International Journal of Neural Systems, Vol. 8, No. 2 (April, 1997) 219–237

ON NEURAL BLIND SEPARATION WITH NOISE SUPPRESSION AND REDUNDANCY REDUCTION

J. KARHUNEN*,†, A. CICHOCKI‡,§, W. KASPRZAK‡,¶, P. PAJUNEN†
†*Helsinki University of Technology, Laboratory of Computer and Information Science, Rakentajanaukio 2 C, P.O. Box 2200, FIN-02015 HUT, Finland*
‡*Frontier Research Program RIKEN, Laboratory for Artificial Brain Systems, Hirosawa 2-1, Saitama 351-01, Wako-shi, Japan*

Received 5 January 1997
Accepted 28 April 1997

Noise is an unavoidable factor in real sensor signals. We study how additive and convolutive noise can be reduced or even eliminated in the blind source separation (BSS) problem. Particular attention is paid to cases in which the number of sensors is larger than the number of sources. We propose various methods and associated adaptive learning algorithms for such an extended BSS problem. Performance and validity of the proposed approaches are demonstrated by extensive computer simulations.

1. Introduction

Blind source separation (BSS) has recently become an active research area both in statistical signal processing and unsupervised neural learning.[1–28] The goal of BSS is to extract statistically independent but otherwise unknown source signals from their linear mixtures without knowing the mixing coefficients. BSS techniques have many potential applications in, for example, data communications, speech processing, and medical signal processing.

Although many neural learning algorithms have been proposed for the BSS problem, in their corresponding models and network architectures it is usually assumed that the number of source signals is known *a priori*. Typically, it should be equal to the number of sensors and outputs of a neural network (separating system). However, in practice these assumptions do not necessarily hold. The problem of on-line determining the number of sources has been considered only recently.[11,12] In most neural approaches to BSS, published in available literature, it is also assumed that there is no noise present in the signal model or that noise can be neglected. However, noise is an unavoidable factor in real-world applications.[16,19,33].

In this paper we extend the basic BSS problem by discussing cases where the number of sensors (inputs of the separation network) is different from the (generally unknown) number of source signals, and where additive noise is present. The question is then how to separate the sources and determine their correct number in a noisy environment.

We propose and study the performance of some network structures in the case of instantaneous mixtures with additive noise. We start in Sec. 2 with the definition of the extended BSS problem. In Secs. 3 and 4 we propose two classes of solutions for the extended mixing model: The first one is based on source separation with pre-whitening, whereas the

*E-mail: Juha.Karhunen@hut.fi
§Author to whom correspondence should be addressed.
On leave from Warsaw University of Technology, Department of Electrical Engineering, Warsaw, Poland.
E-mail: cia@hare.riken.go.jp
¶Current affiliation: Warsaw University of Technology, Institute of Control and Computational Engineering, Warsaw, Poland.

second one tries to cancel noise before separating the sources.[16] For general additive colored noise, it is impossible to separate the unknown noise from the unknown source signals. Therefore, the proposed solutions are based on the assumptions that either the noise has Gaussian distribution (Sec. 3), or that some *a priori* knowledge about the reference noise itself is available (Sec. 4).

2. The Extended BSS problem

2.1. *The mixing model*

Denote by $\mathbf{x}(t) = [x_1(t), \ldots, x_n(t)]^T$ the data vector made up of the n mixtures $x_i(t)$, $i = 1, \ldots, n$, at discrete time or sample value t. The data (mixing) model in BSS can then be written in the vector form as (see Fig. 1):

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) = \sum_{i=1}^{m} s_i(t)\mathbf{a}_i + \mathbf{n}(t)\,. \qquad (1)$$

Here $\mathbf{s}(t) = [s_1(t), \ldots, s_m(t)]^T$ is the source vector consisting of the m unknown source signals at time t. Furthermore, each source signal $s_i(t)$ is assumed to be a stationary zero-mean stochastic process.

$\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]$ is a constant full-rank $n \times m$ mixing matrix whose elements are the unknown coefficients of the mixtures. The vectors $\mathbf{a}_i$ are basis vectors of *Independent Component Analysis* (ICA).[20,24]

None of the source signals $s_i(t)$ are allowed to have a Gaussian distribution if the noise signal $\mathbf{n}(t)$ itself is a Gaussian signal. Otherwise at most one of the sources can be a Gaussian signal. This restriction follows from the fact that it is impossible to separate two or more Gaussian sources from each other. It is also customary to assume in BSS problems that the source signals $s_1(t), \ldots, s_m(t)$ are mutually statistically independent.[1,2,17] In practice, the sources can often be successfully separated even though they are not strictly independent.

2.2. *Neural blind source separation*

In neural and adaptive source separation approaches, a $m \times n$ separating matrix $\mathbf{W}(t)$ is updated so that the m-vector (see Fig. 1)

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

becomes an estimate

$$\mathbf{y}(t) = \hat{\mathbf{s}}(t) \qquad (3)$$

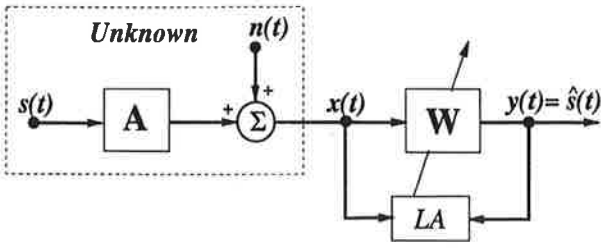


Fig. 1. A schematic diagram of the mixing and blind separation problem. LA means learning algorithm.

of the original source signals. In neural realizations, $\mathbf{y}(t)$ is the output vector of the single-layer feedforward network, and the matrix $\mathbf{W}(t)$ is the total weight matrix between the input and output.

Neural (and adaptive) source separation methods can be divided into two major groups. In the first type of methods, the data vectors $\mathbf{x}(t)$ are prewhitened by using a suitable whitening matrix $\mathbf{V}(t)$:

$$\mathbf{v}(t) = \mathbf{V}(t)\mathbf{x}(t)\,, \quad \text{with} \quad \mathbf{R}_{\mathbf{vv}} = \mathrm{E}[\mathbf{v}(t)\mathbf{v}^T(t)] = \mathbf{I}\,. \qquad (4)$$

After this, the sources are separated by employing a linear network described by:

$$\mathbf{y}(t) = \tilde{\mathbf{W}}(t)\mathbf{v}(t)\,, \qquad (5)$$

where the separating matrix is now denoted by $\tilde{\mathbf{W}}(t)$ for clarity. Pre-whitening has the advantage that the subsequent separation stage becomes somewhat easier. This is because the prewhitened mixtures in $\mathbf{v}(t)$ are already uncorrelated, and the separating matrix $\tilde{\mathbf{W}}(t)$ can be constrained to be orthogonal. On the other hand, whitening can reduce or amplify the noise in certain cases, and separation may be difficult or impossible for ill-conditioned mixing matrices $\mathbf{A}$ and/or weak source signals $s_i(t)$. Therefore, in the other major group of neural methods, the separating matrix $\mathbf{W}(t)$ is sought directly without using pre-whitening.

We do not deal with various neural separating algorithms in more detail here. They are discussed to the necessary extent in several other papers,[1–4,6,10,24,27] and the references of the present paper cover the most significant neural approaches to BSS.

2.3. *Estimation of the number of the sources*

In this paper we shall assume for the most of the time that the actual number of sources m is unknown. We also assume that there exist more sensors than sources, that is $n > m$. In the contrary case where there are less mixtures than sources ($n < m$), there exist to our knowledge very few algorithms that are able to handle this situation in some special cases, e.g. for binary source signals.[5,28,30]

The first approach is applicable to two-stage separation that requires a pre-whitening layer. A schematic diagram of the network structure is shown in Fig. 2. Whitening can be done in many ways.[10,23,24] If the number m of the sources is unknown, it is advantageous to use standard or robust principal component analysis (PCA)[7,15,22] for whitening and data compression, because this simultaneously yields an estimate for m. PCA can also filter out some of the additive Gaussian noise in the case $n > m$. We shall discuss this method in more detail in the next section, extending it to the more difficult and relevant noisy case.

It is also possible to compress the dimensionality of the data from n to the desired m later on in the separation layer instead of the whitening layer. However, this network structure (which is a modification of the network in Fig. 2) is not recommendable if additive noise is present. This is because whitening typically amplifies noise by transforming the variances of all the n components of the whitened vectors $\mathbf{v}(t)$ equal to unity. Thus the separation results in the noisy case are worse than for the network of Fig. 2.

A third approach has been proposed in Ref. 11 (see also Ref. 3) for handling the situation where the number of sources is completely unknown. Here a post-processing layer for elimination of redundant signals is added to the separation network. The number of active sources is determined indirectly by counting the number of non-zero output channels from the final layer. This approach can be applied to the noisy case, too. Thus the applied neural network consists of two or more layers, where the first subnetwork (a single-layer or a multi-layer) separates the sources in parallel manner, and the last post-processing layer eliminates the redundant signals. In this paper we shall investigate how this approach behaves under additive and convolutional noise.

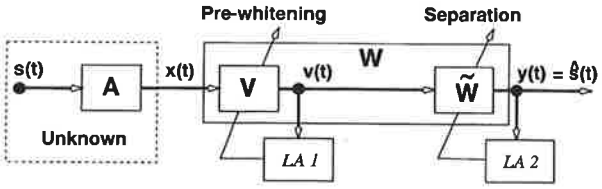


Fig. 2. A schematic diagram of signal compression during pre-whitening.

