Adaptive Blind Signal Processing—Neural Network Approaches

SHUN-ICHI AMARI, FELLOW, IEEE, AND ANDRZEJ CICHOCKI, MEMBER, IEEE

Invited Paper

Learning algorithms and underlying basic mathematical ideas are presented for the problem of adaptive blind signal processing, especially instantaneous blind separation and multichannel blind deconvolution/equalization of independent source signals. We discuss recent developments of adaptive learning algorithms based on the natural gradient approach and their properties concerning convergence, stability, and efficiency. Several promising schemas are proposed and reviewed in the paper. Emphasis is given to neural networks or adaptive filtering models and associated on-line adaptive nonlinear learning algorithms. Computer simulations illustrate the performances of the developed algorithms. Some results presented in this paper are new and are being published for the first time.

Keywords— *Blind deconvolution and equalization, blind separation of signals, independent component analysis (ICA), natural gradient learning, neural networks, self-adaptive learning rates, unsupervised adaptive learning algorithms.*

I. Introduction

There are many potential applications of neural networks to signal processing. This paper treats learning algorithms for blind signal separation and deconvolution of signals. Consider the case in which a number of simultaneous observations of signals are available that are linear or nonlinear superpositions of separate independent signals from different sources. In many cases, source signals are simultaneously filtered and mutually mixed. It is desired to process the observations such that the original source signals are extracted by neural networks (see, e.g., [1]–[18] and [32]).

This problem is known as independent component analysis (ICA), blind source separation (BSS), blind source deconvolution, or blind estimation of multiple independent sources [61], [68], [100], [101]. It can be formulated as the problem of separating or estimating the waveforms of the original sources from an array of sensors or transducers without knowing the characteristics of the transmission channels. For some models, however, there is no guarantee that the estimated or extracted signals can be of exactly the same waveforms as the source signals, and so the requirements are sometimes relaxed to the extent that the extracted waveforms are distorted or filtered versions of the source signals [24], [107].

There are many potential exciting applications of blind signal processing in science and technology, especially in wireless communication, noninvasive medical diagnosis, geophysical exploration, and image enhancement and recognition (such as face recognition) [37], [74], [75], [105]. Acoustic examples include the signals from several microphones in a sound field that is produced by several speakers (the so-called “cocktail-party” problem) or the signals from several acoustic transducers in an underwater sound field from the engine noises of several ships (the sonar problem). Radio examples include the observations corresponding to the outputs of several antenna elements in response to several transmitters; the observations may also include the effects of the mutual couplings of the elements. Other radio communication examples arise in the use of polarization multiplexing in microwave links because the orthogonality of the polarization cannot be maintained perfectly and there is interference among the separate transmissions. In wireless communication the problem arises of recovering communication signals (that are transmitted by unknown channels) from the received signals in the presence of both interuser and intersymbol interferences [19], [23], [61], [88], [106]. Other promising applications are in the area of noninvasive medical diagnosis and biomedical signal analysis, such as EEG, MEG, and ECG [78].

A number of heuristic algorithms for blind signal processing have been proposed as this field of research has been activated [68], [69]. Some recently developed algorithms are surprisingly good [6], [10], [29]–[38]. The heuris-

Manuscript received May 22, 1997; revised March 6, 1998.

S.-I. Amari is with the Brain-Style Information Systems Group, Riken Brain Science Institute, Wako-shi, Saitama 351-0198, Japan (e-mail: amari@brain.riken.go.jp).

A. Cichocki is with the Brain-Style Information Systems Group, Riken Brain Science Institute, Wako-shi, Saitama 351-0198, Japan, on leave from the Warsaw University of Technology, Poland (e-mail: cia@brain.riken.go.jp).

Publisher Item Identifier S 0018-9219(98)06970-9.

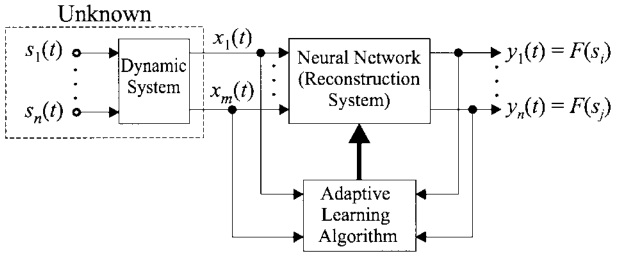


Fig. 1. Functional block diagram illustrating a general blind signal processing problem.

tic ideas have originated not only from traditional signal processing but from various backgrounds such as neural networks, information theory, statistics, systems theory, and information geometry. Now it is time to give unified theoretical foundations.

This paper proposes a unified theoretical basis for this problem based on these results. The theory is constructed on neural networks learning, semiparametric statistical inference, information theory, Lie group theory, and information geometry. We propose the natural gradient learning algorithm instead of the ordinary gradient descent method. In particular, we emphasize two important issues: statistical efficiency and the speed of convergence of algorithms. There are many other important problems, such as the robustness of algorithms and their algebraic properties, which we cannot touch upon here. They are, for example, Liu's approach and Loubuton's approach, where the number of observations is larger than the number of sources in the blind deconvolution/equalization problem [26], [59], [76].

II. The Problem Statements and Associated Neural Network Models

The most general blind signal processing (BSP) problem can be formulated as follows. We observe a set of signals from an MIMO (multiple-input multiple-output) nonlinear dynamic system (see Fig. 1), where its input signals are generated from a number of independent sources. The objective is to find an inverse neural system (also termed a reconstruction system), to see if it exists and is stable and to estimate the original input signals thereby. This estimation is performed on the basis of only the observed output signals where some *a priori* knowledge about the system and the stochastic properties of the source signals might be available. Preferably it is required that the inverse system is constructed adaptively so that it has good tracking capability under nonstationary environments. Alternatively, instead of estimating the source signals directly, it sometimes is more convenient to identify the unknown system (in particular when the inverse system does not exist) and then estimate the source signals implicitly by applying a suitable optimization procedure [62], [102].

A dynamic nonlinear system can be described in many different ways. It is sometimes convenient to describe it either as a lumped system or as a distributed system. This means that the system can be described by a set of ordinary differential (or difference) equations or partial differential equations. Alternatively, one may use linear filters described in the frequency domain. It should be emphasized that such a very general problem is in general intractable, and hence some *a priori* knowledge or assumptions about the system and the source signals are usually necessary.

Four fundamental problems arise:

1) solvability of the problem (in practical terms, this implies the existence of the inverse system and/or identifiability of the system) [20], [100], [101];
2) stability of the inverse model [13], [84], [105];
3) convergence of the learning algorithm and its speed with the related problem of how to avoid being trapped in local minima;
4) accuracy of the reconstructed source signals [8], [14].

Although recently many algorithms have been developed which are able to successfully separate source signals, there are still many problems to be studied.

1) Development of learning algorithms which work:
    a) under nonstationary environments;
    b) when the number of the source signals is unknown;
    c) when the number of source signals are dynamically changing,

    where the properties of the nonstationarities are not known in advance.
2) Optimal choices of the nonlinear activation functions when the distributions of sources are unknown and the mixtures contain sub-Gaussian and super-Gaussian sources.
3) Influence of additive noises and methods for its cancellation or reduction.
4) Global stability and convergence analysis of learning algorithms.
5) Statistical efficiency of learning algorithms.
6) Optimal strategy for deciding the learning rate parameter, especially in a nonstationary environment.

In this paper we will show where the difficulties lie, and we briefly indicate how to solve, at least partially, the corresponding problems. Our main objective in this paper is to develop a theoretical framework for solving these problems and to present associated adaptive on-line learning algorithms.

One direct approach to solve the problem is as follows.

1) Design suitable neural network models for the inverse (separating) problem.
2) Formulate an appropriate loss function (also termed a contrast, cost, or energy function) such that the global minimization or maximization of this function guarantees the correct separation or deconvolution. In particular, this guarantees the statistical independence of the output signals and/or spatial and temporal decorrelation of signals. The loss function should be

a function of the parameters of the neural network model.

3) Apply an optimization procedure to derive learning algorithms. There are many optimization techniques based on the stochastic gradient descent algorithm, such as the conjugate gradient algorithm, quasi-Newton method, and so on. We develop a new algorithm called the natural gradient algorithm [6], [7], [13], which automatically assures the desired equivariant property to be explained later.

A. Blind Separation of Instantaneous Linear Mixtures of Signals

There are many different mathematical or physical models in the mixing processes of unknown input sources $s_j(t)$ $(j = 1, 2, \cdots, n)$ depending on specific applications. In the beginning, we will focus on the simplest model where m observed signals $x_i(t)$ are instantaneous linear combinations of the n $(m \geq n)$ unknown source signals, which are assumed in this paper to be zero-mean and statistically independent. When the observable (sensors) signals are noise-contaminated, we have

$$x_i(t) = \sum_{j=1}^{n} h_{ij} s_j(t) + n_i(t), \qquad (i = 1, 2, \cdots, m) \quad (1)$$

or in the matrix notation [see Fig. 2(a) and (b)]

$$\boldsymbol{x}(t) = \boldsymbol{H}\boldsymbol{s}(t) + \boldsymbol{n}(t) \quad (2)$$

where $\boldsymbol{x}(t) = [x_1(t) \cdots x_m(t)]^T$ is the sensor vector at discrete time t, $\boldsymbol{s}(t) = [s_1(t) \cdots s_n(t)]^T$ is the source signal vector, $\boldsymbol{n}(t) = [n_1(t) \cdots n_m(t)]^T$ is the noise vector, and $\boldsymbol{H}$ is an unknown full rank $m \times n$ mixing matrix.

In order to recover the original source signals from the observed mixtures, we use a simple linear separating system (a feed-forward linear neural network)

$$\boldsymbol{y}(t) = \boldsymbol{W}(t)\boldsymbol{x}(t) \quad (3)$$

where $\boldsymbol{y}(t) = [y_1(t) \cdots y_n(t)]^T$ is an estimate of $\boldsymbol{s}(t)$ and $\boldsymbol{W}(t)$ is a $n \times m$ separating matrix, which is often called the synaptic weight matrix. A number of adaptive learning algorithms have been proposed by which the matrix $\boldsymbol{W}(t)$ is expected to converge to the separating matrix, as will be shown in Section III. However, the problem is ill-conditioned, and we cannot obtain $\boldsymbol{H}^{-1}$ even when $m = n$. There are two types of ambiguities. First, even if we can extract the n independent signals correctly from their m mixtures, we do not know their order of arrangements. That is, we do not know which signal is the first one. There is an inherent indeterminacy within the permutations of their ordering. Second, the scales of the extracted signals are unknown, because when a signal is multiplied by a scalar c it has the same effect as the multiplication of the corresponding column of $\boldsymbol{H}$ by the same constant. Hence, $\boldsymbol{W}(t)$ should converge at best to $\boldsymbol{W}_*$ that satisfies [100], [101]

$$\boldsymbol{T} = \boldsymbol{W}_*\boldsymbol{H} = \boldsymbol{P}\boldsymbol{D} \quad (4)$$

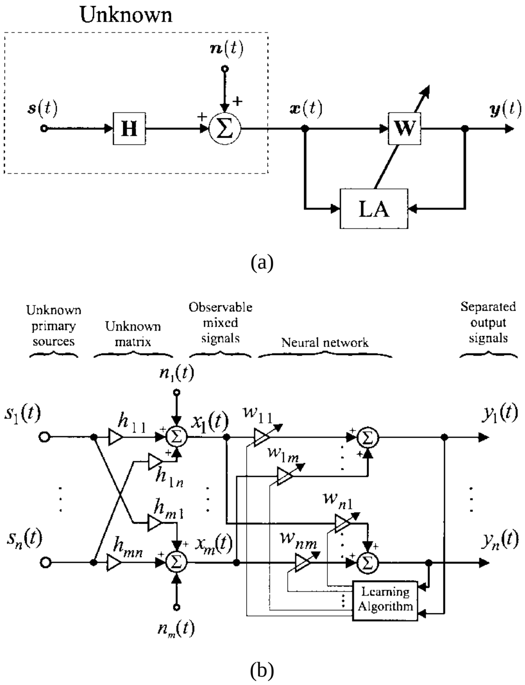


Fig. 2. Illustration of the mixing model and a basic feed-forward neural network for instantaneous blind separation of source signals: (a) general block diagram (LA means learning algorithm) and (b) detailed architecture of mixing and separating models.

where $\boldsymbol{D}$ is a scaling diagonal nonsingular matrix and $\boldsymbol{P}$ is any $n \times n$ permutation matrix.

The performance of the source separation is evaluated by the composite matrix $\boldsymbol{T}(t) = \boldsymbol{W}(t)\boldsymbol{H}$ which describes the total or global mixing-separating model such that $\boldsymbol{y}(t) = \boldsymbol{T}(t)\boldsymbol{s}(t)$. The separation is perfect when it tends to a generalized permutation matrix $\boldsymbol{P}\boldsymbol{D}$ which has exactly one nonzero element in each row and each column [100], [101]. This corresponds to the indeterminacies of the scaling and order of the estimated signals $y_i(t)$. This indeterminacy is not usually a serious problem since the most relevant information about the source signals is contained in the waveforms of the signals rather than their magnitudes and orders in which they are arranged.

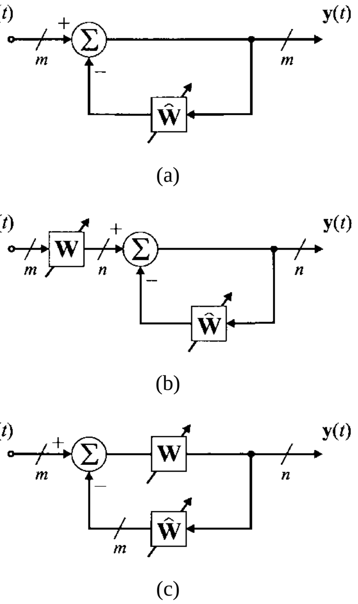
Instead of the feed-forward architecture, we may employ a fully connected feedback (recurrent) neural network shown in Figs. 3(a) and 4(b) in the case of $m = n$ [10], [35]. This is described by

$$y_i(t) = x_i(t) - \sum_{j=1}^{n} \widehat{w}_{ij}(t) y_j(t). \quad (5)$$

It should be noted that, for the ideal memoryless case, both basic models (recurrent and feed-forward) are equivalent under the condition that

$$\boldsymbol{W}(t) = \left[\boldsymbol{I} + \widehat{\boldsymbol{W}}(t)\right]^{-1}. \quad (6)$$

If, however, we take into account some intrinsic dynamic effects such as low-pass filtering, then the dynamic behaviors will be different since the recurrent networks depicted in Fig. 3 can be described by a system of differential (or



(a)

(b)

(c)

Fig. 3. Recurrent neural network models for blind signal separation: (a) simple recurrent neural network, (b) feed-forward–feedback model, and (c) hybrid model.

difference) equations [see Fig. 3(a)–(c), respectively] [10], [35]

$$\boldsymbol{\tau}\,\frac{d\boldsymbol{y}(t)}{dt} = -\,\boldsymbol{y}(t) + \boldsymbol{x}(t) - \widehat{\boldsymbol{W}}(t)\boldsymbol{y}(t) \qquad (7)$$

$$\boldsymbol{\tau}\,\frac{d\boldsymbol{y}(t)}{dt} = -\,\boldsymbol{y}(t) + \boldsymbol{W}(t)\boldsymbol{x}(t) - \widehat{\boldsymbol{W}}(t)\boldsymbol{y}(t) \qquad (8)$$

or

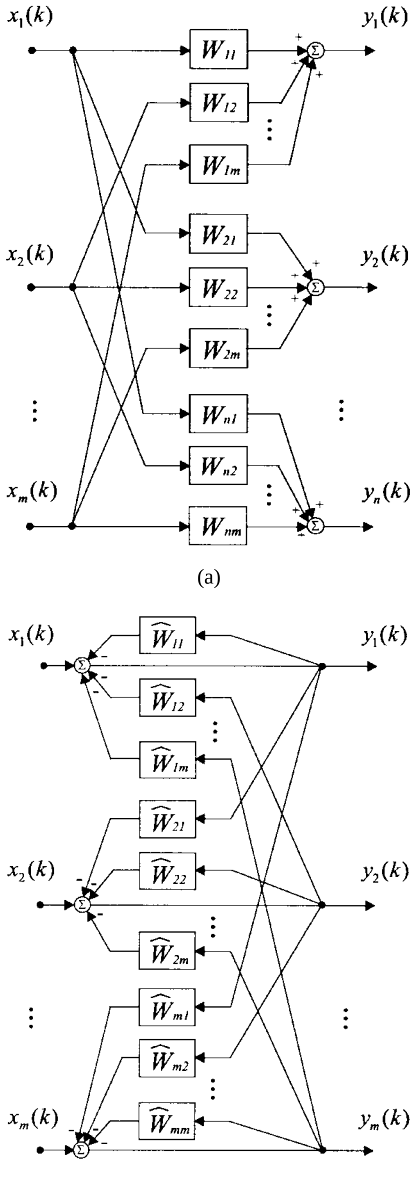
$$\boldsymbol{\tau}\,\frac{d\boldsymbol{y}(t)}{dt} = -\,\boldsymbol{y}(t) + \boldsymbol{W}(t)[\boldsymbol{x}(t) - \widehat{\boldsymbol{W}}(t)\boldsymbol{y}(t)] \qquad (9)$$

where $\boldsymbol{\tau}$ is a diagonal matrix representing the time constants of the network. It should be noted that for the feed-forward architecture of Fig. 2(a) and the hybrid architectures of Fig. 3(c) the number of outputs need not necessarily be equal to the number of sensors. In general, recurrent networks are more robust with respect to additive noises and fluctuations of the parameters [10], [33], [35], [43].

In this paper we treat mostly the case where the number of the outputs is equal to the number of the sensor signals. We also treat the important practical case where the number of the sources is not known in advance but it is greater than or equal to the number of sensors ($m \geq n$). Furthermore, we assume that all the signals are sampled at discrete-time instants $t_k = kT$, where $k = 0, 1, 2, \cdots$ and T is sampling period, assumed to be normalized to unity ($T = 1$).

B. Multichannel Blind Deconvolution/Equalization Problem

In this section we will formulate a more general and physically realistic model where the observed sensors signals are linear combinations of multiply time-delayed versions of the original source signals and/or mixed signals themselves, such as represented by MA, AR and ARMA models [11], [12], [29], [51], [61], [64]. All the coefficients are unknown.



(a)

(b)

Fig. 4. Detailed architectures for multichannel blind deconvolution: (a) feed-forward neural network and (b) recurrent neural network.

