4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), April 2003, Nara, Japan

BLIND SEPARATION OF CONVOLUTIVE AUDIO MIXTURES USING NONSTATIONARITY

Dinh-Tuan Pham

Laboratoire de Modélisation et Calcul,
BP 53X, 38041 Grenoble Cedex, France
Dinh-Tuan.Pham@imag.fr

Christine Servière, Hakim Boumaraf

Laboratoire des Images et des Signaux
BP 46, 38402 St Martin d'Hère Cedex, France
Christine.Serviere@inpg.fr
Hakim.Boumaraf@lis.inpg.fr

ABSTRACT

We consider the blind separation of convolutive mixtures based on the joint diagonalization of time varying spectral matrices of the observation records. The goal is to separate audio mixtures in which the mixing filter has quite long impulse responses and the signals are highly non stationary. We rely on the continuity of the frequency response of the filter to eliminate the permutation ambiguity. Simulations show that our method works well when there is no strong echos in the mixing filter. But if it is not the case, the permutation ambiguity cannot be sufficiently removed.

1. INTRODUCTION

Blind source separation consists in extracting independent sources from their mixtures, without relying on any specific knowledge of the sources. Earlier works have been focused on linear instantaneous mixtures and there are quite a few proposed methods which perform well. However, for audio convolutive mixtures are involved, the performance of various proposed separation methods is often poor, as the convolution filter can have very long impulse response [1]. But audio signals are also highly nonstationary and one can exploit this nonstationarity to separate the sources. In fact, we have relied crucially on this feature to separate the sources, since for computational simplicity reason, we shall adopt a second order statistics based approach, which is well known to be insufficient for blind source separation in the stationarity case. Similar ideas have been introduced in [2, 3]. Using only second order statistics would of course entail some loss of information, but we feel that this would be slight, as most of the information in audio signals seem to be contained in their time frequency spectra.

2. MODEL AND METHODS

From the theoretical point of view, the problem we are considering is the blind separation of convolutive mixtures. Specifically, K observation sequences $\{x_k(t)\},\ k = 1,\dots,K$ are available:

$$x_k(t) = \sum_{n=-\infty}^{\infty} \sum_{j=1}^{K} H_{kj}(n) s_j(t-n) \tag{1}$$

where $\{s_1(t)\},\dots,\{s_K(t)\}$ denote the sources sequences and $\{H_{kj}(n)\}$ are elements of the impulse response matrix $\{\mathbf{H}(n)\}$ of the mixing filter. The goal is to recover the sources through another filtering operation $\mathbf{y}(t) = \sum_{n=-\infty}^{\infty} \mathbf{G}(n)\mathbf{x}(t-n)$ where $\mathbf{x}(t) = [x_1(t) \ \cdots \ x_K(t)]^{\mathbf{T}}$ ($^{\mathbf{T}}$ denoting the transpose), $\mathbf{y}(t) = [y_1(t) \ \cdots \ y_K(t)]^{\mathbf{T}}$ is the recovered source vector and $\{\mathbf{G}(l)\}$ is the impulse response matrix of the separation filter.

In the blind context, the idea is to adjust the filter $\{\mathbf{G}(n)\}$ such that the reconstructed sources $\{y_k(t)\}$ are as mutually independent as it is possible. By adopting a second order approach, we are in fact focused only on the inter-correlations between the reconstructed sources at all lags or equivalently the inter-spectra between them at all frequencies. However, as we are dealing with nonstationary signals, we need to consider the time varying spectra, that is the localized spectra around each given time point. It is precisely the time evolution of these spectra which provides us the extra information to separate the sources[1].

From (1), the time varying spectrum of the vector observation sequence $\{\mathbf{x}(t)\}$ is $S_{\mathbf{x}}(t,f) = \mathbf{H}(f)S_{\mathbf{s}}(t,f)\mathbf{H}^*(f)$ where $\mathbf{H}(f) = \sum_{n=-\infty}^{\infty} e^{inf2\pi}\mathbf{H}(n)$ denotes the frequency response of the mixing filter[2] at frequency f, $S_{\mathbf{s}}(t,f)$ is the diagonal matrix with diagonal elements being the time varying spectra of the sources and * denotes the transpose conjugated. Thus to perform the separation, a natural idea is to find matrices $\mathbf{G}(f)$ such that for each frequency f the matrices $\mathbf{G}(f)\hat{S}_{\mathbf{x}}(t,f)\mathbf{G}^*(f)$, at different

