BLIND SEPARATION OF CONVOLVED MIXTURES IN THE FREQUENCY DOMAIN

Paris Smaragdis
Machine Listening Group
MIT Media Laboratory
20 Ames St. Office E15-401C
Cambridge, MA 02139, USA
paris@media.mit.edu

Abstract

In this paper we employ information theoretic algorithms, previously used for separating instantaneous mixtures of sources, for separating convolved mixtures in the frequency domain. It is observed that convolved mixing in the time domain corresponds to instantaneous mixing in the frequency domain. Such mixing can be inverted using simpler and more robust algorithms than the ones recently developed. Advantages of this approach are improved efficiency and better convergence features.

1 INSTANTANEOUS MIXTURES AND SEPARATION

The problem of blind source separation was traditionally approached by observing *instantaneous mixtures* of sources. Assume that N signals s_i are ordered in a vector $\mathbf{s}^T(t) = [s_1(t) \; \ldots \; s_N(t)]$ where t is a time index. Upon transmission through a medium these signals are collected from N sensors from which we obtain $\mathbf{x}^T(t) = [x_1(t) \; \ldots \; x_N(t)]$. Assuming linear superposition the vector $\mathbf{x}$ can be expressed as:

$$\mathbf{x}(t) = \mathbf{A} \cdot \mathbf{s}(t) \tag{1}$$

where $\mathbf{A}$ is an unknown matrix called the *mixing matrix*. The objective is to recover the original $s_i(t)$ signals given only the $\mathbf{x}(t)$ vectors. These signals can be recovered using $\mathbf{s}(t) = \mathbf{A}^{-1} \cdot \mathbf{x}(t)$. If $\mathbf{A}$ is invertible then separation is feasible and the inverse of this mixing matrix will be called the *unmixing matrix*.

This problem was recently addressed successfully by Bell and Sejnowski (1995) and Amari *et al.* (1996) who introduced fast and robust solutions. Both approaches employed information theoretic principles to find an unmixing matrix that would maximize the statistical independence of the estimated original sources. The computational structure that was used was a matrix multiplication between the estimated unmixing matrix and the mixed inputs. After every sample presentation the estimate of the unmixing matrix was updated using the following learning rules, where $\mathbf{y}(t) = \mathbf{W} \cdot \mathbf{x}(t)$:

$$\Delta\mathbf{W} \propto [\mathbf{W}^{-1}]^T - 2 \cdot f(\mathbf{y}(t)) \cdot \mathbf{x}(t)^T \tag{2}$$

which was derived by Bell and Sejnowski (1995), and:

$$\Delta\mathbf{W} \propto [\mathbf{I} - f(\mathbf{y}(t)) \cdot \mathbf{y}(t)^T] \cdot \mathbf{W} \tag{3}$$

derived by Amari *et al.* (1996) who also performed adaptation accounting for the Riemannian structure of the problem. The matrix $\mathbf{W}$ is our estimate of the inverse of the mixing matrix and the function $f(\cdot)$ is a non-linear sigmoid function. It has been shown that for $f(\cdot) = \tanh(\cdot)$ the algorithm performs very well for zero mean super-Gaussian input data. The unmixing matrix that is obtained this way will recover the original sources, but arbitrarily scaled. In addition the rows of the unmixing matrix might have a different ordering than the true inverse of the mixing matrix, so that:

$$\mathbf{W} \cdot \mathbf{A} = \mathbf{P} \tag{4}$$

Where $\mathbf{P}$ is a scaling and permutation matrix.

Similar work in instantaneous unmixing was also performed by Cichocki *et al.* (1994) and McKay (1996) who used different derivations. Additional approaches using feedback networks for inverting matrices have been also addressed by, Herault and Jutten (1991), Molgedey and Schuster (1994), Amari et al. (1995) as well as batch and algebraic methods proposed by Comon (1992), Cardoso (1993) and Hyvärinen and Oja (1997) but neither will be used in this paper. They are however applicable to the final algorithm presented and could be used to substitute instantaneous algorithms as used in subsequent sections.