In order to recover the source signals, we can also use the network models depicted in Fig. 4(a) and (b) but the synaptic weights w_{ij} or $\widehat{w}_{ij}$ should be generalized to real or complex-valued dynamic filters [e.g., finite impulse response (FIR) or infinite impulse response (IIR) filters] as indicated in Fig. 5. A further variation is to consider the constrained IIR such as the recently introduced gamma or Laguerre filters or other structures which may have some useful properties [see Fig. 5(b) and (c)] [92].

The problem is referred to as that of multichannel blind deconvolution/equalization. In multichannel blind deconvolution and equalization, an m-dimensional vector of the received discrete-time signals $\boldsymbol{x}(k) = [x_1(k) \cdots x_m(k)]^T$ at time k is assumed to be produced from an n-dimensional vector of source signals $\boldsymbol{s}(k') = [s_1(k') \cdots s_n(k')]^T$ for

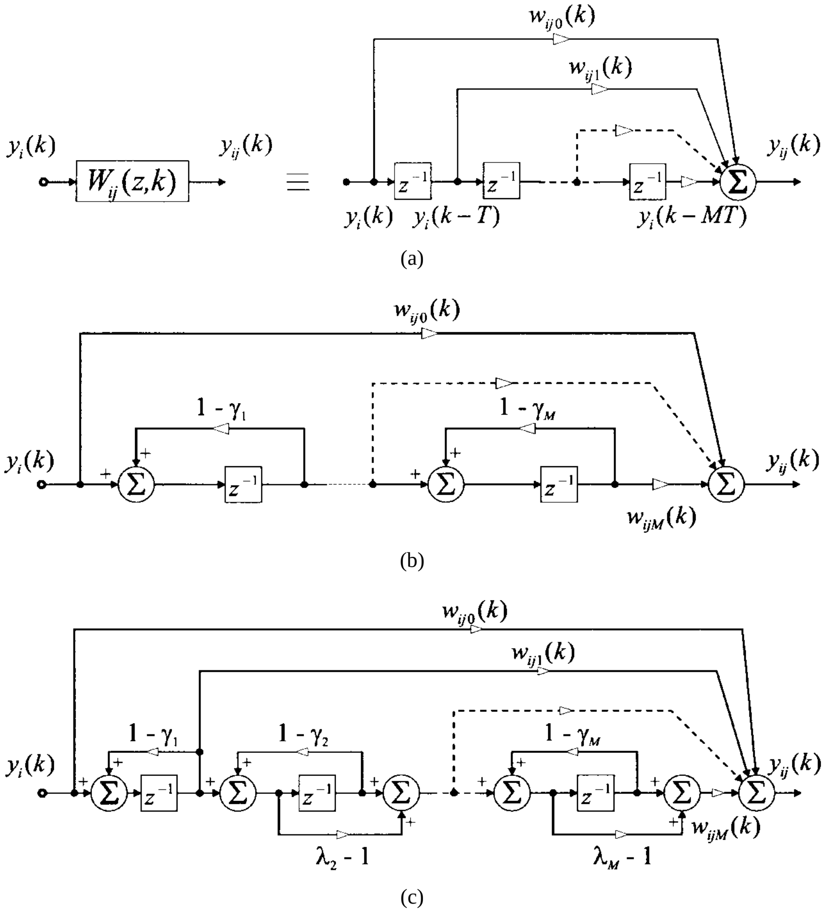


Fig. 5. Exemplary models of synaptic weights: (a) basic FIR filter model, (b) Gamma filter model, and (c) Laguerre filter model.

various times k', $m \geq n$, by using the mixture model

$$\boldsymbol{x}(k) = \sum_{p=-\infty}^{\infty} \boldsymbol{H}_p \boldsymbol{s}(k-p) \tag{10}$$

where $\boldsymbol{H}_p$ is an $(m \times n)$-dimensional matrix of mixing coefficients at lag p. In a causal system, p takes only $L+1$ nonnegative integers, $p = 0, 1, \cdots, L$, but it may take any integer values in the case of spatial filters (for example, for processing of images). The goal is to obtain rescaled and time-delayed estimates of the source signals from the received signals by using approximate knowledge of the distributions and statistics of the source signals. Typically, each source signal $s_i(k)$ is an independent and identically distributed (i.i.d.) sequence that is stochastically independent of all other source sequences [61], [65], [67].

Most methods proposed so far for multichannel blind deconvolution have focused on estimating the channel impulse responses $\{\boldsymbol{H}_p\}$ from the received signals $\boldsymbol{x}(k)$ and then determining the source signals from these estimates [59], [60], [62], [102], [109]. Moreover, the estimation of the channel impulse responses must be carried out before the equalized signals are calculated. In this paper, we consider alternative methods that estimate the source signals directly using a truncated version of a doubly-infinite multichannel equalizer of the form [11], [12], [29]

$$\boldsymbol{y}(k) = \sum_{p=-\infty}^{\infty} \boldsymbol{W}_p(k) \boldsymbol{x}(k-p) \tag{11}$$

where $\boldsymbol{y}(k) = [y_1(k) \cdots y_n(k)]^T$ is a n-dimensional vector of the output signals which are to be estimators of the source signals and $\{\boldsymbol{W}_p(k),\ -\infty \leq p \leq \infty\}$ is a sequence of $(n \times m)$-dimensional coefficient matrices. In the operator form, the input and output of the equalizer are [11], [12]

$$\boldsymbol{x}(k) = \boldsymbol{H}(z)[\boldsymbol{s}(k)] \tag{12}$$

$$\boldsymbol{y}(k) = \boldsymbol{W}(z, k)[\boldsymbol{x}(k)] = \boldsymbol{T}(z, k)[\boldsymbol{s}(k)] \tag{13}$$

where

$$\begin{aligned} \boldsymbol{W}(z, k) &= \sum_{p=-\infty}^{\infty} \boldsymbol{W}_p(k) z^{-p} \\ \boldsymbol{H}(z) &= \sum_{p=-\infty}^{\infty} \boldsymbol{H}_p z^{-p} \\ \boldsymbol{T}(z, k) &= \boldsymbol{W}(z, k) \boldsymbol{H}(z) \end{aligned} \tag{14}$$

are the z-transforms of the channel, equalizer, and total or overall channel-plus-equalizer impulse responses, respectively. Here z^{-1} is the delay operator, such that $z^{-p}[s_i(k)] = s_i(k-p)$. Then the goal of the adaptive deconvolution or equalization task is to adjust $\boldsymbol{W}(z, k)$ such that

$$\lim_{k\to\infty} \boldsymbol{T}(z, k) = \boldsymbol{P}\boldsymbol{D}(z) \tag{15}$$

where $\boldsymbol{P}$ is an $(n \times n)$-dimensional permutation matrix with a single unity entry in any of its rows or columns, $\boldsymbol{D}(z)$ is a diagonal matrix whose (i, i)th entry is $c_i z^{-\Delta_i}$, c_i is a nonzero scaling factor, and Δ_i is an integer delay value. This channel equalization methodology is the multichannel equivalent of the traditional Bussgang blind equalization schemes [61], [88], [103]. A significant shortcoming of the Bussgang techniques in the multichannel case is their relatively slow convergence speed [48], [61], [103], and thus few reports of algorithms have appeared thus far in the literature which are successfully applied to a multichannel equalizer of the form (11) [74], [88], [104], [106], [113]. When $\boldsymbol{H}_p = 0$, except for $p = 0$, the task is reduced to the simpler multichannel source separation of instantaneous signal mixtures.

In this paper, we first discuss general robust algorithms for blind source separation of instantaneous signal mixtures and next extend them to the task of joint signal separation and deconvolution [11], [12]. The natural gradient method is introduced and is proved to have the equivariant property (i.e., uniform properties of the algorithms are independent of the mixing coefficients and scaling factors of source signals [22]). In particular, we extend the natural gradient method described in [6], [7], and [13] in order to apply it to the blind deconvolution algorithms [11], [12]. We also give a class of general divergence measures in the space of probability distributions. This includes most of the loss functions proposed thus far, such as the maximum entropy problem formulation, ICA formulation, maximum likelihood formulation, cumulant methods, etc. The method is also elucidated from the more general framework of estimating functions in semiparametric statistical models [8], [15], [16].

C. Independent Components and Principal Components Analyses

It is useful to remark here that the ICA is different from the standard principal component analysis (PCA) [28], [54], [85], [86], [70], [71], [73]. When the covariance matrix of independent sources is

$$\boldsymbol{R}_{ss} = E\{\boldsymbol{s}(t)\boldsymbol{s}^T(t)\} = \boldsymbol{D} \tag{16}$$

where $\boldsymbol{D}$ is a positive diagonal matrix with diagonal entries d_{ii}, then the covariance matrix of $\boldsymbol{x}(t)$ is

$$\boldsymbol{R}_{xx} = E\big[\boldsymbol{x}(t)\boldsymbol{x}^T(t)\big] = \boldsymbol{H}\boldsymbol{D}\boldsymbol{H}^T. \tag{17}$$

The PCA searches for the orthogonal matrix $\boldsymbol{Q}$ such that the components $y_i(t)$ of

$$\boldsymbol{y}(t) = \boldsymbol{Q}\boldsymbol{x}(t) \tag{18}$$

are uncorrelated, that is

$$\boldsymbol{R}_{yy} = E\big[\boldsymbol{y}(t)\boldsymbol{y}^T(t)\big] = \boldsymbol{Q}\boldsymbol{R}_{xx}\boldsymbol{Q}^T = \boldsymbol{\Lambda}_1 \tag{19}$$

becomes a diagonal matrix $\boldsymbol{\Lambda}_1$ representing eigenvalues of the covariance matrix $\boldsymbol{R}_{xx}$.

However, even though y_i's become uncorrelated

$$E[y_i(t)y_j(t)] = 0, \qquad i \neq j \tag{20}$$

this does not mean that $y_i(t)$ and $y_j(t)$ are independent. To explain this, we consider the simple case where $\boldsymbol{D}$ is the identity matrix, that is

$$d_{ii} = E\big[s_i^2(t)\big] = 1. \tag{21}$$

In this case, for any orthogonal matrix $\boldsymbol{U}$, the components of

$$\boldsymbol{x}(t) = \boldsymbol{U}\boldsymbol{s}(t) \tag{22}$$

are uncorrelated since

$$E\big[\boldsymbol{x}(t)\boldsymbol{x}^T(t)\big] = \boldsymbol{U}\boldsymbol{I}\boldsymbol{U}^T = \boldsymbol{I}. \tag{23}$$

But $x_i(t)$ are not independent except for the special cases where $\boldsymbol{U}$ is a permutation matrix or the case where all s_i are Gaussian random variables.

In the general case, let

$$\boldsymbol{W} = \boldsymbol{U}\sqrt{\boldsymbol{D}^{-1}}\boldsymbol{Q} \tag{24}$$

where $\boldsymbol{U}$ is any orthogonal matrix. Therefore, when $\boldsymbol{Q}$ is an orthogonal matrix such that

$$\boldsymbol{R}_{yy} = \boldsymbol{I} \tag{25}$$

and $\boldsymbol{W}$ is another matrix by which $\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t)$ makes $y_i(t)$ uncorrelated. But there are no guarantee that $y_i(t)$ are independent.

The PCA can obtain independent components in only restricted special cases such as the original $s_i(t)$ are Gaussian, or matrix $\boldsymbol{H}$ is symmetrical and $d_{ii} = 1$ for all i. But we cannot, in general, recover the original $s_i(t)$ by the method of PCA. The present subject searches for the independent components in a more general case by using higher order cumulants and is completely different from the PCA.

It is sometimes useful to use the preprocessing of $\boldsymbol{x}(t)$ into $\tilde{\boldsymbol{x}}(t)$ by

$$\tilde{\boldsymbol{x}}(t) = \tilde{\boldsymbol{Q}}\boldsymbol{x}(t) = \Lambda^{-1/2}\boldsymbol{Q}^T\boldsymbol{x}(t) \tag{26}$$

such that

$$E\{\tilde{\boldsymbol{x}}(t)\tilde{\boldsymbol{x}}^T(t)\} = \boldsymbol{I}. \tag{27}$$

This can be easily performed by using the method of PCA [28], [54], [70], [71], [73], [85], [86]. This is called the prewhitening. We then search for the orthogonal matrix $\boldsymbol{W}$ where

$$\boldsymbol{y}(t) = \boldsymbol{W}\tilde{\boldsymbol{x}}(t) \tag{28}$$

such that $\boldsymbol{W}$ extracts the independent signals. This two-stage process is sometimes very useful. However, we can combine the two processes into one process. Therefore, we do not discuss the prewhitening process in the present paper, although it is important [44], [71], [72], [81], [85].

III. Learning Algorithms for Instantaneous Blind Separation

Stochastic independence of random variables is a more general concept than decorrelation. Therefore, the standard PCA cannot be used for our purpose, although nonlinear PCA has been successfully extended to the ICA [54], [56], [85], [86].

Intuitively speaking, random variables s_i and s_j are stochastically independent if knowledge of the values of s_j tells us nothing about the values of s_j. Mathematically, the independence can be expressed by the relationship $q(s_i, s_j) = q_i(s_i)q_j(s_j)$, where $q(s)$ denotes the probability density function (pdf) of random variable s. More generally, a set of signals $\boldsymbol{s}$ are independent if their joint pdf can be decomposed as

$$q(\boldsymbol{s}) = \prod_{j=1}^{m} q_j(s_j) \tag{29}$$

where $q_j(s_j)$ is the pdf of jth source signal.

In this section, for simplicity of consideration, we assume that all variables are real valued, that the number of source signals is equal to the number of sensors, and that the source signals are zero mean ($E[s_i(t)] = 0$), although some of these assumptions are easily relaxed. Moreover, we assume that, at most, one source distribution is Gaussian [44], [100], [101]. We also assume that additive noises have already been cancelled or reduced in the preprocessing stage to a negligible level [36]. Most learning algorithms have been derived from heuristic considerations based on minimization or maximization of a loss or performance function [6], [10], [30], [44], [68]. It is remarkable that similar types of learning algorithms are derived in common from these heuristic ideas, for example, cumulant minimization, entropy maximization (infomax) [17], [83], ICA [6], [44], and maximization of likelihood (ML) [21], [90], [110], [111].

We show that all of them can be derived from a unified principle as follows. A more general statistical consideration will be given later based on the concept of estimating functions. Let $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$ be estimated source signals and let $p_y(\boldsymbol{y})$ be its pdf which depends on $\boldsymbol{W}$. We search for such $\boldsymbol{W}$ that makes $\boldsymbol{y}$ independent signals. In order to measure the degree of independence, we use an adequately chosen independent probability distribution $q(\boldsymbol{y}) = \prod_i q_i(y_i)$ and consider the Kullback–Leibler (KL) divergence between two probability distributions $p_y(\boldsymbol{y})$ and $q(\boldsymbol{y})$

$$R(\boldsymbol{W}) = \mathrm{KL}[p_y(\boldsymbol{y})||q(\boldsymbol{y})] = \int p_y(\boldsymbol{y}) \log \frac{p_y(\boldsymbol{y})}{q(\boldsymbol{y})}\, d\boldsymbol{y}. \tag{30}$$

This is the risk function in the sense of Bayesian statistics.

The KL divergence has several important properties [5], [22].

1) $\mathrm{KL}(p||q) \geq 0$ with equality if and only if $p = q$.
2) The KL divergence is invariant under an invertible nonlinear transformation f of data samples

$$\mathrm{KL}(p(\boldsymbol{y})||q(\boldsymbol{s})) = \mathrm{KL}(p(\boldsymbol{f}(\boldsymbol{y}))||q(\boldsymbol{f}(\boldsymbol{s}))).$$

iii) The generalized Pythagorean theorem

$$\mathrm{KL}(p||q) = \mathrm{KL}(p||r) + \mathrm{KL}(r||q)$$

holds when three points p, q, and r form a "generalized right triangle" under certain conditions not stated here. This property is fundamental [5], [21], although we do not refer to it explicitly. From this, we can prove that $R(\boldsymbol{W})$ takes critical values at $\boldsymbol{WH} = \boldsymbol{PD}$, (the desired solution where the signals are separated) and it achieves the global minimum under a certain mild conditions on $q(\boldsymbol{y})$ to be stated later.

We show that various algorithms are derived from (30). Entropy maximization (infomax) uses component-wise nonlinear transformation functions $z_i = g_i(y_i)$ and tries to maximize the joint entropy of $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{y})$ [17]. It can be shown that the criterion is equivalent to the minimization of the risk function $R(\boldsymbol{W})$ by putting $q_i(y_i) = dg_i(y_i)/dy_i$. Moreover, when $q_i(y_i)$ is the true pdf of the source signal s_i, minimizing the $R(\boldsymbol{W})$ is equivalent to maximization of the likelihood function. Hence, the derived $\boldsymbol{W}$ is the maximum likelihood estimator. When $q_i(y_i)$ are set equal to the marginal distributions of $p_y(\boldsymbol{y})$ of $\boldsymbol{y}$, we have the ICA criterion, where $q_i(y_i)$ also depends on $\boldsymbol{W}$. The relation between different criteria of the infomax, mutual information, and ML are studied by Yang and Amari [111]. It is illuminating to study this from the viewpoint of information geometry on the manifold of probability distributions [5].

It can be shown that $\boldsymbol{R}(\boldsymbol{W})$ is the expectation of the following instantaneous loss function:

$$l(\boldsymbol{y}, \boldsymbol{W}) = -\tfrac{1}{2}\log\left[\det(\boldsymbol{W}\boldsymbol{W}^T)\right] - \sum_{i=1}^{m} \log q_i(y_i) \tag{31}$$

where $\det \boldsymbol{W}$ is the determinant of matrix $\boldsymbol{W}$, except for a constant term not depending on $\boldsymbol{W}$.

In order to derive a learning algorithm, we apply the standard stochastic gradient descent method [17]

$$\begin{aligned}\boldsymbol{W}(k+1) &= \boldsymbol{W}(k) - \eta(k)\,\frac{\partial l(\boldsymbol{W}(k), \boldsymbol{y}(k))}{\partial \boldsymbol{W}}\\ &= \boldsymbol{W}(k) + \eta(k)\Big[\left[\boldsymbol{W}^T\right]^{-1} - \boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{x}^T(k)\Big]\end{aligned} \tag{32}$$

where $\eta(k)$ is a learning rate at time k and $\boldsymbol{f}(\boldsymbol{y}) = [f_1(y_1) \cdots f_n(y_n)]^T$ is derived from $q_i(y_i)$ as

$$f_i(y_i) = -\frac{d\log(q_i(y_i))}{dy_i} = -\frac{q_i'}{q_i}. \tag{33}$$

How to choose $\boldsymbol{f}$ or q_i will be discussed later.

It is well known that stochastic gradient optimization methods for parameterized systems sometimes suffer from slow convergence. While the quasi-Newton method can be used to overcome these performance limitations [77], it is often costly in computation and may suffer from numerical instability if not implemented properly. Thus, it is desirable to search for new optimization methods that retain the simplicity and robustness of the stochastic gradient method while obtaining good asymptotic convergence performances. It is also desirable that the performance does not depend on the specific mixing system $\boldsymbol{H}$ so that the algorithm works uniformly well even when $\boldsymbol{H}$ is close to a singular matrix. This motivates us to discuss a more general or universal type of algorithm.

The natural gradient [6], [7] search method has emerged as a particularly useful technique when the parameter space is not a uniform Euclidean space but has some structure. In many cases, it has a Riemannian metric structure which includes the Euclidean space as a special case. In the present problem, the parameter space is the set of all nonsingular matrices $\boldsymbol{W}$. Since the product of two nonsingular matrices is again a nonsingular matrix, they form a group (Lie group). For a matrix $\boldsymbol{W}$, if we multiply $\boldsymbol{W}^{-1}$ from the right, it is mapped to the identity matrix $\boldsymbol{I}$. For a small deviation from $\boldsymbol{W}$ to $\boldsymbol{W} + d\boldsymbol{W}$, this multiplication maps $d\boldsymbol{W}$ to

$$d\boldsymbol{X} = d\boldsymbol{W}\boldsymbol{W}^{-1} \tag{34}$$

which is regarded as the corresponding increment measured at $\boldsymbol{I}$. We do not intend to describe mathematical details, but it is worth mentioning that a Riemannian metric structure is automatically introduced in the space of matrices by this group structure. When the parameter space is related to a family of probability distributions, information geometry [5] elucidated its Riemannian structure by using the Fisher information matrix. Information geometry plays a major role also in the manifolds of neural networks (see, e.g., [7]), but we do not discuss its details here.

The ordinary gradient $\partial l/\partial \boldsymbol{W}$ represents the steepest direction of function l when $\boldsymbol{W}$ is the Cartesian coordinate system of a Euclidean space. However, when the parameter space is not Euclidean or when a curvilinear coordinate system is taken in a Euclidean space, this is no more true. We need to modify $\partial l/\partial \boldsymbol{W}$ depending on the local structure of the space. The descent direction of search in such a case is represented by

$$\boldsymbol{G}^{-1}(\boldsymbol{W})\,\frac{\partial l(\boldsymbol{W}, \boldsymbol{y})}{\partial \boldsymbol{W}} \tag{35}$$

where $\boldsymbol{G}(\boldsymbol{W})$ is the Riemannian metric and $\boldsymbol{G}^{-1}(\boldsymbol{W})$ is its inverse. Since we do not intend to enter into Riemannian geometry, the reader may think that they are linear operators depending on $\boldsymbol{W}$, often represented by a matrix multiplication. Therefore, the gradient descent learning algorithm becomes

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) - \eta(k)\boldsymbol{G}^{-1}[\boldsymbol{W}(k)]\,\frac{\partial l\{\boldsymbol{W}(k), \boldsymbol{y}(k)\}}{\partial \boldsymbol{W}}. \tag{36}$$

This is called the natural gradient learning method. How good is the natural gradient method? In the case of neural learning it was proved that natural gradient on-line learning gives asymptotically the best local performance, which is equivalent to the best batch processing. Its behavior is equivalent to the Newton method when the parameter $\boldsymbol{W}$ approaches the equilibrium when the loss function is given by the logarithm of probability. However, the natural gradient is defined everywhere in the parameter space and is generally different from the Newton method. It can be applied to many different types of problems such as statistical estimation, multilayer perceptrons, Boltzmann machines, statistical estimation, and blind source separation and deconvolution (see [7] for details).

Note that the form of the Riemannian metric depends on the current parameter values and may require *a priori* knowledge of the statistical characteristics of the signals to be processed. Moreover, computing the natural gradient direction involves the inversion of $\boldsymbol{G}$ and is generally not easy. However, in the parameter space of matrices (and also of linear dynamic systems to be treated later) where instantaneous blind source separation are performed, the Riemannian metric (see [4] for the geometry of linear systems) is explicitly given by the Lie group structure. Moreover, its inverse is also calculated explicitly in a simple form, and the natural gradient learning algorithm can be formulated for this problem as

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) - \eta(k)\,\frac{\partial l(\boldsymbol{W}(k), \boldsymbol{y}(k))}{\partial \boldsymbol{W}}\,\boldsymbol{W}^T\boldsymbol{W}. \tag{37}$$

Thus, implementation of the natural gradient is particularly simple for the blind separation problem. Since the gradient $\partial l/\partial \boldsymbol{W}$ is given in (32), this leads to the simple robust adaptive learning rule [10], [30]

$$\boldsymbol{W}(k{+}1) = \boldsymbol{W}(k){+}\eta(k)\big[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{y}^T(k)\big]\boldsymbol{W}(k). \tag{38}$$

See Appendix A for its derivation. We can rewrite the algorithm in terms of the composite matrix $\boldsymbol{T}(k) = \boldsymbol{W}(k)\boldsymbol{H}$ (describing the overall mixing-separating process) as

$$\begin{aligned}\boldsymbol{T}(k+1) =& \boldsymbol{T}(k) + \eta(k)\\ &\cdot \big[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{T}(k)\boldsymbol{s}(k))[\boldsymbol{T}(k)\boldsymbol{s}(k)]^T\big]\boldsymbol{T}(k).\end{aligned} \tag{39}$$

This is independent of the mixing matrix $\boldsymbol{H}$, so that the learning behavior is directly described in terms of $\boldsymbol{T}$ and it does not depend on a specific $\boldsymbol{H}$. In other words, even when contributions of some sources are very small, there is no problem for recovering them. This desirable property is called the equivariant property since the asymptotic properties of the algorithm are independent of the mixing matrix and the scaling factors of source signals [22], [35].

Using simple relations $\widehat{\boldsymbol{W}} = \boldsymbol{W}^{-1} - \boldsymbol{I}$ and $d\boldsymbol{W}\boldsymbol{W}^{-1} = -\boldsymbol{W}d\boldsymbol{W}^{-1}$, we can derive a similar equivariant algorithm for the fully recurrent neural network shown in Figs. 3(a) and 4(b) as [10], [35], [38]

$$\widehat{\boldsymbol{W}}(k{+}1) = \widehat{\boldsymbol{W}}(k){-}\eta(k)\Big[\widehat{\boldsymbol{W}}(k) + \boldsymbol{I}\Big]\,\big[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{y}^T(k)\big]. \tag{40}$$

A disadvantage of the above algorithm is that in computer implementation it is required to invert the matrix $\boldsymbol{I} + \widehat{\boldsymbol{W}}$ in each iteration step in order to evaluate the output vector $\boldsymbol{y}(k) = [\boldsymbol{I} + \widehat{\boldsymbol{W}}(k)]^{-1}\boldsymbol{x}(k)$. However, the system computes the outputs without any explicit inverse of matrices in its physical realizations.

The final problem is how to choose the function $\boldsymbol{f}(\boldsymbol{y})$, or equivalently $q_i(y_i)$. If we know the true source pdf, the best choice is to use them, since it gives the maximum likelihood estimator which is Fisher-efficient. One idea is to estimate the true pdf's adaptively. However, the point is that when we misspecify $q_i(y_i)$, the algorithm gives the correct answer (or the consistent estimator in statisticians' terminology) under certain conditions [6], [8], [15]. We give the precise conditions for convergence later. It should be noted here that one can use any function $f_i(y_i)$ which might not be derived from a probability distribution $q_i(y_i)$. This is justified from the theory of estimating functions given later. We state here that suboptimal choices for these nonlinearities still allow the algorithm to perform separation: for sub-Gaussian source signals with negative kurtosis, we can select $f_i(y_i) = \alpha y_i + y_i|y_i|^2$ and for super-Gaussian source signals with positive kurtosis $f_i(y_i) = \alpha y_i + \tanh(\gamma y_i)$, where $\alpha \geq 0$ and $\gamma \geq 2$. In the situation where $\boldsymbol{x}(k)$ contains mixtures of both sub- and super-Gaussian sources, additional techniques are required to enable the system to adapt properly [41], [42], [52], [56]. We give a universally convergent algorithm in the next section.

IV. Equilibrium, Convergence, and Stability of Learning Algorithms

The learning algorithms in the previous section are derived using the stochastic gradient descent method. Therefore, the learning system is globally stable in the sense that, for a sufficiently small learning rate, the loss function decreases on average such that the parameters (synaptic weights) converge to one of the local minima and possibly fluctuate in its neighborhood. Such a system has rather simple convergence behaviors. There are no periodic oscillations and no chaotic behaviors [32] in the averaged learning equations

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) + \eta E\big[\boldsymbol{I} - \boldsymbol{f}\{\boldsymbol{y}(k)\}\boldsymbol{y}^T(k)\big]\boldsymbol{W}(k). \quad (41)$$

Every solution $\boldsymbol{W}(k)$ converges to an equilibrium, except for the measure zero case, which we do not discuss here.

The equilibria are determined by a set of nonlinear equations with respect to $\boldsymbol{W}$

$$E\{\boldsymbol{f}(\boldsymbol{y})\boldsymbol{y}^T\} = \boldsymbol{I} \quad (42)$$

or in the component form

$$E\{f_i(y_i)y_j\} = \delta_{ij} \quad (43)$$

where δ_{ij} is the Kronecker delta. This means that the amplitude of the output signals are automatically scaled in such way that they satisfy condition $E\{f_i(y_i)y_i\} = 1$. As it will be shown in Section V, these constraints can be relaxed or even completely removed.

The equilibrium condition (43) is satisfied when the output signals $\boldsymbol{y}$ becomes a permutation of rescaled source signals, because y_i are independent and zero mean. Hence, such $\boldsymbol{W}$ [which satisfies (4)] is an equilibrium. We call it the correct equilibrium.

However, this does not mean that the correct equilibrium is dynamically stable. It is important to study the stability, because it determines whether or not the system successfully separates the source signals. By the local variational analysis, it can be shown that the correct equilibrium point is stable when the learning rate $\eta(k)$ is sufficiently small and all the eigenvalues of the Hessian of the loss function (31)

$$\boldsymbol{K} = E\left\{\frac{\partial^2 l(\boldsymbol{y}, \boldsymbol{W})}{\partial \boldsymbol{W}\, \partial \boldsymbol{W}}\right\} \quad (44)$$

are positive. It is found that a necessary and sufficient condition for stability is expressed in a relatively simple form [7], [9], [13] as

$$m_i + 1 > 0, \quad (45)$$

$$\kappa_i > 0, \quad (46)$$

$$\sigma_i^2\sigma_j^2\kappa_i\kappa_j > 1, \qquad \text{for all } 1 \leq i \leq m \quad (47)$$

where $m_i = E\{y_i^2 f_i'(y_i)\}$, $\kappa_i = E\{f_i'(y_i)\}$, $\sigma_i^2 = E\{|y_i|^2\}$, y_i is the source signal extracted at the ith output, and $f_i'(y_i)$ denotes the derivative of the activation function $f_i(y_i)$ with respect to y_i (see Appendix B for its derivation). This determines how to choose $\boldsymbol{f}_i$ for obtaining the correct solution. In other words, the stability depends on $\boldsymbol{f}_i$ and the statistical properties of the sources as is expressed mathematically by (45)–(47).

It is interesting to note that, when f_i are monotonically increasing odd nonlinear functions, the above stability conditions are approximately satisfied when [52]

$$\sigma_i^2\kappa_i > \rho_i \qquad \text{holds for all } 1 \leq i \leq m \quad (48)$$

where $\rho_i = E\{y_i f_i(y_i)\}$.

It can be also easily shown that the above stability conditions are satisfied if one selects nonlinearities $f_i(y_i) = \alpha y_i + \tanh(\gamma y_i)$ for super-Gaussian signals and a cubic nonlinearity $f_i(y_i) = \alpha y_i + y_i^3$ for sub-Gaussian signals, (where $\alpha > 0$ and $\gamma > 2$), although such nonlinearities are not optimal in most cases [13], [22], [42], [57].

In the case when the sensor signals are mixtures of both sub- and super-Gaussian source signals there are some problems. Even with some knowledge of the source signal distributions, it is impossible to know *a priori* which source signal will be extracted at which output, and thus it is impossible to select the nonlinearities $f_i(y_i)$ *a priori* to guarantee that all of the signals will be extracted properly. In such a case, we can apply the following strategy proposed by Douglas *et al.* [52]. We prepare nonlinearity vectors $\boldsymbol{f}(\boldsymbol{y}(k)) = [f_{1r}(y_1(k)) \cdots f_{mr}(y_m(k))]^T$ in advance, where $f_{ir}(y)$, $r = 1, 2$ is chosen to be one of the two nonlinearities $f^{(1)}(y)$ and $f^{(2)}(y)$ that are optimized for sub- and super-Gaussian source separation tasks, respectively. To decide nonlinear activation function f_i for each channel, we form the time-averaged estimates

$$\sigma_i^2(k) = (1-\eta)\sigma_i^2(k-1) + \eta|y_i(k)|^2 \quad (49)$$

$$\kappa_{ir}(k) = (1-\eta)\kappa_{ir}(k-1) + \eta f'^{(r)}(y_i(k)) \quad (50)$$

$$\rho_{ir}(k) = (1-\eta)\rho_{ir}(k-1) + \eta y_i(k) f_i^{(r)}(y_i(k)) \quad (51)$$

for $r = \{1, 2\}$ and $1 \leq i \leq m$, where η is a small positive parameter. Then $f_{ir}(y)$ at discrete time k is selected as [52]

$$f_{ir}(y) = \begin{cases} f^{(1)}(y), & \text{if } \sigma_i^2(k)\kappa_{i1}(k) - \rho_{i1}(k) \\ & \quad > \sigma_i^2(k)\kappa_{i2}(k) - \rho_{i2}(k), \\ f^{(2)}(y), & \text{otherwise} \end{cases} \tag{52}$$

and is used to update the coefficient matrix in (38) and (40). It can be seen that, as the coefficients of the system converge, the quantity $\sigma_i^2(k)\kappa_{ir}(k) - \rho_{ir}(k)$ becomes a reliable estimate of the left-hand side of the inequality in (48) for $f_i(y) = f^{(r)}(y)$. Extensive simulations have shown that, so long as there exists a set of nonlinearity assignments such that the stability condition in (48) is satisfied for one ordering of the extracted sources at the outputs, then (52) properly selects the ith nonlinearity over time to enable the system to reliably extract all the source signals regardless of their distributions [52].

More sophisticated methods have been proposed recently to obtain a universally convergent learning algorithms without changing the equilibrium [13], [41], [42], [110]. For example, we use the inverse of the operator $\boldsymbol{K}(\boldsymbol{W})$. Because it has a special block diagonal form, its inverse is obtained explicitly. The universally convergent algorithm can be written in the form [9], [13]

$$\begin{aligned} \boldsymbol{W}(k+1) &= \boldsymbol{W}(k) + \eta(k)\boldsymbol{K}^{-1}\{\boldsymbol{W}(k)\}\big[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{y}^T(k)\big] \\ &\quad \cdot \boldsymbol{W}(k) \\ &= \boldsymbol{W}(k) + \eta(k)\boldsymbol{F}(\boldsymbol{y}(k))\boldsymbol{W}(k) \end{aligned} \tag{53}$$

where elements $f_{ij}(k)$ of the matrix $\boldsymbol{F}(\boldsymbol{y}(k))$ are determined as

$$f_{ij} = -\frac{1}{\sigma_i^2\sigma_j^2\kappa_i\kappa_j - 1}\left\{\sigma_j^2\kappa_i f_i(y_i)y_j - f_j(y_j)y_i\right\} \qquad i \neq j \tag{54}$$

$$f_{ii} = \frac{1}{m_i + 1}\left\{1 - f_i(y_i)y_i\right\}. \tag{55}$$

Here, we need to know σ_i^2, κ_i, and m_i approximately, or need to estimate them adaptively. The above learning algorithm ensures that the correct solution is always a stable equilibrium of the modified learning equation (except for the case $\sigma_i^2\sigma_j^2\kappa_i\kappa_j = 1$).

V. Statistical Properties of Adaptive Algorithms for Blind Source Separation

We have described a family of adaptive learning algorithms. Some are efficient in giving accurate estimators and some are convergent. The reader might wonder what is the general form of efficient learning algorithms. The present section considers this problem from the statistical point of view.

Let

$$p(\boldsymbol{s}) = \prod_{i=1}^{n} p_i(s_i) \tag{56}$$

be the true probability density function of the source signals. Then, the pdf of $\boldsymbol{x} = \boldsymbol{H}\boldsymbol{s}$ is written in terms of $\boldsymbol{W} = \boldsymbol{H}^{-1}$ as

$$p_X(\boldsymbol{x}; \boldsymbol{W}, p) = \det(\boldsymbol{W})p(\boldsymbol{W}\boldsymbol{x}). \tag{57}$$

Given a series of observations $\boldsymbol{x}(1), \cdots, \boldsymbol{x}(T)$, it is a statistical problem to estimate the true $\boldsymbol{W}$ (except for the indeterminacies mentioned before). However, this problem is "ill-posed" in the sense that the statistical model (57) includes not only the parameters $\boldsymbol{W}$, which we want to know, but also n unknown functions $p_i(s_i)$, $i = 1, \cdots, n$. A statistical model is said to be semiparametric when it includes extra unknown parameters of infinite dimensions. Since the unknown functions are of infinite dimensions, this brings some difficulties for estimating $\boldsymbol{W}$.

The estimating function method [15], [16] is applicable to such a problem. Let $\tilde{\boldsymbol{F}}(\boldsymbol{x}, \boldsymbol{W})$ or $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{W})$ be a $n \times n$ matrix function depending on $\boldsymbol{x}$ and $\boldsymbol{W}$ or equivalently on $\boldsymbol{y}$ and $\boldsymbol{W}$. When

$$E[\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{W})] = \boldsymbol{0} \tag{58}$$

is satisfied under whatever source distribution $p(\boldsymbol{s})$ of the form (58) is used for taking the expectation, $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{W})$ is called an estimating function matrix. To avoid trivial solutions we need some regularity conditions such as

$$E\big[\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{W}')\big] \neq \boldsymbol{0} \tag{59}$$

when $\boldsymbol{W}'$ is different from the true parameters $\boldsymbol{W}$ used for taking the expectation. When observations $\boldsymbol{x}(1), \cdots, \boldsymbol{x}(T)$ are given, we can approximate the expectation (58) by the arithmetic mean $(1/T)\sum_{k=1}^{T}\boldsymbol{F}\{\boldsymbol{y}(k), \boldsymbol{W}\}$. Therefore, when an estimating function exists, the solution of the estimating equation

$$\sum_{k=1}^{T}\boldsymbol{F}\{\boldsymbol{y}(k), \boldsymbol{W}\} = \boldsymbol{0} \tag{60}$$

is believed to give a good estimator. It is proved that the solution of (60) is a consistent estimator, that is, an estimator which converges to the true $\boldsymbol{W}$ as T tends to infinity.

We can find a variety of estimating functions. For example

$$\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{W}) = \boldsymbol{F}(\boldsymbol{y}) = \boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y})\boldsymbol{y}^T \tag{61}$$

is an estimating function because

$$E[\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{W})] = \boldsymbol{0} \tag{62}$$

holds when $\boldsymbol{W}$ is the separating solution. The above function is derived from a cost function as its gradient. However, there are estimating functions which are not written as gradients of cost functions. For example, when one of

$$E[\boldsymbol{f}(\boldsymbol{y})] = 0, \quad E[\boldsymbol{g}(\boldsymbol{y})] = \boldsymbol{0} \tag{63}$$

holds, then

$$\boldsymbol{F}(\boldsymbol{y}) = \boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y})\boldsymbol{g}(\boldsymbol{y})^T \tag{64}$$

is an estimating function.

A learning algorithm is easily obtained from an estimating function as

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+\eta(k)\boldsymbol{F}[\boldsymbol{y}(k)]\boldsymbol{W}(k) \tag{65}$$

because the true solution is the equilibrium of (60).

An important problem is to find such an estimating function which gives a good performance. An estimating function $\boldsymbol{F}$ is said to be inadmissible when there exists an estimating function $\tilde{\boldsymbol{F}}$ which gives a better estimator than $\boldsymbol{F}$ does for any probability distributions. We need to obtain the class of admissible estimating functions.

Information geometry [5] is particularly useful for analyzing this type of problem [7]–[9]. When it is applied to the present problem we can obtain all the set of estimating functions. It includes the Fisher efficient one, which is asymptotically the best one. However, the best choice of $\boldsymbol{F}$ again depends on the unknown p, thus we need to use an adaptive method. The following important results [6] are obtained by applying information geometry.

1) The off-diagonal components $f_{ij}(\boldsymbol{y}, \boldsymbol{W})$, $i \neq j$, of an admissible estimating function has the form

$$f_{ij}(\boldsymbol{y}, \boldsymbol{W})=\alpha f_i(y_i)y_j+\beta f_j(y_j)y_i \tag{66}$$

where α and β are suitably chosen constants or variable parameters.

2) The diagonal part $f_{ii}(\boldsymbol{y}, \boldsymbol{W})$ can be arbitrarily assigned.

The above theoretical results have the following implications. Since most learning algorithms have been derived heuristically, one might further try to obtain better rules by searching for an extended class of estimating functions such as $\boldsymbol{f}(\boldsymbol{y})\boldsymbol{g}(\boldsymbol{y})^T$ or more general ones. However, this is not admissible, and we can find a better function for noiseless case in the class of

$$\boldsymbol{F}(\boldsymbol{y})=\alpha \boldsymbol{f}(\boldsymbol{y})\boldsymbol{y}^T+\beta \boldsymbol{y}\boldsymbol{f}(\boldsymbol{y})^T. \tag{67}$$

We have studied the special case of $\beta=0$ in previous sections.

It should be noted that $\boldsymbol{F}(\boldsymbol{y})$ and $K(\boldsymbol{W})\boldsymbol{F}(\boldsymbol{y})$ give the same estimating equations, where $K(\boldsymbol{W})$ is a linear operator. Therefore, $\boldsymbol{F}$ and $\boldsymbol{KF}$ are equivalent when we estimate $\boldsymbol{W}$ by batch processing. However, two learning rules

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+\eta(k)\boldsymbol{F}(\boldsymbol{y}) \tag{68}$$

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+\eta(k)\boldsymbol{K}(\boldsymbol{W})\boldsymbol{F}(\boldsymbol{y}) \tag{69}$$

have different dynamical characteristics, although their equilibria are the same. The natural gradient uses $\boldsymbol{F}(\boldsymbol{y})\boldsymbol{W}$ to improve the performance and to keep the equivariant property. The universally convergent algorithm uses the inverse of the Hessian as $\boldsymbol{K}(\boldsymbol{W})$, so that the convergence is guaranteed. Refer to [8], [15], and [16] to learn how to choose good $\boldsymbol{F}(\boldsymbol{y})$ for theoretical discussions. The statistical efficiency of estimators is analyzed in detail in [8]. It is remarkable that "superefficiency" holds for some cases under certain conditions [9].

The second fact shows that Δw_{ii} can be set arbitrarily. Indeed, this term determines the scales of the output signals which can be arbitrarily assigned. We can rescale them if necessary. A simple choice is $\Delta w_{ii}=0$ or $w_{ii}=1$. This is a hard constraint on w_{ii} and very simple, but it loses the equivariant property. So we will not discuss such an approach in this paper. Another form of the constraints are soft constraints such that Δw_{ii} depends on $\boldsymbol{y}$. For example, we require that the amplitudes of the output signals satisfy

$$E\{\boldsymbol{f}(\boldsymbol{y})\boldsymbol{y}^T\}=\boldsymbol{\Lambda} \tag{70}$$

where $\boldsymbol{\Lambda}$ is any positive definite diagonal matrix. The learning rule (38) then takes the following generalized form:

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+\eta(k)\left[\boldsymbol{\Lambda}-\boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{y}^T(k)\right]\boldsymbol{W}(k). \tag{71}$$

A particularly convenient and interesting case is the special one where $\boldsymbol{\Lambda}=\operatorname{diag}\{f_1(y_1)y_1 \cdots f_n(y_n)y_n\}$. In such a case we remove any constraints (soft and hard) on the diagonal part in the bracket of (71). This new approach improves the convergence properties because, changes in $\boldsymbol{W}(k)$ are shown to be the orthogonal to the equivalence set of $\boldsymbol{W}$ [14].

Furthermore, we discovered that such a learning rule is able to separate the source signals even in the case that the number of sensors is greater than the number of sources and we do not know the exact number of sources. For such a problem we can apply the $m \times m$ matrix $\boldsymbol{W}$, where $m \geq n$ and a slightly modified on-line adaptive algorithm

$$\begin{aligned}\boldsymbol{W}(k+1)=&\,\boldsymbol{W}(k)+\eta(k)\\&\cdot\left[\boldsymbol{\Lambda}-\boldsymbol{y}(k)\boldsymbol{y}^T(k)-\boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{y}^T(k)\right]\boldsymbol{W}(k)\end{aligned} \tag{72}$$

where $\boldsymbol{\Lambda}=\operatorname{diag}\{[(y_1+f_1(y_1))y_1]\cdots[(y_m+f_m(y_m))y_m]\}$. In this new learning rule an auxiliary term $\boldsymbol{y}(k)\boldsymbol{y}^T(k)$ is added in order to simultaneously perform second-order moment decorrelation and higher-order independence of the output signals. Mutual decorrelations due to some redundancy make $(m-n)$ of the output signals converge quickly to zero. This has been confirmed by extensive computer simulations (see Section IX). To achieve the same task, we can alternatively use a symmetrical algorithm which is a generalization of Cardoso–Laheld algorithm [21]

$$\begin{aligned}\boldsymbol{W}(k+1)=&\,\boldsymbol{W}(k)+\eta(k)\left[\boldsymbol{\Lambda}-\boldsymbol{y}(k)\boldsymbol{y}^T(k)-\boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{y}^T(k)\right.\\&\left.+\boldsymbol{y}(k)\boldsymbol{f}(\boldsymbol{y}^T)\right]\boldsymbol{W}(k)\end{aligned} \tag{73}$$

where $\boldsymbol{\Lambda}=\operatorname{diag}\{y_1^2\cdots y_m^2\}$.

Summarizing, all the adaptive learning algorithms with the equivariant property for blind separation of sources can be written in the general form by using estimating functions

— For the feed-forward neural network model [see Fig. 2(a)]

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+\eta(k)\boldsymbol{F}\{(\boldsymbol{y}(k)\}\boldsymbol{W}(k). \tag{74}$$

— For the recurrent neural network model [see Fig. 3(a)]

$$\widehat{\boldsymbol{W}}(k+1) = \widehat{\boldsymbol{W}}(k) - \eta(k)\Big[\widehat{\boldsymbol{W}}(k) + \boldsymbol{I}\Big]\boldsymbol{F}[(\boldsymbol{y}(k)]) \quad (75)$$

where separating matrices $\boldsymbol{W}$ and $\widehat{\boldsymbol{W}}$ are in general of dimension $m \times m$ and the elements of an admissible $m \times m$ matrix $\boldsymbol{F}(\boldsymbol{y}(k))$ can take the generalized form

$$\begin{aligned} f_{ij}(k) =& \lambda_{ii}\delta_{ij} - \alpha_{1i}y_i(k)y_j(k) - \alpha_{2i} \\ &\cdot f[y_i(k)]y_j(k) + \alpha_{3i}y_i(k)f[y_j(k)] \end{aligned} \quad (76)$$

where λ_{ii} are suitably chosen entries of the scaling diagonal matrix $\boldsymbol{\Lambda}$ as explained and α_{ki} are nonnegative parameters in general depending on the statistical properties of the estimated source signals. For example, in the special case for $\alpha_{1i} = \alpha_{2i} = 1$ and $\alpha_{3i} = 0$, the generalized learning rule is simplified to (72). Similarly for $\alpha_{1i} = \alpha_{3i} = \alpha_{3i} = 1$, we obtain algorithm (73). Analogously, we can obtain the algorithm with the universal convergence property (53)–(55) and other algorithms discussed in this paper as special cases.

VI. Blind Signal Extraction

An alternative approach to blind signal separation is to extract the source signals sequentially one by one, rather than to separate all of them simultaneously. This procedure is called the blind signal extraction in contrast to BSS.

Several approaches have been recently developed for blind signal extraction and blind deconvolution [39], [63]–[65], [96]. A single processing unit (artificial neuron) is used in the first step to extract one source signal with specified stochastic properties. In the next step, a deflation technique is used in order to eliminate the already extracted signals from the mixtures. Another technique employs a (multistage) hierarchical neural network which ensures that the signals are extracted in a specified order without extracting the same source signals twice. This is achieved by using inhibitory synapses between the units.

Let us consider a single processing unit [see Fig. 6(a)] described by

$$y_1(k) = \boldsymbol{w}_1^T(k)\boldsymbol{x}_1(k) = \sum_{j=1}^{m} w_{1j}(k)x_{1j}(k) \quad (77)$$

where $\boldsymbol{x}_1 = \boldsymbol{x}$ or $\boldsymbol{x}_1 = \boldsymbol{Q}\boldsymbol{x}$ (where $\boldsymbol{Q}$ is prewhitening matrix). The prewhitening is optional preprocessing in order to improve convergence speed for ill-conditioned problems.

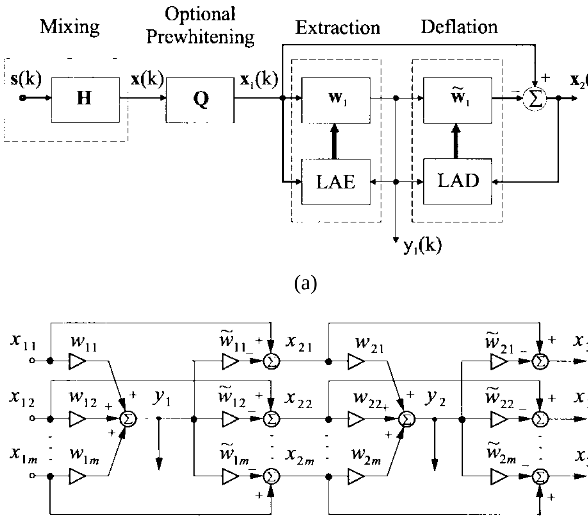
The unit successfully extracts a source signal, say the jth signal, if $\boldsymbol{w}_1(\infty) = \boldsymbol{w}_{1*}$ that satisfies the relation $\boldsymbol{w}_{1*}^T\boldsymbol{H} = \boldsymbol{e}_j^T$, where $\boldsymbol{e}_j$ denotes the jth column of a $n \times n$ nonsingular diagonal matrix.

A possible loss (contrast) function is [39]

$$l(\boldsymbol{w}_1) = -\tfrac{1}{4}|\kappa_4(y_1)| \quad (78)$$

where $\kappa_4(y_1)$ is the normalized kurtosis defined by

$$\kappa_4(y_1) = \frac{E[|y_1|^4]}{\{E[|y_1|^2]\}^2} - 3. \quad (79)$$



(a)

(b)

Fig. 6. Block diagram illustrating blind extraction and deflation of signals: (a) block diagram illustrating extraction and deflation process of the first extracted signal (LAE-learning algorithm for extraction; LAD-learning algorithm for deflation) and (b) cascade neural network for blind extraction of signals and independent component analysis.

It is easy to show that for the prewhitened sensor signals $\boldsymbol{x}_1$ the normalized kurtosis satisfies the following relations:

$$\begin{aligned} \kappa_4(y_1) &= \kappa_4(\boldsymbol{w}_1^T\boldsymbol{H}\boldsymbol{s}) = \kappa_4(\boldsymbol{e}^T\boldsymbol{s}) \\ &= \sum_{i=1}^{m} \frac{e_i^4}{||\boldsymbol{e}||^4}\kappa_4(s_i) = \sum_{i=1}^{m} \bar{e}_i^4\kappa_4(s_i) \end{aligned} \quad (80)$$

where $\boldsymbol{e} = \boldsymbol{w}_1^T\boldsymbol{H}$ and $||\bar{\boldsymbol{e}}|| = 1$.

Thus our normalized kurtosis satisfies the lemma given by Delfosse and Loubaton [46], and the loss function does not contain any spurious local minima so that all the minima correspond to the source signals. In order to derive a global convergent learning rule, we apply the standard stochastic gradient descent technique. Minimization of the loss function (78) leads to a simple learning rule [39]

$$\boldsymbol{w}_1(k+1) = \boldsymbol{w}_1(k) - \eta_1(k)f(y_1(k))\boldsymbol{x}_1(k) \quad (81)$$

where $\boldsymbol{x}$ is prewhitened and the nonlinear activation function is evaluated adaptively as

$$f(y_1) = \mathrm{sign}[\kappa_4(y_1)]\left[y_1 - \frac{m_2(y_1)}{m_4(y_1)}y_1^3\right]\frac{m_4(y_1)}{m_2^3(y_1)}. \quad (82)$$

The higher order moments m_2 and m_4 and the sign of the kurtosis $\kappa_4(y_1)$ can be estimated on-line by using the following averaging formula

$$m_p(k+1) = (1-\eta)m_p(k) + \eta|y_1(k)|^p \quad (83)$$

with $\eta > 0$ and $m_p(k) \cong E[|y_1(k)|^p]$, $p = 2, 4$.

After the successful extraction of the first source signal $y_1(k) \sim s_i(k)$ $(i \in \overline{1, n})$, we can apply a deflation procedure which removes the previously extracted signals from the mixtures. This procedure may be recursively applied to extract sequentially the rest of the estimated source signals. This means that we require an on-line linear transformation [see Fig. 6(a) and (b)] given by [39]

$$\boldsymbol{x}_{j+1}(k) = \boldsymbol{x}_j(k) - \tilde{\boldsymbol{w}}_j(k) y_j(k), \qquad j = 1, 2, \cdots \tag{84}$$

which ensures minimization of the generalized energy (loss) function

$$l_j(\tilde{\boldsymbol{w}}_j) = \frac{1}{p} \, ||\boldsymbol{x}_{j+1}||^p \tag{85}$$

where $y_j = \boldsymbol{w}_j^T \, \boldsymbol{x}_j$ and

$$\boldsymbol{w}_j(k+1) = \boldsymbol{w}_j(k) - \eta_j(k) f[y_j(k)] \boldsymbol{x}_j(k) \tag{86}$$

$$f(y_j) = \text{sign}[\kappa_4(y_j)] \left[y_j - \frac{m_2(y_j)}{m_4(y_j)} \, y_j^3 \right] \frac{m_4(y_j)}{m_2^3(y_j)}. \tag{87}$$

Minimization of the loss function (85) leads to the simple local learning rule

$$\tilde{\boldsymbol{w}}_j(k+1) = \tilde{\boldsymbol{w}}_j(k) + \tilde{\eta}_j(k) y_j(k) \boldsymbol{g}[\boldsymbol{x}_{j+1}(k)], \quad j = 1, 2, \cdots \tag{88}$$

where $\boldsymbol{g}[\boldsymbol{x}_{j+1}] = [g(x_{j+1,1}) \cdots g(x_{j+1,m})]^T = \partial l_j / \partial \boldsymbol{x}_{j+1}$, with $g(x) = |x|^{p-1} \, \text{sign}(x)$. It is easy to show that vector $\tilde{\boldsymbol{w}}_j$ converges to one the column vector of the global mixing matrix $\boldsymbol{A} = \boldsymbol{QH}$.

The procedure can be continued until all the estimated source signals are recovered, i.e., until the amplitude of each signal $\boldsymbol{x}_{j+1}$ is below some threshold. This procedure means that it is not required to know the number of source signals in advance, but it is assumed that the number is constant and the mixing system is stationary (e.g., the sources do not "change places" during the algorithm's convergence).

In order to extract signals in a specified order, for example, in order of the decreasing absolute values of the normalized kurtosis, an auxiliary Gaussian noise ν can be applied to the nonlinear functions $f(y)$, i.e., $f(\tilde{y}_i) = f(y_i + \nu_i)$, where the Gaussian noise is decreasing in time. In this manner, it may be possible to avoid local minima [29], [39].

The blind signal extraction approach may have several advantages over simultaneous blind separation [39].

1) Signals can be extracted in a specified order according to the stochastic features of the source signals, e.g., in the order of decreasing absolute value of normalized kurtosis. Blind extraction can be considered as a generalization of PCA where decorrelated output signals are extracted according to the decreasing order of their variances.
2) Only the "interesting" signals need be extracted. For example, if the source signals are mixed with a large number of Gaussian noise signals, one can extract only those signals with some desired stochastic properties.
3) The learning algorithms are purely local and hence biologically plausible. Typically, they are also simpler than in the case of blind signal separation. (In fact, the learning algorithms described above can be considered as extensions or modifications of the Hebbian/anti-Hebbian learning rule.)

In summary, blind signal extraction is a useful approach when it is desired to extract several signals with specific stochastic properties from a large number of mixtures. Extraction of a single source is closely related to the problem of blind deconvolution [61], [64], [65], [96].

VII. Adaptive Algorithms for Blind Deconvolution

The approach discussed in previous sections can be generalized for blind deconvolution/equalization problems formulated in Section II-B.

A. Learning Algorithms in the Frequency Domain

When the observed signals $\boldsymbol{x}(k)$ are time-delayed multipath mixtures as shown in (10), we need spatial separation and temporal decomposition in order to extract the source signals $\boldsymbol{s}(k)$. A simple way of extending the blind source separation algorithms is to use frequency domain techniques. By using the Fourier transform, (10) and (11) are represented by

$$\begin{aligned} \boldsymbol{X}(\omega) &= \boldsymbol{H}(\omega)\boldsymbol{S}(\omega) \\ \boldsymbol{Y}(\omega) &= \boldsymbol{W}(\omega)\boldsymbol{X}(\omega) \end{aligned} \tag{89}$$

where the Fourier representation is given by

$$\boldsymbol{X}(\omega) = \text{const} \sum_k \boldsymbol{x}(k) e^{j\omega k} \tag{90}$$

and ω denotes the frequency.

In this case, we have linear matrix expressions of $\boldsymbol{H}$ and $\boldsymbol{W}$ for each frequency ω. Let us discretize the frequency axis, and put $\omega_b = b\omega_0$, $b = 1, 2, \cdots$. Then, our basic learning algorithm (71) can be formulated in the frequency domain as follows:

$$\Delta \underline{\boldsymbol{W}}(b) = \eta(b) \big[\underline{\boldsymbol{\Lambda}}(b) - \boldsymbol{f}\{\underline{\boldsymbol{Y}}(b)\} \underline{\boldsymbol{Y}}^H(b) \big] \underline{\boldsymbol{W}}(b) \tag{91}$$

$$\Delta \boldsymbol{W}_p = \frac{1}{N} \sum_{b=0}^{N-1} e^{j\omega_b p} \Delta \underline{\boldsymbol{W}}(b) \tag{92}$$

where b is the new sampling index in the frequency domain, the superscript H denotes the Hermitian conjugate, N is the number of points in the fast Fourier transform (FFT), and the bold underlined symbols represents variables or blocks in the frequency domain (following the convention used in [74] and [75]). In the above algorithm, $\underline{\boldsymbol{Y}} = \text{FFT}\{\boldsymbol{y}(t)\}$ is a block of vector signals of $\boldsymbol{y}$ in the frequency domain. Diagonal matrix $\underline{\boldsymbol{\Lambda}}$ is equal to the identity matrix $\boldsymbol{I}$ or it is a diagonal matrix defined as $\underline{\boldsymbol{\Lambda}}(b) = \text{diag} \boldsymbol{f}\{\underline{\boldsymbol{Y}}(b)\} \underline{\boldsymbol{Y}}(b)$ [cf. (71) and (72)]. Analogously, $\underline{\boldsymbol{W}}$ is a matrix of FIR filters [74]. Unfortunately, the algorithms in the frequency domain are batch or block algorithms and are computationally intensive. Moreover, they are not responsive to changing environments. To avoid this shortcoming, we can use temporal-windowed Fourier expansion like

$$\boldsymbol{X}(\omega, t) = \sum_{t'} w(t - t') e^{j\omega t'} \boldsymbol{X}(t') \tag{93}$$

where $w(t)$ is the window function. We can then apply an on-line learning algorithm

$$\Delta \boldsymbol{W}(\omega, t) = \eta(t)\big[\boldsymbol{\Lambda} - \boldsymbol{f}\{\boldsymbol{Y}(\omega, t)\}\boldsymbol{Y}^H(\omega, t)\big]\boldsymbol{W}(\omega, t). \tag{94}$$

Its performance depends on the choice of the window function. There are a number of researches in this direction [75], where the natural gradient is used in some of them. We will propose a more direct algorithms in the time domain.

B. Adaptive Algorithms in the Time Domain for Multi-Input Multi-Output Blind Deconvolution

In this section we discuss the natural gradient algorithm for adapting $\boldsymbol{W}(z, k)$ in the convolutive model (12) and (13) for the multichannel deconvolution and equalization task. For the derivation of the learning rule, see Appendix C. Refer to [4] for the Riemannian structure of the space of linear systems. We assume that the sources $\{s_i(k)\}$ are i.i.d. and that both $\boldsymbol{H}(z)$ and $\boldsymbol{W}(z, k)$ are stable with no zero eigenvalues on the unit circle $|z| = 1$ in the complex plane of z. In addition, the partial derivatives of quantities with respect to $\boldsymbol{W}(z, k)$ should be understood as a sequence of matrices of the partial derivatives with respect to $\boldsymbol{W}_p(k)$ indexed by the lag value p [65]. We also assume that the number of sources m equals the number of the sensors n and that all signals and coefficients are real valued, although the final form of the algorithm is described for a more general case [11], [12].

For our derivation, we consider the cost or loss function $l[\boldsymbol{W}(z, k)]$ of the form

$$l[\boldsymbol{W}(z, k)] = -\sum_{i=1}^{m} \log q_i(y_i(k)) - \frac{1}{2\pi j}\oint \cdot \log \det |\boldsymbol{W}(z, k)\boldsymbol{W}^T(z, k)| z^{-1}\, dz \tag{95}$$

where $q_i(y_i)$ is any pdf and $j = \sqrt{-1}$. The expectation of (95) is the Kullback–Leibler divergence, except for a constant, between two stochastic processes, one being $\{\boldsymbol{y}(t)\}$ generated from the output of the system and the other being the spatially independent i.i.d. processes specified by the component pdf's $q_i(y_i)$. The expectation of the first term in (95) can be interpreted as an affine measure of the negative of the joint entropy of the time series $\{\boldsymbol{z}(k) = [z_1(k) \cdots z_m(k)]^T\}$, where $z_i(k) = g_i(y_i(k))$ and $q_i(y_i) = dg_i(y_i)/dy_i$. In addition, when $q_i(y_i)$ is the same as the pdf of the source signal $s_i(k)$, the first term on the right-hand side (RHS) of (95) is the negative of the log likelihood. The second term on the RHS of (95) is a constraint term that prevents $\boldsymbol{W}(z, k)$ from converging to 0.

Minimization of the cost function (95) is complicated but is calculated under the same principle as before (see Appendix C). This leads to the learning rule first developed by Amari *et al.* in [11] and [12]

$$\boldsymbol{W}_p(k+1) = \boldsymbol{W}_p(k) + \eta(k)\big[\delta_p \boldsymbol{W}_p(k) - \boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{u}_p^T(k)\big], \quad -\infty < p < \infty \tag{96}$$

where

$$\boldsymbol{u}_p(k) = \sum_{q=-\infty}^{\infty} \boldsymbol{W}_q^T(k)\boldsymbol{y}(k-p-q). \tag{97}$$

In practice, the doubly-infinite noncausal equalizer cannot be implemented, and thus we approximate it by the FIR causal equalizer given by

$$\boldsymbol{y}(k) = \sum_{p=0}^{L} \boldsymbol{W}_p(k)\boldsymbol{x}(k-p). \tag{98}$$

However, even with this restriction, the pth coefficient matrix update in (96) depends on future equalizer outputs $\boldsymbol{y}(k+q)$, $q \leq L-p$, through the definition of $\boldsymbol{u}_p(k)$ in (98) for the truncated equalizer. Although such difficulties might be overcome by suitably approximated calculations and data storage, such solutions are cumbersome. We instead introduce an L-sample delay into the last term on the right-hand side of (96). This delayed update version maintains the same statistical relationships between the signals in the updates and thus can be expected to provide a performance similar to that of (96). The loss in the performance of this approximation can be expected to be similar to that of the delayed LMS algorithm in adaptive filtering tasks [108]. In addition, the calculation of $\boldsymbol{u}_p(k)$ for $0 \leq p \leq L$ is cumbersome because it requires application of the same multichannel filter to $L+1$ delayed versions of the same signal. For small step sizes $\mu(k)$, we can assume that $\boldsymbol{W}_p(k) \approx \boldsymbol{W}_p(k-1) \approx \cdots \approx \boldsymbol{W}_p(k-2L)$ such that

$$\boldsymbol{u}_p(k) \approx \boldsymbol{u}_0(k-p) \tag{99}$$

which avoids these excessive calculations.

Considering these changes and approximations, in the general case where $m \neq n$ and all signals and coefficients are complex valued, the proposed algorithm for multichannel blind deconvolution and equalization is described as follows [11], [12]:

— compute the estimates of the source signals $\boldsymbol{y}(k)$ from the measured signals $\boldsymbol{x}(k)$ using (98);

— form $\boldsymbol{u}(k)$ as

$$\boldsymbol{u}(k) = \sum_{q=0}^{L} \boldsymbol{W}_q^H(k)\boldsymbol{y}(k-q) \tag{100}$$

— update the coefficient matrices as

$$\boldsymbol{W}_p(k+1) = \boldsymbol{W}_p(k) + \eta(k)[\boldsymbol{\Lambda}(k)\delta_p \boldsymbol{W}_p(k) - \boldsymbol{f}(\boldsymbol{y}(k-L))\boldsymbol{u}^H(k-p)] \tag{101}$$

where $\boldsymbol{\Lambda}(k)$ is a positive-definite diagonal matrix, e.g., $\boldsymbol{\Lambda}(k) = \boldsymbol{I}$.

As previously stated, the optimum choice for each $f_i(y_i)$ depends on the statistics of each $y_i(k)$ at the equilibrium. The optimal choices $f_i(y_i) = -d\log(q_i(y_i))/dy_i$ for the true distributions q_i yield the fastest convergence behavior; however, suboptimal choices for these nonlinearities still allow the algorithm to perform separation and deconvolution of the sources. Since typical digital communication signals are complex valued sub-Gaussian with negative kurtosis [defined as $\kappa_4(y) = E\{y^4\} - 3E^2\{y^2\}$], the choices $f_i(y_i) = |y_i|^2 y_i$ yield adequate separation and deconvolution. Similarly, choosing $f_i(y_i) = y_i/|y_i|$ or $f_i(y_i) = \tanh(\gamma y_i)$ for $\gamma > 2$ enables the algorithm to deconvolve mixtures of super-Gaussian sources with positive kurtosis. In the situations where $\boldsymbol{x}(k)$ contains mixtures of both sub- and super-Gaussian sources, techniques similar to that proposed in Section V can be applied.

Although we do not prove it here, the stability analysis of equilibria can be performed by extending our method, and we can obtain the universally convergent algorithm in the case of blind equalization, too.

C. Adaptive Learning Rules for SISO and SIMO Blind Equalization

A similar approach can be applied to the single channel SISO (single-input single-output) and SIMO (single-input multi-channel outputs—fractionally sampled) blind equalization problems [61], [102]. For SISO blind equalization, Douglas *et al.* developed the following adaptive learning algorithms [49]–[51].

1) Filtered-regressor (FR) algorithm [49], [51]:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \eta(k) f(y(k-L))\boldsymbol{u}_k^* \tag{102}$$

where $\boldsymbol{w}(k) = [w_0(k) \cdots w_L(k)]^T$, $y(k) = \sum_{p=0}^{L} w_p(k)x(k-p)$, and $\boldsymbol{u}_k = [u(k) \cdots u(k-L)]^T$ with

$$u(k) = \sum_{p=0}^{L} w_{L-p}^*(k) y(k-p). \tag{103}$$

2) Extended blind separation (EBS) algorithm:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \eta(k)\boldsymbol{F}[\boldsymbol{y}_k]\boldsymbol{w}(k) \tag{104}$$

with $\boldsymbol{y}_k = [y(k) \cdots y(k-L)]^T$, and the $m \times m$ matrix $\boldsymbol{F}[\boldsymbol{y}_k]$ can take one of the following forms:

$$\boldsymbol{F}^T[\boldsymbol{y}_k] = \boldsymbol{\Lambda}(k) - \boldsymbol{f}(\boldsymbol{y}_k)\boldsymbol{y}_k^H \tag{105}$$

$$\begin{aligned}\boldsymbol{F}^T[\boldsymbol{y}_k] = {} & \boldsymbol{\Lambda}(k) - \alpha_1(k)\boldsymbol{y}_k\boldsymbol{y}_k^H - \alpha_2(k)\boldsymbol{f}(\boldsymbol{y}_k)\boldsymbol{y}_k^H \\ & + \alpha_3(k)\boldsymbol{y}_k\boldsymbol{f}(\boldsymbol{y}_k^H)\end{aligned} \tag{106}$$

where $\boldsymbol{\Lambda}(k)$ is a diagonal positive definite matrix, e.g., $\boldsymbol{\Lambda}(k) = \boldsymbol{I}$ or $\boldsymbol{\Lambda}(k) = \text{diag}\{\boldsymbol{f}(\boldsymbol{y}_k)\boldsymbol{y}_k^H\}$ for (105) and $\alpha_i \geq 0$ are suitable nonnegative parameters as discussed above.

VIII. Learning of Learning Rate Parameters

The problem of optimal updating of the learning rate (step size) is a key problem encountered in all the learning algorithms discussed in this paper. Many of the research works related to this problem are devoted to batch and/or supervised algorithms. Various techniques like the conjugate gradient, quasi-Newton, and Kalman filter methods have been applied. However, relatively little work has been devoted to this problem for on-line adaptive unsupervised algorithms [1], [33], [40], [82], [98].

On-line learning algorithms discussed in this paper can be expressed in the general form as [1], [9], [31], [33], [82], [98]

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \eta(k)\tilde{\boldsymbol{g}}(k) \qquad (k = 0, 1, 2, \cdots) \tag{107}$$

where k is the iteration index, $\boldsymbol{\theta}(k) = [\theta_1(k), \theta_2(k), \cdots, \theta_m(k)]^T$ is the m-dimensional vector of the unknown parameters to be updated, $\eta(k) > 0$ is a learning rate (step size), and $\tilde{\boldsymbol{g}}(k) = \tilde{\boldsymbol{g}}\{\boldsymbol{\theta}(k), \boldsymbol{x}(k), \boldsymbol{y}(k)\} = [\tilde{g}_1(k), \tilde{g}_2(k), \cdots, \tilde{g}_n(k)]^T$ is a nonlinear function depending on $\boldsymbol{\theta}(k)$, as well as $\boldsymbol{x}(k)$ and $\boldsymbol{y}(k)$ (respectively, input and output signals) which can be considered as an instantaneous estimate of the gradient.

It is well known that the final misadjustment (often defined in terms of the MSE) increases as the learning rate η increases. However, convergence time increases as the learning rate decreases [1]. For this reason it is often assumed that the learning rate η is a very small positive constant, either fixed or exponentially decreasing to zero as time goes to infinity. Such an approach leads to a relatively slow convergence speed and/or low performance and is not suitable for nonstationary environments.

This inherent limitation of on-line adaptive algorithms represented by (107) imposes a compromise between the two opposing fundamental requirements of the small miss-adjustment and fast convergence demanded in most applications, especially in nonstationary environments [1], [82]. As a result, it is desirable to find an alternative method to improve the performances of such algorithms. Most work has been devoted to variable step size LMS-type (supervised delta rule) algorithms. In this section we will consider a more general case which includes both unsupervised and/or supervised on-line algorithms.

A very old work [1] was devoted to the analysis of on-line dynamic behavior of $\boldsymbol{\theta}(k)$ in the neighborhood of the optimal $\boldsymbol{\theta}_*$ and obtained analytical formulas for the convergence speed to $\boldsymbol{\theta}_*$ and the fluctuation of $\boldsymbol{\theta}(k)$ around $\boldsymbol{\theta}_*$. Moreover, this paper suggested a method for a variable learning rate [1]. In this section we extend this idea of learning of the learning rate by proposing robust, variable step-size on-line algorithms. The main objective is to propose a simple and efficient algorithm which enables automatic update of the learning rate, especially in nonstationary environments.

Recently, we have developed a new family of on-line algorithms with an adaptive learning rate (see Fig. 7) suitable for nonstationary environments [31], [33], [40]

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \eta(k)\tilde{\boldsymbol{g}}(k) \tag{108}$$

$$\hat{\boldsymbol{g}}(k) = (1-\rho_2)\hat{\boldsymbol{g}}(k-1) + \rho_2\tilde{\boldsymbol{g}}(k) \tag{109}$$

$$\eta(k) = (1-\rho_1)\eta(k-1) + \rho_1\beta\psi(||\hat{\boldsymbol{g}}(k)||) \tag{110}$$

or

$$[1 - \rho_1\eta(k-1)]\eta(k-1) + \rho_1\beta\eta(k-1)\psi(||\hat{\boldsymbol{g}}(k)||) \tag{111}$$

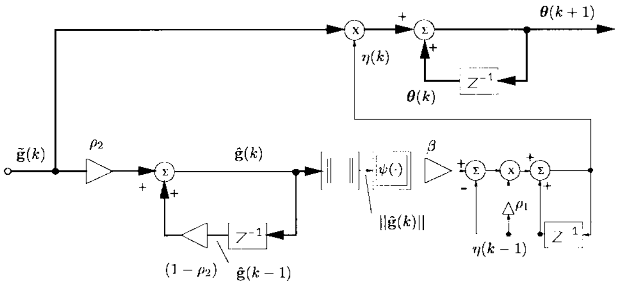


Fig. 7. Implementation of self-adaptive learning of learning rate.

where $0 < \rho_1 < 1$, $0 < \rho_2 < 1$, $\beta > 0$ are fixed coefficients and $\psi(||\hat{\boldsymbol{g}}(k)||)$ is a nonlinear function defined as $\Psi(||\hat{\boldsymbol{g}}(k)||) = (1/m)\sum_{i=1}^{m}|\hat{g}_i(k)|$ or $\psi(||\hat{\boldsymbol{g}}(k)||) = \tanh((1/n)\sum_{i=1}^{n}\hat{g}_i(k)^2)$. It should be noted that (111) has been obtained from (110) by simply replacing the fixed ρ_1 by the self-adaptive term $\rho_1\eta(k-1)$. The above algorithm is related to the very recent research works of Murata *et al.* and Sompolinski *et al.* [82], [98]. It is interesting to note that (109) and (110) describe simple first-order low-pass filters with cut-off frequencies determined by the parameters ρ_1 and ρ_2. The even nonlinear function ψ is introduced to limit the maximum value of the gradient norm $||\hat{\boldsymbol{g}}(k)||$, and the maximal value of the gain β is constrained to ensure stability of the algorithm [33].

It should be noted that for a fixed η the system behaves in such a way that parameters $\theta_i(k)$ never achieve a steady state but will fluctuate around some equilibrium point. In order to reduce such fluctuations we have employed two low-pass filters (LPF's). Intuitively, the self-adaptive system described by (108)–(111) operates as follows. If gradient components $\tilde{g}_i(k)$ have local average (mean) values which differ from zero (which are extracted by the first LPF's), then the learning rate $\eta(k)$ is decreasing or increasing to a certain value determined by the gain β and the norm of the gradient components. However, during the learning process, $|\hat{g}_i(k)|$ decreases to zero, i.e., after some time $\tilde{g}_i(k)$ starts to fluctuate around zero, and then $\eta(k)$ also decreases to zero as desired. If some rapid changes occur in the system, then $|\hat{g}_i(k)|$ suddenly increases and consequently $\eta(k)$ also rapidly increases from a small value so that the system is able automatically to adapt quickly to the new environment.

We have found that these algorithms work with high efficiency and they are easily implemented in very large scale integration (VLSI) technology. However, they require the proper selection of three parameters (ρ_1, ρ_2, β). Although the above algorithms are robust with respect to the values of these parameters, their optimal choice is problem-dependent.

IX. Exemplary Computer Simulation Experiments

Most of the adaptive learning algorithms described on this paper have been investigated and tested by computer simulations. Extensive simulations confirm the validity and high performance of the developed algorithms. Due to limitations of space we present here only three illustrative examples. For more details see also [11], [12], [29]–[43], [51], and [52].

Example 1: In this experiment five color images, as shown in Fig. 8(a), have been mixed by a randomly chosen ill conditioned, nonsingular 5×5 mixing matrix $\boldsymbol{H}$. Two-dimensional mixed images shown in Fig. 8(b) have been converted to one-dimensional zero-mean signals and then separated by feed-forward network with the learning algorithm (38) with nonlinear activation functions of the form $f_i(y_i) = |y_i|^{r_i-1}\,\mathrm{sign}(y_i)$, where parameters r_i change between one and ten depending on value of kurtosis of separating signals. The adaptive algorithm was able to separate images in less than 5000 iterations using the self-adaptive learning rate (108)–(111). Although restoration of the original images is not perfect, the performance index expressed by peak signal to noise ratio (PSNR) was higher than 25 dB [36], [37].

Example 2: In the second experiment, three unknown acoustical signals (two natural speech signals with positive kurtosis and a single tone signal with negative kurtosis) where mixed using randomly selected 7×3 full rank mixing matrix $\boldsymbol{H}$ (see Fig. 9). To sensors signals were added 3% i.i.d. Gaussian noises. The mixing matrix and the number of sources, as well as their statistics, were assumed to be completely unknown. The learning algorithms (72) with self-adaptive learning rate $\eta(k)$ (111) and nonlinear activation functions $f_i(y_i) = y_i/[|y_i|^{2-r_i} + \epsilon]$, where $r_i = 0.5$ and $\epsilon = 0.0001$ if the extracted signal y_i has positive kurtosis and $r_i = 4$, and $\epsilon = 0$ if it has negative kurtosis was able to successfully estimate the number of active sources and their waveforms [14], [41]. In this case four redundant outputs collects only additive sensor noises, while the other three outputs estimate original source signals with reduced noise. In the special case where the sensors signals are noiseless four output signals decay quickly to zero while three other reconstruct original sources.

Example 3: In this more real-world experiment, three natural speech signals are convolved and mixed together by using matrix of FIR filters of order 25. In order to perform multichannel blind deconvolution we apply a feed-forward network with FIR filters of order $L = 256$ with the associated learning algorithm (101). As shown in Fig. 10, the learning algorithm is able to restore the original source signals with a relatively short time less than 11 000 iterations. In order to improve convergence speed, we have added to the output signals auxiliary noise signals gradually attenuating to zero [29].

X. Conclusions, Discussions, and Open Problems

In this paper, we have reviewed new promising approaches to adaptive blind signal processing. A family of robust learning adaptive algorithms are derived or mathematically justified and their properties are briefly analyzed. Exemplary computer simulation experiments are also given to illustrate the performances of the proposed algorithms. Due to wide interest in this fascinating area of research, we expect, in the near future, further developments of computationally efficient separation, deconvolution, equalization

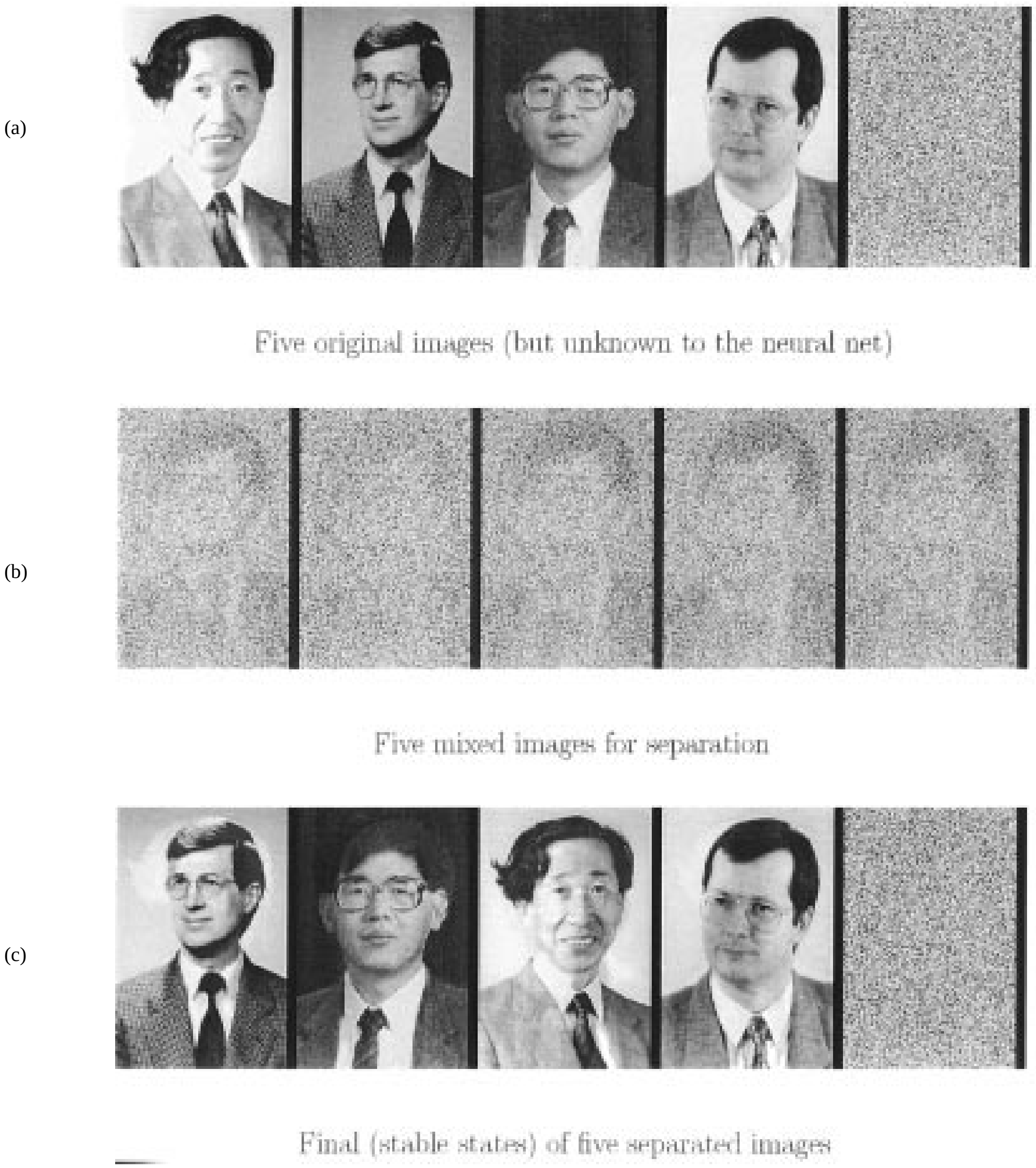
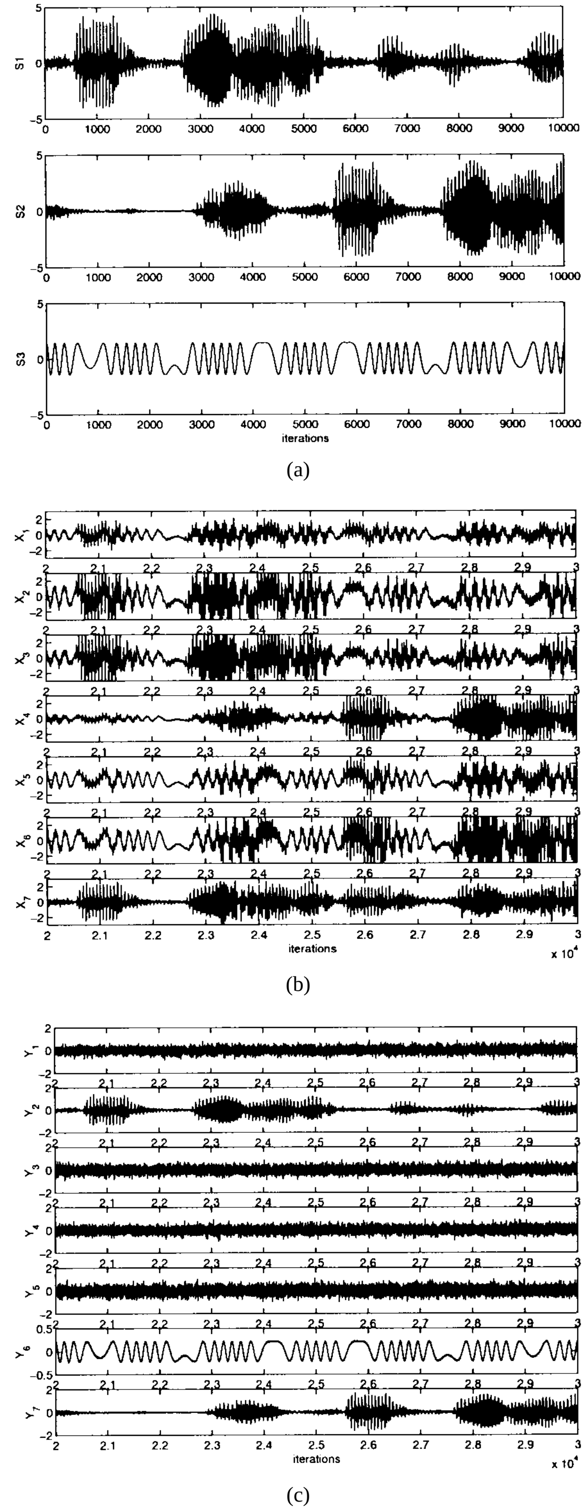

(a)
Five original images (but unknown to the neural net)
(b)
Five mixed images for separation
(c)
Final (stable states) of five separated images

Fig. 8. Example 1: blind separation of face images and noise: (a) original source images, (b) mixed images, and (c) restored images after applying learning algorithm (38) or (72).

self-adaptive or self-organized systems with robust on-line algorithms for many real world applications like wireless communication, the "cocktail party" problem, speech and image recognition, intelligent analysis of medical signals and images, feature extraction, etc.

A number of open problems not addressed in this paper still exist to our knowledge in blind signal processing. We formulate here ten such basic problems.

1) How can one effectively use some *a priori* information about the linear and nonlinear dynamic system in order to successfully separate or extract the source signals?
2) What methods can be effective in the case when there are more source signals than sensors (e.g., for EEG signals)? What *a priori* information is sufficient and/or necessary in such case?
3) How can the number of the sources be reliably estimated in the presence of large amplitude of colored (non-Gaussian) and/or impulsive noises when the reference noise is not available [36], [42], [43].
4) What are the theoretical and practical limitations on blind source separation/deconvolution when the mixtures or channels are nonlinear? It is recognized that there will exist nonlinearities for which no solution exists, but it is desirable to quantify these limitations, and further, to establish the types of problems which are solvable effectively. Developments of the inverse models of nonlinear dynamic systems are necessary [112].

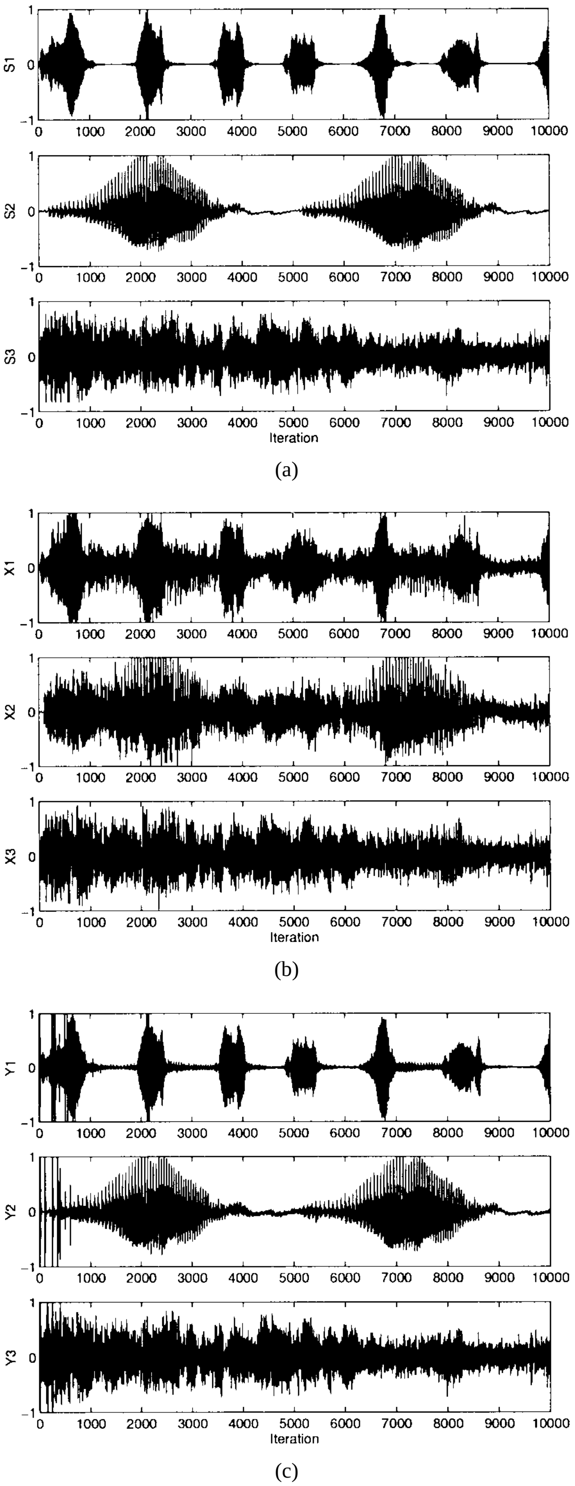


(a)

(b)

(c)

Fig. 9. Example 2: testing the separation abilities when the number of source signals is unknown: (a) three original natural voice signals, (b) seven noisy mixed signals, and (c) restored source signals using the adaptive algorithm (72) with self-adaptive learning rate.



(a)

(b)

(c)

Fig. 10. Example 3 of blind deconvolution of speech signals: (a) original acoustical signals, (b) convolved and mixed signals, and (c) restored signals during the learning process.

5) What are necessary and sufficient conditions for successful blind separation and/or deconvolution problems? What are the practical limitations how to extend or modify the existing criteria, especially when the usual i.i.d. and mutual statistical independence conditions are not fully satisfied or when the data are correlated?

6) Comparison and further developments of computationally efficient separation, deconvolution, equalization adaptive systems with algorithms for real-world applications. For example, in the problem of reverberation and echo or long channel delays, what methods

can be developed to perform fast multichannel blind source separation and deconvolution/equalization?

7) Comparison between time and frequency domain algorithms, especially for ill-conditioned problems with respect to complexity, robustness and biological plausibility.
8) Implementation of adaptive stable recurrent neural networks, especially IIR filters (e.g., Gamma, Laguerre filters) and/or state space models instead of standard FIR channels.
9) Development of the inverse models and adaptive algorithms for time-varying nonstationary systems, i.e., channels with varying characteristic and/or number of sources varying, e.g., an unknown number of sources moving in space and varying in time (e.g., acoustic or speech signals with quiet periods). Estimation of variable time delays for source signals (e.g., acoustic sources moving in space).
10) Robustness and tracking abilities of learning adaptive algorithms, especially in the presence of noise.

Appendix A
Derivation of Basic Learning Rule for BSS

In order to calculate the gradient of $l(\boldsymbol{W}, \boldsymbol{y})$ expressed by (31), we derive the total differential dl of l when $\boldsymbol{W}$ is changed from $\boldsymbol{W}$ to $\boldsymbol{W} + d\boldsymbol{W}$. In component form

$$dl = l(\boldsymbol{y}, \boldsymbol{W} + d\boldsymbol{W}) - l(\boldsymbol{y}, \boldsymbol{W}) = \sum_{i,\, j} \frac{\partial l}{\partial w_{ij}} \, dw_{ij} \quad (112)$$

where the coefficients $\partial l / \partial w_{ij}$ of dw_{ij} represent the gradient of l. Simple algebraic and differential calculus yields

$$dl = -\text{tr}(d\boldsymbol{W}\boldsymbol{W}^{-1}) + \boldsymbol{f}(\boldsymbol{y}^T)\, d\boldsymbol{y} \quad (113)$$

where tr is the trace of a matrix and $\boldsymbol{f}(\boldsymbol{y})$ is a column vector whose components are $f_i(y_i) = -(q_i'(y_i)/q_i(y_i))$. From $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$, we have

$$d\boldsymbol{y} = d\boldsymbol{W}\boldsymbol{x} = d\boldsymbol{W}\boldsymbol{W}^{-1}\boldsymbol{y}. \quad (114)$$

Hence, we put

$$d\boldsymbol{X} = d\boldsymbol{W}\boldsymbol{W}^{-1} \quad (115)$$

whose components dx_{ij} are linear combinations of dw_{ij}. The differentials $\{dx_{ij}\}$ form a basis of the tangent space of nonsingular matrices $\boldsymbol{W}$ since they are linear combinations of the basis $\{dw_{ij}\}$. It should be noted that $d\boldsymbol{X} = d\boldsymbol{W}\boldsymbol{W}^{-1}$ is a nonintegrable differential form so that we do not have a matrix function $\boldsymbol{X}(\boldsymbol{W})$ which gives (115). Nevertheless, the nonholonomic basis $d\boldsymbol{X}$ has a definite geometrical meaning and is very useful. It is effective to analyze the differential in terms of $d\boldsymbol{X}$, since the natural Riemannian gradient [6]–[8] is automatically implemented by it and the equivariant properties investigated in [22] automatically hold in this basis. It is easy to rewrite the results in terms of $d\boldsymbol{W}$ by using (115). The gradient dl in (31) is expressed by the differential form

$$dl = -\text{tr}(d\boldsymbol{X}) + \boldsymbol{f}(\boldsymbol{y}^T)\, d\boldsymbol{X}\boldsymbol{y}. \quad (116)$$

This leads to the stochastic gradient learning algorithm

$$\begin{aligned} \Delta\boldsymbol{X}(k) &= \boldsymbol{X}(k+1) - \boldsymbol{X}(k) \\ &= -\eta(k)\, \frac{dl}{d\boldsymbol{X}} \\ &= \eta(k)[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{y}^T(k)] \end{aligned} \quad (117)$$

in terms of $\Delta\boldsymbol{X}(k) = \Delta\boldsymbol{W}(k)\boldsymbol{W}^{-1}(k)$, or

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) + \eta(k)[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{y}^T(k)]\boldsymbol{W}(k) \quad (118)$$

in terms of $\boldsymbol{W}(k)$.

Appendix B
Stability of the Basic Learning Rule

We consider the expected version of the learning equation

$$\frac{d\boldsymbol{W}}{dt} = \dot{\boldsymbol{W}}(t) = \mu(t) E\big[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y}(t))\boldsymbol{y}^T(t)\big]\boldsymbol{W}(t) \quad (119)$$

where we use the continuous time version of the algorithm for simplicity's sake. By linearizing it at the equilibrium point, we have the variational equation

$$\delta\dot{\boldsymbol{W}}(t) = \mu(t) \frac{\partial\big(E\big[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y}(t))\boldsymbol{y}^T(t)\big]\boldsymbol{W}\big)}{\partial \boldsymbol{W}}\, \delta\boldsymbol{W} \quad (120)$$

where $(\partial/\partial\boldsymbol{W})\delta\boldsymbol{W}$ implies $\sum(\partial/\partial w_{ij})\delta w_{ij}$ in component form. This shows that, only when all the eigenvalues of the operator $\boldsymbol{K}(\boldsymbol{W}, \boldsymbol{y}) = \partial^2 l(\boldsymbol{W}, \boldsymbol{y})/\partial\boldsymbol{W}\partial\boldsymbol{W} = \partial(E[\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y})\boldsymbol{y}^T]\boldsymbol{W})/\partial\boldsymbol{W}$ have negative real parts, the equilibrium is asymptotically stable. Therefore, we need to evaluate all the eigenvalues of the operator. This can be done in terms of $d\boldsymbol{X}$, as follows. Since $\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{y})\boldsymbol{y}^T$ is derived from the gradient dl, we need to calculate its Hessian $d^2 l$

$$d^2 l = \sum \frac{\partial l(\boldsymbol{y}, \boldsymbol{W})}{\partial w_{ij}\, \partial w_{kl}}\, dw_{ij}\, dw_{kl}$$

in terms of $d\boldsymbol{X}$. The equilibrium is stable if and only if the expectation of the above quadratic form is positive definite. We calculate the second total differential, which is the quadratic form of the Hessian of l, as [13]

$$\begin{aligned} d^2 l &= \boldsymbol{y}^T\, d\boldsymbol{X}^T \boldsymbol{f}'(\boldsymbol{y})\, d\boldsymbol{y} + \boldsymbol{f}(\boldsymbol{y}^T)\, d\boldsymbol{X}\, d\boldsymbol{y} \\ &= \boldsymbol{y}^T\, d\boldsymbol{X}^T \boldsymbol{f}'(\boldsymbol{y})\, d\boldsymbol{X}\boldsymbol{y} + \boldsymbol{f}(\boldsymbol{y}^T)\, d\boldsymbol{X}\, d\boldsymbol{X}\boldsymbol{y}. \end{aligned} \quad (121)$$

The expectation of the first term is

$$\begin{aligned} E\big[\boldsymbol{y}^T\, d\boldsymbol{X}^T \boldsymbol{f}'(\boldsymbol{y})\, d\boldsymbol{X}\boldsymbol{y}\big] &= \sum E\big[y_i\, dx_{ji} f_j'(y_j)\, dx_{jk} y_k\big] \\ &= \sum_{j \neq i} E\big[(y_i)^2\big] E\big[f_j'(y_j)\big] (dx_{ji})^2 \\ &\quad + \sum_i E\big[(y_i)^2 f_i'(y_i)\big] (dx_{ii})^2 \\ &= \sum_{j \neq i} \sigma_i^2 \kappa_j (dx_{ji})^2 + \sum_i m_i (dx_{ii})^2. \end{aligned}$$

Here, the expectation is taken at $\boldsymbol{W} = \boldsymbol{A}^{-1}$ where y_i's are independent.

Similarly

$$\begin{aligned} E\big[\boldsymbol{f}(\boldsymbol{y})^T\, d\boldsymbol{X}\, d\boldsymbol{X}\boldsymbol{y}\big] &= \sum E[f(y_i)\, dx_{ij}\, dx_{jk} y_k] \\ &= \sum E[y_i f(y_i)]\, dx_{ij}\, dx_{ji} \\ &= \sum_{i,\,j} dx_{ij}\, dx_{ji} \end{aligned} \tag{122}$$

because $E[y_i f_i(y_i)] = 1$ (the normalization condition). Hence

$$\begin{aligned} E[d^2 l] = \sum_{j \neq i} &\{\sigma_i^2 \kappa_j (dx_{ji})^2 + dx_{ij}\, dx_{ji}\} \\ &+ \sum_i (m_i + 1)(dx_{ii})^2. \end{aligned} \tag{123}$$

For a pair (i, j), $i \neq j$, the summand in the first term is rewritten as

$$k_{ij} = \sigma_i^2 \kappa_j (dx_{ji})^2 + \sigma_j^2 \kappa_i (dx_{ij})^2 + 2\, dx_{ij}\, dx_{ji}. \tag{124}$$

This $k_{ij} (i \neq j)$ is the quadratic form in (dx_{ij}, dx_{ji}), and

$$E[d^2 l] = \sum_{i \neq j} k_{ij} + \sum (m_i + 1)(dx_{ii})^2. \tag{125}$$

The k_{ij} is positive if and only if stability conditions (45)–(47) hold.

Appendix C
Derivation of Learning Algorithm for the Multichannel Blind Deconvolution [11], [12]

We now determine the stochastic learning algorithm, in particular, the natural gradient algorithm for minimizing the expected value of $l[\boldsymbol{W}(z, k)]$ with respect to $\boldsymbol{W}(z, k)$ [11], [12]. To do so, we determine the total differential $l[\boldsymbol{W}(z, k)]$ when $\boldsymbol{W}(z, k)$ undergoes an infinitesimal change $d\boldsymbol{W}(z, k)$ as

$$dl[\boldsymbol{W}(z, k)] = l[\boldsymbol{W}(z, k) + d\boldsymbol{W}(z, k)] - l[\boldsymbol{W}(z, k)] \tag{126}$$

as this corresponds to the gradient of the cost function with respect to $\boldsymbol{W}(z, k)$. Let us define $f_i(y_i) = -d \log q_i(y_i)/dy_i$. Then

$$d\left(-\sum_{i=1}^m \log q_i(y_i(k))\right) = \sum_{i=1}^m f_i(y_i(k))\, dy_i(k) \tag{127}$$

$$= \boldsymbol{f}^T(\boldsymbol{y}(k))\, d\boldsymbol{y}(k) \tag{128}$$

in which $d\boldsymbol{y}(k)$ is given in terms of $d\boldsymbol{W}(z, k)$ as

$$d\boldsymbol{y}(k) = d\boldsymbol{W}(z, k)[\boldsymbol{x}(k)] \tag{129}$$

$$= d\boldsymbol{W}(z, k)\boldsymbol{W}^{-1}(z, k)[\boldsymbol{y}(k)]. \tag{130}$$

Define a modified differential $d\boldsymbol{X}(z, k)$ by

$$d\boldsymbol{X}(z, k) = d\boldsymbol{W}(z, k)\boldsymbol{W}^{-1}(z, k) \tag{131}$$

$$= \sum_{p=-\infty}^{\infty} d\boldsymbol{X}_p(k) z^{-p} \tag{132}$$

where the matrices $\{d\boldsymbol{X}_p(k)\}$ are defined by the inverse-z-transform of $d\boldsymbol{W}(z, k)\boldsymbol{W}^{-1}(z, k)$. With this definition, we have

$$d\left(-\sum_{i=1}^m \log p_i(y_i(k))\right) = \boldsymbol{f}^T(\boldsymbol{y}(k))\, d\boldsymbol{X}(z, k)[\boldsymbol{y}(k)]. \tag{133}$$

Similarly, it can be seen that

$$\begin{aligned} d&\left(\frac{1}{2\pi j}\oint \log|\det \boldsymbol{W}(z, k)| z^{-1}\, dz\right) \\ &= \frac{1}{2\pi j}\oint \operatorname{tr}\big(d\boldsymbol{W}(z, k)\boldsymbol{W}^{-1}(z, k)\big) z^{-1}\, dz \quad (134) \\ &= \operatorname{tr}(d\boldsymbol{X}_0(k)) \quad (135) \end{aligned}$$

where $\operatorname{tr}(\cdot)$ denotes the trace operation. Thus, combining (133) and (135) gives

$$dl[\boldsymbol{W}(z, k)] = \boldsymbol{f}^T(\boldsymbol{y}(k))\, d\boldsymbol{X}(z, k)[\boldsymbol{y}(k)] - \operatorname{tr}(d\boldsymbol{X}_0(k)). \tag{136}$$

The differential in (136) is in terms of the modified coefficient differential matrix $d\boldsymbol{X}(z, k)$. Note that $d\boldsymbol{X}(z, k)$ is a linear combination of the coefficient differentials $dW_{ij}(z, k)$ in the matrix polynomial $\boldsymbol{W}(z, k)$. Thus, so long as $\boldsymbol{W}(z, k)$ is nonsingular, $d\boldsymbol{X}(z, k)$ represents a valid search direction to adjust the polynomial $\boldsymbol{X}(z, k)$ to minimize (95), because $d\boldsymbol{X}(z, k)$ spans the same tangent space of matrices as spanned by $d\boldsymbol{W}(z, k)$. For these reasons, one could use an alternative stochastic gradient search method of the form

$$\Delta\boldsymbol{X}(z, k) = -\eta(k)\,\frac{dl[\boldsymbol{W}(z, k)]}{\Delta\boldsymbol{X}_p}. \tag{137}$$

Hence taking into account (132) we have

$$\boldsymbol{W}_p(k+1) = \boldsymbol{W}_p(k) - \eta(k)\left[\frac{\partial l[\boldsymbol{W}(z, k)]}{\partial \boldsymbol{X}_p(k)}\right]\boldsymbol{W}(z, k) \tag{138}$$

where the right-sided operator $\boldsymbol{W}(z, k)$ acts on the gradient term in brackets only in the time dimension p. This search direction is nothing more than the natural gradient search direction using the Riemannian metric tensor of the space of all matrix filters of the form of $\boldsymbol{W}(z, k)$ in (14) [7]. Now, we note from (132) and (136) that

$$\begin{aligned} \left[\frac{\partial l[\boldsymbol{W}(z, k)]}{\partial \boldsymbol{X}_p(k)}\right]\boldsymbol{W}(z, k) &= \boldsymbol{f}(\boldsymbol{y}(k))\big[\boldsymbol{y}^T(k-p)\big]\boldsymbol{W}(z, k) \\ &\quad - [\boldsymbol{I}\delta_p]\boldsymbol{W}(z, k). \end{aligned} \tag{139}$$

The second term on the RHS of (139) simplifies to $\delta_p \boldsymbol{W}_p(k)$. To express the first term, define

$$\boldsymbol{u}_p(k) = \sum_{q=-\infty}^{\infty} \boldsymbol{W}_q^T(k)\boldsymbol{y}(k-p-q). \tag{140}$$

Then, substituting (139) and (140) into (138) gives the coefficient updates as

$$\begin{aligned} \boldsymbol{W}_p(k+1) = \boldsymbol{W}_p(k) + \eta(k))&\big[\delta_p \boldsymbol{W}_p(k) - \boldsymbol{f}(\boldsymbol{y}(k))\boldsymbol{u}_p^T(k)\big], \\ &-\infty < p < \infty. \end{aligned} \tag{141}$$

The stability analysis of the equilibrium can be performed in a similar way as in Appendix B.

Acknowledgment

The authors wish to thank all present and former researchers associated with the Brain-Style Information Systems Group, Riken Brain Science Institute, Japan, who were involved in this research project. In particular, they wish to thank Dr. S. C. Douglas, Dr. S. Choi, Dr. W. Kasprzak, Dr. J. Cao, Dr. C. Fyfe, Dr. R. Thawonmas, Dr. A. Back, and Dr. N. Murata for their fruitful collaboration and helpful discussions.

References

[1] S. Amari, "A theory of adaptive pattern classifiers," *IEEE Trans. Electro. Comput.*, vol. EC-16, pp. 299–307, Mar. 1967.

[2] S.-I. Amari, "Neural theory of association and concept-formation," *Biological Cybern.*, vol. 26, pp. 175–185, 1977.

[3] ———, "Mathematical theory of neural learning," *New Generation of Computing*, vol. 8, pp. 135–143, 1991.

[4] S. Amari, "Differential geometry of a parametric family of invertible linear systems—Riemannian metric, dual affine connections and divergence," *Mathematical Syst. Theory*, vol. 20, pp. 53–82, 1987.

[5] ———, *Differential-Geometrical Methods of Statistics* (Springer Lecture Notes in Statistics), vol. 28. Berlin: Springer, 1985.

[6] S. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems, Vol. 8, (NIPS-1995).* Boston, MA: MIT Press, 1996, pp. 752–763.

[7] S. Amari "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, pp. 251–276, Jan. 1998.

[8] S. Amari and J. F. Cardoso, "Blind source separation—Semi-parametric statistical approach," *IEEE Trans. Signal Processing*, vol. 45, pp. 2692–2700, Nov. 1997.

[9] S. Amari "Super-efficiency in blind source separation," submitted for publication.

[10] S. Amari, A. Cichocki, and H. H. Yang, "Recurrent neural networks for blind separation of sources," in *Proc. Int. Symp. Nonlinear Theory and its Applications*, NOLTA-95, Las Vegas, Dec. 1995, pp. 37–42.

[11] S. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang, "Multichannel blind deconvolution and equalization using the natural gradient," in *Proc. IEEE Int. Workshop Wireless Communication*, Paris, Apr. 1997, pp. 101–104.

[12] ———, "Novel on-line algorithms for blind deconvolution using natural gradient approach," in *Proc. 11th IFAC Symp. System Identification, SYSID-97*, Kitakyushu City, Japan, July 1997, pp. 1057–1062.

[13] S. Amari, T.-P. Chen, and A. Cichocki, "Stability analysis of adaptive blind source separation," *Neural Networks*, vol. 10, no. 8, pp. 1345–1351, 1997.

[14] ———, "Non-holonomic constraints in learning algorithms for blind source separation," *Neural Computation*, to be published.

[15] S. Amari and M. Kawanabe, "Information geometry of estimating functions in semi-parametric statistical models," *Bernoulli*, vol. 3, no. 1, pp. 29–54, 1997.

[16] ———, "Estimating functions in semi-parametric statistical models," *Estimating Functions* (IMS Monograph Series, vol. 32), I. V. Basawa, V. P. Godambe, and R. L. Taylor, Eds. IMS, 1998, pp. 65–81.

[17] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[18] A. Belouchrani, A. Cichocki, and K. Abed Meraim, "A blind identification and separation technique via multi-layer neural networks," in *Proc. Progress in Neural Information Processing—ICONIP'96*, Hong Kong, Sept. 1996, Springer-Verlag Singapore Ltd., vol. 2, pp. 1195–1200.

[19] A. Benveniste, M. Goursat, and G. Ruget, "Robust identification of a nonminimum phase system: Blind adjustment of a linear equalizer in data communications," *IEEE Trans. Automat. Contr.*, vol. AC-25, pp. 385–399, June 1980.

[20] X.-R. Cao and R.-W. Liu, "General approach to blind source separation," *IEEE Trans. Signal Processing*, vol. 44, pp. 562–571, Mar. 1996.

[21] J.-F. Cardoso, "Infomax and maximum likelihood for blind source separation," *IEEE Signal Processing Lett.*, vol. 4, pp. 109–111, Apr. 1997.

[22] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing*, vol. SP-43, pp. 3017–3029, Dec. 1996.

[23] V. Capdevielle, C. Serviere, and J. Lacoume, "Blind separation of wide-band sources in the frequency domain," in *Proc. ICASSP*, May 1995, pp. 2080–2083.

[24] D. B. Chan, P. J. W. Rayner, and S. J. Godsill, "Multi-channel signal separation," in *Proc. ICASSP*, May 1996, pp. 649–652.

[25] S. Choi, R.-W. Liu, and A. Cichocki, "A spurious equilibria-free learning algorithm for the blind separation of nonzero skewness signals," *Neural Processing Lett.*, vol. 7, no. 2, pp. 61–68, Jan. 1998.

[26] S. Choi and A. Cichocki, "Cascade neural networks for multichannel blind deconvolution," *Electron. Lett.*, vol. 30, no. 12, pp. 1186–1187, 1998.

[27] A. Cichocki, R. Swiniarski, and R. Bogner, "Hierarchical neural network for robust PCA of complex-valued signals," in *Proc. of World Congress on Neural Networks, WCNN'96—1996, Int. Neural Network Soc. Annu. Meeting*, San Diego, USA, Sept. 1996, pp. 818–821.

[28] A. Cichocki and R. Unbehauen, "Robust estimation of principal components in real time," *Electron. Lett.*, vol. 29, no. 21, pp. 1869–1870, 1993.

[29] A. Cichocki, S. Amari, and J. Cao, "Neural network models for blind separation of time delayed and convolved signals," *Japanese IEICE Trans. Fundamentals*, vol. E-82-A, no. 9, pp. 1057–1062, Sept. 1997.

[30] A. Cichocki, R. Unbehauen, and E. Rummert, "Robust learning algorithm for blind separation of signals," *Electronics Lett.*, vol. 30, no. 17, pp. 1386–1387, Aug. 1994.

[31] A. Cichocki, R. Unbehauen, L. Moszczyński, and E. Rummert, "A new on-line adaptive learning algorithm for blind separation of source signals," in *Proc. ISANN-94*, Taiwan, Dec. 1994, pp. 406–411.

[32] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing.* Chichester: Wiley, 1994, ch. 8, pp. 416–471.

[33] A. Cichocki, S. Amari, M. Adachi, and W. Kasprzak, "Self-adaptive neural networks for blind separation of sources," in *Proc. 1996 IEEE Int. Symp. Circuits and Systems*, May 1996, vol. 2, pp. 157–160.

[34] A. Cichocki, S. Amari, and J. Cao, "Blind separation of delayed and convolved signals with self-adaptive learning rate," in *1996 IEEE Int. Symp. Nonlinear Theory and its Applications*, NOLTA-96, Kochi, Japan, Oct. 7–9, 1996, pp. 229–232.

[35] A.Cichocki and R. Unbehauen, "Robust neural networks with on-line learning for blind identification and blind separation of sources," *IEEE Trans. Circuits Syst. I*, vol. 43, pp. 894–906, Nov. 1996.

[36] A. Cichocki, W. Kasprzak, and S. Amari, "Adaptive approach to blind source separation with cancellation of additive and convolutional noise," in *Proc. ICSP'96, 3rd Int. Conf. Signal Processing* , Sept. 1996, vol. I, pp. 412–415.

[37] ———, "Neural network approach to blind separation and enhancement of images," in *Signal Processing VIII. Theories and Applications*, vol. I, Sept. 1996, pp. 579–582.

[38] A. Cichocki, R. E. Bogner, L. Moszczynski, and K. Pope, "Modified Herault–Jutten algorithms for blind separation of sources," *Digital Signal Processing*, vol. 7, no. 2, pp. 80–93, Apr. 1997.

[39] A. Cichocki, R. Thawonmas, and S. Amari, "Sequential blind signal extraction in order specified by stochastic properties," *Electron. Lett.*, vol. 33, no. 1, pp. 64–65, Jan. 1997.

[40] A. Cichocki, B. Orsier, A. D. Back, and S. Amari, "On-line adaptive algorithms in non stationary environments using a modified conjugate gradient approach," in *Proc. IEEE Workshop Neural Networks for Signal Processing*, Sept. 1997, pp. 316–325.

[41] A. Cichocki, I. Sabala, S. Choi, B. Orsier, and R. Szupiluk, "Self adaptive independent component analysis for sub-Gaussian and super-Gaussian mixtures with unknown number of sources and additive noise," in *Proc. 1997 Int. Symp. Nonlinear Theory and its Applications (NOLTA-97)*, Hawaii, Dec. 1997, vol. 2, pp. 731–734.

[42] A. Cichocki, I. Sabala, and S. Amari, "Intelligent neural networks for blind signal separation with unknown number of

sources," in *Proc. Conf. Engineering of Intelligent Systems, ESI-98*, Tenerife, Feb. 11–13, 1998, pp. 148–154.

[43] A. Cichocki, S. Douglas, S. Amari, and P. Mierzejewski, "Independent component analysis for noisy data," in *Proc. Int. Workshop Independence and Artificial Neural Networks*, Tenerife, Feb. 9/10, 1998, pp. 52–58.

[44] P. Comon, "Independent component analysis: A new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.

[45] ———, "Contrast functions for blind deconvolution," *IEEE Signal Processing Lett.*, vol. 3, pp. 209–211, July 1996.

[46] N. Delfosse and P. Loubaton, "Adaptive blind separation of independent sources: A deflation approach," *Signal Processing*, vol. 45, pp. 59–83, 1995.

[47] Y. Deville, "Analysis of the convergence properties of a self-normalized source separation neural network," in *Proc. IEEE Workshop Signal Processing Advances in Wireless Communications*, SPAWC-97, Paris, Apr. 1997, pp. 69–72.

[48] Z. Ding, C. R. Johnson, Jr., and R. A. Kennedy, "Global convergence issues with linear blind adaptive equalizers," in *Blind Deconvolution*, S. Haykin, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1994, pp. 60–120.

[49] S. C. Douglas, A. Cichocki, and S. Amari, "Fast-convergence filtered regressor algorithms for blind equalization," *Electron. Lett.*, vol. 32, no. 23, pp. 2114–2115, Dec. 1996.

[50] S. C. Douglas and A. Cichocki, "Neural networks for blind decorrelation of signals," *IEEE Trans. Signal Processing*, vol. 45, pp. 2829–2842, Nov. 1997.

[51] S. C. Douglas, A. Cichocki, and S. Amari, "Quasi-Newton filtered-error and filtered-regressor algorithms for adaptive equalization and deconvolution," in *Proc. IEEE Workshop Signal Processing and Advances in Wireless Communications*, Paris, Apr. 1997, pp. 109–112.

[52] ———, "Multichannel blind separation and deconvolution of sources with arbitrary distributions," in *Neural Networks for Signal Processing, Proc. 1997 IEEE Workshop (NNSP-97)*. Sept. 1997, pp. 436–445.

[53] M. Girolami and C. Fyfe, "Temporal model of linear anti-Hebbian learning," *Neural Processing Lett.*, vol. 4, no. 3, pp. 1–10, Jan. 1997.

[54] ———, "Stochastic ICA contrast maximization using Oja's nonlinear PCA algorithm," *Int. J. Neural Systems*, to be published.

[55] ———, "Extraction of independent signal sources using a deflationary exploratory projection pursuit network with lateral inhibition," in *Inst. Elect. Eng. Proc. Vision on, Image and Signal Processing J.*, 1997, vol. 14, pp. 299–306.

[56] ———, "Independence are far from normal," in *Proc. Euro. Symp. Artificial Neural Networks*, Bruges, Belgium, 1997, pp. 297–302.

[57] ———, "Kurtosis extrema and identification of independent components: A neural network approach," in *Proc. ICASSP-97*, Apr. 1997, vol. 4, pp. 3329–3333.

[58] V. P. Godambe, Ed., *Estimating Functions*. New York: Oxford, 1991.

[59] A. Gorokhov, P. Loubaton, and E. Moulines, "Second order blind equalization in multiple input multiple output FIR systems: A weighted least squares approach," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, GA, May 1996, vol. 5, pp. 2415–2418.

[60] M. I. Gurelli and C. L. Nikias, "EVAM: An eigenvector-based algorithm for multichannel bind deconvolution of input colored signals," *IEEE Trans. Signal Processing*, vol. 43, pp. 134–149, Jan. 1995.

[61] S. Haykin, Ed., *Blind Deconvolution*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[62] Y. Hua, "Fast maximum likelihood for blind identification of multiple FIR channels," *IEEE Trans. Signal Processing*, vol. 44, pp. 661–672, Mar. 1996.

[63] A. Hyvarinen and E. Oja, "One-unit learning rules for independent component analysis," in *Advances in Neural Information Processing System 9 (NIPS'96)*. Cambridge, MA: MIT Press, 1997, pp. 480–486.

[64] Y. Inouye and T. Sato, "On-line algorithms for blind deconvolution of multichannel linear time-invariant systems," in *Proc. IEEE Signal Processing Workshop on Higher-Order Statistics*, 1997, pp. 204–208.

[65] Y. Inouye, "Criteria for blind deconvolution of multichannel linear time-invariant systems of nonminimum phase," in *Statistical Methods in Control and Signal Processing*, T. Katayama and S. Sugimoto, Eds. New York: Dekker, 1997, pp. 375–397.

[66] Y. Inouye and K. Hirano, "Cumulant-based blind identification of linear multi-input multi-output systems driven by colored inputs," *IEEE Trans. Signal Processing*, vol. 45, pp. 1543–1552, June 1997.

[67] C. R. Johnson, Jr., "Admissibility in blind adaptive channel equalization," *IEEE Control Syst. Mag.*, vol. 11, pp. 3–15, Jan. 1991.

[68] C. Jutten and J. Herault, "Blind separation of sources—Part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, pp. 1–20, 1991.

[69] C. Jutten, H. L. Nguyen Thi, E. Dijkstra, E. Vittoz, and J. Caelen, "Blind separation of sources: An algorithm for separation of convolutive mixtures," in *Proc. Int. Workshop Higher Order Statistics*, Chamrousse, France, July 1991, pp. 275–278.

[70] J. Karhunen and J. Joutsensalo, "Learning of robust principal component subspace," in *Proc. IJCNN'93*, Nagoya, Japan, 1993, vol. 3, pp. 2409–2412.

[71] J. Karhunen, A. Cichocki, W. Kasprzak, and P. Pajunen, "On neural blind separation with noise suppression and redundancy reduction," *Int. J. Neural Syst.*, vol. 8, no. 2, pp. 219–237, Apr. 1997.

[72] J. Karhunen, "Neural approaches to independent component analysis and source separation," in *Proc. Euro. Sym. Artificial Neural Networks, ESANN-96*, Bruges, Belgium, Apr. 1996, pp. 249–266.

[73] J. Karhunen, L. Wang, and R. Vigario, "Nonlinear PCA type approaches for source separation and independent component analysis," in *Proc. ICNN'95*, Perth, Australia, Nov./Dec. 1995, pp. 995–1000.

[74] R. H. Lambert, "Multi-channel blind deconvolution: FIR matrix algebra and separation of multi-path mixtures," Ph.D. dissertation, Dept. Elect. Eng., Univ. of Southern California, Los Angeles, 1996.

[75] T. W. Lee, A. J. Bell, and R. Lambert, "Blind separation of delayed and convolved sources," in *Advances in Neural Information Processing Systems 9*. Cambridge, MA: MIT Press, 1997, pp. 758–764.

[76] R.-W. Liu and G. Dong, "Fundamental theorem for multiple-channel blind equalization," *IEEE Trans. Circuits Syst. I*, vol. 44, pp. 472–473, 1997.

[77] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. Cambridge, MA: MIT Press, 1983.

[78] S. Makeig, A. Bell, T.-P. Jung, and T. J. Sejnowski, "Independent component analysis in electro-encephalographic data," in *Advances in Neural Information Processing Systems 8*, M. Mozer *et al.*, Eds. Cambridge, MA: MIT Press, 1996, pp. 145–151.

[79] O. Macchi and E. Moreau, "Self-adaptive source separation—Part I: Convergence analysis of a direct linear network controlled by Herault–Jutten algorithm," *IEEE Trans. Signal Processing*, vol. 45, pp. 918–926, 1997.

[80] K. Matsuoka, M. Ohya, and M. Kawamoto, "A neural net for blind separation of nonstationary signals," *Neural Networks*, vol. 8, no. 3, pp. 411–419, 1995.

[81] E. Moreau and O. Macchi, "High order contrasts for self-adaptive source separation," *Int. J. Adaptive Contr. Signal Processing*, vol. 10, no. 1, pp. 19–46, Jan. 1996.

[82] N. Murata, K. Mueller, A. Ziehle, and S.-I. Amari, "Adaptive on-line learning in changing environments," *Advances in Neural Information Processing Systems 9*. Cambridge, MA: MIT Press, 1997, pp. 312–318.

[83] J. P. Nadal and N. Parga, "ICA: Conditions on cumulants and information theoretic approach," in *Proc. Euro. Symp. Artificial Neural Networks, ESANN'97*, Bruges, Apr. 16/17, 1997, pp. 285–290.

[84] H. L. Nguyen Thi and C. Jutten, "Blind source separation for convolved mixtures," *Signal Processing*, vol. 45, no. 2, pp. 209–229, 1995.

[85] E. Oja and J. Karhunen, "Signal separation by nonlinear Hebbian learning," in *Computational Intelligence—A Dynamic System Perspective*, M. Palaniswami *et al.*, Eds. New York: IEEE, Nov. 1995, pp. 83–97.

[86] E. Oja, "The nonlinear PCA learning rule in independent component analysis," *Neurocomputing*, vol. 17, pp. 25–45, 1997.

[87] E. Oja, J. Karhunen, L. Wang, and R. Vigario, "Principal and independent components in neural networks—Recent developments," in *Proc. 7th Italian Workshop on Neural Networks (WIRN-95)*, Vietri sul Mare, Italy, May 1995, pp. 20–26.

[88] C. B. Papadias and A. Paulraj, "A constant modulus algorithm for multi-user signal separation in presence of delay spread using antenna arrays," *IEEE Signal Processing Lett.*, vol. 4, pp. 178–181, June 1997.

[89] B. A. Pearlmutter and L. C. Parra, "A context-sensitive generalization of ICA," in *Proc. Int. Conf. Neural Information Processing, ICONIP'96,* Hong Kong, Sept. 24–27, 1996, pp. 151–156.

[90] D. T. Pham, P. Garrat, and C. Jutten, "Separation of a mixture of independent sources through a maximum likelihood approach," in *Proc. EUSIPCO'92,* 1992, pp. 771–774.

[91] J. C. Platt and F. Faggin, " Networks for the separation of sources that are superimposed and delayed," *NIPS,* vol. 4, pp. 730–737, 1992.

[92] J. Principe, C. Wang, and H. Wu, "Temporal decorrelation using teacher forcing anti-Hebbian learning and its applications in adaptive blind source separation," in *Proc. IEEE Workshop Neural Networks for Signal Processing,* Kyoto, Japan, 1996, pp. 413–422.

[93] O. Shalvi and E. Weinstein, "New criteria for blind deconvolution of nonminimum phase systems," *IEEE Trans. Inform. Theory,* vol. 36, pp. 312–321, Mar. 1990.

[94] ——, "Universal method for blind deconvolution," in *Blind Deconvolution,* S. Haykin, Ed. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1994.

[95] ——, "Super-exponential methods for blind deconvolution," *IEEE Trans. Signal Processing,* vol. 39, pp. 504–519, Mar. 1993.

[96] J. J. Shynk and R. P. Gooch, "The constant modulus array for co-channel signal copy and direction finding," *IEEE Trans Signal Processing,* vol. 44, pp. 652–660, Mar. 1996.

[97] D. T. M. Slock, "Blind joint equalization of multiple synchronous mobile users oversampling and/or multiple antennas," in *Proc. 28th Asimilar Conf.,* Pacific Grove, CA, Oct. 31–Nov. 2, 1994, pp. 1154–1158.

[98] H. Sompolinsky, N. Barkai, and H. S. Seung, "On-line learning of dichotomies: Algorithms and learning curves," in *Neural Networks: The Statistical Mechanics Perspective,* J.-H. Oh, C. Kwon, and S. Cho, Eds. Singapore: World Scientific, 1995, pp. 105–130.

[99] K. Torkkola, "Blind separation of convolved sources based on information maximization," in *Proc. IEEE Workshop Neural Networks and Signal Processing,* Kyoto, Japan, Sept. 1996, pp. 423–432.

[100] L. Tong, R.-W. Liu, V. C. Soon, and J. F. Huang, "Indeterminacy and identifiability of blind identification," *IEEE Trans. Circuits Syst.* vol. 38, pp. 409–509, 1991.

[101] L. Tong, Y. Inouye, and R.-W. Liu, "Waveform preserving blind estimation of multiple independent sources," *IEEE Trans. Signal Processing,* vol. 41, pp. 2461–2470, 1993.

[102] L. Tong, G. Xu, and T. Kailath, "Blind identification and equalization based on second-order statistics: A time domain approach," *IEEE Trans. Inform. Theory,* vol. IT-40, no. 2, pp. 340–349, Mar. 1994.

[103] J. R. Treichler and M. G. Larimore, "New processing techniques based on constant modulus adaptive algorithms," *IEEE Trans. Acoustic, Speech, Signal Processing,* vol. ASSP-33, no. 2, pp. 420–431, Apr. 1985.

[104] J. K. Tugnait, "Blind equalization and channel estimation for multiple-input multiple-output communications systems," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing,* Atlanta, GA, May 1996, vol. 5, pp. 2443–2446.

[105] S. Van Gerven, "Adaptive noise cancellation and signal separation with applications to speech enhancement," Ph.D. dissertation, Catholic University Leuven, Leuven, Belgium, Mar. 1996.

[106] A.-J. van der Veen, S. Talvar, and A. Paulraj, "A subspace approach to blind space-time signal processing for wireless communication systems," *IEEE Trans. Signal Processing,* vol. 45, pp. 173–190, Jan. 1997.

[107] E. Weinstein, M. Feder, and A. Oppenheim "Multichannel signal separation by decorrelation," *IEEE Trans. Speech Audio Processing,* vol. 1, no. 4, pp. 405–413, 1993.

[108] B. Widrow and E. Walach, *Adaptive Inverse Control.* Englewood-Cliffs, NJ: Prentice-Hall, 1996.

[109] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," *IEEE Trans. Signal Processing,* vol. 43, pp. 2982–2983, Dec. 1995.

[110] L. Xu, C.-C. Cheung, J. Ruan, and S. Amari, "Nonlinearity and separation capability: Further justification for the ICA algorithm with a learned mixture of parametric densities," in *Proc. Euro. Symp. Artificial Neural Networks, ESANN'97,* Bruges, Apr. 16/17, 1997, pp. 291–296.

[111] H. H. Yang and S. Amari, "Adaptive on-line learning algorithms for blind separation—Maximum entropy and minimum mutual information," *Neural Computation,* vol. 7, no. 9, pp. 1457–1482, 1997.

[112] H. H. Yang, S. Amari, and A. Cichocki, "Information-theoretic approach to blind separation of sources in nonlinear mixture," *Signal Processing,* vol. 64, pp. 291–300, 1998.

[113] D. Yellin and E. Weinstein, "Multichannel signal separation: Methods and analysis," *IEEE Trans. Signal Processing,* vol. 44, pp. 106–118, Jan. 1996.

Shun-ichi Amari (Fellow, IEEE) was born in Tokyo, Japan, on January 3, 1936. He graduated from the University of Tokyo in 1958, having majored in mathematical engineering, and he received the Dr.Eng. degree from the University of Tokyo in 1963.

He was an Associate Professor at Kyushu University, an Associate and then Full Professor at the Department of Mathematical Engineering and Information Physics, University of Tokyo, and is now Professor-Emeritus at the University of Tokyo. He is the Director of the Brain-Style Information Systems Group, RIKEN Brain Science Institute, Saitama, Japan. He has been engaged in research in wide areas of mathematical engineering and applied mathematics, such as topological network theory, differential geometry of continuum mechanics, pattern recognition, mathematical foundations of neural networks, and information geometry.

Dr. Amari served as President of the International Neural Network Society, Council member of Bernoulli Society for Mathematical Statistics and Probability Theory, and Vice President of the Institute of Electrical, Information and Communication Engineers. He was founding Coeditor-in-Chief of *Neural Networks*. He has been awarded the Japan Academy Award, the IEEE Neural Networks Pioneer Award, and the IEEE Emanuel R. Piore Award.

Andrzej Cichocki (Member, IEEE) was born in Poland. He received the M.Sc. (with honors), Ph.D., and Habilitate Doctorate (Dr.Sc.) degrees, all in electrical engineering and computer science, from Warsaw University of Technology, Poland, in 1972, 1975, and 1982, respectively.

Since 1972 he has been with the Institute of Theory of Electrical Engineering and Electrical Measurements at the Warsaw University of Technology, where he became a Full Professor in 1991. He spent a few years at the University Erlangen-Nuernberg as an Alexander Humboldt Research Fellow and Guest Professor. From 1995 to 1996 he worked as a Team Leader of the Laboratory for Artificial Brain Systems at the Frontier Research Program RIKEN, Saitama, Japan, in the Brain Information Processing Group. Currently he is Head of the Laboratory for Open Information Systems at the Brain Science Institute. His current research interests include signal and image processing (especially biomedical signal/image processing), neural networks and their electronic implementations, learning theory and algorithms, dynamic independent and principal component analysis, optimization problems, and nonlinear circuits and systems theory and applications. He is the co-author of two books: *MOS Switched-Capacitor and Continuous-Time Integrated Circuits and Systems* (Springer-Verlag, 1989) and *Neural Networks for Optimization and Signal Processing* (Wiley, 1993) and author or co-author of more than 150 scientific papers.

Dr. Cichocki is a reviewer of IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, *Biological Cybernetics*, *Electronics Letters*, *Neurocomputing*, and *Neural Computation*, and he is the Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS. He is a member of several international scientific committees.