BLIND SOURCE SEPARATION ALGORITHM FOR MIMO CONVOLUTIVE MIXTURES

Kamran Rahbar and James P. Reilly

Electrical & Computer Eng.
Mcmaster University, Hamilton, Ontario, Canada
Email: kamran@reverb.crl.mcmaster.ca, reillyj@mcmaster.ca

ABSTRACT

We consider the problem of blind source separation of MIMO convolutive mixtures for the general case where the number of sensors are greater than or equal to the number of sources. We assume that sources are non-stationary signals. The separation is performed in the frequency domain by joint minimization of the off–diagonal elements of observed signal's cross-spectral density matrices over different epochs. We propose an efficient Newton– based algorithm over the complex Steifel manifold to minimize an appropriate cost function. We resolve the permutation problem using a novel tree structured diadic detection scheme. We find and correct wrong permutations at each frequency bin based on cross frequency correlation between diagonal elements of the output cross spectral matrices. We demonstrate the performance of the new algorithm using synthetic mixtures and real word recordings. The method has the additional advantage of fast convergence.

1. INTRODUCTION

Blind source separation deals with the case where statistically independent sources are mixed through an unknown channel where only the channel outputs (observed signals) are measurable. The objective is: based on the information contained in observed signals design a separation network to extract the original sources. In this paper we consider only the FIR convolutive mixture case. There have been several algorithms in the literature which discuss this problem: [1], [2], [3]. The main approaches taken so far can be divided in two groups. Higher order statistics methods are discussed in [1], [3], [4] while [2] and [5] propose second–order statistics algorithms. We can also divide the proposed methods into time domain [2], [3] and frequency domain [4] [5] approaches. In this paper we propose an algorithm in the frequency domain using a second–order statistics approach. The key assumption used in this paper is that the sources are non-stationary. We consider the general case when the number of observed signals can be equal to or greater than the number of the sources. We estimate the separating matrix at each frequency bin by minimizing the off–diagonal elements of the cross spectral density matrices of the observed signals over a range of epochs. Using the information contained in diagonal elements of the diagonalized output covariance matrices we then align the permutations of the columns of the separating matrices across frequency. Similar work is done in [6] based on cross correlation between temporal trajectories of the separated output signals. The differences are: here we don't need to estimate these temporal trajectories; also, we propose a diadic sorting scheme for adjusting the permutations. In this way we can prevent catastrophic errors (as will be explained later in this paper) that can happen as a result of a missed or wrongly adjusted permutation. To support our arguments and to demonstrate the performance of the algorithm we present numerical simulations using synthetic sources and real speech data.

2. PROBLEM STATEMENT

We consider the following N-source J-sensor MIMO linear model for the received signal for the convolutive mixing problem:

$$\mathbf{x}(k) = \sum_{l=-\infty}^{\infty} \mathbf{H}(l)\mathbf{s}(k-l) + \mathbf{n}(k) \qquad k \in \mathcal{Z} \tag{1}$$

where $\mathbf{x}(k) = (x_1(k), \cdots, x_J(k))^T$ is the vector of observed signals, $\mathbf{s}(k) = (s_1(k), \cdots, s_N(k))^T$ is the vector of sources, $\mathbf{H}(k)$ is the $J \times N$ channel matrix and $\mathbf{n}(k) = (n_1(k), \cdots, n_J(k))^T$ is the additive noise vector. We make the following assumptions on the model:

A0: $J \geq N$; i.e, we can have more sensors than sources.

A1: The sources $\mathbf{s}(k)$ are real, zero mean, non-stationary random signals and the cross–spectral density matrices of the sources, $\mathbf{D}_s(\omega, m)$ are diagonal for all ω and m, where ω denotes frequency and m denotes epoch.

A2: $\mathbf{H}(k)$ is real, causal and has finite length L_h with L_h being unknown. We also assume that $\mathbf{H}(\omega)$ has full column rank for all $\omega \in [0, 2\pi)$.

A3: The noise $\mathbf{n}(k)$ is zero mean, *iid* across sensors, with unknown power and is independent of the sources.

