Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 71 (2008) 2344–2355
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Blind source extraction: Standard approaches and extensions to noisy and post-nonlinear mixing

Wai Yie Leong[a,*], Wei Liu[b], Danilo P. Mandic[a]

[a]Communications and Signal Processing Group, Department of Electrical and Electronic Engineering, Imperial College London SW7 2AZ, UK
[b]Communications Research Group, Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield S1 3JD, UK

Available online 29 February 2008

Abstract

We provide an overview of blind source extraction (BSE) algorithms whereby only one source of interest is separated at the time. First, BSE approaches for linear instantaneous mixtures are reviewed with a particular focus on the "linear predictor" based approach. A rigorous proof of the existence BSE paradigm is provided, and the mean-square prediction error (MSPE) is identified as a unique source feature. Both the approaches based on second-order statistics (SOS) and higher-order statistics (HOS) are included, together with extensions for BSE in the presence of noise. To help circumvent some of the problems associated with the assumption of linear mixing, an extension in the form of post-nonlinear mixing system is further addressed. Simulation results are provided which confirm the validity of the theoretical results and demonstrate the performance of the derived algorithms in noiseless, noisy and nonlinear mixing environments.

Keywords: Blind source extraction; Linear predictor; Nonlinear predictor; Noisy mixtures; Post-nonlinear model

1. Introduction

Due to its wide range of potential applications including those in biomedical engineering, sonar, radar, speech enhancement and telecommunications [34], blind source separation (BSS) has been studied extensively and has become one of the most important research topics in the signal processing area [1,4,8,12,15–17]. This is a technique which aims at recovering the original sources from all kinds of their mixtures, without the need for prior knowledge of the mixing process and the sources themselves. Fig. 1 shows a block diagram of the BSS process, whereby there are n sources $s_1(k), s_2(k), \ldots, s_n(k)$ passed through an unknown mixing system with additive noise $\mathbf{vn}(k)$; by m sensors we acquire the received mixed signals $x_1(k), x_2(k), \ldots, x_m(k)$. By BSS, the original signals are recovered from their mixtures, subject to the ambiguities of permutation and scaling. Depending on the nature of mixing, the sources are mixed in a linear or nonlinear manner. The linear case can be further divided into two categories: *instantaneous* mixing and *convolutive* mixing. In the convolutive mixing, the mixtures are modelled as sums of filtered versions of the sources. Compared to instantaneous mixing, this is a more complex problem and is beyond the scope of this paper.

Within the instantaneous mixing model, the mixtures $\mathbf{x}(k)$ are modelled as weighted sums of individual sources without dispersion or time delay, given by

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{vn}(k), \tag{1}$$

with

$$\begin{aligned}
\mathbf{s}(k) &= [s_1(k), s_2(k), \ldots, s_n(k)]^{\mathrm{T}},\\
\mathbf{x}(k) &= [x_1(k), x_2(k), \ldots, x_m(k)]^{\mathrm{T}},\\
[\mathbf{A}]_{i,j} &= a_{i,j}, \quad i = 1, \ldots, m, \;\; j = 1, \ldots, n,\\
\mathbf{vn}(k) &= [vn_1(k), vn_2(k), \ldots, vn_n(k)]^{\mathrm{T}},\\
\mathbf{y}(k) &= [y_1(k), y_2(k), \ldots, y_n(k)]^{\mathrm{T}},
\end{aligned} \tag{2}$$

where $\mathbf{A}$ is the mixing matrix and $\mathbf{y}$ are separated outputs. We normally assume that the sources are zero-mean and the elements of $\mathbf{vn}(k)$ are white Gaussian and independent of the source signals.

*Corresponding author at: SIMTech, A*STAR, 71 Nanyang Drive, Singapore 638075, Singapore. Tel./fax: +65 90841108.
E-mail addresses: waiyie@ieee.org (W.Y. Leong), w.liu@sheffield.ac.uk (W. Liu), d.mandic@imperial.ac.uk (D.P. Mandic).

0925-2312/$ - see front matter
doi:10.1016/j.neucom.2007.10.023

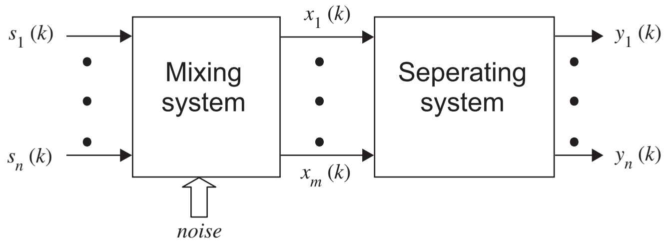


Fig. 1. Block diagram of the blind source separation problem.

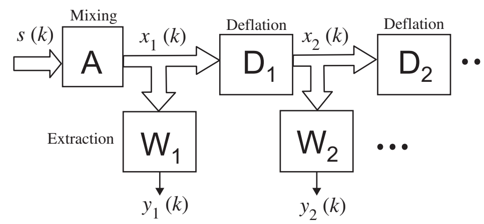


Fig. 2. Blind source extraction (BSE) and deflation.

In general, by BSS all the n sources are obtained simultaneously; this introduces large computational burden for a large number of mixtures. In many practical situations; however, we may choose to extract only a single source or a subset of sources at a time [11,7,25,6,9,2]. This process can be repeated until the last source or the last desired source/subset of sources is extracted [5,20,26,10,21]. The BSS approach operating in this way is also called blind source extraction (BSE) [4,22–24,28]. Compared to the general simultaneous BSS for multiple sources, BSE provides us with more freedom in separation. For instance, different algorithms can be used at different stages of the extraction process, which is achieved according to the features of the source signal we want to extract at a particular stage. In addition, by extracting only a limited set of signals of interest (instead of separating all of the sources), much of the unnecessary computation is avoided. This is particularly beneficial when the spatial dimension of observed mixtures is large and the number of signals of interest is small.

The aim of this paper is two-fold: (i) to provide an overview and theoretical justification for a class of BSE algorithms, (ii) to suggest feasible practical extensions of standard BSE. The focus is on the approaches for noisy instantaneous linear/post-nonlinear (PNL) mixtures. The paper is organised as follows. In Section 2, a brief review of the BSE for instantaneous linear noise-free mixtures is given, including algorithms based on both the higher- and second-order statistics (SOS). By analysis of the kurtosis based method for noisy mixtures, a BSE method for instantaneously mixed noisy mixtures is provided. In Section 3, we extend the BSE for linearly mixed sources to the case of PNL mixtures. Simulation results are given in Section 4 to illustrate the validity of the proposed methodology. Finally, conclusions are drawn in Section 5.

2. BSE for noisy linear mixtures

Fig. 2 shows a general BSE architecture for extracting one single source at a time; it consists of two principal stages: extraction and deflation [11]. The mixtures first undergo the extraction stage to have one of the sources recovered (as indicated by $\mathbf{w}_1, \ldots, \mathbf{w}_n$); after deflation, the so extracted source is removed from the mixtures $(\mathbf{D}_1, \ldots, \mathbf{D}_n)$. The new "deflated" mixtures of the remaining sources then undergo the next extraction stage to recover the second source; this process is repeated until the last source of interest is recovered. In this section, we overview a class of standard BSE algorithms, and provide a rigorous proof of the existence of the BSE solution. After establishing the conditions for uniqueness, an extension to the case with noisy mixtures is provided.

2.1. Review of BSE for noise-free instantaneous mixtures

2.1.1. Kurtosis based BSE algorithms

When source signals are independent, ICA methods can be conveniently used to perform the extraction. As the mixtures of independent Gaussian sources mixed by an orthogonal mixing matrix remain independent, the ICA approaches are only applicable to non-Gaussian sources, or when at most only one source is Gaussian. Under this condition, the extraction of one original source signal is equivalent to extracting an independent component from the mixtures. From (1), for noise-free cases, we therefore have

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k). \tag{3}$$

In the extraction stage, a linear combination of the mixed signals is considered, given by

$$y(k) = \mathbf{w}^{\mathrm{T}}\mathbf{x}(k) = \mathbf{g}^{\mathrm{T}}\mathbf{s}(k), \tag{4}$$

where $\mathbf{g}^{\mathrm{T}} = \mathbf{w}^{\mathrm{T}}\mathbf{A}$ is the global demixing vector.

To extract one of the source signals, it is sufficient to maximise the non-Gaussianity of $y(k)$. A classic measure of non-Gaussianity is the kurtosis, which for a zero-mean random variable y is defined as [17]

$$kt(y) = E\{y^4\} - 3(E\{y^2\})^2, \tag{5}$$

where $E\{\cdot\}$ denotes the statistical expectation operator.

The kurtosis of a Gaussian random variable is zero, whereas for most non-Gaussian random variables it is non-zero. Random variables with positive kurtosis are called super-Gaussian, whereas they are termed sub-Gaussian for a negative kurtosis. One such kurtosis based algorithm, is based on the minimisation of the cost function given by [7]

$$C(\mathbf{w}) = -\frac{\beta kt(y)}{4(E\{y^2\})^2}, \tag{6}$$

where $\beta = 1$ for the extraction of source signals with positive kurtosis and $\beta = -1$ for sources with negative kurtosis. Applying standard gradient descent yields

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\phi(y(k))\mathbf{x}(k), \tag{7}$$

where μ is the stepsize and

$$\phi(y(k)) = \beta\left[\frac{m_2(y)}{m_4(y)}y^3(k) - y(k)\right] \tag{8}$$

with

$$m_q(y)(k) = (1-\lambda)m_q(y)(k-1) + \lambda|y(k)|^q, \quad q = 2,4, \tag{9}$$

where λ is the forgetting factor.

