NEW FIXED-POINT ICA ALGORITHMS FOR CONVOLVED MIXTURES

N. Mitianoudis, M. Davies

King's College, London
Strand
WC2R 2LS London, UK

ABSTRACT

One of the most powerful techniques applied to blind audio source separation is Independent Component Analysis (ICA). For the separation of audio sources recorded in a real environment, we need to model the mixing process as convolutional. Many methods have been introduced for separating convolved mixtures, the most successful of which require working in the frequency domain [1], [2], [3], [4]. Most of these methods perform efficient separation of convolved mixtures, however they are relatively slow. The authors propose two fixed-point algorithms for performing fast frequency domain ICA.

1. INTRODUCTION

Suppose we have N discrete audio sources $s_i[n]$. We produce M observation signals $x_i[n]$, by calculating M linear combinations of the N audio sources. The whole procedure can be modeled by the following equation:

$$\underline{x}[n] = A\underline{s}[n] \qquad (1)$$

where $\underline{s}[n]$ is a vector representing the audio sources, $\underline{x}[n]$ is a vector representing the observed signals and A is the *mixing matrix*. The problem of blind source separation is defined as the procedure of calculating an unmixing matrix W, using information retrieved from the observation signals $\underline{x}[n]$, so as to separate the original audio sources, using the formula:

$$\underline{u}[n] = W\underline{x}[n] \qquad (2)$$

It is clear that in order to perform separation, the unmixing matrix W should approximate A^{-1}. Moreover, in order to simplify our analysis, we assume that the number of observed signals M is equal to the number of input sources N. We also ignore any additive noise present during the mixing procedure of the original sources.

Many techniques have been applied to solve this mathematical problem. ICA methods estimate the unmixing matrix W, exploiting the non-gaussianity of audio signals, as well as the statistical independence of the separated signals $\underline{u}$. As measures of non-gaussianity, some ICA methods employ higher-order moments *(kurtosis)* or negentropy, which measures the distance from the gaussian distribution [7], [8], [9]. Other methods try to separate the audio sources by minimizing the mutual information conveyed by the separated sources [5]. Others employ maximum likelihood methods, imposing probabilistic priors to model the sources [6], [7].

All these methods separate instantaneous mixtures. However, if we try to apply these techniques on observation signals acquired from microphones in a real room environment, we will see that all actually fail to separate the audio sources. This is mainly because we didn't take into account the room acoustics in the previous mixing model. In a real recording environment, sensors (microphones) record delayed, attenuated versions of the source signals, apart from direct path signals, due to reflections. However, the observation signals cannot be regarded as linear combinations of the source signals and can be modeled as follows:

$$x_i[n] = \sum_{j=1}^{N}\sum_{k=1}^{L} a_{jk} s_j[n-k], \quad \mathrm{i} = 1,\mathrm{N} \qquad (3)$$

where L denotes the maximum delay in terms of discrete points. Looking at the equation above, we can see that it is actually the summation of the *convolution* of the N sources with N filters of maximum length L. Applying common ICA methods to convolved mixtures, we can possibly achieve separation of direct path signals only, leaving the reflections untouched. As a consequence, a new approach has to be established.

2. PREVIOUS WORK ON FREQUENCY DOMAIN ICA

In order to solve the problem of convolution, Smaragdis [1] [2] proposed applying a STFT to the mixture signals $\underline{x}[n]$, using windows of greater length than L, and work in the frequency domain. The motivation behind moving to the frequency domain is that the discrete fourier transform can turn the convolution into multiplication. As a consequence, the whole separation problem is divided into N linear complex source separation problems, one for every frequency bin. There are many ICA methods to perform source separation of linear mixtures. Smaragdis [1] [2] applied the natural gradient ICA algorithm [6] to complex source separation, using a complex, non-linear activation function *φ(u)*.

$$\Delta W(\omega) = \delta(I - \varphi(\underline{u})\underline{u}^H)W(\omega) \qquad (4)$$

where δ denotes the learning rate.

One inherent ambiguity in all ICA methods is the *permutation* problem. Permutation problems are of minor importance in the instantaneous mixtures case. However, it's absolutely essential to keep the same permutation in the frequency domain ICA, so as not to end up with signals with mixed frequency content. Smaragdis proposed a heuristic coupling of adjacent frequency bins, noting that it didn't prove to be very effective.