The objective is to design a causal, stable MIMO separation network $\mathbf{B}(l)$ of length L_B with outputs given as:

$$\mathbf{y}(k) = \sum_{l=0}^{L_B} \mathbf{B}(l)\mathbf{x}(k-l), \tag{2}$$

This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC), Centre for Information Technology Ontario (CITO)

such that for each output $y_i(k)$ we have:

$$y_i(k) = \lambda_{ii}(k) * s_{\pi(i)}(k) + \zeta(k) \qquad i = 1, \cdots, N, \quad (3)$$

where $\pi(i) \in \{1, \cdots, N\}$, is a permutation function, $*$ is the convolution operator, $\lambda_{ii}(k)$ is the impulse response of the filtering operation done on $s_{\pi(i)}(k)$, and $\zeta(k)$ represents the residual noise. In the frequency domain, the separation network must satisfy

$$\mathbf{B}(\omega)\mathbf{H}(\omega) = \mathbf{\Lambda}(\omega)\mathbf{\Pi}(\omega) \quad (4)$$

where $\mathbf{\Lambda}(\omega)$ is a diagonal matrix with diagonal elements $\lambda_{ii}(\omega)$ and $\mathbf{\Pi}(\omega)$ is a permutation matrix. Notice that equation (4) although a necessary condition, is not a sufficient condition for separation unless we have $\mathbf{\Pi}(\omega) = \mathbf{\Pi}_c$ where $\mathbf{\Pi}_c$ is a constant permutation matrix. In next sections we propose a frequency domain separation criteria and a method for obtaining uniform permutations across all frequency bins.

3. PREWHITENING AND JOINT DIAGONALIZATION CRITERIA

The separation criteria is based on minimizing the off–diagonal elements of cross–spectral density matrices of the observed signals over M epochs. The signal is assumed to be stationary over each epoch. The cross–spectral density matrix of the observed signals $\mathbf{x}(k)$ at epoch m is given by

$$\tilde{\mathbf{P}}_x(\omega, m) = \mathbf{H}(\omega)\mathbf{D}_s(\omega, m)\mathbf{H}^\dagger(\omega) + \sigma_n^2\mathbf{I}, \quad (5)$$

where $\dagger$ represents the Hermitian transpose operation, $\mathbf{D}_s(\omega, m)$ is the cross–spectral density matrix of the sources, which by assumption is diagonal for all $\omega \in [0, 2\pi)$, $m \in \{0, \cdots, M-1\}$ and σ_n^2 is the power of the noise $\mathbf{n}(t)$. For $J > N$ and under high signal to noise ratios, σ_n^2 can be estimated from the $J - N$ smallest eigenvalues of $\mathbf{P}_x(\omega, m)$ and we can set:

$$\mathbf{P}_x(\omega, m) = \tilde{\mathbf{P}}_x(\omega, m) - \hat{\sigma}_n^2\mathbf{I}. \quad (6)$$

We can estimate $\mathbf{B}(\omega)$ up to a paraunitary filter by whitening the observed signals. That is, we can write

$$\mathbf{B}(\omega) = \mathbf{\Phi}(\omega)\tilde{\mathbf{W}}(\omega) \quad (7)$$

where $\mathbf{\Phi}(\omega)$ is a $N \times N$ paraunitary matrix and $\tilde{\mathbf{W}}(\omega)$ is the $N \times J$ polynomial whitening filter, which can be estimated from:

$$\tilde{\mathbf{W}}(\omega) = \mathbf{\Sigma}(\omega)^{-\frac{1}{2}}\mathbf{V}^\dagger(\omega) \quad (8)$$

where $\mathbf{\Sigma}(\omega)$ is an $N \times N$ diagonal matrix whose N diagonal values correspond to the N largest eigenvalues of

$$\mathbf{R}(\omega) = \sum_{m=0}^{M-1} \mathbf{P}_x(\omega, m), \quad (9)$$

and $\mathbf{V}(\omega)$ is an $J \times N$ matrix with orthonormal columns consisting of the N corresponding eigenvectors. Notice that $\tilde{\mathbf{W}}(\omega)$ given by (8) spatially and temporarily whitens the observed signals. To prevent temporal whitening of the outputs when the sources are colored we use the following ad-hoc but convenient method for normalizing the $\tilde{\mathbf{W}}(\omega)$ given by (8) as follows:

$$\mathbf{W}(\omega) = \frac{\tilde{\mathbf{W}}(\omega)}{||\tilde{\mathbf{W}}(\omega)||_F}. \quad (10)$$

Applying $\mathbf{W}(\omega)$ to $\mathbf{P}_x(\omega, m)$ we can write:

$$\mathbf{P}_w(\omega, m) \triangleq \mathbf{W}(\omega)\mathbf{P}_x(\omega, m)\mathbf{W}^\dagger(\omega). \quad (11)$$

The paraunitary matrix $\mathbf{\Phi}(\omega)$ can be estimated from the orthogonal joint diagonalization of $\mathbf{P}_w(\omega, m)$ over all $m \in \{0, \cdots, M-1\}$. Since in practice we have only an estimate of $\mathbf{P}_w(\omega, m)$ this diagonalization will be approximate and can be achieved by jointly minimizing the off-diagonal elements of $\mathbf{P}_w(\omega, m)$ over the range of epochs $m = 0, \cdots, M-1$.

