Blind separation of non-negative source signals using multiplicative updates and subspace projection

Alexander Bertrand†, Marc Moonen†

Dept. of Electrical Engineering (ESAT-SCD) - Katholieke Universiteit Leuven
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
E-mail: alexander.bertrand@esat.kuleuven.be
marc.moonen@esat.kuleuven.be
Phone: +32 16 321899, Fax: +32 16 321970

Abstract

In this paper, we consider a noise-free blind source separation problem with independent non-negative source signals, also known as non-negative independent component analysis (NICA). We assume that the source signals are well-grounded, which means that they have a non-vanishing pdf in a positive neighborhood of zero. We propose a novel algorithm, referred to as multiplicative NICA (M-NICA), which uses multiplicative updates together with a subspace projection based correction step to reconstruct the original source signals from the observed linear mixtures, and which is based only on second order statistics. A multiplicative update has the facilitating property that it preserves non-negativity, and does not depend on a user-defined learning rate, as opposed to gradient based updates such as in the non-negative PCA (NPCA) algorithm. We provide batch mode simulations of M-NICA and compare its performance to NPCA, for different types of signals. It is observed that M-NICA generally yields a better unmixing accuracy, but converges slower than NPCA. Especially when the amount of data samples is small, M-NICA signficantly outperforms NPCA. Furthermore, a sliding window implementation of both algorithms is described and simulated, where M-NICA is observed to provide the best results.

Keywords: Non-negative blind source separation, Non-negative independent component analysis (NICA), Multiplicative updating, Multi-channel signal processing

† EURASIP member

I. INTRODUCTION

Assume that we observe a set of instantaneous linear mixtures of mutually independent source signals. The goal of independent component analysis (ICA) is then to reconstruct the original source signals from the observed mixtures. This problem is widely studied in literature (see [1], [2] for a survey), usually under the general assumption that the source signals are nongaussian and that the mixing matrix is full rank. However, if some prior knowledge on the source signals is available, this knowledge may be exploited to design more efficient unmixing algorithms. In this paper, we consider an ICA problem with *non-negative* sources, i.e. we collect observations of a J-channel signal $\mathbf{y}$ that satisfies

$$\mathbf{y} = \mathbf{A}\mathbf{s} \tag{1}$$

where $\mathbf{s} = [s_1 \dots s_N]^T$ is a vector of N mutually independent source signals with $s_n \geq 0, \forall n \in \{1 \dots N\}$, and where $\mathbf{A}$ is an unknown $J \times N$ mixing matrix with $J \geq N$. In [3], this problem is referred to as the *non-negative independent component analysis* (NICA) problem. Non-negativity arises in many practical problems, e.g. source activity detection [4], unmixing spectral data [5], hyperspectral imaging [6], [7], chemistry [8], environmetrics [9], music transcription [10], etc.

A closely related problem is 'non-negative matrix factorization' (NMF) [11], [12], in which a non-negative matrix is factorized in two smaller non-negative matrices. This corresponds to the case where the mixing matrix $\mathbf{A}$ is also assumed to be non-negative. However, NMF does not take independence of the sources into account, and therefore NMF algorithms often yield suboptimal results when applied to the NICA problem.

By making additional assumptions on the source signals, several algorithms are proposed to solve the NICA problem [3], [13], [14]. In this paper, we add the assumption that the sources are well-grounded, as also done in [3]. This means that all sources have a non-zero pdf in any positive neighborhood of zero, i.e. $\forall\, \delta > 0$: $Pr(s_n < \delta) > 0$, for all source signals s_n, $n = 1 \dots N$. Well-groundedness of the sources is a weaker assumption than the locally-dominant assumption[1] in [13], [14], and it is often satisfied in practice, e.g. when the sources have an on-off behavior or when the source signals are sparse. The locally-dominant assumption is more easily violated, especially for short time windows. We will consider two different algorithms to solve the NICA problem with well-grounded sources: the non-negative PCA algorithm (NPCA), which is introduced in [3], and the multiplicative NICA algorithm (M-NICA), which is a novel approach to tackle the NICA problem.

[1]The locally-dominant assumption states that for each source s_j in a set of N source signals $\{s_1, \dots, s_N\}$, there is a sample time t_j in the data set such that $s_j[t_j] \neq 0$ and $s_i[t_j] = 0$ for all $i \neq j$.

The NPCA algorithm [3] first decorrelates the data by applying a whitening procedure without taking the non-negativity into account. In a second step, the algorithm computes a rotation matrix that restores the non-negativity of the data. This is done by means of a gradient-descent algorithm with additional correction steps to preserve orthogonality. The learning rate of the NPCA algorithm is a critical parameter to obtain satisfying results. If the learning rate is chosen too small, the algorithm can have extremely slow convergence. On the other hand, if the learning rate is too high, it is possible that the NPCA algorithm does not converge at all.

The M-NICA algorithm, on the other hand, decorrelates the data while at the same time maintaining non-negativity, by means of a multiplicative update step. Multiplicative updating is a popular technique to solve non-negative optimization problems, e.g. [12], [15]. Since a multiplicative update results in data that is not in the original signal subspace, it requires a correction step based on a subspace projection to restore the original signal subspace. The M-NICA algorithm has the facilitating property that it does not depend on a user-defined learning rate, as opposed to the NPCA algorithm. This is particularly relevant in applications for which a step-size search is impossible or undesirable.

NPCA and M-NICA have similar computational complexity. We will compare the performance of both algorithms by means of simulations with different types of signals. As will be demonstrated, the convergence speed and the unmixing accuracy of both algorithms heavily depends on the type of signals involved. By averaging over multiple experiments, it is observed that M-NICA generally provides a better unmixing accuracy, but at a slower convergence rate than NPCA. The difference between the unmixing accuracy of M-NICA and NPCA becomes more significant in cases where the amount of available data samples is small, where the former is observed to provide the best results. Also in an adaptive sliding window implementation, M-NICA clearly outperforms NPCA in terms of unmixing accuracy, at a slightly slower adaptation speed.

The outline of the paper is as follows. In section II, the NPCA algorithm is briefly reviewed. The batch mode M-NICA algorithm is described in section III. In section IV, a sliding window implementation of the M-NICA algorithm is described. Batch mode simulations of M-NICA and NPCA are provided in section V. The performance of the sliding window implementations of M-NICA and NPCA are analyzed in section VI. Conclusions are given in section VII.

II. NON-NEGATIVE PCA (NPCA)

In [16], the following theorem is proven:

Theorem 2.1: Let $\mathbf{s}$ be an N-dimensional vector of non-negative and well-grounded mutually independent source signals with unit variance, and let $\mathbf{z} = \mathbf{U}\mathbf{s}$ be an orthonormal rotation of $\mathbf{s}$ where

$\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}_N$, with $\mathbf{I}_N$ denoting the $N \times N$ identity matrix. Then $\mathbf{z}$ is a permutation of $\mathbf{s}$ if and only if the signals in $\mathbf{z}$ are non-negative with probability 1.

This theorem states that any orthogonal mixture of well-grounded mutually independent non-negative sources, that preserves the non-negativity, must be a permutation of the sources. It is noted that, although the well-groundedness of the source signals is not explicitly exploited in the algorithms described in the sequel, it is a crucial assumption. The intuition behind this is that, if the source signals are well-grounded, there is only one possible rotation to completely fit the rectangular (decorrelated) data cloud in the positive orthant [16].

In [3], Theorem 2.1 is used to derive the non-negative PCA algorithm (NPCA), which is able to solve NICA problems with well-grounded sources. The algorithm uses only second order statistics, which makes it very efficient compared to ICA algorithms that use higher order statistics. The outline of the NPCA algorithm is as follows (here described in batch mode):

1) Let $\mathbf{C}_y = E\{(\mathbf{y}-\overline{\mathbf{y}})(\mathbf{y}-\overline{\mathbf{y}})^T\}$, where $E\{.\}$ denotes the expectation operator and where $\overline{\mathbf{y}} = E\{\mathbf{y}\}$. Compute the eigenvalue decomposition of the covariance matrix $\mathbf{C}_y$, i.e. $\mathbf{C}_y = \mathbf{E}\mathbf{D}\mathbf{E}^T$ with $\mathbf{D}$ a diagonal matrix containing the eigenvalues of $\mathbf{C}_y$ on its diagonal, and with $\mathbf{E}$ containing the corresponding eigenvectors in its columns. Since $\mathbf{C}_y$ is a rank N matrix, we can write $\mathbf{C}_y = \overline{\mathbf{E}}\,\overline{\mathbf{D}}\,\overline{\mathbf{E}}^T$, with $\overline{\mathbf{D}}$ an $N \times N$ diagonal matrix, containing the N non-zero eigenvalues of $\mathbf{C}_y$ on its diagonal, and with $\overline{\mathbf{E}}$ the $J \times N$ matrix containing the corresponding eigenvectors.
2) Whiten the signal $\mathbf{y}$ with a whitening matrix[2]

$$\mathbf{V} = \overline{\mathbf{D}}^{-\frac{1}{2}}\overline{\mathbf{E}}^T \tag{2}$$

yielding the whitened compressed signal $\mathbf{v} = \mathbf{V}\mathbf{y}$.
3) Assume w.l.o.g. that the sources s_n, $n = 1 \ldots N$, have unit variance[3], such that $\mathbf{C}_s = E\{(\mathbf{s}-\overline{\mathbf{s}})(\mathbf{s}-\overline{\mathbf{s}})^T\} = \mathbf{I}_N$. Then the matrix $\mathbf{Z} = \mathbf{V}\mathbf{A}$ is orthogonal, since $\mathbf{Z}\mathbf{Z}^T = \mathbf{V}\mathbf{A}\mathbf{A}^T\mathbf{V}^T = \mathbf{V}\mathbf{A}\mathbf{C}_s\mathbf{A}^T\mathbf{V}^T = \mathbf{V}\mathbf{C}_y\mathbf{V}^T = \mathbf{I}_N$. According to Theorem 2.1, it is then sufficient to find an orthogonal matrix $\mathbf{W}$ such that $\mathbf{z} = \mathbf{W}\mathbf{v} = \mathbf{W}\mathbf{Z}\mathbf{s}$ is non-negative with probability 1. This matrix $\mathbf{W}$ can be computed by means of the following learning rule:

$$\mathbf{W}^{\text{temp}} = \mathbf{W}^i - \eta\mathbf{M}^i\mathbf{W}^i \tag{3}$$

[2]In [3], a symmetric whitening matrix was chosen, i.e. $\mathbf{V} = \mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T$. This is however only possible when $\mathbf{y}$ is an N-dimensional vector, i.e. when the mixing matrix $\mathbf{A}$ is square. If $J > N$, the whitening matrix (2) performs a dimension reduction, in addition to a decorrelation.