2 CONVOLVED MIXTURES AND SEPARATION

Unfortunately instantaneous mixing is very rarely encountered in real-world situations, due to the extensive filtering imposed on sources by their environment, differences between sensors and propagation delays. Instead we observe *convolved mixtures*. To express this mixing process and maintain consistency with previous notation we will use *FIR Linear Algebra* notation (Lambert 1996). In FIR Linear Algebra matrices are composed of FIR filters instead of scalars and multiplication between two such FIR matrix elements is defined as their convolution. By implication the multiplication of two FIR matrices will involve a convolve and accumulate procedure replacing the dot products we would compute for ordinary matrices. FIR Filter matrices are notated as underlined matrices (e.g. $\underline{\mathbf{A}}$). To illustrate:

$$\underline{\mathbf{A}} \cdot \underline{\mathbf{B}} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & \dots & b_{1k} \\ \vdots & \ddots & \vdots \\ b_{n1} & \dots & b_{nk} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} a_{1i} * b_{i1} & \dots & \sum_{i=1}^{n} a_{1i} * b_{ik} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} a_{mi} * b_{i1} & \dots & \sum_{i=1}^{n} a_{mi} * b_{ik} \end{bmatrix} \quad (5)$$

Where $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ are m by n and n by k FIR matrices respectively, a_{ij} and b_{ij} are FIR filters and the star operator denotes convolution.

Using this notation we can express the convolved mixing case, which using scalar notation is:

$$x_i(t) = \sum_j \sum_k a_{ij}(k) s_j(t-k) \quad (6)$$

in a more elegant form as:

$$\mathbf{x}(t) = \underline{\mathbf{A}} \cdot \mathbf{s}(t) \quad (7)$$

Using a similar reasoning as in the instantaneous case, we can define an unmixing matrix $\underline{\mathbf{W}}$ and $\mathbf{y}(t) = \underline{\mathbf{W}} \cdot \mathbf{x}(t)$ to derive learning rules which are notationally identical to the ones in the previous section. We produce:

$$\Delta\underline{\mathbf{W}} \propto [\underline{\mathbf{W}}^{-1}]^T - 2 \cdot f(\mathbf{y}(t)) \cdot \mathbf{x}(t)^T \quad (8)$$

for the Bell rule, in equation (2), and

$$\Delta\underline{\mathbf{W}} \propto [\mathbf{I} - f(\mathbf{y}(t)) \cdot \mathbf{y}(t)^T] \cdot \underline{\mathbf{W}} \quad (9)$$

for the Amari rule, in equation (3).

Upon proper convergence the matrix $\underline{\mathbf{W}}$ should be such that:

$$\underline{\mathbf{W}} \cdot \underline{\mathbf{A}} = \underline{\mathbf{P}} \quad (10)$$

Where $\underline{\mathbf{P}}$ is a scaling, delay and permutation FIR matrix. That means that the output of the separation equation will be the original sources, arbitrarily scaled, permuted and delayed. In practice, for some algorithms, there is also some alteration of the spectral envelope of the original sources (spectral whitening).

Several variations of this approach have been developed by Bell (1996), Torkkola (1996) and even though they perform adequately they require significant computational power due to the convolutions that have to be executed. In addition to the computational power required, some approaches to this problem have the side effect of also whitening the spectrum of the output data, which sometimes acts as a local minimum that hinders convergence.

3 FREQUENCY DOMAIN SEPARATION

Various authors have used techniques that transform the data to the frequency domain, in order to perform training (Lee *et al.* 1997). In this paper we consider an different frequency domain approach to this problem. It is possible to transform the FIR filter mixing matrix, by performing a frequency transform on its elements, to an *FIR polynomial matrix*. FIR polynomial matrices are matrices whose elements are complex valued polynomials (Lambert 1996). Multiplication between FIR polynomial matrices is defined similarly as for normal matrices with the exception that the scalar multiplications become element-wise multiplications of the polynomials. The FIR polynomial matrix of $\underline{\mathbf{A}}$ will be notated as $\hat{\underline{\mathbf{A}}}$. Bearing these points in mind we can rewrite the mixing equation (7) as:

$$\hat{\underline{\mathbf{X}}} = \hat{\underline{\mathbf{A}}} \cdot \hat{\underline{\mathbf{S}}} \tag{11}$$

where the FIR polynomial matrices $\hat{\underline{\mathbf{X}}}$ and $\hat{\underline{\mathbf{S}}}$ contain the frequency transforms of **x** and **s** respectively, and $\hat{\underline{\mathbf{A}}}$ is the mixing matrix.