2.4. *Separation in additive Gaussian noise*

Consider now the effect of additive noise on blind source separation. Such noise is often unknown, and can for example describe the possible imperfections in the data model (1). In most papers dealing with BSS, it is assumed that the noise term $\mathbf{n}(t)$ in (1) is zero, and sometimes noise is regarded as an extra source.[14] Generally, this does not hold for the signal model (1), which can be seen by expressing the noise vector $\mathbf{n}(t)$ in the form

$$\mathbf{n}(t) = \sum_{i=1}^{m} n_i(t)\mathbf{a}_i + \mathbf{e}(t)\,. \tag{6}$$

Here $n_i(t)$ is the projection of $\mathbf{n}(t)$ onto the ith ICA basis vector $\mathbf{a}_i$, and $\mathbf{e}(t)$ denotes the portion of the noise vector $\mathbf{n}(t)$ that lies in the subspace orthogonal to the m basis vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ of ICA. Inserting (6) into the data model (1) shows that generally noise adds a component $n_i(t)$ to each source.

2.5. *Separation in additive convolutional noise*

In this paper we also investigate a special case of colored noise. In many practical situations we can measure or model the environmental noise. We shall denote in the following such noise as *reference noise* $n_R(t)$ or a vector of reference noises [for each separate sensor $n_i(t)$] (Fig. 3). We assume here that the noise obeys *Finite Impulse Response* (*FIR*) model, which describes additive and convolutional noise.[16,19,33] [Fig. 3(b)]:

$$n_i(t) = \sum_{j=0}^{N} b_{ij} z^{-j} n_R(t) = \sum_{j=0}^{N} b_{ij} n_R(t - jT)\,, \tag{7}$$

where $z^{-1} = e^{-sT}$ is the unit delay. Such a model is generally regarded as a realistic (real-world) model

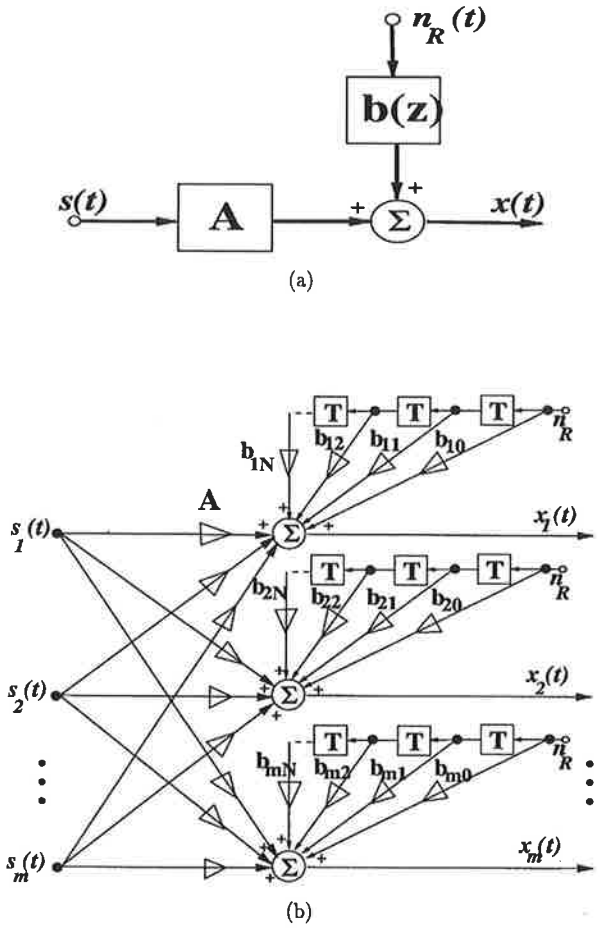


Fig. 3. The source mixture with additive noise and the simple FIR noise model: (a) source mixture with additive noise vector, (b) modeling the unknown noise by a convolutional additive noise (n_R is the reference noise, the coefficients b_{ij} are unknown).

in both signal and image processing.[19,33] In this model we assume that known reference noise is added to each sensor (mixture of sources) with different unit delays T and various but unknown coefficients $b_{ij}(t)$. In other words, we assume that noise is convolutional and the data model (1) now becomes[16]

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{b}(z)n_R(t)\,, \tag{8}$$

where $\mathbf{b}(z) = [b_1(z), b_2(z), \ldots, b_n(z)]^T$ with

$$b_i(z) = b_{i0} + b_{i1}z^{-1} + \cdots + b_{iN}z^{-N}\,. \tag{9}$$

3. PCA Based Separation in Additive Gaussian Noise

Our general observation with separation algorithms, designed for the basic BSS problem (without noise), was that they can tolerate small amounts of unknown noise. However, noise destroys nice theoretical properties of some algorithms such as independence of the condition number of the mixing matrix $\mathbf{A}$ and the ability to separate even very weak sources.[2,6,10] When the amount of noise increases, performance of the separation algorithms begins to degrade rapidly. If the power of the noise is large, separation algorithms typically yield output signals that are useless in practice. Maybe one or two of the output signals somehow resemble some source signal while the others are completely unrecognizable.

3.1. *PCA based pre-whitening in noisy conditions*

If the number n of mixtures is greater than the number m of sources, it is usually possible to filter some of the noise out. This can be done by projecting the input vectors $\mathbf{x}(t)$ onto their m-dimensional signal subspace,[29] which is in our case defined as the subspace spanned by the m basis vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ of ICA. Now these basis vectors and thus the signal subspace are generally unknown. A standard and in practice often the best way to estimate the signal subspace is to use PCA.

3.1.1. *Standard procedure*

The practical procedure is as follows. First, estimate the $n \times n$ covariance matrix $\mathbf{R}_{xx} = E\{\mathbf{x}(t)\mathbf{x}(t)^T\}$ of the zero-mean data vectors $\mathbf{x}(t)$. Then compute the n eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ and the respective eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ of $\mathbf{R}_{xx}$. These eigenvalues and eigenvectors define PCA. The subspace spanned by the m first PCA eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$ is an approximation to the true signal subspace. This PCA subspace is optimal in the sense that it provides the best lower dimensional approximation of the data vectors. More specifically, the projection of the data vectors $\mathbf{x}(t)$ onto the PCA subspace has the least the mean-square error among

all the subspaces of the same dimensionality. Connections between the signal subspace and the PCA subspace are explained in more detail in Refs. 23 and 29.

The data vectors $\mathbf{x}(t)$ can be conveniently projected onto the estimated signal subspace in context with PCA pre-whitening. The PCA whitening matrix $\mathbf{V}$ is given by

$$\mathbf{V} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T, \tag{10}$$

where $\mathbf{\Lambda} = \text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_m]$ is the diagonal matrix containing the m largest PCA eigenvalues, and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]$ is the orthogonal matrix that contains the respective PCA eigenvectors. It is easy to see that the covariance matrix $\mathbf{R}_{vv}$ of the whitened vectors $\mathbf{v}(t) = \mathbf{V}(t)\mathbf{x}(t)$ is then $m \times m$ unit matrix, as required in whitening.

3.1.2. *Neural and robust PCA*

Instead of computing the PCA whitening matrix (10) using standard numerical software, one can use neural approaches for PCA whitening.[23] They are, however, usually less accurate due to the stochastic gradient learning rules used, and may require long convergence times especially if the number of sources m is large.

It is more convenient to use a robust PCA approach for extraction of principal components, corresponding to useful signals.[7,15] In this approach, principal components are extracted sequentially as long as the eigenvalues λ_i are larger than some chosen threshold value.

We assume that minor components for $i > m$ correspond to additive noise. Using the concept of self-supervising principle (replication) and cascade (hierarchical) neural network architecture we can easily derive the following robust learning algorithm[7,15]:

$$\mathbf{u}_i(t+1) = \mathbf{u}_i(t) + \eta_i(t)\hat{v}_i(t)\Psi[\mathbf{e}_i(t)], \quad (i = 1, 2, \ldots, n), \tag{11}$$

where

$$\mathbf{e}_i = \mathbf{e}_{i-1} - \mathbf{u}_i\hat{v}_i, \quad \mathbf{e}_0(t) = \mathbf{x}(t), \tag{12}$$

$$\hat{v}_i(t) = \mathbf{u}_i^T\mathbf{e}_{i-1}, \tag{13}$$

$$\Psi(\mathbf{e}_i) = [\Psi(e_{i1}), \Psi(e_{i2}), \ldots, \Psi(e_{in})]^T, \tag{14}$$

$$\Psi(e_{ij}) = \frac{\partial\rho(e_i)}{\partial e_{ij}}, \tag{15}$$

e.g. $\Psi(e_{ij}) = \tanh(e_{ij}/\beta)$, $\beta > 0$, for the loss function[9]:

$$\Psi(\mathbf{e}_i) = \beta \ln(\cosh(\mathbf{e}_i/\beta)).$$

The optimal choice of the activation function $\Psi(\mathbf{e}_i)$ depends on the distribution of noise. For Gaussian noise, the linear function $\Psi(\mathbf{e}_i) = \mathbf{e}_i$ is optimal. In this case the above algorithm simplifies to the well-known Oja's rule (for $i = 1$):

$$\mathbf{u}_i(t+1) = \mathbf{u}_i(t) + \eta_i(t)\hat{v}_i(t)\mathbf{e}_i(t). \tag{16}$$

In contrast to Sanger's GHA algorithm (see e.g. Refs. 9 and 22) we extract signals using a deflation technique, i.e. $\hat{v}_i = \mathbf{u}_i^T\mathbf{e}_{i-1}$ but not $\hat{v}_i = \mathbf{u}_i^T\mathbf{x}$. This lead to a more stable and accurate algorithm.[7,15]

The output signals $\hat{v}_i(t)$ after applying the above learning procedure will be uncorrelated with variances $\lambda_i = E[\hat{v}_i^2]$, $(i = 1, 2, \ldots, n)$. In order to normalize them to unit variance we can apply the following procedure

$$v_i(t) = \lambda_i^{-\frac{1}{2}}\hat{v}_i(t) = \lambda_i^{-\frac{1}{2}}\mathbf{u}_i^T(t)\mathbf{e}_i(t). \tag{17}$$

Some other approaches and references to robust PCA can be found in Ref. 22. Various robust loss or criterion functions are discussed in more detail in Refs. 7, 9, 15 and 21.