[1] In the *stationary case*, it is well known that the use of second order statistics alone is insufficient to separate convolutive mixtures, unless there is some contraint such as a known (small) maximum the number of lags

[2] For simplicity, we use the same symbol to denote the impulse or the frequency response of a filter, depending to its argument

Thanks to the European Bliss Project for funding.

time points t, are as close to diagonal as is possible, where $\hat{S}_{\mathbf{x}}(t,f)$ are estimates of $S_{\mathbf{x}}(t,f)$. This idea has been exploited by Para and Spence [2, 3], but they use a different diagonality criterion as ours. The one we use is the same an in [4], which is also considered in [5] but without using the nonstationarity idea. It arises naturally from the likelihood or the mutual information approach and experience in the case of instantaneous mixture shows that it is a powerful criterion. Besides, we have developed a simple and very fast algorithm to perform joint approximate diagonalization based on minimizing this criterion [6]. For a single matrix $\mathbf{G}(f)\hat{S}_{\mathbf{x}}(t,f)\mathbf{G}^*(f)$ the diagonality measure is given by

$$\frac{1}{2}\{\log\det\operatorname{diag}[\mathbf{G}(f)\hat{S}_{\mathbf{x}}(t,f)\mathbf{G}^*(f)] - \log\det[\mathbf{G}(f)S_{\mathbf{x}}(t,f)\mathbf{G}^*(f)]\}$$

where $\operatorname{diag}(\cdot)$ denotes the operator which builds a diagonal matrix from its argument. But the last term equals $2\log|\det\mathbf{G}(f)| + \log\det\hat{S}_{\mathbf{x}}(t,f)$ and the term $\log\det\hat{S}_{\mathbf{x}}(t,f)$, being constant, can be dropped. Therefore a global diagonality criterion can be written as

$$\sum_t \left\{\frac{1}{2}\log\det\operatorname{diag}[\mathbf{G}(f)\hat{S}_{\mathbf{x}}(t,f)\mathbf{G}^*(f)] - \log\det|\mathbf{G}(f)|\right\} \tag{2}$$

where the summation is over the time points of interest. This criterion is to be minimized with respect to $\mathbf{G}(f)$ to obtain the frequency response of the separation filter.

The above frequency domain approach has the great advantage that the calculations can be done in each frequency bin separately and independently. This is very important since in the present application the number of these bins must be very large or equivalently the sequence of the impulse response of the separation filter must be very long. A time domain approach would require the minimization of some criterion with respect to a very large number of parameters, which is too costly, not to mention the difficulty of ensuring the convergence of the minimization algorithm in such a situation. By contrast, in the frequency approach, for each frequency bin, one only has a small minimization problem, which can be solved very quickly. There is however a price to be paid for this. The joint diagonalization only provides the matrices $\mathbf{G}(f)$ up to a scale change and a permutation: if $\mathbf{G}(f)$ is a solution then so is $\mathbf{\Pi}(f)\mathbf{D}(f)\mathbf{G}(f)$ for any diagonal matrix $\mathbf{D}(f)$ and any permutation matrix $\mathbf{\Pi}(f)$. Thus, one only gets a separation filter of frequency response matrix of the form

$$\mathbf{G}(f) = \mathbf{\Pi}(f)\mathbf{D}(f)\hat{\mathbf{H}}^{-1}(f) \tag{3}$$

where $\hat{\mathbf{H}}(f)$ is a consistent estimator of $\mathbf{H}(f)$ but $\mathbf{\Pi}(f)$ and $\mathbf{D}(f)$ are *arbitrary* permutation and diagonal matrices.

It should be note that the above ambiguity problem is not really related to the frequency approach but to the use of a criterion such as (2) which expresses the mutual dependence of the signals in a decoupling way in the frequency domain. The scale ambiguity is however intrinsic to the blind separation of convolutive mixtures and can only be lifted with some a priori knowledge about the sources or the mixing mechanism. The permutation ambiguity will be dicussed in next section.

3. THE PERMUTATION AMBIGUITY PROBLEM

It is a difficult problem which is still open to a satisfying solution, particularly for room acoustics. Several ideas emerged. The first one consists in constraining the separating filters with short support FIR structures in time domain as permutations induce IIR filters or with very long tails. It may be not easy to handle, as for long responses the inverse is usually longer [5]. Other ideas are to add frequency coupling for example in the adaptation parameters to preserve the same permutation [7, 8], or to exploit a continuity assumption on the frequency response of the unmixing filters [5, 7]. In the time domain approach, the permutation ambiguity does not arise explicitly because one often works with filters of rather short impulse response, which implicitly imposes a continuity condition on its frequency response. In this paper, we also exploited this continuity idea. But we have found that it has some limit in this audio application.