Davies [3] introduced a time-frequency model to solve the permutation problem. This is performed by adding a time dependent β(t) term to the frequency model of the separated sources.

$$\beta_k(t) = \frac{1}{N}\sum_{\omega} |u_k(\omega,t)| \qquad (5)$$

The β(t) term can be interpreted as a time average over frequency. In other words, it measures the overall signal amplitude along the frequency axis. Its main purpose is to impose frequency coupling between frequency bins. Incorporating this term in the frequency model alters the natural gradient algorithm as follows:

$$\Delta W(\omega) = \delta(I - \beta(t)^{-1}\varphi(u(\omega,t))u(\omega,t)^H)W \qquad (6)$$

$$\beta(t) = diag(\beta_1(t), \beta_2(t), ...\beta_N(t)) \qquad (7)$$

where φ(u) is a nonlinear complex activation function. Assuming laplacian priors for the sources, we can use the following activation function [3]:

$$\varphi(u) = u/|u|, \quad \text{for all u} \neq 0 \qquad (8)$$

There are many methods proposed to overcome the permutation problem. Davies [3] applies a likelihood ratio jump solution. This technique compares the likelihood of the unmixing matrix W with that of [0 1; 1 0] W. For the 2x2 case, we calculate LR using the following formula and if LR<1, we have to permute W.

$$LR = \frac{\gamma_{12}\gamma_{21}}{\gamma_{11}\gamma_{22}} \qquad (9)$$

where

$$\gamma_{ij} = \sum_{t=1}^{T}\frac{|u_i(t)|}{\beta_j(t)} \qquad (10)$$

This method tends to sort out the permutation problem in the majority of the cases. However, it gets rather complicated to form LR expressions in the general NxN case.

In this paper, we present two efforts to replace the natural gradient algorithm with a fixed-point algorithm in the frequency domain ICA framework

3. THE FIRST FIXED POINT SOLUTION

Hyvarinen et al proposed a family of fixed point ICA algorithms for performing ICA of instantaneous mixtures [7] [8] [9]. Their basic feature is that they converge much faster than gradient descent algorithms with the same separation quality. Nevertheless, they are more computationally expensive, but as the number of iterations for convergence is much decreased, they tend to be faster then common ICA techniques. In addition, they tend to be much more stable.

In [8], Hyvarinen explored the relation of his fixed-point algorithm with the natural gradient algorithm [6]. The fixed-point algorithm is basically a deflation algorithm, isolating one independent component every time. It employs a decorrelation scheme to prevent the algorithm converging to the same maximum. The one-unit learning rule for the fixed-point algorithm is the following:

$$\underline{w}^{+} \leftarrow C^{-1}E\{\underline{x}\varphi(\underline{w}^T\underline{x})\} - E\{\varphi'(\underline{w}^T\underline{x})\}\underline{w} \qquad (11)$$

where $C = E\{\underline{x}\underline{x}^T\}$ and $\varphi(\underline{u})$ is an non-linear activation function. Making certain assumptions on x, Hyvarinen shows that the learning rule in (11) can be represented by the following learning rule:

$$W^{+} \leftarrow W + D[diag(-\alpha_i) + E\{\varphi(\underline{u})\underline{u}^T\}]W \qquad (12)$$

where $\alpha_i = E\{u_i\varphi(u_i)\}$, $D = diag(1/(\alpha_i - E\{\varphi'(\underline{u})\}))$ and $W = [\underline{w}_1\ \underline{w}_2 \ldots \underline{w}_N]$. Comparing equations (12) and (4), we can see that the two methods look very similar. Actually, we can say that (12) is a more adaptive version of (4). We apply an optimal step size in terms of *D,* instead of a constant learning rate δ. Hyvarinen [9] states that replacing I with the term *diag(-α_i)* is also beneficial for convergence speed. If we use pre-whitened data x, then the formula in (14) is equivalent to the original fixed-point algorithm, while it is expressed in terms of the natural gradient algorithm.

This algorithm works efficiently on instantaneous mixtures, providing accurate separation with faster convergence, compared to the natural gradient algorithm. In this paper, we wish to replace the natural gradient algorithm with the fixed-point algorithm, as described in (12), in the frequency domain framework, so as to accelerate the convergence of audio separation algorithms.