3.1.3. *Noise reduction during pre-whitening*

We can analyze the ability of PCA to present signal information in noisy conditions somewhat by assuming that the noise term $\mathbf{n}(t)$ in (1) is uncorrelated with the sources $s_1(t), \ldots, s_m(t)$, which are assumed to be mutually independent (or at least uncorrelated with each other). Then the data covariance matrix has the form

$$\mathbf{R}_{xx} = \mathbf{R}_{ss} + \mathbf{R}_{nn} = \sum_{i=1}^{m} \mathrm{E}\{s_i(t)^2\}\mathbf{a}_i\mathbf{a}_i^T + \mathbf{R}_{nn}, \tag{18}$$

where $\mathbf{R}_{nn}$ is the covariance matrix of the noise vector $\mathbf{n}(t)$, and $\mathbf{R}_{ss}$ is the covariance matrix of the signal part. If we first assume that the noise vector $\mathbf{n}(t)$ and consequently $\mathbf{R}_{nn}$ are zero, it is easy to

see[29] that only the m largest PCA eigenvalues are nonzero, and the corresponding PCA eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$ lie exactly in the signal subspace. If the power of noise is small so that $\mathbf{R}_{nn}$ is "small" compared to $\mathbf{R}_{ss}$, the m first PCA eigenvectors still approximate well the true signal subspace.

A common special case occurs when the components of the zero-mean noise vector $\mathbf{n}(t)$ are mutually uncorrelated and have all equal variances σ^2 (for example in the case of white Gaussian noise). Then the noise covariance matrix

$$\mathbf{R}_{nn} = \sigma^2 \mathbf{I}, \tag{19}$$

where $\mathbf{I}$ is the unit matrix. It is easy to see that then the noise term does not affect to the directions of the PCA eigenvectors, and the m first PCA eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$ theoretically lie in the signal subspace.[29] Because the data covariance matrix $\mathbf{R}_{xx}$ must in practice be estimated from a finite number of samples, this property does not hold exactly in practice.

3.2. *Estimation of the number of sources during pre-whitening*

In practical situations, the correct number m of the sources is often unknown. PCA is useful also in estimating m, and this can be conveniently carried out in context with PCA-based pre-whitening and/or noise filtering.

3.2.1. *Heuristic approach*

Recall first that in the ideal noiseless case only the m largest "signal" eigenvalues $\lambda_1, \ldots, \lambda_m$ of the data covariance matrix $\mathbf{R}_{xx}$ are nonzero while the remaining "noise" eigenvalues are zero. This yields directly an estimate to the number m of the sources. If the powers of the sources are much larger than the power of noise, the m largest signal eigenvalues are still clearly larger than noise eigenvalues, and it is straightforward to determine m from the breakpoint. However, if some of the sources are weak or the power of the noise is not small, it is generally hard to see what is the correct number m of sources just by inspecting the eigenvalues. The main practical difficulty in this approach is how to correctly set the threshold which divides the eigenvalues into the m signal eigenvalues and the remaining $n - m$ noise eigenvalues. If the threshold has been selected roughly correctly, this approach yields good results in estimating the number m of sources, otherwise not.

3.2.2. *AIC and MDL criteria*

Instead of setting the threshold between the signal and noise eigenvalues using some heuristic procedure or a rule of thumb, we can use two well-known information theoretic criteria, namely Akaike's information criterion (AIC) or the minimum description length (MDL) criterion. Wax and Kailath[32] have evaluated explicit expressions of the AIC and MDL criteria for estimating the number m of signals in the model (1). This subspace model is used widely also in sinusoidal frequency estimation and array processing with different assumptions,[29] and the same model order determination problem appears also there.

The formulas of Wax and Kailath have the form[29,32]

$$\begin{aligned} \mathrm{AIC}(m) &= -2K(n-m)\,\ln \varrho(m) \\ &\quad + 2m(2n-m)\,, \end{aligned} \tag{20}$$

$$\begin{aligned} \mathrm{MDL}(m) &= -K(n-m)\,\ln \varrho(m) \\ &\quad + 0.5m(2n-m)\,\ln K\,. \end{aligned} \tag{21}$$

Here K is the number of the data vectors $\mathbf{x}(t)$ used in estimating the data covariance matrix $\mathbf{R}_{xx}$, and

$$\varrho(m) = \frac{(\lambda_{m+1}\lambda_{m+2}\cdots\lambda_n)^{\frac{1}{n-m}}}{\frac{1}{n-m}(\lambda_{m+1}+\lambda_{m+2}+\cdots+\lambda_n)} \tag{22}$$

is the ratio of the geometric mean of the $n-m$ smallest PCA eigenvalues to their arithmetic mean. The estimate $\hat{m}$ of the number of the signals (in our case sources) is chosen so that it minimizes either the AIC or MDL criterion.

A problem with the AIC and MDL criteria given above is that they have been derived by assuming that the data vectors $\mathbf{x}(t)$ have a Gaussian distribution.[29,32] This is done for mathematical tractability, making it possible to derive closed form expressions. The Gaussianity assumption does not usually hold exactly in BSS and other signal processing applications. Therefore, the MDL and AIC criteria yield suboptimal estimates only, but provide anyway formulas that have turned out useful in model order estimation in lack of better criteria.

3.2.3. *Practical approach*

One might at first sight think that the MDL and AIC criteria cannot be applied to the BSS problem, because there we assume that the source signals $s_i(t)$ are non-Gaussian. However, it should be noted that the components of the data vectors $\mathbf{x}(t)$ are mixtures of the sources, and therefore often have distributions that are actually not so far from the Gaussian one. In our practical experiments, the MDL and AIC criteria have quite often performed very well in estimating the number m of the sources in the BSS problem. We have found two practical requirements for their successful use. First, the number n of mixtures must be larger than the number m of the sources. (If $n = m$, the ratio (22) cannot be computed.) The second requirement is that there must be at least a small amount of noise present. This guarantees that also the noise eigenvalues $\lambda_{m+1}, \ldots, \lambda_n$ are nonzero. It is obvious that zero eigenvalues cause difficulties in formulas (20) and (21).

3.3. *Separation of pre-whitened signals*

In the separation stage, following the pre-whitening stage, we have applied the *nonlinear PCA* rule and the *bi-gradient* rule. The *Nonlinear PCA subspace rule*[27,24] employs the following update rule for the orthogonal separating matrix $\tilde{\mathbf{W}}$:

$$\tilde{\mathbf{W}}(t+1) = \tilde{\mathbf{W}}(t) + \eta(t)\mathbf{g}[\mathbf{y}(t)][\mathbf{v}(t) - \tilde{\mathbf{W}}^T(t)\mathbf{g}[\mathbf{y}(t)]]^T , \tag{23}$$

where $\mathbf{v}(t) = \mathbf{V}(t)\mathbf{x}(t)$, $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$, and $\mathbf{y}(t) = \tilde{\mathbf{W}}(t)\mathbf{v}(t)$. The notation $\mathbf{g}[\mathbf{y}(t)]$ is used for the column vector whose ith component is $g_i[y_i(t)]$, where $g_i(t)$ is usually an odd and monotonically increasing nonlinear activation function. The learning rate $\eta(t)$ must be positive for stability reasons. The separation properties of (23) have been analyzed mathematically in simple cases in Ref. 27. In our experiments we have applied alternatively the following activation functions:

$$\begin{aligned} g[y(t)] &= y^3(t) && \text{for super-Gaussian sources} \\ g[y(t)] &= \tanh[1.3y(t)] && \text{for sub-Gaussian sources} \end{aligned} \tag{24}$$

Another algorithm, the *bi-gradient algorithm*[24,31] can also be applied after pre-whitening for learning the orthogonal separating matrix $\tilde{\mathbf{W}}$:

$$\tilde{\mathbf{W}}(t+1) = \tilde{\mathbf{W}}(t) + \gamma(t)\mathbf{g}[\mathbf{y}(t)]\mathbf{v}^T(t) + \eta(t)[\mathbf{I} - \tilde{\mathbf{W}}(t)\tilde{\mathbf{W}}^T(t)]\tilde{\mathbf{W}}(t) . \tag{25}$$

For sub-Gaussian sources, we can use either a small negative learning parameter $\gamma(t)$ together with the cubic nonlinearity $g(y) = y^3$, or a small positive $\gamma(t)$ with the sigmoidal nonlinearity $g(y) = \tanh(\alpha y)$. Here α is a possible scaling constant (with pre-whitening $\alpha = 1.3$ provides good convergence). For super-Gaussian sources the same choices with reversed sign of the learning parameter $\gamma(t)$ are applicable.

4. Separation with Cancellation of Convolutional Colored Noise

The problem of efficiently separating the sources if additive colored noise cannot any longer be neglected, can be stated alternatively: How to cancel or suppress additive colored noise. In general, the problem is rather difficult because we have $2 \times n$ unknown signals (where n is the number of sensors). Hence the problem is highly under-determined, and without any *a priori* information about the mixture model and/or noise it is very difficult or even impossible to solve it.[19]

4.1. *Reference noise*

However, in many practical situations we can measure or model the environmental noise. We shall denote in the following such noise as *reference noise* $n_R(t)$ or a vector of reference noises [for each separate sensor $n_i(t)$] (Fig. 3).[16] For example, in acoustic *cocktail party* problems we could measure such a noise during a short salience period (when no one speaks), or we could measure and record it online by an extra isolated microphone. In a similar way one can measure noise in biomedical applications like EEG or ECG by extra electrodes, placed appropriately.