Our approach is to exploit the continuity of the frequency response $\mathbf{H}(f)$ of the mixing filter. It has the advantage that it relies only on this weak assumption and requires a very little computational cost. Thus, we impose that the frequency response $\mathbf{G}(f)$ of separating filter be continuous in f. Since a permutation function cannot be continuous unless it is a constant function, this constraint reduces the ambiguity with respect to a permutation *varying with the frequency* to that with respect to a fixed one. One cannot do better in the context of blind source separation, since permuting the sources and/or applying a filter to each of them does not change in anyway their mutual independence.

In practice, $\mathbf{G}(f)$ will be available only over a finite regular grid of frequcuencies $f_0 < \ldots < f_L$, say. To detect if a jump of $\mathbf{\Pi}(f)$ has occurred in (f_{l-1}, f_l), we look the "ratio" $\mathbf{G}(f_l)\mathbf{G}^{-1}(f_{l-1})$ and test for its closeness to a diagonal matrix. Indeed, by using the representation (3), this ratio, which we denote by $\mathbf{R}(f_l, f_{l-1})$, can be written as:

$$\mathbf{\Pi}(f_l)[\mathbf{D}(f_l)\hat{\mathbf{H}}^{-1}(f_l)\hat{\mathbf{H}}(f_{l-1})\mathbf{D}^{-1}(f_{l-1})]\mathbf{\Pi}^{-1}(f_{l-1})$$

Since the function $\hat{\mathbf{H}}(f)$ is continuous, $\hat{\mathbf{H}}^{-1}(f_l)\hat{\mathbf{H}}(f_{l-1})$ is nearly the identity matrix, hence the product in the above square bracket [ ] is nearly a diagonal. Therefore, the matrix

$$\mathbf{\Delta}(f_l, f_{l-1}) = \mathbf{\Pi}(f_{l-1})[\mathbf{D}(f_l)\hat{\mathbf{H}}^{-1}(f_l)\hat{\mathbf{H}}(f_{l-1})\mathbf{D}^{-1}(f_{l-1})]\mathbf{\Pi}^T(f_{l-1}),$$

which is the same matrix but with its rows and column permuted by the same permutation, is also nearly diagonal. Thus, since $\mathbf{\Pi}(f_{l-1})$ is orthogonal, $\mathbf{R}(f_l, f_{l-1}) = \mathbf{\Pi}(f_l)\mathbf{\Pi}^{-1}(f_{l-1})\mathbf{\Delta}(f_l, f_{l-1})$, which appears as a product of a permutation matrix with a nearly diagonal matrix.

The above method is much better than just comparing $\mathbf{G}(f_l)$ with $\mathbf{G}(f_{l-1})$, since a nearly diagonal matrix is easy to recognize. By contrast, the matrix $\mathbf{H}^{-1}(f_l)$ may have two rows nearly proportional and then a permutation of the rows in $\mathbf{G}(f_l)$, accompanied by an appropriate rescaling, would not change it by much and thus $\mathbf{G}(f_l)$ can be quite similar to $\mathbf{G}(f_{l-1})$ even if a jump in $\mathbf{\Pi}(f)$ has occurred. Another advantage of our method is that we can cancel the permutation jump by looking for a permutation matrix $\mathbf{\Pi}$ such that the matrix $\mathbf{\Pi}\mathbf{R}(f_l, f_{l-1})$ be the closest to a diagonal matrix, according to some criterion. If the obtained matrix, denoted by $\mathbf{\Pi}(f_l, f_{l-1})$, is the identity matrix, this means that $\mathbf{\Pi}(f_l) = \mathbf{\Pi}(f_{l-1})$ and we has nothing to do, otherwise, there is a permutation jump in (f_{l-1}, f_l) and we can cancel it out by pre-multiplying $\mathbf{G}(f_l)$ by $\mathbf{\Pi}(f_l, f_{l-1})$.

