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. \mathbf{z} is always equal to $[0, \frac{2\pi}{3}]^T$.⁶ \mathbf{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 \mathbf{f} has no discontinuities. \mathbf{X} is then generated according to the data model: $\mathbf{X} = \mathbf{M}(\mathbf{A} \odot (\mathbf{zf}^T))$.

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 \mathbf{M}), while the elements of $\hat{\mathbf{z}}$ are of the form $e^{i\alpha}$ with α taken from the Uniform($[0, \frac{\pi}{2}]$) distribution. The elements of $\hat{\mathbf{f}}$ are also of the form $e^{i\beta}$, with β uniformly distributed between 0 and 2π .

⁶ This choice of \mathbf{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).

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

- 100 noiseless datasets: 2 sources, 4 sensors, 100 time points, no noise.
- 100 noisy datasets: same as above 1, but with added complex Gaussian white noise as in equation (3). The noise power is such that the Signal-to-Noise Ratio (SNR) of the data is 20 dB.

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 $\hat{\mathbf{M}}$ (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 \mathbf{G} with the sum of the squares of its off-diagonal elements. This criterion is called Signal-to-Interference Ratio (SIR).

Also, $\hat{\mathbf{A}}$ will be compared to \mathbf{A} through visual inspection for one dataset with a SIR close to the average SIR 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 SIR criterion are presented in Table 1. This table also shows results for other choices of $\lambda_{\mathbf{A}}$, which are discussed in section 5.1. 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.

Data	SIR (dB)		
	$\lambda_{\mathbf{A}} = 0.3$	$\lambda_{\mathbf{A}} = 3$	$\lambda_{\mathbf{A}} = 30$
Dataset 1	23.93 \pm 12.72	24.67 \pm 12.27	17.34 \pm 10.46
Dataset 2	24.37 \pm 13.02	25.20 \pm 12.29	19.33 \pm 10.96

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. We used $\lambda_{\mathbf{f}} = 1$ for all the entries of the table.

These results illustrate that PLMF can separate phase-locked sources in both the noiseless and the noisy condition. Furthermore, they show that there is no performance hit due to the presence of a small amount of noise, suggesting that PLMF has good robustness against small perturbations.

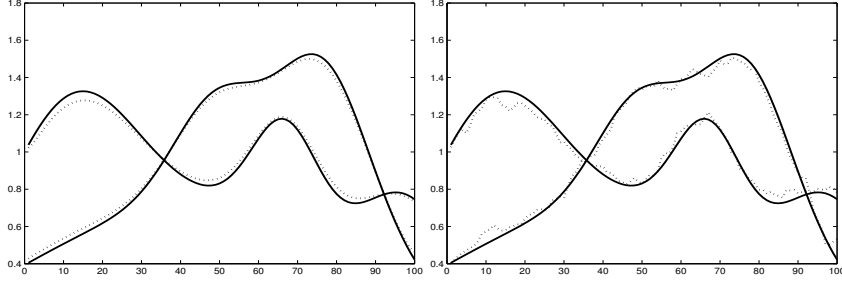


Fig. 2. Visual comparison of the estimated amplitudes $\hat{\mathbf{A}}$ (dashed lines) with the true amplitudes \mathbf{A} (solid lines), for a representative dataset, after both are normalized so that they have unit means over the observation period. (Left) Results for a noiseless dataset: the two estimated amplitudes are close to the true values. Note that the estimated amplitudes have slower variations than the true amplitudes, due to the term with $\lambda_{\mathbf{A}}$. (Right) Results for a noisy dataset: 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. For this figure, the values $\lambda_{\mathbf{A}} = 3$ and $\lambda_{\mathbf{f}} = 1$ were used.

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.

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.

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, Φ will have a rank higher than 1 (in most cases, it will have the maximum possible rank, which is N), which makes a representation $\Phi = \mathbf{z}\mathbf{f}^T$ 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 Φ . We are currently testing also a more general algorithm, which optimizes Φ 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 $\mathbf{X} = \mathbf{M}_1(\mathbf{A}_1 \odot (\mathbf{z}_1 \mathbf{f}_1^T)) = \mathbf{M}_2(\mathbf{A}_2 \odot (\mathbf{z}_2 \mathbf{f}_2^T))$ (i.e., two factor-

izations 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 indeterminacies of BSS apply to PLMF as well, namely the indeterminacies of permutation, scaling, and sign of the estimated sources. There is at least one further indeterminacy: starting from a given solution $\mathbf{X} = \mathbf{M}_1 (\mathbf{A}_1 \odot (\mathbf{z}_1 \mathbf{f}_1^T))$, 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 $\mathbf{S}_1 = \mathbf{A}_1 \odot (\mathbf{z}_1 \mathbf{f}_1^T) = \mathbf{A}_2 \odot (\mathbf{z}_2 \mathbf{f}_2^T) = \mathbf{S}_2$, thus the estimated sources are exactly the same.

5.1 Choice of parameters λ_A and λ_f

The values of the parameters that we chose were somewhat *ad hoc*. However, PLMF is rather robust to the choice of λ_A . Table 1 shows not only the values for $\lambda_A = 3$ (and $\lambda_f = 1$), but also for cases where λ_A is one order of magnitude smaller (0.3) or greater (30). Those results show that the SIR does not change too much when this parameter varies by two orders of magnitude.

λ_A has the effect of penalizing large variations in \mathbf{A} which can be due to the presence of noise. Therefore, if this parameter is too large, the algorithm will underestimate the variations present in the true amplitudes, as illustrated in figure 3. In this figure, the shape of the estimated amplitudes is similar to the shape of the true amplitudes, but the variations are smaller. This effect was already present in figure 2 for $\lambda_A = 3$, but in that case the error was very slight, whereas in figure 3 the effect is very noticeable. This is the reason why $\lambda_A = 30$ yields somewhat lower SIR values when compared to $\lambda_A = 3$: the algorithm will yield estimated amplitudes which are smaller than the true amplitudes.

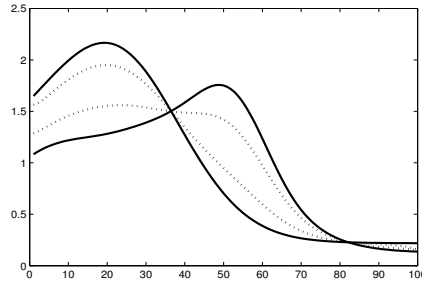


Fig. 3. A typical result of choosing a value of λ_A which is too large (in this case, $\lambda_A = 30$). The true amplitudes are shown in solid lines, whereas estimated amplitudes are shown in dashed lines. The estimated amplitudes are similar in shape to the true ones, but have lower variations, since the penalty term is too strong.

One might then think that the correct way to choose λ_A would be to pick the smallest possible value, which is $\lambda_A = 0$. The results for $\lambda_A = 0.3$, from table 1, might encourage that decision. However, $\lambda_A = 0$ is a poor choice, because the λ_A term has the

indirect effect of preventing \mathbf{A} from exploding to infinity and \mathbf{M} from shrinking to zero, while keeping the product $\mathbf{X} = \mathbf{M}(\mathbf{A} \odot (\mathbf{z}\mathbf{f}^T))$ constant, a situation which causes severe numerical problems.⁷ Therefore, one should pick a small but positive value for $\lambda_{\mathbf{A}}$.

The effect of $\lambda_{\mathbf{f}}$ is easier to understand, because the indirect effect mentioned in the last paragraph is not present. This parameter has a much smaller effect if \mathbf{f} is smooth, as is the case studied in this paper. A non-zero value helps with numerical conditioning of the problem in the presence of noise, because it prevents the fast variations of the noise from contaminating the estimated \mathbf{f} . However, in contrast to the poor choice $\lambda_{\mathbf{A}} = 0$ discussed in the previous paragraph, $\lambda_{\mathbf{f}} = 0$ is a perfectly valid choice.

6 CONCLUSION

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.

⁷ These numerical problems are the reason why no results for $\lambda_{\mathbf{A}} = 0$ are shown in this paper.

Bibliography

- [Almeida et al., 2010]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 et al., 2011a]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 et al., 2011b]Almeida, M., Vigario, R., and Bioucas-Dias, J. (2011b). Phase locked matrix factorization. In *Proc. of the EUSIPCO conference*.
- [Ben-Israel and Greville, 2003]Ben-Israel, A. and Greville, T. (2003). *Generalized inverses: theory and applications*. Springer-Verlag.
- [Boyd and Vandenberghe, 2004]Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- [Gold et al., 1973]Gold, B., Oppenheim, A. V., and Rader, C. M. (1973). Theory and implementation of the discrete hilbert transform. In Rabiner, L. R. and Rader, C. M., editors, *Discrete Signal Processing*.
- [Hyvärinen et al., 2001]Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons.
- [Lee and Seung, 2001]Lee, D. and Seung, H. (2001). Algorithms for non-negative matrix factorization. In *Advances in NIPS*, volume 13, pages 556–562.
- [Nunez et al., 1997]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 et al., 2005]Palva, J. M., Palva, S., and Kaila, K. (2005). Phase synchrony among neuronal oscillations in the human cortex. *J. of Neuroscience*, 25(15):3962–3972.
- [Pikovsky et al., 2001]Pikovsky, A., Rosenblum, M., and Kurths, J. (2001). *Synchronization: A universal concept in nonlinear sciences*. Cambridge Nonlinear Science Series. Cambridge University Press.
- [Schoffelen et al., 2008]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 and Compo, 1998]Torrence, C. and Compo, G. P. (1998). A practical guide to wavelet analysis. *Bull. of the Am. Meteorological Society*, 79:61–78.
- [Uhlhaas and Singer, 2006]Uhlhaas, P. J. and Singer, W. (2006). Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology. *Neuron*, 52:155–168.
- [Vigário et al., 2000]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 and Müller, 1998]Ziehe, A. and Müller, K.-R. (1998). TDSEP - an efficient algorithm for blind separation using time structure. In *Proc. of ICANN*, pages 675–680.

The role of whitening for separation of synchronous sources

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

miguel.almeida@lx.it.pt, ricardo.vigario@aalto.fi, jose.bioucas@lx.it.pt

¹Institute of Telecommunications, Instituto Superior Técnico, Lisbon, Portugal

²Adaptive Informatics Research Centre, Aalto University School of Science, Finland

Abstract. The separation of synchronous sources (SSS) is a relevant problem in the analysis of electroencephalogram (EEG) and magnetoencephalogram (MEG) synchrony. Previous experimental results, using pseudo-real MEG data, showed empirically that prewhitening improves the conditioning of the SSS problem. Simulations with synthetic data also suggest that the mixing matrix is much better conditioned after whitening is performed. Unlike in Independent Component Analysis (ICA), synchronous sources can be correlated. Thus, the reasoning used to motivate whitening in ICA is not directly extendable to SSS. In this paper, we analytically derive a tight upper bound for the condition number of the equivalent mixing matrix after whitening. We also present examples with simulated data, showing the correctness of this bound on sources with sub- and super-gaussian amplitudes. These examples further illustrate the large improvements in the condition number of the mixing matrix obtained through prewhitening, thus motivating the use of prewhitening in real applications.

Keywords: whitening, source separation, independent component analysis (ICA), synchrony, phase-locking factor (PLF), condition number

1 Introduction

Research on the topic of synchrony has gained momentum in recent years. It can be studied under an elegant mathematical framework applicable to many different fields such as laser interferometry, the pull of interstellar objects, and the human brain [9]. Synchrony is believed to play an important role in the interaction of distinct brain regions. For example, a muscle’s electromyogram oscillates coherently with several brain regions, when a person is involved in a motor task [8, 10]. Memorization, learning, autism, Alzheimer’s, Parkinson’s, and epilepsy are examples of neuroscience topics associated with synchrony [11].

Inference about the synchrony of the networks present in the brain (or other real-world systems) requires access to the dynamics of the individual oscillators (the “sources”). However, in the brain’s electroencephalogram (EEG) and

* Corresponding author.

magnetoencephalogram (MEG), the signals from individual oscillators are not directly measurable; one has only access to a superposition of the sources. This is known as the “cross-talk effect” in the field of EEG and MEG research [7]. In this case, spurious synchrony occurs, as has been shown both empirically and analytically [2].

Reversing this superposition is usually called blind source separation (BSS). Usually, it is assumed that the mixing process is linear and instantaneous, a valid approximation in, *e.g.*, brain signals [12]. Let the vector of sources be denoted by $\mathbf{s}(t)$ and the vector of measurements by $\mathbf{y}(t)$. They are related through the model $\mathbf{y}(t) = \mathbf{M}\mathbf{s}(t)$, where \mathbf{M} is a real-valued mixing matrix. The BSS problem has infinitely many solutions. Therefore, assumptions are necessary to adequately pose the problem, such as the independence of the sources, as in Independent Component Analysis (ICA) [6]. However, in the case discussed here, independence is not a valid assumption, because synchronous sources are highly dependent.

We have previously introduced two algorithms to perform Synchronous Source Separation (SSS): Independent Phase Analysis (IPA), a data-driven approach [2], and Phase Locked Matrix Factorization (PLMF), a model-driven approach [3]. Furthermore, we have empirically verified, both with simulated data [2] and with pseudo-real MEG data [1], that prewhitening the data severely improves the quality of the results obtained with these algorithms. However, those empirical findings had no theoretical support. The goal of this paper is to study why prewhitening improves the results of SSS algorithms. We will derive a tight upper bound for the condition number of the problem after prewhitening. We will also present experimental evidence that corroborate this analytical result.

2 Background

2.1 Phase-Locking Factor

Let $\phi_j(t)$ and $\phi_k(t)$, for $t = 1, \dots, T$, be the time-dependent phases of signals j and k . The real-valued Phase Locking Factor (PLF) between those signals is

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

where $\langle \cdot \rangle$ is the time average operator, and $i = \sqrt{-1}$. Note that $0 \leq \varrho_{jk} \leq 1$. Importantly, the value $\varrho_{jk} = 1$ corresponds to two signals that are fully synchronized: their phase lag, defined as $\phi_j(t) - \phi_k(t)$, is constant. The value $\varrho_{jk} = 0$ is attained if $\phi_j(t) - \phi_k(t)$ is uniformly distributed in $[0, 2\pi)$. 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 , $\varrho_{jj} = 1$.

2.2 Whitening

Assume that there is an N by T source matrix \mathbf{S} , such that its (j, t) element is the j -th complex-valued source at time t , $s_j(t)$. Each component of $\mathbf{s}(t)$ is

assumed to have zero mean, *i.e.*, $E[\mathbf{s}(t)] = \mathbf{0}$. We also assume that these sources are unknown, but that we can observe a set of measurements $\mathbf{y}(t)$, which are obtained from the sources through $\mathbf{y}(t) = \mathbf{M}\mathbf{s}(t)$ (note that $\mathbf{y}(t)$ also has zero mean), where \mathbf{M} is a square real-valued matrix with full rank.¹ If one also stores successive samples of $\mathbf{y}(t)$ in a matrix \mathbf{Y} , then $\mathbf{Y} = \mathbf{M}\mathbf{S}$.