In addition to this normalised kurtosis based algorithm, other kurtosis-related algorithms include the KuicNet algorithm [13] and FastICA algorithm [18]. For more details, refer to [4].

2.1.2. BSE algorithms using a linear predictor

Kurtosis based algorithms are only applicable to independent non-Gaussian sources (or at most one Gaussian). However, if the sources are not correlated with one another and every source has a different temporal structure, it is still possible to perform successive separation. More specifically, assume

$$\begin{aligned}\mathbf{R}_{ss}(0) &= E\{\mathbf{s}(k)\mathbf{s}^{\mathrm{T}}(k)\} \\ &= \mathrm{diag}\{\rho_0(0), \rho_1(0), \ldots, \rho_{n-1}(0)\},\end{aligned} \tag{10}$$

with $\rho_i(0) = E\{s_i(k)\cdot s_i(k)\}$, $i = 0, 1, \ldots, n-1$, and

$$\begin{aligned}\mathbf{R}_{ss}(\Delta k) &= E\{\mathbf{s}(k)\mathbf{s}^{\mathrm{T}}(k-\Delta k)\} \\ &= \mathrm{diag}\{\rho_0(\Delta k), \rho_1(\Delta k), \ldots, \rho_{n-1}(\Delta k)\}\end{aligned} \tag{11}$$

with $\rho_i(\Delta k) \neq 0$ for some non-zero delay Δk.

One convenient way to exploit this property is to employ a linear predictor within the BSE structure, as shown in Fig. 3 [6,2,5,26,24], where the weighted sum $y(k) = \mathbf{w}^{\mathrm{T}}\mathbf{x}(k)$ is passed through a linear predictor with coefficient $\mathbf{b}$ and of length P. The "output" error in this case is given by

$$e(k) = y(k) - \mathbf{b}^{\mathrm{T}}\mathbf{y}(k), \tag{12}$$

where

$$\begin{aligned}\mathbf{b} &= [b_1, b_2, \ldots, b_P]^{\mathrm{T}}, \\ \mathbf{y}(k) &= [y(k-1), y(k-2), \ldots, y(k-P)]^{\mathrm{T}}.\end{aligned} \tag{13}$$

In general, different source signals have different prediction errors for a given linear predictor. A direct use of the prediction error as a cost function is not convenient, since the error values depend on the magnitudes of signals; instead consider the normalised mean squared prediction error (MSPE) of the extracted signal $y(k)$, given by

$$J(\mathbf{w}) = \frac{E\{e^2(k)\}}{E\{y^2(k)\}}. \tag{14}$$

The MSPE $E\{e^2(k)\}$ can be expressed as

$$E\{e^2(k)\} = \mathbf{w}^{\mathrm{T}}\mathbf{A}\hat{\mathbf{R}}_{ss}\mathbf{A}^{\mathrm{T}}\mathbf{w}, \tag{15}$$

where $\hat{\mathbf{R}}_{ss}$ is a diagonal matrix given by

$$\hat{\mathbf{R}}_{ss} = \mathbf{R}_{ss}(0) - 2\sum_{p=1}^{P} b_p\mathbf{R}_{ss}(p) + \sum_{p,q=1}^{P} b_p b_q \mathbf{R}_{ss}(q-p). \tag{16}$$

The diagonal elements of the matrix $\hat{\mathbf{R}}_{ss}$ represent the MSPEs related to the corresponding source signals; under the assumption $\mathbf{R}_{ss}(0) = \mathbf{I}$, those become the normalised MSPEs of the source signals (for the particular linear predictor with coefficients $\mathbf{b}$).

Substituting (15) and $E\{y^2(k)\} = \mathbf{w}^{\mathrm{T}}\mathbf{R}_{xx}(0)\mathbf{w}$ into (14), we have

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{A}\hat{\mathbf{R}}_{ss}\mathbf{A}^{\mathrm{T}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{R}_{ss}(0)\mathbf{A}^{\mathrm{T}}\mathbf{w}}. \tag{17}$$

By minimising the cost function $J(\mathbf{w})$ with respect to the demixing vector $\mathbf{w}$, it has been proven that the global demixing vector $\mathbf{g} = \mathbf{w}^{\mathrm{T}}\mathbf{A}$ tends to have only one non-zero element and consequently *only the source signal with the smallest normalised MSPE for the fixed linear predictor* $\boldsymbol{b}$ *will be extracted* [24].

Applying standard gradient descent optimisation to minimise $J(\mathbf{w})$, and using standard stochastic approximations [14], an online update [24] can be obtained as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\mu}{\sigma_y^2(k)}\left(e(k)\hat{\mathbf{x}}(k) - \frac{\sigma_e^2(k)}{\sigma_y^2(k)}y(k)\mathbf{x}(k)\right), \tag{18}$$

with μ being the learning rate and

$$\begin{aligned}\hat{\mathbf{x}}(k) &= \mathbf{x}(k) - \sum_{p=1}^{P} b_p\mathbf{x}(k-p), \\ \sigma_e^2(k) &= \beta_e\sigma_e^2(k-1) + (1-\beta_e)e^2(k), \\ \sigma_y^2(k) &= \beta_y\sigma_y^2(k-1) + (1-\beta_y)y^2(k),\end{aligned} \tag{19}$$

where β_e and β_y are the corresponding forgetting factors.

The so far introduced two classes of BSE algorithms have been designed for the case of noise-free mixtures. Although these can be applied directly to a general case, their performance will inevitably be degraded due to the presence of noise or nonlinearity for which they are not designed.

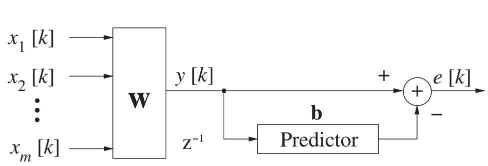


Fig. 3. BSE based on a linear predictor.

2.2. BSE with a linear predictor in noisy environments

In a noisy environment, to extract one of the sources, a demixing vector $\mathbf{w}$ is applied to noisy mixtures $\mathbf{x}(k)$, to obtain the source

$$y(k) = \mathbf{w}^{\mathrm{T}}\mathbf{x}(k) = \mathbf{g}^{\mathrm{T}}\mathbf{s}(k) + \mathbf{w}^{\mathrm{T}}\mathbf{v}\mathbf{n}(k). \tag{20}$$

If the sources are independent, the method from Section 2.1.1 is not perfectly suited, due to the presence of noise. However, if the variance of this noise can be estimated, the effect of noise can be removed from within the cost function [22]. More specifically, as the kurtosis of a Gaussian random variable is zero, the kurtosis of an extracted signal, $kt(y(k))$ will be the same as in the case with no noise; this way only the denominator of (6) need to be modified. This is equivalent to minimising

$$C(\mathbf{w}) = -\frac{\beta kt(y)}{4(E\{y^2\} - \mathbf{w}^{\mathrm{T}}\mathbf{R}_{\mathrm{vn}}\mathbf{w})^2}, \tag{21}$$

where $\mathbf{w}^{\mathrm{T}}\mathbf{R}_{\mathrm{vn}}\mathbf{w}$ is the contribution to the variance of the extracted signal $y(k)$ due to noise.

This idea can be extended to the case of BSE based on linear predictor and the effect of noise can be removed directly from the cost function (14) [23]; thus the generic form of the cost function remains the same as that in the noise-free case.

2.2.1. The cost function

Assume that noise $\mathbf{vn}(k)$ is uncorrelated with the source signals and that the noise correlation matrix is given by

$$\mathbf{R}_{vv}(\Delta k) = E\{\mathbf{vn}(k)\mathbf{vn}^{\mathrm{T}}(k-\Delta k)\} = \begin{cases} \mathbf{0} & \text{for } \Delta k \neq 0, \\ \mathbf{R}_{\mathrm{vn}} & \text{for } \Delta k = 0. \end{cases} \tag{22}$$

The MSPE at a linear prediction from Fig. 3. can now be expressed as

$$E\{e^2(k)\} = \mathbf{w}^{\mathrm{T}}\mathbf{R}_{xx}(0)\mathbf{w} - 2\sum_{p=1}^{P} b_p \mathbf{w}^{\mathrm{T}}\mathbf{R}_{xx}(p)\mathbf{w} + \sum_{p,q=1}^{P} b_p b_q \mathbf{w}^{\mathrm{T}}\mathbf{R}_{xx}(q-p)\mathbf{w}, \tag{23}$$

where $\mathbf{R}_{xx}(\Delta k)$ is the correlation matrix of the observed mixtures. After accounting for the effect of noise, from (1), we have

$$\begin{aligned}\mathbf{R}_{xx}(\Delta k) &= \mathbf{A}E\{\mathbf{s}(k)\mathbf{s}^{\mathrm{T}}(k-\Delta k)\}\mathbf{A}^{\mathrm{T}} + E\{\mathbf{vn}(k)\mathbf{vn}^{\mathrm{T}}(k-\Delta k)\} \\ &= \mathbf{A}\mathbf{R}_{\mathrm{ss}}(\Delta k)\mathbf{A}^{\mathrm{T}} + \mathbf{R}_{vv}(\Delta k).\end{aligned} \tag{24}$$

Thus, $E\{e^2(k)\}$ can be divided into two parts

$$E\{e^2(k)\} = e_{\mathrm{s}}^2 + e_{\mathrm{vn}}^2, \tag{25}$$

whereby the term $e_{\mathrm{s}}^2(k)$ is associated with the source signals, and is given by

$$\begin{aligned}e_{\mathrm{s}}^2(k) &= \mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{R}_{\mathrm{ss}}(0)\mathbf{A}^{\mathrm{T}}\mathbf{w} - 2\sum_{p=1}^{P} b_p \mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{R}_{\mathrm{ss}}(p)\mathbf{A}^{\mathrm{T}}\mathbf{w} \\ &\quad + \sum_{p,q=1}^{P} b_p b_q \mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{R}_{\mathrm{ss}}(q-p)\mathbf{A}^{\mathrm{T}}\mathbf{w} \\ &= \mathbf{w}^{\mathrm{T}}\mathbf{A}\hat{\mathbf{R}}_{\mathrm{ss}}\mathbf{A}^{\mathrm{T}}\mathbf{w}\end{aligned} \tag{26}$$

and $\hat{\mathbf{R}}_{\mathrm{ss}}$ is given in (16).