The noise $n_R(t)$ may influence each sensor in some unknown manner due to environmental effects. Hence, such effects like delays, reverberation, echo, nonlinear distortion etc. may occur. It can be assumed that the reference noise is processed by some unknown nonlinear dynamic system before it reaches each sensor (Fig. 3). We could consider ARMA,

NARMA or FIR (convolutive) noise models. In the simplest case, a convolutive model of noise can be assumed, that is, the reference noise is processed by some FIR (finite impulse response) filters, whose parameters need to be estimated.[16] Hence, the additive noise in the ith sensor is modeled as [Fig. 3(b)][16]:

$$n_i(t) = \sum_{j=0}^{N} b_{ij} z^{-j} n_R(t) \tag{26}$$

where $z^{-1} = e^{-sT}$ is the unit delay. Equation (26) can be written in the time domain as

$$n_i(t) = b_{i0} n_R(t) + b_{i1} n_R(t-T) + \cdots + b_{iN} n_R(t-NT)\,. \tag{27}$$

In this model, we assume that known reference noise is added to each sensor (mixture of sources) with different unit delays T and various but unknown coefficients $b_{ij}(t)$. In other words, we assume that noise is convolutional and the reference noise n_R is known. The mixing matrix $\mathbf{A}$, the coefficient vector $\mathbf{b}(z)$, and the number of time delay units N (maximum order of the FIR filters) are completely unknown.

4.2. *Learning algorithm for noise cancellation*

In our simple model the noise cancellation and source separation stages are performed sequentially. We first attempt to cancel the noise contained in the mixtures and then to separate the sources (Fig. 4). Thus the output signals are derived from:

$$\begin{aligned}\mathbf{y}(t) &= \mathbf{W}[\mathbf{x}(t) - \mathbf{h}(z) n_R] \\ &= \mathbf{W}\mathbf{A}\mathbf{s}(t) + \mathbf{W}\mathbf{b}(z) n_R - \mathbf{W}\mathbf{h}(z) n_R\,. \end{aligned} \tag{28}$$

It is obvious that $\mathbf{y}(t) \simeq \mathbf{s}(t)$ if (problems of signal scaling and permutation are omitted):

$$\mathbf{W}\mathbf{A} = \mathbf{I} \quad \text{and} \quad \mathbf{h}(z) = \mathbf{b}(z)\,. \tag{29}$$

In order to cancel additive noise and to develop an adaptive learning algorithm for unknown coefficients $h_{ij}(t)$ we can apply the concept of minimization of generalized output energy of output signals $\tilde{\mathbf{x}}(t) = [\tilde{x}_1(t), \tilde{x}_2(t), \ldots, \tilde{x}_n(t)]^T$. In other words, we can formulate the following cost function (generalized energy)[16]:

$$J(\mathbf{h}) = \sum_{i=1}^{n} \rho_i(\tilde{x}_i) \tag{30}$$

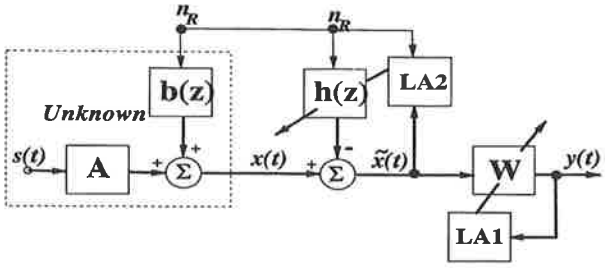


Fig. 4. The model for blind source separation with noise cancellation as pre-processing.

where $\rho_i(\tilde{x}_i)$ is a suitably chosen loss function, typically:

$$\rho_i(\tilde{x}_i) = \frac{1}{\beta} \ln \cosh(\beta \tilde{x}_i) \quad \text{or} \quad \rho_i(\tilde{x}_i) = \frac{1}{p} |\tilde{x}_i|^p \tag{31}$$

and

$$\tilde{x}_i(t) = x_i(t) - \sum_{j=1}^{n} h_{ij} n_R(t - jT)\,, \quad \forall i\,. \tag{32}$$

Minimization of this cost function using a standard stochastic gradient descent approach leads to the learning algorithm (compare Fig. 4)[16]:

$$\begin{aligned} h_{ij}(t+1) &= h_{ij}(t) - \tilde{\eta}(t) \frac{\partial J(\mathbf{h})}{\partial h_{ij}} \\ &= h_{ij}(t) + \tilde{\eta}(t) f_R[\tilde{x}_i(t)] n_R(t - jT)\,. \end{aligned} \tag{33}$$

Here $f_R(\tilde{x}_i(t))$ is a suitably chosen nonlinear function:

$$f_R(\tilde{x}_i(t)) = \frac{\partial \rho_i(\tilde{x}_i)}{\partial \tilde{x}_i}\,. \tag{34}$$

Typical choices are $f_R(\tilde{x}_i(t)) = \tilde{x}_i^3(t)$ or $f_R(\tilde{x}_i(t)) = \tanh(\alpha \tilde{x}_i(t))$.

4.3. *Separation layer*

In order to separate noiseless signals we can alternatively apply two relative simple and powerful learning algorithms without any preprocessing[2,8,10,14]:

- the global robust learning rule[2,8,9]:

$$\begin{aligned} \mathbf{W}(t+1) &= \mathbf{W}(t) + \eta(t) \\ &\quad \times \{\mathbf{I} - \mathbf{f}[\mathbf{y}(t)]\mathbf{g}[\mathbf{y}^T(t)]\}\mathbf{W}(t)\,, \end{aligned} \tag{35}$$

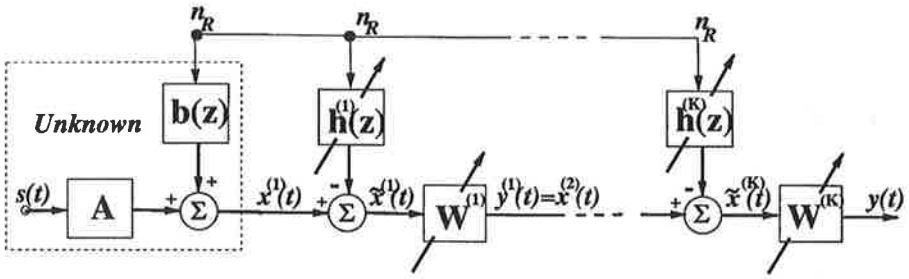


Fig. 5. Multi-layer neural network model for improved source separation with noise cancellation.

which can be written in scalar form as:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) \times \left[w_{ij}(t) - f_i[y_i(t)] \sum_{k=1}^{m} w_{kj}(t) g_k[y_k(t)] \right], \quad (36)$$

- or the local learning rule[11,14]:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) \times \{\delta_{ij} - f_i[y_i(t)] g_j[y_j(t)]\}. \quad (37)$$

In the above equations, δ_{ij} is the *Kronecker* delta, $\eta(t) > 0$ is the adaptive learning rate and $\mathbf{I}$ is the $n \times n$ identity matrix. As before, $\mathbf{f}(\mathbf{y}) = [f(y_1), \ldots, f(y_n)]^T$ and $\mathbf{g}(\mathbf{y}^T) = [g(y_1), \ldots, g(y_n)]$ denote vectors of nonlinear activation functions, where $f(y)$, $g(y)$ is a pair of nonlinear functions chosen to match the type of sources (for example images, sound signals, and speech signals).

For $g(y) = y$ the rule (35) was theoretically derived and justified by Amari *et al.*[1-3]

4.4. *Multi-layer models*

In order to improve the learning performance multi-layer neural networks can be used (Fig. 5).[11,16] Employing several layers clearly improves the separation results if we want to apply a local learning rule for separation of mixtures in which some of the source signals are very weak or the mixing matrix $\mathbf{A}$ is ill-conditioned. A multi-layer model might be a proper solution also if the initialization and decreasing speed of the learning rates $\eta(t)$ and $\tilde{\eta}$ have not been chosen optimally.

In case there are more sensors than sources ($n > m$), there may appear redundant signals on some outputs. This redundancy can be eliminated by adding a post-processing layer, which in course of learning suppresses eventual redundant signals. An appropriate solution for this problem was proposed in Refs. 11 and 12.

5. Computer Simulation Results

5.1. *Performance measures*

In order to estimate the quality of separation we assume that the original sources and the mixing matrix are known. However, these quantities are unknown to the learning algorithms. The separation results can be assessed qualitatively from figures showing the original sources, mixtures, and separated sources. The accuracy of the obtained results can be measured quantitatively using some suitable criterion.

For sound signals one such measure is $\mathrm{snr}[Y_i, S_j]$ — the *signal-to-noise* ratio between the error of source S_j reconstruction by signal Y_i and the corresponding original source S_j:

$$\mathrm{snr}[Y_i, S_j] = -10 \log_{10}(\mathrm{MSE}[Y_i, S_j]), \quad (38)$$

where MSE is the *mean square error* of source reconstruction:

$$\mathrm{MSE}[Y_i, S_j] = \frac{1}{N} \sum_{k=1}^{N} (y_{jk} - s_{jk})^2. \quad (39)$$

In order to made this measure independent of scaling factors before calculation of the signal-to-noise ratios the amplitudes of the Y_i and S_j signals are always normalized by $\max(|y_{jk}|) = \max(|s_{jk}|) = 1$.

For image sources it is common to use a different but related index — the *peak signal-to-noise-ratio* defined as:

$$\mathrm{psnr}[Y_i, S_j] = 10 \log_{10} \left(\frac{A^2}{\mathrm{MSE}[Y_i, S_j]} \right), \quad (40)$$

where $A = s_{\max} - s_{\min}$ is the amplitude interval of source signal. Also in this case the maximum signal amplitudes of the Y_i and S_j are always normalized to 1.

One of the above quality factors is computed alternatively for each source reconstruction by each output signal.