A batch algorithm for approximate joint diagonalization has been discussed in [7]. In this paper, we present an an alternative algorithm which uses optimization over the complex Steifel manifold for this purpose, that has the advantage of being *adaptive*. Since the Frobineous norm of $\mathbf{P}_w(\omega, m)$ is invariant to the orthogonal transformation $\mathbf{\Phi}(\omega)$, minimizing the sum of squares of off-diagonal elements is equivalent to maximizing the sum of squares of the diagonal elements. The joint diagonalization procedure can therefore be reformatted and written as following optimization problem for each $\omega_k \in [0, \pi)$ [8]:

$$\begin{array}{ll} \min\limits_{\mathbf{\Phi}(\omega_k)} & -\frac{1}{2}\sum_{m=0}^{M-1}\sum_{i \neq j=1}^{N}(a_{ii}(\omega_k, m) - a_{jj}(\omega_k, m))^2 \\ \text{Subject to} & \mathbf{\Phi}(\omega_k)\mathbf{\Phi}^\dagger(\omega_k) = \mathbf{I} \qquad k = 0, \cdots, K, \end{array} \quad (12)$$

where $a_{ii}(\omega_k, m)$ represents the i_{th} diagonal value of $\mathbf{\Phi}(\omega_k)\mathbf{P}_w(\omega_k, m)\mathbf{\Phi}(\omega_k)^\dagger$ and K is the number of frequency points of interest.

In [8] a method was proposed for solving a problem similar to (12), for the case of real orthogonal matrices. As indicated in that paper, the orthogonality constraint can be represented over the Steifel manifold and an optimization problem with orthogonality constraints can be solved using unconstrained optimization methods over this manifold. In this paper we extend the method in [8] to complex matrices. A rigorous treatment of the complex Steifel manifold is done in [9]. In this paper we use a Newton method over the complex Steifel manifold to optimize (12).

4. NEWTON METHOD OVER COMPLEX STEIFEL MANIFOLD

In this section we derive gradient descent and Newton methods over the complex Steifel manifold to solve the constrained optimization problem given in (12). The Newton method has the advantage of fast convergence when close to optimal solution while gradient descent has the advantage of lower complexity and tracking ability. In and [9] [10] the authors discuss general forms for unconstrained optimization methods over Steifel manifold. Following the ideas in [8], we derive adaptive algorithms for the complex version of the orthogonal joint diagonalization problem. For brevity we limit ourselves to the necessary derivations and final results. For more information we refer interested readers to the above references.

The update rule for minimizing the function given in (12) over the complex Steifel manifold is given by:

$$\mathbf{\Phi}_t(\omega_k) = \mathbf{\Phi}_{t-1}(\omega_k) exp\Bigg[\alpha \mathbf{\Phi}^{\dagger}_{t-1}(\omega_k)\mathbf{\Gamma}_{t-1}(\omega_k)\Bigg] \tag{13}$$

where α is the step size and $\mathbf{\Gamma}_{t-1}(\omega_k)$ is the search direction at the iteration $t-1$ and at the frequency bin ω_k. For gradient descent the search direction is $\mathbf{\Gamma}(\omega_k) = -\mathbf{G}(\omega_k)$ where $\mathbf{G}(\omega_k)$ is the gradient of the cost function in (12) over the complex Steifel manifold given by following equation:

$$\mathbf{G}(\omega_k) = \mathbf{F}_{\Phi}(\omega_k) - \mathbf{\Phi}(\omega_k)\mathbf{F}^{\dagger}_{\Phi}(\omega_k)\mathbf{\Phi}(\omega_k) \tag{14}$$

where $\mathbf{F}_{\Phi}(\omega_k)$ is matrix of partial derivatives of the cost function in (12) with respect to $\varphi_{ij}(\omega_k)$ which is the ij_{th} element of $\mathbf{\Phi}(\omega_k)$.

$$\mathbf{F}_{\Phi}(\omega_k) = -\sum_{m=0}^{M-1} \frac{\partial a_{ii}(\omega_k, m)}{\partial \varphi_{ij}(\omega_k)} \sum_{i \neq j} \big(a_{ii}(\omega_k, m) - a_{jj}(\omega_k, m)\big), \tag{15}$$

which can be calculated as:

$$\mathbf{F}_{\Phi}(\omega_k) = -2\sum_{m=0}^{J-1} \mathbf{\Lambda}(\omega_k, m)\mathbf{\Phi}(\omega_k)\ bP_w(\omega_k, m) \tag{16}$$

where

$$\mathbf{\Lambda}(\omega_k, m) = N\mathbf{D}_y(\omega_k, m) - c(\omega_k, m)\mathbf{I}, \tag{17}$$

$$\mathbf{D}_y(\omega_k, m) = diag(\mathbf{\Phi}(\omega_k)\mathbf{P}_w(\omega_k, m)\mathbf{\Phi}^{\dagger}(\omega_k)) \tag{18}$$

and $c(\omega_k, m)$ is the trace of $\mathbf{P}_w(\omega_k, m)$ for each ω_k and m.