[3]If this is not the case, the source signals can be scaled accordingly, yielding a reciprocal scaling of the columns of the mixing matrix $\mathbf{A}$

$$\mathbf{W}^{i+1} = \left(\mathbf{W}^{\text{temp}}\mathbf{W}^{\text{temp}\,T}\right)^{-\frac{1}{2}}\mathbf{W}^{\text{temp}} \tag{4}$$

with

$$\mathbf{M}^i = E\{f(\mathbf{z}^i)\mathbf{z}^{i\,T} - \mathbf{z}^i f(\mathbf{z}^{i\,T})\} \tag{5}$$

where $\mathbf{z}^i = \mathbf{W}^i\mathbf{v}$, $f(z_n) = \min(0, z_n)$ and with η denoting a positive learning rate.

Since (3) does not enforce orthogonality of $\mathbf{W}$, the correction step (4) is added to guarantee orthogonality of $\mathbf{W}$. Let $\mathbf{y}[k]$ denote the observation of $\mathbf{y}$ at time k, and let M denote the number of observations of $\mathbf{y}$. Then the expected value in (5) can be computed by simple averaging over the M transformed observations $\mathbf{z}^i[k]$, $k = 1 \ldots M$. Assuming that $M \gg N$, then (5) is the computationally most expensive step of the NPCA algorithm, yielding an overall complexity of $O(N^2M)$.

It is observed that the learning rate η is a crucial parameter for the algorithm to converge, i.e. its value should be small enough to guarantee convergence. However, a too small η results in a very slow convergence. In [17], an adaptive strategy is proposed to update η. Although convergence can be enforced in this way, the strategy is observed to yield rather conservative learning rates. It remains unclear how an optimal value for η can be chosen automatically to provide a fast convergence.

III. Multiplicative NICA (M-NICA)

In this section, we present a new algorithm to solve the NICA problem with well-grounded sources. It is based on the following corollary, which follows straightforwardly from Theorem 2.1:

Corollary 3.1: Let $\mathbf{s}$ be an N-dimensional vector of non-negative and well-grounded mutually independent source signals, and let $\mathbf{y} = \mathbf{A}\mathbf{s}$ with $\mathbf{A}$ a full column rank $J \times N$ mixing matrix. Let $\mathbf{z} = \mathbf{K}\mathbf{y}$ where $\mathbf{K}$ is a $N \times J$ unmixing matrix. Then $\mathbf{z}$ is a permutation of $\mathbf{s}$ if and only if the signals in $\mathbf{z}$ are mutually uncorrelated and non-negative with probability 1.

Proof: We only prove the '$\Leftarrow$' direction, since the '$\Rightarrow$' direction is trivially proved. We thus assume that $\mathbf{z}$ is non-negative and that its signals are mutually uncorrelated, i.e.

$$E\{(\mathbf{z} - \overline{\mathbf{z}})(\mathbf{z} - \overline{\mathbf{z}})^T\} = \mathbf{I}_N\ . \tag{6}$$

Since $\mathbf{z} = \mathbf{K}\mathbf{y}$ and $\mathbf{y} = \mathbf{A}\mathbf{s}$, expression (6) can be rewritten as

$$\mathbf{K}\mathbf{A}E\{(\mathbf{s} - \overline{\mathbf{s}})(\mathbf{s} - \overline{\mathbf{s}})^T\}\mathbf{A}^T\mathbf{K}^T = \mathbf{I}_N\ . \tag{7}$$

Assume w.l.o.g. that the source signals in $\mathbf{s}$ have unit variance. Since these source signals are mutually independent, they are uncorrelated, and therefore (7) becomes

$$\mathbf{U}\mathbf{U}^T = \mathbf{I}_N \tag{8}$$

where $\mathbf{U} = \mathbf{K}\mathbf{A}$. Expression (8) shows that $\mathbf{U}$ is a $N \times N$ orthogonal matrix. Since $\mathbf{z}$ is non-negative and $\mathbf{z} = \mathbf{U}\mathbf{s}$, Theorem 2.1 shows that $\mathbf{z}$ is a permutation of $\mathbf{s}$.

To solve the NICA problem (1), it is thus sufficient to find an $N \times J$ unmixing matrix $\mathbf{K}$ that results in N non-negative uncorrelated signals. Notice that the first step of the NPCA algorithm decorrelates the data by applying a straightforward whitening procedure without taking the non-negativity into account. In the second step, the algorithm computes a rotation matrix that restores the non-negativity of the data, while preserving the decorrelation. In the M-NICA algorithm described infra, we will decorrelate the data while preserving the non-negativity. This has several advantages. Since we use a multiplicative update, the algorithm does not require any user-defined learning rate. Furthermore, since we explicitly minimize the correlation under non-negativity constraints, the algorithm is more robust than NPCA when using small sample sets (as will be demonstrated in section V-D). For the sake of an easy exposition, we will first describe the M-NICA algorithm in batch mode. A sliding window algorithm will be described in section IV.

A. Multiplicative decorrelation with subspace projection

Assume we collect a $J \times M$ data matrix $\mathbf{Y}$ that contains M observations $\mathbf{y}[k]$, $k = 1 \dots M$, in its columns. We will try to find an unmixing matrix $\mathbf{K}$ such that the rows of the $N \times M$ matrix $\mathbf{S} = \mathbf{K}\mathbf{Y}$ are uncorrelated and only contain non-negative values. Notice that $\mathbf{S}$ does not necessarily contain the samples $\mathbf{s}[k]$, $k = 1 \dots M$, in its columns, since it depends on the choice of $\mathbf{K}$ (even when $\mathbf{K}$ yields perfect unmixing, there remains a scaling and permutation ambiguity compared to the signals in $\mathbf{s}$).

Define $\mathbf{C}_S = (\mathbf{S} - \overline{\mathbf{S}})(\mathbf{S} - \overline{\mathbf{S}})^T$, where $\overline{\mathbf{S}}$ denotes the $N \times M$ matrix for which each column contains the sample mean of the rows of $\mathbf{S}$, i.e. $\overline{\mathbf{S}} = \frac{1}{M}\mathbf{S}\,\mathbf{1}_M\mathbf{1}_M^T$, where $\mathbf{1}_M$ denotes an M-dimensional column vector in which each entry is 1. For notational convenience, we introduce the matrix $\mathbf{P} = \mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T$, to write $\mathbf{C}_S = (\mathbf{S} - \overline{\mathbf{S}})(\mathbf{S} - \overline{\mathbf{S}})^T = \mathbf{S}\mathbf{P}\mathbf{P}^T\mathbf{S}^T = \mathbf{S}\mathbf{P}\mathbf{S}^T$. Let

$$F(\mathbf{S}) = \sum_{n,m} \frac{\left[\mathbf{S}\mathbf{P}\mathbf{S}^T\right]_{nm}^2}{\left[\mathbf{S}\mathbf{P}\mathbf{S}^T\right]_{nn}\left[\mathbf{S}\mathbf{P}\mathbf{S}^T\right]_{mm}} \tag{9}$$

i.e. the function $F(\mathbf{S})$ evaluates the sum of the squared (cross-)correlation coefficients of the rows of $\mathbf{S}$.

According to Corollary 3.1, to obtain the original source signals in the rows of $\mathbf{S}$, it is sufficient to construct $\mathbf{S} = \mathbf{KY}$ such that $\mathbf{S} \geq 0$ and $\mathbf{C}_S$ is a diagonal matrix. This is translated into the following optimization problem:

$$\min_{\mathbf{S}} F(\mathbf{S}) \tag{10}$$

$$\text{s.t.} \begin{cases} \mathbf{S} \geq 0 \\ \exists\, \mathbf{K} \in \mathbb{R}^{N \times J} : \mathbf{S} = \mathbf{KY} \ . \end{cases} \tag{11}$$

The first constraint in (11) enforces non-negativity and the second constraint links the matrix $\mathbf{S}$ to the observations in $\mathbf{Y}$, such that both have the same row space. The minimization of the cost function[4] $F(\mathbf{S})$ yields decorrelation of the rows of $\mathbf{S}$. The function $F(\mathbf{S})$ has multiple global minima, which are all equal to N, corresponding to the case where all cross-correlation coefficients are zero. Since the cost function $F(\mathbf{S})$ is non-convex, it has multiple stationary points. However, as shown by the following theorem, every local minimizer $\mathbf{S}^*$ corresponds to perfectly uncorrelated source signals in the rows of $\mathbf{S}^*$.

Theorem 3.2: Let $\mathbf{S}^*$ denote a local minimizer of $F(\mathbf{S})$ (without taking the constraints (11) into account). Then the rows of $\mathbf{S}^*$ are uncorrelated, i.e. $\mathbf{C}_S^* = (\mathbf{S}^* - \overline{\mathbf{S}}^*)(\mathbf{S}^* - \overline{\mathbf{S}}^*)^T$ is a diagonal matrix.

Proof: In the sequel, we ignore the points $\mathbf{S}$ for which $F(\mathbf{S})$ does not exist, i.e. the case where $\mathbf{S}$ has one or more zero-variance rows. The gradient of the cost function (9) is

$$\nabla F(\mathbf{S}) = 4 \left(\mathbf{\Lambda}_1^{-1} \mathbf{SPS}^T \mathbf{\Lambda}_1^{-1} - \mathbf{\Lambda}_1^{-1} \mathbf{\Lambda}_2 \right) \mathbf{SP} \tag{12}$$

with

$$\mathbf{\Lambda}_1 = D\left\{ \mathbf{SPS}^T \right\} \tag{13}$$

$$\mathbf{\Lambda}_2 = D\left\{ \left(\mathbf{\Lambda}_1^{-1} \mathbf{SPS}^T \right)^2 \right\} \tag{14}$$

and with $D\{\mathbf{X}\}$ denoting the operator that sets all off-diagonal elements of $\mathbf{X}$ to zero. Let $\mathbf{S}^*$ denote a local minimizer of F, and therefore it satisfies $\nabla F(\mathbf{S}^*) = 0$, which is equivalent to

$$\mathbf{\Lambda}_1^{*-1} \mathbf{S}^* \mathbf{P} \mathbf{S}^{*\,T} \mathbf{\Lambda}_1^{*-1} \mathbf{S}^* \mathbf{P} = \mathbf{\Lambda}_1^{*-1} \mathbf{\Lambda}_2^* \mathbf{S}^* \mathbf{P} \tag{15}$$

where $\mathbf{\Lambda}_1^*$ and $\mathbf{\Lambda}_2^*$ are defined by (13)-(14) with $\mathbf{S}$ replaced by $\mathbf{S}^*$.