5.2. *PCA based separation in noise*

We first illustrate the effect of a Gaussian unknown additive noise on the separation results, and next demonstrate how PCA filtering can be used for removing this type of noise. In the following experiments, we applied the PCA pre-whitening approach, and then learned the orthogonal separating matrix $\tilde{\mathbf{W}}$ using the bigradient algorithm.[24,31] We express the relative noise level contained in the noisy signal mixtures by the following general *signal-to-noise* ratio SNR, defined as:

$$\mathrm{SNR} = -10 \log_{10} \left(\frac{\mathrm{E}\{\mathbf{n}^2(t)\}}{\mathrm{E}\{\mathbf{s}^2(t)\}} \right) . \qquad (41)$$

The first row in Fig. 6 shows four original images and the second one contains separation results for four noiseless mixtures. The last two rows show

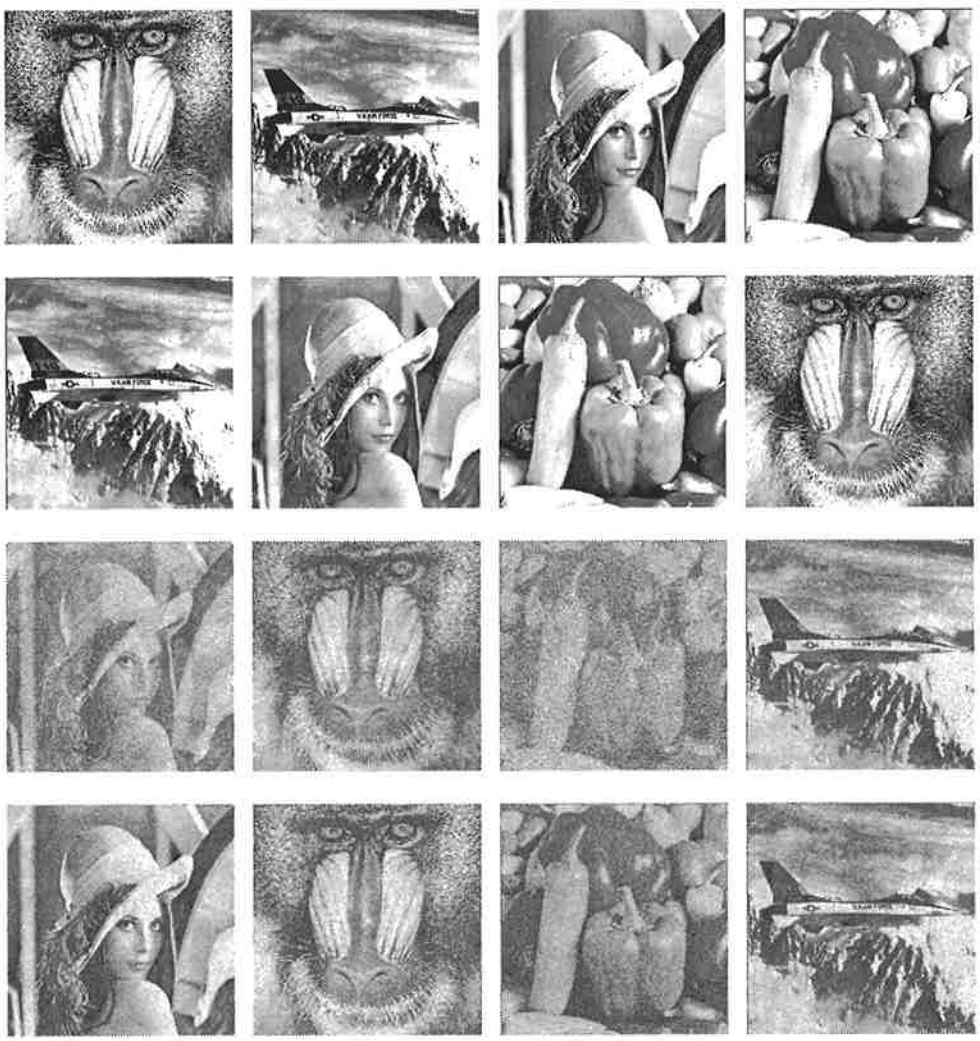


Fig. 6. Separation results in Gaussian additive white noise: (1st row) original 4 images (sub-Gaussian sources), (2nd row) results for 4 mixtures in noiseless case, (3rd row) results for 4 noisy mixtures, (4th row) results for 8 noisy mixtures and when PCA was used to whiten and compress the data vectors to 4-D vectors prior to blind separation.

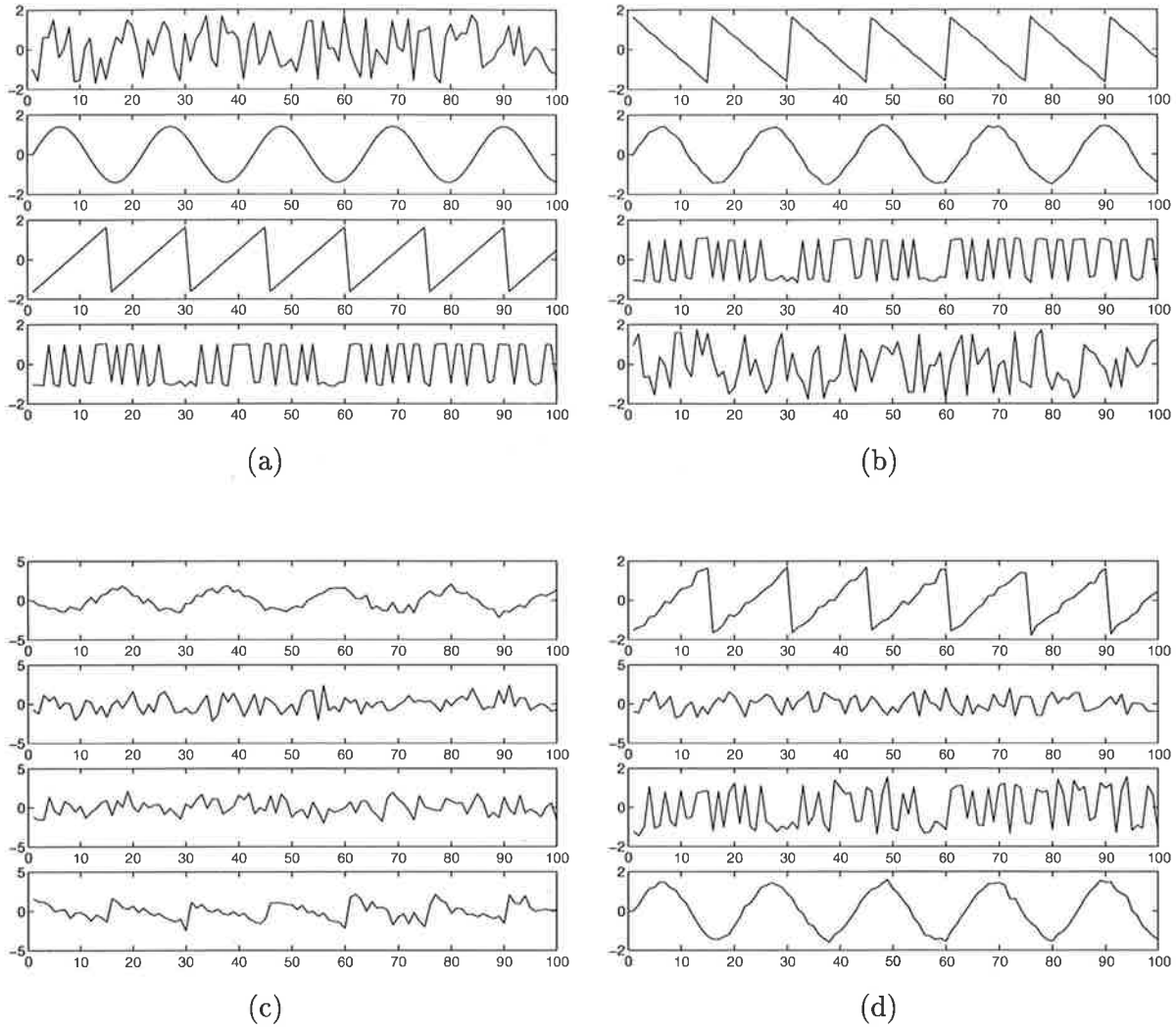


Fig. 7. Separation results in Gaussian additive white noise: (a) original 4 sub-Gaussian sources, (b) separation results in noiseless case, (c) separation results for 4 noisy mixtures, (d) results for 8 noisy mixtures when PCA was used to whiten and compress the data vectors to 4-D vectors prior to blind separation.

separation results for 4 and 8 noisy mixtures, respectively, when the noise level was SNR = 8 [dB]. In the last case, prior to separation (during whitening) the mixed signal dimension was compressed from 8 to 4. In this case, there is a significant improvement in separation quality of noisy mixtures, although the separation quality of the noiseless case can not be reached. The total number of data samples was 10 000.

Figure 7(a) shows 100 samples of the 4 sub-Gaussian original sources (uniformly distributed white noise, a sinusoid, a ramp signal, and a binary signal) used in this experiment. Figure 7(b) depicts the good separation results (outputs of the network) in the case where there were 4 noiseless mixtures, and no data compression took place in the whitening stage. The total number of data samples was 5000.

When some noise was added to the 4 mixtures (components of the data vectors) so that the signal-to-noise ratio of each mixture was about 26 [dB], the separation results shown in Fig. 7(c) are much poorer. Only some of the sources (sinusoid, ramp) are crudely recognizable. Figure 7(d) shows the

respective result when there were 8 similar noisy mixtures available, and the dimensionality of the data vectors was compressed from 8 to 4 using PCA whitening. The quality of the separated sources is now much better than in Fig. 7(c), and only slightly worse than in the noiseless case in Fig. 7(b).