For the Newton method over the complex Steifel manifold we use the method suggested in [9], which requires calculating the first and second derivative (Hessian) of the cost function given in (12). To calculate the Hessian we first rewrite (12) as:

$$\min_{\mathbf{\Phi}(\omega_k)} -\frac{1}{2}\sum_{m=0}^{M-1}\sum_{i \neq j=1}^{N}\Bigg(\mathrm{vec}\{\mathbf{\Phi}(\omega_k)\}^{\dagger}\mathbf{A}^{ij}(\omega_k, m)\mathrm{vec}\{\mathbf{\Phi}(\omega_k)\}\Bigg)^2$$
$$\text{Subject to} \quad \mathbf{\Phi}(\omega_k)\mathbf{\Phi}^{\dagger}(\omega_k) = \mathbf{I} \qquad k = 0, \cdots, K \tag{19}$$

where

$$\mathbf{A}^{ij}(\omega_k, m) = \mathbf{P}^{*}_w(\omega_k, m) \otimes \mathbf{E}^{ij}, \tag{20}$$

the symbol * represents the complex conjugate operation, and $\mathbf{E}^{ij}$ is an $N \times N$ diagonal matrix whose i_{th} diagonal element is equal to 1, whose j_{th} diagonal element is equal to -1, and all other elements are equal to zero. The Hessian of (19) is easily found to be:

$$\begin{aligned}\mathbf{H}_{\Phi}(\omega_k, m) = -\sum_{m=0}^{M-1}\sum_{i \neq j=1}^{N}\Bigg\{ \\ 4\mathbf{A}^{ij}(\omega_k, m)\mathrm{vec}\{\mathbf{\Phi}(\omega_k)\}\mathrm{vec}\{\mathbf{\Phi}(\omega_k)\}^{\dagger}\mathbf{A}^{ij}(\omega_k, m) \\ + 2\mathrm{vec}\{\mathbf{\Phi}(\omega_k)\}^{\dagger}\mathbf{A}^{ij}(\omega_k, m)\mathrm{vec}\{\mathbf{\Phi}(\omega_k)\}\mathbf{A}^{ij}(\omega_k, m)\Bigg\}\end{aligned} \tag{21}$$

Using the first and second order derivatives $\mathbf{F}_{\Phi}(\omega_k)$ and $\mathbf{H}_{\Phi}(\omega_k)$, we calculate the Newton step over the complex Steifel manifold using the algorithm given in [9]. We can then use the obtained Newton step to update the $\mathbf{\Phi}(\omega_k)$ using (13) with $\alpha = 1$.

A procedure which has been found to work well empirically, from the viewpoint of computational efficiency and fast convergence, is to first, use a few gradient descent steps to get close to the solution, and then apply the Newton algorithm to quickly converge to the optimum.

Joint diagonalization Using the Newton method over the complex Steifel manifold

1. Initialize $\mathbf{\Phi}(\omega_k)$ to some arbitrary value for all $k = 1, \cdots, K$ such that $\mathbf{\Phi}(\omega_k)\mathbf{\Phi}(\omega_k)^{\dagger} = \mathbf{I}$
2. Compute $\mathbf{F}_{\Phi}(\omega_k)$ and $\mathbf{H}_{\Phi}(\omega_k)$, the first and second order derivatives of cost function (12), as given by equations (16) and (21).
3. Using the derivative information calculate the search direction $\mathbf{\Gamma}(\omega_k)$ using the function *tpoint* as defined in Table 1 in [9]
4. Update the values of $\mathbf{\Phi}(\omega_k)$ using the (13)
5. Test for convergence. If the test fails, go to step 2.