The term e_{vn}^2 corresponds to the noise term from within the mixtures, and is given by

$$\begin{aligned}e_{\mathrm{vn}}^2 &= \mathbf{w}^{\mathrm{T}}\mathbf{R}_{\mathrm{vn}}\mathbf{w} + \sum_{p=1}^{P} b_p^2 \mathbf{w}^{\mathrm{T}}\mathbf{R}_{\mathrm{vn}}\mathbf{w} \\ &= \sum_{p=0}^{P} b_p^2 \mathbf{w}^{\mathrm{T}}\mathbf{R}_{\mathrm{vn}}\mathbf{w},\end{aligned} \tag{27}$$

where $b_0 = 1$.

From (24), $E\{y^2(k)\}$ becomes

$$\begin{aligned}E\{y^2(k)\} &= \mathbf{w}^{\mathrm{T}}\mathbf{R}_{xx}(0)\mathbf{w} \\ &= \mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{R}_{\mathrm{ss}}(0)\mathbf{A}^{\mathrm{T}}\mathbf{w} + \mathbf{w}^{\mathrm{T}}\mathbf{R}_{\mathrm{vn}}\mathbf{w}.\end{aligned} \tag{28}$$

To remove the effects of noise from (14), a new cost function J_1, can be constructed, as [23]

$$J_1(\mathbf{w}) = \frac{E\{e^2(k)\} - e_{\mathrm{vn}}^2(k)}{E\{y^2(k)\} - \mathbf{w}^{\mathrm{T}}\mathbf{R}_{\mathrm{vn}}\mathbf{w}}. \tag{29}$$

Based on the above analysis, (29) can be written as

$$J_1(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{A}\hat{\mathbf{R}}_{\mathrm{ss}}\mathbf{A}^{\mathrm{T}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{R}_{\mathrm{ss}}(0)\mathbf{A}^{\mathrm{T}}\mathbf{w}}. \tag{30}$$

Observe that $J_1(\mathbf{w})$ from (30) is exactly the same as (17) (for more detail see [24]). We can therefore state that by minimising $J_1(\mathbf{w})$ with respect to $\mathbf{w}$, successful extraction of sources from noisy mixtures is achieved.

To build the cost function J_1, we first need to know $\mathbf{R}_{\mathrm{vn}}$. In practice, it is reasonable to assume $\mathbf{R}_{\mathrm{vn}} = \sigma_{\mathrm{vn}}^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix and σ_{vn}^2 is the variance of the noise. As

$$\mathbf{R}_{xx}(0) = \mathbf{A}\mathbf{R}_{\mathrm{ss}}(0)\mathbf{A}^{\mathrm{T}} + \sigma_{\mathrm{vn}}^2\mathbf{I}, \tag{31}$$

when $m > n$, (that is, the number of mixtures is larger than the number of sources), a subspace method can be used to estimate σ_{vn}^2 (which in this case represents the smallest eigenvalue of $\mathbf{R}_{xx}(0)$ [14]). Alternatively, for noisy mixtures, an adaptive principal component analysis method may be employed to estimate σ_{vn}^2 online [17].

2.2.2. BSE algorithms for noisy mixtures

To simplify the derivation of the adaptive algorithms, a normalisation of $\mathbf{w}$ will be performed after each update, which can be expressed as

$$\mathbf{w}(k+1) \leftarrow \mathbf{w}(k+1)/\sqrt{\mathbf{w}^{\mathrm{T}}(k+1)\mathbf{w}(k+1)}, \tag{32}$$

which also helps to avoid the critical case when the norm of $\mathbf{w}(k)$ becomes too small. Thus, we have

$$e_{\mathrm{vn}}^2 = \sum_{p=0}^{P} b_p^2 \sigma_{\mathrm{vn}}^2 = \sigma_c^2 \sigma_{\mathrm{vn}}^2, \tag{33}$$

where $\sigma_c^2 = \sum_{p=0}^{P} b_p^2$ is a constant. Now $J_1(\mathbf{w})$ becomes

$$J_1(\mathbf{w}) = \frac{E\{e^2(k)\} - \sigma_c^2\sigma_{\mathrm{vn}}^2}{E\{y^2(k)\} - \sigma_{\mathrm{vn}}^2}. \tag{34}$$

Applying standard gradient descent to $J_1(\mathbf{w})$, yields

$$\nabla_{\mathbf{w}} J_1 = \frac{2}{E\{y^2(k)\} - \sigma_{\text{vn}}^2}(E\{e(k)\hat{\mathbf{x}}(k)\} - \frac{E\{e^2(k)\} - \sigma_c^2\sigma_{\text{vn}}^2}{E\{y^2(k)\} - \sigma_{\text{vn}}^2} E\{y(k)\mathbf{x}(k)\}, \tag{35}$$

where $E\{e^2(k)\}$ and $E\{y^2(k)\}$ can be estimated, respectively, by

$$\sigma_e^2(k) = \beta_e\sigma_e^2(k-1) + (1-\beta_e)e^2(k),$$
$$\sigma_y^2(k) = \beta_y\sigma_y^2(k-1) + (1-\beta_y)y^2(k), \tag{36}$$

where β_e and β_y are the corresponding forgetting factors with $0 \leqslant \beta_e, \beta_y < 1$.

Using again standard stochastic approximations, the following update of the demixing vector is obtained [23]

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{2\mu}{\sigma_y^2(k) - \sigma_{\text{vn}}^2} \times \left(e(k)\hat{\mathbf{x}}(k) - \frac{\sigma_e^2(k) - \sigma_c^2\sigma_{\text{vn}}^2}{\sigma_y^2(k) - \sigma_{\text{vn}}^2} y(k)\mathbf{x}(k) \right). \tag{37}$$

The above algorithm is a direct gradient method and is suitable for online implementation (as there is no preprocessing involved). Notice that by a combination of preprocessing and normalisation of the demixing vector $\mathbf{w}(k)$, the variance of the extracted signal $y(k)$ can be reduced to a fixed value, which can greatly simplify the cost function and the associated adaptive algorithm.

Note that $\mathbf{R}_{xx}(0)$ can be decomposed as

$$\mathbf{R}_{xx}(0) = \mathbf{E}\mathbf{D}_x\mathbf{E}^{\mathrm{T}}, \tag{38}$$

where $\mathbf{E}$ is an orthogonal matrix whose columns are the eigenvectors of $\mathbf{R}_{xx}(0)$, and $\mathbf{D}_x$ is a diagonal matrix whose diagonal elements are the corresponding eigenvalues. Multiplying the mixtures by the whitening matrix $\mathbf{P} = \mathbf{D}_x^{-1/2}\mathbf{E}^{\mathrm{T}}$, we obtain the prewhitened vector $\tilde{\mathbf{x}}$ as

$$\tilde{\mathbf{x}} = \mathbf{D}_x^{-1/2}\mathbf{E}^{\mathrm{T}}\mathbf{x}. \tag{39}$$

The correlation matrix of the prewhitened mixtures $\mathbf{R}_{\tilde{x}\tilde{x}}(0)$ now becomes

$$\begin{aligned}\mathbf{P}_{\tilde{x}\tilde{x}}(0) &= \mathbf{P}\mathbf{R}_{xx}(0)\mathbf{P}^{\mathrm{T}} \\ &= \mathbf{P}\mathbf{A}\mathbf{R}_{\text{ss}}(0)\mathbf{A}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}} + \mathbf{P}\mathbf{R}_{\text{vn}}\mathbf{P}^{\mathrm{T}} \\ &= \hat{\mathbf{A}}\mathbf{R}_{\text{ss}}(0)\hat{\mathbf{A}}^{\mathrm{T}} + \hat{\mathbf{R}}_v = \mathbf{I}\end{aligned} \tag{40}$$

with $\hat{\mathbf{A}} = \mathbf{P}\mathbf{A}$ and $\hat{\mathbf{R}}_v = \mathbf{P}\mathbf{R}_{\text{vn}}\mathbf{P}^{\mathrm{T}}$.

