Adaptive Speech Separation Using Hybrid Approach

YAN LI
Department of Mathematics and Computing
University of Southern Queensland
Toowoomba
AUSTRALIA

Abstract: - A hybrid iterative learning algorithm for recurrent neural networks based on higher-order statistics to blind signal separation is presented in this paper. Fourth-order statistics are used as the separation criterion to train an RNN to perform the separation. Some simulation results for both artificially convoluted audio signals and real recordings demonstrate that the proposed approach is promising.

Keywords: - Recurrent neural networks, Blind signal separation, Higher-order statistics, Independent component analysis, Cross-cumulants, Probability density functions

1 Introduction

In blind signal separation (BSS), the goal is to extract independent source signals from their mixtures using a minimum of priori information. In the general case, it assumes that the sources are non-Gaussian signals and statistically independent of one another. Higher-order statistics (HOS) methods and theories were one of the most important fields in signal processing theory. The HOS can be considered as an important complement of the classic second order statistics methods (power, variance, covariance and spectra) to solve many recent and important telecommunication problems [1], such as blind identification or equalization, blind separation of sources and time delay estimation.

In the literature, various criteria based on HOS are used for solving the problem of blind separation of sources. In the case of instantaneous linear mixtures, the first solution, proposed in 1985, was based on cancellation of higher-order moments. Other criteria based on minimization of cost functions, such as the sum of square fourth-order cumulants [2] or contrast functions [3] have been used by several researchers. It was proved in [4] that the minimization or cancellation of fourth-order cross-cumulants, leads to a set of solutions to signal separation.

In this paper, we present the performances of speech signal separation of using a recurrent neural network (RNN) through higher-order statistics (fourth-order statistics) separation criterion. An unsupervised learning algorithm using the RNN to speech signal separation is introduced in the next section. In section 3, some simulation results for both artificially mixed audio signals and real recordings using the proposed approach are presented.

2 Separation Algorithm

BSS is the process of extracting unknown source signals from sensor measurements, which are unknown combinations of the sources. The term "blind" is used as the source signals and the combinations are unknown.

2.1 The Data Model

There are two kinds of data models used by many researchers. One is linear model. Another type is non-linear model.

In traditional linear independent component analysis (ICA) model, all received signals for instantaneous mixing are the linear superposition of the sources, namely, the outputs of microphones. Suppose that sources is denoted by a vector:

$$\underline{S}(t)=[S_1(t), S_2(t), \ldots, S_n(t)]^T, \; t = 0, 1, 2\ldots$$

and the observed signals are denoted by

$$\underline{X}(t)=[X_1(t), X_2(t), \ldots, X_n(t)]^T, \; t = 0, 1, 2\ldots,$$

then, we have general equations:

$$\underline{X}(t)=\underline{A}(t)\cdot\underline{S}(t) \quad (1)$$

$$\underline{Y}(t)=\underline{W}(t)\cdot\underline{X}(t) \quad (2)$$

here, $\underline{A}(t)$ is the unknown mixing matrix and $\underline{Y}(t)=[Y_1(t), Y_2(t), \ldots, Y_n(t)]^T$

denotes the separated signals so that

$\underline{W}(t)=\underline{A}^{-1}(t) \quad \Rightarrow \quad \underline{Y}(t) = \underline{S}(t).$

Thus the task is to recover the original sources by finding a matrix $\underline{W}(t)$ that is in general time-varying and a permutation and rescaling of the inverse of the unknown matrix $\underline{A}(t)$, so that $\underline{Y}(t)$ is as close as possible to $\underline{S}(t)$.

In convolutive BSS, a source is corrupted by time-delayed versions of itself and other source signals. In this case, the equations (1) and (2) hold in the frequency domain. Taking the *z*-transform of equations (1) and (2), we have

$$\underline{X}(z)=\underline{A}(z)\underline{S}(z) \quad (3)$$
$$\underline{Y}(z)=\underline{W}(z)\,\underline{X}(z) \quad (4)$$

for convolutive BSS.

The basic linear models (1), (2) and (3), (4) are often too simple for describing the observed data $\underline{X}(t)$ adequately. A natural extension of the linear ICA (or BSS) models is to assume that the components of the data vectors $\underline{X}(t)$ depend nonlinearly on some statistically independent components (source signals) $\underline{S}(t)$. Thus the instantaneous mixtures

$$\underline{X}(z)=F(\underline{S}(t)) \quad (5)$$

where, F: $R^m \rightarrow R^m$ is unknown nonlinear mixing function.

The nonlinear ICA problem consists of finding an inverse mapping G: $R^m \rightarrow R^m$, which gives estimates of the independent components as:

$$\underline{Y}(t)= G(\underline{X}(t)) \quad (6)$$

Solutions of the nonlinear ICA problems are usually highly non-unique [5]. For getting more unique solutions, various constraints have been introduced, but it is not clear which constraints are most meaningful in different situations.

In this paper, a non-linear ICA data model is used and we constrain the diagonal entries of the $\underline{W}$ matrix to unity: $w_{ij} = 1$, when $i=j$. Non-linear ICA tends to be computationally demanding. The computational load usually increases very rapidly with the dimensionality of the problem, preventing in practice the applications of non-linear BSS methods to high-dimensional data sets.

We assume that each component of $\underline{S}(t)$ is independent of each other. The independence of the sources is defined by their joint probability density function (PDF) $P_s(S)$:

$$P_s(S)=\prod_{i=1}^{n} P_{S_i}(S_i)$$

and source signals are independent if and only if this identity holds.

2.2 Higher-Order Statistics

The higher-order statistics (usually the higher-order cumulants) can be used as a natural measure of the degree of the independence. Equivalently, the problem of the source extraction becomes the task of separating the observed joint PDF into the independent source PDF's that generate the former through a linear/nonlinear transformation.

Cumulants, and their associated Fourier transforms, not only reveal amplitude information about a process, but also reveal phase information. This is important, because, as is well known, second-order statistics (i.e., correlation) are phase blind. Cumulants, on the other hand, are blind to any kind of a Gaussian process, whereas correlation is not; hence, cumulant-based signal processing methods handle colored Gaussian measurement noise automatically and cumulant-based methods boost signals-to-noise ratio when signals are corrupted by Gaussian measurement noise.

An important benefit of basing ICA on 4th-order cumulants becomes apparent in that as 4th-order cumulants are polynomial in the parameters. If the random signals $x_i(t)$ and $x_j(t)$ are mutually statistically independent, then the cross-cumulants of any order must be equal to zero. It has been proved by many authors that the statistics up to the fourth-order are sufficient.

Let $x_i(t)$ be the zero-mean observed signals, and let $M_{lm}(x_i, x_j) = E[x_i^l x_j^m]$ be the *(l+m)th*-order cross-moments. At the fourth-order, the cross-cumulants of two independent signals [4] are:

$$\begin{aligned} Cum_{13}(x_i, x_j) &= M_{31}(x_i, x_j) - 3M_{20}(x_i, x_j)M_{11}(x_i, x_j) \\ &= E(x_i^3 x_j) - 3E(x_i^2)E(x_i x_j) \end{aligned} \quad (7)$$

$$\begin{aligned} Cum_{22}(x_i, x_j) &= M_{22}(x_i, x_j) - M_{20}(x_i, x_j)M_{02}(x_i, x_j) \\ &\quad - 2M_{11}^2(x_i, x_j) \\ &= E(x_i^2 x_j^2) - E(x_i^2)E(x_j^2) - 2(E(x_i x_j))^2 \end{aligned} \quad (8)$$

$$Cum_{31}(x_i,\ x_j) = M_{13}(x_i, x_j) - 3M_{02}(x_i, x_j)M_{11}(x_i, x_j)$$
$$= E(x_i\, x_j^3) - 3E(x_j^2)E(x_i x_j) \quad (9)$$

The equations (7), (8) and (9) are used as the separation criterion in this paper to minimizing or canceling the mutual information within the sources. A hybrid iterative learning algorithm based on higher-order statistics for an RNN with time varying weights is introduced in the next section. The weights of the RNN can be estimated in real-time, and the fourth-order cross-cumulants can be reduced close to zero. The algorithm drives Cum_{31}, Cum_{13} and Cum_{22} close to zero recursively in this paper.

2.3 Recurrent Neural Networks And The Learning Algorithm

In an RNN, basic processing units are connected arbitrarily so that there are both feedforward and feedback paths. Nodes in an RNN are generally classified into three categories (instead of layers): input, output, and hidden nodes. Input nodes receive external input signals, and output nodes send off output signals calculated through the network. Hidden nodes neither receive external input signals nor send off output signals, but rather exchange internal signals with other nodes. In this paper, we use processing nodes to represent all the output nodes and hidden nodes. Processing nodes in an RNN are usually fully connected: they receive output signals from all nodes including themselves. Fig. 1 shows the topology of the RNN.

There are two sets of synaptic connections in RNNs. The first set of connections link the input and the processing nodes. Their weights constitute the inter-weights matrix $\underline{W}_2=\{w_{ij}\}$. The weight $w_{ij}(t)$ $\forall i \in \underline{U}$ and $j \in \underline{I}$ (where $\underline{U}$ and $\underline{I}$ are the sets of

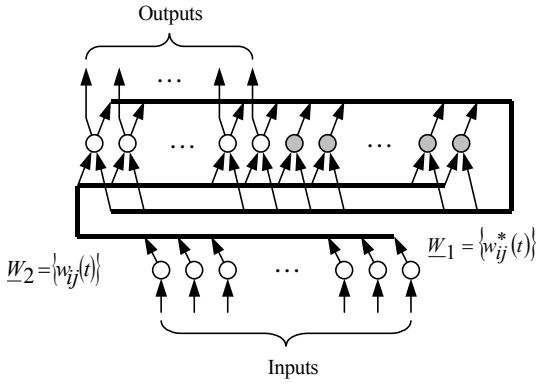

Fig. 1 The topology of the RNN

processing and input nodes, respectively) denotes the strength of the connection from the j^{th} input node to the i^{th} processing node, at time t. The second set of connections forms the feedback paths. Therefore, each processing node is connected to all other processing nodes, including itself. Their weights constitute the intra-weight matrix $\underline{W}_1=\{w_{ij}^*\}$. Similarly, $w_{ij}^*(t)$ denotes the strength of the connection from the j^{th} processing node to the i^{th} processing node for $\forall i, j \in \underline{U}$, at time t.

Let $\underline{y}(t)=\{y_i(t)\}$ denote the outputs of the processing nodes and $\underline{u}(t)=\{u_j(t)\}$ denote the external inputs. Then, we use the corresponding learning rule (See [6] for the details) of the form:

$$\Delta \underline{W}(t+1) = \underline{C}^{-1}(t) Cum[\underline{x}(t)^T \underline{x}(t)]^{-1} \underline{x}(t)^T \quad (10)$$

where, Cum is the fourth-order cross-cumulants of outputs shown in equation (7), (8), and (9), and

$$\underline{C}^{-1}(t) = [diag(f'(\underline{\xi}_1), f'(\underline{\xi}_2), ..., f'(\underline{\xi}_n))]^{-1},$$
$$\underline{\xi} = [\underline{\xi}_1, \underline{\xi}_2, ..., \underline{\xi}_n]^T = \underline{W}(t)\underline{x}(t),$$

$$\underline{x}(t) = \begin{bmatrix} \underline{y}(t) \\ \underline{u}(t) \end{bmatrix},\ \underline{y}(t+1) = f[\underline{W}(t)\underline{x}(t)]$$

$\underline{W}(t+1)=\underline{W}(t)+\Delta\underline{W}(t)$ with constraint $w_{ij}=1$, when $i=j$, and

$$\underline{W}(t+1)=[\underline{W}_1(t+1)\,|\,\underline{W}_2(t+1)].$$

For each Cum_{13}, Cum_{31} and Cum_{22}, the algorithm drives the corresponding cross-cumulant to zero recursively. In this way, we can obtain the outputs of the network:

$$\underline{y}(t+1) = f[\underline{W}(t)\underline{x}(t)]$$

The algorithm is effective as the iterative learning algorithm is able to drive the fourth-order cross-cumulants to zero. Comparing (10) with other neural algorithms in BSS/ICA, the real-time learning algorithm has a dynamic learning rate of $\underline{C}^{-1}(t)[\underline{x}(t)^T \underline{x}(t)]^{-1}$.

3 Simulation Results

The simulation results for two sets of audio signals using the proposed approach are given in this section. The results are then compared with those obtained by our other approach – using output decorrelation and time-delay [6] in terms of the signal to noise ratio (SNR) and computational

complexity. The observed signals are convolutively mixed signals, and real environment recordings.

The proposed algorithm in this paper is an on-line algorithm that separates the signals sample by sample. Furthermore, the approach used in this study is non-linear. It is not appropriate to use the weights as the main objective measure as many researchers have done, such as in [7]. The process is non-linear and the final weight matrix doesn't mean anything directly useful, viz $WA^{-1} \neq P\Lambda$. Instead, the experiments are assessed qualitatively by listening to and viewing the waveforms, and are quantitatively evaluated by SNRs of the separated signals.

Assume *s(t)* is the desired signals, and *y(t)* is the estimated source signals. *s(t)* and *y(t)* have the same energy. Then *n(t)=y(t)-s(t)* estimates the undesired components (the noise). The SNR of the separated output signals is defined by the following formula:

$$SNR = 10\log\left(\frac{E[s(t)^2]}{E[n(t)^2]}\right)$$

where *E[.]* is the mean of the arguments. The SNR will show how much louder the desired sources are than the undesired sources, with a SNR of 15dB sounding perfectly separated, 6dB being effectively separated, and 3dB being perceptibly separated [8].

The simulations were carried out in MATLAB 6.0.0.88 (R12) on Red Hat Linux.

Experiment 1 Convolutively Mixed Signals

One recording of male speech $S_1(t)$ and one recording of female speech $S_2(t)$ were taken from a CD available with Daniel Schobben's thesis [9]. They were convolutively mixed together. The mixed signals $X_1(t)$ and $X_2(t)$ are in the form of

$$x_1(t)=s_1(t)+0.2s_2(t)+0.1s_1(t-1)+0.6s_1(t-3)+0.04s_2(t-2)$$
$$x_2(t)=s_2(t)+0.3s_1(t)+0.5s_2(t-1)+0.3s_2(t-2)+0.03s_1(t-1)$$

The original and convolutively mixed signals are shown in figures 2 and 3. The SNRs for the separated signal 1 in figure 4 is 9.8dB, and 11.2dB for the separated signal 2. The running time for the experiment is 530 seconds. It is observed that the cross-cumulant Cum_{22} has less effect on the performance of separation than Cum_{31} and Cum_{13} in that case. For the same convolutively mixed signals, the output decorrelation and time-delay approach in [6] produces 10.2dB SNR for the separated signal 1 and 10.6dB for the separated signal 2. It is noted that the SNRs are at the same levels with both methods. And there are no audible differences between two sets of separated signals when they are listened to.

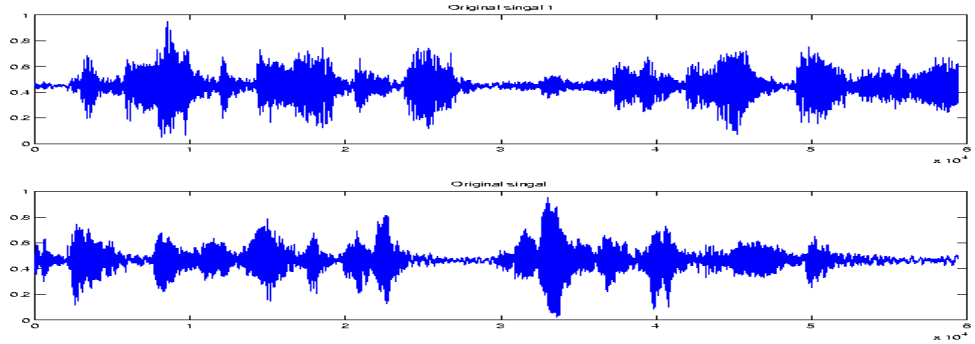

Fig. 2 The source signals

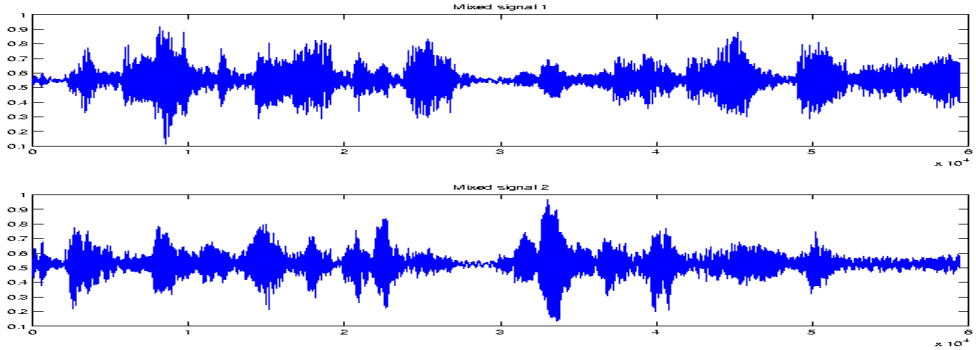

Fig. 3. The Mixed Signals

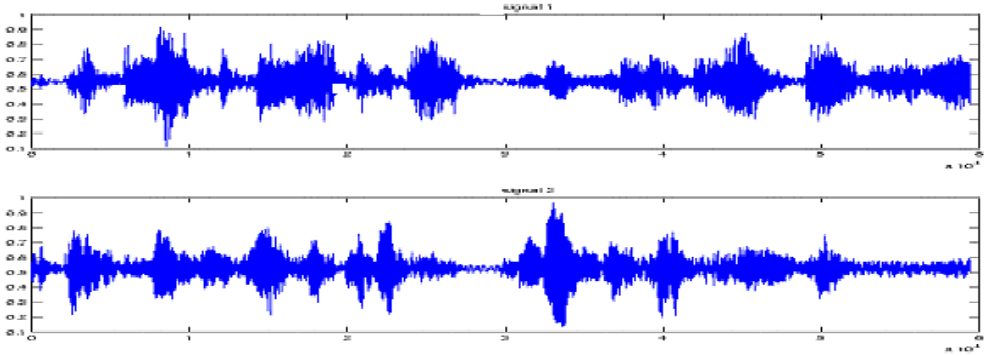

Fig. 4. The Separated Signals

Experiment 2 Real Recordings

There are several sets of speech recordings made in our ordinary offices. One set of recordings with two-input/two-output were presented in this paper. The two sources, one male speech and one female speech are pre-recorded .wav files, reading news items. They were taken from a commercial CD set (LDC/NIST WSJ0) [10] professionally recorded for speech recognition purpose, and were regarded as clean recordings. The two files were merged into one stereo sound data using AudioEditor software package. The male speech is in the right channel of the file, and female's voice is in the left channel. One PC, connecting with two speakers was selected

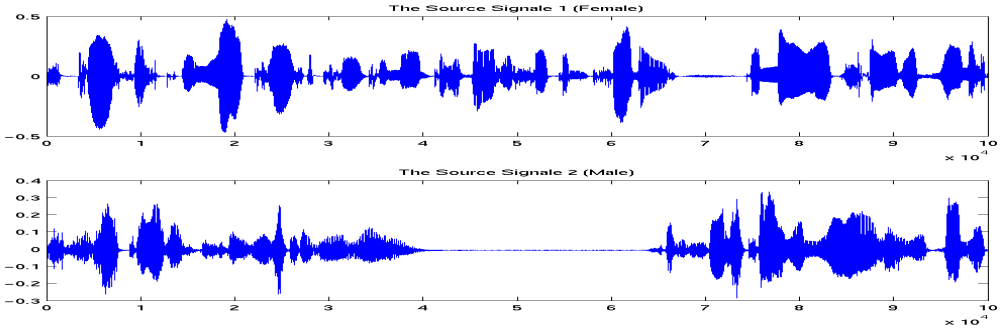

Fig. 5 The Source Signals

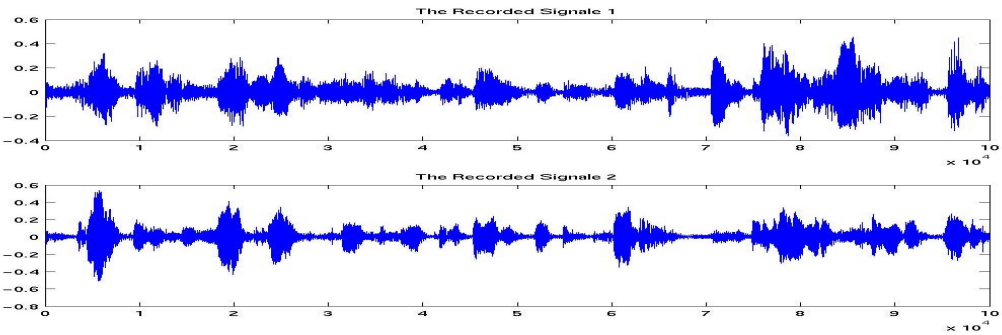

Fig. 6 The Recorded Signals

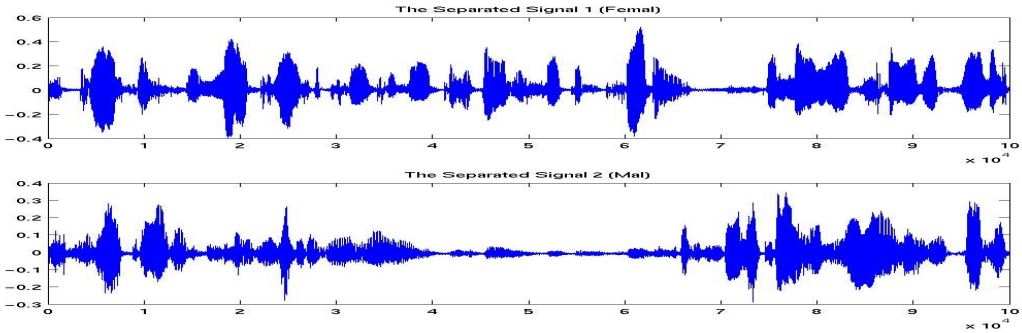

Fig.7 The Separated Signals

to play the sound files. Other two PCs were used to record the sound data. To minimizing the recording noise from the three computer drivers, the three computers are in one room, the two loud speakers and microphones, hanging from the ceiling, are located in another room, which is diagonally opposite to the computer room. The two rooms are approximately 3 meters apart. The two speakers are 120cm away and two microphones are 80cm apart. The right channel of each recording PC receives input from a microphone, the left channel is a direct line input from a loud speaker. This setup helps synchronise all the signals. The source speeches shown in figure 5 were used for comparing with the separation results using the proposed algorithm. The recordings and the separated signals are shown in figure 6 and 7.

The SNR for the separated signal 1 (male speech) in figure 7 is 4.9dB, and 6.7dB for the separated signal 2 (female speech). It is noted that the separated female voice is a bit louder than the male's. In separating the set of real recordings, the algorithm tends to be unstable sometimes. Without the weight constraint, the algorithm doesn't work appropriately. Unfortunately, the output decorrelation and time delay approach in [6] did not produce perceptible separation results, and Bell and Sejnowski's infomax algorithm in [7] also failed in separating this set of real recordings.

4 Discussion

Many different approaches have been attempted by numerous researchers using neural networks, artificial learning, higher order statistics, minimum mutual information, beam-forming and adaptive noise cancellation, each claiming various degrees of success. But the separation of speech in real environments is still very challenging. There are many facts such as, synchronizing, surroundings in the recorded room playing great roles in separating real recordings.

This paper aims to exploit the application of RNNs using higher-order statistics to blind signal separation. An iterative learning algorithm using an RNN is presented. In Experiment 1, the performance of the proposed approach is stable. It is observed that the cross-cumulant Cum_{22} has less affects on the performances of separation than Cum_{31} and Cum_{13} in that case. The cross-cumulant Cum_{22} tends to close to zero when successfully minimizing Cum_{31} and Cum_{13} recursively. Comparing with Experiment 1, separating recordings is much more challenge. To make the evaluation easier, we use the clean pre-recorded signals as the sources instead of 'living' speech, which allows us to compare the quality (SNRs) of separation results by re-playing and re-recording. This proved very effective in giving us an objective performance standard, which has been lacking in previous BSS research.

References:

[1] Comon P., Independent component analysis- a new concept?, signal processing, vol. 36, 1994 pp. 287-314.

[2] Mansour A. and Ohnishi N. (1999), Multichannel Blind Separation of Sources Algorithm Based on Cross- Cumulant and the Levenberg-Marquardt Method, Vol. 47, No. 11, November, pp. 3172-3179.

[3] Mansour A and Jutten C (1995) Fourth Order Criteria for Blind Sources Separation. IEEE Transactions on Signal Processing, Vol. 43, No. 8 August 1995. pp. 2022-2025.

[4] Nguyen Thi H.L. and Jutten C. (1995) Blind Source Separation for Convolutive Mixtures, Signal Processing Vol. 45, pp. 209-229.

[5] Lin J., Grier D. and Cowan J. (1997), Faithful representation of separable distributions, Neural Computation, Vol. 9, pp.1305-1320.

[6] Li Y., Powers D. and Wen P. (2001), Separation and Deconvolution of Speech Using Recurrent Neural Networks, pp. 1303--1309, Vol. III, Proceedings of the International Conference on Artificial Intelligence (IC-AI'01), June 25-28, 2001, Las Vegas, Nevada, USA.

[7] Bell, A. J. and T. J. Sejnowski (1995). An information-maximisation approach to blind separation and blind deconvolution, Neural Computation, 7(6), 1004-1034.

[8] Westner, A. G. (1999). Object based audio capture: Separating acoustically mixed sounds. Master's thesis, MIT.

[9] Schobben, D. (1999), Efficient adaptive multi-channel concepts in acoustics: Blind signal separation and echo cancellation, PhD thesis.

[10] Contiunous Speech Recognition Corpus (WSJ0) produced by NIST and LDC (Linguistic Data Consortium), sponsored by ARPA, Collected by SRI, TI & MIT, NIST Speech Disc No.11, File No. 011C02 (Female) and 431A01 (Male).