5. RESOLVING PERMUTATIONS

One potential problem with the cost function in (12) is that it is insensitive to permutations of columns of $\mathbf{\Phi}(\omega_k)$; i.e., if $\mathbf{\Phi}_{opt}(\omega_k)$ is an optimum solution to (12) then $\mathbf{\Pi}(\omega_k)\mathbf{\Phi}_{opt}(\omega_k)$, where $\mathbf{\Pi}(\omega_k)$ is a permutation matrix, will also be a optimum solution. Since in general $\mathbf{\Pi}(\omega_k)$ will vary with frequency, poor overall separation performance can result. One solution to this problem, as has been suggested by [5], is to constrain the length of the separation filters. Adding such a constraint, as has been mentioned in [11], limits the the overall separation performance, specially for long mixing filters. In this paper we suggest a novel solution for solving the permutation problem which exploits the cross-frequency correlation between diagonal values of $\mathbf{D}_y(\omega_k, m)$ given by (18). $\mathbf{D}_y(\omega_k, m)$ can be considered an estimate of the sources' cross-power spectral density at epoch m. We can show that for white non-stationary signals the temporal trajectories of spectral density are correlated over frequency. For non-white signals, this correlation still holds, but for some frequencies it may be zero if the power spectrum of the sources at those frequencies vanish.

Using this correlation we can adjust the permutations as shown in the following example. Basically assume that we have two sources and $\mathbf{D}_y(\omega_k, m), m = 0, \cdots, M-1$ represents the estimated cross-spectral density matrix of the sources at frequency bin ω_k. We want to adjust the permutation at frequency ω_j such that it has the same permutation as in frequency bin ω_i. To do so we first calculate the cross frequency correlation between all diagonal values of $\mathbf{D}_y(\omega_j, m)$ and $\mathbf{D}_y(\omega_i, m)$ using following measure:

$$\rho_{qp}(\omega_i, \omega_j) = \frac{\sum_{m=0}^{M-1} d_q(\omega_i, m) d_p(\omega_j, m)}{\sqrt{\sum_{m=0}^{M-1} d_q^2(\omega_i, m)}\sqrt{\sum_{m=0}^{M-1} d_p^2(\omega_j, m)}} \quad (22)$$

where $\rho_{qp}(\omega_i, \omega_j)$ represents the cross frequency correlation between $d_q(\omega_i, m)$, the q_{th} diagonal element of $\mathbf{D}_y(\omega_i, m)$, and $d_p(\omega_j, m)$, the p_{th} diagonal element of $\mathbf{D}_y(\omega_j, m)$. If the frequency bins ω_i and ω_j have the same permutation then we expect that

$$\frac{\rho_{11}(\omega_i, \omega_j) + \rho_{22}(\omega_i, \omega_j)}{\rho_{12}(\omega_i, \omega_j) + \rho_{21}(\omega_i, \omega_j)} > 1; \quad (23)$$

otherwise, we must change the permutation at one of the frequency bins ω_i or ω_j such that the above condition is satisfied. We can apply the above ratio test to all frequency bins to detect and adjust the wrong permutations. In general when the number of sources is greater than two, the ratio test given in (23) can be written as the following discrete optimization problem

$$\max_{\mathbf{\Pi}_{ij} \in \mathcal{P}} \quad \text{trace}\big(\mathbf{E}^T(\omega_i)\mathbf{E}(\omega_j)\mathbf{\Pi}_{ij}\big) \quad (24)$$

where $\mathcal{P}$ is the set of $N \times N$ permutation matrices including the identity matrix, and $\mathbf{E}(\omega_k)$ is an $M \times N$ matrix whose l_{th} column is given by

$$\mathbf{e}_l(\omega_k) = (d_l(\omega_k, 0), \cdots, d_l(\omega_k, M-1))^T \quad (25)$$

where $\mathbf{e}_l(\omega_k)$ has been normalized to have unit norm; i.e, $\mathbf{e}_l(\omega_k) = \frac{\mathbf{e}_l(\omega_k)}{||\mathbf{e}_l(\omega_k)||_2}$. A more computationally efficient but suboptimal approach to estimate the permutation matrix between the two frequency bins is given by the following algorithm.

Algorithm I: Unifying the permutations at two frequency bins ω_i and ω_j

1. Initialize the $N \times N$ matrix $\mathbf{\Pi}_{ij}$ to zero.
2. Setup matrices:

$$\mathbf{E}(\omega_i) = [\mathbf{e}_1(\omega_i), \cdots, \mathbf{e}_N(\omega_i)] \quad (26)$$

   and likewise $\mathbf{E}(\omega_j)$, where

$$\mathbf{e}_l(\omega_i) = (d_l(\omega_i, 0), \cdots, d_l(\omega_i, M-1))^T \quad (27)$$

   has been normalized to unit norm.
3. Form the multiplication

$$\mathbf{T}_{ij} = \mathbf{E}^T(\omega_i)\mathbf{E}(\omega_j). \quad (28)$$

4. Find the row r_{max} and column c_{max} corresponding to largest (in absolute value) element of $\mathbf{T}$. Delete all elements corresponding to this row and column and set

$$\mathbf{\Pi}_{ij}(c_{max}, r_{max}) = 1. \quad (29)$$

5. Repeat step 4 for the remaining elements of matrix $\mathbf{T}_{ij}$ until only one element remains. Set

$$\mathbf{\Pi}_{ij}(c_f, r_f) = 1 \quad (30)$$

   where r_f and c_f are the corresponding row and column of the remaining element.