It remains to define a criterion of non diagonality for an arbitrary square complex matrix. Since the matrices $\hat{\mathbf{G}}(f_l)$ are intrinsically ambiguous in terms of scale, it is desirable to use a criterion which is scale invariant in the sense that it is unchanged when the matrix to which it is applied is pre- or post-multiplied by a diagonal matrix. A simple criterion which meets this requirement and is well suited to our algorithm is

$$d(\mathbf{M}) = \max_{(i_1,\dots,i_k)\neq(1,\dots,K)} \frac{|M_{i_1 1}| \cdots |M_{i_1 K}|}{|M_{11}| \cdots |M_{KK}|}$$

where M_{ij} denotes the general element of the $K \times K$ matrix $\mathbf{M}$ and the maximum is over all permutations $(i_1, \dots, i_K)$ of $1, \dots, K$ distinct from the identity permutation. With this criterion, our algorithm can be summarized as follows.

(i) Compute the matrices $\mathbf{R}(f_l, f_l) = \mathbf{G}(f_l)\mathbf{G}^{-1}(f_{l-1})$, $l = 1, \dots, L$.

(ii) for each $l = 1, \dots, L$, determine the permutation $(i_1, \dots, i_K)$ such that the product $\prod_{j=1}^{K} |\hat{\mathbf{R}}(f_l, f_{l-1})_{i_j j}|$ be the maximum among all other permutations, $\mathbf{R}(f_l, f_{l-1})_{i_j j}$ denoting the i_j, j element of the matrix $\mathbf{R}(f_l, f_{l-1})$.

(iii) If $(i_1, \dots, i_K)$ is the identity permutation, do nothing. Otherwise permute the rows $1, \dots, K$ of $\mathbf{G}(f_l), \dots, \mathbf{G}(f_L)$ into rows $i_1, \dots, i_K$.

Unfortunately, the stage (ii) of this algorithm can be quite time consuming for K large since one has to search all permutations. One may note however that the matrix $\mathbf{R}(f_l, f_{l-1})$ should be close to the product of a permutation and a diagonal matrix. Therefore, defining the indices $i_1, \dots, i_K$ by $i_j = \arg\max_{i=1,\dots K} |\mathbf{R}(f_l, f_{l-1})_{ij}|$, there is a good chance that these indices are distinct and thus define precisely the permutation we are looking for. If this is not the case, one must in principle examine again all permutations. But this case is exceptional and since the matrix $\mathbf{R}(f_l, f_{l-1})$ would then be not close to the product of a permutation and a diagonal matrix and the permutation *is not well determined anyway*. We have however developed simple suboptimal solution but still optimal for $K \leq 3$.

Another much simpler method, relying on the particular behavior of our joint (approximate) diagonalization algorithm, which we find to work quite well, is as follows. Instead of jointly diagonalizing the matrices $\hat{S}_{\mathbf{x}}(t, f_l)$ we jointly diagonalize the matrices $G(f_{l-1})\hat{S}_{\mathbf{x}}(t, f_l)G^*(f_{l-1})$, where $G(f_{l-1})$ is the solution to the previous problem of joint diagonalization of the $\hat{S}_{\mathbf{x}}(t, f_{l-1})$. By continuity, we expect that the matrices $G(f_{l-1})\hat{S}_{\mathbf{x}}(t, f_l)G^*(f_{l-1})$ are already rather close to diagonal so that a solution to their joint diagonalization problem is nearly the identity matrix. Further, our joint diagonalization algorithm operates by transforming successively the matrices to be diagonalized by pre- and post- multiplying them by an appropriate matrix, and each time between two candidates for such a matrix, differing only by a permutation, the one which is closer to the identity matrix (in some sense) is chosen [6]. Thus, the algorithm would produce a matrix ratio $G(f_l)G^{-1}(f_{l-1})$ close to the identity matrix and hence no subsequent permutation is needed. In practice, we find that this method generally agrees with the "permutation algorithm" described earlier. The case when they differ significantly is when the matrix $G(f_l)$ changed too fast (with l), but then permutation is not well detected anyway. One can also apply the present method in conjunction with the permutation algorithm. This may speed up the calculation since the joint diagonalization algorithm is initialized at a point closer to its solution.

4. SPECTRAL ESTIMATION

The first step in the separation procedure is to estimate the spectral matrix of the observation sequences. Since we are dealing with time varying spectrum, the simplest way is to subdivide the data sequence into consecutive blocks and estimate the spectrum as if the data inside each block come from stationary processes. It is possible to consider moving (overlapping) blocks. But this increases the computational cost and we have found, in practice, that this can only marginally improve the performance.