Whitening is a process which involves multiplying the data $\mathbf{y}(t)$ by a square matrix \mathbf{B} , such that the resulting vector, $\mathbf{z}(t) \equiv \mathbf{B}\mathbf{y}(t)$, has as covariance the identity matrix. There are infinitely many possible matrices \mathbf{B} which can achieve this; one possibility² is to have $\mathbf{B} = \mathbf{C}_{\mathbf{Y}}^{-1/2}$, where $\mathbf{C}_{\mathbf{Y}}$ is the covariance matrix of $\mathbf{y}(t)$. The original BSS problem $\mathbf{Y} = \mathbf{M}\mathbf{S}$ is thus transformed into an equivalent one $\mathbf{Z} = \mathbf{B}\mathbf{M}\mathbf{S}$; $\mathbf{B}\mathbf{M}$ is called the equivalent mixing matrix.

The BSS community, in particular the users of ICA, have advocated the use of whitening as a preprocessing step [6], because if the sources $\mathbf{s}(t)$ have the identity matrix as their covariance matrix (which is always true if they are independent, up to trivial scalar factors), then the equivalent mixing matrix $\mathbf{B}\mathbf{M}$ is necessarily an orthogonal matrix. This means that ICA algorithms can restrict themselves to finding an orthogonal matrix, which makes the ICA problem considerably easier [6]. However, in SSS the sources are highly dependent.

2.3 Condition number

It is well known that the difficulty of solving linear inverse problems, such as ICA and SSS, can be roughly characterized by the condition number of matrix \mathbf{M} [4]. The condition number of a matrix \mathbf{M} is defined³ as the quotient $\rho = \frac{\sigma_{max}}{\sigma_{min}}$, where σ_{max} is the largest singular value of \mathbf{M} and σ_{min} is its smallest singular value. 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 [4].

The condition number of a BSS problem depends on the unknown matrix \mathbf{M} . In ICA, after prewhitening the inverse problem has $\rho = 1$ [6]. Such is not the case for SSS; however, we will show that an upper bound for this condition number can be derived using prewhitening. We also show experimentally that large improvements on the condition number can be obtained through this process.

3 Upper bound for condition number after prewhitening

3.1 Notation and Assumptions

Let \mathbf{S} denote a complex-valued N -by- T matrix with the value of the sources, where the (j, t) element of \mathbf{S} contains $s_j(t)$. We decompose $s_j(t) \equiv a_j(t)e^{i\phi_j(t)}$

¹ This reasoning can be easily extended to \mathbf{M} having more rows than columns [1, 6].

² In this paper, square roots are taken only of Hermitian positive semidefinite matrices. If $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^H$ is the eigendecomposition of \mathbf{A} , we define $\mathbf{A}^{1/2} \equiv \mathbf{V}\mathbf{D}^{1/2}\mathbf{V}^H$.

³ Other definitions of condition number exist. The one presented here is quite common, and will be used throughout the paper.

where $i = \sqrt{-1}$, $a_j(t)$ is the real-valued, non-negative amplitude, and $\phi_j(t)$ is a real number in the interval $[0, 2\pi)$, called phase. The amplitudes of each of the N sources are considered random variables A_j , *i.e.*, each time point $a_j(t)$ is i.i.d., drawn from a certain probability density distribution. The phases of the sources are also considered random variables, although not independent of each other, as detailed below.

Our goal is to study the simplest case applicable to SSS. We make the following assumptions:

- A_j is independent of A_k for $j \neq k$;
- A_j is independent of ϕ_k for any j and k , including for $j = k$;
- All A_j have the same distribution, which is generally denoted as A ;
- ϕ_j and ϕ_k have maximum PLF, *i.e.* they have a constant phase lag;
- The previous point implies that $\phi_j(t) = \phi_j(1) + \Phi(t)$ for all j and t .

We assume that $\Phi(t)$ is uniformly distributed in $[0, 2\pi)$.

Note that this is still a harder problem than ICA, because $\phi_j(t)$ and $\phi_k(t)$ are strongly dependent. Nevertheless, algorithmic solutions exist that can extract the matrix \mathbf{S} using only information from the observations $\mathbf{Y} = \mathbf{MS}$ [2, 3].

3.2 Upper bound

Multiplying the data by $\mathbf{B} = \mathbf{C}_{\mathbf{Y}}^{-1/2}$ will, in general, result in an equivalent mixing matrix \mathbf{BM} which is complex, even if \mathbf{M} is real. This is a disadvantage, if one aims to use algorithms which search for real separation matrices [2, 3]. To take this into account, one can consider the two following formulations, which are equivalent to $\mathbf{Y} = \mathbf{MS}$ with the constraint of real \mathbf{M} :

$$\begin{bmatrix} \mathbf{Y}_R \\ \mathbf{Y}_I \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{S}_R \\ \mathbf{S}_I \end{bmatrix} \text{ or } [\mathbf{Y}_R \ \mathbf{Y}_I] = \mathbf{M} [\mathbf{S}_R \ \mathbf{S}_I], \quad (2)$$

where $\mathbf{S}_R \equiv \text{Real}(\mathbf{S})$, $\mathbf{S}_I \equiv \text{Imag}(\mathbf{S})$, and similarly for \mathbf{Y}_R and \mathbf{Y}_I . $\mathbf{0}$ is a matrix filled with zeros with the same size as \mathbf{M} .

Although both formulations allow the derivations done below, the equivalent mixing matrix is, on average, farther from the bound (and thus, better conditioned) on the second case; therefore, we use that formulation and define $\mathbf{S}_{RI} \equiv [\mathbf{S}_R \ \mathbf{S}_I]$ and similarly for \mathbf{Y}_{RI} .⁴

In this formulation, prewhitening involves multiplying the new data \mathbf{Y}_{RI} by a new whitening matrix $\mathbf{B} \equiv \mathbf{C}_{\mathbf{Y}_{RI}}^{-1/2}$ such that $\mathbf{Z}_{RI} \equiv \mathbf{B}\mathbf{Y}_{RI} = \mathbf{BMS}_{RI}$ has as correlation the identity matrix. Since

$$\mathbf{C}_{\mathbf{Z}_{RI}} = \mathbf{BMC}_{\mathbf{S}_{RI}}\mathbf{M}^T\mathbf{B}^H = (\mathbf{BMC}_{\mathbf{S}_{RI}}^{1/2})(\mathbf{BMC}_{\mathbf{S}_{RI}}^{1/2})^H = \mathbf{I}, \quad (3)$$

⁴ The second formulation has a trickier interpretation, since $[\mathbf{S}_R \ \mathbf{S}_I]$ is no longer a matrix whose columns are realizations of one random variable. Consequently, the term “correlation matrix” for $\mathbf{C}_{\mathbf{S}_{RI}}$ is somewhat abusive. Formally, we define the “correlation matrix” of \mathbf{S}_{RI} as $\mathbf{C}_{\mathbf{S}_{RI}} \equiv \frac{\mathbf{S}_{RI}\mathbf{S}_{RI}^T}{2T}$; technically, it corresponds to the correlation matrix of a random variable which can take the value $\text{Real}(\mathbf{s})$ and $\text{Imag}(\mathbf{s})$ with equal probability. Similar considerations hold for $\mathbf{C}_{\mathbf{Y}_{RI}}$ and $\mathbf{C}_{\mathbf{Z}_{RI}}$.

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

We can now study the singular values of the equivalent mixing matrix $\mathbf{B}\mathbf{M}$. It holds that $\mathbf{B}\mathbf{M} = \mathbf{R}\mathbf{C}_{\mathbf{S}_{RI}}^{-1/2}$, and that the singular values of $\mathbf{R}\mathbf{C}_{\mathbf{S}_{RI}}^{-1/2}$ are the same as those of $\mathbf{C}_{\mathbf{S}_{RI}}^{-1/2}$, since they differ only by a multiplication by a unitary matrix. Therefore, the conditioning of the equivalent source separation problem can be studied by studying the singular values of $\mathbf{C}_{\mathbf{S}_{RI}}^{-1/2}$.

Note that $\mathbf{C}_{\mathbf{S}_{RI}} = 1/2(\mathbf{C}_{\mathbf{S}_R} + \mathbf{C}_{\mathbf{S}_I})$. Using the assumptions from Section 3.1 yields, for the diagonal elements of $\mathbf{C}_{\mathbf{S}_{RI}}$,

$$[\mathbf{C}_{\mathbf{S}_{RI}}]_{jj} = \frac{1}{2}\mathbb{E}[A^2], \quad (4)$$

whereas the off-diagonal elements of $\mathbf{C}_{\mathbf{S}_{RI}}$ can be shown to be

$$[\mathbf{C}_{\mathbf{S}_{RI}}]_{jk} = \frac{1}{2}\mathbb{E}[A]^2 \cos(\phi_j(1) - \phi_k(1)). \quad (5)$$

Since $\mathbb{E}[A^2] = \text{Var}[A] + \mathbb{E}[A]^2$, for any random variable A , we get

$$\mathbf{C}_{\mathbf{S}_{RI}} = \frac{\text{Var}[A]\mathbf{I} + \mathbb{E}[A]^2\mathbf{F}}{2}, \quad (6)$$

where \mathbf{I} is the identity matrix and $\mathbf{F}_{jk} \equiv \cos(\phi_j(1) - \phi_k(1))$.

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 $\mathbf{F} = \text{Re}(\mathbf{G})$, with $\mathbf{G} \equiv \mathbf{x}\mathbf{x}^H$, where the vector \mathbf{x} has in its j -th component $x_j \equiv e^{i\phi_j(1)}$. \mathbf{G} has simple eigenvalue $\lambda_{\mathbf{G}} = N$ (the number of sources), and an eigenvalue $\lambda_{\mathbf{G}} = 0$ 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 $\mathbf{v}^H = \mathbf{v}^T$. Then,

$$\mathbf{v}^H\mathbf{G}\mathbf{v} = \mathbf{v}^T\mathbf{G}\mathbf{v} = \mathbf{v}^T\mathbf{F}\mathbf{v} + \mathbf{v}^T\text{Im}(\mathbf{G})\mathbf{v} = \mathbf{v}^T\mathbf{F}\mathbf{v}, \quad (7)$$

where $\mathbf{v}^T\text{Im}(\mathbf{G})\mathbf{v} = 0$ because \mathbf{G} is Hermitian, thus its imaginary part is skew-symmetric. The leftmost expression is valued between 0 and N , since those are the smallest and largest eigenvalues of \mathbf{G} . Thus the rightmost expression must also be within those values. Therefore, the eigenvalues of \mathbf{F} obey $0 \leq \lambda_{\mathbf{F}} \leq N$.

With these bounds for $\lambda_{\mathbf{F}}$, one can immediately conclude that the eigenvalues of $\mathbf{C}_{\mathbf{S}_{RI}}$ obey

$$\frac{\text{Var}[A]}{2} \leq \lambda_{\mathbf{C}_{\mathbf{S}_{RI}}} \leq \frac{\text{Var}[A] + N\mathbb{E}[A]^2}{2}. \quad (8)$$

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

$$\rho(\mathbf{B}\mathbf{M}) = \rho(\mathbf{C}_{\mathbf{S}_{RI}}^{-1/2}) = \sqrt{\rho(\mathbf{C}_{\mathbf{S}_{RI}}^{-1})} = \sqrt{\rho(\mathbf{C}_{\mathbf{S}_{RI}})} \leq \sqrt{1 + N \frac{\mathbb{E}[A]^2}{\text{Var}[A]}}. \quad (9)$$

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 full of ones, and its eigenvalues are exactly 0 and N . It is very simple to see that in that case, $\rho(\mathbf{C}_{\mathbf{S}_{RI}}^{-1/2}) = \sqrt{1 + N \frac{\mathbb{E}[A]^2}{\text{Var}[A]}}$ holds.

4 Experiments

The above result is derived for the ideal case, where the assumptions of Section 3.1 are valid. However, in real data these assumptions will never hold, because the number of time points is finite.⁵ Therefore, we now study whether this upper bound expression is useful in practice, using small simulated examples.

We generate each set of data in the following way: the initial phase for each source, $\phi_j(1)$, is randomly drawn from a uniform distribution between 0 and 2π . The common phase oscillation, $\Phi(t)$, is given by $\Phi(t) = \omega t$ with $\omega = 0.02\pi$. The amplitudes $a_j(t)$ are independently drawn from the Gamma probability distribution with unit scale parameter and shape parameter equal to 1, 2, 3. A similar range was used for the Irwin-Hall probability distribution. This corresponds to the sums of one, two, or three Exponential or Uniform distributions, thus representing different values of kurtosis. The mixing matrix \mathbf{M} has each of its elements independently drawn from a Uniform(-1,1) distribution.

Each of these datasets has $T = 10000$ time points and $N = 4$ sources and measurements. We generate 1000 such datasets for each of the six distributions. We then make a scatter plot comparing the condition number of the original mixing matrix \mathbf{M} (in the horizontal axis) with the condition number of the equivalent mixing matrix $\mathbf{B}\mathbf{M}$ (in the vertical axis). Each of these plots also shows the theoretical value of the upper bound from eq. (9), drawn as a horizontal line. These plots are shown in Figure 1.

At first glance it might seem unexpected that some points are slightly above the line of the upper bound. This is justified by the difference between the ideal case with $T = \infty$, which was used to derive the bound, and the experimental case with a finite T . In other words, it is a consequence of using a sample covariance matrix instead of the true covariance matrix. Nevertheless, the fraction of points above the line is very small, as is the vertical gap between those points and the line. As the number of time points T approaches infinity, the fraction of points above the line and their gap tends to zero.

5 Discussion

Several directions can be taken to extend this result. One such direction is to derive an upper bound for cases where the PLF between the sources is smaller than 1. The case of zero PLF includes the ICA case; in that case, the condition

⁵ This is similar to the ICA case, where although independence of the sources is assumed, it is not verified precisely in real cases.