The above algorithm calculates the required matrix $\mathbf{\Pi}_{ij}$ such that two frequency bins ω_i and ω_j will have the same permutations. Using this we can devise a sorting algorithm to align permutations over all frequency bins. A simple approach is a sequential sorting method, where starting from frequency bin 1, we adjust the permutation of each bin to that of its previous bin. This approach, although simple, has a major drawback as follows: consider the situation where we cannot detect the correct permutation matrix for frequency bin ω_k. In this case all frequency bins succeeding ω_k will receive a different permutation than the ones preceding ω_k. In the worst case, we will have half the frequency bins having one permutation matrix, and the other half having a different permutation matrix, resulting in very poor or virtually no separation performance. To prevent such a catastrophe we propose following hierarchical algorithm to sort the permutations across all frequency bins.

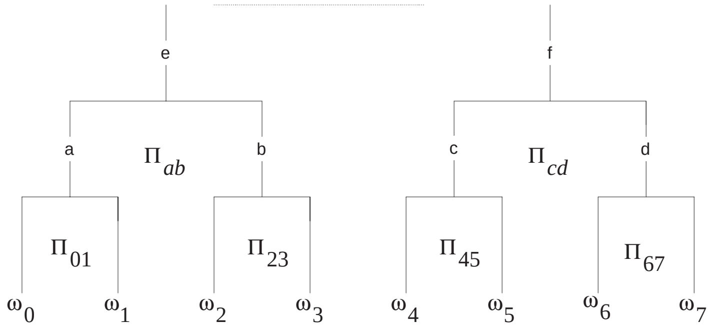

Figure 1: Tree diagram of the permutation sorting algorithm

Algorithm II: Sorting algorithm for unifying the permutations across all frequency bins

1. Divide the frequency bins into groups of two bins each as is shown in figure 1.
2. Use algorithm I for each group to estimate the permutation matrix $\mathbf{\Pi}_{ij}$. Update the order of diagonal values of $\mathbf{D}_y(\omega_i, m)$, where ω_i is one of the frequency bins inside the group, and the matrix $\mathbf{\Phi}(\omega_i)$ using

$$\begin{aligned} \mathbf{D}_y(\omega_i, m) &= \mathbf{\Pi}_{ij}\mathbf{D}_y(\omega_i, m)\mathbf{\Pi}_{ij}^T \\ \mathbf{\Phi}(\omega_i) &= \mathbf{\Pi}_{ij}\mathbf{\Phi}(\omega_i). \end{aligned} \quad (31)$$

3. For each group choose a representative by averaging the diagonal values of $\mathbf{D}(\omega_i, m)$ over the frequency bins in that group:
$$\mathbf{D}_k(m) = \mathbf{D}(\omega_i, m) + \mathbf{D}(\omega_{i+1}, m) \tag{32}$$
4. Move up one level in the hierarchy: Divide the set of representatives into groups of two elements and repeat step 2. Notice this time the update given in (31) is done for all bins corresponding to each representative.
5. Repeat this process for each level until only two representatives remain at the last level.

Figure 1 shows the tree structure of the algorithm II for the case that the total number of frequency bins are 8. The success of this algorithm is based on the assumption that the probability of detecting the wrong permutations increases as the algorithm progresses up the hierarchy. This assumption is justified by the fact that through (32), more information is available upon which to detect wrong permutations at the higher levels.

6. SIMULATION RESULTS

In this section we present two sets of numerical results and demonstrate the performance of the algorithm using real world recorded speech signals. In [12] the authors demonstrate separation of non-stationary Gaussian *iid* sequences for instantaneous mixtures. Here we demonstrate the result for the convolutive case. For this experiment we use three *iid* Gaussian signals and we create the nonstationarity by multiplying them by slowly varying sine, cosine and slowly decaying exponential waveforms:

$$\begin{aligned} s_1(k) &= \sin(-2\pi k/T1)n_1(k) \\ s_2(k) &= \cos(-2\pi k/T2)n_2(k) \\ s_3(k) &= \exp(-2\pi k/T3)n_3(k) \end{aligned} \tag{33}$$

where $n_i(t)$ is a Gaussian *iid* signal. The signals are then mixed using an 8-tap, 3 × 3 FIR polynomial matrix whose coefficients for each element are chosen randomly from a Gaussian distribution. To estimate the cross-spectral density matrices of the observed signal, we first divide the input sequence into M epochs and we use the following formula to estimate the cross-spectral density matrix at the m_{th} epoch:

$$\hat{\mathbf{P}}_x(\omega, m) = \frac{1}{N_s}\sum_{i=0}^{N_s-1} \mathbf{x}_i(\omega, m)\mathbf{x}_i^{\dagger}(\omega, m) \tag{34}$$

where

$$\mathbf{x}_i(\omega, m) = \sum_{n=-\infty}^{\infty} \mathbf{x}(n)w(n - iT_s - mT_b)e^{-j\omega n} \tag{35}$$

where m is the epoch index, N_s is number of overlapping windows inside each epoch, T_b is the size of each epoch, T_s is the time shift between two overlapping windows and $w(n)$ is the windowing sequence (Hanning window for these simulations). These estimated cross-spectral density matrices were then used as the input to our algorithm. We used both the Newton and the gradient descent algorithms for the joint diagonalization part of the method. We also used the diadic structure, described in previous section, to resolve the permutation problem across the frequency bins. Since in simulations we know the mixing channel, we can estimate the separation performance by first convolving the separation matrix $\mathbf{B}(k)$ with the channel $\mathbf{H}(k)$ to obtain

$$\mathbf{C}(k) = \mathbf{B}(k) * \mathbf{H}(k). \tag{36}$$

Next we quantify the performance using the formula:

$$\eta(i) = 20\log\Big(\frac{\{\sum_{j=1}^{N} ||\mathbf{c}_{ij}||_2^2\} - \max_j ||\mathbf{c}_{ij}||_2^2}{max_j ||\mathbf{c}_{ij}||_2^2}\Big) \quad i = 1, .., N \tag{37}$$

where $\eta(i)$ represents the row-wise interference to signal ratio for i_{th} output, and

$$\mathbf{c}_{ij} = (C_{ij}(0), \cdots, C_{ij}(L_c - 1))^T \tag{38}$$

where $C_{ij}(k)$ represents the ij_{th} element of the $\mathbf{C}(k)$. Table 1 shows the performance results before and after the permutation algorithm was applied. As can be seen, permutation error severely degrades the performance of the algorithm. Also figure 2 shows the separation results for this experiment.

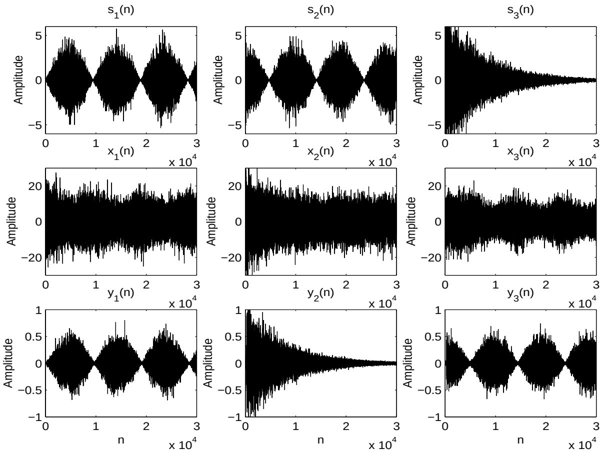

Figure 2: Results of separation: original sources $s_1(n)$, $s_2(n)$ and $s_3(n)$, observed signals $x_1(n)$, $x_2(n)$ and $x_3(n)$, separated outputs $y_1(n)$, $y_2(n)$ and $y_3(n)$.

In the next experiment we imposed a harder, more realistic situation. We used two male speech signals which were mixed through a 4 × 2 polynomial matrix with order 20 (each element of the mixing matrix is a 20 tap FIR filter). We also added white Gaussian noise at a controlled level to the result of this mixture. The output signal to interference ratios (SIRs) were then calculated for different values of noise power using 50 Monte Carlo runs. The average signal to noise ratio for each source is given by:

$$SNR_j = \frac{\ll \sum_{i=1}^{J}(h_{ij}(n) * s_j(n))^2 \gg}{J \ll n_i(k)^2 \gg} \tag{39}$$

where $\ll . \gg$ represents the time average operation. Figure 3 shows the resulting curves of output SIR versus average signal to noise ratio of the sources. The results for this experiment can be heard from [13].

Output ISR (dB)	$\eta(1)$	$\eta(2)$	$\eta(3)$
Without Resolv. Permut.	-0.58dB	2.52dB	-0.75dB
With Resolv. Permut.	-21.9dB	-24.1dB	-21.6dB

Table 1: Output SIRs before and after correcting the permutations.

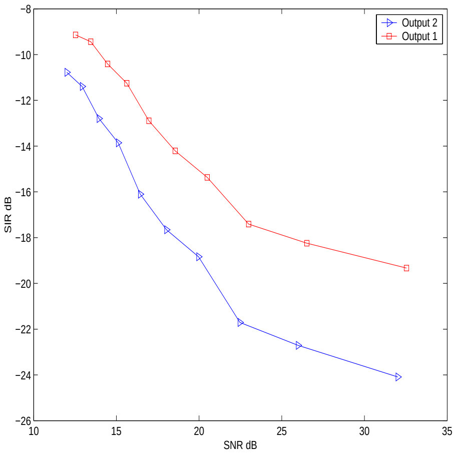

Figure 3: Output SIR versus the average signal to noise ratio (SNR) of the corresponding source

We also applied our method to real world data recorded from the room experiments contributed by [14]. In this experiment, since we don't have access to the true room impulse responses nor the true original sources, a quantitative measurement is not possible. However, the output speech can be heard from [13].

Further, it was found that the method proposed in this paper is significantly faster in convergence than that proposed in [5], for problems of similar size.

7. CONCLUSION

In this paper we presented a new method for blind source separation of MIMO convolutive mixtures using the assumption that the sources are non-stationary. We presented the method as a three stage algorithm (Whitening, Orthogonal joint diagonalization, and Adjusting Permutations) where the first two stages calculate the separation matrix at each frequency bin and the last stage detects and removes the wrong permutations. For orthogonal joint diagonalization we designed an adaptive algorithm based on the Newton method over the complex Steifel manifold. For aligning permutations we presented a novel diadic algorithm which exploits cross frequency correlations between diagonal values of the output cross–spectral density matrices. We tested our algorithm using synthetic data and real speech data in different mixing scenarios. We also applied the algorithm to real world recorded speech samples. In all tests, excellent performance was achieved.

8. ACKNOWLEDGEMENTS

The authors wish to acknowledge and thank the following institutions for their support of this project: Mitel Corporation, Kanata, Ontario, Canada; The Centre for Information Technology Ontario (CITO); and the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors also acknowledge the helpfull comments from Dr. Jonathan H. Manton, the corresponding author of [9].

9. REFERENCES

[1] D. Yellin and E. Weinstein, “Multichannel signal separation: methods and analysis,” *IEEE Transactions on Signal Processing*, vol. 44, pp. 106–118, Jan. 1996.

[2] H. Sahlin and H. Broman, “MIMO signal separation for fir channels: a criterion and performance analysis,” *IEEE Transactions on Signal Processing*, vol. 48, pp. 642–649, March. 2000.

[3] J. K. Tugnait, “Adaptive blind separation of convolutive mixtures of independent linear signals,” *Signal Processing*, vol. 73, pp. 139–152, 1999.

[4] J. Pesquet, B. Chen, and A. P. Petropulu, “Frequency-domain contrast functions for separation of convolutive mixtures,” in *Proc. ICASSP2001*, May 2001.

[5] L. Parra and C. Spence, “Convolutive blind separation of non-stationary sources,” *IEEE Trans. Speech and Audio Processing*, vol. 8, pp. 320–327, May. 2000.

[6] S. Ikeda and N. Murata, “A Method Of ICA in Time-Frequency Domain ,” in *Proc. ICA pp.365-370*, Jan. 1999.

[7] J. Cardoso and A. S. Soulomiac, “Jacobi angles for simultaneous diagonalization,” *SIAM Journal on Matrix Analysis and Applications*, vol. 17, pp. 161–164, no. 1 1996.

[8] K. Rahbar and J. Reilly, “Geometric optimization methods for blind source separation of signals,” in *Proc. ICA pp.375-380*, June 2000.

[9] J. H. Manton, “Optimization Algorithms Exploiting Complex Orthogonality Constraints,” *IEEE Transactions on Signal Processing*, submitted.

[10] A. Edelman, T. Arias, and S. T. Smith, “The geometry of algorithms with orthogonality constraints,” *SIAM Journal on Matrix Analysis and Applications*, vol. 20, pp. 303–353, Feb. 1998.

[11] M. Z. Ikram and D. R. Morgan, “Exploring Permutation Inconsistency in Blind Separation Of Signals In a Reverberant Environment,” in *Proc. ICASSP2000*, May 2000.

[12] D. Pham and J. Cardoso, “Blind separation of instantaneous mixtures of non-stationary sources,” in *Proc. ICA pp.187-190*, June 2000.

[13] www.ece.mcmaster.ca/∼reilly/kamran/index.htm.

[14] www.humanism.org/∼lucas/research/.