More specifically, we are going to divide our observation signals into overlapping windowed frames, and apply STFT on them, forming a time-frequency representation x(ω,t). We pre-whiten x(ω,t) before proceeding. The next step is to estimate the unmixing matrix for every frequency bin. This is achieved by iterating the following learning rule, using random initial value for W.

$$W^{+} \leftarrow W + D[diag(-\alpha_i) + E\{\varphi(\underline{u})\underline{u}^H\}]W \qquad (13)$$

All the parameters in the equation above are calculated as discussed earlier. However, we should pay attention to the choice of the activation function φ(u). A proper activation function for the processing of complex data is (8), as introduced by Davies [3]. At this point, we should note that the discontinuity of the activation function φ(u) at u=0 doesn't appear to cause any problems. By differentiating, we get the derivative of *φ*:

$$\varphi'(u) = |u|^{-1} - u^2|u|^{-3}, \quad \text{for all u} \neq 0 \qquad (14)$$

Another important factor in frequency domain ICA is the permutation problem. In order to solve the permutation problem in this case, we can follow a method similar to the one described earlier. Firstly, we enhance frequency coupling by incorporating a time dependent *β(t)* term to the frequency model of the separated sources, as we did in the natural gradient method. If we look at the maximum likelihood learning law at (6), we can see that the *β(t)* term can be incorporated in the activation function *φ(u)*. Therefore, in order to impose frequency coupling in the fixed-point algorithm in (13), we can use the following activation function:

$$\varphi(u) = \frac{u}{\beta(t)|u|} \qquad (15)$$

As a second step, we can use the likelihood ratio jump solution, as presented in (9), (10), in order to get the same permutation of the separated sources for every frequency bin.

Another important task when performing frequency domain ICA is to return the separated signals $\underline{u}$ to their original space (represented by $\underline{x}$ vectors). More specifically, if W_f is the unmixing matrix for the frequency bin f, we can write:

$$x_i s_j(f,t) = W_{f_{ij}}^{-1} u_j(f,t)\text{, for i ,j = 1…N} \quad (16)$$

Note that for pre-whitened sources, we also need to return the sources to the original space before pre-whitening. Suppose that V_f is the pre-whitening matrix for each frequency bin. We have:

$$[x_1 s_j \; \ldots \; x_N s_j]^T = V_f^{-1}[x_1 s_j \; \ldots \; x_N s_j]^T \text{ ,j=1...N} \quad (17)$$

After performing all the essential linear transformations, we can group the $x_i s_j$ signals to form the separated outputs as follows:

$$\tilde{u}_j(f,t) = \sum_i x_i s_j(f,t)\text{, for j = 1…N} \quad (18)$$

The fixed-point frequency domain algorithm is summarised as follows:

1. Pre-whiten input data
2. Incorporate β(t) function in the activation function, i.e. use formula (15)
3. For the derivative of (15), use (14) as an approximation
4. Use the learning rule presented in (13), to estimate the unmixing matrices for every frequency bin.
5. Return separated signals to the observation space, as well as re-decorrelate separated signals. Finally, group the corresponding $x_i s_j$ signals.

4. THE SECOND FIXED POINT SOLUTION

In [10], Bingham et al proposed a "fast" fixed-point algorithm for independent component analysis of complex valued signals. This algorithm is designed to separate instantaneous mixtures of complex data, providing fast convergence as well as great separation quality. It's an extension of the FastICA algorithm [7], [8] to complex signals. The difference between the two fixed-point algorithms lies in the different contrast function employed in the optimisation problem. In the first fixed-point algorithm, the contrast function is $G(w^H x)$, where as in the second fixed-point algorithm the contrast function is $G(|w^H x|^2)$, where $\varphi(u) = dG(u)/du$.