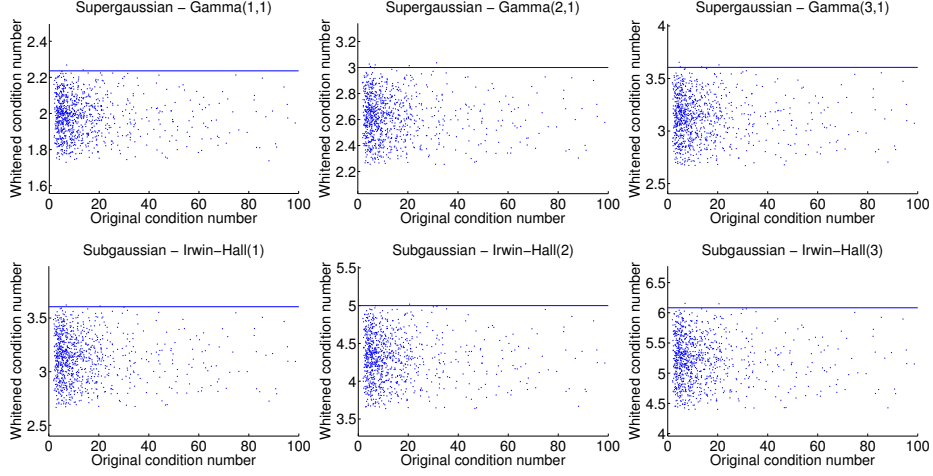


Fig. 1. Experimental confirmation of the upper bound. The condition number of the mixing matrix is displayed on the horizontal axis, and that of the equivalent mixing matrix on the vertical axis, for six different distributions for A . The horizontal line corresponds to the upper bound in eq. (9).

number of the equivalent mixing matrix is known to be 1 [6]. In general, we believe that the upper bound for the $\text{PLF} < 1$ case will be smaller than the one derived here; however, it will probably depend on the specific form of the sources' phases: there may be i.i.d. random phase noise, as we studied previously [1], there may be phase slips [9], or other types of imperfect phase-locking.

Another very important direction is to obtain a result on the probability of a case with a finite number of points T , to have a condition number higher than the bound derived here for $T = \infty$. In other words, it would be interesting to know in anticipation how many points will, on average, end up above the horizontal line in Figure 1. This result would necessarily depend on T , on the distribution of the amplitudes A , and on how the mixing matrix \mathbf{M} is generated.

Yet another useful extension is to remove the assumption that all amplitudes are drawn from the same distribution A . As long as the amplitudes are independent of each other and of the phases, the reasoning used throughout this paper stands, although the mathematical expressions involved become less elegant.

Finally, one could also relax the assumption that the amplitudes and phases are independent, since studies have shown that the power of given brain oscillations may be locked to the phases of other oscillations [5]. This requires the assumption of a specific dependency between the amplitudes and phases.

6 Conclusion

We have derived an upper bound for the condition number of an SSS problem, if whitening is performed as pre-processing. Experimental results confirm the

validity of this upper bound. The main conclusion is that in virtually any situation, it is advantageous to use whitening as a pre-processing step, even when there is a degree of dependence in the sources.

Acknowledgements: *This work was supported by FCT project PEst-OE / EEI / LA0008 / 2011, under internal project DECA-Bio, and by the Academy of Finland through its Centres of Excellence Program 2006-2011.*

References

1. Almeida, M., Bioucas-Dias, J., Vigário, R.: Detection and separation of phase-locked subspaces with phase noise. *Signal Processing* (submitted) (2011)
2. Almeida, M., Schleimer, J.H., Bioucas-Dias, J., Vigário, R.: Source separation and clustering of phase-locked subspaces. *IEEE Transactions on Neural Networks* 22(9), 1419–1434 (2011)
3. Almeida, M., Vigario, R., Bioucas-Dias, J.: Phase locked matrix factorization. In: *Proc. of the EUSIPCO conference* (2011)
4. Bertero, M., Boccacci, P.: *Introduction to Inverse Problems in Imaging*. Taylor & Francis (1998)
5. Canolty, R., Edwards, E., Dalal, S., Soltani, M., Nagarajan, S., Kirsch, H., Berger, M., Barbaro, N., Knight, R.: High gamma power is phase-locked to theta oscillations in human neocortex. *Science* 313, 1626–1628 (2006)
6. Hyvärinen, A., Karhunen, J., Oja, E.: *Independent Component Analysis*. John Wiley & Sons (2001)
7. Nunez, P.L., Srinivasan, R., Westdorp, A.F., Wijesinghe, R.S., Tucker, D.M., Silberstein, R.B., Cadusch, P.J.: EEG coherency I: statistics, reference electrode, volume conduction, laplacians, cortical imaging, and interpretation at multiple scales. *Electroencephalography and clinical Neurophysiology* 103, 499–515 (1997)
8. Palva, J.M., Palva, S., Kaila, K.: Phase synchrony among neuronal oscillations in the human cortex. *Journal of Neuroscience* 25(15), 3962–3972 (April 2005)
9. Pikovsky, A., Rosenblum, M., Kurths, J.: *Synchronization: A universal concept in nonlinear sciences*. Cambridge Nonlinear Science Series, Cambridge University Press (2001)
10. Schoffelen, J.M., Oostenveld, R., Fries, P.: Imaging the human motor system's beta-band synchronization during isometric contraction. *NeuroImage* 41, 437–447 (2008)
11. Uhlhaas, P.J., Singer, W.: Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology. *Neuron* 52, 155–168 (Oct 2006)
12. Vigário, R., Särelä, J., Jousmäki, V., Hämäläinen, M., Oja, E.: Independent component approach to the analysis of EEG and MEG recordings. *IEEE Trans. On Biom. Eng.* 47(5), 589–593 (May 2000)

A comparison of algorithms for separation of synchronous subspaces

Miguel Almeida*, José Bioucas-Dias[†], Ricardo Vigário[‡], Erkki Oja[§]

April 24, 2012

Abstract

Independent Subspace Analysis (ISA) consists in separating sets (subspaces) of dependent sources, with different sets being independent of each other. While a few algorithms have been proposed to solve this problem, they are all completely general in the sense that they do not make any assumptions on the intra-subspace dependency.

In this paper, we address the ISA problem in the specific context of Separation of Synchronous Sources (SSS), *i.e.*, we aim to solve the ISA

*Corresponding author. M. Almeida is with Instituto de Telecomunicações, Instituto Superior Técnico, Technical University of Lisbon, Portugal, and with the Department of Information and Computer Science, Aalto University, Finland. Email: malmeida@lx.it.pt

[†]J. Bioucas-Dias is with Instituto de Telecomunicações, Instituto Superior Técnico, Technical University of Lisbon, Portugal. Email: bioucas@lx.it.pt

[‡]R. Vigário is with the Department of Information and Computer Science, Aalto University, Finland. Email: ricardo.vigario@aalto.fi

[§]E. Oja is with the Department of Information and Computer Science, Aalto University, Finland. Email: erkki.oja@aalto.fi

problem when the intra-subspace dependency is known to be perfect phase synchrony between all sources in that subspace. We compare multiple algorithmic solutions for this problem, by analyzing their performance on an MEG-like dataset.

Keywords: phase-locking, synchrony, source separation, subspaces, Independent Component Analysis (ICA), Independent Subspace Analysis (ISA), magnetoencephalogram (MEG)

1 Introduction

In the human brain, it has been shown that synchrony [19] is associated with important phenomena. For example, there is evidence that autism, Parkinson's, and Alzheimer's disease are associated with a decrease in the synchrony of some brain regions, whereas some types of epilepsy are associated with an anomalous increase in synchrony [23].

It is also known that electrophysiological brain signals, such as electroencephalogram (EEG) or magnetoencephalogram (MEG) signals, are not a direct measurement of the activity of individual brain regions, but rather the result of a superposition of those regions' activities [14]. Due to the low frequencies involved (below 1 kHz), the approximation of a linear and instantaneous mixing process is valid [25].

Since synchrony is an important phenomenon in the human brain, and measurements are the result of a mixing process, we argue that it is important to perform Blind Source Separation (BSS) on these types of measurements,

to be able to extract relevant networks of synchronous activity. If, as a first approximation, one considers those different networks to be independent, then finding the original networks can be cast as an Independent Subspace Analysis (ISA) task, where each one of the mutually independent subspaces has a specific type of internal dependency, namely, phase synchrony.

The goal of this paper is to compare the performance of various algorithms on a partial ISA task (defined in Section 2.3) when the subspaces are composed of synchronous sources. Some of these algorithms are dedicated ISA algorithms, while others are Independent Component Analysis (ICA) algorithms which we nevertheless apply to the partial ISA task. These algorithms are compared on data which is created to mimic brain MEG recordings.

This paper is organized as follows. In Section 2 we provide an overview of the BSS problem, with specific focus on the ICA and ISA problems; the models of these problems are detailed there. In Section 3 we provide an overview of some of the algorithms available to perform ISA. Section 4 compares the performance of these algorithms on a set of MEG-like. These results are discussed in Section 5, and conclusions are drawn in Section 6.

2 Methodology

2.1 Blind Source Separation (BSS)

One important subclass of signal processing problems is the topic of blind source separation (BSS). In BSS, one has access to a set of signals called mea-

measurements $\mathbf{y}(t)$, which are the result of a superposition of another set of signals $\mathbf{s}(t)$, which are called the sources, and which are not directly observable. The goal of BSS is to recover the sources using only the measurements.

Let $\mathbf{s}(t)$ denote a vector of N sources, $\mathbf{s}(t) \equiv (s_1(t), \dots, s_N(t))^T$. Suppose that these sources undergo a linear and instantaneous mixing process, described by a P -by- N mixing matrix \mathbf{M} , resulting in P mixed signals contained in the vector $\mathbf{y}(t) = \mathbf{M}\mathbf{s}(t)$, $\mathbf{y}(t) \equiv (y_1(t), \dots, y_P(t))^T$. The goal of (linear and instantaneous) BSS is therefore to recover the original sources \mathbf{s} using only the observed mixtures \mathbf{y} . In this paper, we will limit ourselves to the case where $P = N$, *i.e.*, the case where the number of sources is equal to the number of observed signals.¹

In general, the BSS problem is ill-posed, in the sense that it has an infinity of solutions. To solve this problem, one must assume extra conditions on the sources \mathbf{s} , the mixing matrix \mathbf{M} , or both. One possibility is to assume that, for each time instant t , the vector $\mathbf{s}(t)$ is an i.i.d. realization of a random vector whose components are statistically independent. This is the fundamental assumption of Independent Component Analysis (ICA) [7, 11]. Other possibilities include assuming that the mixing matrix and the sources are non-negative, which is known as Non-negative Matrix Factorization (NMF) [10, 13], or assuming that the sources have perfect phase

¹The case $P > N$ is called the overdetermined case and can be solved using techniques similar to the ones described here by reducing the problem to an equivalent problem with a square mixing matrix [11]. The case $P < N$ is called the underdetermined case and is, in general, harder to solve [11].

synchrony, which leads to Separation of Synchronous Sources [3].

2.2 Independent Component Analysis (ICA)

In ICA, the goal is to find a demixing matrix \mathbf{W} such that the estimated sources, given by $\mathbf{x} \equiv \mathbf{W}\mathbf{y} = \mathbf{W}\mathbf{M}\mathbf{s}$, are equal to the original sources, which are assumed to be mutually independent. If no other information besides independency is used, some indeterminacies are unavoidable: the estimated sources are equal to the original sources only up to permutation and arbitrary scaling. Equivalently, the gain matrix, defined as $\mathbf{G} \equiv \mathbf{W}\mathbf{M}$, must be a permutation of the rows of a diagonal matrix.

In an informal way, one can say that the ICA problem is identifiable in the following sense: under mild assumptions, if $\mathbf{y} = \mathbf{M}\mathbf{s}$ and one finds a matrix \mathbf{W} such that all components of $\mathbf{x} = \mathbf{W}\mathbf{y}$ are independent, then $\mathbf{x} = \mathbf{s}$ up to permutation and arbitrary scaling [11]. Unfortunately, independence is not easy to measure from a finite sample; therefore, most ICA algorithms replace that criterion with other contrast functions, such as entropy, kurtosis or lagged correlations. A good overview of ICA algorithms and theory can be found in [7, 11].

2.3 Independent Subspace Analysis (ISA)

Independent Subspace Analysis (ISA) is a generalization of ICA, where one has independent random *vectors* instead of independent random (scalar)

variables. Thus, when all the random vectors are uni-dimensional, ISA reduces to ICA.

Since ISA follows a different model than ICA, we likewise introduce new notation: let

$$\mathbf{s} \equiv \begin{bmatrix} \mathbf{s}^1 \\ \mathbf{s}^2 \\ \vdots \\ \mathbf{s}^K \end{bmatrix}, \text{ where } \mathbf{s}^k \equiv \begin{bmatrix} s_1^k \\ \vdots \\ s_{N_k}^k \end{bmatrix}.$$

Note that \mathbf{s} is a random vector composed of K random subvectors \mathbf{s}^k , $k = 1, \dots, K$. The size of subvector \mathbf{s}^k is N_k . The crucial assumption in ISA is that any two random subvectors are independent, *i.e.*, \mathbf{s}^k and \mathbf{s}^l are independent for any $k \neq l$. Naturally, one must have $N_1 + \dots + N_K = N$. Because realizations of the random subvector \mathbf{s}^k span an N_k dimensional subspace, in the following for brevity \mathbf{s}^k is called the k -th subspace of \mathbf{s} .

A naïve approach to perform ISA would be to minimize the mutual information between the various subspaces:

$$\min I(\hat{\mathbf{s}}^1, \dots, \hat{\mathbf{s}}^K)$$

where $\hat{\mathbf{s}}^k$ is the estimate of the k -th subspace. This is a valid approach which has seen some use; however, in general, this approach is a combinatorial optimization problem [6], since one does not know which of the estimated sources should be grouped together [1, 18]. One must then either test all

possible groupings, which grow very quickly with K and rapidly become intractable, or solve a discrete optimization problem by following, *e.g.*, a greedy approach. Nevertheless, this approach is very prone to local minima.

Furthermore, the aforementioned approach involves the computation of the entropy of random vectors of dimension N_k , for $k = 1, \dots, K$. This computation is non-trivial for $N_k \geq 2$ [5], further increasing the computational complexity of this approach. Nevertheless, this approach has been tackled, *e.g.*, by estimating the entropy of multi-dimensional components using minimum spanning trees [18], or using variational Bayes approaches [1].

In this paper we divide ISA into three successive parts, and consider only the first one, similarly to what other groups have done [9, 12, 20]. Thus, the goal is to solve an easier problem than the full ISA problem; we call this easier problem “partial ISA”. This partial problem is not combinatorial.

2.4 Partial ISA and Full ISA

The ISA procedure can be split into three parts. The first part could be called inter-subspace separation, and is the primary focus of this study. The goal of this first part is to obtain a demixing matrix \mathbf{W}_{inter} , such that the gain matrix, $\mathbf{W}_{inter}\mathbf{M}$, is a permutation of a block diagonal matrix with blocks corresponding to the subspaces.

For example, suppose that there are three subspaces ($K = 3$), the first of which has three components ($N_1 = 3$) while the second and third subspaces have two components ($N_2 = N_3 = 2$). In this case, the goal is to find a matrix

\mathbf{W}_{inter} of the form $\mathbf{W}_{inter} \equiv \mathbf{P}\mathbf{B}_{inter}$ where \mathbf{P} is a permutation matrix and \mathbf{B}_{inter} is such that

$$\mathbf{B}_{inter}\mathbf{M} = \begin{bmatrix} \mathbf{U}^1 & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 2} \\ \mathbf{0}_{2 \times 3} & \mathbf{U}^2 & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 2} & \mathbf{U}^3 \end{bmatrix}$$

. Here, $\mathbf{0}_{m \times n}$ is the m -by- n zero matrix, \mathbf{U}^1 is a 3-by-3 invertible matrix, and \mathbf{U}^2 and \mathbf{U}^3 are 2-by-2 invertible matrices. In this case, each entry of the random vector $\mathbf{x}_{inter} \equiv \mathbf{W}_{inter}\mathbf{y}$ is a linear combination of sources from one subspace only.

