Włodzimierz KASPRZAK
Warsaw University of Technology
Institute of Control and Computation Engineering
ul. Nowowiejska 15 –19, PL - 00-665 Warszawa, Poland
Phone: +48 – 22 – 660 7866. Fax: +48 – 22 – 825 – 3719
W.Kasprzak@ia.pw.edu.pl

BLIND SOURCE DECONVOLUTION BY HOMOMORPHIC FILTERING IN FOURIER SPACE

Abstract.
An approach to multi-channel blind deconvolution is developed, which uses an adaptive filter that performs blind source separation in the Fourier space. The approach keeps (during the learning process) the same permutation and provides appropriate scaling of components for all frequency bins in the frequency space. Experiments verify a proper blind deconvolution of convolution mixtures of sources.

Keywords: *Adaptive filter, Image filtering, Source separation, Unsupervised learning.*

1 INTRODUCTION

Source signals, like in speech, seismology or medicines get mixed and distorted if they are transmitted over disperse environment. The simplest case of a mixing model is an *instantaneous (linear) mixing* of source signals [1, 2, 5], but this is a practically no feasible model. In general, the nature of the transmission environment is dynamic and nonlinear. The goal of *blind source deconvolution* is to reconstruct from many distorted signals the estimates of original sources [2]. Some ambiguity is inherent, i.e. the permutation order, the scaling and delay factors cannot be reliably predicted. 1-D signals are the main application field of blind signal processing techniques. But there appear some possible applications in image processing as well [2, 4]: (1) the extraction of sparse binary images (e.g. documents), (2) contrast strengthening of “smoothed” images in selected regions, (3) encryption of transmitted images.

In this paper we solve the source deconvolution problem by repetitive use of blind source separation method in the frequency space. It is difficult to combine the independently learned weights for all frequency bins into one learning process. In our approach we avoid non-compatible output permutations and different component scales for different frequencies.

2 THE BSS/MBD PROBLEMS

The blind source separation task. Assume that there exist m zero-mean source signals, $s_1(t)$, ... ,$s_m(t)$, that are scalar-valued and mutually (spatially) statistically independent (or as independent as possible) at each time instant or index value t. The original sources $s_j\,(t)$ are unknown to the observer, who has to deal with n possibly noisy but different linear mixtures, $x_1(t)$,..., $x_n(t)$, of the sources (usually for n >= m). The mixing coefficients are some unknown constants. The task of blind source separation (BSS) is to find the waveforms $\{s_I\,(t)\}$ of the sources, knowing only the mixtures $x_j(t)$ and the number m of sources [1, 2]. Denote by $\mathbf{x}(t)= [x_1(t),...,x_n(t)]^T$ the n-dimensional t-th mixture data vector, at discrete index value (time) t. The BSS mixing model is equal to:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) = \sum_{i=1}^{m} s_i(t) a_i + \mathbf{n}(t) \tag{1}$$

Let us assume further that in the general case the noise signal has a Gaussian distribution but none of the sources is Gaussian. In standard neural source separation approach, an $m \times n$ separating matrix $\mathbf{W}(t)$ is updated so that the m-vector,

$$\mathbf{y}(t) = \mathbf{W}(t)\, \mathbf{x}(t), \tag{2}$$

becomes an estimate of the original independent source signals. $\mathbf{y}(t)$ is the output vector of the network and the matrix $\mathbf{W}(t)$ is the total weight matrix between the input and output layers. The rank of the mixing matrix must be at least equal to the number of sources and the number of outputs is at least equal to the number of independent sources

Gradient based optimization. A well-known iterative optimization method is the *stochastic gradient* (or *gradient descent*) search [3]. In this method the basic task is to define a criterion $J(\mathbf{W}(k))$, which obtains its minimum for some $\mathbf{W}_{opt}$ if this $\mathbf{W}_{opt}$ is the expected optimum solution. The iterative rule in gradient descent search computes the $\mathbf{W}(k+1)$ by moving from $\mathbf{W}(k)$ along the gradient descent, i.e. $-\nabla J(\mathbf{W}(k))$:

$$\mathbf{W}(k+1) = \mathbf{W}(k) - \eta(k)\, \nabla J(\mathbf{W}(k)), \tag{3}$$

where $\eta(k)$ is a positive-valued step-scaling coefficient. For the BSS problem another gradient approach was developed recently – the *natural gradient descent* [1, 2]. It takes the form:

$$\mathbf{W}(l+1) = \mathbf{W}(l) - \eta(l) \frac{\partial E\{J(\mathbf{W})\}}{\partial \mathbf{W}} \mathbf{W}^T(l) \cdot \mathbf{W}(l)\,. \tag{4}$$