The algorithm proposed is a deflation algorithm separating one independent component at a time. When we need to calculate a new component, we can prevent the algorithm from converging to the same stationary point by using a decorrelation scheme. The proposed fixed-point algorithm is summarized in the following formula:

$$\begin{aligned} \underline{w}^+ \leftarrow & E\{\underline{x}(\underline{w}^H \underline{x})^* \varphi(|\underline{w}^H \underline{x}|^2)\} - \\ & - E\{\varphi(|\underline{w}^H \underline{x}|^2) + |\underline{w}^H \underline{x}|^2 \varphi'(|\underline{w}^H \underline{x}|^2)\}\underline{w} \end{aligned} \quad (19)$$

$$\underline{w}^+ \leftarrow \underline{w}^+ / \| \underline{w}^+ \| \quad (20)$$

where $\varphi(\underline{u})$ is an activation function. Instead of calculating every independent component separately, it's preferable for many applications to calculate all components simultaneously. We can use different one-unit algorithms (19) for all independent components and apply a symmetric decorrelation to prevent the algorithms from converging to the same components. This can be accomplished by using a symmetric decorrelation :

$$W \leftarrow W(W^H W)^{-1/2} \quad (21)$$

where $W = [\underline{w}_1 \; \underline{w}_2 \ldots \underline{w}_N]$ is the matrix of the vectors $\underline{w}_i$.

Bingham proposes a set of activation functions that can be applied to this fixed-point algorithm. As we can see the problem involves real data, therefore it is easier to choose an activation function. From the set of the proposed activation functions, we are going to use the following:

$$\varphi(u) = 1/(0.1+u) \quad (22)$$

The derivative of the above is:

$$\varphi'(u) = -1/(0.1+u)^2 \quad (23)$$

This method achieves fast and accurate separation of complex signals. In this paper, we would like to adapt this method to a frequency-domain separation framework. The main advantage of this algorithm is that it performs separation of complex-valued mixtures, being therefore easier to adapt directly in a frequency domain framework.

In other words, the observation signals are transformed into a time-frequency representation using a Short-Time Fourier Transform. As before, we prewhiten the $\underline{x}(\omega,t)$. Then, we have to calculate the unmixing matrix W_f for every frequency bin. We randomly initialize N learning rules, as described in (19) and (20) for every frequency bin and iterate until convergence. However, there is nothing in this algorithm to tackle the permutation problem explained earlier.

We can solve the permutation problem firstly, by incorporating the time dependent prior *β(t)* in the learning rule, in order to impose frequency coupling. As we have seen in [3], the *β(t)* term can be actually integrated in the activation function *φ(u)*. In section 3, we saw that Hyvarinen transformed the basic fixed-point algorithm to a form that was similar to the natural gradient algorithm and we gathered that we could incorporate *β(t)* in the activation function *φ(u)* of the fixed-point algorithm, so as to impose frequency coupling. This is the main motivation behind incorporating the *β(t)* term in the activation function of the second fixed-point algorithm. Therefore, equations (22) and (23) are now transformed in the following form.

$$\varphi(u_k) = 1/(\beta_k(t)(0.1+u_k)) \quad (24)$$

$$\varphi'(u_k) = -1/(\beta_k(t)(0.1+u_k)^2) \quad (25)$$

where $\beta_k(t)$ refers to the corresponding separated component u_k, as introduced in (5).

The second step is to apply the likelihood ratio jump solution, described in (9), (10), so as to keep the same source permutation along the frequency axis. The likelihood ratio jump solution can be directly applied to the second fixed-point algorithm, without any adaptation. It is also worth noting that the β(t) term doesn't have a strong probabilistic interpretation as in the first fixed-point solution and in [3].

The next step would be to return the separated sources to the observation space $\underline{x}$, by using a formula slightly different to the one described in (16). This is because Hyvarinen defines the unmixing procedure as $\underline{u} = W^H\underline{x}$.

$$x_i s_j(f,t) = (W_{f_{ij}}^H)^{-1} u_j(f,t), \text{ for } i,j = 1\ldots N \quad (26)$$

We also need to remove the effects of prewhitening, by using the formula described in (17) and also sum also the components to construct the separated sources, as described in (18).

The second fixed-point frequency domain algorithm can be summarised as follows:

1. Pre-whiten input data
2. Initiate N one-unit learning procedures, one for each component, with random initialisation.
3. Incorporate each component's $\beta_k(t)$ in the corresponding learning rule, according to (24) and (25).
4. Decorrelate the separate outputs in every iteration using (21).
5. Return separated signals to the observation space, as well as remove prewhitening effects on separated signals, according to (26), (17), (18).

