

# Comparison of Blind Source Separation Algorithms

Yan Li, David Powers and James Peach  
School of Informatics and Engineering  
The Flinders University of South Australia, Australia  
GPO Box 2100, Adelaide, SA 5001, Australia  
{yan.li, david.powers}@flinders.edu.au

*Abstract:* - A set of experiments are designed to evaluate and compare the performances of three well known blind source separation algorithms in this paper. The specific algorithms studied are two group of neural networks algorithms, Bell and Sejnowski's infomax algorithm and Hyvärinen's fixed-point family, and J. F. Cardoso's joint approximate diagonalization of eigen-matrices algorithm. In this paper, the algorithms are quantitatively evaluated and compared using the three measures, MATLAB flops (floating point operations), the difference between the mixing and separating matrices and the signal-to-noise ratios of the separated signals in this paper.

*Key-Words:* - blind source separation, information maximisation, fixed-point algorithm, JADE

## 1 Introduction

The goal of blind source separation (BSS) is to recover independent sources given only sensor observations that are linear mixtures of independent source signals. Due to a number of interesting applications in communications, speech and medical signal processing, BSS has received lots of attention. 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.

Bell and Sejnowski [1] put BSS problem into an information theoretic framework and demonstrated the separation and deconvolution of mixed sources. Their adaptive methods are more plausible from a neural networks processing perspective than the cumulant-based cost functions proposed by Comon [2]. A similar adaptive method for source separation was proposed by Cardoso and Laheld [3]. Cardoso and Laheld [3] demonstrated that their joint approximate diagonalization of eigen-matrices (JADE) is equivalent to informatic approaches. The contrast function used in their paper is effectively a measure of mutual information between the cross-cumulants of signals. Recently, many BSS algorithms tend to come together with neural networks to perform the learning rules, probably because of neural networks are powerful in nonlinear mapping and learning ability. Aapo Hyvärinen *et al* [4] has presented a family of neural algorithms called "fixed-point algorithm", which have proved very successful.

In order to have a better insight into BSS algorithms, a group of experiments are designed to evaluate and compare the performances of some main algorithms in this paper. The specific algorithms evaluated are Bell and Sejnowski's infomax algorithm [1] with the modified natural gradient learning rule proposed by Amari *etc* [5], J. F. Cardoso's JADE algorithm [6] and Hyvärinen's fixed-point algorithm [7]. The algorithms are quantitatively compared using the three measures, MATLAB flops (floating point operations) which indicates the computational complexity, the difference between the mixing and separating matrices which indicates separation quality, and the signal-to-noise ratio (SNR) of the separated signals. The quality of the implementation could have a large impact on the results, high quality implementations written by members of the research groups that originally produced the algorithms were utilised for these experiments. This was done so that the results could be more readily attributed to the algorithms themselves, rather than to flaws or idiosyncrasies in their implementations.

The rest of the paper is organised as follows: The specific algorithms studied in this paper are introduced in the next section. The performance evaluations of the algorithms on four different experiments are presented in section 3. Finally, discussions are included in section 4.

## 2 Blind Source Separation Algorithms

The BSS algorithms studied in this paper are the infomax, fixed-point and JADE algorithms.

## 2.1 Information maximisation algorithms

The information maximisation algorithm (often known as infomax) developed by Bell and Sejnowski [1] catalysed a surge of interest in using information theory to perform blind source separation. In 1995, Bell and Sejnowski proposed an adaptive learning algorithm that maximizes the information passed through a neural networks. The paper shows that a neural network is capable of resolving the independent components in the inputs, that is, the neural network can perform independent component analysis. The main idea is that maximizing the joint entropy  $H(y)$  of the outputs of a neural processor can approximately minimize the mutual information among the output components.