Generally speaking, PCA yields good results in filtering Gaussian additive noise, if there are more mixtures than sources and enough samples from the mixtures. A general rule of thumb in estimation theory is that there should be at least 5–10 times more samples than parameters to be estimated for getting acceptable results. In PCA, one must in practice first estimate the $n \times n$ data covariance matrix $\mathbf{R}_{xx} = \mathrm{E}\{\mathbf{x}(t)\mathbf{x}(t)^T\}$ from the data vectors $\mathbf{x}(t)$. Due to the symmetry, $\mathbf{R}_{xx}$ has $n(n+1)/2$ free parameters to be estimated. Thus if for example $n = 8$ we should have at least several hundred samples for obtaining good noise filtering results in context with PCA whitening. Of course, the more noise, the more samples are needed for good accuracy.

5.3. *PCA based source number determination in noise*

Another requirement for obtaining good noise filtering results using PCA is that the correct number m of sources is known or can be estimated reliably. In the following, we present some experimental results on the performance of the MDL criterion (21) and the AIC criterion (20) in estimating the number of sources in a BSS problem. Generally speaking, these information-theoretic criteria are applicable provided that there is noise in the mixtures and the number n of mixtures is larger than the number m of sources. If there is no noise or $n \leq m$, the criteria do not yield good results.

We used the same 4 sub-Gaussian sources (having negative kurtoses) as in Fig. 7 in experiments with varying numbers of mixtures, sources and samples for different noise levels. Tables 1 and 2 show the results at different signal-to-noise ratios when the number of samples was 100 or 1000, respectively, and there were 5 noisy mixtures ($n = 5$) available. A total of 100 experiments were made in each case where the correct number of sources was either $m = 1$, 2, 3 or 4. The figures in the tables give the number of times when the MDL or AIC criteria provided a correct estimate for m. Thus the numbers in the tables are directly

Table 1. Percentages of correct estimates of the number m of sources given by the MDL and AIC criteria at different signal-to-noise ratios. The number of samples was 100.

SNR	No. of Sources			
	$m = 1$	$m = 2$	$m = 3$	$m = 4$
Estimates with MDL criterion (in [%])				
34.5 dB	99	99	97	96
24.5 dB	100	95	87	70
17.5 dB	99	50	37	20
Estimates with AIC criterion (in [%])				
34.5 dB	92	87	85	99
24.5 dB	84	88	85	85
17.5 dB	90	73	55	47

Table 2. Percentages of correct estimates of the number m of sources given by the MDL and AIC criteria at different signal-to-noise ratios. The number of samples was 1000.

SNR	No. of Sources			
	$m = 1$	$m = 2$	$m = 3$	$m = 4$
Estimates with MDL criterion (in [%])				
34.5 dB	100	100	100	98
24.5 dB	100	100	98	86
17.5 dB	100	94	68	55
Estimates with AIC criterion (in [%])				
34.5 dB	86	89	81	100
24.5 dB	89	84	85	95
17.5 dB	87	86	80	76

percentages of correctly estimated number of sources in each case.

From Tables 1 and 2 one can see that the MDL criterion performs very well, and is more reliable than the AIC criterion for high signal-to-noise ratios. Furthermore, the results given by the MDL criterion improve when the number of samples increases, while this effect is not so clear in the case of AIC. The reason for this behavior is obvious: The MDL criterion is consistent, yielding asymptotically correct results, while the AIC criterion does not have this theoretically desirable property.[32] However, at lower signal-to-noise ratios the AIC criterion is sometimes better. We also observed that when the MDL criterion failed, it gave almost always an estimate of m that was one too small ($\hat{m} = m - 1$). For example at SNR = 17.5 [dB] with 1000 samples the MDL criterion estimated the number $m = 4$ of sources correctly in 55% of cases, and in 45% of cases it

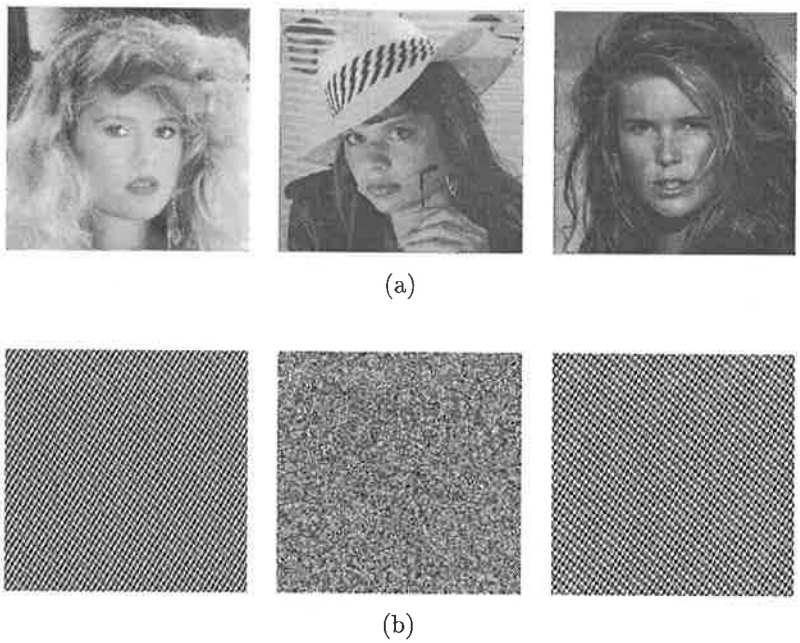


Fig. 8. Three original images (a) and three alternative colored reference noise signals (one noise signal per test) (b) used for generation of additive convolutional noise signals.

gave a wrong estimate $\hat{m} = 3$. The AIC criterion either overestimated or underestimated the number of sources (usually by one) when it failed.

Our simulations show that both criteria, especially MDL, can be successfully applied to the estimation of the number m of sources when the amount of unknown additive noise is small or moderate, even though the sources are not Gaussian as assumed in the theoretical derivations. The criteria may fail in estimating m if large noise is present. But then the subsequent blind separation task also becomes too difficult for the currently existing approaches.

5.4. *Separation with convolutional noise cancellation*

In an illustrative example three (unknown) natural images [Fig. 8(a)] were mixed by a randomly chosen matrix $\mathbf{A}$ (with the condition number $\text{cond}(\mathbf{A}) = 18.09$):

$$\mathbf{A} = \begin{bmatrix} 1.0 & 0.7 & 0.3 \\ 1.5 & 0.7 & 0.9 \\ 1.2 & 0.8 & 0.8 \end{bmatrix} \tag{42}$$

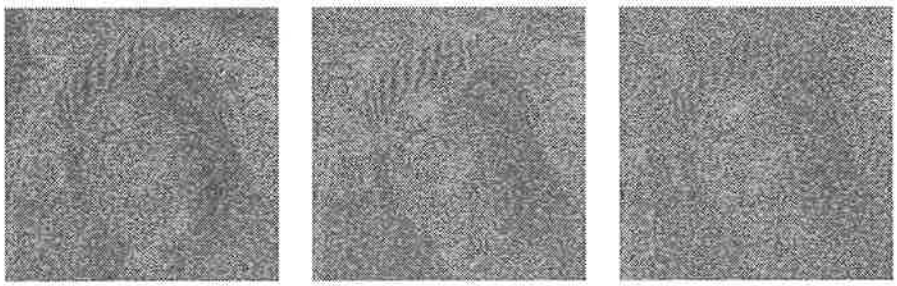
and convolutional noise signals were added, originated from one of three alternative reference noises [Fig. 8(b)]. All the sources were zero-mean signals, and sub-Gaussian with a negative kurtosis value. Nonlinear activation functions f, f_R were chosen to be: $f(y) = y^3$, $f_R(\tilde{x}) = \tilde{x}^3$.

The sources 1 and 2 as well as 2 and 3 were correlated with each other (by 14–22%). However, none of the sources was in practice correlated with any of the three noise signals, the source-to-noise correlations being below 1%.

Convolutional noise signals were modeled using two tenth order unknown FIR filters ($N = 10$), the first one is introducing small noise and the second one large noise (see the top row on Fig. 9). The FIR matrix for generation of large convolutional noises was:

$$\mathbf{B} = \begin{bmatrix} 1.2 & 1.4 & 1.1 & 1.0 & 0.9 & 0.8 & 0.7 & 0.55 & 0.40 & 0.3 \\ 1.3 & 1.2 & 1.5 & 1.1 & 0.95 & 0.84 & 0.77 & 0.65 & 0.54 & 0.4 \\ 1.1 & 1.2 & 1.3 & 1.15 & 0.99 & 0.74 & 0.87 & 0.85 & 0.94 & 1.0 \end{bmatrix} \tag{43}$$

Three mixed images (with small additive and convolutional noise 1)



Three mixed images (with large additive and convolutional noise 2)

(a)

(b) Mixed images after noise cancellation

(c) Output images after subsequent blind separation

Fig. 9. Example of source separation with colored noise cancellation for an image mixture containing additive, convolutional noise.

In the noise cancellation model the number M of delay units was chosen to be equal to 25. The noise cancellation results are very accurate. After 2000 iterations, the FIR $h_i(z)$ filters are already as follows:

$$\begin{aligned}
\mathbf{h}_1 &= [1.202, 1.401, 1.096, 0.999, 0.897, 0.795, 0.699, 0.547, 0.400, 0.299,\\
&\quad -0.011, -0.005, -0.003, -0.003, 0.002, -0.001, -0.001, 0.000,\\
&\quad -0.012, -0.007, -0.008, -0.007, -0.002, -0.001, -0.002]\\
\mathbf{h}_2 &= [1.301, 1.203, 1.499, 1.101, 0.949, 0.835, 0.770, 0.648, 0.541, 0.401,\\
&\quad -0.011, -0.004, -0.002, -0.003, 0.002, -0.001, -0.001, 0.002,\\
&\quad -0.012, -0.005, -0.007, -0.007, -0.002, -0.002, -0.003]\\
\mathbf{h}_3 &= [1.102, 1.203, 1.299, 1.151, 0.989, 0.735, 0.871, 0.849, 0.941, 1.001,\\
&\quad -0.010, -0.003, -0.002, -0.003, 0.002, -0.000, -0.001, 0.001,\\
&\quad -0.011, -0.005, -0.006, -0.006, -0.001, -0.002, -0.003]
\end{aligned} \tag{44}$$

Table 3. Quality factors for separation and noise cancellation of image signals.