Note that $\mathbf{S}^*\mathbf{P} = (\mathbf{S}^* - \overline{\mathbf{S}}^*)$ has full row rank. This can be shown by contradiction as follows. Assume

[4]Notice that we do not use the cost function $\sum_{n,m} \left[\mathbf{SPS}^T - \mathbf{I}_N \right]_{nm}^2$. Although this cost function would yield simpler updating formulas, cost function (9) is observed to yield a better performance due to its implicit normalization. This normalization makes the resulting updating formulas independent of the variance of the elements in $\mathbf{S}$.

that $\mathbf{S}^*\mathbf{P}$ does not have full row rank. Then either $\mathbf{S}^*$ has a zero variance row, which can be excluded since then $F(\mathbf{S}^*)$ does not exist, or $\mathbf{S}^*$ has at least one row which is a linear combination of the other rows. Let the i-th row $[\mathbf{S}]_{i,:}$ denote such a row which is a linear combination of the other rows. Let $\mathbf{e}^T$ be an $M \times 1$ row vector with random numbers, which are uncorrelated with any row in $\mathbf{S}$. Then adding the vector $\alpha\mathbf{e}^T$ to the row $[\mathbf{S}]_{i,:}$, with α denoting any positive number, will result in a decrease of the cost function F. This shows that there exists a descent direction in $\mathbf{S}^*$. However, since $\mathbf{S}^*$ is a local minimizer of F, no such direction can exist in the point $\mathbf{S}^*$.

Since $\mathbf{S}^*\mathbf{P} = (\mathbf{S}^* - \overline{\mathbf{S}}^*)$ has full row rank, $\mathbf{S}^*\mathbf{P}\mathbf{P}^T{\mathbf{S}^*}^T = \mathbf{S}^*\mathbf{P}{\mathbf{S}^*}^T$ has full rank and non-zero elements on its diagonal. Using this, and since both $\mathbf{\Lambda}_1^*$ and $\mathbf{\Lambda}_2^*$ are diagonal matrices, (15) is equivalent to

$$\mathbf{S}^*\mathbf{P}{\mathbf{S}^*}^T = \mathbf{\Lambda}_1^*\mathbf{\Lambda}_2^* \,. \tag{16}$$

Since $\mathbf{S}^*\mathbf{P}{\mathbf{S}^*}^T = (\mathbf{S}^* - \overline{\mathbf{S}}^*)(\mathbf{S}^* - \overline{\mathbf{S}}^*)^T = \mathbf{C}_S^*$, and since the righthand side of (16) is a diagonal matrix, the theorem is proven.

Theorem 3.2 implies that any local minimizer $\mathbf{S}^*$ of F satisfies $F(\mathbf{S}^*) = N$ and hence is a global minimizer. It is thus sufficient to find a local minimum of (9) that satisfies the constraints (11), to solve the NICA problem.

The first constraint of (11) is a non-negativity constraint on the matrix $\mathbf{S}$. A popular way to minimize a cost function $F(\mathbf{S})$ under non-negativity constraints, is to use multiplicative update rules (cfr. e.g. [12], [15]). Multiplicative optimization algorithms are usually easy to implement compared to general contrained optimization (CO) techniques, and they do not require any step size search. The multiplicative update rules can be derived if the gradient of the cost function $F(\mathbf{S})$ can be split into a positive part and a negative part, i.e.

$$\nabla F(\mathbf{S}) = \nabla^+ F(\mathbf{S}) - \nabla^- F(\mathbf{S}) \tag{17}$$

with $[\nabla^- F(\mathbf{S})]_{nm} \geq 0$ and $[\nabla^+ F(\mathbf{S})]_{nm} \geq 0$, $n = 1 \ldots N$, $m = 1 \ldots M$, then the following multiplicative update rule can be used [15]:

$$[\mathbf{S}]_{nm} \leftarrow [\mathbf{S}]_{nm} \frac{[\nabla^- F(\mathbf{S})]_{nm}}{[\nabla^+ F(\mathbf{S})]_{nm}} \,. \tag{18}$$

Notice that, if $\mathbf{S}$ is initialized with non-negative numbers, all of its elements remain non-negative under the update (18), and the non-negativity constraint of (11) is automatically satisfied. There exist two kinds of fixed points for (18). The first satisfies $\nabla^+ F(\mathbf{S}) = \nabla^- F(\mathbf{S})$, yielding $\nabla F(\mathbf{S}) = 0$, i.e. a stationary point of the cost function $F(\mathbf{S})$. The other is $[\mathbf{S}]_{nm} = 0$, $n = 1 \ldots N$, $m = 1 \ldots M$. Notice that the updating procedure (18) cannot converge to a stationary point of F if certain elements of $\mathbf{S}$ that are

non-zero in any stationary point, are set to zero. Indeed, any element that has a value of zero remains zero in all future iterations. We will refer to this as 'false zeros'.

It is generally difficult to prove convergence of multiplicative update formulas of the form (18). However, for many cost functions F, update (18) is found to converge to a local minimizer of F. This can be explained intuitively as follows. The variable $[\mathbf{S}]_{nm}$ decreases when $[\nabla^+ F(\mathbf{S})]_{nm} > [\nabla^- F(\mathbf{S})]_{nm}$, i.e. when $[\nabla F(\mathbf{S})]_{nm} > 0$. This means that the value changes in the opposite direction of the gradient. Therefore (18) is similar to a gradient descent update, where the step-size is different for each variable and in each step. More specifically, (18) is equivalent to a *natural* gradient descent update, as pointed out in [15]. A natural gradient learning algorithm has the convenient property that it has isotropic convergence around any local optimum, independent of the model parametrization or the signals being processed [18].

By applying this technique to the gradient of $F(\mathbf{S})$, as given in (12)-(14), the following updating formula for the matrix $\mathbf{S}$ is found[5]:

$$[\mathbf{S}]_{nm} \leftarrow [\mathbf{S}]_{nm} \frac{\left[\overline{\mathbf{S}}\mathbf{S}^T\mathbf{\Lambda}_1^{-1}\mathbf{S} + \mathbf{S}\mathbf{S}^T\mathbf{\Lambda}_1^{-1}\overline{\mathbf{S}} + \mathbf{\Lambda}_2\mathbf{S}\right]_{nm}}{\left[\overline{\mathbf{S}}\mathbf{S}^T\mathbf{\Lambda}_1^{-1}\overline{\mathbf{S}} + \mathbf{S}\mathbf{S}^T\mathbf{\Lambda}_1^{-1}\mathbf{S} + \mathbf{\Lambda}_2\overline{\mathbf{S}}\right]_{nm}} . \tag{19}$$

Notice that this update does not take the second constraint of (11) into account. Therefore, an additional correction step is required after each update (19). To enforce the second constraint of (11), the rows of $\mathbf{S}$ are projected onto the signal subspace $\mathcal{S}$, which is equal to the row space of $\mathbf{Y}$:

$$\mathbf{S} \leftarrow \mathscr{P}_{\mathcal{S}}\{\mathbf{S}\} \tag{20}$$

where $\mathscr{P}_{\mathcal{S}}\{\mathbf{X}\}$ denotes the projection operator that projects the rows of the matrix $\mathbf{X}$ onto the signal subspace $\mathcal{S}$.

Notice that the projection $\mathscr{P}_{\mathcal{S}}\{\mathbf{S}\}$ can result in negative values in $\mathbf{S}$. To preserve non-negativity, $\mathbf{S}$ should actually be projected onto $\mathcal{S}^+ = \mathcal{S} \cap \mathbb{P}$ where $\mathbb{P}$ denotes the positive orthant, i.e.

$$\mathbf{S} \leftarrow \mathscr{P}_{\mathcal{S}^+}\{\mathbf{S}\} . \tag{21}$$

This projection can be iteratively computed with Dykstra's algorithm [19]. However, to reduce the complexity of the M-NICA algorithm, we use a heuristic procedure instead, as described in the next section.

Remark: It is noted that general constrained optimization (CO) techniques can also be used to solve the

[5]We replaced $\mathbf{S}\mathbf{P}\mathbf{S}^T$ with $(\mathbf{S}-\overline{\mathbf{S}})\mathbf{S}^T$, instead of the equivalent substitution $\mathbf{S}\mathbf{P}\mathbf{S}^T = \mathbf{S}\mathbf{P}\mathbf{P}^T\mathbf{S}^T = (\mathbf{S}-\overline{\mathbf{S}})(\mathbf{S}-\overline{\mathbf{S}})^T$.

problem $\min_{\mathbf{K}} F(\mathbf{KY})$ s.t. $\mathbf{KY} \geq 0$, which is equivalent to (10)-(11). Experiments[6] indicate that only the interior point (IP) method [20] gives good results that are comparable to the unmixing performance of M-NICA and NPCA. However, for the experiments in section V, the computation time of the IP method is roughly the double[7] of the computation time of M-NICA and NPCA. Furthermore, the M-NICA algorithm (see section III-B) is much simpler to implement compared to an IP method, where in each iteration the Hessian matrix must be evaluated (i.e. second order numerical differentiation) or approximated, and a corresponding IP-KKT system must be solved with a subsequent step-size search. Each IP-KKT system is of large dimension due to the large amount of inequality constraints that inforce non-negativity of each unmixed sample.

B. The multiplicative NICA algorithm (M-NICA)

The following fixed-point algorithm is used to solve (9)-(11), and is referred to as multiplicative non-negative ICA (M-NICA):

1) *Initialization*:

   a) $\forall\, n = 1 \ldots N, \forall\, m = 1 \ldots M:\ [\mathbf{S}]_{nm} \leftarrow |[\mathbf{Y}]_{nm}|$

   b) Replace $\mathbf{Y}$ by its best rank N approximation by means of the singular value decomposition (SVD), i.e.

$$\{\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}\} \leftarrow \text{SVD}\,(\mathbf{Y}) \tag{22}$$

$$\mathbf{Y} \leftarrow \overline{\mathbf{U}}\ \overline{\mathbf{\Sigma}}\ \overline{\mathbf{V}}^T \tag{23}$$

   where $\overline{\mathbf{\Sigma}}$ is the $N \times N$ diagonal matrix containing the N largest singular values[8] of $\mathbf{Y}$ on its diagonal, and where the corresponding left and right singular vectors are stored in the columns of $\overline{\mathbf{U}}$ and $\overline{\mathbf{V}}$ respectively.