5. EXPERIMENTS

In order to record the performance of the proposed algorithms, we tested them using various data sets. For the whole set of experiments, the specifications of the STFT were a frame size of 2048 samples with 50% overlapping using a hamming window. The FFT length was 2048 points.

Initially, we applied the algorithms on some real data available from [11] of two people speaking simultaneously in a room, as they are commonly used in ICA benchmarks. Our first conclusion is that the fixed-point algorithm takes about 40-50 iterations to converge, which is much faster compared to common maximum likelihood algorithms. The second fixed-point algorithm converges in about the same number of iterations with the fixed-point algorithm. Commonly, the solutions proposed by Davies [3] and Smaragdis [1] [2] require usually about 200–300 iterations to converge for the same quality of separation. Convergence speed has become a quite important factor for frequency domain ICA, as previous approaches required considerable time to run. As far as the separation quality is concerned, we can say that you can hear almost no cross-talk.

The difference between the two fixed-point algorithms is that the second fixed-point is a little bit faster as some frequency bins converge faster than others.

In order to demonstrate the separation quality of the algorithm, we constructed a synthetic mixture of two speech signals. The mixtures contained delayed components of 25ms maximum, as well as the direct path signals. Both the fixed-point and the second fixed-point algorithms managed to separate the input sources quite well. We can see the spectrograms of the original and separated sources in figures 1,2.

The separation quality is quite good, and almost all the harmonic components of the original sources are preserved in the separated outputs. We can clearly see that there are no permutation problems visible or audible in these spectrograms. The permutation problems are well described in [3], where it is shown that although some algorithms perform reasonable separation for every frequency bin, we can see source permutation changes at certain frequencies. As a result, each source estimate contains large proportions of both sources which are both audible.

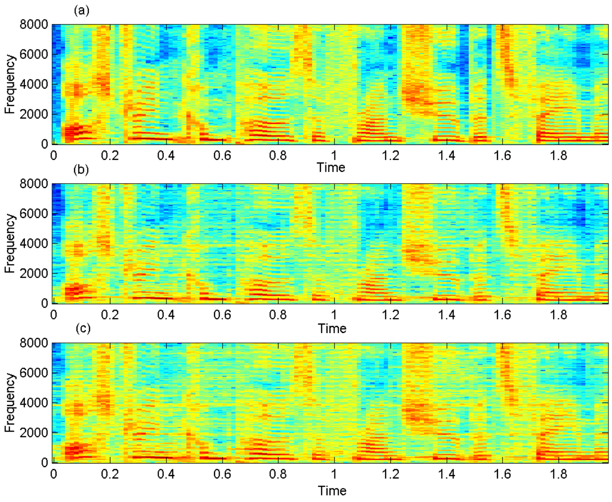

Figure 1. (a) Spectrogram of the original source and spectrogram of the separated source using (b) the fixed-point algorithm and (c) the second fixed-point algorithm

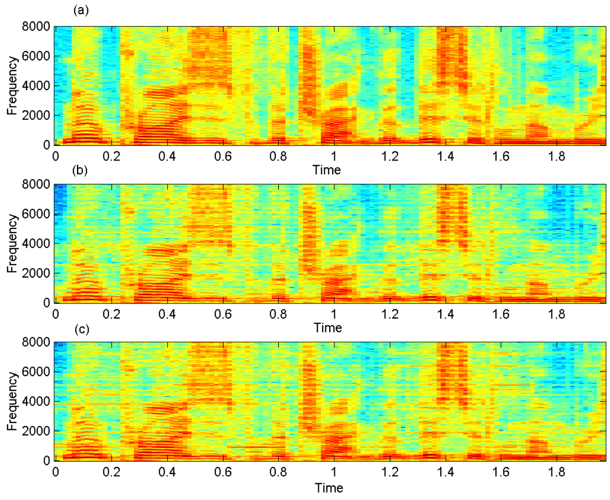

Figure 2. (a) Spectrogram of the original source and spectrogram of the separated source using (b) the first fixed-point algorithm and (c) the second fixed-point algorithm

We can further test the algorithms' performance on the permutation problem, using the dataset introduced in [3] that demonstrated permutation problems in Smaragdis's algorithm. In figure 3, we can see the spectrogram of one of the sources, separated by the fixed-point and the second fixed-point algorithm. We can see no changes in permutation along the frequency axis, which implies that the time-frequency model and the likelihood ratio jump solution are efficiently incorporated in the algorithms. As far as separation quality is concerned, the methods seem to have separated the signals quite successfully preserving all the harmonic structures. We can also spot almost no difference in the performance of the two algorithms.

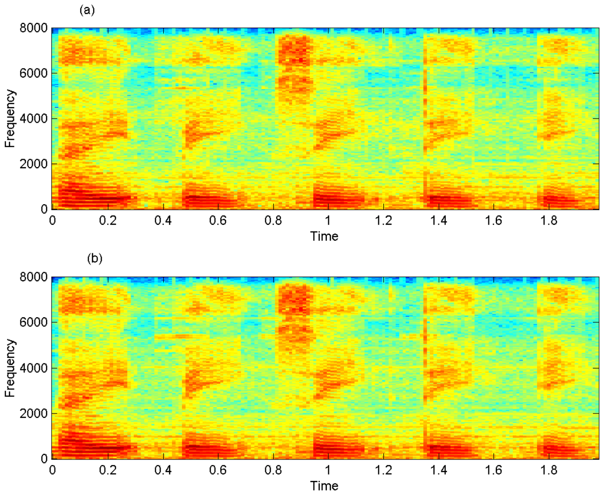

Figure 3. Spectrogram of a separated source using (a) the first fixed-point, and (b) the second fixed-point algorithm. Refer to the dataset in [3].

We also wanted to test the algorithms with a more challenging task of instrument separation. We recorded two guitars playing triads in unison and created a synthetic delayed mixture adding a tap delay of 25ms to each source. This is a quite highly correlated mixture as the two guitars are playing notes in unison, making it even difficult for the human ear to separate.

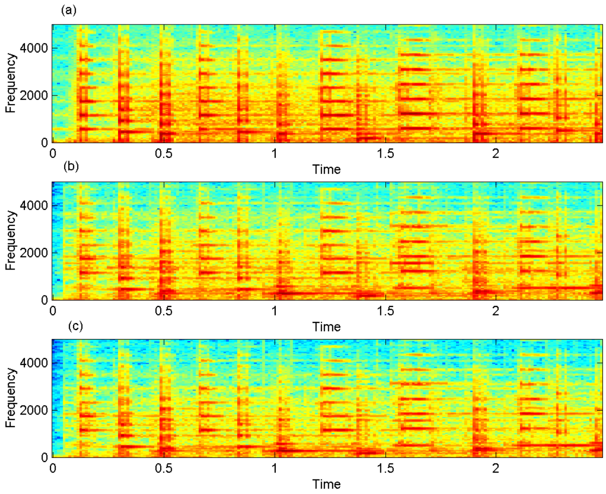

Figure 4. (a) Spectrogram of the original guitar source and spectrogram of the separated guitar source using (b) the first fixed-point algorithm and (c) the second fixed-point algorithm

However, the results are very good, as presented in figure 4. These highly correlated signals are well separated by the two algorithms, with the second fixed-point being a little bit more robust this time, with negligible crosstalk in the background.

Finally, we wanted to test these algorithms in a difficult reverbant environment. Therefore, we used Westner's [12] MATLAB routine *roommix.m*, which simulates the room acoustics of a conference room. Generally, the mixtures generated by these models are highly reverbant and difficult to separate. We constructed two mixtures using the following MATLAB expression, which defines the position of sources and sensors in the conference room:

```
[x,f]=roommix(x,[1 2],[2 1])
```

We had to re-adjust the STFT settings to a frame of 4096 samples with 75% overlapping, so that the algorithms can cope with greater tap delays. The FFT length was 4096 points. The results acquired were quite promising, although not perfect. The second fixed-point algorithm managed to perform better separation from the first fixed-point algorithm, suppressing the crosstalk to a considerable amount, as depicted in figure 5. In order to evaluate the algorithm's performance, we had to compare the separated outputs with each of the original signals simulated alone in the synthetic room environment.

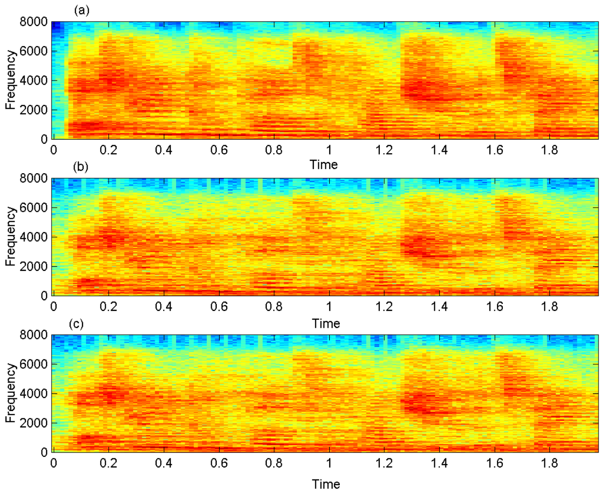

Figure 5. (a) Spectrogram of the original source in the simulated room environment and spectrogram of the separated source using (b) the first fixed-point algorithm and (c) the second fixed-point algorithm

6. CONCLUSIONS

In this paper, we have introduced two fixed-point algorithms for frequency domain source separation of convolved mixtures: the fixed-point and the second fixed-point algorithm.

The algorithms proved to be more stable and faster compared to former maximum likelihood approaches, as they are based on a second order optimization method. The quality of the separation is good, although the separation quality is dependent on how reverbant the recording environment is.

The two algorithms performed similarly in all tests, with the second fixed-point being a little bit faster, more robust and producing slightly better audible results. This is due to the fact that the algorithms basically exploit the same second order optimization method.

Furthermore, the introduction of a time-frequency prior in the source model combined with the likelihood ratio jump solution, introduced in [3], proved to be able to solve the permutation problem, in the two fixed-point frameworks. However, this likelihood test becomes more complicated for more than two sources.

In future, we hope to formulate this likelihood ratio jump test for more sources. In addition, we are interested in replacing the STFT analysis with a multi-resolution analysis framework, aiming to improve the separation quality. Moreover, we hope to improve the performance and speed of these fixed-point solutions.

7. REFERENCES

[1] Smaragdis P., "Blind Separation of convolved mixtures in the frequency domain", *International Workshop on Independence & Artificial Neural Networks*, University of La Laguna, Tenerife, Spain, February 9 - 10, 1998.

[2] Smaragdis P., "Information Theoretic Approaches to Source Separation", May 1997, Masters Thesis, *MIT Media Lab*.

[3] Davies M., "Audio Source Separation", *Mathematics in Signal Processing V*, 2000.

[4] Ikeda S., Murata N., "A method of ICA in Time-Frequency Domain", *International Conference on Independent Component Analysis and Signal Separation*, pp 365—371, Jan 1999.

[5] Bell A., Sejnowski T., "An information-maximization approach to blind separation and blind deconvolution", *Neural Computation*, 7: 1129-1159, 1995.

[6] Amari S., Cichocki A., Yang H. H., "A new learning algorithm for blind source separation", Advances in *Neural Information Processing Systems*, pp. 757-763, MIT Press, Cambridge MA, 1996.

[7] Hyvarinen A., "Survey on Independent Component Analysis", *Neural Computing Surveys* 2:94--128, 1999.

[8] Hyvärinen A., Oja E., "A Fast Fixed-Point Algorithm for Independent Component Analysis", *Neural Computation*, 9(7):1483-1492, 1997.

[9] Hyvarinen A., "The Fixed-point Algorithm and maximum likelihood estimation for Independent Component Analysis", *Neural Processing Letters*, 10(1):1-5.

[10] Bingham E., Hyvarinen A., " A fast fixed-point algorithm for independent component analysis of complex-valued signals", *Int. J. of Neural Systems*, 10(1):1-8, 2000.

[11] http://www.cnl.salk.edu/~tewon/ica_cnl.html

[12] http://www.media.mit.edu/~westner/

[13] Schobben D., Torkkola K., Smaragdis P., "Evaluation of blind signal separation methods", Proceedings of the *Workshop on Independent Component Analysis and Blind Signal Separation*, Aussois, France, January 11-15 1999.