There exist different theoretical justifications of the BSS, like the Kullback-Leibler divergence minimization, the information maximization and the mutual information minimization. All of them lead to the same cost function [2, 6]:

$$J(\mathbf{W}, y) = -\log \det(\mathbf{W}) - \sum_{i=1}^{n} \log p_i(y_i)\,, \tag{5}$$

where the $p_i(y_i)$-s are pdf's of signals y_i respectively, $\det(\mathbf{W})$ - the determinant of matrix $\mathbf{W}$.

Applying the natural gradient approach we may derive the learning rule for BSS:

$$\Delta\mathbf{W}(t) = -\eta(t) \frac{\partial \mathrm{E}\{J(\mathbf{W}(t))\}}{\partial \mathbf{W}(t)} \cdot \mathbf{W}^T(t) \cdot \mathbf{W}(t) = \eta(t)\left[\mathbf{I} - \left\langle f(\mathbf{y}(t)) \cdot \mathbf{y}^T(t) \right\rangle\right]\mathbf{W}(t)\,. \tag{6}$$

The MBD problem. The *multi-channel blind deconvolution problem* (MBD) can be considered as a natural extension of the instantaneous blind separation problem (BSS). An m-dimensional vector of sensor signals (in discrete time), $\mathbf{x}(k) = [x_1(k), \dots ,x_m(k)]^T$, is assumed to originate from an n-dimensional vector of source signals, $\mathbf{s}(k) = [s_1(k), \dots , s_n(k)]^T$ ($m > n$), as:

$$\mathbf{x}(k) = \sum_{p=-\infty}^{\infty} \mathbf{H}_p \, \mathbf{s}(k-p) = \mathbf{H} * \mathbf{s}(k) = \mathbf{H}(z)[\mathbf{s}(k)], \tag{7}$$

where $\mathbf{H} = \{\mathbf{H}_p\}$ is a set of (m× n) matrices of mixing coefficients at lag p, which represents the time-domain impulse response of the mixing filter. The BSS feed-forward network can now be generalized to a deconvolution filter with the impulse response $\mathbf{W} = \{ \mathbf{W}_p \}$:

$$\mathbf{y}(k) = \mathbf{W} * \mathbf{x}(k) = \sum_{p=-\infty}^{\infty} \mathbf{W}_p \mathbf{x}(k-p) = \mathbf{W}(z)[\mathbf{x}(k)] \tag{8}$$

3 THE BSS-FT METHOD

Homomorphic filter. A *linear filter* is useful for noise reduction or signal feature extraction if the signal is distorted by additive noise. The *homomorphic* system is a generalization of a *linear* system. It is more useful to use such a system than a linear filter if the signals are combined in a non-additive fashion, like the convolution operation. The approach of homomorphic systems consists of following steps:

1. Transform a non-additive combination into an additive one by applying a transformation T_c (e.g. $\mathbf{X}(\omega, \kappa) = T_c(\mathbf{x}(k)) = \log [FT [\mathbf{x}(k)]]$), called the *characteristic transformation*.
2. Perform a linear transformation T_L (e.g. some linear filter transform, for example the BSS transform) of the transformed signal $\mathbf{X}$ (e.g. $\hat{\mathbf{Y}}(\omega, \kappa) = \hat{\mathbf{W}}(\omega)\, \mathbf{X}(\omega, \kappa)$)
3. Make signal inversion by T_c^{-1} (e.g. $T_c^{-1}(\hat{\mathbf{Y}}) = FT^{-1}[\exp[\hat{\mathbf{Y}}]]$).