The joint entropy of  $n$  variable,  $y_1, y_2, \dots, y_n$  (Assume they are the outputs of a neural network) may be written as:

$$H(y_1, \dots, y_n) = H(y_1) + \dots + H(y_n) - I(y_1, \dots, y_n) \quad (1)$$

If the nonlinear transfer function of a neural network matches the probability density function of the inputs and the joint entropy  $H(y_1, \dots, y_n)$  of the outputs is maximized, the mutual information  $I(y_1, \dots, y_n)$  among the outputs is then minimized. The output signals are assumed to be independent. Bell and Sejnowski noted that many real-world analog signals including the speech signals are super-gaussian and satisfies the independence condition assumed if the transfer function of the neurons in the neural network is a sigmoidal or hyperbolic tangent function.

The learning rule for a single layer feedforward neural network to implement the separation is

$$\Delta \mathbf{W} \propto [\mathbf{W}^T]^{-1} + (\mathbf{1} - 2\mathbf{y})\mathbf{x}^T \quad (2)$$

$$\Delta \mathbf{w}_0 \propto \mathbf{1} - 2\mathbf{y} \quad (3)$$

where  $y = f(\mathbf{W}\mathbf{x} + \mathbf{w}_0)$  and  $f(u)$  is a sigmoid contrast function, usually  $f(u) = 1 + e^{-u^{-1}}$  or  $f(u) = \tanh(u)$ .

A similar learning algorithm was derived by Amari *et al* [5] using the natural gradient in the parameter space instead of the descent gradient. This learning rule is

$$\Delta \mathbf{W} \propto [\mathbf{I} - f(\mathbf{u}) \cdot \mathbf{u}^T] \cdot \mathbf{W}. \quad (4)$$

where  $f(x) = \frac{3}{4}x^{11} + \frac{25}{4}x^9 - \frac{14}{3}x^7 - \frac{47}{4}x^5 + \frac{29}{4}x^3$ . This rule speeds up the algorithm by removing the computationally expensive matrix inversion in equation 2.

## 2.2 Fixed-Point Algorithms

Aapo Hyvärinen has introduced a family of algorithms that are grouped under the epithet “fixed-point algorithms”. The members of this family are differentiated firstly by the algorithmic approach and secondly by the contrast function used. The key to all the variations is finding independent components by separately maximising the negentropy of each mixture [4][7].

There are mainly two algorithmic approaches in the fixed-point algorithm class. The symmetric approach uses a modified rule for the update of  $\mathbf{W}$  that enables simultaneous separation of all independent components, whereas the deflation approach updates the columns of  $\mathbf{W}$  individually, finding the independent components one at a time. Either of these approaches is able to use almost any non-quadratic contrast function to provide estimates of negentropy [4]. The original algorithm used kurtosis [7], but more recent versions use the hyperbolic tangent, exponential or cubic functions.

The update rule for the deflation method is given by [4]:

$$\mathbf{w}^*(k) = \mathbf{C}^{-1} E\{\mathbf{x}g(\mathbf{w}(k-1)^T \mathbf{x})\} \quad (5)$$

$$- E\{g'(\mathbf{w}(k-1)^T \mathbf{x})\} \mathbf{w}(k-1) \quad (6)$$

$$\mathbf{w}(k) = \mathbf{w}^*(k) / \sqrt{\mathbf{w}^*(k)^T \mathbf{C} \mathbf{w}^*(k)} \quad (7)$$

where  $g$  can be any suitable non-quadratic contrast function, with derivative  $g'$ , and  $\mathbf{C}$  is the covariance matrix of the mixes,  $\mathbf{x}$ .

The most interesting feature of the fixed-point family of algorithms is that the deflation method can be used to find independent sources one at a time. Hyvärinen notes that the ordering of the discovered sources is determined by the contrast function and that by using kurtosis as the contrast function, the super-Gaussian sources tend to be found first. Given that speech signals are strongly super-Gaussian, this suggests the possibility of using the fixed-point deflation algorithm with kurtosis as the contrast function and simply finding the first source signal. With some fine-tuning of the contrast function, this algorithm may be able to guarantee that the first source found is the speech signal.

## 2.3 JADE

The JADE algorithm presents a subtle contrast to both the infomax and fixed-point algorithms [6][8] in that previous algorithms optimise a particular transform of the input data, whereas JADE optimises a transform of a particular set of statistics about the data. The starting point for the JADE algorithm is the realization that BSS algorithms generally require an estimation of the distributions of the independent sources or have such an assumption built into the algorithm [8]. Cardoso notes that optimising cumulant approximations of data implicitly performs this, leading him to present a number of approximations to information theoretic algorithms that operate on second and fourth order cross-cumulants.

All of the information theoretic measures can be calculated by operations on cumulants, eg. variance and kurtosis are the second and fourth order auto-cumulants [8]. The advantage of operating on cross-cumulants is that the algorithms do not require gradient descent and thus avoid any convergence problems. A side effect of this is that the

JADE algorithm requires no parameter tuning for good performance. A disadvantage of this approach is that estimating a complete set of fourth order cross-cumulants requires storage of  $O(n^4)$  cumulant matrices. The cumulant matrix with elements  $[Q^X(M)]_{ij}$  is defined in [8] as:

$$[Q^X(\mathbf{M})]_{ij} = \sum_{k=j=1}^n Cum(\mathbf{X}_i, \mathbf{X}_j, \mathbf{X}_k, \mathbf{X}_l) \mathbf{M}_{kl} \quad (8)$$

where  $\mathbf{M}$  is an  $n \times n$  matrix and  $\mathbf{X}$  is an  $n \times 1$  random vector.

The JADE algorithm uses second and fourth order cumulants. The second order cumulant is used to ensure the data is “white” (ie. decorrelated). This produces a whitening matrix  $\hat{\mathbf{W}}$  and the whitened sources  $\mathbf{Z}$ . A set of cumulant matrices is estimated from the whitened sources. Cardoso demonstrates that the separating matrix can be estimated as  $\mathbf{V}' \cdot \hat{\mathbf{W}}$ , where  $\mathbf{V}$  is a rotation matrix “that makes the cumulant matrices as diagonal as possible” according to the JADE contrast function [8]. The JADE contrast function [8] is the sum of squared fourth order cross cumulants from the set defined in equation 8:

$$\phi^{JADE}(Y) = \sum_{ijkl \neq iikl} (Q_{ijkl})^2 \quad (9)$$

This contrast function is effectively a measure of mutual information between the cross-cumulants. Making the cumulants diagonal as possible can be translated as making the data as independent as possible [8]. The matrix that performs the diagonalisation on cumulants can be translated to perform separation on the mixed data.

### 3 Performance Evaluation

For comparing the performances of the three main separation algorithms mentioned above, this section describes four experiments, each with tasks of increasing difficulty. The first experiment is based on an artificially created instantaneous signals, whereas the second one is an instantaneous separation task using real recorded sounds. The third experiment examines the performance of the separation of simulated “moving sources”, and the final one presents real recordings separation. All the experimental results are implemented in Matlab scripts.

The criteria used to evaluate the performances of the BSS algorithms are the evaluations and comparisons of source separation algorithms performed by Gannakopoulos [9] and Westner [10]. Gannakopoulos [9] examined the computational complexity and separation quality of BSS algorithms for instantaneously mixed sources. The two main measures used are MATLAB flops (floating point operations) which indicates the computational complexity, and the difference between the mixing and separating matrices which indicates

separation quality. If separation is perfect, then the separating matrix is the inverse of the mixing matrix and this error distance ( $E_1$ ) is zero. As  $E_1$  increases, the separating matrix is progressively farther from the inverse of the mixing matrix, so separation quality is correspondingly poorer. The difference between the mixing and separating matrices is calculated with the following equation [5]:

$$E_1 = \sum_{i=1}^n \left( \left( \sum_{j=1}^n \frac{|p_{ij}|}{\max_k |p_{ik}|} \right) - 1 \right) + \sum_{j=1}^n \left( \left( \sum_{i=1}^n \frac{|p_{ij}|}{\max_k |p_{jk}|} \right) - 1 \right) \quad (10)$$

where  $\mathbf{P} = (p_{ij}) = \mathbf{W}\mathbf{A}$ .

Another measure of separation quality used was the SNRs of the separated outputs, with a SNR of 15dB sounding perfectly separated, 6dB being effective, and 3dB being perceptibly improved [10]. The equation describing the SNR is:

$$SNR = 10 \log_{10} \left( \frac{E[s(t)^2]}{E[n(t)^2]} \right) \quad (11)$$

where  $E(\cdot)$  is the mean of the arguments,  $s(t)$  is the desired signal and  $n(t) = y(t) - s(t)$  is the noise, indicating the undesired signal. Here  $y(t)$  is the estimated source signals, and  $s(t)$  and  $y(t)$  are at the same energy.

It must be mentioned that the number of FLOPS are an ill-defined notion and don’t translate readily between different computer architectures. Similarly, small SNRs and large values for the error distance,  $E_1$ , are not in themselves indicators of failure. In particular, an error distance of zero is technically the perfect identification of the mixing matrix, but acceptable results are usually achieved well before this occurs. The use of multiple performance indicators in this evaluation provides a rounded picture of the examined algorithms.

The sampling rate of the sound data did not remain constant across experiments, however the performances of the various algorithms are independent of sample rates. Each source is considered a vector of samples and the measures of independence used have no regard for the time quantisation period. Varying the number of sources significantly affects the difficulty of the problem, so for most experiments the number of sources was fixed at four.

#### 3.1 Performance On Artificial Created Data

The aim of this test was to establish a baseline of relative performance for the algorithms in question. The task was to separate four artificial signals, consisting of 8000 data points, mixed by a random matrix. The signals used were two smooth curves, a sawtooth wave and a noise function, as shown in Figure 1.

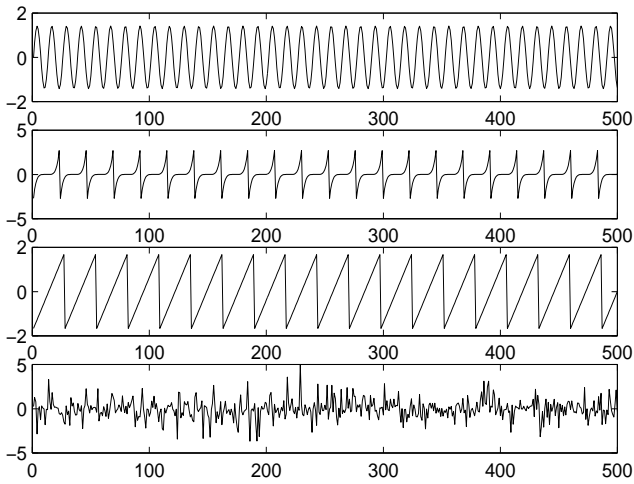


Figure 1. The first 500 points of the artificial sources

Algorithm	Flops	Error ( $E_1$ )	SNR (dB)			
			1	2	3	4
JADE	4,254,672	0.097	42.8	53.6	43.2	34.6
InfoMax	4,330,103	5.397	3.6	6.0	10.2	11.1
FP (tanh)	3,427,304	0.134	46.6	47.2	51.5	30
FP (pow3)	2,437,989	0.102	40.8	60.3	51.5	33.3
FP (gauss)	3,621,475	0.110	46.1	56.5	58.7	30.9

Table 1. Results for experiment 3.1

The main surprise of this experiment was the poor performance of the infomax algorithm. It was expected that this would achieve high quality separation. However, it can be seen from the results in Table 1 that it performed considerably worse on both the error distance ( $E_1$ ) and the SNR metrics. The fixed-point algorithms were the fastest despite their having larger  $E_1$  scores than the JADE algorithm, and tended to produce better SNRs. Consistent with this, visual comparison of plots of the separated sources indicated that the fixed-point and JADE algorithms performed well, while the infomax algorithm showed marked irregularities. It appeared to have the most difficulty with the the first source, which was barely recognisable as sinusoid.

### 3.2 Performance On Real Data With Artificial Mixing

In this experiment, the task was made harder by feeding each algorithm real wave data mixed by the same process as in the previous experiment. The sounds used were a fragment of classical music, a sample of bubbling water, a sample of a telephone conversation and a “zap” sound effect (Figure 2). This choice of sources was intended to be representative of a real environment. The sounds used were samples at 11,025 Hz using 16 bits/sample. One second of audio (11025 data points) was examined by the algorithms.

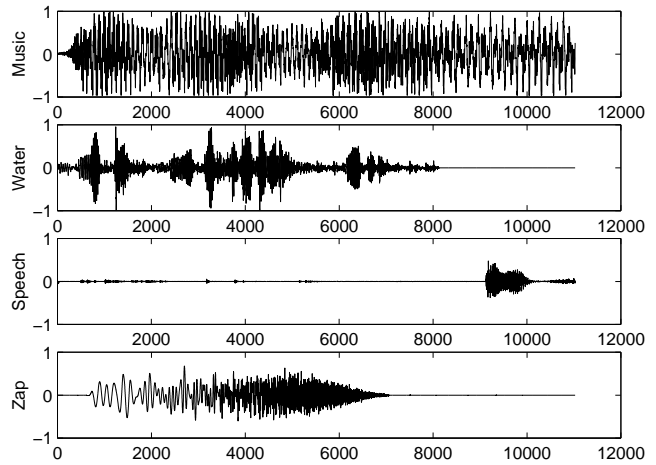


Figure 2. The sound samples used in experiment 2

Algorithm	Flops	Error ( $E_1$ )	SNR (dB)			
			1	2	3	4
JADE	5,859,779	0.5902	18	26.1	42	28.5
InfoMax	88,848,925	0.3887	19.3	30	46.8	34.6
FP (tanh)	8,582,581	$6.661 \times 10^{-15}$	21.9	32	46.4	34.6
FP (pow3)	7,332,525	$9.326 \times 10^{-15}$	18.4	24.7	42	26.1
FP (gauss)	51,660,782	$9.992 \times 10^{-15}$	22.7	33	46.6	34.7

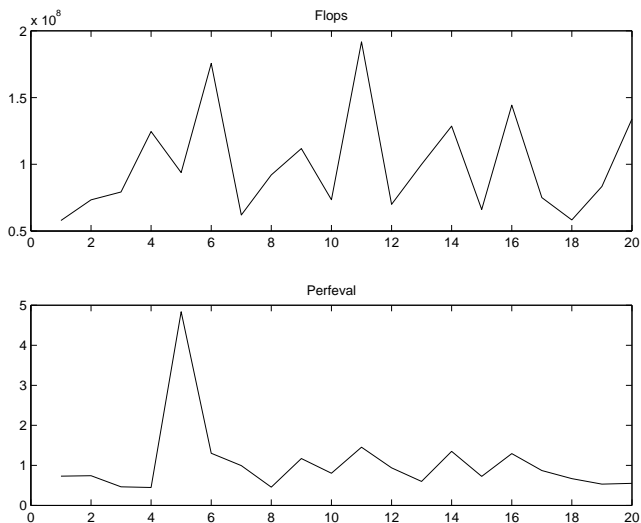
Table 2. Results for experiment 3.2.

The results of this experiment (summarised in Table 2) were closer to those expected. The infomax algorithm performed very good separation, but took approximately ten times more floating point operations than the more efficient algorithms, fixed-point (tanh & pow3). The fixed-point algorithm using the tanh non-linearity also made a sudden leap in computational cost. A big surprise was the performance of the fixed-point algorithms on the error distance metric. All of these came exceedingly close to identifying the mixing matrix perfectly (see Table 2). Interestingly this appeared to have little impact on the SNRs, indicating that increasing the accuracy of the separating matrix need not correspondingly increase the accuracy of separation. The main reason is probably the whitening process. All algorithms performed very good separation, with high SNR measurements and only small SNR differences between the algorithms. As expected from the high SNRs, an aural comparison of the outputs found the sources to be almost perfectly separated in every case. The only noticeable overlap was at the end of the “zap” sound effect where there was a significant period of silence, in which the classical music sample was faintly audible. The noticeable effect of the separation is that in all the cases the outputs are louder than the original sources.

### 3.3 Performance With Simulated Moving Sources

This experiment aimed to determine what degree of on-line learning is required to separate moving sources. If an unmixing matrix is learned for a particular microphone and source configuration, the system cannot continue to use the same unmixing matrix if the configuration changes as the configuration of the microphones and sources is implicitly described by the mixing matrix. As any changes to this configuration are reflected as changes to the mixing, then to the unmixing matrix (since  $\mathbf{W} = \mathbf{A}^{-1}$ ), it is expected that a real world implementation will need to calculate updates to the unmixing matrix according to its learning rule, and to perform the actual separation (ie.  $\mathbf{Ax}$ ).

In order to test this, an iterative scheme was developed where the unmixing matrix learned in the previous iteration was the initial guess for learning the unmixing matrix on the next iteration. Between iterations, the mixing matrix was updated by adding to it a matrix of small random numbers to simulate moving the sources. The algorithms tested in this experiment were the infomax algorithm with Amari's modified learning rule and the fixed-point algorithm using the symmetric approach and the tanh, cubic and exponential contrast functions. The JADE algorithm was not used as there is no provision for specifying an initial guess because learning is done in a statistical rather than a data domain. Performances were measured by the number of flops and the error distance  $E_1$ .



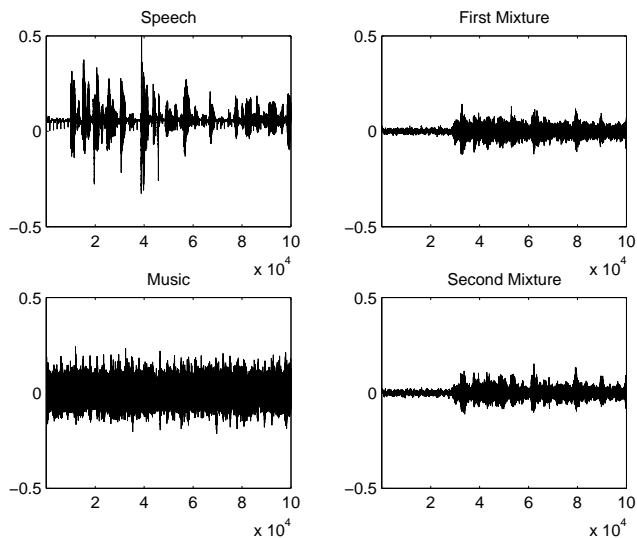
**Figure 3.** Results of 20 iterations of the infomax algorithm with simulated moving sources

The results from this experiment were surprising. It was hoped that after an initial performance hit, the computational load would settle to a lower level than that of the previous experiment and that the accuracy would remain relatively constant between iterations. However, the

results were not a lot different from running the algorithms with a random initial guess. The computational cost was inconsistent and fluctuated wildly. Separation quality was as good as the previous experiment, but no consistent advantage was gained by knowing the results of the previous separation. Figure 3 shows the results of one test using the infomax algorithm. The lower graph is the error distance, and shows it to be reasonably consistent and quite small. The upper graph shows the computational cost ranging over a 10,000,000 flop field.

### 3.4 Performance On Real Data With Real Mixing

In order to demonstrate the performances of the algorithms on real data, a set of experiments using real recordings were conducted. The data was recorded through two microphones in a standard office environment with two known sound sources: a rock music CD and a male voice. The recording was made at 16 bits/sample with sampling rate of 16,000Hz. Only two sources were used due to limited availability of microphones and amplification equipment at that stage. The microphones used were high quality condenser microphones provided by the Leon Audio Company. One microphone was mounted against a perspex surface to minimise the reception of reflected sounds (boundary microphone technique) and the second one (floating technique) had the microphone



**Figure 4.** Comparison of sources and mixes detected with the boundary microphone technique. The sources are on the left and the mixtures on the right.

hanging from the ceiling so that it would detect larger amounts of reflected sound. A comparison of the original sources and the mixtures received by the microphone in the boundary configuration is shown in Figure 4. The leading noise in the

mixtures is caused by the difficulty of starting the playback of two sources simultaneously.

The standard measures used to evaluate source separation in the previous experiments were found to be virtually useless in this experiment. For each algorithm, the error distance  $E_1$  could not be measured as the mixing matrix was unknown, but the SNR was uniformly low, even negative in some cases (indicating that the separation process actually *worsened* the sound quality). Aural and visual comparisons of the separated sources showed little change from the observed mixtures.

These results, though unsatisfactory, were not wholly surprising due to the recording environment for the data collection. The background noise and degree of convolution was significant as the recording environment was a typical office, complete with noisy computers and plenty of hard sound-reflecting surfaces. To have a good separation for any algorithm, a clean set of data is essential. Further experiments recording with more microphones may achieve better separation as noise sources may be separated out.

## 4 Discussions

One substantial difference between the three algorithms was the number of tuning parameters required to achieve a good separation performance. The implementation of the infomax algorithm used had around 15 tunable parameters, JADE had no parameters and the fixed-point variations had only two or three. Achieving good separation quality from infomax was highly dependent on tuning its many parameters. This makes the use of the infomax algorithm problematic in a real-world system as the changing environment means its parameters would need to be continuously updated in a sophisticated manner.

These experiments show that all the studied algorithms achieve excellent separation within the bounds of the basic mixing models. The infomax algorithm tends to be more expensive than the others, and had an anomalous failure in the first experiment. It is suspected that this was due to the differing gaussianity of the sources in this case. In that experiment, the sources had kurtosis in the range 2 - 4 while in the later experiments, the kurtosis tended to be around 15 - 40. In the development of infomax algorithm, it is assumed that the sources are non-gaussian [1], and it appears that this assumption affects the operation of the algorithm. The JADE and fixed-point algorithms were able to separate all instantaneous mixtures regardless of gaussianity.

In experiment 2, the silence at the end of the "zap" sound effect means that for the rest of the separation there were effectively only three sources. It is investigated that parts of the other sources "leaked" into the silence part of the separated "zap". It appears that current algorithms need to perform better in adapting to changing environments. Some extra facility is needed to detect the number of sources and

adjust accordingly.

Although experiment 3 is an ill-considered experiment for simulating the case of "moving sources", it is noted that the initial weights have little influence on the performance (such as training time, iterations *etc*) of a neural algorithm. Even using a correct unmixing weight matrix as an initial weights for a little changed mixed source signals, a similar training time and iterations are still needed for a similar accurate separated result. It could be explained by: (1) The optimization training surface for an algorithm is not very regular, and so the algorithm jumps from near one optimum to another. This is especially probable if real data is used. (2) Whitening changes the weight vectors, and most neural approaches are based on after whitening. Finally, none of the algorithms succeeded for real ordinary environmental recordings. Further work is needed to focus on this area.

## References:

- [1] A. J. Bell and T. J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, *Neural Computation*, Vol. 7, 1995, pp. 1129-1159.
- [2] P. Comon, Independent component analysis - a new concept ?, *Signal Processing*, Vol. 36, No.3, 1994, pp. 287-314
- [3] J-F Cardoso and P. Comon, Equivariant adaptive source separation, *IEEE Transactions on Signal Processing*, Vol. 45, No. 2, 1996, pp. 434-444.
- [4] A. Hyvärinen, A family of fixed-point algorithms for independent component analysis. *IEEE Conference on Acoustics, Speech, and Signal Processing*, 1997, 3917-3920.
- [5] S. Amari, A. Cichocki and H. H. Yang, A new learning algorithm for blind signal separation, *Advances In Neural Information Processing Systems*, Vol. 8, 1996, pp. 752-763, MIT Press.
- [6] J. F. Cardoso, and A. Souloumiac, Blind beamforming for non-Gaussian signals. *IEE Proceeding Part F*, Vol.140, No. 6, 1993, pp. 362-370.
- [7] A. Hyvärinen E Oja, A fast fixed-point algorithm for independent component analysis, *Neural Computation*, Vol. 9, 1997, pp.1482-1492
- [8] J. F. Cardoso, High-order contrasts for independent component analysis, *Neural Computation*, Vol. 11, No. 1, 1999, pp. 157-192.
- [9] X. Giannakopoulos, Comparison of adaptive independent component analysis algorithms, *Master's thesis, Helsinki University of Technology*, 1998.
- [10] A. G. Westner, Object based audio capture: separating acoustically mixed sounds, *Master's thesis, MIT*, 1999.