2) *Decorrelation step*:

$$\begin{gathered}\forall\, n = 1 \ldots N, \forall\, m = 1 \ldots M: \\ \left[\mathbf{S}^{\text{temp}}\right]_{nm} \leftarrow [\mathbf{S}]_{nm} \frac{\left[\overline{\mathbf{S}}\mathbf{S}^T\mathbf{\Lambda}_1^{-1}\mathbf{S} + \mathbf{S}\mathbf{S}^T\mathbf{\Lambda}_1^{-1}\overline{\mathbf{S}} + \mathbf{\Lambda}_2\mathbf{S}\right]_{nm}}{\left[\overline{\mathbf{S}}\mathbf{S}^T\mathbf{\Lambda}_1^{-1}\overline{\mathbf{S}} + \mathbf{S}\mathbf{S}^T\mathbf{\Lambda}_1^{-1}\mathbf{S} + \mathbf{\Lambda}_2\overline{\mathbf{S}}\right]_{nm}}\end{gathered} \tag{24}$$

[6]We also tried an active set method and a Levenberg-Marquardt based algorithm [20]. Both methods give good results in some cases, but their separation performance varies significantly over multiple Monte-Carlo experiments. Especially in scenarios with many sources ($N > 2$) and/or overdetermined observations ($J > N$), both methods generally yield very poor results.

[7]Based on Matlab's *fmincon* command.

[8]Notice that, if noise were present, this step will remove some noise from the observations. In the noise-free case, $\mathbf{Y}$ has exactly N non-zero singular values.

with

$$\overline{\mathbf{S}} = \frac{1}{M}\mathbf{S}\,\mathbf{1}_M\mathbf{1}_M^T \tag{25}$$

$$\mathbf{C}_S = (\mathbf{S} - \overline{\mathbf{S}})(\mathbf{S} - \overline{\mathbf{S}})^T \tag{26}$$

$$\mathbf{\Lambda}_1 = D\left\{\mathbf{C}_S\right\} \tag{27}$$

$$\mathbf{\Lambda}_2 = D\left\{\left(\mathbf{\Lambda}_1^{-1}\mathbf{C}_S\right)^2\right\}\ . \tag{28}$$

3) *Signal subspace projection step*:

$$\forall\, n = 1 \ldots N, \forall\, m = 1 \ldots M:$$

$$[\mathbf{S}]_{nm} \leftarrow \max\left(\left[\mathbf{S}^{\text{temp}}\ \overline{\mathbf{V}}\ \overline{\mathbf{V}}^T\right]_{nm}, 0\right)\ . \tag{29}$$

4) Return to step 2.

In step 3, the algorithm computes a projection onto $\mathcal{S}$, followed by a projection onto $\mathbb{P}$, instead of computing the exact projection $\mathscr{P}_{\mathcal{S}^+}\{\mathbf{S}\}$ as given in (21). This significantly reduces the computational load, and it is observed to work fine in our simulations, since the negative values that appear after the projection onto $\mathcal{S}$ are observed to be very sparse. After several iterations of the algorithm, the number of negative values after the projection onto $\mathcal{S}$ becomes negligible. Notice that $\mathbf{S}$ is initialized with absolute values of the elements of $\mathbf{Y}$. The absolute value guarantees that the initial $\mathbf{S} \in \mathbb{P}$, which is required when using multiplicative updates. Furthermore, by initialising $\mathbf{S}$ with (positive) elements of $\mathbf{Y}$, the initial matrix $\mathbf{S}$ will be 'close' to the subspace $\mathcal{S}$. Notice that, if the mixing matrix $\mathbf{A}$ is non-negative, then $\mathbf{Y}$ is non-negative, and hence the initial matrix $\mathbf{S}$ starts in the solution space $\mathcal{S}^+$, defined by the constraints (11).

The M-NICA algorithm is a fixed-point type algorithm, which has the facilitating property that it does not depend on any user-defined stepsize parameter, as opposed to the NPCA algorithm described in section II. The algorithm searches for a good approximation of the solution of (9)-(11), i.e. a common fixed point of (24) and (29). Notice that many of the false zeros of (24) are eliminated since they are reset to a non-zero value due to (29) and therefore, they can again be updated by the multiplicative decorrelation process. In extensive simulations with different types of signals, the M-NICA algorithm was always observed to converge. This is stated here as an observation, since a formal proof is not available. Once the algorithm has converged, the mixing matrix[9] $\hat{\mathbf{A}}$ and the unmixing matrix $\mathbf{K}$ can be

[9]We add a hat to denote that $\hat{\mathbf{A}}$ is an estimate of the actual mixing matrix $\mathbf{A}$.

computed as

$$\hat{\mathbf{A}} = \mathbf{Y}\mathbf{S}^T \left(\mathbf{S}\mathbf{S}^T\right)^{-1} \tag{30}$$

$$\mathbf{K} = \mathbf{S}\overline{\mathbf{V}}\ \overline{\mathbf{\Sigma}}^{-1}\overline{\mathbf{U}}^T\ . \tag{31}$$

Notice that there always remains a permutation and scaling ambiguity between the columns of $\hat{\mathbf{A}}$ and the rows of $\mathbf{S}$.

Assuming that $M \gg N$, then the overall complexity of the M-NICA algorithm is $O(N^2M)$, which is the same as the NPCA algorithm.

IV. Sliding-window M-NICA

In this section, we describe an adaptive version of the M-NICA algorithm, which corresponds to a sliding window implementation of the batch mode version of M-NICA. The window slides over the observed signal $\mathbf{y}$, sample by sample. After each window shift, a new sample is added to the window, and an old sample is removed. A sample that enters the window is first unmixed with an unmixing matrix computed from the previous samples. After each window shift, a single iteration[10] of the batch mode M-NICA algorithm is performed on the samples that are currently in the window.

Let K denote the number of samples in the sliding window. We use Matlab notation to denote rows and columns, i.e. $[\mathbf{M}]_{i,:}$ and $[\mathbf{M}]_{:,j}$ respectively denote the i-th row and the j-th column of the matrix $\mathbf{M}$. The adaptive implementation of M-NICA is then given as follows. For notational convenience, we omit all universal quantifiers. Index n is always assumed to attain all values from 1 to N and index k is assumed to attain all values from 1 to K.

1) *Initialization*:
   a) $[\mathbf{W}_Y]_{:,k} \leftarrow \mathbf{y}[k]$
   b) $[\mathbf{W}_S]_{nk} \leftarrow |[\mathbf{W}_Y]_{nk}|$
   c) $\mathbf{K} \leftarrow \mathbf{W}_S\mathbf{W}_Y^{\dagger}$, where $\mathbf{W}_Y^{\dagger}$ denotes the pseudo-inverse of $\mathbf{W}_Y$.
   d) $i \leftarrow K-1$
2) *Window updates*:
   a) $c \leftarrow (i \mod K) + 1$
   b) $i \leftarrow i+1$
   c) $[\mathbf{W}_Y]_{:,c} \leftarrow \mathbf{y}[i]$

[10]To achieve faster convergence, multiple iterations can be performed after each window shift.

d) Replace $\mathbf{W}_Y$ by its best rank N approximation by means of the singular value decomposition (SVD), i.e.

$$\{\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}\} \leftarrow \text{SVD}\left(\mathbf{W}_Y\right) \tag{32}$$

$$\mathbf{W}_Y \leftarrow \overline{\mathbf{U}}\, \overline{\mathbf{\Sigma}}\, \overline{\mathbf{V}}^T \tag{33}$$

where $\overline{\mathbf{\Sigma}}$ is the $N \times N$ diagonal matrix containing the N largest singular values of $\mathbf{W}_Y$ on its diagonal, and where the corresponding left and right singular vectors are stored in the columns of $\overline{\mathbf{U}}$ and $\overline{\mathbf{V}}$ respectively.

e) $[\mathbf{W}_S]_{nk} \leftarrow \max\left([\mathbf{K}\mathbf{W}_Y]_{nk}, 0\right)$

3) *Decorrelation step*: compute (24) where $\mathbf{S}$ and $\mathbf{S}^{\text{temp}}$ are replaced by $\mathbf{W}_S$ and $\mathbf{W}_S^{\text{temp}}$, respectively.

4) *Computation of unmixing matrix*:

$$\mathbf{K} \leftarrow \mathbf{W}_S^{\text{temp}} \overline{\mathbf{V}}\, \overline{\mathbf{\Sigma}}^{-1} \overline{\mathbf{U}}^T$$

$$[\mathbf{K}]_{n,:} \leftarrow \frac{[\mathbf{K}]_{n,:}}{\| [\mathbf{K}]_{n,:} \|}$$

5) *Estimation of sample* $\mathbf{s}[i]$:

$$\hat{\mathbf{s}}[i] = \mathbf{K}\mathbf{y}[i]$$

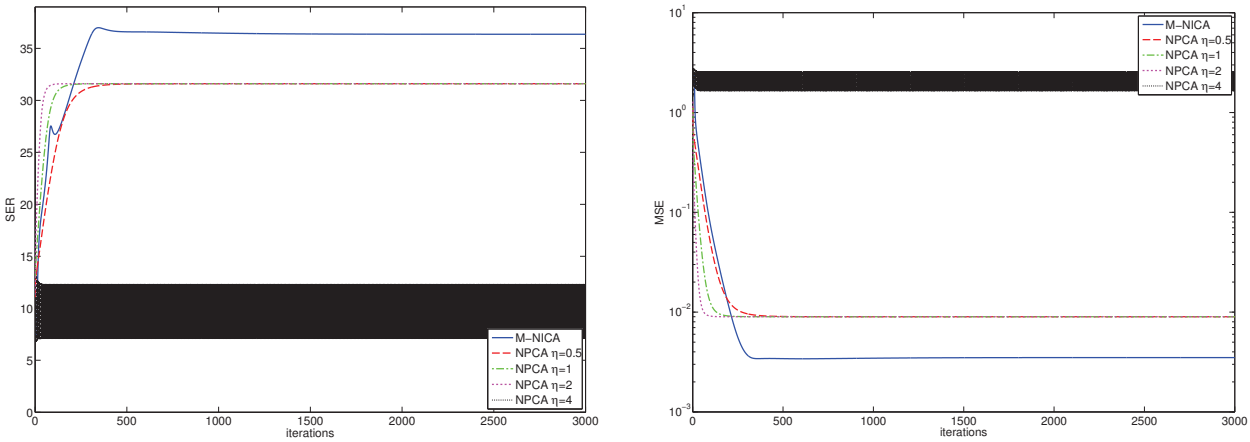
6) Return to step 2.

Notice that step 2(e) corresponds to the signal subspace projection step in the batch mode algorithm. Step (32) can be implemented efficiently by means of sliding window subspace tracking methods, e.g. [21], [22]. Also notice that the rows of the unmixing matrix $\mathbf{K}$ are normalized in each iteration to remove the scaling ambiguity in the NICA problem. Notice that $\mathbf{K}$ is normalized rather than $\mathbf{W}_S$, since a normalization of $\mathbf{W}_S$ would result in an unmixing matrix that varies over time when the signals in $\mathbf{s}$ are non-stationary, which is undesirable in view of the sample by sample unmixing in step 5 of the algorithm.

The window length K introduces a trade-off: it should be large enough such that the window contains enough samples to compute a reliable estimate of the correlation coefficients, and to make sure that the independence assumption is not violated. On the other hand, it should be small enough to achieve sufficient tracking performance.

V. Batch mode simulations

In this section, we provide batch mode simulation results for M-NICA and NPCA with different types of signals. We use two different measures to assess the performance of these algorithms: the signal-to-error



(a) SER (b) MSE

Fig. 1. SER and MSE for random signals that are uniformly distributed on the unit interval.

ratio (SER) and the mean squared error (MSE), i.e.

$$\text{SER} = \frac{1}{N}\sum_{n=1}^{N} 10\log_{10}\frac{E\{s_n^2\}}{E\{(s_n-\hat{s}_n)^2\}} \tag{34}$$

and

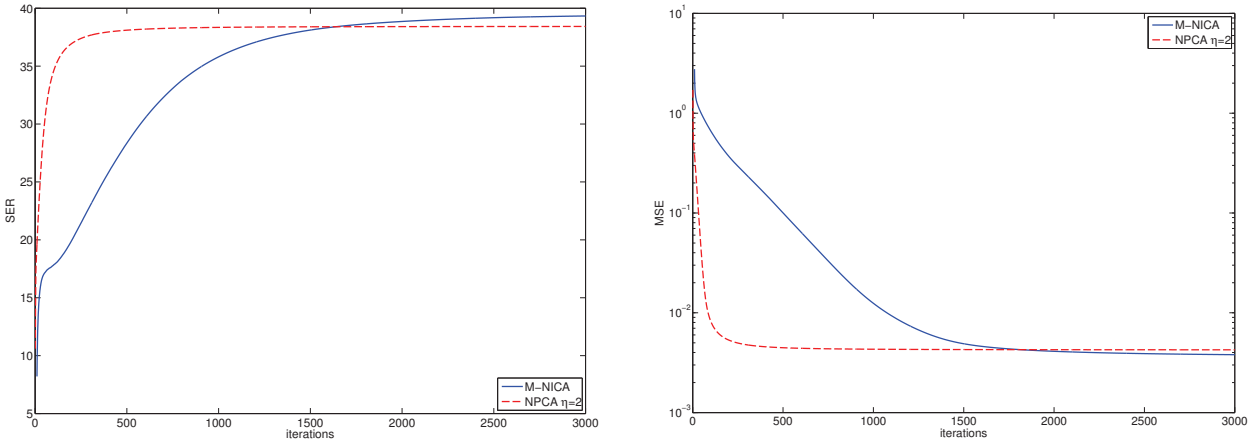
$$\text{MSE} = \frac{1}{N}\sum_{n=1}^{N} E\{(s_n-\hat{s}_n)^2\} \tag{35}$$

where $\hat{s}_n$ denotes the reconstruction of the n-th source signal, after an optimal (least squares) rescaling to resolve the scaling ambiguity between s_n and $\hat{s}_n$. Notice that NPCA does not explicitly enforce the unmixed signals to be non-negative, whereas M-NICA enforces this in (29). To obtain a fair comparison between both algorithms, we half-wave rectify the signals obtained by NPCA, i.e. negative values are set to zero.

A. Uniformly distributed random signals on the unit interval

In this experiment, we used a uniformly distributed random process on the unit interval to generate $M = 1000$ samples of the $N = 3$ source signals. The mixing matrix $\mathbf{A}$ is a 10×3 (J=10) matrix with random numbers drawn from a zero-mean normal distribution.

In Fig. 1(a) and 1(b) the SER and the MSE of both algorithms are plotted versus the number of iterations. It is observed that the convergence rate of NPCA depends on the choice of η. If η is set to a proper value, NPCA converges faster than M-NICA. However, if the value for η is too large, i.e. $\eta = 4$ in this case, NPCA does not converge and results in a suboptimal unmixing (in Fig. 1(a) and 1(b), this



(a) SER (b) MSE

Fig. 2. SER and MSE for random signals that are uniformly distributed on the unit interval, averaged over 1000 experiments.

results in a black band due to the oscillation of the SER and MSE over the different iterations). Despite the slower convergence, M-NICA has a higher unmixing accuracy.

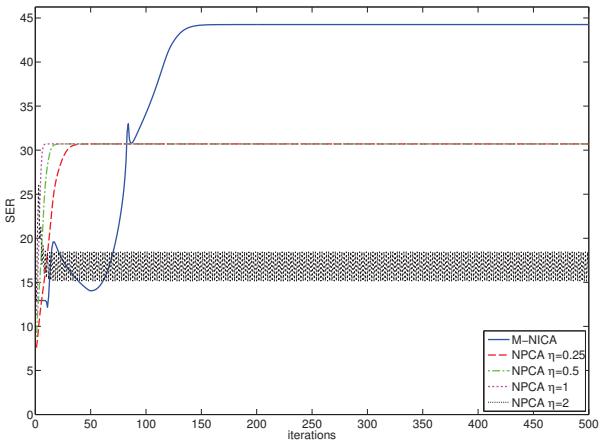
It should be noted that the convergence speed and the accuracy of the algorithms varies over different experiments. To draw more general conclusions, we performed 1000 Monte-Carlo simulations and averaged out the results. The learning rate for NPCA is set to $\eta = 2$, which is observed to provide the best results (both in terms of convergence and accuracy). The average SER and MSE versus the number of iterations are shown in Fig. 2(a) and 2(b). It is observed that NPCA generally converges much faster than M-NICA, but M-NICA slightly outperforms NPCA in terms of unmixing accuracy.

B. Sparse signals on the unit interval

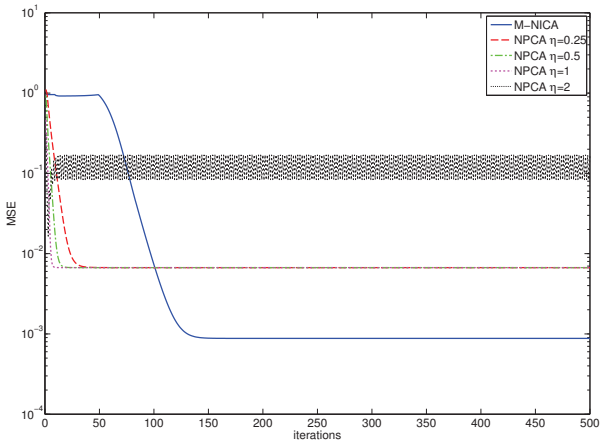
In this experiment, we model sparse random processes, i.e. $\exists\, \alpha > 0,\ \forall\, \delta > 0 :\ Pr(0 \leq s_n < \delta) > \alpha$. This model can be used when the sources have an on-off behaviour, or when analyzing signal spectra that are known to be sparse, e.g. [4], [8]. Notice that the well-grounded assumption is very well satisfied for this type of signals.

For the simulations, we use a signal that is similar to what we used in the previous section, but we modify it to model on-off behavior of the sources, i.e. the signal contains clusters of zero valued samples corresponding to the source being 'off' during a certain time segment[11]. To model this, the following random process is repeated for each of the $N = 3$ sources signals, until $M = 1000$ samples are generated

[11]For example, this is similar to the power of a speech signal analyzed in time, where pauses in between words and sentences create bursts of zeros [4].

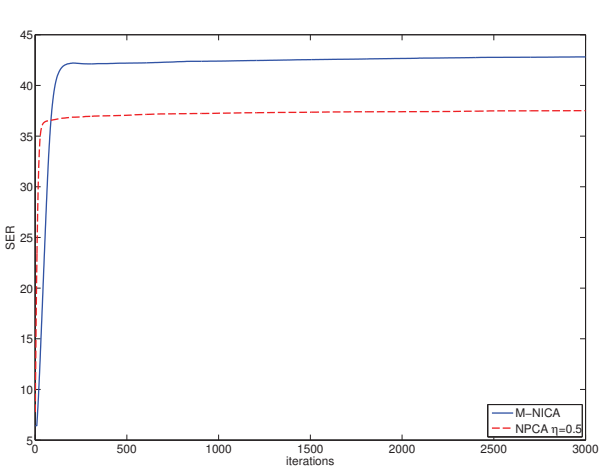


(a) SER

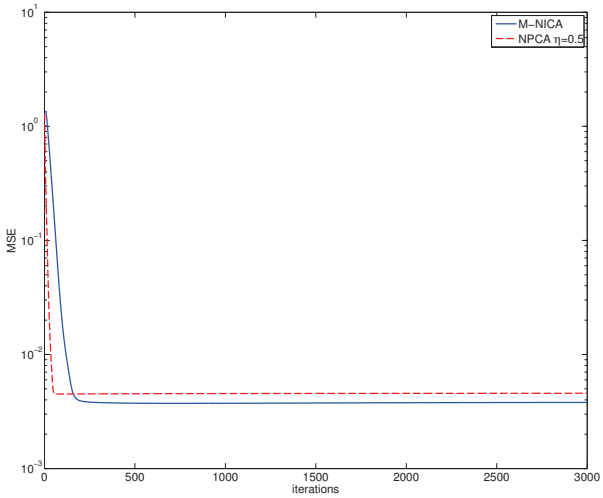


(b) MSE

Fig. 3. SER and MSE for random sparse signals on the unit interval.



(a) SER



(b) MSE

Fig. 4. SER and MSE for random sparse signals on the unit interval, averaged over 1000 experiments.

1) Let p define a binary random variable that can attain the values 0 or 1 with equal probability. Let q define an integer random variable that can attain values from 1 to 10 with equal probability.
2) Draw a sample P from p. If $P = 0$, go to step 3, and if $P = 1$, go to step 4.
3) Draw a sample Q from q. The next Q samples of the signal s are zero. Then go back to step 2.
4) Draw a sample Q from q. The next Q samples of the signal s are drawn from a uniformly distributed random process on the unit interval. Then go back to step 2.

Notice that the total time during which the source is switched off is approximately equal to the time during which the source is active. The mixing matrix is constructed as in the experiment described in

section V-A.

Fig. 3(a) and 3(b) plot the SER and MSE versus the number of iterations for both algorithms. It is observed that NPCA converges faster than M-NICA. However, M-NICA again yields a better unmixing accuracy. As opposed to the previous experiment, the learning rate of NPCA should now be set to a smaller value to obtain convergence.

To draw more general conclusions, we again performed 1000 Monte-Carlo simulations and averaged out the results. The learning rate of NPCA is set to $\eta = 0.5$. Larger values are observed to often cause NPCA not to converge. The average SER and MSE are shown in Fig. 4(a) and 4(b). Both algorithms converge much faster compared to the previous experiment (compare with Fig. 2(a) and 2(b)), which is due to the sparsity of the signal. It is again observed that NPCA converges fastest, but that M-NICA outperforms NPCA in terms of unmixing accuracy. The difference in unmixing accuracy between both algorithms appears to be more significant for sparse signals, i.e. more than 5 dB in SER (compare to Fig. 2).

C. Images

In this experiment, we generate 3 non-negative mixtures of 3 color images. Notice that the pixel values of images are non-negative, and therefore this defines a NICA problem. The original images and the unmixed images by M-NICA are shown in Fig. 5. Fig. 6 shows the SER versus the iteration index for M-NICA and NPCA. It is observed that M-NICA yields a significantly more accurate unmixing.

D. Effect of sample size

In the following Monte-Carlo experiment, we want to analyze the performance of M-NICA and NPCA for different amounts of available data samples M. Fig. 7(a) and Fig. 7(b) show the resulting SER for data generated as in section V-A (uniformly distributed signals) and section V-B (sparse signals) respectively. The results are averaged over 200 experiments. We performed 3000 iterations of M-NICA and NPCA with the uniformly distributed data, and 600 iterations with the sparse data since the latter yields faster convergence.

In Fig. 7(a), i.e. the case of uniformly distributed signals, it is observed that M-NICA outperforms NPCA if the amount of available samples is small. A possible reason for this is the fact that the samples of the original source signals are slightly correlated due to using finite sample sets. Since the decorrelation process of M-NICA is based on an explicit minization process that satisfies a non-negativity constraint, this correlation between the original samples will partly remain in the unmixed data. On the other hand, NPCA starts by perfectly decorrelating the data samples with a whitening matrix while ignoring this



Fig. 5. Three mixtures (first row) of the three original images (second row), and the corresponding unmixed images with the M-NICA algorithm (third row).

non-negativity constraint. This removes all correlation that was present in the original samples of the unmixed source signals, yielding an unavoidable distortion. If the amount of data samples is sufficiently large[12], NPCA has a similar (or better) unmixing accuracy compared to M-NICA. In Fig. 7(b), it is again observed that M-NICA outperforms NPCA, and that this effect is more significant when using small sample sizes. For $M = 100$, the relative difference in SER is approximately 20%, whereas this is

[12]For very large data sets (i.e. $M > 10000$), the results are not shown here since M-NICA needs more than 3000 iteration to converge in this case. This is not the case for sparse signals, as observed in Fig. 7(b), since M-NICA converges much faster on this type of data.

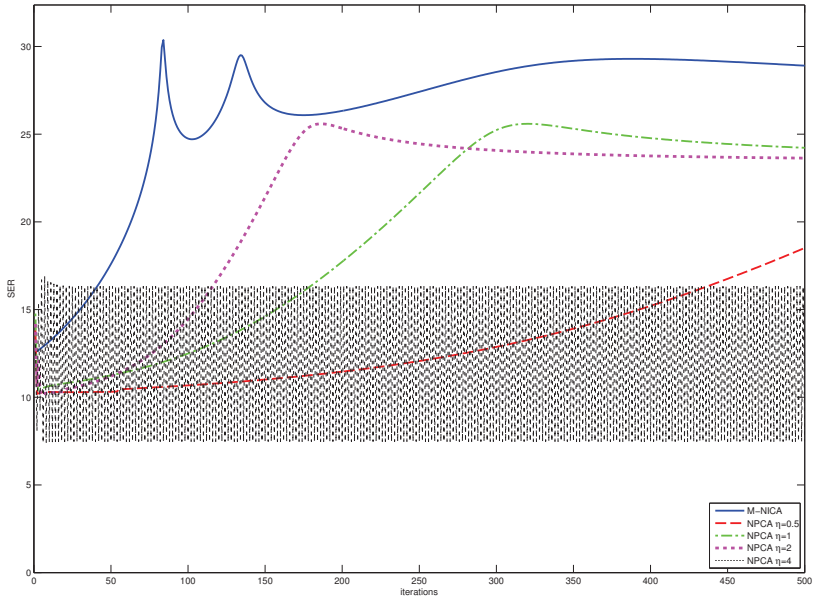
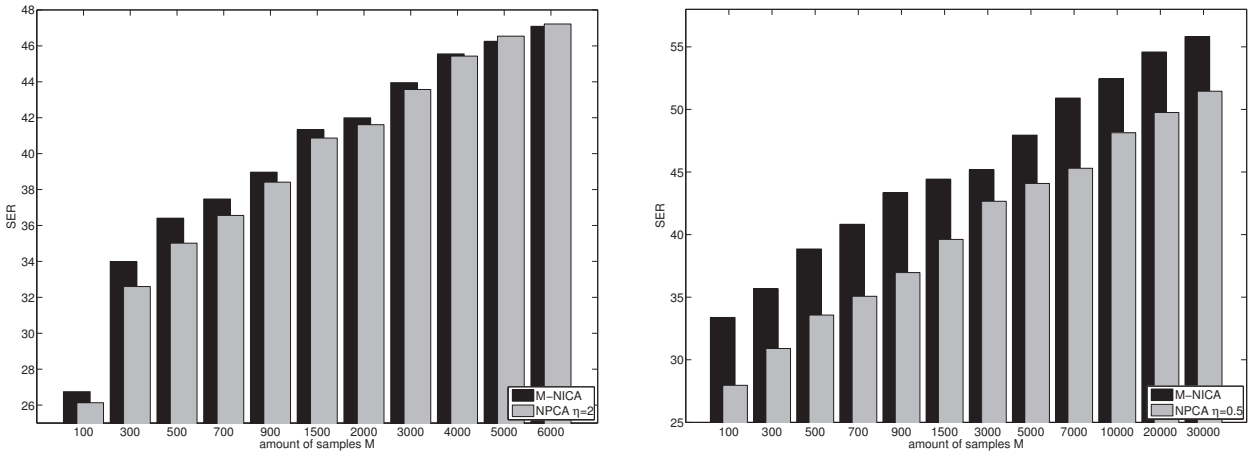


Fig. 6. SER versus iteration index for images.



(a) Uniformly distributed (b) Sparse

Fig. 7. SER, averaged over 200 experiments, as a function of sample size.

approximately 8% when $M = 30000$.

E. Conclusions

The above experiments demonstrate that the behavior of NPCA heavily depends on the choice of the learning rate η. The proper choice of η depends on the signals that are involved, and should be tuned by the user to ensure convergence and to obtain a good separation performance. The major advantage of the M-NICA algorithm is that it does not depend on any user-defined parameter. Furthermore, although

the M-NICA algorithm is usually slower than the NPCA algorithm, it generally yields a better separation performance than NPCA, especially when the amount of available data samples is small. In the case of sparse signals, M-NICA has good convergence properties and a significantly better unmixing accuracy than NPCA.

VI. Sliding window simulations

In this section, we provide simulation results of a sliding window implementation of M-NICA and NPCA with different types of signals. We use the same measures as in section V to assess the performance of the algorithms, i.e. the SER and the MSE. However, since we consider a sliding window implementation, both measures are computed over a window of length K and vary over time.

The sliding window implementation of M-NICA is described in section IV. We add $K-1$ zeros at the beginning of each signal, to be able to estimate each sample of $\mathbf{s}[i]$ starting from $i=0$. This means that the windows $\mathbf{W}_Y$ and $\mathbf{W}_S$ are initialized with $K-1$ all-zero columns.

The sliding window implementation of NPCA corresponds to its batch mode version described in section II, where now one iteration is performed for each position of the sliding window. This means that each time a new sample is added, the whitening matrix is updated according to (2), and the rotation matrix $\mathbf{W}$ is updated according to (3)-(5), where the expectation operator is replaced by an averaging over the samples in the window.

A. Uniformly distributed random signals on the unit interval

The signal and mixing matrix generation for this experiment is the same as in section V-A. However, to show the adaptation capabilities of the algorithms, we change the mixing matrix $\mathbf{A}$ after 1000 samples to another mixing matrix. A window length of $K=200$ seems to provide a good balance between adaptation speed and unmixing accuracy.

Fig. 8, shows the variation in SER, MSE and the cross-correlation between the estimated source signals, over time. The cross correlation is computed as the sum of the absolute values of the cross-correlation coefficients between the estimated sources signals. This is only shown for M-NICA since the cross-correlation is always zero in the case of NPCA, due to the whitening procedure. The drop in the SER, and the increase in the MSE and the cross-correlation at sample time 1000 is due to the sudden change of the mixing matrix $\mathbf{A}$. Again, it is observed that NPCA breaks down if the learning rate η is set too large. M-NICA provides the best unmixing accuracy.

To draw more general conclusions, we performed 1000 Monte-Carlo simulations of this experiment and averaged out the results. We set the learning rate of NPCA to $\eta=2$, which is observed to provide best

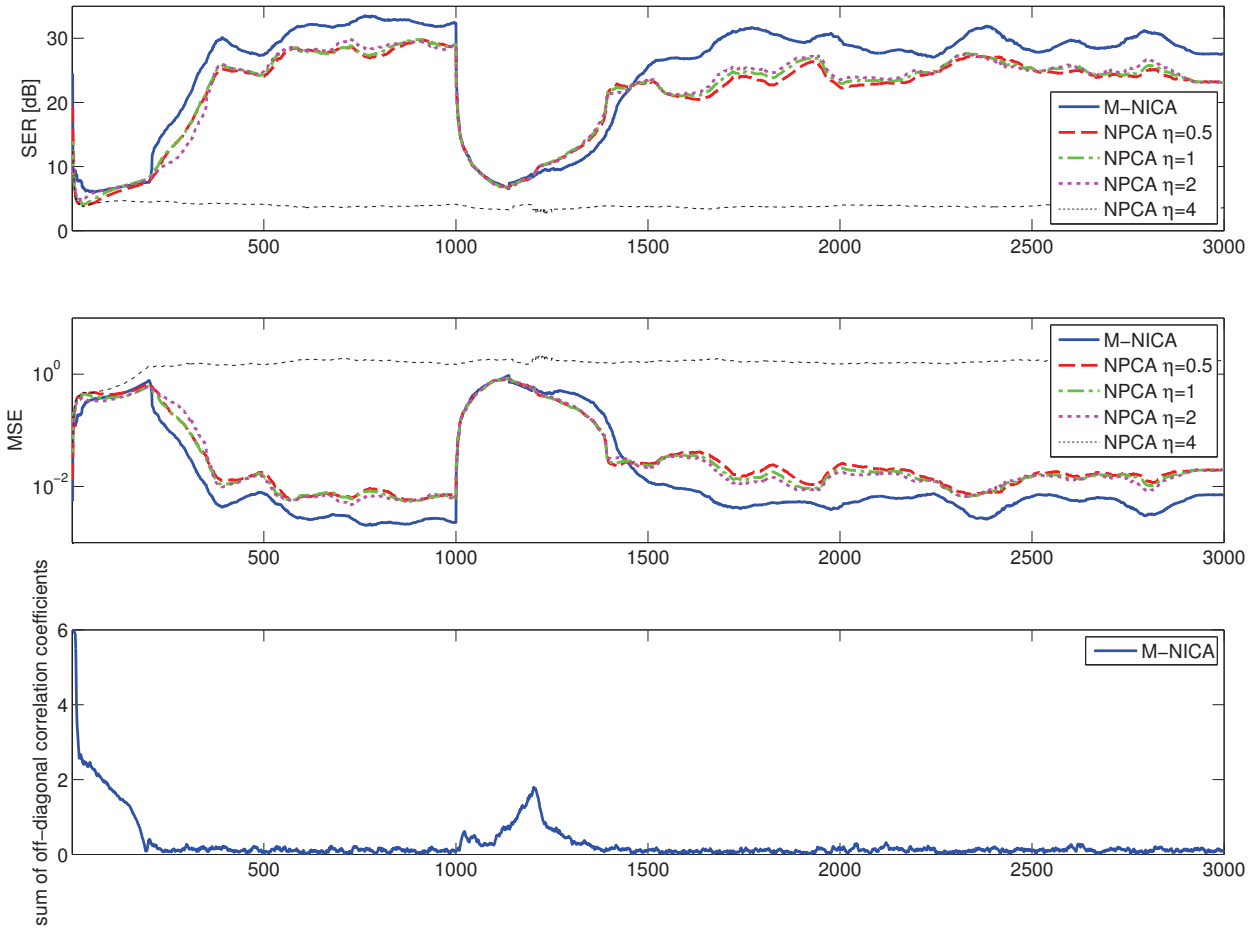


Fig. 8. The SER (above), MSE (middle) and the absolute value of the sum of the cross-correlation coefficients between the sources (below). All three measures are computed over the samples in the sliding window buffer.

results. The average SER and MSE over time is shown in Fig. 9. It is observed that, in general, M-NICA performs significantly better than NPCA in terms of unmixing accuracy, which may be explained by the fact that the window contains only a small amount of data samples. The difference in convergence speed between both algorithms is less distinct compared to the batch mode experiments (compare with Fig. 2).

B. Sparse signals on the unit interval

In this experiment, we analyze the performance of sliding window M-NICA and NPCA for sparse signals, generated in the same way as in section V-B. Again, we change the mixing matrix $\mathbf{A}$ after 1000 samples, and the window length is again set to $K = 200$.

Fig. 8 shows the variation in SER, MSE and the cross-correlation between the estimated source signals, over time. Again it is observed that M-NICA yields a better reconstruction of the source signal, compared to NPCA.

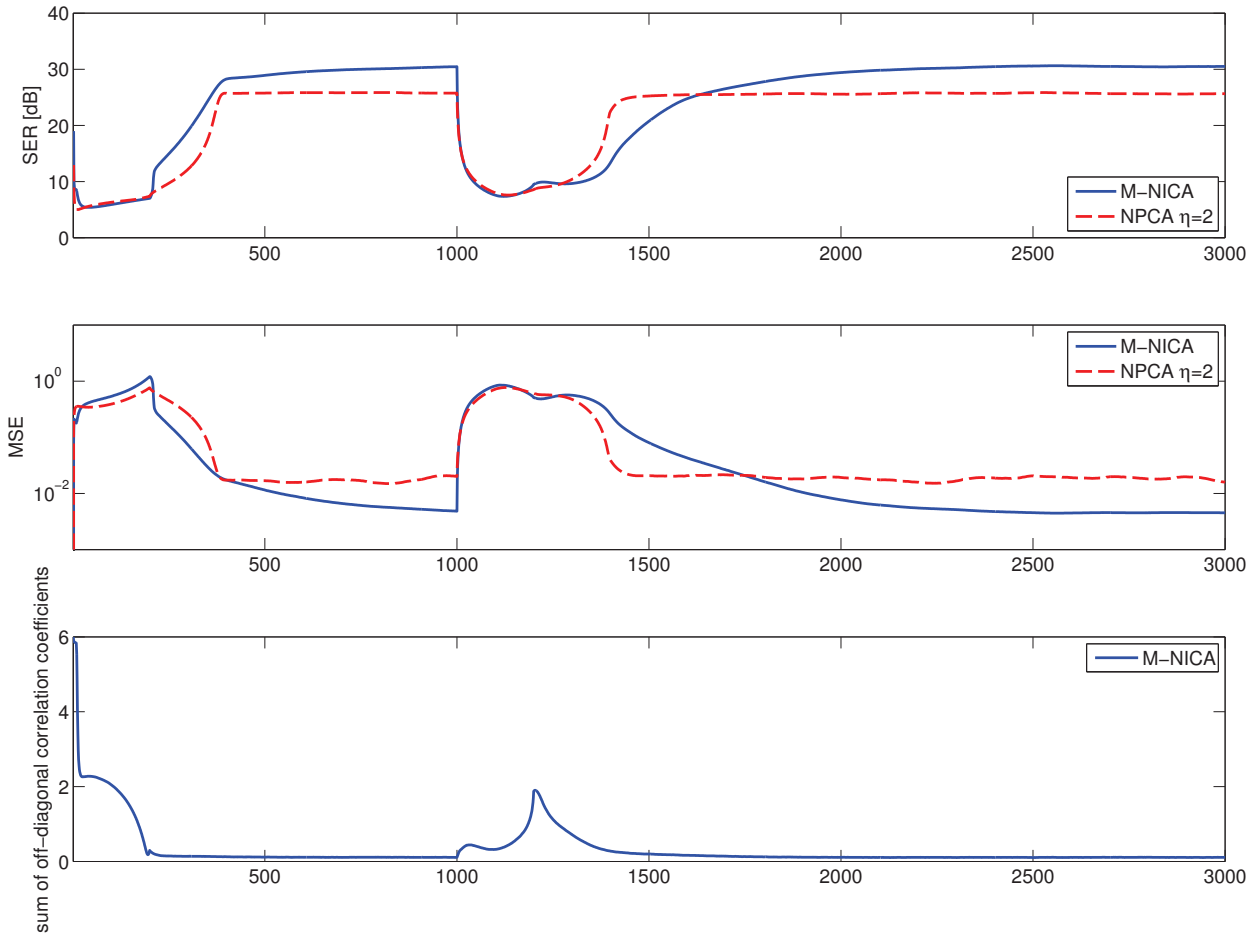


Fig. 9. The averaged SER (above), MSE (middle) and the absolute value of the sum of the cross-correlation coefficients between the sources (below). All three measures are computed over the samples in the sliding window buffer, and averaged over 1000 experiments.

The averaged results of 1000 Monte-Carlo simulations is shown in Fig. 11. The learning rate for NPCA is set to $\eta = 0.5$. Again it is observed that, in general, M-NICA performs significantly better than NPCA in terms of unmixing accuracy.

In [4], both sliding window algorithms are applied to track the power of multiple simultaneous speech signals. The results are consistent with the experiments in this paper, i.e. M-NICA significantly outperforms NPCA at the cost of a slightly slower adaptation speed.

VII. CONCLUSIONS

In this paper, we have proposed a new algorithm, referred to as M-NICA, to solve non-negative ICA problems with well-grounded sources. The M-NICA algorithm is based on multiplicative update rules which preserve non-negativity, together with a subspace projection based correction step. It has the

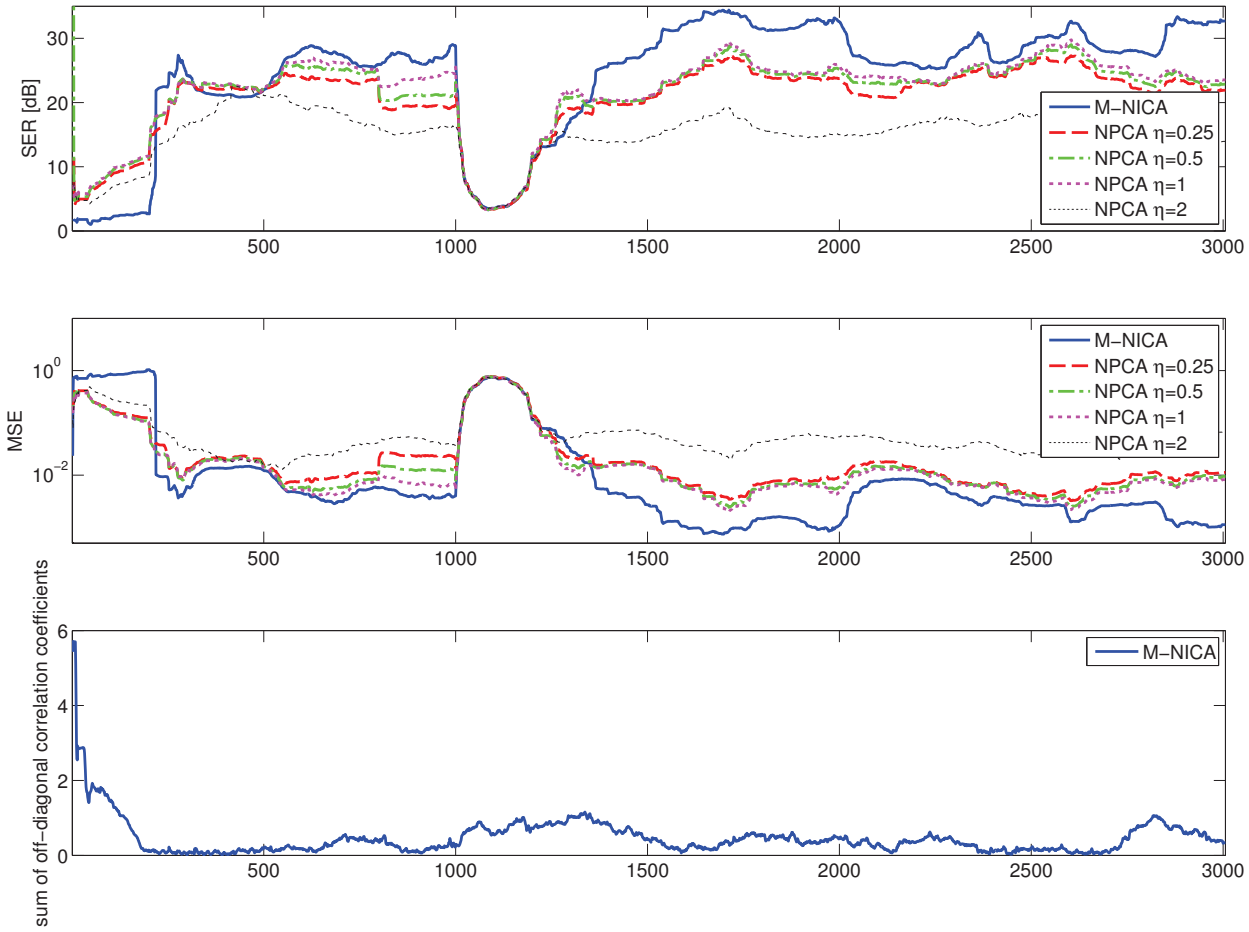


Fig. 10. The SER (above), MSE (middle) and the absolute value of the sum of the cross-correlation coefficients between the sources (below). All three measures are computed over the samples in the sliding window buffer.

facilitating property that it does not depend on a user-defined learning rate, as opposed to gradient based techniques such as the NPCA algorithm, where a proper choice for the learning rate is crucial to provide satisfying results.

The performance of M-NICA has been demonstrated by means of simulation results with different types of signals. Batch mode simulations indicated that M-NICA has a better unmixing accuracy than NPCA, but with slower convergence. In the case of sparse signals, M-NICA has good convergence properties, and significantly outperforms NPCA in terms of unmixing accuracy. It is also observed that M-NICA is best suited when the amount of available data samples is small. A sliding window implementation of both algorithms has also been described and validated, again showing that M-NICA significantly outperforms NPCA.

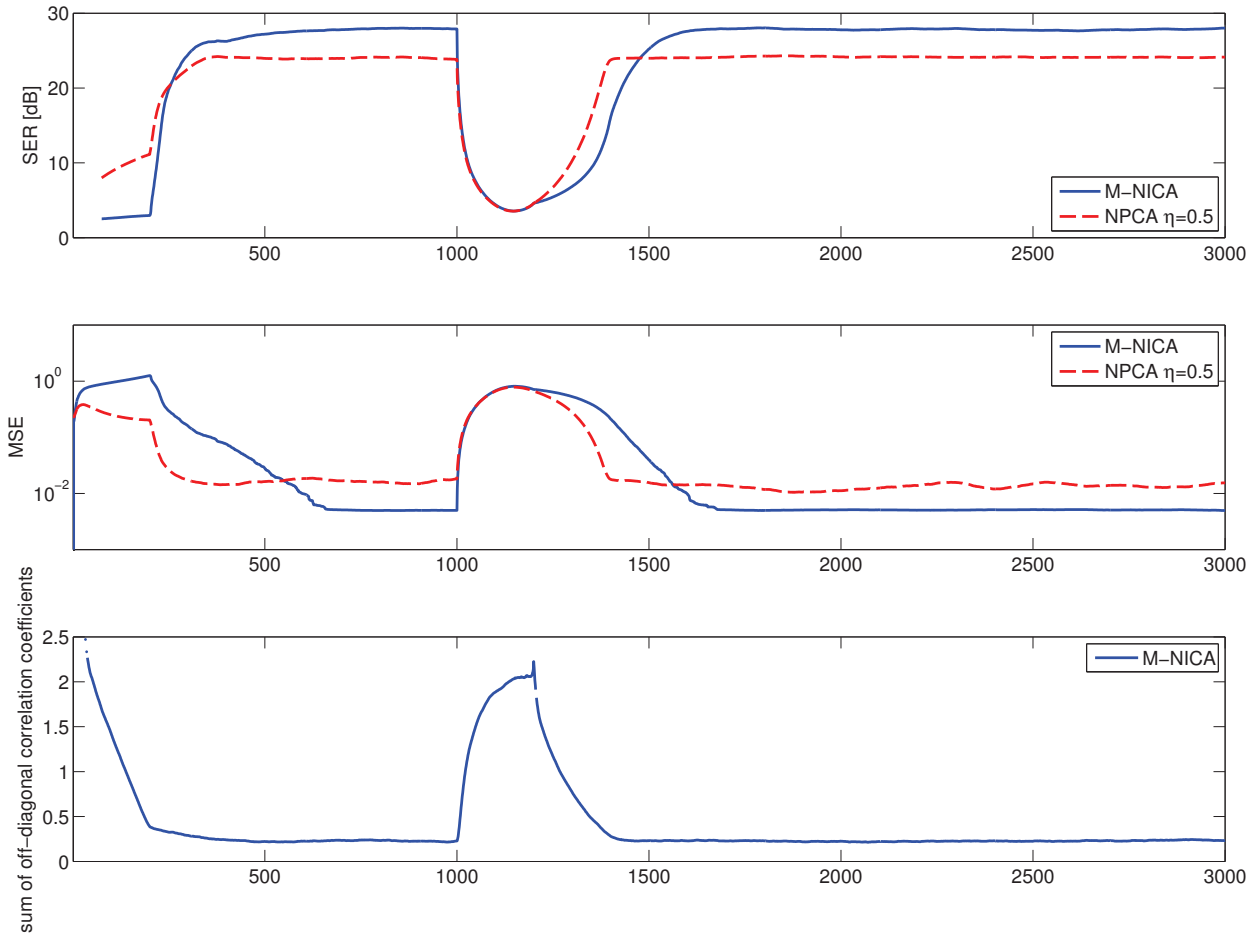


Fig. 11. The averaged SER (above), MSE (middle) and the absolute value of the sum of the cross-correlation coefficients between the sources (below). All three measures are computed over the samples in the sliding window buffer, and averaged over 1000 experiments.

VIII. Acknowledgements

Alexander Bertrand is a Research Assistant with the I.W.T. (Flemish Institute for the Promotion of Innovation through Science and Technology). This research work was carried out at the ESAT Laboratory of Katholieke Universiteit Leuven, in the frame of K.U.Leuven Research Council CoE EF/05/006 Optimization in Engineering (OPTEC), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, 'Dynamical systems, control and optimization', 2007-2011), Research Project FWO nr. G.0600.08 ('Signal processing and network design for wireless acoustic sensor networks'). The scientific responsibility is assumed by its authors. The authors would like to thank the anonymous reviewers for their useful comments and suggestions.

REFERENCES

[1] P. Comon and C. Jutten, *Handbook of Blind Source Separation*. New York: Academic Press, 2010.

[2] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*. New York: J. Wiley, 2002.

[3] E. Oja and M. Plumbley, "Blind separation of positive sources using non-negative PCA," in *Proc. of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, (Nara, Japan), April 2003.

[4] A. Bertrand and M. Moonen, "Energy-based multi-speaker voice activity detection with an ad hoc microphone array," in *Acoustics, Speech and Signal Processing, 2010. ICASSP 2010. IEEE International Conference on*, pp. 85–88, March 2010.

[5] P. Pauca, R. Plemmons, M. Giffin, and K. Hamada, "Unmixing spectral data for space objects using independent component analysis and nonnegative matrix factorization," in *Proceedings Amos Technical Conference*, 2004.

[6] J. Nascimento and J. Dias, "Does independent component analysis play a role in unmixing hyperspectral data?," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, pp. 175–187, January 2005.

[7] L. Parra, C. Spence, P. Sajda, A. Ziehe, and K.-R. Muller, "Unmixing hyperspectral data," in *in Advances in Neural Information Processing 12 (Proc. NIPS*99)*, pp. 942–948, MIT Press, 2000.

[8] D. Nuzillard and J.-M. Nuzillard, "Application of blind source separation to 1-D and 2-D nuclear magnetic resonance spectroscopy," *Signal Processing Letters, IEEE*, vol. 5, pp. 209–211, August 1998.

[9] R. C. Henry, "Multivariate receptor models–current practice and future trends," *Chemometrics and Intelligent Laboratory Systems*, vol. 60, no. 1-2, pp. 43 – 48, 2002.

[10] S. Abdallah and M. Plumbley, "Polyphonic transcription by non-negative sparse coding of power spectra," in *Proc. Int. Conf. Music Information Retrieval (ISMIR)*, (Barcelona, Spain), pp. 318–325, October 2004.

[11] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization.," *Nature*, vol. 401, pp. 788–791, October 1999.

[12] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *In NIPS*, pp. 556–562, MIT Press, 2000.

[13] F. Y. Wang, C. Y. Chi, T. H. Chan, and Y. Wang, "Blind separation of positive dependent sources by non-negative least-correlated component analysis," in *Machine Learning for Signal Processing, 2006. Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on*, pp. 73–78, September 2006.

[14] T.-H. Chan, W.-K. Ma, C.-Y. Chi, and Y. Wang, "Blind separation of non-negative sources by convex analysis: Effective method using linear programming," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 3493–3496, April 2008.

[15] Z. Yang and J. Laaksonen, "Multiplicative updates for non-negative projections," *Neurocomputing*, vol. 71, no. 1-3, pp. 363 – 373, 2007. Dedicated Hardware Architectures for Intelligent Systems; Advances on Neural Networks for Speech and Audio Processing.

[16] M. Plumbley, "Conditions for nonnegative independent component analysis," *Signal Processing Letters, IEEE*, vol. 9, pp. 177–180, June 2002.

[17] M. Ye, "Global convergence analysis of a discrete time nonnegative ICA algorithm," *Neural Networks, IEEE Transactions on*, vol. 17, pp. 253–256, January 2006.

[18] S. Amari and S. Douglas, "Why natural gradient?," in *Acoustics, Speech and Signal Processing, 1998. ICASSP 1998. IEEE International Conference on*, vol. 2, pp. 1213–1216, May 1998.

[19] J. Boyle and R. Dykstra, "A method for finding projections onto the intersection of convex sets in Hilbert spaces," *Lecture Notes in Statistics*, vol. 37, pp. 28–47, 1986.

[20] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 ed., 2006.

[21] R. Badeau, G. Richard, and B. David, "Sliding window adaptive SVD algorithms," *Signal Processing, IEEE Transactions on*, vol. 52, January 2004.

[22] P. Strobach, "Sliding window adaptive SVD using the unsymmetric householder partial compressor," *Signal Processing*, 2009, DOI: 10.1016/j.sigpro.2009.07.002.

View publication stats