We can additionally break equation (11) down to:

$$\mathbf{X}_f(t) = \mathbf{A}_f \cdot \mathbf{S}_f(t), \quad \forall f, t \tag{12}$$

where $\mathbf{S}_f(t)$ and $\mathbf{X}_f(t)$ are vectors with one element for every source, which is the frequency transform element of s_i and x_i at frequency f and time t, and $\mathbf{A}_f$ is a matrix containing the elements of the frequency transforms of the mixing filters at frequency f †. This can also be interpreted as performing convolution in the frequency domain (using a short time Fourier transform) where element-wise multiplication of the frequency elements must be performed. By closer examination of equation (12) we notice a resemblance to equation (1). The frequency elements we observe from our sensors are in fact instantaneous mixtures of the original frequency elements of the sources.

To better illustrate this consider the FIR polynomial matrix $\hat{\underline{\mathbf{X}}}$ in the case of two sources. For every frequency and time it will contain a 2×1 complex valued vector with the frequency values of the two observable signals at that time. Each of these matrices will originate from a matrix multiplication of the corresponding frequency element matrices from $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ at the same time. That matrix product is a instantaneous mixing process.

Knowledge that the frequency bins of the observables are instantaneous mixtures of the original sources' frequency bins suggests that we can use the original instantaneous unmixing algorithms on every frequency track to achieve separation from convolved mixtures. To do so we need to reformulate these algorithms for complex-domain data (recall that the elements of the frequency polynomials are complex numbers).

In order to deal with complex-valued data we need to perform two modifications. First we must change matrix transpositions to hermitian transpositions (conjugate transpose). With this modification the learning rules, transform to:

$$\Delta\mathbf{W} \propto [\mathbf{W}^{-1}]^H - 2 \cdot f(\mathbf{y}(t)) \cdot \mathbf{x}(t)^H \tag{13}$$

for Bell's rule, and:

$$\Delta\mathbf{W} \propto [\mathbf{I} - f(\mathbf{y}(t)) \cdot \mathbf{y}(t)^H] \cdot \mathbf{W} \tag{14}$$

for Amari's rule.

The other important point is to choose a proper sigmoid function. The widely-used logistic and hyperbolic tangent sigmoids perform very poorly in the complex domain due to their singularities and are a bad choice in this case. Georgiou and Koutsogeras (1992) have developed a set of properties that complex-domain activation functions must fulfill. The most prominent one being that the function must be bounded. In the case of the hyperbolic tangent (and similarly the logistic), the function is undefined at $(k + 1/2)\pi i$, for $k = 0, 1, 2, \ldots$, which will introduce numerical problems and seriously hinder convergence (see figure (1)).

An alternative activation function for the complex domain is proposed which is defined as $f(z) = \tanh(Re\{z\}) + \tanh(Im\{z\}) \cdot i$ (shown in figure (2)). This function fulfills all complex activation function requirements set by Georgiou and Koutsogeras and performs very well in complex-domain networks. It can also be justified more intuitively by pointing out that this function is a better fit to the CDF of the data‡(assuming that the

†. It is also possible to have **A** being a time varying matrix to simulate moving sources, in which case the mixing equation includes a subscript t for the mixing matrix. Whether **A** is constant or not, the algorithm remains the same and for simplicity's sake we consider the static case.

‡. Bell (1996) points out that the hyperbolic tangent is a good approximation to the Gaussian CDF and that by mapping a Gaussian process through that function we flatten the outcoming PDF, thus maximizing information.

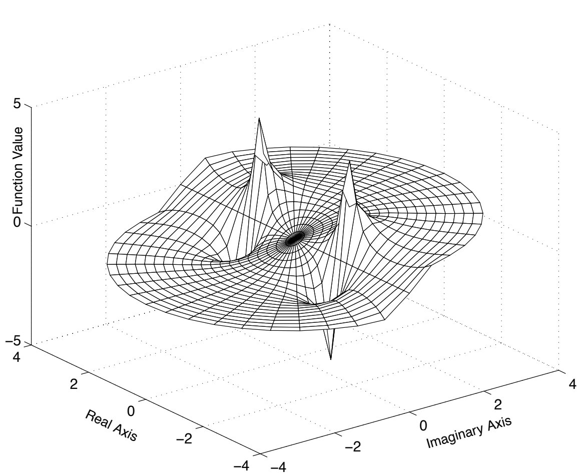

Figure 1: The hyperbolic tangent in the complex-number domain. Due to the singularities seen at $(k+1/2)\cdot\pi i$ it is an unsuitable activation function for training complex-domain networks.