Hence, the sequence of transformations is:

$$\mathbf{y}(k) = T_c^{-1}[T_L[T_c[\mathbf{x}(k)]]]. \tag{9}$$

Fourier space. The homomorphic filter uses the well-known principle that a convoluted mixture in the time domain corresponds to an instantaneous mixture of complex-valued signals in the frequency domain. We shall use a $2L$-point Fast Fourier Transform to convert each time domain signal $x_i(t)$ into a series of Fourier coefficients $\{ X_i(\omega, \kappa) \}$ in the frequency space

$$X_i(\omega,\kappa) = \sum_{k=0}^{2L-1} x_i(k) e^{-j\omega} \cdot w(k - \kappa\Delta), \qquad \text{with: } \omega = 0, \frac{1\cdot 2\pi}{2L}, \frac{2\cdot 2\pi}{2L}, \dots, \frac{(N-1)\cdot 2\pi}{2L}, \tag{10}$$

where w is a window function with $2L$ nonzero elements and Δ is a shift interval between consecutive window positions. The number of coefficients is equal to $2L$ and all the frequencies ω are multiplies of the basic frequency $2\pi / 2L$.

The learning (adaptive filtering) process. Let $L = 2^p$ be the basic length of samples in one block and at the same time the number of time-delayed filter weights in each channel. In order to avoid end effects of the Fourier Transform, we shall use a $2L$-point FFT, with half of the samples padded to zero. The impulse response of some ij-th channel is:

$$\hat{w}_{ij} = FFT([w_{ij1},...,w_{ijL},0,...,0]) . \quad (11)$$

The sensor vector data are grouped to blocks, indexed by κ, where each block $\mathbf{x}^B(\kappa)$ contains *L* (vector) samples, up to time index *k*:

$$\mathbf{x}^B(\kappa) = (\mathbf{x}(k\text{-}L\text{+}1), \dots , \mathbf{x}(k)),\ \ \mathbf{x}^B(\kappa\text{-}1) = (\mathbf{x}(k\text{-}2L\text{+}1), \dots , \mathbf{x}(k\text{-}L)), \quad (12)$$

The block of output signals, which are computed in the κ-th iteration of the learning process, contains the samples indexed by time instants up to *k*:

$$\hat{\mathbf{X}}(\kappa) = \mathrm{FFT}([\mathbf{x}(\kappa-1), \mathbf{x}(\kappa)]) . \quad (13)$$

The output signals in the frequency space are calculated next:

$$\hat{Y}(\kappa) = \hat{\mathbf{W}}(\kappa) \,.\!*\, \hat{\mathbf{X}}(\kappa) , \quad (14)$$

where the operation “.*” means that we perform a set of matrix multiplications, one multiplication for each single frequency bin ω. Now we could apply the nonlinear function (e.g. $f(y) = y^3$) to current $\hat{\mathbf{Y}}(\kappa)$ in the frequency space, but this leads to independent learning (permutation and scaling) of weights for each frequency bin. The proper way is to transform the frequency output to the time domain. Hence, the output signals are calculated next:

$$[\mathbf{y}(\kappa-1), \mathbf{y}(\kappa)] = \mathrm{FFT}^{-1}(\hat{Y}(\kappa)) . \quad (15)$$

Finally, the block of transformed output signals in the time domain is transformed again into the Fourier space:

$$\hat{f}(\hat{\mathbf{Y}}(\kappa)) = \mathrm{FFT}[f(\mathbf{y}(\kappa-1)), f(\mathbf{y}(\kappa))] . \quad (16)$$

<u>The BSS learning (update) rule in Fourier space.</u> Now, for each frequency bin ω one can apply the BSS update rule in order iteratively to learn the matrix $\hat{\mathbf{W}}(\omega)$ and to estimate the output signals (in frequency space) $\hat{\mathbf{Y}}(\omega, \kappa)$. One of such learning (weight update) rules is based on the principle of natural gradient (equation 4) and it takes the form:

$$\hat{W}(\omega, \kappa+1) = \hat{W}(\omega, \kappa) + \eta \left[\Gamma(\omega) - f(\hat{Y}(\omega, \kappa)) \cdot \hat{Y}^H(\omega, \kappa) \right] \hat{W}(\omega, \kappa) \ \ \text{, for all } \omega, \quad (17)$$

where the superscript *H* denotes the Hermitian conjugate. In order to keep the balance between signal component energies in particular frequency bandwidths the learning process converges to different values Λ(ω), (for all ω), that are put into relation to each other:

$$\Lambda(\omega_p) = \frac{E\{\langle X(\omega_p, \kappa) \cdot X(\omega_p, \kappa) \rangle\}}{E\{\langle X(\omega_0, \kappa) \cdot X(\omega_0, \kappa) \rangle\}} , \qquad (p = 0, 1, 2, \dots , L\text{-}1). \quad (18)$$

From above coefficients we form an appropriate set of diagonal matrices:

$$\Gamma(\omega_p) = \begin{bmatrix} \Lambda(\omega_p) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Lambda(\omega_p) \end{bmatrix} , \qquad (p = 0, 1, 2, \dots , L\text{-}1). \quad (19)$$

4 EXPERIMENTAL RESULTS

The approach described in section 3 was implemented in Matlab and it was tested on some examples of image sources (Fig. 1). Obviously 1-D signals can also be applied as well, but the image data gives better and more impressive illustration of the results. From Fig. 2 it can be seen that difficult, convoluted mixtures were computed. The three sensor signals in both tests were nearly the same, i.e. the cross-correlation factor of pairs of mixed signals was in the range of 95 – 98%.

The results of the blind source deconvolution process, applied to these mixtures in two separate tests, are shown on Fig. 3. The three synthetic sources correspond to step- and sinusoidal signals and they are only to some small amount cross-correlated. Hence, for them a high quality deconvolution effect was achieved. In opposite, the natural sources are strongly cross-correlated, which is against the theory requirements of source independency. The deconvolution results in this case are only of average quality, but they show appropriately the limitations of the blind processing theory (related to unsupervised learning).

ACKNOWLEDGMENTS

This work was supported by the *Research Center for Control and Information-Decision Technology* (CATID) at Warsaw University of Technology, Poland.

REFERENCES

[1] S. Amari, S.C. Douglas, A. Cichocki, H.Y. Yang: *Novel on-line adaptive learning algorithms for blind deconvolution using the natural gradient approach*. IEEE Signal Proc. Workshop on Signal Processing Advances in Wireless Communications, April 1997, Paris, 107–112.

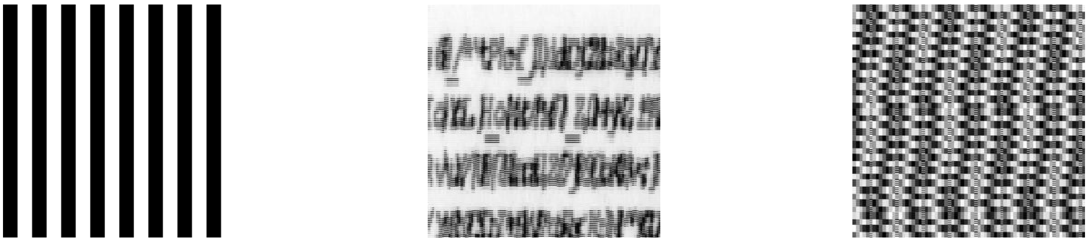
[2] A. Cichocki, S. Amari: *Adaptive Blind Signal and Image Processing*, John Wiley, Chichester, UK, 2002.

[3] R.O. Duda, P.E. Hart: *Pattern classification and scene analysis*, John Wiley & Sons, New York, 1973.

[4] W. Kasprzak, A. Cichocki: *Hidden Image Separation From Incomplete Image Mixtures by Independent Component Analysis*, 13th Int. Conf. on Pattern Recognition, ICPR'96, Proceedings. IEEE Computer Society Press, Los Alamitos CA, 1996, vol. II, 394-398.

[5] W. Kasprzak, A. Cichocki, S. Amari: *Blind Source Separation with Convolutive Noise Cancellation*, Neural Computing & Applications, Springer, London, 6(1997), 127-141.

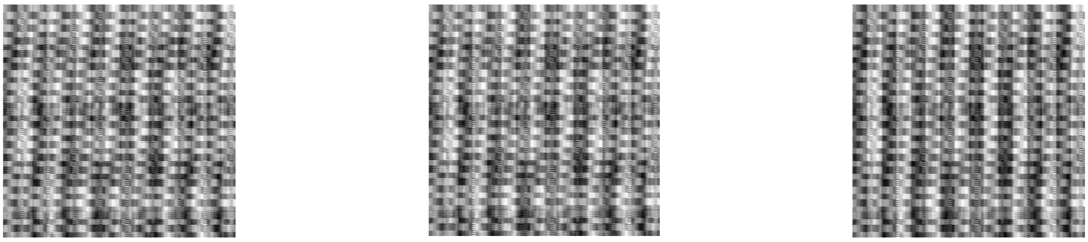
[6] I. Sabała: *Multichannel Deconvolution and separation of statistically independent signals for unknown dynamic systems*. Ph.D. Thesis, Warsaw University of Technology, Department of Electrical Engineering, Warsaw, 1998.



Test 1: three synthetic image sources

Test 2: three natural image sources

Fig. 1: Original sources (assumed to be unknown).



Convoluted mixtures (input signals)

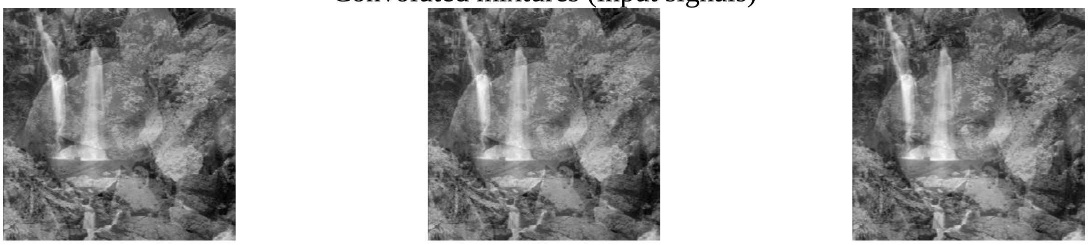
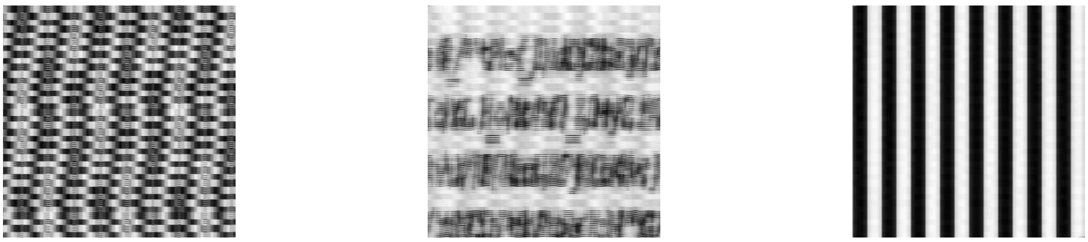


Fig. 2: Examples of sensor signals.



Deconvolved (output) signals

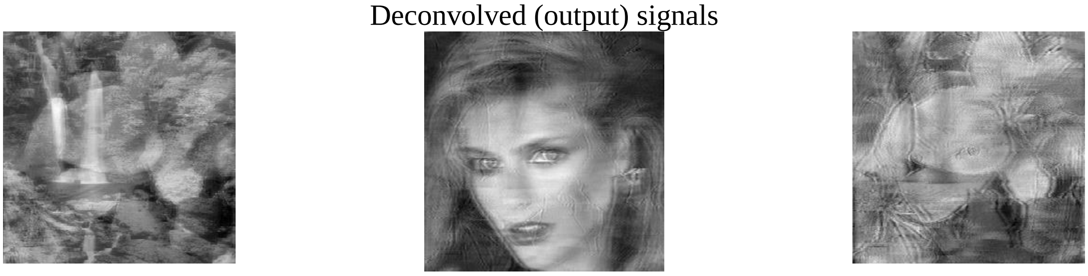


Fig. 3: Examples of output signals.