Due to prewhitening, the numerator and the denominator of (14) become, respectively

$$\begin{aligned}E\{e^2(k)\} &= e_{\text{s}}^2 + e_{\text{vn}}^2 \\ &= \mathbf{w}^{\mathrm{T}}\hat{\mathbf{A}}\hat{\mathbf{R}}_{\text{ss}}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{w} + \sum_{p=0}^{P} b_p^2\mathbf{w}^{\mathrm{T}}\hat{\mathbf{R}}_v\mathbf{w},\end{aligned} \tag{41}$$

$$\begin{aligned}E\{y^2(k)\} &= \mathbf{w}^{\mathrm{T}}\mathbf{R}_{\hat{x}\hat{x}}(0)\mathbf{w} = \mathbf{w}^{\mathrm{T}}\mathbf{w} \\ &= \mathbf{w}^{\mathrm{T}}\hat{\mathbf{A}}\mathbf{R}_{\text{ss}}(0)\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{w} + \mathbf{w}^{\mathrm{T}}\hat{\mathbf{R}}_v\mathbf{w}.\end{aligned} \tag{42}$$

From (30), after the multiplication of $\mathbf{P}$ with the mixing matrix $\mathbf{A}$, the cost function $J_1(\mathbf{w})$ becomes

$$\begin{aligned}J_1(\mathbf{w}) &= \frac{E\{e^2(k)\} - \sum_{p=0}^{P} b_p^2\mathbf{w}^{\mathrm{T}}\hat{\mathbf{R}}_v\mathbf{w}}{E\{y^2(k)\} - \mathbf{w}^{\mathrm{T}}\hat{\mathbf{R}}_v\mathbf{w}} \\ &= \frac{E\{e^2(k)\} - \sigma_c^2\mathbf{w}^{\mathrm{T}}\hat{\mathbf{R}}_v\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{w} - \mathbf{w}^{\mathrm{T}}\hat{\mathbf{R}}_v\mathbf{w}} \\ &= \frac{E\{e^2(k)\} - \sigma_c^2\mathbf{w}^{\mathrm{T}}\hat{\mathbf{R}}_v\mathbf{w}}{\mathbf{w}^{\mathrm{T}}(\mathbf{I} - \hat{\mathbf{R}}_v)\mathbf{w}}.\end{aligned} \tag{43}$$

Further, if the denominator $\mathbf{w}^{\mathrm{T}}(\mathbf{I} - \hat{\mathbf{R}}_v)\mathbf{w}$ is used to normalise $\mathbf{w}$ at every update, that is

$$\mathbf{w}(k+1) \leftarrow \mathbf{w}(k+1)/\sqrt{\mathbf{w}^{\mathrm{T}}(\mathbf{I} - \hat{\mathbf{R}}_v)\mathbf{w}}, \tag{44}$$

then

$$E\{y^2(k)\} - \mathbf{w}^{\mathrm{T}}\hat{\mathbf{R}}_v\mathbf{w} = \mathbf{w}^{\mathrm{T}}(\mathbf{I} - \hat{\mathbf{R}}_v)\mathbf{w} = 1. \tag{45}$$

This way, the cost function is simplified to

$$\begin{aligned}J_1(\mathbf{w}) &= E\{e^2(k)\} - \sigma_c^2\mathbf{w}^{\mathrm{T}}\hat{\mathbf{R}}_v\mathbf{w} \\ &= E\{e^2(k)\} - \sigma_c^2(E\{y^2(k)\} - 1).\end{aligned} \tag{46}$$

In this case, for linear predictor with fixed coefficients, after applying standard gradient descent to $J_1(\mathbf{w})$, the following online update rule is obtained [23]

$$\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu(e(k)\hat{\tilde{\mathbf{x}}}(k) - \sigma_c^2 y(k)\tilde{\mathbf{x}}(k)), \tag{47}$$

where $\hat{\tilde{\mathbf{x}}}(k)$ is given by

$$\hat{\tilde{\mathbf{x}}}(k) = \tilde{\mathbf{x}}(k) - \sum_{p=1}^{P} b_p\tilde{\mathbf{x}}[k-p]. \tag{48}$$

This update equation is followed by the normalisation operation given in (44). The simulation results are shown in Section 4.

3. Post-nonlinear BSE

The algorithms introduced in Section 2 are designed for linear mixtures, a condition which is not realistic for a range of real world sensors; for instance those with saturation type nonlinearity. To help mitigate this problem, nonlinear models ought to be considered [29,30]; one such model is the PNL mixing model [30], a focus of this Section.

3.1. PNL mixtures

In the PNL mixing model, the n unknown sources are first mixed linearly by a mixing matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, the measurements $\mathbf{x}(k)$ are obtained through a nonlinear vector function $\mathbf{M}(\cdot) = [M_1(\cdot), \ldots, M_m(\cdot)]^{\mathrm{T}}$, as shown in Fig. 4 given by

$$\mathbf{x}(k) = \mathbf{M}(\mathbf{A}\mathbf{s}(k)).$$

Our goal is to extract the sources of interest without any prior knowledge of their distributions and the features of

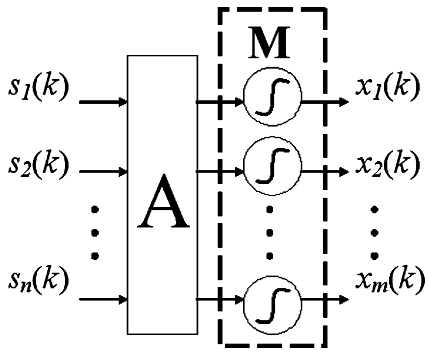


Fig. 4. Block diagram of the post-nonlinear mixing model.

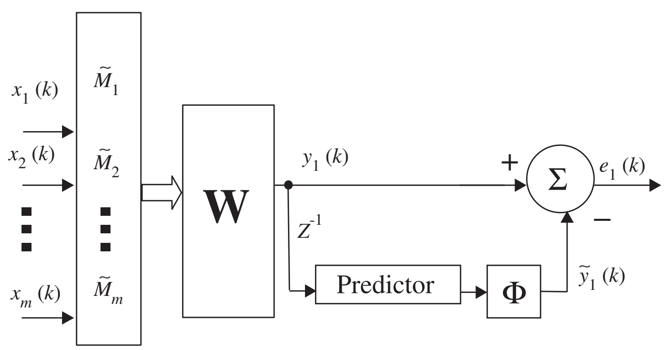


Fig. 5. A BSE structure for the extraction of a single source using a nonlinear adaptive FIR filter with nonlinear activation function Φ.

the nonlinear mixing mechanism. By extending the results from Section 2, source extraction in this case can be achieved by first estimating the inverse of the nonlinearities in the mixing process and then applying a demixing vector $\mathbf{w}_1$ to extract the desired signal, given by

$$y_1(k) = \mathbf{w}_1^{\mathrm{T}}\mathbf{M}^{-1}(\mathbf{x}(k)) = \mathbf{w}_1^{\mathrm{T}}\tilde{\mathbf{M}}(\mathbf{x}(k)), \tag{49}$$

where

$$\tilde{\mathbf{M}} = \mathbf{M}^{-1} = [\tilde{M}_1(\cdot), \ldots, \tilde{M}_m(\cdot)]^{\mathrm{T}},$$
$$\mathbf{w}_1 = [w_{11}(k), \ldots, w_{1m}(k)]^{\mathrm{T}}. \tag{50}$$

The learning rule involves both the estimation of the demixing vector $\mathbf{w}_1$ and the nonlinearity. The estimation of $\mathbf{w}_1$ is based on the nonlinear predictability of the sources, as shown in Fig. 5.

3.2. Estimating the inverse of the nonlinearity

To estimate $\tilde{\mathbf{M}}$, the alternating conditional expectation (ACE) algorithm [3,19,27,31–33] may be exploited to approximately invert the component-wise nonlinear functions within $\mathbf{M}$. Using ACE, to make the inverse of the nonlinear functions M_1 and M_2, the quantity to be minimised is

$$corr(\tilde{M}_1(x_1), \tilde{M}_2(x_2)) \tag{51}$$

with respect to some nonlinear functions $\tilde{M}_1$ and $\tilde{M}_2$ [3]. The aim here is to find the transformations $\tilde{M}_1$ and $\tilde{M}_2$ such that the relationship between the transformed variables becomes as linear as possible.

The ACE method finds the optimal transformation $\tilde{M}_1^*$ and $\tilde{M}_2^*$ which maximise (51) [3]. For fixed $\tilde{M}_1$ the optimal $\tilde{M}_2$ is given by

$$\tilde{M}_2(x_2) = E\{\tilde{M}_1(x_1)|x_2\}$$

and conversely, for fixed $\tilde{M}_2$ and the optimal $\tilde{M}_1$ is

$$\tilde{M}_1(x_1) = E\{\tilde{M}_2(x_2)|x_1\}. \tag{52}$$

The idea of the iterative ACE algorithm is then to compute these conditional expectations alternatively until convergence has been reached. The optimality is defined by the least squares criterion, which yields

$$\varepsilon_r^2 = \min_{\tilde{M}_1,\tilde{M}_2} \frac{E\{\tilde{M}_1(x_1) - \tilde{M}_2(x_2)\}^2}{E\{\tilde{M}_1^2(x_1)\}}, \tag{53}$$

The minimisation of ε_r^2 is carried out through a series of single function minimisations. The optimal transformation in the least squares sense is equivalent to the transformation with maximal correlation,

$$\max_{\tilde{M}_1,\tilde{M}_2} corr(\tilde{M}_1(x_1), \tilde{M}_2(x_2)), \tag{54}$$

This equivalence, which was proven in [3], also reveals the ways to extend the ACE algorithm to the multivariate case. The underlying idea for applying the ACE framework to the PNL problem is based on the fact that we can approximate the inverse of the post-nonlinearities by searching for those nonlinear transformations which *maximise* the *linear* correlations between the *nonlinearly* transformed observed variables (more detail can be found in the Appendix).

3.3. Blind extraction of sources based on nonlinear predictability

Now that we are able to find estimates of nonlinear transformations $\tilde{\mathbf{M}}$ from Fig. 5, a standard extraction process (as shown in Section 2) with extracting coefficients $\mathbf{w}_1(k)$ is used to extract one signal (denoted by $y_1(k)$) from the mixtures. A nonlinear adaptive finite impulse response (FIR) filter with coefficients $\mathbf{b}_1(k)$ is used as a predictor to assist the extraction. Such a nonlinear predictor is particularly important to support the ACE algorithm in eliminating the effects of the remaining nonlinearities. In Fig. 5, the filter output $\tilde{y}_1(k)$ is an estimate of the extracted signal $y_1(k)$ and the filter nonlinearity $\Phi(\cdot)$ is typically a sigmoid function. The estimation of the extracted signal $y_1(k)$ is naturally accompanied by the corresponding prediction error, defined by

$$e_1(k) = y_1(k) - \tilde{y}_1(k), \tag{55}$$

where

$$\tilde{y}_1(k) = \Phi(\mathbf{y}_1^{\mathrm{T}}\mathbf{b}_1(k)),$$
$$\mathbf{b}_1(k) = [b_{11}(k), b_{12}(k), \ldots, b_{1p}(k)]^{\mathrm{T}},$$
$$\mathbf{y}_1(k) = [y_1(k-1), y_1(k-2), \ldots, y_1(k-P)]^{\mathrm{T}}. \quad (56)$$

To adjust the time varying filter coefficients $\mathbf{b}_1(k)$ and the extracting coefficients $\mathbf{w}_1(k)$, a gradient descent algorithm is next introduced; this is based on the minimisation of the prediction error $e_1(k)$. Similarly to the previous cases, the cost function for the first extracted signal is therefore defined as

$$J(\mathbf{w}_1(k), \mathbf{b}_1(k)) = \tfrac{1}{2}e_1^2(k). \quad (57)$$

The extracted signal $y_1(k)$ and its estimate $\tilde{y}_1(k)$ can be found from

$$y_1(k) = \sum_{i=1}^{m} x_i(k)w_{1i}(k),$$
$$\tilde{y}_1(k) = \Phi\left(\sum_{p=1}^{P} b_{1p}(k)y_1(k-p)\right)$$
$$= \Phi\left(\sum_{p=1}^{P} b_{1p}(k)\sum_{i-1}^{m} x_i(k-p)w_{1i}(k-p)\right). \quad (58)$$

To derive a gradient descent adaptation for every element $b_{1p}(k)$, $p = 1, 2, \ldots, P$ of the filter coefficient vector $\mathbf{b}_1$ and every element $w_{1i}(k)$, $i = 1, 2, \ldots, m$ of the extracting coefficient vector $\mathbf{w}_1$, we need to calculate the corresponding gradients within the updates

$$b_{1p}(k+1) = b_{1p}(k) - \mu_b \nabla_{b_{1p}} J(\mathbf{w}_1(k), \mathbf{b}_1(k)), \quad (59)$$
$$w_{1i}(k+1) = w_{1i}(k) - \mu_w \nabla_{w_{1i}} J(\mathbf{w}_1(k), \mathbf{b}_1(k)), \quad (60)$$

where μ_b and μ_w denote, respectively, the learning rates for the adaptation of $\mathbf{b}_1$ and $\mathbf{w}_1$. From the prediction error term

$$e_1(k) = y_1(k) - \tilde{y}_1(k)$$
$$= \sum_{i=1}^{m} x_i(k)w_{1i}(k) - \Phi\left(\sum_{p=1}^{P} b_{1p}(k) \times \sum_{i-1}^{m} x_i(k-p)w_{1i}(k-p)\right) \quad (61)$$

we have

$$\nabla_{b_{1p}} J(\mathbf{w}_1(k), \mathbf{b}_1(k)) = e_1(k)\frac{\partial e_1(k)}{\partial b_{1p}(k)} = -e_1(k)\Phi'(k)y_1(k-p), \quad (62)$$

where, for convenience, $\Phi(k)$ denotes the value of the nonlinear function $\Phi(\mathbf{b}_1^{\mathrm{T}}(k)\mathbf{w}_1(k))$ from (61) at time instant k, $\Phi'(k)$ denotes its derivative, and

$$\nabla_{w_{1i}} J(\mathbf{w}_1(k), \mathbf{b}_1(k)) = e_1(k)x_i(k). \quad (63)$$

The updates for the filter and the extracting coefficients now become

$$b_{1p}(k+1) = b_{lp}(k) + \mu_b(k)e_1(k)\Phi'(k)y_1(k-p),$$
$$w_{1i}(k+1) = w_{1i}(k) - \mu_w e_1(k)x_i(k) \quad (64)$$

which can be expressed more compactly in the vector form as

$$\mathbf{b}_1(k+1) = \mathbf{b}_1(k) + \mu_b(k)e_1(k)\Phi'(k)y_1(k),$$
$$\mathbf{w}_1(k+1) = \mathbf{w}_1(k) - \mu_w e_1(k)\mathbf{x}(k). \quad (65)$$

This completes the derivation of the adaptive BSE algorithm for post-nonlinearly mixed sources.

4. Experiments

We now provide experimental support for the presented BSE algorithms. We shall consider the cases of noisy mixtures and PNL mixtures separately. This is because each have their own specific merits. Blind PNL extraction of noisy mixtures is a difficult and open problem which is beyond the scope of this work.

4.1. Simulations for noisy mixtures

Three source signals shown in Fig. 6 were used in simulations (these can be found in the ICALAB toolbox [4]). The coefficients of the linear predictor with a length of $P = 3$ were randomly generated, and are given by

$$\mathbf{b} = [0.2\ 0.9\ 0.1]. \quad (66)$$

This way, the obtained normalised prediction errors of the three source signals were, respectively, $\{1.61, 1.05, 0.05\}$. The 4×3 mixing matrix $\mathbf{A}$ was randomly generated and is

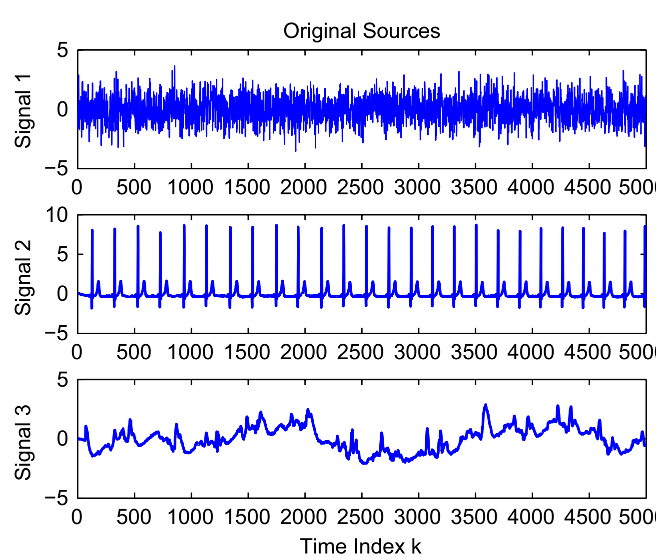


Fig. 6. The three source signals used in the simulations.

given by

$$\mathbf{A} = \begin{bmatrix} -0.2121 & 0.0724 & -0.3481 \\ -2.0734 & -0.6688 & 1.2110 \\ 1.2410 & -0.2463 & -1.0882 \\ -1.7048 & -1.0232 & 0.0708 \end{bmatrix}. \quad (67)$$

As there is one more mixture than the number of sources, this additional degree of freedom was used to estimate the variance of the additive white Gaussian noise. To further illustrate the proposed approach, two different noise levels were used. For the first set of simulations, the variance of the noise was $\sigma_{\mathrm{vn}}^2 = 0.01$. By minimising the normalised MSPE, it is expected that the signal with the smallest normalised prediction error would be extracted, in our case this is the third signal.

For comparison, the direct BSE algorithm (designed for noisy mixtures) given in Eq. (37) was first tested. The forgetting factors were $\beta_e = \beta_y = 0.98$ and the stepsize $\mu = 0.0017$. The learning curve for this case is shown in Fig. 7, whereby the performance index was defined as [4]

$$PI = 10\log_{10}\left(\frac{1}{n-1}\left(\sum_{m=0}^{n-1}\frac{g_m^2}{\max\{g_0^2, g_1^2, \ldots, g_{n-1}^2\}} - 1\right)\right), \quad (68)$$

with $\mathbf{g} = \mathbf{A}^{\mathrm{T}}\mathbf{w} = [g_0 \quad g_1 \quad \cdots \quad g_{n-1}]^{\mathrm{T}}$. The resulting signal to interference plus noise ratio (SINR) was about 16.5 dB. As the performance index reached the level well below −30 dB on the average, we can say signal 3 had been extracted successfully, as shown in Fig. 8.

To further illustrate the performance of the proposed algorithm, in the next experiment, the variance of noise was

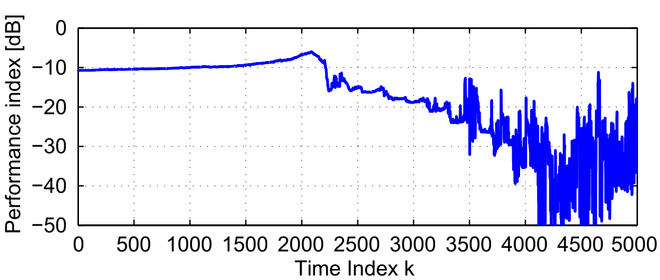


Fig. 7. The learning curve using the algorithm (37) with $\sigma_{\mathrm{vn}}^2 = 0.01$.

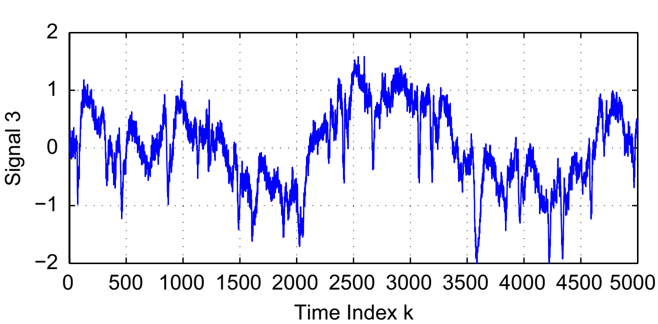


Fig. 8. The extracted signal 3 using the algorithm (37).

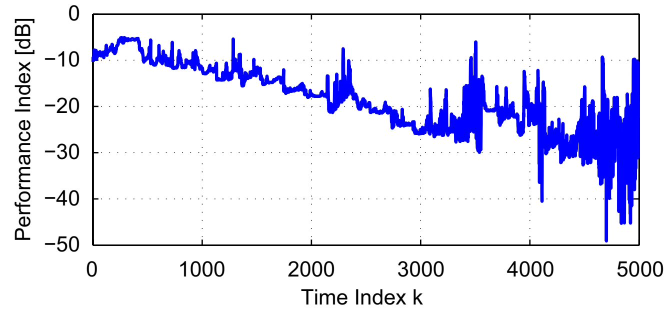


Fig. 9. The learning curve using the algorithm (37) with $\sigma_{\mathrm{vn}}^2 = 0.04$.

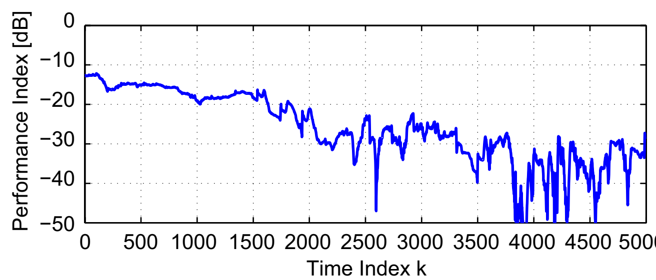


Fig. 10. The learning curve using the algorithm (47) with $\sigma_{\mathrm{vn}}^2 = 0.01$.

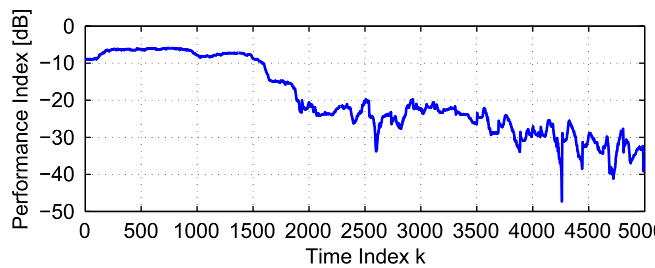


Fig. 11. The learning curve using the algorithm (47) with $\sigma_{\mathrm{vn}}^2 = 0.04$.

increased to $\sigma_{\mathrm{vn}}^2 = 0.04$. The associated learning curve is shown in Fig. 9 with the resulting SINR value of approximately 10.6 dB. The corresponding value of the performance index at the steady state (well below 20 dB on the average) indicates a successful extraction.

Next, the same experiments were constructed for the algorithm given in (47). The stepsize was chosen to be $\mu = 3$. It may at first seem surprising to have such a large stepsize, but this is a direct consequence of a very small instantaneous gradient value $2(e(k)\hat{\mathbf{x}}(k) - \sigma_c y(k)\mathbf{x}(k))$, which takes values in the region of 10^{-3} during the adaptation. This stepsize is also much larger (about 1000 times) than that for the direct approach (the first set of simulations). This is also consistent with the observation for the noise-free BSE algorithms [24], where the stepsize when prewhitening was performed was about 10^3 times the value of that when using the normalised MSPE as the cost function. This clearly indicates a close relationship between the prewhitening and the value of the stepsize. The corresponding learning curves for the two different noise levels are shown, respectively, in Figs. 10 and 11.

The resulting SINRs were, respectively, about 15.9 and 11.1 dB, and in both cases the algorithm was able to extract the source successfully.

Finally, a comparison between the algorithm given in (37) and that in (18) is provided. The number of sources was increased to four and the fourth source had similar characteristics to signal 1 (Fig. 6), with a normalised MSPE of 1.88. The noise variance was $\sigma_{\mathrm{vn}}^2 = 0.16$. The learning curves of the two algorithms with a stepsize $\mu = 0.0024$ are shown in Fig. 12, which were obtained by averaging 1000 independent simulations, each with a randomly generated mixing matrix and a randomly generated initial value of the demixing vector. Observe that some of the trials may have had an ill-conditioned or nearly ill-conditioned mixing matrix; this affected the performance of both the algorithms. From Fig. 12 notice that in the case of noisy mixtures, the algorithm designed for the noisy case outperformed the original one, as illustrated by the performance index of the proposed method reaching a lower level.

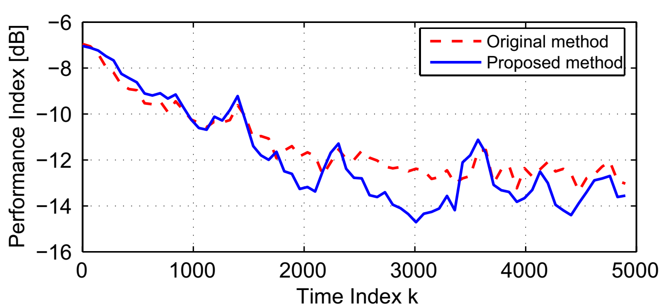


Fig. 12. The learning curves of the algorithm in (37) and the original algorithm in (18), obtained by averaging 1000 independent trials.

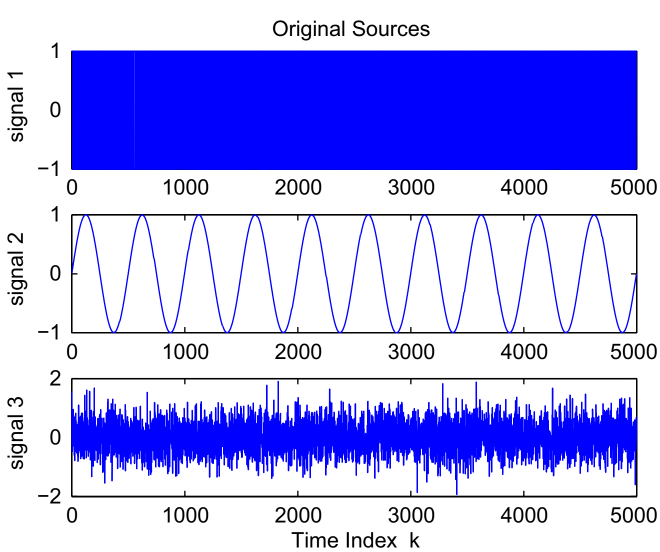


Fig. 13. Source signals with binary (top), sine waveform (middle) and Gaussian distribution (bottom).

4.2. Simulations for PNL mixtures

To illustrate the operation of BSE from nonlinear mixtures, the simulations were based on three source signals: s_1 with binary distribution, s_2 a sine waveform and s_3 with Gaussian distribution, as shown in Fig. 13. The signals s_1 and s_2 had positive kurtosis ($\beta = 1$), the adaptive filter length was $P = 3$, and learning rates μ_b and $\mu_w = 0.0001$. Monte Carlo simulations with 5000 iterations of independent trials were performed.

The initial values of the predictor and the demixing vector $\mathbf{w}_1(0)$ coefficients were randomly generated for each run. The simulations were conducted without prewhitening. Matrix $\mathbf{A}$ was a 3×3 mixing matrix and was randomly

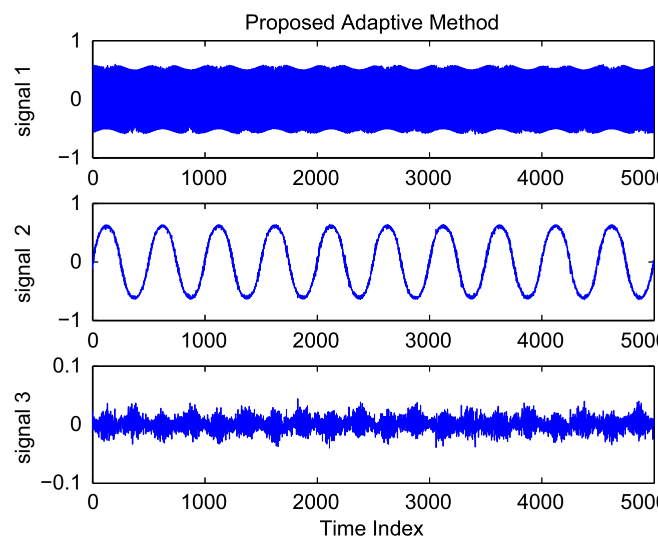


Fig. 14. Extracted output signals with binary (top), sine waveform (middle) and Gaussian distribution (bottom) using the post-nonlinear BSE.

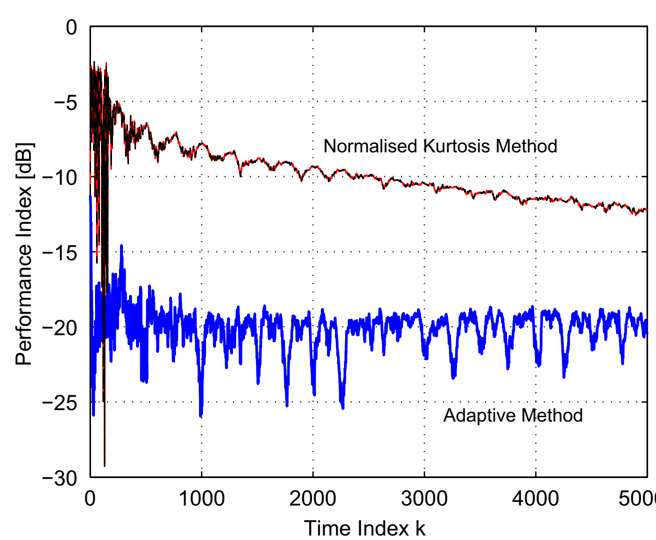


Fig. 15. Learning curves of the BSE for post-nonlinear mixtures; standard BSE vs post-nonlinear BSE.

generated, given by

$$\mathbf{A} = \begin{bmatrix} 0.0413 & 0.1025 & -0.9389 \\ -0.4786 & -0.3694 & 0.8490 \\ 0.6005 & -0.0585 & -0.0180 \end{bmatrix}. \tag{69}$$

The PNL mixtures were based on a saturation type nonlinearity, in this case

$$\mathbf{x}(k) = \mathbf{tanh}(\mathbf{A}\mathbf{s}(k)). \tag{70}$$

The waveforms of the sequentially extracted signals by the proposed algorithm are given in Fig. 14, matching the

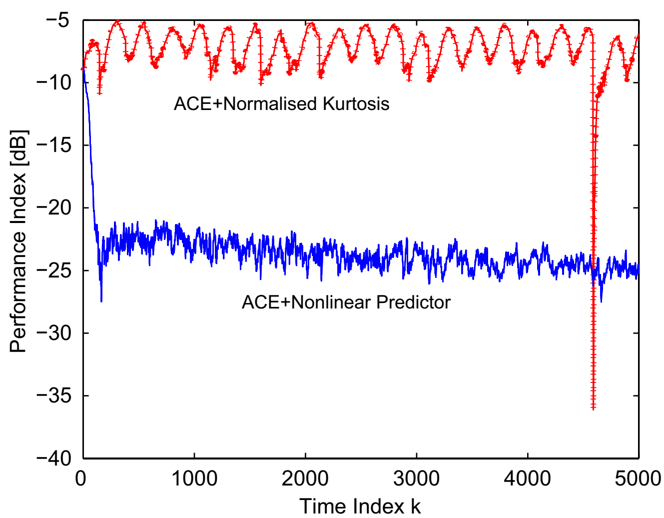


Fig. 16. Comparison between ACE + normalised kurtosis and ACE + nonlinear predictor algorithms.

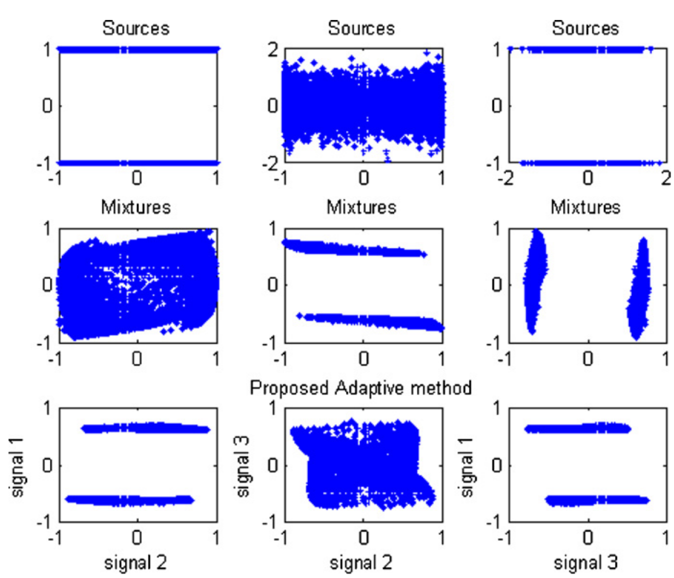


Fig. 17. Scatter plots comparing the independence of the output signals in noiseless environment. Column 1: signals 1 and 2; Column 2: signals 2 and 3; Column 3: signals 1 and 3. Row 1: original sources; Row 2: mixtures; Row 3: the proposed adaptive PNL method.

first source signal. Fig. 15 illustrates the values of PI, along the adaptation, for both the normalised kurtosis based [22] and the proposed PNL extraction approach. For the PNL extraction, the PI reached a level of around −30 dB after 1000 iterations, indicating successful PNL blind extraction; the extraction using the normalised kurtosis was not successful. Next, the comparison between ACE + normalised kurtosis and ACE + nonlinear predictor is shown in Fig. 16. The PNL approach exhibited better qualitative performance clearly indicating the usefulness of the nonlinear predictor in assisting the extraction process.

To illustrate the *qualitative* performance of the introduced algorithms, scatter plots of the original sources and the recovered output signals are displayed in Fig. 17. These scatter plots illustrate the degree of independence between the outputs, whereby every point on the diagram corresponds to a data vector. Conforming with the above results, the extracted output signals using the PNL method outperformed the normalised kurtosis [22] based extraction.

5. Conclusions

An overview of BSE algorithms has been provided and some of their extensions are proposed. First, BSE approaches for linear instantaneous mixtures are reviewed with an emphasis on the linear predictor based approach, including algorithms for both noise-free and noisy mixing. To help circumvent some of the problems associated with the assumption of linear mixing, an extension in the form of PNL mixing system is addressed, this is achieved for a special case of invertible sensor nonlinearities. Simulation results have been provided to support the introduced algorithms for both the case of linear, PNL and noisy mixtures.

Appendix

To implement the ACE algorithm, estimation of the conditional expectations from the data is very important. These conditional expectations are computed by data smoothing for which numerous techniques exist [3]. In the following, applying $\tilde{M}_1^*$ and $\tilde{M}_2^*$ to the mixed signals x_1 and x_2 removes the effect of the nonlinear functions M_1 and M_2. To that end, we show that for $F_1 = A_{11}s_1 + A_{12}s_2$ and $F_2 = A_{21}s_1 + A_{22}s_2$ functions $\tilde{M}_1^*$ and $\tilde{M}_2^*$ obtained from the ACE procedure are the desired inverse functions given that F_1 and F_2 are jointly normally distributed. In other words, the following relationship holds:

$$h_1^*(F_1) := \tilde{M}_1^*(M_1(F_1)) \propto F_1,$$
$$h_2^*(F_2) := \tilde{M}_2^*(M_2(F_2)) \propto F_2. \tag{71}$$

The above statement proceeds exactly along the idea from [3,19,31–33]. Noticing that the correlation between two signals does not change, if we scale one or both signals and following, it can be stated that

$$\tilde{M}_1^*(x_1) \propto E\{\tilde{M}_2^*(x_2)|x_1\},$$
$$\tilde{M}_2^*(x_2) \propto E\{\tilde{M}_1^*(x_1)|x_2\}. \tag{72}$$

Here the conditional expectation $E\{\tilde{M}_2^*(x_2)|x_1\}$ is a function of x_1 and the expectation is taken with respect to x_2, analogously with the second expression.

Since $\tilde{M}_1^*(x_1) = h_1^*(F_1)$ and $\tilde{M}_2^*(x_2) = h_2^*(F_2)$, furthermore $x_1 = M_1(F_1)$ and $x_2 = M_2(F_2)$ we obtain

$$h_1^*(F_1) \propto E\{h_2^*(z_2)|M_1(F_1)\},$$
$$h_2^*(F_2) \propto E\{h_1^*(z_1)|M_2(F_2)\}. \tag{73}$$

Since M_1 and M_2 are invertible functions, they can be omitted from the conditional expectation, leading to

$$h_1^*(F_1) \propto E\{h_2^*(F_2)|F_1\},$$
$$h_2^*(F_2) \propto E\{h_1^*(F_1)|F_2\}. \tag{74}$$

Assuming that the vector $(F_1, F_2)^{\mathrm{T}}$ is normally distributed and the correlation coefficient $corr(F_1, F_2)$ is non-zero, a straightforward calculation shows

$$E\{F_2|F_1\} \propto F_1,$$
$$E\{F_1|F_2\} \propto F_2 \tag{75}$$

meaning that F_1 and F_2 satisfy (74), which then immediately implies (73).

The above assumptions are usually fulfilled in nonlinear mixtures since mixtures of independent signals have a distribution that is closer to the Gaussian than the most non-Gaussian ones and they are also more correlated than the unmixed signals. This suggests that the ACE algorithm may be able to equalise the nonlinearities, even if the assumptions are not perfectly met, though there might be difficulties in the presence of large variance noise or low correlations.

References

[1] S. Amari, Natural gradient works efficiently in learning, Neural Comput. 10 (1998) 251–276.

[2] A.K. Barros, A. Cichocki, Extraction of specific signals with temporal structure, Neural Comput. 13 (2001) 1995–2003.

[3] L. Breiman, J.H. Friedman, Estimating optimal transformations for multiple regression and correlation, J. Amer. Statist. Assoc. 80 (1985) 580–598.

[4] A. Cichocki, S. Amari, Adaptive Blind Signal and Image Processing, Wiley, New York, 2003.

[5] A. Cichocki, T. Rutkowski, K. Siwek, Blind signal extraction of signals with specified frequency band, in: Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, September 2002, pp. 515–524.

[6] A. Cichocki, R. Thawonmas, On-line algorithm for blind signal extraction of arbitrarily distributed, but temporally correlated sources using second order statistics, Neural Process. Lett. 12 (2000) 91–98.

[7] A. Cichocki, R. Thawonmas, S. Amari, Sequential blind signal extraction in order specified by stochastics properties, IEE Electron. Lett. 33 (1997) 64–65.

[8] P. Comon, Independent component analysis: a new concept? (special issue on higher-order statistics), Signal Process. 36 (3) (1994) 287–314.

[9] S. Cruces, A. Cichocki, L. Castedo, Blind signal extraction in Gaussian noise, in: Proceedings of the International Workshop on Independent Component Analysis and Blind Source Separation, June 2002, pp. 63–68.

[10] S.A. Cruces-Alvarez, A. Cichocki, S. Amari, From blind signal extraction to blind instantaneous signal separation: criteria, algorithms, and stability, IEEE Trans. Neural Networks 15 (4) (2004) 859–873.

[11] N. Delfosse, P. Loubaton, Adaptive blind separation of independent sources: a deflation approach, Signal Process. 95 (1995) 59–83.

[12] S.C. Douglas, Blind signal separation and blind deconvolution, in: Y.H. Hu, J.N. Hwang (Eds.), Handbook of Neural Network Signal Processing, CRC Press, Boca Raton, Florida, 2002 (Chapter 7).

[13] S.C. Douglas, S.Y. Kung, Kuicnet algorithms for blind deconvolution, in: Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, Cambridge, UK, 1998, pp. 3–12.

[14] S. Haykin, Adaptive Filter Theory, third ed., Prentice-Hall, Englewood Cliffs, NJ, 1996.

[15] S. Haykin, Unsupervised Adaptive Filtering, Blind Source Separation, vol. 2, Wiley, New York, 2000.

[16] S. Haykin, Unsupervised Adaptive Filtering, Blind Deconvolution, vol. 2, Wiley, New York, 2000.

[17] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley, New York, 2001.

[18] A. Hyvarinen, E. Oja, A fast fixed-point algorithm for independent component analysis, Neural Comput. 9 (1997) 1483–1492.

[19] C. Jutten, J. Karhunen, L. Almeida, S. Harmeling, Separation methods for nonlinear mixtures, Technical Report, Institut National Polytechnique of Grenoble, France, October 8, 2003.

[20] Y.Q. Li, J. Wang, Sequential blind extraction of instantaneously mixed sources, IEEE Trans. Signal Process. 50 (2002) 997–1006.

[21] Y.Q. Li, J. Wang, A. Cichocki, Blind source extraction from convolutive mixtures in ill-conditioned multi-input multi-output channels, IEEE Trans. Circuits Syst. II Regular Pap. 51 (2004) 1814–1822.

[22] W. Liu, D.P. Mandic, A normalised kurtosis based algorithm for blind source extraction from noisy measurements, Signal Process. Elsevier 86 (7) (2006) 1580–1585.

[23] W. Liu, D.P. Mandic, A. Cichocki, Blind second-order source extraction of instantaneous noisy mixtures, IEEE Trans. Circuits Syst. II Express Briefs 53 (9) (2006) 931–935.

[24] W. Liu, D.P. Mandic, A. Cichocki, Blind source extraction based on a linear predictor, IET Signal Process. 1 (1) (2007) 29–34.

[25] Z. Malouche, O. Macchi, Adaptive unsupervised extraction of one component of a linear mixture with a single neuron, IEEE Trans. Neural Networks 9 (1998) 123–138.

[26] D.P. Mandic, A. Cichocki, An online algorithm for blind extraction of sources with different dynamical structures, in: 4th International Symposium on Independent Component Analysis and Blind Signal Separation, Nara, Japan, April 2003, pp. 645–650.

[27] K. Shah, E. Jonckheere, S. Bohacek, Dynamic modeling of internet traffic: linear versus nonlinear canonical correlation analysis of HTTP versus FTP traffic, Technical Report, Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA, December 2001.

[28] Z.W. Shi, C.S. Zhang, Semi-blind source extraction for fetal electrocardiogram extraction by combining non-Gaussianity and time-correlation, Neurocomputing 70 (2007) 1574–1587.

[29] A. Taleb, Source separation in structured nonlinear models, in: Conference on Acoustics, Speech, and Signal Processing, vol. 6, 2001, pp. 3516–3613.

[30] A. Taleb, C. Jutten, Source separation in post-nonlinear mixtures, IEEE Trans. Signal Process. 49 (10) (1999) 2807–2820.

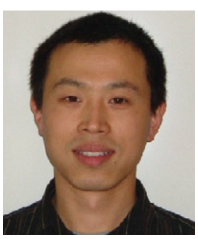
[31] D.L. Wang, M. Murphy, Estimating optimal transformations for multiple regression using the ACE algorithm, J. Data Sci. 2 (2004) 329–346.

[32] D.L. Wang, M. Murphy, Identifying nonlinear relationship in regression using the ACE algorithm, J. Appl. Statist. 32 (2005) 243–258.

[33] A. Ziehe, M. Kawanabe, S. Harmeling, K.R. Muller, Blind separation of post-nonlinear mixtures using linearizing transformations and temporal decorrelation, J. Mach. Learn. Res. 4 (2003) 1319–1338.

[34] W.Y. Leong, J. Homer, An implementation of nonlinear multiuser detection in Rayleigh fading channel, EURASIP J. Wireless Commun. Network. 2006 (2006) 9 (Article ID 45647).

Wai Yie Leong received the Bachelor degree in Electrical Engineering and the Ph.D. degree in Electrical Engineering from The University of Queensland, Australia in 2002 and 2006, respectively. In 1999, she was a System Engineer at the Liong Brothers Poultry Farm. From 2002 to 2005, she was appointed as Research Assistant and Teaching Fellow of the School of Information Technology and Electrical Engineering at The University of Queensland, Australia. She is also the Teaching Fellow of St. John's College, Australia. In 2005, she joined the School of Electronics and Electrical Engineering, Imperial College London, UK as a Post-doctoral Research Fellow. Between his B.Eng. and Ph.D. studies, she has been actively involving in research commercialisation. She is now the Research Engineer of A*STAR Corporate, Singapore. Her research interests include blind source separation, blind extraction, smart sensor, wireless communication systems, smart antennas and biomedical engineering.



Wei Liu was born in Hebei, China, in January 1974. He received the B.Sc. degree in Space Physics (minor in Electronics) in 1996, L.L.B. degree in Intellectual Property Law in 1997, both from Peking University, China, M.Phil. degree from the Department of Electrical and Electronic Engineering, University of Hong Kong, in 2001, and Ph.D. degree in 2003 from the Communications Research Group, School of Electronics and Computer Science, University of Southampton, UK. He then worked as a Postdoc in the same group and later in the Communications and Signal Processing Group, Department of Electrical and Electronic Engineering, Imperial College London. Since September 2005, he has been with the Communications Research Group, Department of Electronic and Electrical Engineering, University of Sheffield, UK, as a Lecturer. His research interests are mainly in array signal processing, blind source separation/extraction and multirate systems. He has now authored and co-authored more than 40 journal and conference publications.

Danilo P. Mandic is a Reader in Signal Processing at Imperial College London. He has been working in the area of nonlinear adaptive signal processing and nonlinear dynamics. His publication record includes two research monographs (Recurrent Neural Networks for Prediction, and Complex Valued Nonlinear Adaptive Filters) with Wiley, an edited book on Signal Processing for Information Fusion (Springer 2007) and more than 200 publications in Signal and Image Processing. He has been a Member of the IEEE Technical Committee on Machine Learning for Signal Processing, Associate Editor for the IEEE Transactions on Circuits and Systems II, IEEE Transactions on Signal Processing and International Journal of Mathematical Modelling and Algorithms. Dr. Mandic has produced award winning papers and products resulting from his collaboration with Industry. He is a Senior Member of the IEEE and Member of the London Mathematical Society. Dr. Mandic has been a Guest Professor in KU Leuven Belgium, TUAT Tokyo, Japan and Westminster University UK, and Frontier Researcher in RIKEN Japan.