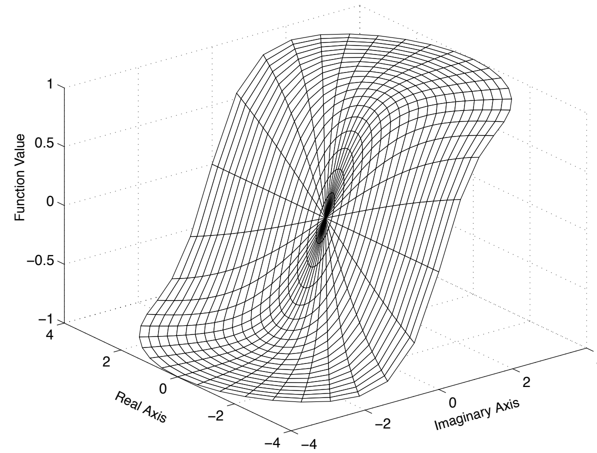

Figure 2: Proposed activation function for complex domain processing. It fulfills the required set of properties and provides a better fit to the CDF of complex Gaussian data.

frequency bin values exhibit Gaussian-like PDFs).

We form an algorithm which uses these modifications and apply it independently to each frequency track obtained from a Short Time Fourier Transform of the inputs. From this we will obtain one complex valued unmixing matrix for every frequency bin of the transform. The corresponding elements of these unmixing matrices will form the frequency responses of the filter elements of the unmixing FIR matrix. The algorithm is depicted in figure (3). In cases where the unmixing filters need to be long, there is the additional problem of the block delay introduced from the FFT based convolution. It is possible to modify the algorithm to perform zero delay FFT convolution to facilitate better real-time response (Gardner 1994).

One major advantage of this approach is that by performing convolution at the frequency domain we can improve computational performance by a large factor, especially for real world cases where separating filters should be long and there is limited computational power. To illustrate consider the 2 by 2 case where the separating filters contain 1024 taps and the sampling rate is 44.1 kHz. In the time domain adaptation and unmixing would require roughly 3.6×10^{8} flops versus 5.2×10^{6} using the frequency domain approach.

An additional advantage is the fact that we are performing adaptation in an orthogonal domain (the Fourier coefficients). This means that adaptation of one parameter will not interfere with the other parameters. This is extremely valuable in the case of long filters since for M taps long separating filters on the N sources problem we will require M instantaneous networks of size N that will be independent of each other. So we will be increasing the number of separating networks but their algorithmic complexity and convergence properties will remain the same (so for a two source problem we will have equally easy convergence whether we use any length of separating filters, the only difference will be in the required computational power which, as shown before, scales well for large filters).

Finally to deal with non-minimum phase mixing filters it is also possible to perform reordering of the time data to implement non-causal unmixing filters (Lee *et al.* 1997).

With the frequency domain approach there are however some extra complications. The algorithms shown in the first section and used to separate the frequency bins are invariant to scaling and permutation (equation (4)). The scaling invariance means that the scaling of every frequency band can be different, which will of course result in spectral deformation of the original sounds. This problem can be remedied by forcing the determinant of the unmixing matrices to unity by:

$$\mathbf{W}_f^{norm} = \mathbf{W}_f^{orig} \cdot \left|\mathbf{W}_f^{orig}\right|^{-\frac{1}{N}} \tag{15}$$

where $\mathbf{W}_f^{norm}$ and $\mathbf{W}_f^{orig}$ are respectively the normalized and original $N\times N$ unmixing matrices at frequency f. This ensures volume conservation for every unmixing matrix and an almost unaltered spectral envelope, while preserving the separation.