Noise	Signals	psnr [dB]		
		Face 1	Face 2	Face 3
Separation with noise cancellation				
Small	**y**	20.32	24.50	31.54
Large	**y**	20.30	27.53	32.29
Separation of a noise-free mixture				
None	**y**	23.37	26.19	28.36

From Eqs. (43) and (44) one can clearly see that the approximation error of matrix **B** [Eq. (43)] is 0.1%–1.0%.

The separation performance is summarized in Table 3. It should be emphasized that the detailed performance values may change from experiment to experiment depending on the learning rates $\eta(t)$, $\tilde{\eta}(t)$ and on proper setting of their initial values and decay speed.

We repeated the same experiment for sound sources [Fig. 10(a)] and convolutional noise signals, generated from some reference noise [Fig. 10(b)]. All the sources were super-Gaussian with a positive kurtosis value. In this case the appropriate nonlinear activation functions are: $f(y) = \tanh(1.3y)$, $f_R(\hat{x}) = \tanh(10\hat{x})$.

Now the source and the noise were in practice uncorrelated with each other, i.e. all correlation factors were below 1%, so we expected that the separation and cancellation stages should be substantially easier than the same operations were for the image sources. The experiments verified this conjecture,

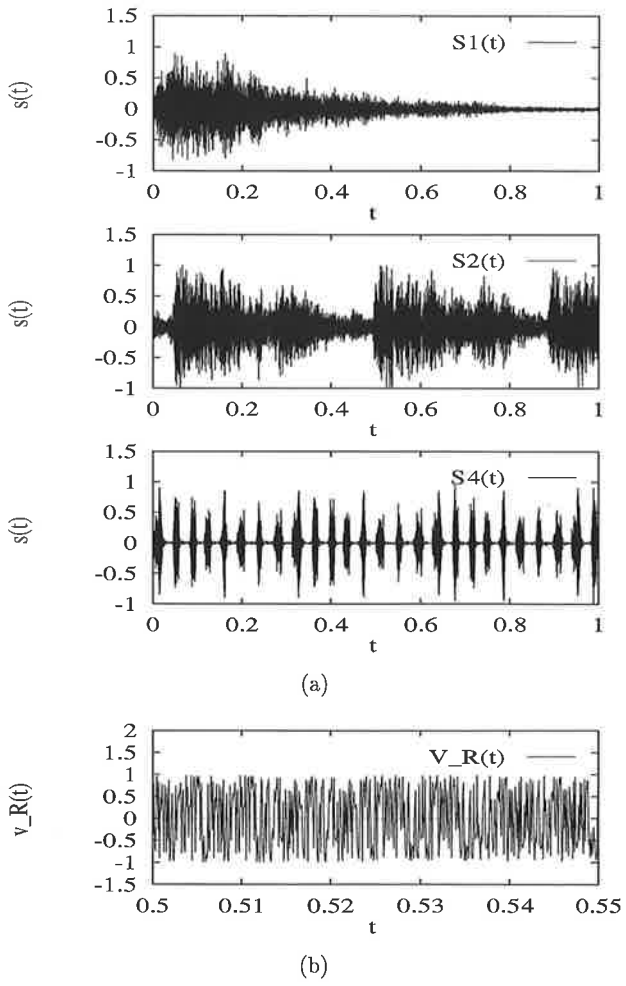
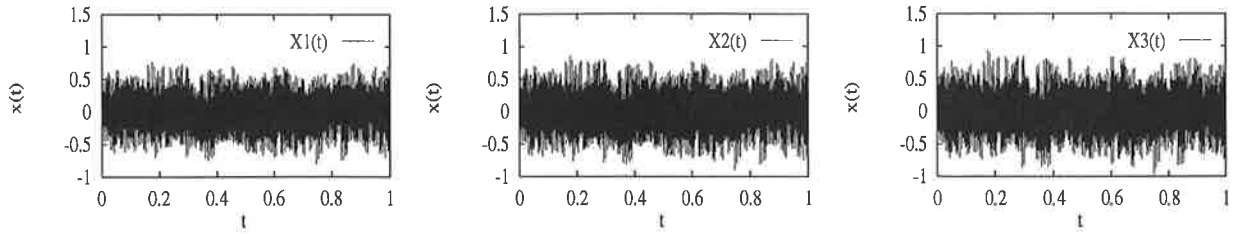
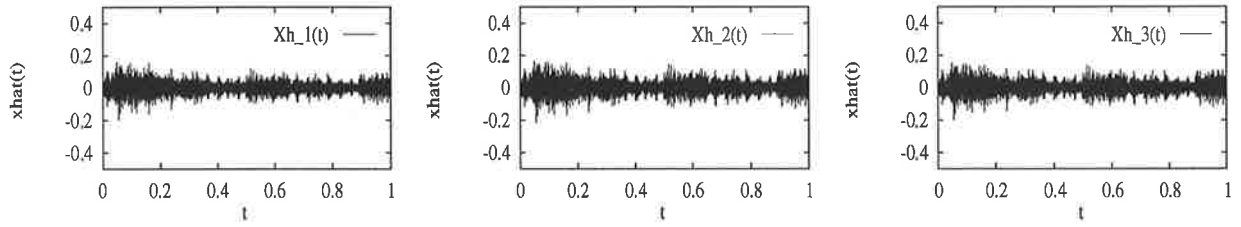


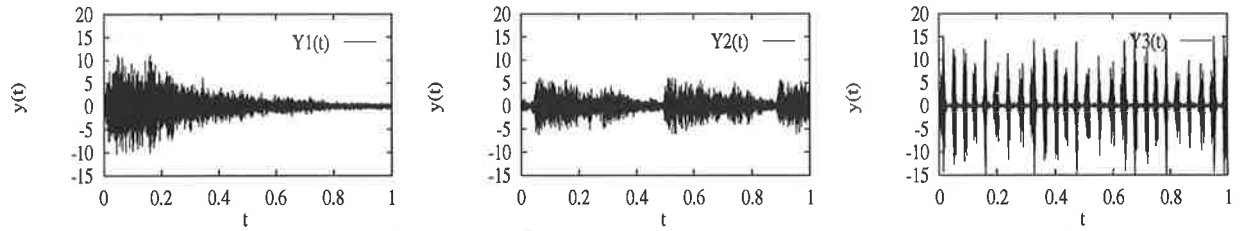
Fig. 10. Three sound sources (a) and one reference noise (b) used for generation of additive convolutional noise signals.



(a) Three mixed sounds of three sources with large additive, convolutional noise



(b) After noise cancellation the additive source mixtures are recovered



$snr[Y1, S1] = 19.9[dB]$ $snr[Y2, S2] = 27.1[dB]$ $snr[Y3, S3] = 25.2[dB]$

(c) Separated signals (reconstructed sources)

Fig. 11. Example of blind source separation and noise cancellation of a sound mixture with additive, convolutional noise, assuming that reference noise is available.

because high quality separation results were achieved in this case (see the example on Fig. 11).

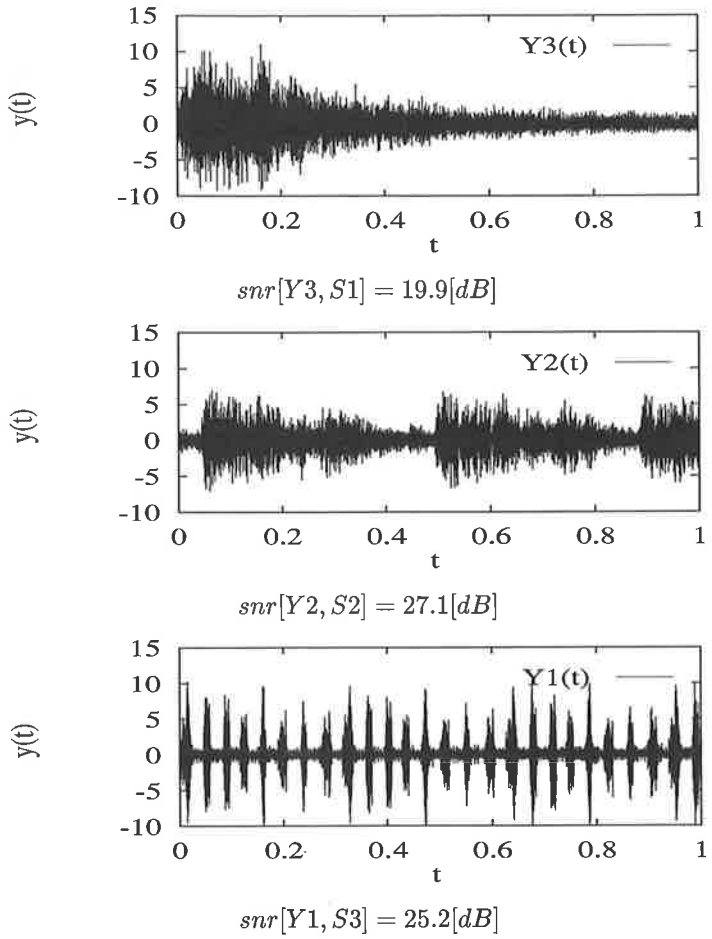
5.5. *Noise cancellation for more sensors than sources*

We also investigated such a case where only some convolutional noise signal generated by an unknown primary reference noise is measurable. In this case the cancellation of the noise is no longer perfect although it still gives good quality results. Usually the noise "returns back" (is amplified) during the subsequent blind separation, and instead of a desired source signal a noise signal appears on one or more outputs.