The second part (which is not studied in this paper) is called permutation compensation or subspace detection. The goal is to group the entries of the random vector \mathbf{x}_{inter} so that the first N_1 entries of \mathbf{x}_{inter} are linear combinations of sources from the first subspace, the next N_2 entries are linear combinations of sources from the second subspace, and so on. This can be achieved by multiplying \mathbf{x}_{inter} by a suitable permutation matrix, \mathbf{Q} . In general, this step involves the use of a measure of dependency or independency, and therefore will depend on the specific type of dependencies between sources in each subspace.

After the subspace detection is completed, one can define

$$\mathbf{y}^1 \equiv \mathbf{Q}_{(1:N_1, :)} \mathbf{x}_{inter} \quad (1)$$

$$\mathbf{y}^2 \equiv \mathbf{Q}_{(N_1+1:N_1+N_2, :)} \mathbf{x}_{inter} \quad (2)$$

$$\vdots \quad (3)$$

$$\mathbf{y}^k \equiv \mathbf{Q}_{(N-N_k+1:N, :)} \mathbf{x}_{inter}, \quad (4)$$

where $\mathbf{Q}_{(a:b, :)}$ is a matrix composed of the rows a to b of matrix \mathbf{Q} .

The third and last part (which is also not studied here) is called intra-subspace separation, and involves finding square matrices \mathbf{W}^k (of size N_k by N_k) such that $\mathbf{s}^k = \mathbf{W}^k \mathbf{y}^k$. There are K such matrices to be found, and each can be estimated separately once the inter-subspace separation and the subspace detection have been performed.

Unfortunately, the available literature is not quite clear on what the term “Independent Subspace Analysis” means. Some authors (*cf.*, [17, 21]) define “ISA” as the task of performing all three steps, while others (*cf.*, [12, 20]) define the same term as solving the first or the first two steps only. To prevent confusion, throughout this paper we will define “Full ISA” as the task of performing all three steps, and “Partial ISA” as the task of performing the first step alone.

ISA (full and partial) has seen increasing interest from the scientific community in recent years. It is known by different names, including “Independent Subspace Analysis” [9, 12, 18, 22], “Subspace Independent Component

Analysis” [20], among others. Relevant theoretical results have been published about this topic, such as sufficient conditions on the distribution of the sources for full ISA to be achievable through maximization of kurtosis [21] or minimization of mutual information [17]. Under these conditions, then, simple ICA algorithms which maximize kurtosis (such as some variants of FastICA) or minimize mutual information (such as Infomax) can be safely used to perform the full ISA task, even though the assumption of independence of the sources is violated. On the other hand, dedicated algorithms for partial ISA have also been proposed; see, *e.g.*, [12] and [20]. Techniques for subspace detection have also been presented recently [9].

2.5 Phase synchrony

Motivated by studies of the human brain, where synchrony plays an important role, we assume here that the ISA subspaces are composed of synchronous sources. The external measurements such as EEG or MEG are the result of superpositions of brain regions’ activities, and due to the low frequencies involved, the mixing process can be assumed to be linear and instantaneous.

We can thus safely assume that the sources follow the ISA model and, in addition, they are assumed to have perfect phase synchrony within each subspace, as measured by the Phase Locking Factor (PLF) [3]; in other words, within each subspace all pairwise PLFs are 1. Such signals follow a specific model: it can be shown that the sources of each subspace k are of the form

[4]

$$s_i^k(t) \equiv a_i^k(t) \cos(\alpha_i^k(t)), \text{ where } \alpha_i^k(t) \equiv \phi_i^k + \psi^k(t), \quad (5)$$

where $a_i^k(t)$ are non-negative amplitudes, and $\alpha_i^k(t)$ is called the phase of source s_i^k . Informally, in each subspace the phase difference between any two sources is constant, *i.e.*, $\alpha_i^k(t) - \alpha_j^k(t) = \phi_i^k - \phi_j^k = \text{const.}$ [3, 4]. For this reason, $\psi^k(t)$ is sometimes called the common oscillation of subspace k (although it does not have to be an oscillatory signal), and ϕ_i^k is called the phase lag of source s_i^k .

3 Algorithms

3.1 ICA algorithms

We now give a brief overview of the specific ICA algorithms used in this study. Note that ICA algorithms have not been designed to be used for ISA tasks, therefore it is not necessarily expected that they perform ideally. The Infomax algorithm [11] can be derived as the maximum likelihood estimator (MLE) of \mathbf{W} , if the observations \mathbf{y} follow the ICA model $\mathbf{y} = \mathbf{M}\mathbf{s}$, with \mathbf{s} having independent components. The optimization of the likelihood function is usually done with gradient steps. In our experiments, we use the implementation of Infomax in the MISEP package² [2] which not only estimates the demixing matrix \mathbf{W} but is also adaptive in the probability density func-

²Available from <http://www.lx.it.pt/~lbalmeida/ica/mitoolbox.html>

tions of the sources, thus avoiding the usual parametrization between sub- and super-gaussian densities [11].

The FastICA algorithm [11] can also be derived as an MLE, just like Infomax; however, the optimization is done using fixed-point iterations instead of gradient steps. FastICA is actually a family of algorithms: one can separate the sources using as contrast function the kurtosis of the estimated sources or their entropy, among other possibilities. We use the authors' implementation of FastICA³; the symmetric version with the cubic nonlinearity is used.

The Second Order Blind Identification (SOBI) algorithm and the Temporal Decorrelation SEparation (TDSEP) algorithms [26] are somewhat different from the previous two. They are based on the following principle: if the estimated sources \mathbf{x} are independent from time-lagged versions of each other, then the correlation matrix $\mathbf{C}(\tau) \equiv \mathbb{E}[\mathbf{x}(t)\mathbf{x}(t - \tau)^T]$, where \mathbb{E} is the expectation value operator, should be diagonal for any value of the time lag τ . SOBI and TDSEP choose a set of M lags τ_1, \dots, τ_M and find \mathbf{W} such that the correlation matrices $\mathbf{C}(\tau_1), \dots, \mathbf{C}(\tau_M)$ are as diagonal as possible. If the data does not follow the ICA model exactly due to, *e.g.*, noise, it is in general impossible to exactly diagonalize these matrices, and algorithms for approximate joint diagonalization must be employed.⁴For more details of these algorithms, see [7, 11].

³Available from <http://research.ics.tkk.fi/ica/fastica/>

⁴We use the TDSEP implementation in [26].

3.2 ISA algorithms

We now briefly discuss the ISA algorithms used in this study. FastISA [12] assumes that the dimensions of the subspaces are known. It is an algorithm which searches for a matrix \mathbf{W} such that the norms of the projections of \mathbf{x} onto each of the K subspaces are independent. This can be viewed as FastICA applied to a scalar function (the norm) of the vector variables \mathbf{x}^k . The goal of FastISA is to perform Partial ISA only.

In [20], it was argued that the kurtosis of a scalar random variable can be generalized to random vectors. The authors prove that, if the dimensions of all subspaces are equal and known (*i.e.*, if $N_1 = \dots = N_K$ and K is known), then one of the stationary points of $\text{vkurt}(\mathbf{x}^1) + \dots + \text{vkurt}(\mathbf{x}^K)$, where vkurt is the vector kurtosis function in [20], corresponds to a solution of Partial ISA.

The third algorithm we use is sJADE, a generalization of JADE [11] based on joint block diagonalization [9]. Unlike the previous two algorithms, sJADE does not assume that the subspace sizes are known.

While these three algorithms only perform Partial ISA, algorithms for subsequent subspace detection have been proposed recently as well [9].

4 Experimental results

4.1 MEG-like data

The choice of the data to use in this study is non-trivial. On one hand, knowledge of the true sources in real EEG and MEG requires simultaneous data from outside the scalp (EEG or MEG, which correspond to the mixed signals \mathbf{y}) and from inside the scalp (intra-cranial recordings, which are more directly related to the sources \mathbf{s}). Acquisition of intra-cranial recordings is, obviously, a very invasive procedure which is seldom available. When they are not available, results can only be assessed qualitatively by human experts, who can tell whether the extracted sources are meaningful or not. On the other hand, synthetic data is too simple to assess the usefulness of these algorithms in real data.

In order to retain the advantages of using simulated and real data, we use MEG-like data, generated from actual MEG recordings. These data were originally used in [24]; they consist of data from 122 MEG channels, each with 17730 time samples. The sampling frequency is 297 Hz, and the data had already been subjected to a low-pass filtering with cutoff at 90 Hz. We begin by selecting 4 of the 122 channels randomly. Those 4 channels are bandpass filtered in the band $[16, 30]$ Hz. The resulting filtered data is then downsampled with a factor of $\lfloor \frac{0.25}{B} \rfloor$, where $B = 14/297$ is the passband width in relative frequency units. This downsampling is performed to make the size of the data manageable by some of the more computationally intensive

algorithms. We then compute the Hilbert Transform [16] of those 4 channels and use it to obtain the corresponding analytic signals, $\hat{u}_i(t)$, $i = 1, \dots, 4$. Note that $\hat{u}_i(t)$ are complex-valued signals.

Since the 4 MEG channels are not independent, we split the observation period into two parts, one with $t = 1, \dots, \frac{17730}{2}$, and one with $t = \frac{17730}{2} + 1, \dots, 17730$. The first half is used to generate data for the first subspace, whereas the second half is used for the second subspace.

The common oscillation of the first subspace is taken as $\psi^1(t) \equiv e^{i \text{angle}(\hat{u}_1(t))}$. The phase lags ϕ_1^1 and ϕ_2^1 are drawn from a $\text{Uniform}(0, 2\pi)$ distribution. The amplitudes are simply taken as the amplitudes of the first and second analytic signals: $a_{1,2}^1(t) \equiv |\hat{u}_{1,2}(t)|$. Recall that, for the first subspace, only the first half of the observation period is used.

The second subspace is generated in a similar manner, with the following differences: the common oscillation is now generated from the third MEG channel, $\psi^2(t) \equiv e^{i \text{angle}(\hat{u}_3(t))}$; the amplitudes are now taken as the amplitudes of the third and fourth analytic signals, $a_{1,2}^2(t) \equiv |\hat{u}_{3,4}(t)|$; and, as mentioned earlier, the second half of the observation period is used instead of the first half.

Furthermore, to ensure that the mixing process is physiologically plausible, we use the EEGIFT [8] package with its example dataset and default options to estimate 20 independent components on a dataset of 64 EEG recordings. We select a random 4-by-4 submatrix from the 20-by-64 mixing matrix estimated by EEGIFT and use it to mix the MEG-like sources

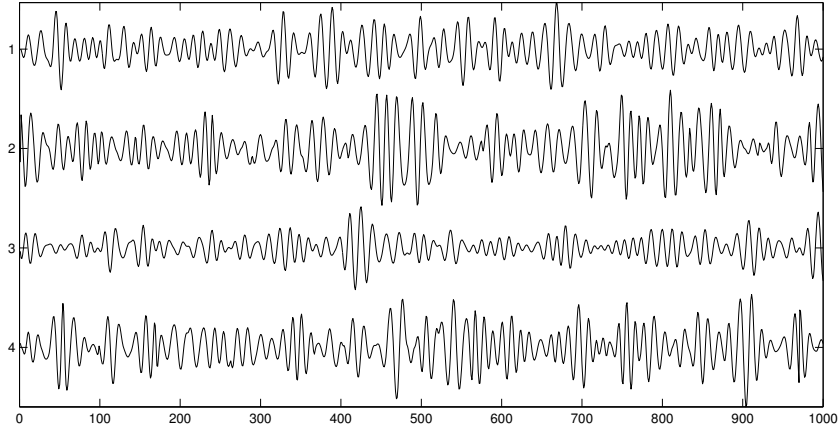


Figure 1: Excerpt of one dataset with 2+2 structure. The figure shows the first 1000 time points of the four sources.

described in the previous paragraphs.

The above process generates a set of 4 sources, in which sources 1 and 2 form a subspace and sources 3 and 4 form another subspace. An example of such sources is shown in figure 1. We call this a “2+2” structure. We used an analogous procedure to generate two other datasets: one in a “3+3” structure, *i.e.*, a set of 6 sources where sources 1, 2, and 3 form one subspace, as do sources 4, 5, and 6; and one in a “2+2+2” structure, *i.e.*, a set of 6 sources where sources 1 and 2 form one subspace, sources 3 and 4 form another subspace, and sources 5 and 6 form yet another subspace.

By choosing randomly which MEG channels are used to generate the sources and which sub-matrix from the 20-by-64 mixing matrix estimated by EEGIFT is used to mix the sources, we can generate different sets of data. We generate 500 datasets of each of the structures mentioned in the previous paragraph and run the six algorithms mentioned above on them.

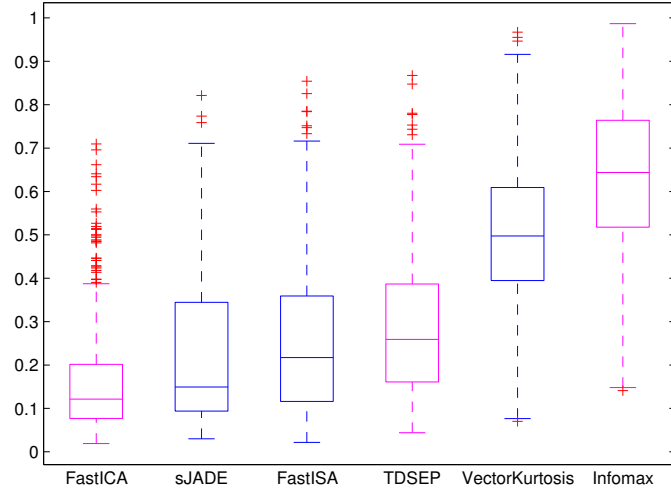


Figure 2: $[2+2]$ Values of the subspace Amari Performance Index (SAPI) for the six algorithms applied to the 500 datasets with $2+2$ structure. The box plots represent the following quantiles: 0 (bottom whisker), 0.25 (bottom edge of the box), 0.5 (solid line within the box), 0.75 (top edge of the box) and 1 (top whisker). If some values of the SAPI would make the whiskers longer than 1.5 times the height of the corresponding box, they are considered outliers, are represented as individual “+” symbols. They are not considered in the box plots. Magenta boxes correspond to ICA algorithms whereas blue boxes correspond to ISA algorithms.