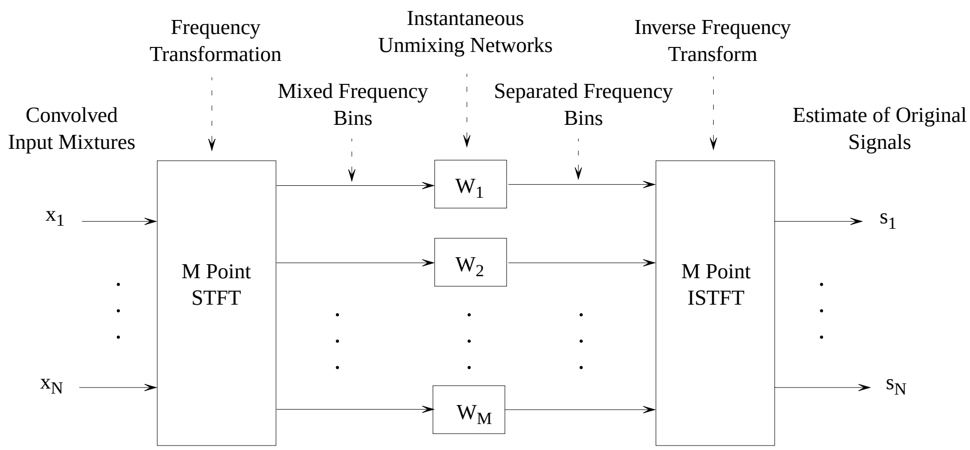

Figure 3: Flow diagram of the frequency domain algorithm.

The permutation invariance is a more difficult problem which is still open to a satisfying and rigorous solution. Fortunately this problem is not evident in all cases. For simple mixing filters all unmixing matrices converge to the same permutation. Using more complicated filters however problems might rise. In such cases careful selection of the adaptation parameters in addition to decaying learning rate and momentum are very helpful since they can eliminate random permutation changes in training and preserve the same permutation throughout. The required parameter values do however delay convergence by a serious amount. Another important parameter is adequate zero padding when performing the frequency transformation. Since zero padding before the frequency transformation results in smoother spectra, we ensure that contiguous frequency values of the inputs will be close to each other. That way convergence over the frequency bins will occur on similar surfaces and will eventually lead to the same permutations given the similarity of the inputs for neighboring unmixing matrices. In order to enforce this further we can use an *influence factor* for the update of every bin so that:

$$\Delta_a \mathbf{W}_{f+1} = \Delta_e \mathbf{W}_{f+1} + k \cdot \Delta_e \mathbf{W}_f \tag{16}$$

or

$$\Delta_a \mathbf{W}_{f+1} = \Delta_e \mathbf{W}_{f+1} + k \cdot \Delta_a \mathbf{W}_f \tag{17}$$

where $\Delta_a \mathbf{W}_f$ and $\Delta_e \mathbf{W}_f$ are respectively the applied and estimated weight updates for the unmixing matrices at frequency f, and $0 < k < 1$ is the influence factor. Although this is a simple heuristic it seems to work fine under many situations (but not in very complicated problems).

4 RESULTS

The algorithm was evaluated with synthetic tests on three different situations: instantaneous mixing, delayed mixing and finally convolved mixing. The inputs were two speech signals from news broadcasts at a sampling rate of 22050 Hz. The learning rule used was equation (14) with the incorporation of learning rate and momentum. On-line training was performed, only once through the training data in real-time. After training, the unmixing matrices for all frequency bins were collected to form the unmixing FIR matrix.

To evaluate performance the estimated unmixing FIR matrix was multiplied with the FIR mixing matrix to obtain the *performance FIR matrix*. This matrix indicates how well separation was achieved. On perfect separation its diagonal elements are impulse responses and the remaining elements are zero. In order to obtain a better sense of performance the frequency transforms of the performance matrix elements are shown in the figures. Because of the relatively sparse high frequency content in the training data the corresponding high-frequency weights are not well trained and as a result we observe the formation of highpass filters in place of zero elements in the performance matrices. This however does not imply poor performance since there is little high frequency input to separate anyway.

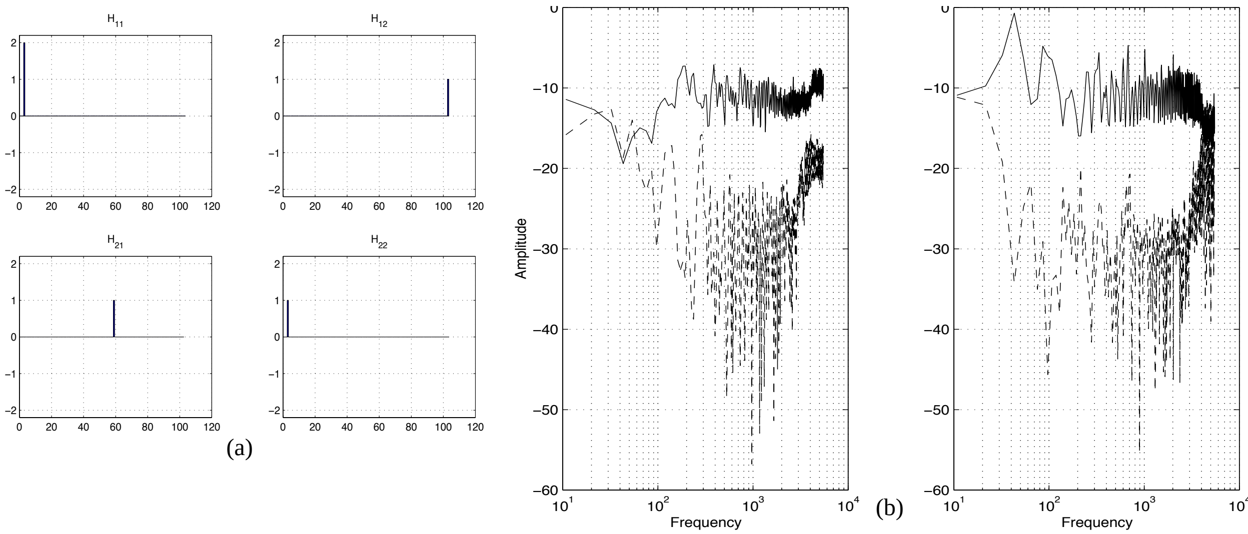

Figure 4: Performance matrix for delayed mixtures. Subplot (a) shows the elements of the mixing FIR matrix. A sense of the unmixing performance is given in subplot (b) where we examine the outputs by looking at the frequency domain. The solid line on top is the response of the unmixing system to one source at the corresponding output and the dashed line under it is the response to the interfering source at the same output. On average the responses of the desired source vs. the interfering source differ by 20dB of attenuation.

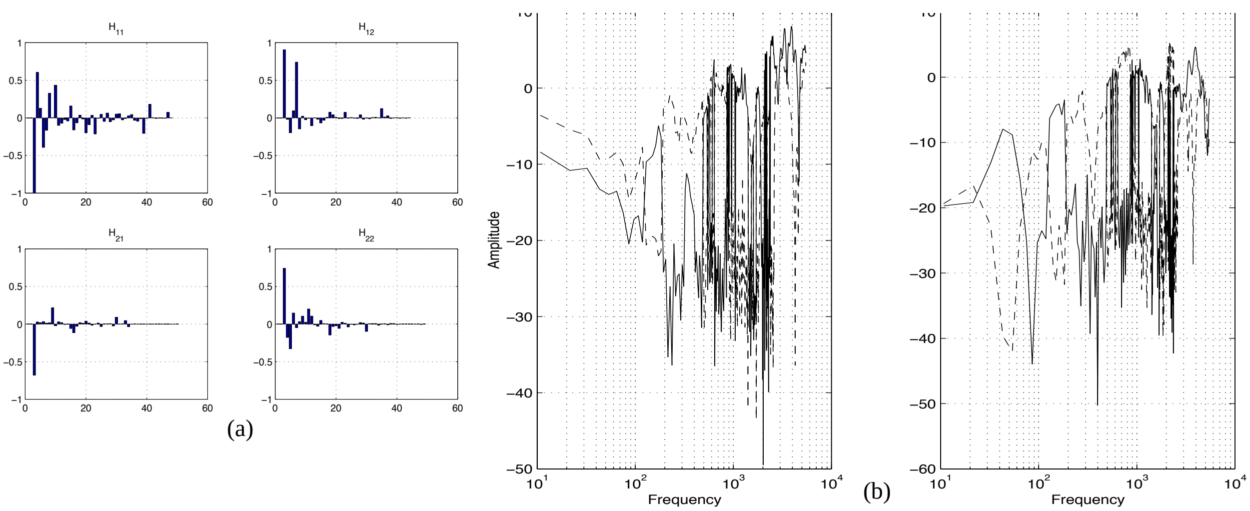

Figure 5: Performance matrix for convolved mixtures. The FIR mixing matrix, containing decaying Cauchy noise, is shown in subplot (a) and the performance FIR matrix in subplot (b). Note in subplot (b) that for many frequency regions the permutation of the unmixing matrices is incorrect. This is due to badly selected adaptation parameters. Also note the seemingly poor performance at the high frequency regions. Since the training data were speech signals there was little frequency content in these regions which resulted in underconstrained training. This is however not a problem given that there is no excitation at these levels. Excluding permutation problems for the frequency range of the inputs (100 Hz - 1000 Hz) the algorithm works fine.

The first example was set to examine if the algorithm worked at all. To do so we used the instantaneous mixing matrix:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \tag{18}$$

Results were just as good as for instantaneous separation algorithms and the interfering sources were successfully suppressed by more than 30dB. Convergence was of course a slower than instantaneous algorithms since training data was presented once for every frequency transform rather for every sample. In order to speed convergence time the overlap of the frequency transformations was increased (by a reasonable amount) and it was possible though to get sufficient separation in about two seconds of

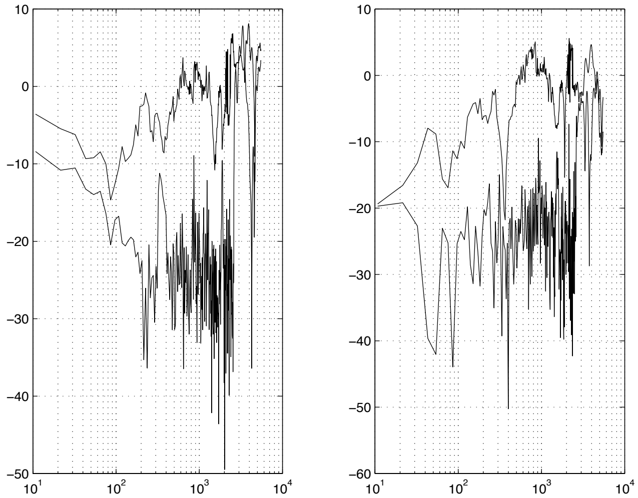

Figure 6: Performance matrix for Cauchy noise mixing matrix shown in figure (5). This time better parameter selection took place in addition to use of the influence factor. It is shown that separation is good averaging 20dB of interference suppression

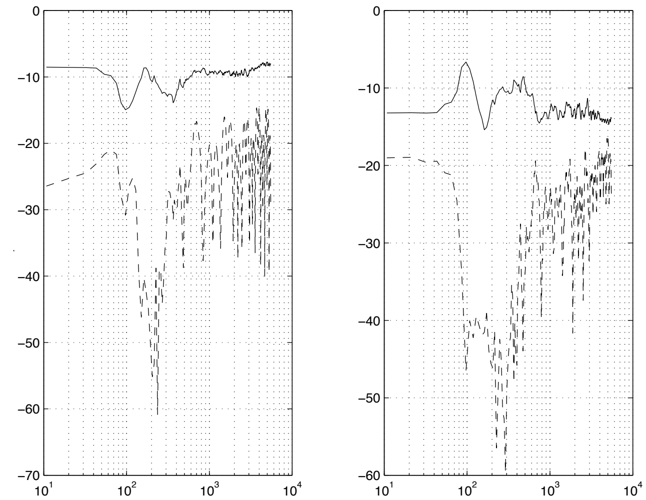

Figure 7: Results from an additional case of convolved mixtures using decaying Cauchy noise mixing filters.It is clear that separation is very good (up to 30 dB) especially at the frequency band that the inputs dominated (100 Hz - 1000 Hz).

sound.

The second example was setup to see how the algorithm performs with delayed mixtures, which is the simplest form of convolved mixing. In this example we used the mixing FIR filter matrix depicted in figure (4.a). The diagonal elements are delays of 56 and 100 samples; with the exception of the first element which has a value of 2 all others have a value of 1. The performance matrix in the frequency domain is shown in figure (4.b). Interference was suppressed to inaudible levels, very quickly, with the exception of occasional loud consonants.

Afterwards a more complex problem was set up. The sources were mixed with filters which had exponentially decaying Cauchy noise as impulse responses. Such filters are good for emulating room responses since they exhibit a few loud echoes, perceived as early reflections, and many more faint echoes which simulate the room ambience. The mixing matrix is shown in figure (5.a) and the corresponding results in figures (5.b) and (6). In the first figure we have a case where the unmixing matrices have permutation disparities. In figure (6) the same mixture was separated with more conservative adaptation parameter selection and by using the influence factor described in the previous section. Resulting separation was almost inaudible in this case too.

The same problem was setup again only this time the mixing filters were non-minimum phase which necessitated the use of non-causal unmixing filters. Results in that case were also very satisfying and are shown in figure (7).

All experiments were done with separating filters of 1024 taps, a zero padding factor of 2 and an STFT hop size of 512 points. For the last example a zero padding factor of 4 was used to facilitate the influence factor. In addition to this the last example also used non-causal unmixing filters.

5 CONCLUSIONS

It was shown that the problem of separating convolved mixtures can be approached using techniques developed for separating instantaneous mixtures. The algorithm developed is considerably more efficient than its time domain counterparts, with a complexity of $N \cdot \log N$ vs. N^2, that permits implementations that can run in real time on personal computers. In addition to efficiency improvements, convergence properties are very attractive. Due to the fact that the filter parameters in the frequency domain are orthogonal to each other, updating one parameter does not influence the rest. This is not the case in time domain adaptation where filter parameters are linearly dependent. This independence allows us to perform adaptation using longer filters without complicating convergence or introducing new potential minima.

It should also be stressed that the performance of this approach is very dependent on the algorithm that is being used to separate the frequency coefficients. For the examples in this paper the modified version of the learning rule by Amari *et al*. (1996) was used (equation (14)). Performance of this approach in a more complicated situation, such as one featuring more sources, or the case where we have more inputs than outputs, is highly dependent on the characteristics of the instantaneous unmixing algorithm that is used. Also dependent to this feature is the ability to deal with sub or super Gaussian inputs (although note that separation takes place in the frequency domain and that we should be observing the PDF of the frequency bins rather than the one of the time domain signal).

However, this algorithm is not foolproof. Problems appear with dense and badly conditioned filters where the unmixing matrices of different frequency tracks are led to converge to different permutations. Although there are heuristics that can be applied to avoid this problem they are not always successful and further work to provide alternative solutions is required. Incorporation of other instantaneous separation algorithms to separate the frequency bins would be of benefit since the convergence characteristics of such approaches might eliminate current problems. The advantages of this approach are certainly attractive enough to encourage such work.

Acknowledgments

The author would like to thank the Machine Listening group at the MIT Media Lab as well as V. Michael Bove and Kari Torkkola for useful discussions and comments.

References

Amari, S-I., A. Cichocki and HH. Yang. 1995. "Recurrent Neural Networks for Blind Separation of Sources". In *Proceedings 1995, International symposium on Nonlinear Theory and Applications NOLTA 95*, Vol. **5**, pp. 37-42. Tokyo, Japan.

Amari, S-I., A. Cichocki, and H.H. Yang. 1996. "A new learning algorithm for blind signal separation". In *Advances in Neural Information Processing Systems **8***. MIT Press, Cambridge, MA.

Bell, A.J. and T.J. Sejnowski. 1995. "An information maximization approach to blind separation and blind deconvolution". In *Neural Computation **7***. pp. 1129-1159. MIT Press, Cambridge, MA.

Cardoso, J-F. 1993. "Blind beamforming for non gaussian signals". In *IEE-Proceedings-F*, vol. **140**, no **6**, pp. 362-370.

Cichocki, A., R.Unbehauen and E. Rummert. 1994. "Robust Learning Algorithm for Blind Separation of Signals". In *Electronics Letters*, vol. **30**, No. **17**, pp. 1386-1387.

Comon, P. 1992. "Independent component analysis, a new concept?" In *Signal Processing* **36**, pp.287-314.

Gardner, W. G. 1994. "Efficient convolution without input-output delay". In *Journal of Audio Engineering Society **43 (3)***, pp. 127-136.

Herault, J. and C. Jutten. 1991. "Blind Separation of Sources, Part I: An adaptive Algorithm based on Neuromimetic Architecture". *Signal Processing **24***, pp. 1-10.

Hyvärinen, A. and E. Oja. 1997. "A fast fixed-point algorithm for independent component analysis". In *Neural Computation* **9**, pp. 1483-1492, MIT Press Cambridge, MA.

Lambert, R. H. 1996. "Multichannel blind deconvolution: FIR matrix algebra and separation of multipath mixtures". Ph.D. dissertation, University of Southern California, EE dept. May 1996.

Lee T-W., A.J. Bell and R. Lambert. 1997. "Blind separation of delayed and convolved sources". In *Proceedings of NIPS*9*.

McKay, D. 1996. "Maximum Likelihood and Covariant Algorithms for Independent Component Analysis". *Draft paper* available at: `ftp://wol.ra.phy.cam.ac.uk/pub/mackay/ica.ps.gz`

Molgedey L. and H.G. Schuster (1994). "Separation of a mixture of Independent Signals using Time Delayed Correlations". In *Physical Review Letters **72***, pp. 3634-3637.

Torkkola, K. 1996. "Blind separation of convolved sources based on information maximization". In *Neural Networks for Signal Processing **VI***. Kyoto Japan, IEEE press.