There are several ways to estimate the spectrum of a (multivariate) signal [9]. We focus on frequency domain methods as time domain methods are too costly since a large number of lags would be needed. A first method is to subdivide again the data block into elementary blocks, compute the Fourier transform in each of them to form the "short term" periodogram (see definition below) and then average the periodograms over all elementary blocks. It is discarded because of the "end effect" problem: the mixing filter has a quite long impulse response, hence the Fourier transform of a short segment of the filtered signal cannot be well approximated by the product of the frequency response of the filter and the Fourier transform of the same segment of the signal. Thus we adopt a second method which consists in computing the Fourier transform of the whole data block, forming the periodogram and then averaging it over consecutive frequencies. Specially, let the block length be N, we first compute the periodogram of the $(k+1)$-th data block by

$$P_{\mathbf{x}}(k{+}1, f) = \frac{1}{N}\Big[\sum_{t=kN+1}^{kN+N} \mathbf{x}(t)e^{2\pi i f t}\Big]\Big[\sum_{t=kN+1}^{kN+N} \mathbf{x}(t)e^{2\pi i f t}\Big]^*.$$

The frequencies[3] are taken to be of the form $f = n/N, n = 0, \ldots, N/2$ to take advantage of the Fast Fourier Transform, as N will be chosen to be a power of 2. The time varying spectrum at the mid point $t_k = kN - (N-1)/2$ of the k-th block is then simply estimated by

$$\hat{S}_{\mathbf{x}}\Big(t_k, \frac{n}{N}\Big) = \frac{1}{2m+1}\sum_{l=n-m}^{n+m} P_{\mathbf{x}}\Big(k, \frac{l \bmod N}{N}\Big).$$

where m is a bandwidth parameter.

One added advantage of the above approach over the "short term" periodogram is that one obtains the estimated spectrum over a fine grid of frequencies, which is helpful for the detection of permutation.

5. DESIGN AND SIMULATION RESULTS

We considered mixtures of real sound sources from premeasured room impulse responses. The last are provided by the matlab routine roomix.m of Alex Westner (found in http://sound.media.mit.edu/ica-bench), which uses a library of impulse responses measured off a real 3.5m $\times$ 7m $\times$ 3m conference room. The user specifies the positions of the sensors and the sources (using 8 preset positions). The sources signals and room responses are sampled at 22.05 kHz with a duration of 9 seconds.

The two sources are a speech and a music sound wave, with a length limited to 61440 samples (2.7864 sec). As for the mixing filter, we find that the impulse response provided by roomix is too long (up to 16383 lags) but can be truncated to about 512 lags without losing any "peak" corresponding to echos. Therefore we have, at first, taken this filter as a basis for our simulation. But we find that, although our separation algorithm works quite well for each frequency, the

[3] actually the relative frequency with respect to the sampling frequency

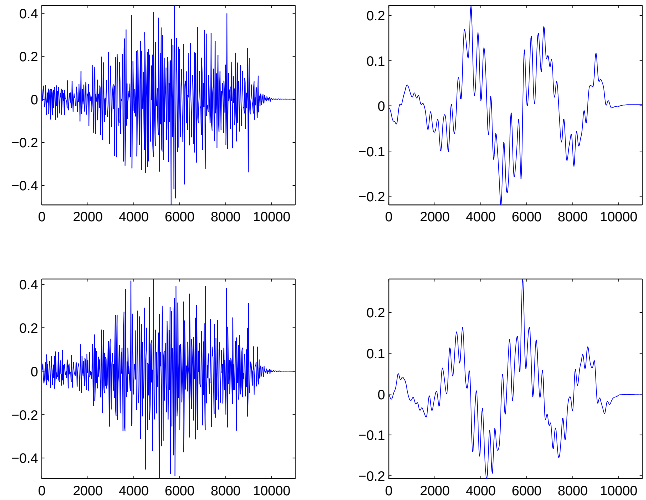

Fig. 1. Frequency response of $\mathbf{H}_{21}$ for the 512-lags filter (left panel) and the modified filter (right panel); upper row = real part, lower row = imaginary part

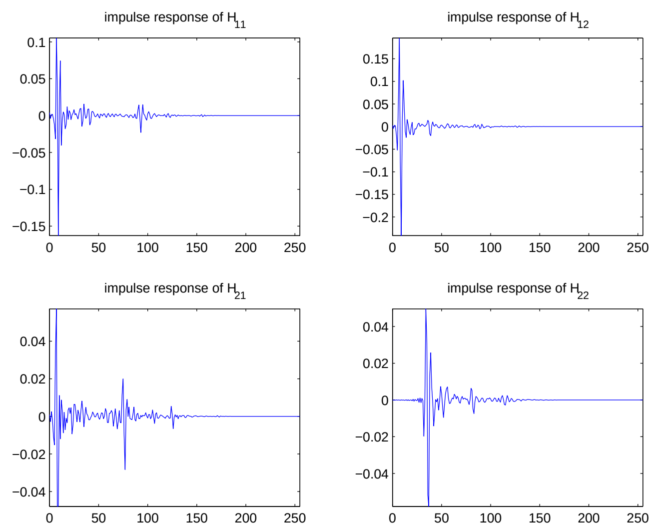

Fig. 2. Impulse response of the modified filter

permutation ambiguity cannot be eliminated. This problem is principally due to the presence of strong echos, which induce very rapid oscillations of the frequency response of the mixing filter (see figure 1). Therefore we actually test our method with a modified mixing filter which essentially attenuates the echos. We first note that the impulse responses have many leading zeros (or close to zero) due to delay of responses, therefore we shift the sources and also the second sensor with respect to the first to eliminate most of them. We then attenuate the responses by multiplying with a lag window vanishing at lag 256 and beyond. Figure 2 plots the impulse responses of the modified filter. Its frequency response can be seen from figure 1 to be much smoother than that of the 512-lags filter.

We take as block length $N = 4096$ (yielding 15 time

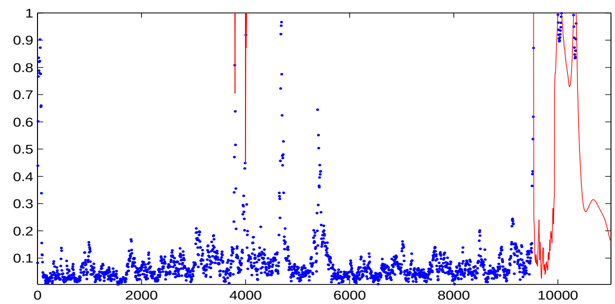

Fig. 3. Separation index (blue dots) and its inverse (solid red), truncated at 1

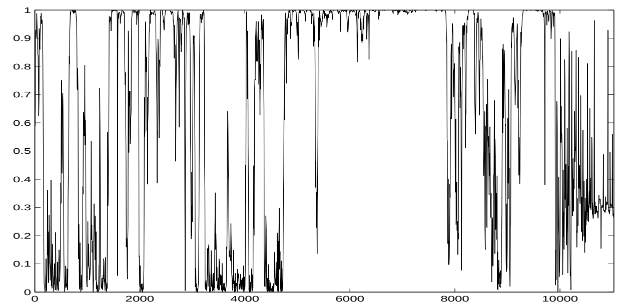

Fig. 5. Coherence (the 512-lags filter)

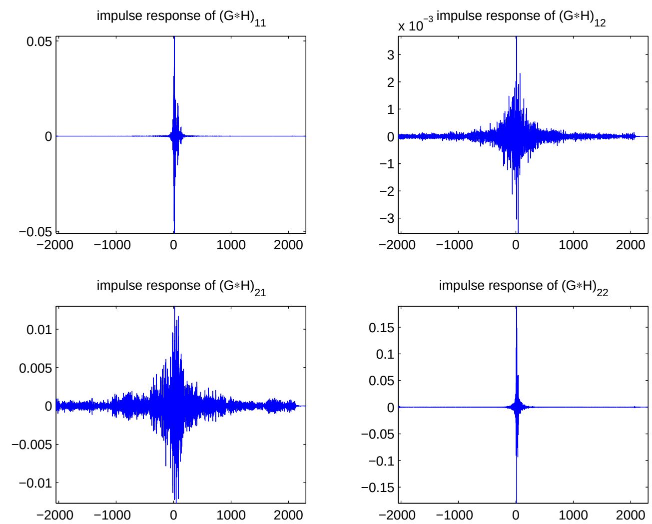

Fig. 6. Impulse response of the global filter $(\mathbf{G} * \mathbf{H})(n)$

blocks) and estimate the spectral matrices by averaging over 9 frequencies ($m = 4$). A first indicator of the performance of the method is the separation index, defined as

$$r(f) = |(\mathbf{GH})_{12}(f)(\mathbf{GH})_{21}(f)/[(\mathbf{GH})_{11}(f)(\mathbf{GH})_{22}(f)]|^{1/2}$$

where $(\mathbf{GH})_{ij}(f)$ is the ij element of the matrix $\mathbf{G}(f)\mathbf{H}(f)$. For a good separation, this index should be close to 0 or infinity (in this case the estimated sources are permuted). Figure 3 plots $\min(r, 1)$ and $\min(1/r, 1)$ versus frequency (in Hz). When r crosses the value 1, this means that a permutation has occurred. One can see from figure 3 that this happens only at some isolated frequencies near 4KHz and at high frequencies, which is unimportant since there is no signal energy there. The same plot for the 512-lags filter (not shown) however indicates a lot of permutations.

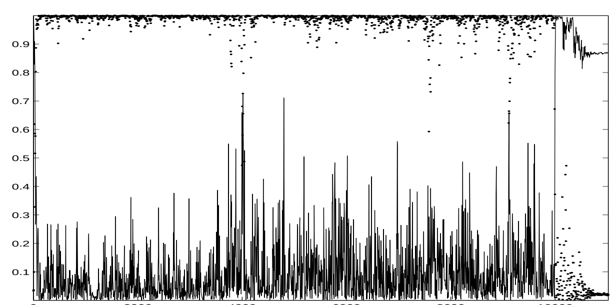

Fig. 4. Coherences (the modified filter)

Another performance index is the coherence between the 1st estimated source and the contribution to the 1st sensor of the two true sources. Good separation will result in coherence near 0 and 1 and permutation occurs when the two curves cross. Figure 4 shows that the coherence is good and permutations occur rarely, in the case of the modified filter. For the 512-lags filter however, the coherence switches constantly between 0 and 1, as is seen in figure 5 (only one coherence is plotted for clarity)

The impulse response of the global filter $(\mathbf{G} * \mathbf{H})(n)$ is shown in figure 6. One can see that $(\mathbf{G} * \mathbf{H})_{12}(n)$ is much smaller than $(\mathbf{G} * \mathbf{H})_{11}(n)$ and $(\mathbf{G} * \mathbf{H})_{21}(n)$ is much smaller than $(\mathbf{G} * \mathbf{H})_{22}(n)$, meaning that the sources are nearly separated. This can be confirmed by looking at the original sources, the mixtures and the separated sources, displayed in figure 7.

6. DISCUSSION AND CONCLUSION

Our method proves to work quite well if the mixing filter does not contain strong echos. The presence of echos is a real problem as it breaks our permutation algorithm (but otherwise the separation is good over all). In fact the rapid variation of $\mathbf{H}(f)$ forces us to work with very narrow frequency bin, both to avoid bias in spectral estimation and to ensure small variation from $\mathbf{G}(f_{l-1})$ to $\mathbf{G}(f_l)$. But increasing the frequency resolution would result in more variability. In fact the problem also comes from the fact that the separating matrix $\mathbf{G}(f)$ cannot be good at *all* frequencies. At some frequencies, it is bad, meaning that the estimated sources (at this frequency) are still mixtures so it would be hard to determine if they correspond to source 1 or 2. On top of that, the mixing matrices $\mathbf{H}(f)$ are ill conditioned at

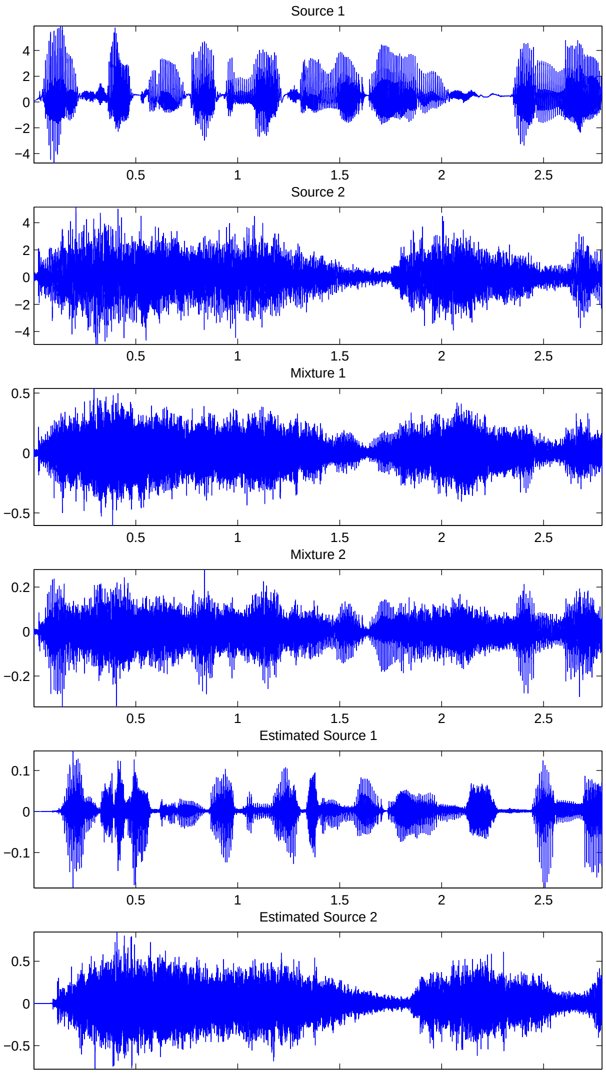

Fig. 7. Sources, mixtures and estimated sources

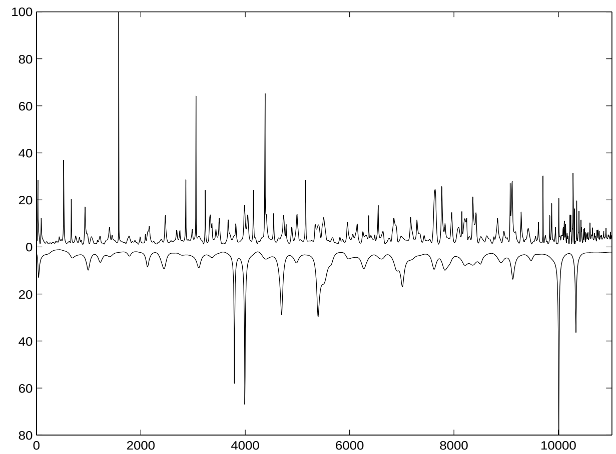

Fig. 8. Condition number of the frequency response matrix of the 512-lags filter (upper half) and the modified filter (lower half, downward)

many frequencies (see figure 8), meaning that the columns of $\mathbf{H}(f)$ at these frequencies are nearly proportional. Finally, even a single permutation cannot be tolerated unless it occurs at the low or high frequency range or it is quickly canceled by another permutation. We are working to improve the method, either by considering non adjacent frequencies as in [7, 8] or a two stages procedure where the first stage tries to remove crudely the echos.

7. REFERENCES

[1] R. Mukai, S. Araki, and S. Makino, "Separation and dereverberation performance of frequency domain blind source separation," in *Proceeding of ICA 2001 Conference*, San-Diego, USA, Dec. 2001, pp. 230–235.

[2] L. Parra and C. Spence, "On line blind source separation of non stationary signals," *J. VLSI Signal Proc. Systems for Signal, Images and Video Tech.*, vol. 26, no. 8, pp. 15–24, 2000.

[3] L. Parra and C. Spence, "Convolutive blind source separation of non-stationary sources," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 3, pp. 320–327, May 2000.

[4] D. T. Pham and J.-F. Cardoso, "Blind separation of instantaneous mixtures of non stationary sources," *IEEE Trans. Signal Processing*, vol. 49, no. 9, pp. 1837–1848, 2001.

[5] H.-C. Wu and J. C. Principe, "Simultaneous diagonalization in the frequency domain (SDIF) for source separation," in *Proceeding of ICA 1999 Conference*, Aussois, France, Jan. 1999, pp. 245–250.

[6] D. T. Pham, "Joint approximate diagonalization of positive definite matrices," *SIAM J. on Matrix Anal. and Appl.*, vol. 22, no. 4, pp. 1136–1152, 2001.

[7] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," in *International Workshop on Independence & Artificial Neural Networks*, University of La Laguna, Tenerife, Spain, Feb. 1998.

[8] J. Anemüler and B. Kollmeier, "Amplitude modulation decorrelation for convolutive blind source separation," in *Proceeding of ICA 2000 Conference*, Helsinki, Finland, June 2000, pp. 215–220.

[9] A. Oppenheim and R. Schaffer, *Digital Signal Processing*, Prentice Hall, 1975.