4.2 Results

The results of these experiments are summarized in figures 2, 3, and 4. We use the Subspace Amari Performance Index (SAPI; see, *e.g.*, [21]) to measure the quality of the inter-subspace separation. This quantity is zero if and only if the gain matrix \mathbf{WM} is a permutation of a block diagonal matrix, with block sizes corresponding to the subspace dimensions. Otherwise, it is positive, with greater values of the SAPI corresponding to worse separations.

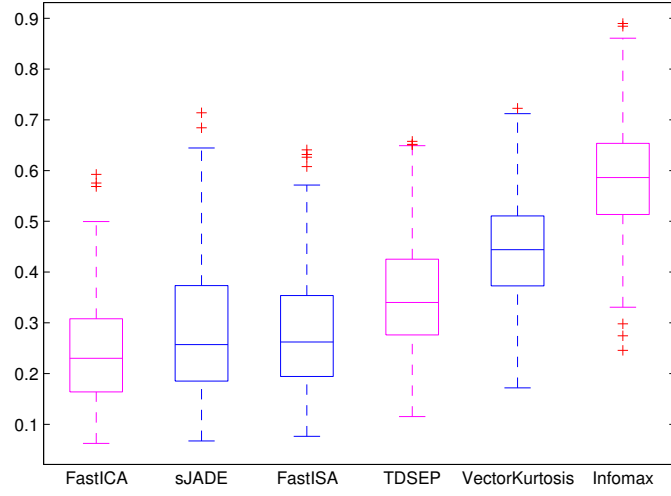


Figure 3: $[2+2+2]$ Values of the subspace Amari Performance Index (SAPI) for the six algorithms on the 500 datasets with $2+2+2$ structure. The box plots represent similar quantities as figure 2.

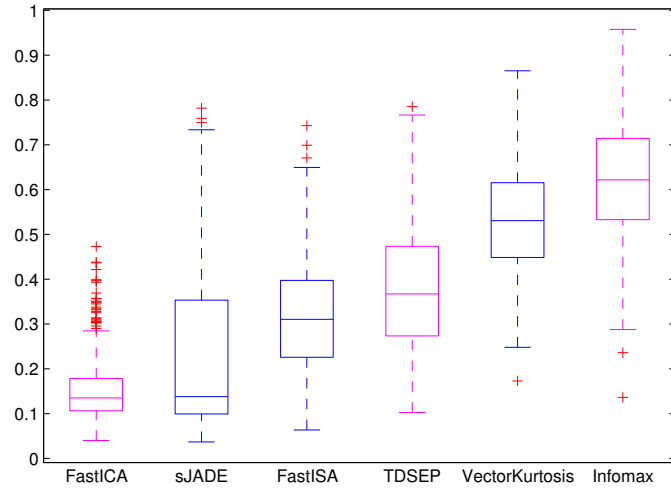


Figure 4: $[3+3]$ Values of the subspace Amari Performance Index (SAPI) for the six algorithms on the 500 datasets with $3+3$ structure. The box plots represent similar quantities as figure 2.

The main conclusion is very interesting, albeit unexpected: the best overall algorithm for this type of data appears to be FastICA, an algorithm which was not tailored for tackling subspaces. The performance depends on the random elements (phase lags, choice of MEG channels, choice of submatrix), but the typical SAPI value for FastICA is of the order of 0.1 to 0.2 for the 2+2 and 3+3 cases, which correspond to good separations. The 2+2+2 case worsens the results on all algorithms, but FastICA's results of 0.18 to 0.3 are still acceptable.

5 Discussion

The algorithms used here require some parameter choices. We found that the results shown here are relatively robust to such choices: for example, FastICA can be used with several non-linearities, all of which yield similar results. It can also be used in a symmetric or deflation mode, and again, both yield similar results. The choice of lags in TDSEP is the only somewhat critical parameter; however, since we downsample the data according to the sampling frequency and the filter bandwidth, we simply used all lags from 0 to 8. The reader is referred to the references given in Section 3 for more information on the effect of these parameters and on how they should be chosen.

The results shown here allow an important conclusion for the goal of performing full ISA when each subspace is composed of synchronous sources.

FastICA yields good results despite being an ICA algorithm and not an ISA algorithm. Algorithms for the intra-subspace separation already exist for this type of sources [3, 4]; thus, the full ISA task can, in theory, be satisfactorily performed if one devises a proper subspace detection procedure. Note, however, that this may depend strongly on the type of dependency in each subspace, since other authors have found rather different results on other types of data [9].

In practice, some questions still need to be addressed. The most important of these is the performance of these methods on actual EEG/MEG data. The MEG-like data used here allows a quantitative measurement of the quality of the separation; such measurement is not available on real EEG or MEG. On the other hand, natural EEG/MEG data contain disturbing properties (such as artifacts) which were simplified in the present data generation.

Another relevant issue is the performance when the number of sources and measurements is large (more than 100 as in EEG/MEG): in this case, one can compress the data onto a smaller dimension using, *e.g.*, Principal Component Analysis [11], but the effective number of sources will still likely be larger than 6. Unfortunately, due to time limitations, we could not run simulations for larger numbers of sources in time for this submission. Finally, the choice of the frequency band to analyze is not innocuous, since different frequency bands contain information about different brain phenomena [15].

6 Conclusion

We have studied the performance of several algorithms for the separation of subspaces of synchronous sources, using MEG-like data to assess the quality of the separation. We have found that FastICA yields the best results on this type of data, despite the fact that it is an ICA algorithm and not an ISA algorithm. Nevertheless, sJADE (in general) or FastISA (if the subspace dimensions are known in advance) also seem to be able to deal reasonably well with the problem of partial ISA.

Acknowledgements

This work was supported by FCT project PEst-OE/EEI/LA0008/2011, under internal project DECA-Bio, and by the Academy of Finland through its Centres of Excellence Program 2012-2017.

References

- [1] E. Alhoniemi, A. Honkela, K. Lagus, and J. Seppä. Compact modeling of data using independent variable group analysis. *IEEE Transactions on Neural Networks*, 18:1762–1776, 2007.
- [2] L.B. Almeida. MISEP - linear and nonlinear ICA based on mutual information. *Journal of Machine Learning Research*, 4:1297–1318, 2004.

- [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*, 22(9):1419–1434, 2011.
- [4] M. Almeida, R. Vigário, and J. Bioucas-Dias. Estimation of the common oscillation for phase locked matrix factorization. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 2012.
- [5] J. Beirlant, E. Dudewicz, L. Györfi, and E. van der Meulen. Nonparametric entropy estimation: an overview. *International Journal of Mathematical and Statistical Sciences*, 6:17–39, 1997.
- [6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [7] A. Cichocki and S. Amari. *Adaptive blind signal and image processing - Learning algorithms and applications*. John Wiley & Sons, 2002.
- [8] T. Eichele, S. Rachakonda, B. Brakedal, R. Eikeland, and V. D. Calhoun. EEGIFT: Group independent component analysis for event-related EEG data. *Computational Intelligence and Neuroscience*, 2011:1–9, 2011.
- [9] H.W. Gutch, J. Krumsiek, and F.J. Theis. An ISA algorithm with unknown group sizes identifies meaningful clusters in metabolomics

- data. In *Proceedings of the European Signal Processing Conference (EU-SIPCO)*, 2011.
- [10] P. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.
 - [11] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.
 - [12] A. Hyvärinen and U. Köster. FastISA: A fast fixed-point algorithm for independent subspace analysis. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, 2006.
 - [13] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13, pages 556–562, 2001.
 - [14] Paul L. Nunez, Ramesh Srinivasan, Andrew F. Westdorp, Ranjith S. Wijesinghe, Don M. Tucker, Richard B. Silberstein, and Peter 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.
 - [15] P.L. Nunez and R. Srinivasan. *Electric Fields of the Brain: the neurophysics of EEG*. Oxford University Press, 2006.
 - [16] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice-Hall International Editions, 1999.

- [17] J.A. Palmer and S. Makeig. Blind separation of dependent sources and subspaces by minimum mutual information. Technical report, University of California, San Diego, 2010.
- [18] B. Póczos and A. Lörincz. Independent subspace analysis using geodesic spanning trees. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.
- [19] A. Pikovsky, M. Rosenblum, and J. Kurths. *Synchronization: A universal concept in nonlinear sciences*. Cambridge Nonlinear Science Series. Cambridge University Press, 2001.
- [20] A. Sharma and K.K. Paliwal. Subspace independent component analysis using vector kurtosis. *Pattern Recognition*, 39:2227–2232, 2006.
- [21] Z. Szabó, B. Póczos, and A. Lorincz. Undercomplete blind subspace deconvolution. *Journal of Machine Learning Research*, 8:1063–1095, 2007.
- [22] F.J. Theis. Towards a general Independent Subspace Analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.
- [23] P. J. Uhlhaas and W. Singer. Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology. *Neuron*, 52:155–168, Oct 2006.
- [24] R. Vigário, V. Jousmäki, M. Hämäläinen, R. Hari, and E. Oja. Independent component analysis for identification of artifacts in magnetoencephalographic recordings. In *Advances in NIPS*, 1997.

- [25] 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 Biom. Eng.*, 47(5):589–593, May 2000.
- [26] A. Ziehe and K.-R. Müller. TDSEP - an efficient algorithm for blind separation using time structure. In *International Conference on Artificial Neural Networks*, pages 675–680, 1998.

Separation of Synchronous Sources through Phase Locked Matrix Factorization

Miguel Almeida, Ricardo Vigário, José Bioucas-Dias

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 (ICA) 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 PLMF's 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 non-noisy) case, and with various levels of additive noise in the observed signals and of phase jitter in the sources.

Index Terms—phase-locking, synchrony, source separation, independent component analysis, matrix factorization

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 blind source separation (see [1] for an extensive overview). Non-linear 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 non-essential indeterminations which are typical of BSS problems (see section II-E). Good overviews of ICA can be found in [6], [7], [8].

A generalization of ICA is Independent Subspace Analysis (ISA) [9], [10], [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 non-negative. This problem is known as Non-negative 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, *e.g.*, [19], [20], [21]). As an example, it has been shown that muscle activity measured with an electromyogram (EMG) 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 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 this mixing needs to be undone.

M. Almeida is with Instituto de Telecomunicações, Lisbon, Portugal. Email: malmeida@lx.it.pt.

R. Vigário is with Aalto University School of Science, Espoo, Finland. Email: ricardo.vigario@aalto.fi

J. Bioucas-Dias is with Instituto de Telecomunicações, Lisbon, Portugal. Email: bioucas@lx.it.pt

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, or 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 TDSEP¹ [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 intra-subspace 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 \mathbf{y} and \mathbf{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 work we present an alternative technique to solve the SSS problem, named Phase Locked Matrix Factorization (PLMF) [28], [29]. Compared to IPA, PLMF has no singular solutions and can deal with higher amounts of noise and with non-square 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 PLMF's robustness to deviations of this model, induced, *e.g.*, by additive noise or by imperfect synchronization between the sources.

The paper is organized as follows. In Section II we provide an overview of some theoretical concepts that will be needed throughout the 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 conclusions are 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-valued² Phase Locking Factor (PLF) between these two signals is defined as

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

where $|\cdot|$ and $\langle \cdot \rangle$ are the absolute value and time average operators, and $i = \sqrt{-1}$. 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 non-zero 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, mathematical characterization: it was shown in [25] that if $\mathbf{s}(t)$ is a set of such sources and if we define $\mathbf{y}(t) \equiv \mathbf{M}\mathbf{s}(t)$ 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 \mathbf{y} to be equal to \mathbf{s} up to permutation and scaling, a typical, non-restrictive indeterminacy in source separation problems (indeterminacies are further discussed in section II-E).

This effect is illustrated in Figure 1, which shows four perfectly synchronized sources and their PLFs. That 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.

This property illustrates 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

¹TDSEP 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].

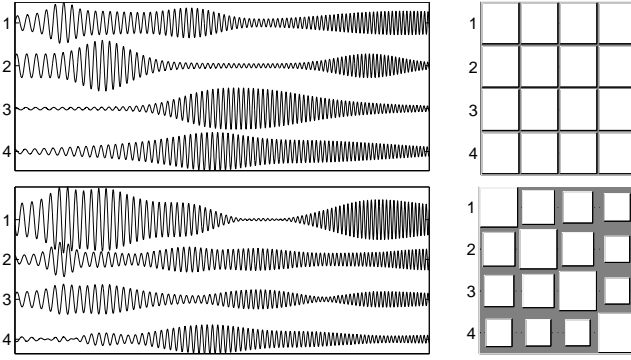


Fig. 1. (Top row) Three sources (left) and PLFs between them (right). On the right column, the area of the square in position (i, j) is proportional to the PLF between the signals i and j . Therefore, large squares represent PLFs close to 1, while small squares represent values close to zero. The smallest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

separated, the synchrony values will not be accurate. Established BSS methods such as Independent Component Analysis (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 elementwise (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 $\Phi_j(t) \equiv e^{i \arg[s_j(t)]} \equiv e^{i \phi_j(t)}$. The elements of \mathbf{A} are non-negative, 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

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

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 $\text{rank}(\Phi) = 1$, 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 j -th element of \mathbf{z} and the t -th 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:

$$\mathbf{Y} \equiv \mathbf{M} \mathbf{S} + \mathbf{N}, \quad (3)$$

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

$$\mathbf{Y} \equiv \mathbf{M} \mathbf{S} = \mathbf{M} [\mathbf{A} \odot (\mathbf{z} \mathbf{f}^T)] = \mathbf{M} \mathbf{D}_{\mathbf{z}} \mathbf{A} \mathbf{D}_{\mathbf{f}}, \quad (4)$$

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

E. Indeterminacies

Blind source separation problems usually do not have unique solutions. For example, in Independent Component Analysis it is well known that the order of the estimated sources might not be the same as their original order. This happens because if $\mathbf{Y} = \mathbf{M} \mathbf{S}$, 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 non-zero 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 $e^{i\gamma} \mathbf{z}$ and $e^{-i\gamma} \mathbf{f}$, 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 take this indeterminacy into account.

III. ALGORITHM

We now present Phase Locked Matrix Factorization, 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

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

for each subproblem, which 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

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

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 non-convex problem [32] due to the presence of a product of several variables in the cost function and of non-convex constraints. Non-convex 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 non-negative entries, PLMF does not explicitly force that assumption. As we shall see, even if the sources follow the model from section II-D with non-negative amplitudes, it is not necessary to impose this constraint on the optimization problem.

To solve the minimization problem in (5), 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 (5). In the second subproblem (Section III-C), we use the estimated \mathbf{f} , keep it fixed, and solve (5) 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.

⁴This contrasts with our original PLMF approach presented in [29]. In that work, we optimized the cost function (5) 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.

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 $\rho = \frac{\sigma_{max}}{\sigma_{min}}$, where σ_{max} is the largest singular value of \mathbf{M} and σ_{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 doesn’t 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 $\mathbf{C}_Y \equiv \frac{1}{T} \mathbf{Y} \mathbf{Y}^H$. Usually, prewhitening involves multiplying the data \mathbf{Y} on the left by a matrix which we define as

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

where \mathbf{D} is a $N \times N$ diagonal matrix containing the nonzero eigenvalues of \mathbf{C}_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 $\mathbf{Y} = \mathbf{M}\mathbf{S}$, on the left, by this matrix transforms the original source separation problem $\mathbf{Y} = \mathbf{M}\mathbf{S}$ into a new problem $\mathbf{B}\mathbf{Y} = \mathbf{B}\mathbf{M}\mathbf{S}$, where $\mathbf{B}\mathbf{Y}$ can be interpreted as new data, and $\mathbf{B}\mathbf{M}$ 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 $\mathbf{B}\mathbf{M}$. 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 $\mathbf{B}\mathbf{M}$ 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 $\mathbf{B}\mathbf{M}$ 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

⁵While 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.

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

⁷In 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 $\mathbf{Y} = \mathbf{M}\mathbf{S}$ can be turned into an equivalent real problem in two different ways:

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

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

- \mathbf{M} and \mathbf{S} both have maximum rank.
- There is no additive noise; thus, $\mathbf{Y} = \mathbf{M}\mathbf{S}$ holds.
- The amplitudes of each source, $a_j(t)$, are i.i.d. realizations of a random variable which we denote by A_j ;
- A_j is independent of A_k for $j \neq k$;
- A_j is independent of ϕ_k for any j and k , including $j = k$;
- All A_j have the same distribution (we denote by A a random variable with that distribution);
- 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 ;
- 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 $\rho(\mathbf{B}\mathbf{M})$, obeys

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

where N is the number of sources, $\mathbb{E}[\cdot]$ is the expected value operator and $\text{Var}[\cdot]$ is the variance operator. Furthermore, this upper bound is tight, meaning that in some cases equation (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 $\mathbf{B}\mathbf{Y}$ as new data, and we designate these new data simply as \mathbf{Y} and the equivalent mixing matrix $\mathbf{B}\mathbf{M}$ simply as \mathbf{M} .

B. Estimation of \mathbf{f}

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

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

\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 $\mathbf{M}\mathbf{D}_z$ into the matrix \mathbf{H} , which is now allowed to be any complex matrix. This relaxation⁸ means that the subproblem in Eq. (9) is easier to tackle than the original problem in Eq. (5). If the sources exactly follow the model in Eq. (4), a factorization of the form $\mathbf{Y} = \mathbf{H}\mathbf{A}\mathbf{f}$ 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 (9) 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 $\mathbf{Y} = \mathbf{H}\mathbf{A}\mathbf{f}$, then \mathbf{f} is correctly estimated through the minimization in (9), 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 $\mathbf{Y} = \mathbf{H}_1\mathbf{A}_1\mathbf{D}_{f1}$ with $\mathbf{H}_1 \in \mathbb{C}^{P \times N}$, $\mathbf{A}_1 \in \mathbb{R}^{N \times T}$, $\mathbf{D}_{f1} \in \mathbb{D}_1^T$, 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, $\mathbf{Y} = \mathbf{H}_2\mathbf{A}_2\mathbf{D}_{f2}$, then necessarily one has $\mathbf{D}_{f2} = \mathbf{E}\mathbf{D}_{f1}$ 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

⁸Recall 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 $\mathbf{M}\mathbf{D}_z$ does not span the space of all possible complex matrices \mathbf{H} . Therefore, this is a relaxed version of the original problem (5).

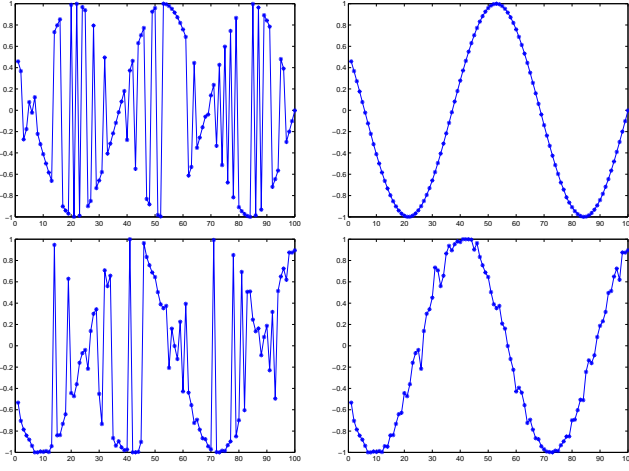


Fig. 2. Typical result of the phase correction procedure described at the end of section III-B. Top row: results for noiseless data. We show the real part of \mathbf{f} before the correction is made (left) and the real part of \mathbf{f} after that correction (right). Bottom row: similar to the top row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

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. 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 $t = 1, \dots, T-1$, the quantity

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

where $f_R(t)$ is the real part of the t -th 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 $\gtrsim \sqrt{2}$ if such a sign change occurs from t to $(t+1)$. In our algorithm we determine that there is a change in sign when

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

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 figure 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 Eq. (5) reduces to a second subproblem,

$$\min_{\mathbf{M}, \mathbf{A}, \mathbf{z}} \frac{1}{2} \|\mathbf{Y}\mathbf{D}_{\mathbf{f}}^* - \mathbf{M}\mathbf{D}_{\mathbf{z}}\mathbf{A}\|_F^2, \quad (12)$$

$$\text{s.t.: } 1) \max_{i,j} |m_{ij}| = 1 \\ 2) |\mathbf{z}_j| = 1 \text{ for all } j$$

(note that $\mathbf{D}_{\mathbf{f}}^* = \mathbf{D}_{\mathbf{f}}^{-1}$). Constraints should be interpreted as in the original formulation (5).

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. Further 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 is generated according to equation (4) with non-negative amplitudes (i.e. \mathbf{A} has non-negative entries), Theorems 2 and 3 can be combined to produce an interesting result: the original PLMF problem, as stated in (5), has a unique solution with non-negative amplitudes. This is stated in the next theorem.

Theorem 4. *Let \mathbf{Y} be data generated according to the model in equation (4) with non-negative 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 non-negative, the constraints of problem (5) 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. ■

⁹A sketch of the proof is as follows: the previous derivation remains valid until equation (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.

Given this theorem, and since in our experiments we will indeed use data generated using non-negative 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 itself suggests that one should split the problem into two subproblems.

E. Optimization procedures

The PLMF algorithm is presented in Table I. We now explain in further detail how each of the two subproblems is tackled. We employ the block nonlinear Gauss-Seidel [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 (9) and (12), 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 (9) and (12) are convex in some variables and non-convex 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 sub-optimal procedure, the fact that the two subproblems are non-convex 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 taking the first constraint of (9) into account. 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 (9) and (12) 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).

PHASE LOCKED MATRIX FACTORIZATION	
1:	Given: data \mathbf{Y}
0:	WHITENING
2:	Whiten data \mathbf{Y}
I:	ESTIMATION OF \mathbf{f}
3:	for run $\in \{1, 2, \dots, \text{MaxRuns}_f\}$, do
4:	Randomly initialize $\hat{\mathbf{H}}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{f}}$
5:	for iter $\in \{1, 2, \dots, \text{MaxIter}_f\}$, 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_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 $\in \{1, 2, \dots, \text{MaxRuns}_{\mathbf{M}, \mathbf{A}, \mathbf{z}}\}$, do
15:	Randomly initialize $\hat{\mathbf{M}}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{z}}$
16:	for iter $\in \{1, 2, \dots, \text{MaxIter}_{\mathbf{M}, \mathbf{A}, \mathbf{z}}\}$, 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 $\text{MaxRuns}_{\mathbf{M}, \mathbf{A}, \mathbf{z}}$ 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}

TABLE I
THE PHASE LOCKED MATRIX FACTORIZATION ALGORITHM.

IV. EXPERIMENTAL RESULTS

A. Data generation

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

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

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 Equation (4).

We generate 1000 datasets for each set of parameters that we study. For each dataset, 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 $[0, \Delta\phi, \dots, (N-1)\Delta\phi]^T$ ($\Delta\phi$ is determined below), and the common oscillation \mathbf{f} is generated as a sinusoid: $\mathbf{f} = [0, \exp(i\Delta t), \exp(i2\Delta t), \dots, \exp(i(T-1)\Delta t)]$, 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 lowpass filtering a Gaussian white noise signal. This is appropriate

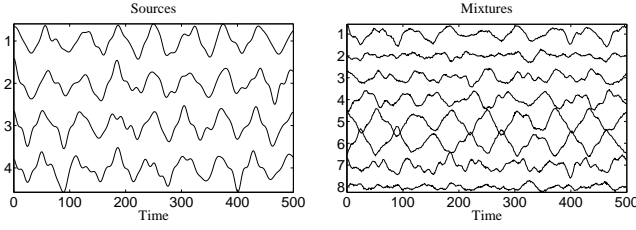


Fig. 3. Left: the real part of a typical set of four sources generated as described in section IV-A, with no phase jitter. Right: the real part of a corresponding set of eight mixtures, with an input signal-to-noise ratio (SNR) of 20 dB. Note that in most of the following experiments, only 100 points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

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 non-negative¹⁰, 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 depicted in figure 3, where we present an extended time period ($T = 500$) to better illustrate the structure of the signals.

We study the effect of the following variables:

- 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.
- Phase jitter \mathbf{J} . We study two types of jitter:
 - 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 $\sigma_{iid} = 0, 0.02, 0.04, \dots, 0.1$. 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} .
 - 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 elementwise. 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.

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

- Phase lag $\Delta\phi$. We study the cases of $\Delta\phi = \pi/50, 2\pi/50, \dots$, up to $12\pi/50$.
- Number of sources N and number of sensors P . We study the cases $N = 2, 4, \dots, 10$, with $P = N$ and with $P = 2N$.
- 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 $\Delta\phi = \pi/10$.

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 datasets 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 datasets, we empirically compute $\mathbb{E}[A]^2$ and $\text{Var}[A]$ to compute the value of the bound.

We also generate 1000 datasets, 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 datasets where each time sample of \mathbf{A} is drawn independently from a uniform distribution. These datasets have $T = 10000$, to ensure that the sample covariance matrix is very similar to the true one. These datasets 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.

Figure 4 shows the results for these four types of datasets. 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.

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 datasets 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

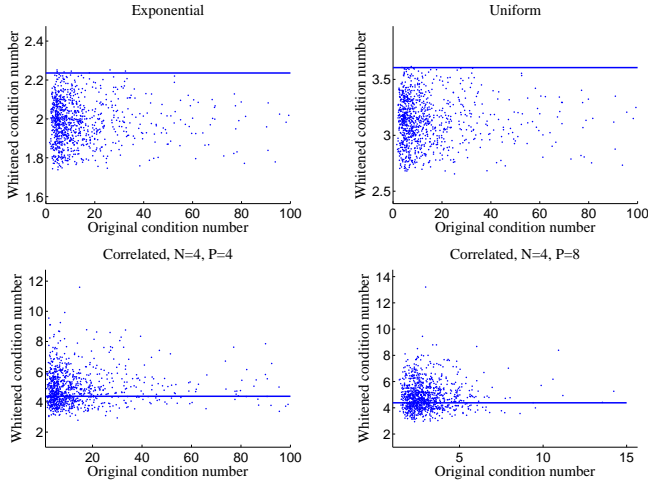


Fig. 4. Comparison of the condition number of the original mixing matrix, $\rho(\mathbf{M})$ (horizontal axis), with that of the equivalent mixing matrix after whitening, $\rho(\mathbf{BM})$ (vertical axis). Top left: results for datasets which follow the assumptions of theorem 1, and for which each 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 datasets 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.

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, $\text{MaxRuns}_f = \text{MaxRuns}_{\mathbf{M}, \mathbf{A}, \mathbf{z}} = 5$. In other words, we solved each subproblem 5 times and kept only the solution which yielded the lowest cost value. We used $\text{MaxIter}_f = 100$ 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 isn't used. Figure 5 shows a typical result of this comparison. On both sub-problems, 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 $\hat{\mathbf{B}}\mathbf{M}$ to distinguish it from the true equivalent mixing matrix $\mathbf{B}\mathbf{M}$.¹² We begin by computing the gain matrix, defined as $\mathbf{G} \equiv (\hat{\mathbf{B}}\mathbf{M})^\dagger \mathbf{B}\mathbf{M}$, 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

¹¹There 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 $P \rightarrow \infty$, 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 figure 4.

¹²As 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.

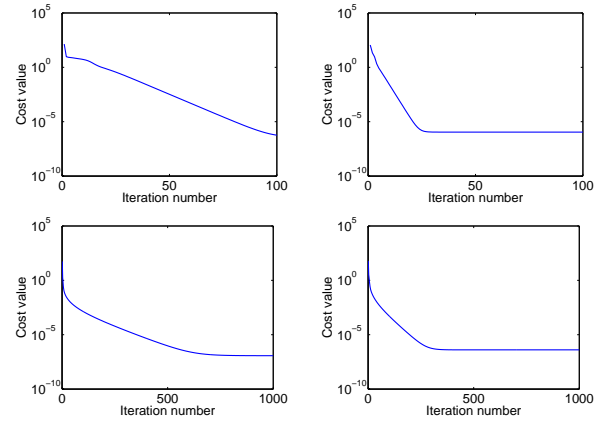


Fig. 5. Influence of whitening on the convergence of PLMF. The top row shows the evolution of the cost function of the first sub-problem (equation (9)) as a function of the number of iterations, without whitening (left) and with whitening (right). The top-left and top-right subfigures used the exact same data. The bottom row shows the same for the second sub-problem (equation (12)); again, both sub-figures used the same data. Clearly, whitening improves the speed of convergence.

simply “separation quality”, is defined as

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

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 non-diagonal elements. The $10 \log(\cdot)$ function allows us to express that quotient in dB, a unit typically used for the signal-to-noise ratio (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 illustrated in figure 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

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

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 j -th 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 $\mathbf{B}\mathbf{M}$ and its estimation. Finally, note that if $\hat{\mathbf{B}}\mathbf{M}$ is a good estimate of $\mathbf{B}\mathbf{M}$, that allows us to recover a good estimate of the original sources, by directly computing

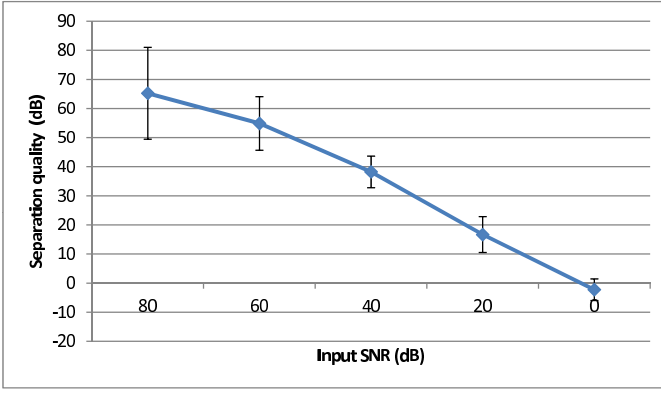


Fig. 6. Separation quality versus input SNR. Under heavy noise, PLMF can recover the sources with about as much noise as they had in the input. Error bars correspond to one standard deviation.

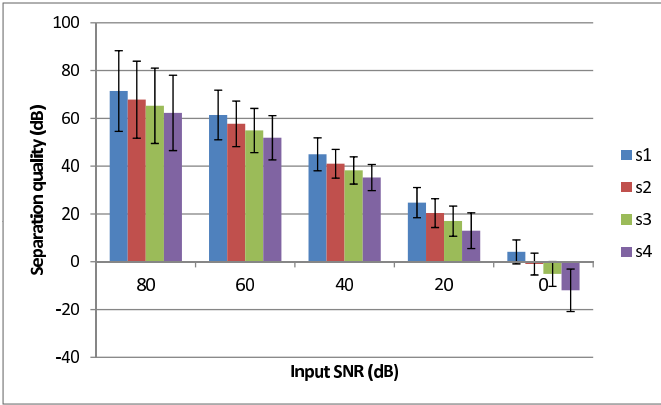


Fig. 7. Separation quality per source versus input SNR. In each group of bars, s_1 represents mean and standard deviation of the SNR of the best-estimated source for the 1000 runs of PLMF. s_2 represents the same for the second-best-estimated source, and so on. This graph illustrates that all sources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

$\hat{\mathbf{S}} \equiv (\mathbf{B}\hat{\mathbf{M}})^\dagger \mathbf{B}\mathbf{Y}$. If estimates of \mathbf{A} , \mathbf{z} and \mathbf{f} are desired, those can easily be obtained from $\hat{\mathbf{S}}$.

Figure 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 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 Figure 7. It can be seen 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.

Figure 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

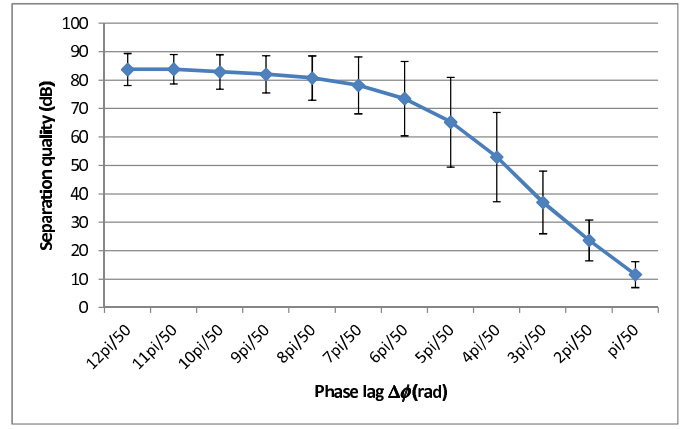


Fig. 8. Separation quality versus phase lag. PLMF's results are, in general, good, but they deteriorate progressively as one approaches the case where $\Delta\phi = 0$, where theorem 3 fails to hold.

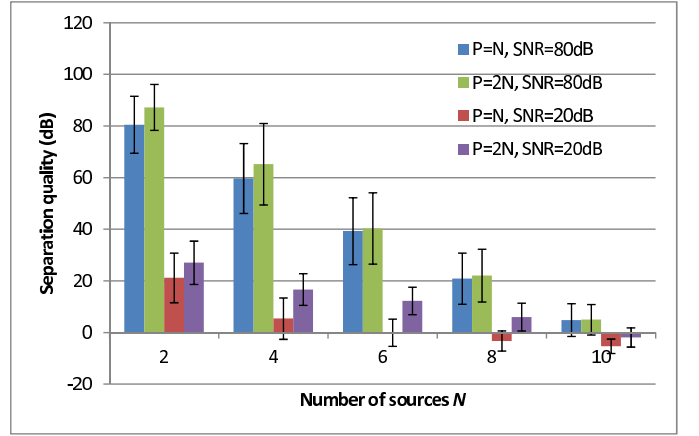


Fig. 9. Separation quality versus number of sources (N), number of sensors (P), and input SNR. For the low noise case, having twice as many sensors only brings a negligible benefit. However, when there is considerable noise, having more sensors improves the results considerably, especially for $N = 4,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

phase lags (smaller than $\frac{2\pi}{50}$, or 7.2 degrees, which yields a separation quality of 23.7 dB).

Figure 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 \mathbf{M} , \mathbf{A} and \mathbf{z} increases. When there is very little noise (input SNR of 80 dB), there is 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.

Figures 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.

Figure 12 shows that, in some situations, increasing the number of time samples T is beneficial. The results indicate

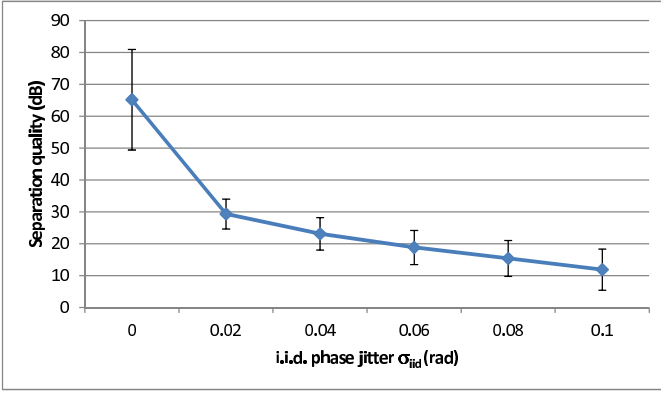


Fig. 10. Separation quality versus i.i.d. phase jitter. The horizontal axis represents the standard deviation of the phase of each entry of \mathbf{J} .

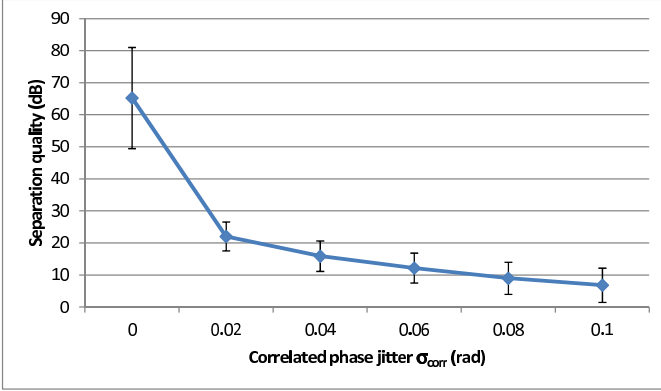


Fig. 11. Separation quality versus correlated phase jitter. The horizontal axis represents the amplitude of the correlated phase jitter.

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 to 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

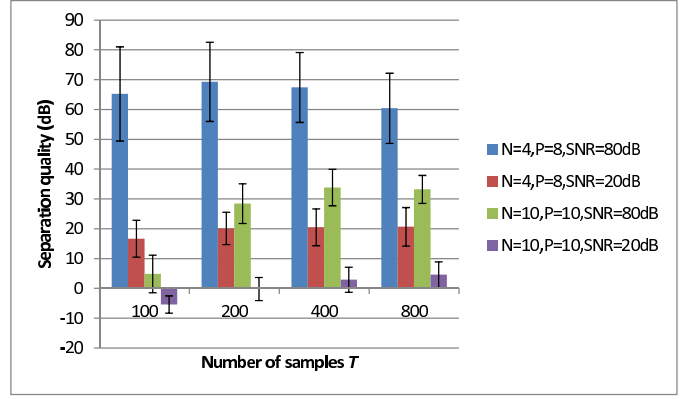


Fig. 12. Separation quality versus number of time samples T . We study four cases: $N = 4$, $P = 8$ and $N = P = 10$, each with input SNR of 80 and 20 dB.

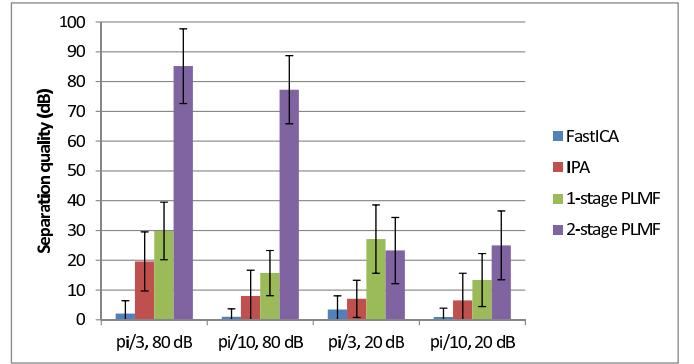


Fig. 13. Comparison of FastICA [6], IPA [25], one-stage PLMF [29], and two-stage PLMF (this work). The two-stage PLMF algorithm clearly dominates the other two algorithms, except for one situation ($\Delta\phi = \pi/3$, input SNR of 20 dB) where it is essentially tied for first place.

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 figure 13. FastICA performs poorly¹³ compared to all SSS algorithms, a consequence of the strong inter-dependence 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 to the other two SSS algorithms.

Figure 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.

¹³We 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.

V. DISCUSSION

We begin our discussion by briefly mentioning the runtime of PLMF: it takes about 3 seconds in total, on a typical modern desktop computer, to run PLMF in MATLAB for a dataset with $N = 4$ sources, $P = 8$ sensors, and $T = 100$ time samples. The time grows to about 11 seconds for $N = 10$, $P = 20$ and $T = 100$ (more sources and sensors, same number of samples), and to about 6 seconds 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 (5), penalizing large fluctuations in the values of \mathbf{A} and \mathbf{f} . In that case, the problem becomes

$$\min_{\mathbf{M}, \mathbf{A}, \mathbf{z}, \mathbf{f}} \frac{1}{2} \|\mathbf{Y} - \mathbf{M}\mathbf{D}_z\mathbf{A}\mathbf{D}_f\|_F^2 + \lambda_A \|\mathbf{A}\mathbf{L}_A\|_F^2 + \lambda_f \|\mathbf{L}_f\mathbf{f}\|_2^2, \quad (16)$$

with the same constraints as before, where \mathbf{L}_A and \mathbf{L}_f are the first-order difference operators of appropriate size, such that the entry (j, t) of $\mathbf{A}\mathbf{L}_A$ is given by $a_j(t+1) - a_j(t)$, and the k -th entry of $\mathbf{L}_f\mathbf{f}$ is given by $f_{(k+1)} - f_k$. The two parameters λ_A and λ_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 Figures 10 and 11. The results from figures 10 and 11 might suggest, at first, that both types of phase jitter cause PLMF's performance to deteriorate. However, a detailed inspection of the estimated variables reveals a significant difference in behavior. Figures 14 and 15 present a typical result of the first subproblem for two cases which are similar in all aspects, except that the first one has i.i.d. phase jitter with a very high standard deviation ($\sigma_{iid} = 0.2$ rad), whereas the second one has very strong correlated jitter (with an amplitude of $\sigma_{corr} = 0.2$).¹⁴ Let $\psi(t)$ denote the phase of the t -th 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 figure 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 figure 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 block nonlinear Gauss-Seidel (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

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

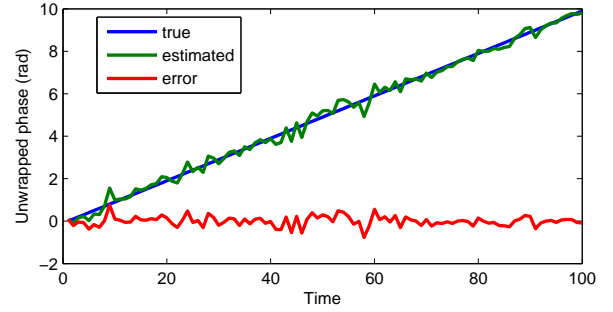


Fig. 14. Typical results of the first subproblem (estimation of \mathbf{f}) in the presence of strong i.i.d. phase jitter. While there is considerable error in the estimation of \mathbf{f} , this error could be significantly reduced using a simple low-pass filter.

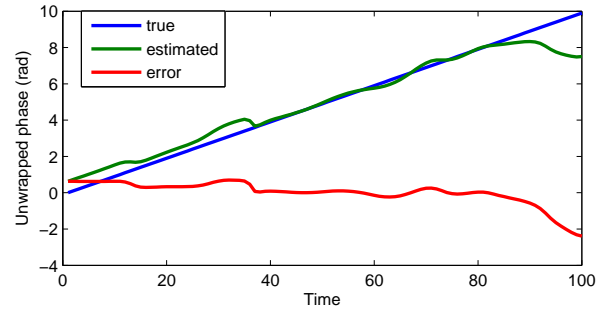


Fig. 15. Typical results of the first subproblem (estimation of \mathbf{f}) in the presence of strong correlated phase jitter. There is considerable error in the estimation of \mathbf{f} , and this error could not be significantly reduced using a low-pass filter.

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 indeterminacies 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 previous work [36] we addressed this issue by constructing pseudo-real 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 pseudo-real data is a direction we intend to address in future work.

VI. CONCLUSION

We have presented Phase Locked Matrix Factorization, an algorithm to perform separation of synchronous sources. 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 8, 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 to previous algorithms for the SSS problem.

ACKNOWLEDGEMENTS

This work was partially supported by project DECA-Bio of Instituto de Telecomunicacoes, PEst-OE/EEI/LA0008/2011.

REFERENCES

- [1] M. S. Pedersen, J. Larsen, U. Kjems, and L. C. Parra, *Springer Handbook of Speech Processing*. Springer Press, 2008, ch. A Survey of Convolutional Blind Source Separation methods, pp. 1065–1084.
- [2] C. Jutten and J. Karhunen, “Advances in nonlinear blind source separation,” in *Proc. of the 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003)*, 2003.
- [3] L. B. Almeida, “Synthesis lectures on signal processing,” in *Nonlinear Source Separation*. Morgan & Claypool, 2006.
- [4] C. Jutten and J. Herault, “Blind separation of sources: an adaptive algorithm based on neuromimetic architecture,” *Signal Processing*, vol. 24, pp. 1–10, 1991.
- [5] P. Comon, “Independent component analysis, a new concept?” *Signal Processing*, vol. 36, pp. 287–314, 1994.
- [6] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley & Sons, 2001.
- [7] A. Cichocki and S. Amari, *Adaptive blind signal and image processing - Learning algorithms and applications*. John Wiley & Sons, 2002.
- [8] P. Comon and C. Jutten, Eds., *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press, 2010.
- [9] B. Póczos and A. Lörincz, “Independent subspace analysis using geodesic spanning trees,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2005, pp. 673–680.
- [10] F. Theis, “Towards a general independent subspace analysis,” in *Advances in Neural Information Processing Systems*, 2007, pp. 1361–1368.
- [11] Z. Szabó, B. Póczos, and A. Lörincz, “Undercomplete blind subspace deconvolution,” *Journal of Machine Learning Research*, vol. 8, pp. 1063–1095, 2007.
- [12] A. Sharma and K. Paliwal, “Subspace independent component analysis using vector kurtosis,” *Pattern Recognition*, vol. 39, pp. 2227–2232, 2006.
- [13] J.-F. Cardoso, “Multidimensional independent component analysis,” in *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP ’98)*, 1998.
- [14] P. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.
- [15] D. Lee and H. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems*, vol. 13, 2001, pp. 556–562.
- [16] H. Kameoka, N. Ono, K. Kashino, and S. Sagayama, “Complex NMF: a new sparse representation for acoustic signals,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009.
- [17] B. J. King and L. Atlas, “Single-channel source separation using complex matrix factorization,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 2591–2597, 2011.
- [18] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A universal concept in nonlinear sciences*, ser. Cambridge Nonlinear Science Series. Cambridge University 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, pp. 194–208, 1999.
- [20] P. Tass, M. G. Rosenblum, J. Weule, J. Kurths, A. Pikovsky, J. Volkmann, A. Schnitzler, and H.-J. Freund, “Detection of n:m phase locking from noisy data: Application to magnetoencephalography,” *Physical Review Letters*, vol. 81, pp. 3291–3294, 1998.
- [21] J. M. Palva, S. Palva, and K. Kaila, “Phase synchrony among neuronal oscillations in the human cortex,” *Journal of Neuroscience*, vol. 25, no. 15, pp. 3962–3972, April 2005.
- [22] B. A. Conway, D. M. Halliday, S. F. Farmer, U. Shahani, P. Maas, A. I. Weir, and J. R. Rosenberg, “Synchronization between motor cortex and spinal motoneuronal pool during the performance of a maintained motor task in man,” *Journal of Physiology*, vol. 489, pp. 917–924, 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, pp. 437–447, 2008.
- [24] P. J. Uhlhaas and W. Singer, “Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology,” *Neuron*, vol. 52, 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 Transactions on Neural Networks*, vol. 22, no. 9, pp. 1419–1434, 2011.
- [26] M. Almeida, J. Bioucas-Dias, and R. Vigário, “Independent phase analysis: Separating phase-locked subspaces,” in *Proceedings of the Latent Variable Analysis Conference*, 2010, pp. 189–196.
- [27] A. Ziehe and K.-R. Müller, “TDSEP - an efficient algorithm for blind separation using time structure,” in *International Conference on Artificial Neural Networks*, 1998, pp. 675–680.
- [28] M. Almeida, R. Vigario, and J. Bioucas-Dias, “Phase locked matrix factorization,” in *Proc. of the EUSIPCO conference*, 2011, pp. 1728–1732.
- [29] M. Almeida, R. Vigário, and J. Bioucas-Dias, “Estimation of the common oscillation for phase locked matrix factorization,” in *Proceedings of the International Conference on Pattern Recognition Applications and Methods (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., 1973.
- [31] B. Boashash, “Estimating and interpreting the instantaneous frequency of a signal - part i: Fundamentals,” *Proceedings of the IEEE*, vol. 80, no. 4, pp. 519–538, April 1992.
- [32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [33] L. Grippo and M. Sciandrone, “On the convergence of the block nonlinear Gauss-Seidel method under convex constraints,” *Operations Research Letters*, vol. 26, pp. 127–136, 2000.
- [34] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. Taylor & Francis, 1998.
- [35] M. Almeida, R. Vigário, and J. Bioucas-Dias, “The role of whitening for separation of synchronous sources,” in *Proceedings of the Latent Variable Analysis Conference*, 2012, pp. 139–146.
- [36] M. Almeida, J. Bioucas-Dias, and R. Vigário, “Separation of phase-locked sources in pseudo-real MEG data,” *EUSIPCO Journal on Advances in Signal Processing*, vol. 32, 2013.
- [37] J. Bioucas-Dias and G. Valadao, “Phase unwrapping via graph cuts,” *IEEE Transactions on Image Processing*, vol. 16, pp. 698–709, 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*. Springer, 2013, pp. 51–66.
- [39] A. Ben-Israel and T. Greville, *Generalized inverses: theory and applications*. Springer-Verlag, 2003.

APPENDIX

A. Proof of Theorem 1

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

$$\mathbf{C}_{\mathbf{Z}_{RI}} \equiv \mathbf{Z}_{RI}\mathbf{Z}_{RI}^T = \quad (17)$$

$$= \mathbf{B}\mathbf{Y}_{RI}\mathbf{Y}_{RI}^T\mathbf{B}^T \quad (18)$$

$$= \mathbf{B}\mathbf{C}_{\mathbf{Y}_{RI}}\mathbf{B}^T \quad (19)$$

$$= \mathbf{D}^{-1/2}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{V}\mathbf{D}^{-1/2} \quad (20)$$

$$= \mathbf{I}, \quad (21)$$

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

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

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

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

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

$$[\mathbf{C}_{\mathbf{S}_R}]_{jk} \equiv \mathbb{E}[\text{Re}[s_j(t)]\text{Re}[s_k(t)]] \quad (23)$$

$$= \mathbb{E}[A_j(t)\cos[\phi_j(t)]A_k(t)\cos[\phi_k(t)]]. \quad (24)$$

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

$$[\mathbf{C}_{\mathbf{S}_R}]_{jk} = \mathbb{E}[A_j(t)]\mathbb{E}[A_k(t)]\mathbb{E}[\cos[\phi_j(t)]\cos[\phi_k(t)]] \quad (25)$$

$$= \mathbb{E}[A]^2\mathbb{E}\left[\frac{1}{2}\cos[\phi_j(t) + \phi_k(t)] + \frac{1}{2}\cos[\phi_j(t) - \phi_k(t)]\right] \quad (26)$$

$$= \mathbb{E}[A]^2\mathbb{E}\left[\frac{1}{2}\cos[\phi_j(1) + \phi_k(1) + 2\phi(t)] + \frac{1}{2}\cos[\phi_j(1) - \phi_k(1)]\right] \quad (27)$$

$$= \frac{1}{2}\mathbb{E}[A]^2\mathbb{E}[\cos[\phi_j(1) - \phi_k(1)]] \quad (28)$$

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 $\mathbf{C}_{\mathbf{S}_{RI}} = (\mathbf{C}_{\mathbf{S}_R} + \mathbf{C}_{\mathbf{S}_I})/2$, the expressions for the (j, k) element of $\mathbf{C}_{\mathbf{S}_{RI}}$ are similar to those in (28) and (29) with the factor $\frac{1}{2}$ omitted.

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

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

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 $\mathbf{F} = \text{Re}(\mathbf{G})$, with $\mathbf{G} \equiv \mathbf{x}\mathbf{x}^H$, where the vector \mathbf{x} has components $x_j \equiv e^{i\phi_j(1)}$. \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 $\mathbf{v}^T\text{Im}(\mathbf{G})\mathbf{v} = 0$ because \mathbf{G} is Hermitian, and therefore its imaginary part is skew-symmetric. Then,

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

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 $\mathbf{C}_{\mathbf{S}_{RI}}$, using the result from (30):

$$0 \leq \lambda(\mathbf{F}) \leq N \quad (32)$$

$$0 \leq \lambda\left(\frac{\mathbb{E}[A]^2}{2}\mathbf{F}\right) \leq N\frac{\mathbb{E}[A]^2}{2} \quad (33)$$

$$\frac{\text{Var}[A]}{2} \leq \lambda(\mathbf{C}_{\mathbf{S}_{RI}}) \leq \frac{\text{Var}[A] + N\mathbb{E}[A]^2}{2}. \quad (34)$$

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

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

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, $\rho(\mathbf{C}_{\mathbf{S}_{RI}}^{-1/2}) = \sqrt{1 + N\frac{\mathbb{E}[A]^2}{\text{Var}[A]}}$.

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 $\mathbf{D}_{f1}^{-1} \equiv \mathbf{D}_{f1}^*$, where $*$ represents the entrywise complex conjugate. We thus obtain a new equation $\mathbf{A}_1 = \mathbf{H}_0 \mathbf{A}_2 \mathbf{D}_{f0}$ where $\mathbf{H}_0 \equiv \mathbf{H}_1^\dagger \mathbf{H}_2 \in \mathbb{C}^{N \times N}$ has full rank and $\mathbf{D}_{f0} \equiv \mathbf{D}_{f2} \mathbf{D}_{f1}^* \in \mathbb{D}_1^T$.

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

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

where $e^{i\psi(t)}$ is the t -th diagonal element of \mathbf{D}_{f0} . 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

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

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 $e^{i\psi(t_1)}$ and $e^{i\psi(t_2)}$ 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 $\{-e^{i\gamma}, +e^{i\gamma}\}$, where γ is a real number. Multiplying both sides of this last equation on the right by $\mathbf{D}_{f1} \equiv \mathbf{D}_{f1}^{*-1}$ yields $\mathbf{D}_{f2} = \mathbf{E} \mathbf{D}_{f1}$, as desired.

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

$$\mathbf{M}_1 \mathbf{D}_{z1} \mathbf{A}_1 = \mathbf{M}_2 \mathbf{D}_{z2} \mathbf{A}_2 \quad (38)$$

$$\mathbf{M}_1 \mathbf{E}_1^{-1} \mathbf{E}_1 \mathbf{D}_{z1} \mathbf{A}_1 = \mathbf{M}_2 \mathbf{E}_2^{-1} \mathbf{E}_2 \mathbf{D}_{z2} \mathbf{A}_2 \quad (39)$$

$$\mathbf{M}'_1 \mathbf{D}'_{z1} \mathbf{A}_1 = \mathbf{M}'_2 \mathbf{D}'_{z2} \mathbf{A}_1, \quad (40)$$

where $\mathbf{M}'_{1,2} \equiv \mathbf{M}_{1,2} \mathbf{E}_{1,2}^{-1}$ and $\mathbf{D}'_{z1,2} \equiv \mathbf{E}_{1,2} \mathbf{D}_{z1,2}$. Note that the assumption that all diagonal entries of \mathbf{D}_{z1} are different modulo π ensures that all diagonal entries of \mathbf{D}'_{z1} are different, and the same for \mathbf{D}'_{z2} . Let us split (40) into its real and imaginary parts. Since by definition the real part of $\mathbf{D}'_{z1,2}$ is the identity matrix, we obtain

$$\begin{aligned} \mathbf{M}'_1 \mathbf{A}_1 &= \mathbf{M}'_2 \mathbf{A}_2 \\ \mathbf{M}'_1 \text{Im}(\mathbf{D}'_{z1}) \mathbf{A}_1 &= \mathbf{M}'_2 \text{Im}(\mathbf{D}'_{z2}) \mathbf{A}_2 \end{aligned} \quad (41)$$

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

$$\mathbf{M}'_1 \text{Im}(\mathbf{D}'_{z1}) \mathbf{M}_1'^{-1} = \mathbf{M}'_2 \text{Im}(\mathbf{D}'_{z2}) \mathbf{M}_2'^{-1} \quad (42)$$

Since $\text{Im}(\mathbf{D}'_{z1})$ is a diagonal matrix, the left-hand side can be interpreted as an eigenvalue decomposition of some matrix, where $\text{Im}(\mathbf{D}'_{z1})$ contains the eigenvalues in its diagonal and \mathbf{M}'_1 contains the corresponding eigenvectors in its columns.¹⁵

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

A similar interpretation can be given to the right-hand side. Furthermore, since all diagonal elements of $\mathbf{D}'_{z1,2}$ are different from one another, all eigenvalues have algebraic and geometric multiplicities of 1 on the left-hand side and on the right-hand side. Therefore, $\text{Im}(\mathbf{D}'_{z1}) = \text{Im}(\mathbf{D}'_{z2})$ 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 $\mathbf{A}_1 = \mathbf{P} \mathbf{A}_2$, where \mathbf{P} is a permutation of a diagonal matrix with nonzero entries on the diagonal.

We now have, from (38),

$$\mathbf{M}_1 \mathbf{D}_{z1} \mathbf{P} \mathbf{A}_2 = \mathbf{M}_2 \mathbf{D}_{z2} \mathbf{A}_2 \quad (43)$$

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

$$\mathbf{M}_1 \mathbf{D}_{z1} \mathbf{P} = \mathbf{M}_2 \mathbf{D}_{z2}. \quad (44)$$

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

$$e^{i\alpha_1 m} \mathbf{m}_{1m} = e^{i\alpha_2 n} \mathbf{m}_{2n}, \quad (45)$$

where $e^{i\alpha_1 m}$ is the (m, m) -th entry of \mathbf{D}_{z1} , $e^{i\alpha_2 n}$ is the (n, n) -th entry of \mathbf{D}_{z2} , \mathbf{m}_{1m} is the m -th column of \mathbf{M}_1 , and \mathbf{m}_{2n} is the n -th 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 $e^{i\theta} \mathbf{z}_1$ and $e^{i\theta} \mathbf{z}_2$ all have non-zero 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 $e^{i\theta} \mathbf{z}_1$ and $e^{i\theta} \mathbf{z}_2$ everywhere.

D. Proof of theorem 4

Define $\mathbf{H}_1 \equiv \mathbf{M}_1 \mathbf{D}_{z1}$ and $\mathbf{H}_2 \equiv \mathbf{M}_2 \mathbf{D}_{z2}$. Theorem 2 can be applied to the factorizations $\mathbf{H}_1 \mathbf{A}_1 \mathbf{D}_{z1}$ and $\mathbf{H}_2 \mathbf{A}_2 \mathbf{D}_{z2}$. Therefore, $\mathbf{D}_{f1} = \mathbf{E} \mathbf{D}_{f2}$, where $\mathbf{E} = e^{i\gamma} \mathbf{I}_\pm$, 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}_{f1} , we get the two factorizations

$$\mathbf{Y} = \mathbf{M}_1 \mathbf{D}_{z1} e^{i\gamma} \mathbf{A}_1 \mathbf{I}_\pm \mathbf{D}_{f2} \quad \text{and} \quad \mathbf{Y} = \mathbf{M}_2 \mathbf{D}_{z2} \mathbf{A}_2 \mathbf{D}_{f2}.$$

Define $\mathbf{D}'_{z1} \equiv \mathbf{D}_{z1} e^{i\gamma}$ and $\mathbf{A}'_1 \equiv \mathbf{A}_1 \mathbf{I}_\pm$. Then, theorem 3 is applicable to the two factorizations $\mathbf{Y} \mathbf{D}_{f2}^* = \mathbf{M}_1 \mathbf{A}'_1 \mathbf{D}'_{z1}$ and $\mathbf{Y} \mathbf{D}_{f2}^* = \mathbf{M}_2 \mathbf{A}_2 \mathbf{D}_{z2}$.

Therefore, we get $\mathbf{M}_1 = \mathbf{M}_2$, $\mathbf{z}'_1 = \mathbf{z}_2$ (i.e., $\mathbf{z}_1 = e^{i\gamma} \mathbf{z}_2$, 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 non-negative by assumption, the last of these equalities implies that $\mathbf{A}_1 = \mathbf{A}_2$ up to permutation and positive scaling.