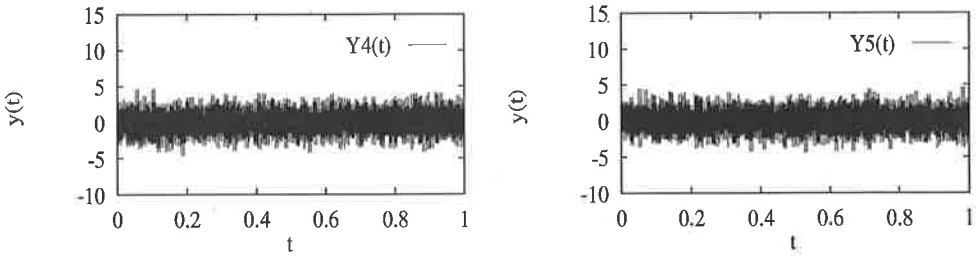
We can improve the performance if we apply more sensors than sources. Figure 12 illustrates an example of five sensors and only three sources. The applied neural network with learning rules described in Sec. 4 was able not only to cancel the additive, convolutive noise but also to extract all the sources successfully. In this example, additive noise signals appeared on two auxiliary outputs.

6. Conclusions

In this paper, we have developed adaptive approaches for blind separation of unknown source signals from their linear mixtures distorted by additive noise. The first approach is based on PCA, and is best applicable to completely unknown noise which is Gaussian or close to it. It was shown that if more mixtures than sources are available, considerably better separation results can be achieved in



(a) Three separated signals (reconstructed sources)



(b) Noise signals appear on two outputs

Fig. 12. Sound separation with noise cancellation results in an over-determined sensor case: five sensors, three sound sources.

noisy environment by first compressing the input vectors (mixed signals) onto PCA subspace. This can be done conveniently in context with PCA whitening. However, for obtaining good results the dimensionality of selected PCA subspace must be equal to the correct number of sources. In our experiments the separation results generally improved in noisy conditions when the number n of mixtures was larger

than the number m of sources, and PCA whitening was used to reduce the dimensionality of the input vectors from n to m. However, when the number of mixtures n was chosen too large, the separation results became worse. This effect was caused by the estimation errors in the covariance matrix due to the limited number of available samples. Generally, the number of samples should be at least 5–10 times the number $n(n+1)/2$ of free parameters in the data covariance matrix for getting good separation and noise filtering results using the PCA pre-whitening approach.

It was also shown that the number of the sources can be estimated with high reliability using the MDL criterion on certain conditions. In our experiments it turned out that when PCA-based pre-whitening had failed in estimating the number of sources correctly, the separation results were too noisy to be of much practical value. If the mixtures contain a large amount of general unknown noise, the whole blind separation task becomes in general very difficult or impossible.

The second approach simultaneously performs noise cancellation and source separation, assuming that reference noise is available. This approach is valid under the assumption that the unknown noise can be modeled as a convolutional noise mixture of a known reference noise. The proposed noise model could be extended to IIR adaptive filters, gamma filters, and other more sophisticated nonlinear neural network models of noise, like NARMAX.

Open problems are how to suppress or to cancel non-additive noise, and how to proceed if no reference noise is available or if there is no *a priori* knowledge about the statistics of noise.

The computer experiments presented demonstrate the validity and performance of proposed algorithms. The approaches were tested on various source signals including image and sound signals, but they are generally applicable to various classes of non-Gaussian signals, such as speech and biomedical signals.

References

1. S. Amari 1997, "Natural gradient works efficiently in learning," *Neural Computation* (in print).
2. S. Amari, A. Cichocki and H. Yang 1996, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems*, **8** (MIT Press, Cambridge, MA), pp.757–763.
3. S. Amari and A. Cichocki 1998, "Adaptive blind signal processing — neural network approaches," Invited paper in Proceedings of the IEEE, Special Issue: Blind System Identification and Estimation, eds. Ruey-wen Liu and Lang Tong (in print).
4. A. J. Bell and T. J. Sejnowski 1995, "An information maximization approach to blind operation and blind deconvolution," *Neur. Comput.* **7**, 1129–1159.
5. A. Belouchrani and J.-F. Cardoso 1995, "Maximum likelihood source separation by the expectation-maximization technique: Deterministic and stochastic implementation," in *Proceedings of Int. Symposium on Nonlinear Theory and its Applications (NOLTA-95)*, Las Vegas, USA, December, 49–53.
6. J. F. Cardoso and B. Laheld 1996, "Equivariant adaptive source separation," *IEEE Trans. on Sig. Processing* **43**, 3017–3029.
7. A. Cichocki and R. Unbehauen 1993, "Robust estimation of principal components in real time," *Electronics Letters* **29**, No. 21, 1869–1870.
8. A. Cichocki, R. Unbehauen and E. Rummert 1994, "Robust learning algorithm for blind separation of signals," *Electronics Letters* **30**(17), 1386–1387.
9. A. Cichocki and R. Unbehauen 1994, *Neural Networks for Optimization and Signal Processing*, 2nd edition (John Wiley, New York), pp. 461–471.
10. A. Cichocki and R. Unbehauen 1996, "Robust neural networks with on-line learning for blind identification and blind separation of sources," *IEEE Trans. Circ. Syst. I* **43**, 894–906.
11. A. Cichocki and W. Kasprzak 1996, "Local adaptive learning algorithms for blind separation of natural images," *Neural Network World* **6**, (IDG Co., Prague), 515–523.
12. A. Cichocki, W. Kasprzak and S. Amari 1996, "Neural network approach to blind separation and enhancement of images," *Signal Processing VIII. Theories and Applications* (EURASIP/LINT Publ., Trieste, Italy), Vol. I, pp. 579–582.
13. A. Cichocki, S. Amari, M, Adachi, W. Kasprzak 1996, "Self-adaptive neural networks for blind separation of sources," *Proc. of 1996 IEEE International Symposium on Circuit and Systems*, Vol. 2, pp. 157–160.
14. A. Cichocki, R. E. Bogner, L. Moszczynski and K. Pope 1997, "Modified Herault-Jutten algorithms for blind separation of sources," *Digital Signal Processing* **7**(2), 80–93.
15. A. Cichocki, W. Kasprzak and W. Skarbek 1996, "Adaptive learning algorithm for principal component analysis with partial data," in *Cybernetics and Systems '96* (Austrian Society for Cybernetic Studies, Vienna), pp. 1014–1019 (on-line at `http://www.bip.riken.go.jp/absl/publPCA.html`).
16. A. Cichocki, W. Kasprzak and S.-I. Amari 1996, "Adaptive approach to blind source separation with cancellation of additive and convolutional noise," in *Third Int. Conf. Signal Processing, ICSP'96* (IEEE Press/PHEI Beijing, China), pp. 412–415.
17. P. Comon, C. Jutten and J. Herault 1991, "Blind

separation of sources, Part II: Problems statement," *Sig. Processing* **24**, 11–20.

18. N. Delfosse and P. Loubaton 1995, "Adaptive blind separation of independent sources: A deflation approach," *Sig. Processing* **45**, 59–83.
19. S. Van Gerven 1996. "Adaptive noise cancellation and signal separation with applications to speech enhancement," Ph.D. Dissertation, Katholieke Universiteit Leuven, Dept. Elektrotechnik, Belgium.
20. C. Jutten and J. Herault 1991, "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture," *Sig. Processing* **24**, 1–20.
21. J. Karhunen and J. Joutsensalo 1994, "Representation and separation of signals using nonlinear PCA type learning," *Neural Networks* **7**, 113–127.
22. J. Karhunen and J. Joutsensalo 1995, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks* **8**, 549–562.
23. J. Karhunen, E. Oja, L. Wang, R. Vigario and J. Joutsensalo 1997, "A class of neural networks for independent component analysis," *IEEE Trans. Neural Networks* **12**, (accepted).
24. J. Karhunen 1996, "Neural approaches to independent component analysis and source separation," in *Proc. 4th European Symp. on Artificial Neural Networks (ESANN'96)* (Bruges, Belgium), pp. 249–266.
25. J. L. Lacoume and P. Ruiz 1992, "Separation of independent sources from correlated inputs," *IEEE Trans. Sig. Processing* **40**, 3074–3078.
26. E. Moreau and O. Macchi 1996, "High-order contrasts for self-adaptive source separation," *Int. J. Adaptive Control and Signal Processing* **10**, 19–46.
27. E. Oja and J. Karhunen 1995, "Signal separation by nonlinear Hebbian learning," in *Computational Intelligence — A Dynamic System Perspective*, eds. M. Palaniswami, Y. Attikiouzel, R. Marks, D. Fogel and T. Fukuda (IEEE Press, New York), pp. 83–97.
28. P. Pajunen 1997, "A competitive learning algorithm for separating binary sources," in *Proc. European Symposium on Artificial Neural Networks (ESANN'97)* (Bruges, Belgium), 255–260.
29. C. Therrien 1992, *Discrete Random Signals and Statistical Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ).
30. A.-J. van der Veen 1997, "Analytical method for blind binary signal separation," *IEEE Transaction on Signal Processing* **45**, 1078–1082.
31. L. Wang, J. Karhunen and E. Oja 1995, "A bigradient optimization approach for robust PCA, MCA, and source separation," in *Proc. 1995 IEEE Int. Conf. Neural Networks* (Perth, Australia), pp. 1684–1689.
32. M. Wax and T. Kailath 1985, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoustics, Speech, and Signal Processing* **33**, 387–392.
33. B. Widrow and S. D. Stearns 1985, *Adaptive Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ).