INVITED PAPER

Blind Deconvolution of Dynamical Systems: A State-Space Approach

Liqing Zhang and Andrzej Cichocki

Laboratory for Open Information Systems,
Brain Science Institute, RIKEN
Wako-shi, Saitama 351-0198, Japan
E-mail: {zha/cia}@open.brain.riken.go.jp

Abstract **In this paper we present a general framework of the state space approach for blind deconvolution. First, we review the current state of the art of blind deconvolution using state-space models, then give a new insight into blind deconvolution in the state-space framework. The cost functions for blind deconvolution are discussed and adaptive learning algorithms for updating external parameters are developed by minimizing a certain cost function, which is derived from mutual information of output signals. The information backpropagation approach is developed for training the internal parameters. In order to compensate for the model bias and reduce the effect of noise, we introduce the Kalman filter to the blind deconvolution setting. A new concept, called hidden innovation, is introduced so as to numerically implement the Kalman filter. Thus we propose a new method: the two-stage approach to blind deconvolution. Finally we suggest how to extend the information backpropagation approach to the nonlinear case. Computer simulations are given to show the validity and effectiveness of the state-space approach.**

Keywords: independent component analysis, multichannel blind deconvolution, state-space models, unsupervised learning algorithms, nonlinear systems, information backpropagation

1. Introduction

Blind separation/deconvolution of source signals has been a subject under consideration for more than a decade [1]-[14]. There are significant potential applications of blind separation/deconvolution in various fields, such as wireless telecommunication systems, sonar and radar systems, audio and acoustics, image enhancement and biomedical signal processing (EEG/MEG signals) [15]-[25]. In those applications, several unknown but independent temporal signals propagate through a mixing and/or filtering, natural or synthetic medium. The blind source separation/deconvolution problem is to recover independent sources from sensor outputs without assuming any a priori knowledge of the original signals besides certain statistic features [6]-[8], [26].

Existing statistical blind deconvolution or equalization algorithms can generally be classified into two categories: the mutual information minimization/entropy maximization and the cumulant-based algorithms. The Bussgang algorithms [16]-[17], [27] and the natural gradient algorithm [9], [11], [28]-[31] are two typical examples of the first category. The Bussgang techniques are iterative equalization schemes that employ stochastic gradient descent procedures to minimize non-convex cost functions depending on the equalizer output signals. The Bussgang algorithms are simple and easy to implement. However, they might converge to wrong solutions which results in poor performance of the equalizer. The natural gradient approach was developed by S. Amari to overcome the drawback of the Bussgang algorithm [1, 28]. It has been proven that the natural gradient algorithm is an efficient algorithm in blind separation and blind deconvolution [1]. In the Cumulant Fitting Procedure (CFP) [32]-[34], the channel identification process directly employs the minimization of higher-order cumulant-based nonlinear cost functions. The underlying cost functions in CFP are multimodal, as in the case of Bussgang algorithms.

Although there exist a number of models and methods for separating blindly independent sources, such as the infomax, natural gradient approach and equivariant adaptive algorithms, there still exist several challenges in generalizing mixtures to dynamic and nonlinear systems, as well as in developing more rigorous and effective algorithms with general convergence. For example, in most practical applications the mixtures not only involve the instantaneous mixing but also delays or filtering of primary sources. The seismic data, the cocktail problem and biomedical data such as EEG signals are typical examples of such mixtures.

The state-space description of systems [35]-[37] is a new generalized model for blind separation and deconvolution. There are several reasons why the state-space models are advantageous for blind deconvolution. Although transfer function models are equivalent to the state-space ones in the linear case, it is difficult to exploit any common features that may be present in the real dynamic systems. The main advantage of the state space description for blind deconvolution is that it not only gives the internal description of a system, but there are various equivalent types of state-space realizations for a system, such as balanced realization and observable canonical forms. In particular, it is known how to parameterize some specific classes of models which are of interest in applications. In addition, it is easy to tackle the stability problem of state-space systems using the Kalman Filter. Moreover, the state-space model enables a much more general description than standard finite impulse response (FIR) convolutive filtering. All of the known filtering models, such as AR, MA, ARMA, ARMAX and Gamma filterings, could also be considered as special cases of flexible state-space models.

The state space formulation of blind source separation/deconvolution was discussed by Salam et al [38]-[40], Zhang et al [41]-[43] and Cichocki et al [44]-[47]. An efficient learning algorithm was developed by Zhang and Cichocki [41] to train the output matrices by minimizing the mutual information. In order to compensate for the model bias and reduce the effect of noise, a state estimator approach [43] was also proposed by using the Kalman filter. Cichocki et al extended the state space approach to nonlinear system [44], and an effective two-stage learning algorithm was presented [45] for training the parameters in demixing models.

In this paper we present a general framework of state space approach for multichannel blind deconvolution of both linear and nonlinear systems. First we give a general formulation of blind deconvolution in the state space model. We discuss some theoretical problems, such as recoverability, and cost function for blind deconvolution in the state space framework. Then we present two approaches, the information backpropagation and the two-stage approaches. The two approaches are developed for different purposes. The information backpropagation algorithm is suitable for the blind deconvolution of minimum phase systems. However, in the non-minimum phase systems, it is not easy to accomplish the blind deconvolution in one step, since the delays in different recovered channels of the demixing models are unknown [48].

In this paper, we divide the parameters in demixing models into two types: The internal parameters and external parameters. They are trained in different ways. The internal parameters are independent of the individual signal separation problems; they are usually trained in an off-line manner, according to a set of signal separation problems. In contrast, the external parameters are trained individually for each separation problem.

2. General Formulation

Assume that unknown source signals $\mathbf{s}(t) = (s_1(t), \cdots, s_n(t))^T$ are stationary zero-mean i.i.d processes and mutually statistically independent. Suppose that the unknown source signals $\mathbf{s}(k)$ are mixed by a stable nonlinear dynamic system

$$\begin{aligned} \overline{\mathbf{x}}(k+1) &= \overline{\boldsymbol{\mathcal{F}}}(\overline{\mathbf{x}}(k), \mathbf{s}(k), \boldsymbol{\xi}_P(k)) & (1) \\ \mathbf{u}(k) &= \overline{\mathbf{G}}(\overline{\mathbf{x}}(k), \mathbf{s}(k)) + \boldsymbol{\theta}(k) & (2) \end{aligned}$$

where $\overline{\boldsymbol{\mathcal{F}}}$ and $\overline{\mathbf{G}}$ are two unknown nonlinear mappings, $\overline{\mathbf{x}}(k) \in \mathbf{R}^N$ is the state vector of the system, and $\mathbf{u}(k) \in \mathbf{R}^n$ is the vector of sensor signals, which are available to signal processing. $\boldsymbol{\xi}_P(k)$ and $\boldsymbol{\theta}(k)$ are the process noises and sensor noises of the mixing system, respectively. In this paper, we present another dynamic system as a demixing model

$$\begin{aligned} \mathbf{x}(k+1) &= \boldsymbol{\mathcal{F}}_N(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta}) & (3) \\ \mathbf{y}(k) &= \mathbf{G}(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Phi}) & (4) \end{aligned}$$

where $\mathbf{u}(k) \in \mathbf{R}^n$ is the vector of sensor signals, $\mathbf{x}(k) \in \mathbf{R}^M$ is the state vector of the system, $\mathbf{y}(k) \in \mathbf{R}^n$ is designated to recover source signals in certain sense, $\boldsymbol{\mathcal{F}}_N$ is a nonlinear mapping, described by a general nonlinear capability neural network, $\boldsymbol{\Theta}$ is the set of parameters (synaptic weights) of the neural network, $\mathbf{G}$ is a nonlinear mapping with non-singularity of derivative $\frac{\partial \mathbf{G}}{\partial \mathbf{u}}$, and $\boldsymbol{\Phi}$ is the weights of $\mathbf{G}$. The dimension M of the state vector is the order of the demixing system.

Since the mixing system is blind, we neither know the nonlinear mappings $\overline{\boldsymbol{\mathcal{F}}}$ and $\overline{\mathbf{G}}$, nor the dimension N of the state vector $\overline{\mathbf{x}}(k)$. We need to estimate the order and approximate nonlinear mappings of the demixing system. In the blind deconvolution, the dimension M is difficult to determine and is usually overesitimated, i.e. $M > N$. The overestimation of the order M may produce delays in output signals, but this is acceptable in blind deconvolution. There are a number of neural networks such as Radial Based Function, Support Vector Machine and multilayer perceptron, which can be used as demixing models. In this paper, we employ the Support Vector Machine to estimate the nonlinear mapping $\boldsymbol{\mathcal{F}}_N(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta})$ in the demixing model.

If both of the mappings, $\boldsymbol{\mathcal{F}}_N$ and $\mathbf{G}$ are linear, the nonlinear state space model will reduce to the standard multichannel blind deconvolution. In this paper, we first discuss the linear case, then extend the algorithms to the nonlinear blind deconvolution. For simplicity, we will discuss the nonlinear models of the following form,

$$\begin{aligned} \mathbf{x}(k+1) &= \boldsymbol{\mathcal{F}}_N(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta}) & (5) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) & (6) \end{aligned}$$

In this demixing model, the output equation is assumed to be linear. The restriction is reasonable since in many practical problems, the measurement is a linear combination of certain variables. The simplification of the demixing model is simply for clarity of presentation and for easier derivation of learning algorithms. One can extend the results to the general case.

2.1 Internal Representation

The state space description [35]-[37] allows us to divide the variables into two types: the internal state variable $\mathbf{x}(k)$, which produces the dynamics of the system, and the external variables $\mathbf{u}(k)$ and $\mathbf{y}(k)$, which represent the input and output of the system, respectively. The vector $\mathbf{x}(k)$ is known as the state of the dynamic system, which summarizes all the information about the past behavior of the system that is needed to uniquely predict its future behavior, except for the purely external input $\mathbf{u}(k)$. The term state plays a critical role in mathematical formulation of a dynamical system. It allows us to realize the internal structure of the system and to define the controllability and observability of the system as well. In the state space framework, it becomes much easier to discuss the stability, controllability and observability of dynamical systems.

We formulate the demixing model in the framework of the state space models for blind deconvolution. The parameters in the state equation of the demixture are referred to as internal representation parameters (or simply internal parameters), and the parameters in the output equation as external ones. Such a distinction enables us to train the demixing model in two stages: internal representation and output separation. In the internal representation stage, we will make the state space as sparse as possible such that the output signals can be represented as a linear combination of the state vector $\mathbf{x}(k)$ and input vector $\mathbf{u}(k)$. In this paper, we will employ two approaches to train the internal parameters and to estimate the state vector by using information backpropagation and Kalman filtering.

In the state space framework, we suggest that the separation of sources should be made in two stages: the first one involves training the state space parameters, such that the system is of sparse information representation, the second one involves fixing the internal parameters and training the external parameters by blind deconvolution algorithms.

2.2 Linear System Case

In order to illustrate the flexibility of the state-space model for blind deconvolution, we elaborate the mixing model and demixing model within the framework of linear state-space representation. Suppose that the mixing model is described by a linear state discrete-time system

$$\overline{\mathbf{x}}(k+1) = \overline{\mathbf{A}}\overline{\mathbf{x}}(k) + \overline{\mathbf{B}}\mathbf{s}(k) + \overline{\mathbf{L}}\boldsymbol{\xi}_P(k) \quad (7)$$
$$\mathbf{u}(k) = \overline{\mathbf{C}}\overline{\mathbf{x}}(k) + \overline{\mathbf{D}}\mathbf{s}(k) + \boldsymbol{\theta}(k) \quad (8)$$

where $\overline{\mathbf{x}} \in \mathbf{R}^N$ is the state vector of the system, $\mathbf{s}(k) \in \mathbf{R}^n$ is the vector of input signals, $\mathbf{u}(k) \in \mathbf{R}^m$ is the vector of sensor signals, $\overline{\mathbf{A}} \in \mathbf{R}^{N\times N}$ is the state mixing matrix, $\overline{\mathbf{B}} \in \mathbf{R}^{N\times n}$ is an input mixing matrix, $\overline{\mathbf{C}} \in \mathbf{R}^{m\times N}$ is the output mixing matrix and $\overline{\mathbf{D}} \in \mathbf{R}^{m\times n}$ is the input-output mixing matrix. If we ignore the noise terms in the mixing model, its transfer function matrix is described by an $m \times n$ matrix of the form

$$\mathbf{H}(z) = \overline{\mathbf{C}}(z\mathbf{I} - \overline{\mathbf{A}})^{-1}\overline{\mathbf{B}} + \overline{\mathbf{D}} \quad (9)$$

where z^{-1} is a time delay operator.

We formulate the blind separation problem as a task to recover original signals from observations $\mathbf{u}(k)$ without prior knowledge on the source signals or state space matrices $[\overline{\mathbf{A}}, \overline{\mathbf{B}}, \overline{\mathbf{C}}, \overline{\mathbf{D}}]$ besides certain statistical features of the source signals. Since the mixing model is a linear state-space system, we propose that the demixing model here is another linear state-space system, which is described as follows, (see Fig. 1)

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}\boldsymbol{\xi}_R(k) \quad (10)$$
$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (11)$$

where $\boldsymbol{\theta} = [\mathbf{A}, \mathbf{B}, \mathbf{L}]$, the input $\mathbf{u}(k)$ of the demixing model is simply the output (sensor signals)

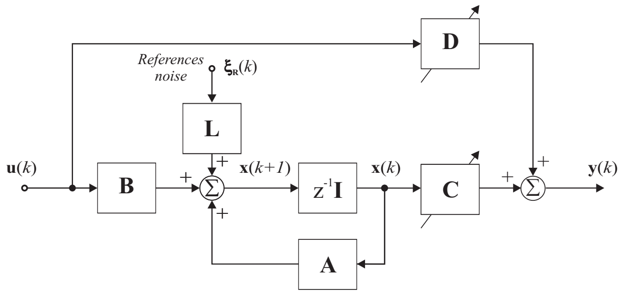

Fig. 1 General linear state-space model for blind deconvolution

of the mixing model and the $\boldsymbol{\xi}_R(k)$ is the reference model noise. Generally, the set of matrices $[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{L}]$ are parameters to be determined in a learning process [41]-[47]. For simplicity, we assume that the noise terms both in the mixing and demixing models are negligibly small. The transfer function of the demixing model is $\mathbf{W}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$. The output $\mathbf{y}(k)$ is designed to recover the source signals in the following sense

$$\mathbf{y}(k) = \mathbf{W}(z)\mathbf{H}(z)\mathbf{s}(k) = \mathbf{P}\boldsymbol{\Lambda}(z)\mathbf{s}(k) \quad (12)$$

where $\mathbf{P}$ is any permutation matrix and $\boldsymbol{\Lambda}(z)$ is a diagonal matrix with $\lambda_i z^{-\tau_i}$ in diagonal entry (i,i), here λ_i is a nonzero constant and τ_i is any nonnegative integer. The recoverability of the linear demixing model and nonlinear demixing model will be discussed in the next section.

It is easy to see that the linear state space model mixture is an extension of the instantaneous mixture. When the matrices $\overline{\mathbf{A}}, \overline{\mathbf{B}}, \overline{\mathbf{C}}$ in the mixing model and $\mathbf{A}, \mathbf{B}, \mathbf{C}$ in the demixing model are null matrices, the problem is simplified to the standard independent component analysis (ICA) problem [6], [8], [9].

2.3 Canonical Forms

Canonical forms for linear systems are of great importance since they provide a unique state space representation of linear systems [35]-[37]. Therefore they play a major role in system identification where a unique parameterization of the systems in the model set in essential to avoid identifiability problems. Various types of canonical forms for linear systems have been introduced and studied. Most of these canonical forms for multi-variable systems are generalizations of the observer or controller canonical form for single-input single-output systems. The usefulness of a canonical form depends on its properties. One of the standard canonical forms, the controller canonical form, is of particular significance since the parameters of the canonical form have a direct interpretation as their coefficients of the transfer function. Moreover, this canonical form permits a straightforward proof of the pole-shift theorem [37]. There are, however drawbacks of the controller canonical forms, particularly concerning the resulting parameterization of linear systems. The set of parameters in the controller form that lead to a minimal realization is very complicated. Fortunately, in blind deconvolution, the minimal realization of the demixing system is neither necessary nor practical. Another canonical form is the balanced canonical form, which has a geometrically well-behavior parameter space. One of the main advantages of the balanced canonical form is its geometric properties, for example, the stable minimum phase system can be represented in the Lyapunov Balanced Canonical Form.

Controller Canonical Form
If the transfer function of a system is given by

$$\mathbf{H}(z) = \mathbf{P}(z)\mathbf{Q}^{-1}(z) \tag{13}$$

where $\mathbf{P}(z) = \sum_{i=0}^{N} \mathbf{P}_i z^{-i}$ and $\mathbf{Q}(z) = \sum_{i=0}^{N} \mathbf{Q}_i z^{-i}$, $\mathbf{Q}_0 = \mathbf{I}$.

The matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ for the canonical controller form are represented as follows.

$$\mathbf{A} = \begin{bmatrix} -\boldsymbol{\mathcal{Q}} & -\mathbf{Q}_N \\ \mathbf{I}_{n(N-1)} & \boldsymbol{\mathcal{O}} \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} \mathbf{I}_n \\ \boldsymbol{\mathcal{O}} \end{bmatrix} \tag{14}$$

$$\mathbf{C} = (\mathbf{P}_1\ \mathbf{P}_2\ \cdots\ \mathbf{P}_N), \quad \mathbf{D} = \mathbf{P}_0 \tag{15}$$

where $\boldsymbol{\mathcal{Q}} = (\mathbf{Q}_1\ \mathbf{Q}_2,\ \cdots,\ \mathbf{Q}_{N-1})$ is an $n \times n(N-1)$ matrix, $\boldsymbol{\mathcal{O}}$ is an $n(N-1) \times n$ null matrix, $\mathbf{I}_n$ and $\mathbf{I}_{n(N-1)}$ are the $n \times n$ and $n(N-1) \times n(N-1)$ identity matrices, respectively. In particular, if the system is FIR, i.e. $\mathbf{H}(z) = \mathbf{P}(z)$, both $\mathbf{A}$ and $\mathbf{B}$ are constant matrices.

Lyapunov Balanced Canonical Form
The system $\mathbf{W} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ is called Lyapunov-balanced if the positive definite solutions $\mathbf{Y}$ and $\mathbf{Z}$ to the Lyapunov equations,

$$\mathbf{A}\mathbf{Z} + \mathbf{Z}\mathbf{A}^* + \mathbf{B}\mathbf{B}^* = \mathbf{0} \tag{16}$$

$$\mathbf{A}^*\mathbf{Y} + \mathbf{Y}\mathbf{A} + \mathbf{C}^*\mathbf{C} = \mathbf{0} \tag{17}$$

are such that

$$\mathbf{Y} = \mathbf{Z} = diag(\sigma_1, \sigma_2, \cdots, \sigma_n) := \mathbf{\Sigma}_n \tag{18}$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$. The nonsingular diagonal matrix $\mathbf{\Sigma}_s$ is called the Lyapunov-grammian of the system. The positive numbers $\sigma_1, \sigma_2, \cdots, \sigma_n$ are called the Lyapunov -singular values of the system. The canonical form of a stable minimal system will be Lyapunov balanced. This is the main advantage of the balanced canonical form. Such a parameterization for stable minimum phase systems is of importance in time series analysis, where the innovation of state space model is made based an observed time series. However, a disadvantage of the balanced parameterization is that it does not include an atlas for the manifold of systems. This leads to a new parameterization: An overlapping parameterization. Refer to [37] for the detailed realization of balanced canonical forms.

3. Cost Functions for Blind Deconvolution

Our objective is to train the demixing model such that the output signals are as spatially mutually independent and temporarily i.i.d. as possible. There are a number of cost functions which fulfill blind deconvolution task. Commonly used cost functions include maximum entropy, minimum mutual information and maximum high order cumulants.

3.1 Maximum Entropy

Assume that $p(\mathbf{y}, \mathbf{W})$ and $p_i(y_i, \mathbf{W})$ are the joint probability density function of $\mathbf{y}$ and marginal pdf of y_i, $(i = 1, \cdots, n)$ respectively. The differential entropy of the output of the demixing model is defined by

$$H(\mathbf{y}, \mathbf{W}) = -\int p(\mathbf{y}, \mathbf{W}) \log p(\mathbf{y}, \mathbf{W}) d\mathbf{y} \tag{19}$$

The entropy $H(\mathbf{y}, \mathbf{W})$ is generally not upper bounded. This means that the maximum of $H(\mathbf{y}, \mathbf{W})$ may not exist. In order to overcome this difficulty, Bell and Sejnowski proposed that after using a component nonlinear transformation on $\mathbf{y}$, the entropy will become upper bounded and the maximum of entropy will always exist. Refer to [8] for a detailed discussion.

3.2 Minimum Mutual Information

Another cost function for blind deconvolution is the Kullback-Leibler Divergence which defines an asymmetrical measure of two probability functions [49], [50]. Let $p_{\mathbf{y}}(\mathbf{y})$ and $q_{\mathbf{y}}(\mathbf{y})$ denote two different probability density functions. We then define the Kullback-Leibler Divergence between $p_{\mathbf{y}}(\mathbf{y})$ and $q_{\mathbf{y}}(\mathbf{y})$ as follows:

$$\mathcal{D}(p,q) = \int p_{\mathbf{y}}(y) \log\left(\frac{p_{\mathbf{y}}(\mathbf{y})}{q_{\mathbf{y}}(\mathbf{y})}\right) d\mathbf{y} \tag{20}$$

The Kullback-Leibler Divergence can be presented in the context of differential geometry as the Riemannian metric in the space of the distributions [49], [50]. The expression (20) can be interpreted as a quasi-distance because it is always non-negative and is equal to zero if and only if $p_{\mathbf{y}}(\mathbf{y}) = q_{\mathbf{y}}(\mathbf{y})$. Because of this property, we can also use the Kullback-Leibler Divergence to measure the mutual independence of output signals $\mathbf{y}$. Let $p_i(y_i)$ be the i-th marginal probability density function of component y_i, which is defined by

$$p_i(y_i) = \int p_{\mathbf{y}}(\mathbf{y}) d\mathbf{y}^{(i)},\ i = 1, \cdots, n \tag{21}$$

where $\mathbf{y}^{(i)}$ is the $(n-1)$-dimensional vector after removing the $i-$th element from vector $\mathbf{y}$. If the output $\mathbf{y}$ is spatially mutually independent, then

$$p_{\mathbf{y}}(\mathbf{y}) = \prod_{i=1}^{n} p_i(y_i) \tag{22}$$

The Kullback-Leibler Divergence between $p_{\mathbf{y}}(\mathbf{y})$ and $q_{\mathbf{y}}(\mathbf{y}) = \prod_{i=1}^{n} p_{y_i}(y_i)$ is given by

$$\mathcal{D}(p,q) = \int p_{\mathbf{y}}(y) \log\left(\frac{p_{\mathbf{y}}(\mathbf{y})}{\prod_{i=1}^{n} p_i(y_i)}\right) d\mathbf{y} \tag{23}$$

or we rewrite it into the mutual information form

$$l(\mathbf{W}) = -H(\mathbf{y}, \mathbf{W}) + \sum_{i=1}^{n} H(y_i, \mathbf{W}) \tag{24}$$

where

$$\begin{aligned} H(\mathbf{y}, \mathbf{W}) &= -\int p(\mathbf{y}, \mathbf{W}) \log p(\mathbf{y}, \mathbf{W}) d\mathbf{y} \\ H(y_i, \mathbf{W}) &= -\int p_i(y_i) \log p_i(y_i) dy_i \end{aligned}$$

The divergence $l(\mathbf{W})$ is a nonnegative functional, which measures the mutual independence of the output signals $y_i(k)$. The output signals $\mathbf{y}$ are mutually independent if and only if $l(\mathbf{W}) = 0$. Therefore, the Kullback-Leibler Divergence $\mathcal{D}(p, \prod_{i=1}^{n} p_i(y_i))$ can be used as a cost function for blind deconvolution. However, there are several unknowns in the cost function: the joint probability density function $p_{\mathbf{y}}(\mathbf{y})$ and the marginal probability density functions $p_i(y_i)$.

As shown in the Appendix 13.2, we can express the entropy $H(\mathbf{y})$ as

$$H(\mathbf{y}) = -\log|det(\mathbf{D})| + const \tag{25}$$

The cost function derived from the mutual information can be simplified as

$$l(\mathbf{y}, \mathbf{W}) = -\log|det(\mathbf{D})| - \sum_{i=1}^{n} H(y_i, \mathbf{W}) \tag{26}$$

In order to implement the statistical on-line learning, we reformulate the cost function as

$$l(\mathbf{y}, \mathbf{W}) = -\log|det(\mathbf{D})| - \sum_{i=1}^{n} \log q(y_i) \tag{27}$$

where $q(y_i)$ is an estimation of the true probability density function of source signals. There are a number of methods, such as the Edgeworth series [51] and Gram-Charlier series [9], to estimate the probability density function $p(y_i)$ of y_i. The Gram-Charlier series is an expansion of a probability density function at the neighborhood of the normalized Gaussian distribution $\phi(y_i) = \frac{1}{\sqrt{2\pi}} e^{-y_i^2/2}$,

$$p_i(y_i) = \phi(y_i) \left[1 + \sum_{j=3}^{\infty} c_j H_j(y_i)\right] \tag{28}$$

where $H_j(y_i)$ are Hermite polynomials, and the coefficients c_j are defined in terms of the cumulants of y_i. Refer to [9] and [52] for further information.

3.3 High Order Cumulants

Alternative function for blind deconvolution is the high order statistics of the output [53]. High order information can be expressed by using cumulants. Denote $\mathcal{C}_r(y_i)$ the cumulant of order r of the random variable y_i. Then we can define a cost function as follows

$$l(\mathbf{y}, \mathbf{W}) = \sum_{i=1}^{n} |\mathcal{C}_r(y_i)| \tag{29}$$

Usually such a cost function has a disadvantage in that we have to perform the whitening procedure first, and then maximize the above cost function.

4. Invertiability by State Space Model

Assume that the number of sensor signals equals the number of source signals, i.e. $m = n$. In the following discussion, we restrict the mixing model to the following form,

$$\mathbf{x}(k+1) = \overline{\mathcal{F}}(\mathbf{x}(k), \mathbf{s}(k)) \quad (30)$$
$$\mathbf{u}(k) = \overline{\mathbf{C}}\mathbf{x}(k) + \overline{\mathbf{D}}\mathbf{s}(k) \quad (31)$$

where the state equation is a nonlinear dynamic system, and the output equation is a linear one. From a theoretical point of view, we can easily find the inverse of the state space models in the same form, if the matrix $\overline{\mathbf{D}}$ is invertible. In fact, the inverse system is expressed by

$$\mathbf{x}(k+1) = \overline{\mathcal{F}}(\mathbf{x}(k), \overline{\mathbf{D}}^{-1}(\mathbf{y}(k) - \overline{\mathbf{C}}\mathbf{x}(k))) \quad (32)$$

$$\mathbf{s}(k) = \overline{\mathbf{D}}^{-1}(\mathbf{u}(k) - \overline{\mathbf{C}}\mathbf{x}(k)) \quad (33)$$

This means that if the mixing model is expressed by (30) and (31), we can recover the source signals using the inverse system (32) and (33). There is an advantage to the state space model in that we do not need to inverse any nonlinear functions explicitly.

4.1 Linear Case

Now we concentrate on the linear system and discuss the recoverability and representation of the inverse system. From the definition, we have

$$\mathbf{u}(k) = \mathbf{H}(z)\mathbf{s}(k) \quad (34)$$
$$\mathbf{y}(k) = \mathbf{W}(z)\mathbf{u}(k) \quad (35)$$

The independent sources are recoverable from the demixing model (10) and (11) if and only if

$$\mathbf{W}(z)\mathbf{H}(z) = \mathbf{P}\mathbf{\Lambda}(z) \quad (36)$$

where $\mathbf{P}$ and $\mathbf{\Lambda}(z)$ are defined as in (12). The question here is whether matrices $[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$ exist in the demixing model (10) and (11), such that its transfer function $\mathbf{W}(z)$ satisfies (36). The answer is affirmative. If the matrix $\overline{\mathbf{D}}$ in the mixing model satisfies $rank(\overline{\mathbf{D}}) = n$, and $\mathbf{W}_0(z)$ is the inverse of $\mathbf{H}(z)$, which is defined in the Appendix, then any state-space realization $[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$ of a new transfer function $\mathbf{W}(z) = \mathbf{P}\mathbf{\Lambda}(z)\mathbf{W}_0(z)$ meets equation (11). Therefore, we have the following theorem:

Theorem 1 *If the matrix $\overline{\mathbf{D}}$ in the mixing model is of full rank, i.e. $rank(\overline{\mathbf{D}}) = n$, then there exist matrices $[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$, such that the output signals $\mathbf{y}$ of state-space system (4) and (5) recovers the independent source signals in the sense of (6).*

In blind deconvolution problems, we do not know matrices $[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$, or the state space dimension M. Before we train the matrices $[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$ in the state-space model, we must estimate the dimension M of the system if one needs to obtain a canonical solution. There are several criteria for estimating the dimension of a system in system identification, such as AIC and FPE criteria. The order estimation problem in blind deconvolution is quite difficult, but interesting. It remains an open problem that is not discussed in this paper. It should be noted that if the dimension of state vector in the demixing model is overestimated, i.e., it is larger than that in the mixing model, the separated signals may contain some auxiliary time-delays, which are acceptable in blind deconvolution.

5. Basic Learning Algorithms

In this section, we develop learning algorithms to update the external parameters $\mathbf{W} = [\mathbf{C}, \mathbf{D}]$ in the demixing model. The basic idea is to use the gradient descent approach to update these parameters. In order to obtain an improved learning performance, we define a new search direction, which is related to the natural gradient, developed by Amari [49] and Amari and Nagaoko [50].

5.1 Gradient Descent Algorithm

For simplicity we suppose that the matrix $\mathbf{D}$ in the demixing model (11) is a nonsingular $n \times n$ matrix. The cost function for learning is derived from mutual information as

$$l(\mathbf{y}, \mathbf{W}) = -\log|det(\mathbf{D})| - \sum_{i=1}^{n} \log q_i(y_i) \quad (37)$$

where $det(\mathbf{D})$ is the determinant of the matrix $\mathbf{D}$. $q_i(y_i)$ is an approximation of pdf $p_i(y_i)$. For the gradient of the cost function $l(\mathbf{y}, \mathbf{W})$ with respect to $\mathbf{W}$, we calculate the total differential $dl(\mathbf{y}, \mathbf{W})$ of $l(\mathbf{y}, \mathbf{W})$ when we take a differential $d\mathbf{W}$ on $\mathbf{W}$,

$$dl(\mathbf{y}, \mathbf{W}) = l(\mathbf{y}, \mathbf{W} + d\mathbf{W}) - l(\mathbf{y}, \mathbf{W}) \qquad (38)$$

Following Amari's derivation for the natural gradient method [2, 31], we have

$$dl(\mathbf{y}, \mathbf{W}) = -tr(d\mathbf{D}\mathbf{D}^{-1}) + \boldsymbol{\varphi}^T(\mathbf{y})d\mathbf{y} \qquad (39)$$

where tr is the trace of a matrix and $\boldsymbol{\varphi}(\mathbf{y})$ is a vector of nonlinear activation functions

$$\varphi_i(y_i) = -\frac{d \log q_i(y_i)}{dy_i} = -\frac{q_i'(y_i)}{q_i(y_i)} \qquad (40)$$

Taking a differential of $\mathbf{y}$ in equation (11), we have the following relation

$$d\mathbf{y} = d\mathbf{C}\mathbf{x}(k) + d\mathbf{D}\mathbf{u}(k) + \mathbf{C}d\mathbf{x}(k) \qquad (41)$$

Hence, we obtain the derivatives

$$\begin{aligned} \frac{\partial l(\mathbf{y}, \mathbf{W})}{\partial \mathbf{C}} &= \boldsymbol{\varphi}(\mathbf{y})\mathbf{x}^T & (42) \\ \frac{\partial l(\mathbf{y}, \mathbf{W})}{\partial \mathbf{D}} &= -\mathbf{D}^{-T} + \boldsymbol{\varphi}(\mathbf{y})\mathbf{u}^T & (43) \end{aligned}$$

Finally the gradient descent learning algorithm is described by

$$\begin{aligned} \Delta\mathbf{C}(k) &= -\eta(k)\boldsymbol{\varphi}(\mathbf{y})\mathbf{x}^T & (44) \\ \Delta\mathbf{D}(k) &= \eta(k)(\mathbf{D}^{-T} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{u}^T) & (45) \end{aligned}$$

where $\eta(k)$ is the learning rate. In this algorithm, we must calculate the inverse matrix $\mathbf{D}^{-1}$. In order to reduce the computing cost and improve the learning efficiency, in [41] we employed the natural gradient technique by multiplying a positive definite matrix $\mathbf{D}^T\mathbf{D}$ in the learning algorithm for $\mathbf{D}$. The modified learning rule for $\mathbf{D}$ is described by

$$\Delta\mathbf{D}(k) = \eta(k)(\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{u}^T\mathbf{D}^T)\mathbf{D} \qquad (46)$$

The modification reduces the computing cost, but does not significantly improve the learning efficiency because the modified search direction is not the natural gradient one. The algorithm includes an unknown score function $\boldsymbol{\varphi}(\mathbf{y})$. The optimal one is given by equation (40) with $q_i(y_i) = p_i(y_i)$, if we can estimate the true source probability distribution $p_i(y_i)$ adaptively. Another solution is to give a score function according to the statistics of source signals. Typically if a source signal y_i is super-Gaussian, one can choose $\varphi_i(y_i) = tanh(y_i)$. Respectively, if it is sub-Gaussian, one can choose $\varphi_i(y_i) = y_i^3$ [54, 55]. A question will be raised as to whether the learning algorithm will converge to a true solution if the approximation score functions are used. The theory of the semi-parametric model for blind separation/deconvolution ([26], [56], [57], [58]) shows that even through a misspecified pdf is used in learning, learning algorithms can still converge to the true solution if certain stability conditions are satisfied [54].

5.2 Natural Gradient Algorithm

Stochastic gradient optimization methods for parameterized systems suffer from slow convergence due to the statistical correlation of the processes signals. While quasi-Newton and related methods can be used to improve convergence, they also suffer from heavy computation and numerical instability, as well as local convergence problems.

The natural gradient search scheme proposed by Amari [1], [49] is an efficient technique for solving iterative estimation problems. For a cost function $l(\mathbf{y}, \mathbf{W})$, the natural gradient $\tilde{\nabla} l(\mathbf{y}, \mathbf{W})$ is the steepest ascent direction of the cost function $l(\mathbf{y}, \mathbf{W})$. In this paper we derive the extended natural gradient by introducing a new search direction.

From linear output equation (6), we have

$$\mathbf{u}(k) = \mathbf{D}^{-1}(\mathbf{y}(k) - \mathbf{C}\mathbf{x}(k)) \qquad (47)$$

Substituting (47) into (41), we obtain

$$d\mathbf{y} = (d\mathbf{C} - d\mathbf{D}\mathbf{D}^{-1}\mathbf{C})\mathbf{x} + d\mathbf{D}\mathbf{D}^{-1}\mathbf{y} \qquad (48)$$

In order to improve the computing efficiency of learning algorithms, we introduce a new search direction defined as

$$\begin{aligned} d\mathbf{X}_1 &= d\mathbf{C} - d\mathbf{D}\mathbf{D}^{-1}\mathbf{C} & (49) \\ d\mathbf{X}_2 &= d\mathbf{D}\mathbf{D}^{-1} & (50) \end{aligned}$$

It is easy to obtain the derivatives of the loss function l with respect to matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ as

$$\begin{aligned} \frac{\partial l(\mathbf{y}, \mathbf{W})}{\partial \mathbf{X}_1} &= \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{x}^T(k) & (51) \\ \frac{\partial l(\mathbf{y}, \mathbf{W})}{\partial \mathbf{X}_2} &= \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k) - \mathbf{I} & (52) \end{aligned}$$

Using the standard gradient descent method, we deduce a learning rule for $\mathbf{X}_1$ and $\mathbf{X}_2$

$$\begin{aligned} \Delta\mathbf{X}_1(k) &= -\eta\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{x}^T(k) \quad (53)\\ \Delta\mathbf{X}_2(k) &= -\eta(\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k) - \mathbf{I}) \quad (54)\end{aligned}$$

where η is a learning rate. From (49) and (50), we obtain a novel learning algorithm to update matrices $\mathbf{C}$ and $\mathbf{D}$ as

$$\Delta\mathbf{C}(k) = \eta\left((\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T)\mathbf{C} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{x}^T\right) \quad (55)$$

$$\Delta\mathbf{D}(k) = \eta\left(\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T\right)\mathbf{D}(k) \quad (56)$$

In fact, the relation between the natural gradient and the ordinary gradient can be defined by

$$\tilde{\nabla} l = \nabla l \left[\begin{array}{cc} \mathbf{I} + \mathbf{C}^T\mathbf{C} & \mathbf{C}^T\mathbf{D} \\ \mathbf{D}^T\mathbf{C} & \mathbf{D}^T\mathbf{D}\end{array}\right] \quad (57)$$

where $\nabla l = \left[\frac{\partial l(\mathbf{y},\mathbf{W})}{\partial \mathbf{C}} \ \frac{\partial l(\mathbf{y},\mathbf{W})}{\partial \mathbf{D}}\right]$. Therefore, the learning algorithm can be rewritten equivalently in the following form

$$[\Delta\mathbf{C} \ \Delta\mathbf{D}] = -\eta(k)\tilde{\nabla} l(\mathbf{y}, \mathbf{W}). \quad (58)$$

It is easy to see that the preconditioning matrix

$$\left[\begin{array}{cc} \mathbf{I} + \mathbf{C}^T\mathbf{C} & \mathbf{C}^T\mathbf{D} \\ \mathbf{D}^T\mathbf{C} & \mathbf{D}^T\mathbf{D}\end{array}\right]$$

is symmetric positive definite, and this expression is the extension of Amari's natural gradient to the state space model.

The natural gradient provides a form by which the analysis of stability becomes much easier. The equilibrium points of the learning algorithm satisfy the following equations

$$E[\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{x}^T(k)] = 0 \quad (59)$$

$$E\left[\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k)\right] = 0 \quad (60)$$

This means that separated signals $\mathbf{y}$ could achieve a mutual independence as high as possible if the nonlinear activation function $\boldsymbol{\varphi}(\mathbf{y})$ is suitably chosen. From (55) and (56), we see that the natural gradient learning algorithm [9] is covered as a special case of the learning algorithm when the mixture is simplified to an instantaneous case.

On the other hand, if the output signals $\mathbf{y}$ of (11) are spatially mutually independent and temporarily i.i.d. signals, it is easy to verify that $\mathbf{y}(k)$ satisfies (59) and (60). In fact, from (10) and (11) we have

$$\mathbf{x}(k+1) = \sum_{l=0}^{k-1} \tilde{\mathbf{B}}\,\tilde{\mathbf{A}}^l\mathbf{y}(k-l) \quad (61)$$

where $\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$, $\tilde{\mathbf{B}} = \mathbf{B}\mathbf{D}^{-1}$ and we have assumed that $\mathbf{x}(0) = \mathbf{0}$. Substituting (61) into (59), we deduce that (59) is satisfied for the i.i.d. property of $\mathbf{y}(k)$. In next section we will derive the stability conditions of the natural gradient algorithm for the state space model.

5.3 Lagrange Multiplier Approach

Salam and Erten [40] proposed another learning algorithm for the state space model by using the Lagrange Multiplier. Assume that the demixing model is described by (5) and (6). The augmented cost function becomes

$$J(\mathbf{U}) = \sum_{k=1}^{L}\left(l(k) + \lambda_{k+1}^T(\boldsymbol{\mathcal{F}}_N(k) - \mathbf{x}(k+1))\right) \quad (62)$$

where $\boldsymbol{\mathcal{F}}_N(k) = \boldsymbol{\mathcal{F}}_N(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta})$, $\mathbf{U} = (\mathbf{W}, \boldsymbol{\Theta})$ and $l(k) = l(\mathbf{y}(k), \mathbf{U})$ is given by (27), which depends on the parameters $\mathbf{U}$ of the demixing model. The Hamiltonian was defined as

$$H^k = l(k) + \lambda_{k+1}^T\boldsymbol{\mathcal{F}}_N(k) \quad (63)$$

By using the standard gradient descent approach they presented a learning algorithm [40]

$$\mathbf{x}(k+1) = \frac{\partial H^k}{\partial \lambda_{k+1}} = \boldsymbol{\mathcal{F}}_N(k) \quad (64)$$

$$\lambda_k = \frac{\partial \boldsymbol{\mathcal{F}}_N(k)}{\partial \mathbf{x}(k)}^T \lambda_{k+1} + \frac{\partial l(k)}{\partial \mathbf{x}(k)} \quad (65)$$

$$\begin{aligned} \Delta\boldsymbol{\Theta} &= -\eta(k)\frac{\partial \boldsymbol{\mathcal{F}}_N(k)}{\partial \boldsymbol{\Theta}}^T \lambda_{k+1} \quad (66)\\ \Delta\mathbf{W} &= -\eta(k)\frac{\partial l(k)}{\partial \mathbf{W}} \quad (67)\end{aligned}$$

Refer to [40] for a detailed derivation. It should be noted that if the demixing model is a linear system, the above learning algorithm for $\mathbf{W}$ reduces to the ordinary gradient descent algorithm (44) and (45). However, the learning algorithm (66) is not easy to implement on-line because the system (65) is non-causal.

6. Stability of Learning Algorithm

In this section we discuss the stability of the natural gradient algorithm (55) and (56). Since the algorithm is derived from (53) and (54), we only need to discuss the stability of (53) and (54). Consider its learning rule

$$\Delta \mathbf{X}_1(k) = -\eta \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{x}^T(k), \quad (68)$$
$$\Delta \mathbf{X}_2(k) = -\eta(\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k) - \mathbf{I}) \quad (69)$$

The equilibrium points of the dynamical system satisfy

$$E[\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{x}^T(k)] = 0 \quad (70)$$
$$E\left[\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k)\right] = 0 \quad (71)$$

Clearly, the true solution $\mathbf{C}$ and $\mathbf{D}$ is the solution of (70) and (71). However, this does not guarantee that the $\mathbf{C}(k)$ and $\mathbf{D}(k)$ converges to the true solution even locally. This is because if the true solution is an unstable equilibrium point of (55) and (56), the learning sequence $\mathbf{C}(k)$ and $\mathbf{D}(k)$ will never converge to it.

Assume that $\mathbf{y}(k)$ is the recovered signal, which is spatially mutually independent and temporarily i.i.d. We will prove that $\mathbf{y}(k)$ is a locally stable equilibrium point of the learning algorithm (68) and (69) under certain conditions on the source signals.

Consider the average version of the learning algorithm

$$\Delta \mathbf{X}_1(k) = \eta \mathbf{F}_1(\mathbf{X}) \quad (72)$$
$$\Delta \mathbf{X}_2(k) = \eta \mathbf{F}_2(\mathbf{X}) \quad (73)$$

where $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2]$, $\mathbf{F}_1(\mathbf{X}) = -E\left[\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{x}^T(k)\right]$, and $\mathbf{F}_2(\mathbf{X}) = \mathbf{I} - E\left[\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k)\right]$. Taking a small variation at the equilibrium point, we have

$$\Delta\delta\mathbf{X}_1(k) = \eta\left(\frac{\partial \mathbf{F}_1}{\partial \mathbf{X}_1}\delta\mathbf{X}_1 + \frac{\partial \mathbf{F}_1}{\partial \mathbf{X}_2}\delta\mathbf{X}_2\right) \quad (74)$$
$$\Delta\delta\mathbf{X}_2(k) = \eta\left(\frac{\partial \mathbf{F}_2}{\partial \mathbf{X}_1}\delta\mathbf{X}_1 + \frac{\partial \mathbf{F}_2}{\partial \mathbf{X}_2}\delta\mathbf{X}_2\right) \quad (75)$$

This shows that only when all the eigenvalues of the matrix

$$\begin{bmatrix} \frac{\partial \mathbf{F}_1}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{F}_1}{\partial \mathbf{X}_2} \\ \frac{\partial \mathbf{F}_2}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{F}_2}{\partial \mathbf{X}_2} \end{bmatrix} \quad (76)$$

have negative real parts, the derived equilibrium point is asymptotically stable. In fact, it becomes much easier to analyze the stability by using the variation instead of derivatives in this case. From the definition of $\mathbf{X}_1$ and $\mathbf{X}_2$, we have

$$\delta\mathbf{y}(k) = \delta\mathbf{X}_1\mathbf{x}(k) + \delta\mathbf{X}_2\mathbf{y}(k) \quad (77)$$

Consider the following variation

$$\delta E\left[\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{x}^T(k)\right] = \boldsymbol{\varphi}'(\mathbf{y})\delta\mathbf{y}(k)\mathbf{x}(k)^T$$
$$= \boldsymbol{\varphi}'(\mathbf{y})\left(\delta\mathbf{X}_1\mathbf{x}(k) + \delta\mathbf{X}_2\mathbf{y}(k)\right)\mathbf{x}(k)^T \quad (78)$$

Using the i.i.d. property of $\mathbf{y}(k)$ and (78), we derive the variational equation of $\mathbf{X}_1$

$$\Delta\delta X_{1,ij} = -\eta E\left[\varphi'(y_i(k)\sum_{p=1}^{N}\delta X_{1,ip}x_p x_j\right]$$
$$= -\eta\left[E[\varphi'(y_i(k))]\sum_{p=1}^{N}E[x_p x_j]\delta X_{1,ip}\right] \quad (79)$$

for $i = 1, \cdots, n,\ j = 1, \cdots, N$, or write it in a compact form

$$\Delta\delta\mathbf{X}_{1,i} = -\eta E\left[\varphi(y_i)\right]E[\mathbf{x}\mathbf{x}^T]\delta\mathbf{X}_{1,i} \quad (80)$$

for $i = 1, \cdots, n$, where $\mathbf{X}_{1,i}$ is the vector of the i-th row of matrix $\mathbf{X}_1$. We can easily derive the stability conditions for (80). If $E\left[\varphi(y_i)\right] > 0$ and $E[\mathbf{x}\mathbf{x}^T]$ is positive definite, the matrix $-\eta E\left[\varphi(y_i)\right]E[\mathbf{x}\mathbf{x}^T]$ is a negative definite matrix; therefore, all the eigenvalues of the matrix are negative.

Following a similar procedure in deriving (80), we derive the variational equation of $\mathbf{X}_2$

$$\Delta\delta\mathbf{X}_2 = -\eta E\left[diag(\boldsymbol{\varphi}'(\mathbf{y}))\delta\mathbf{X}_2\mathbf{y}\mathbf{y}^T\right]$$
$$-\eta E\left[\boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T\delta\mathbf{X}_2^T\right] \quad (81)$$

where $diag(\boldsymbol{\varphi}'(\mathbf{y}))$ is the diagonal matrix of vector $\boldsymbol{\varphi}'(\mathbf{y})$. For simplicity we omit the discrete time index k in the above equation. The equation (81) can be rewritten into a two-dimensional subsystem with a self-closed form. We take the following notation

$$\sigma_i^2 = E[y_i^2] \quad (82)$$
$$\kappa_i = E[\varphi_i'(y_i)] \quad (83)$$
$$m_i = E[y_i^2\varphi_i(y_i)] \quad (84)$$

where $\varphi_i' = \frac{d\varphi_i}{dy_i}$. Using the normalization condition (60) and the spatially mutual independence of $\mathbf{y}$, we simplify (81) to the following component form

$$\Delta\delta X_{2,ij} = -\eta\left(\kappa_i\sigma_j^2\delta X_{2,ij} + \delta X_{2,ji}\right) \quad (85)$$

$$\Delta\delta X_{2,ji} = -\eta\left(\kappa_j\sigma_i^2\delta X_{2,ji} + \delta X_{2,ij}\right) \quad (86)$$

for $i \neq j$, and $i,j = 1,\cdots,n$. Similarly, we derive the variational equations for the diagonal elements

$$\Delta\delta X_{2,ii} = -\eta(m_i+1)\delta X_{2,ii}. \quad (87)$$

Then the stability conditions for (85)-(87) are summarized as

$$\begin{aligned} m_i+1 &> 0,\ for\ i=1,\cdots,n & (88)\\ \kappa_i &> 0,\ for\ i=1,\cdots,n & (89)\\ \kappa_i\kappa_j\sigma_i^2\sigma_j^2 &> 1,\ for\ i,j=1,\cdots,n & (90)\end{aligned}$$

which have similar form to the one given by Amari et al [54]. In summary we have the following theorem

Theorem 2 *If the covariance matrix $E[\mathbf{x}\mathbf{x}^T]$ is positive definite and the conditions (88)-(90) are satisfied, the true solution is the asymptotically stable equilibrium point of the learning algorithm.*

If the mixing system is linear, the condition that the covariance matrix $E(\mathbf{x}\mathbf{x}^T)$ is positive definite can be further simplified. From the mixing model we have

$$\mathbf{x}(k) = \sum_{p=1}^{\infty}\tilde{\mathbf{A}}^{p-1}\tilde{\mathbf{B}}\mathbf{y}(k-p) \quad (91)$$

Using expression (91) and the i.i.d. property of $\mathbf{y}(k)$, we have

$$E\left[\mathbf{x}\mathbf{x}^T\right] = \sum_{p=1}^{\infty}\mathbf{\Psi}_p diag(\sigma_1^2,\cdots,\sigma_n^2)\mathbf{\Psi}_p^T \quad (92)$$

where $\mathbf{\Psi}_p = \tilde{\mathbf{A}}^{p-1}\tilde{\mathbf{B}}$. If the mixing system is controllable, then the matrix

$$[\mathbf{\Psi}_1\ \mathbf{\Psi}_2\ \cdots\ \mathbf{\Psi}_N] \quad (93)$$

is of full rank and the covariance matrix $E\left[\mathbf{x}\mathbf{x}^T\right]$ is positive definite.

7. Information Backpropagation

In order to develop a learning algorithm for matrices $\mathbf{A}$ and $\mathbf{B}$, we use the information backpropagation approach. Combining (39) and (41), we express the gradient of $l(\mathbf{y},\mathbf{W})$ with respect to $\mathbf{x}(k)$ as

$$\frac{\partial l(\mathbf{y},\mathbf{W})}{\partial \mathbf{x}(k)} = \mathbf{C}^T\boldsymbol{\varphi}(\mathbf{y}(k)) \quad (94)$$

Therefore, we can calculate the derivative of $l(\mathbf{y},\mathbf{W})$ with respect to $\mathbf{A}$ and $\mathbf{B}$ as

$$\frac{\partial l(\mathbf{y},\mathbf{W})}{\partial \mathbf{A}} = \sum_{l=1}^{N}\frac{\partial l(\mathbf{y},\mathbf{W})}{\partial x_l(k)}\frac{\partial x_l(k)}{\partial \mathbf{A}} \quad (95)$$

$$\frac{\partial l(\mathbf{y},\mathbf{W})}{\partial \mathbf{B}} = \sum_{l=1}^{N}\frac{\partial l(\mathbf{y},\mathbf{W})}{\partial x_l(k)}\frac{\partial x_l(k)}{\partial \mathbf{B}} \quad (96)$$

where $\frac{\partial x_l(k)}{\partial \mathbf{A}}$ and $\frac{\partial x_l(k)}{\partial \mathbf{B}}$ are obtained by the following on-line iterations

$$\frac{\partial x_l(k+1)}{\partial a_{ij}} = \sum_{p=1}^{N} a_{lp}\frac{\partial x_p(k)}{\partial a_{ij}} + \delta_{li}x_j(k) \quad (97)$$

$$\frac{\partial x_l(k+1)}{\partial b_{iq}} = \sum_{p=1}^{N} a_{lp}\frac{\partial x_p(k)}{\partial b_{iq}} + \delta_{li}u_q(k) \quad (98)$$

for $l,i,j = 1,\cdots,N$ and $q = 1,\cdots,n$, where δ_{li} is the Kronecker delta function. The minimization of the loss function (37) by the gradient descent method deduces a mutual information backpropagation learning algorithm as follows

$$\Delta a_{ij}(k) = -\eta(k)\boldsymbol{\varphi}(\mathbf{y}(k))^T\sum_{l=1}^{N}\mathbf{C}_l\frac{\partial x_l(k)}{\partial \mathbf{a}_{ij}} \quad (99)$$

$$\Delta b_{iq}(k) = -\eta(k)\boldsymbol{\varphi}(\mathbf{y}(k))^T\sum_{l=1}^{N}\mathbf{C}_l\frac{\partial x_l(k)}{\partial \mathbf{b}_{iq}} \quad (100)$$

for $i,j = 1,\cdots,N$ and $q = 1,\cdots,n$, where $\eta(k)$ is a learning rate and $\mathbf{C}_l$ is the l-th column vector of matrix $\mathbf{C}$.

Since matrices $\mathbf{A}$ and $\mathbf{B}$ are quite sparse in the canonical forms, we do not need to update all elements in the matrices. Here we elaborate the learning algorithm for the controller canonical form. In the controller canonical form, the matrix $\mathbf{B}$ is a constant matrix, and only the first n rows of matrix $\mathbf{A}$ are variable parameters. Denote the vector of l-row of matrix $\mathbf{A}$ by $\mathbf{a}_l$, $l = 1, \cdots, N$, and define

$$\frac{\partial \mathbf{x}(k)}{\partial \mathbf{a}_l} = \left(\frac{\partial x_i(k)}{\partial a_{lj}}\right)_{N\times N} \tag{101}$$

The derivative matrix $\frac{\partial \mathbf{x}(k)}{\partial \mathbf{a}_l}$ can be calculated by the following iteration

$$\frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{a}_l} = \mathbf{A}\frac{\partial \mathbf{x}(k)}{\partial \mathbf{a}_l} + \mathbf{\Phi}_l(k) \tag{102}$$

where $\mathbf{\Phi}_l(k) = (\delta_{li} x_j(k))_{N\times N}$. Substituting the above representation into (99) and (100), we have the following learning rule for $\mathbf{a}_l$,

$$\Delta \mathbf{a}_l = -\eta(k)\boldsymbol{\varphi}(\mathbf{y}(k))^T \mathbf{C}\frac{\partial \mathbf{x}(k)}{\partial \mathbf{a}_l} \tag{103}$$

The learning algorithm updates the internal parameters of the dynamical system on-line. The dynamical system (97) and (98) is the variational system of the demixing model with respect to $\mathbf{A}$ and $\mathbf{B}$. The purpose of the system is to estimate on-line the derivatives of $\mathbf{x}(k)$ with respect to $\mathbf{A}$ and $\mathbf{B}$. It should be noted that we must choose very carefully the initial value of the matrices $\mathbf{A}$ and $\mathbf{B}$ in numerical implementation. If a suitable initial value is not chosen, the demixing system or its variational system becomes unstable. The stability is the common problem in dynamical system identification. One solution is to formulate the demixing model in the Lyapunov balanced canonical form [37].

8. State Estimator – The Kalman Filter

There is a drawback in training $\mathbf{A}$ and $\mathbf{B}$ using the information backpropagation algorithm (99) and (100). It may suffer from instability of the systems, i.e. the eigenvalues of matrix $\mathbf{A}$ may be located outside of the unit cycle during learning. In order to overcome the problem, we employ the Kalman filter to estimate the state of the system. From output equation (11), it is observed that if we can accurately estimate the state vector $\mathbf{x}(k)$ of the system, then we can separate mixed signals using the learning algorithm (55) and (56).

8.1 Kalman Filter

The Kalman filter is a powerful approach for estimating the state vector in state-space models. The function of the Kalman filter is to generate on-line the state estimate of the state $\mathbf{x}(k)$. The Kalman filter dynamics are given as follows

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{K}\mathbf{r}(k) + \boldsymbol{\xi}_R(k) \tag{104}$$

where $\mathbf{K}$ is the Kalman filter gain matrix, and $\mathbf{r}(k)$ is called the innovation or residual which measures the error between the measured(or expected) output $\mathbf{y}(k)$ and the predicted output $\mathbf{C}\mathbf{x}(k)+\mathbf{D}\mathbf{u}(k)$. There are a variety of algorithms with which to update the Kalman filter gain matrix $\mathbf{K}$ as well as the state $\mathbf{x}(k)$; refer to [36] and [59] for more details.

However, in the blind deconvolution problem there exists no explicit residual $\mathbf{r}(k)$ to estimate state vector $\mathbf{x}(k)$ because the expected output $\mathbf{y}(t)$ here means the source signals, and we cannot measure the source signals. In order to solve the problem, we present a new concept called hidden innovation in order to implement the Kalman filter in the blind deconvolution case. Since updating matrices $\mathbf{C}$ and $\mathbf{D}$ will produce an innovation in each learning step, we introduce a hidden innovation as follows

$$\mathbf{r}(k) = \Delta \mathbf{y}(k) = \Delta \mathbf{C}\mathbf{x}(k) + \Delta \mathbf{D}\mathbf{u}(k) \tag{105}$$

where $\Delta \mathbf{C} = \mathbf{C}(k+1) - \mathbf{C}(k)$ and $\Delta \mathbf{D} = \mathbf{D}(k+1) - \mathbf{D}(k)$. The hidden innovation presents the adjusting direction of the output of the demixing system and is used to generate an a posteriori state estimate. Once we define the hidden innovation, we can employ the commonly used Kalman filter to estimate the state vector $\mathbf{x}(k)$, as well as to update the Kalman gain matrix $\mathbf{K}$. The updating rule in this paper is described as follows:

(1) Compute the Kalman gain

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^T + \mathbf{R}_k)^{-1}$$

(2) Update estimate with hidden innovation

$$\hat{\mathbf{x}}_k = \mathbf{x}_k + \mathbf{K}_k \mathbf{r}(k)$$

(3) Update the error covariance

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k k)\mathbf{P}$$

(4) evaluate the state vector ahead

$$\mathbf{x}_{k+1} = \mathbf{A}_k \hat{\mathbf{x}}_k + \mathbf{B}_k \mathbf{u}_k$$

(5) evaluate the error covariance ahead

$$\mathbf{P}_{k+1} = \mathbf{A}_k \hat{\mathbf{P}}_k \mathbf{A}_k^T + \mathbf{Q}_k$$

where $\mathbf{Q}_k$ and $\mathbf{R}_k$ are the covariance matrices of the noise vector $\boldsymbol{\xi}_R$ and output measurement noise $\mathbf{n}_k$, respectively.

The theoretic problems such as convergence and stability remain to be analyzed. Simulation experiments show that the algorithm, based on the Kalman filter, can separate the convolved signals very well.

9. Two-stage Separation Algorithm

In this section we present a novel two-stage separation algorithm for state-space models. In this approach we decompose the separation problem into the following two stages. First we separate the mixed signals in the following sense

$$\mathbf{W}(z)\mathbf{H}(z) = \mathbf{P}\mathbf{Q}(z) \tag{106}$$

where $\mathbf{Q}(z) = diag(q_1(z), \cdots, q_n(z))$ is a diagonal matrix with polynomials of z^{-1} in its diagonal entity. At this stage the output signals are mutually independent but in single channel convolution. Therefore, we need only to apply single channel equalization methods, such as the natural gradient approach or Bussgang methods, to obtain the temporarily i.i.d. recovered signals.

The question here is whether matrices $[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$ exist in the demixing model (10) and (11), such that its transfer function $\mathbf{W}(z)$ satisfies (106). The answer is affirmative. Suppose that there is a inverse filter $\mathbf{W}_0(z)$ of $\mathbf{H}(z)$ in the sense of (10). Since $\mathbf{W}_0(z)$ is a rational polynomial of z^{-1}, we know that there is a state-space realization $[\mathbf{A}_0, \mathbf{B}_0, \mathbf{C}_0, \mathbf{D}_0]$ of $\mathbf{W}_0(z)$. Then we rewrite $\mathbf{W}_0(z)$ into following form

$$\begin{aligned} \mathbf{W}_0(z) &= \mathbf{D}_0 + \mathbf{C}_0(z\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{B}_0 \\ &= \sum_{i=0}^{N} \mathbf{P}_i z^{-i} / q(z^{-1}) \end{aligned} \tag{107}$$

We can construct a linear system with transfer function $\sum_{i=0}^{N} \mathbf{P}_i z^{-i}$ as follows

$$\mathbf{A} = \begin{bmatrix} \mathcal{O}^T & \mathbf{O}_n \\ \mathbf{I}_{n(N-1)} & \mathcal{O} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{I}_n \\ \mathcal{O} \end{bmatrix} \tag{108}$$

$$\mathbf{C} = (\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_n), \quad \mathbf{D} = \mathbf{P}_0 \tag{109}$$

where $\mathbf{I}_{n(N-1)}$ is an $n(N-1) \times n(N-1)$ identity matrix, $\mathbf{O}_n$ are an $n \times n$ zero matrix, and $\mathcal{O}$ is an $n(N-1) \times n$ zero matrix, respectively. Then we deduce that $\mathbf{W}(z) = \mathbf{D} + \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = \mathbf{W}_0(z)q(z^{-1})$. Thus we have

$$\mathbf{W}(z)\mathbf{H}(z) = \mathbf{P}\mathbf{\Lambda}(z)q(z^{-1}) = \mathbf{P}\mathbf{Q}(z) \tag{110}$$

where $\mathbf{Q}(z)$ is a diagonal matrix with polynomials of z^{-1} in its diagonal entities. It is easily seen that both $\mathbf{A}$ and $\mathbf{B}$ are constant matrices. Therefore, we have only to develop a learning algorithm to update $\mathbf{C}$ and $\mathbf{D}$ so as to obtain the separated signals in the sense of (106).

On the other hand, we know that if the matrix $\overline{\mathbf{D}}$ in the mixing model satisfies $rank(\overline{\mathbf{D}}) = n$, then there exist matrices $[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$, such that the output signal $\mathbf{y}$ of state-space system (10) and (11) recovers the independent source signals in the sense of (106). Therefore, we have the following theorem:

Theorem 3 *If the matrix $\overline{\mathbf{D}}$ in the mixing model satisfies $rank(\overline{\mathbf{D}}) = n$, then for given specific matrices $\mathbf{A}$ and $\mathbf{B}$ as (108), there exist matrices $[\mathbf{C}, \mathbf{D}]$, such that the transfer matrix $\mathbf{W}(z)$ of the system (10) and (11) meets equation (106).*

The two-stage blind deconvolution is realized in the following way: first we give the matrices $\mathbf{A}$ and $\mathbf{B}$ of the state equation in the form (108), and then employ the natural gradient algorithm to update $\mathbf{C}$ and $\mathbf{D}$. We intend to make the output of the demixing model as spatially mutually independent as possible. After the first stage the outcome signals are in the following form

$$\hat{y}_i(k) = q(z)s_i(k), \text{ for } i = 1, \cdots, n \tag{111}$$

Then we employ the natural gradient algorithm for double finite FIR filter, to remove the convolved signals. From computer simulations we see that the two-stage approach can also recover the source signals mixed by a non-minimum phase system.

10. Learning Algorithm for Nonlinear Models

In this section we employ the mutual information backpropagation approach to train the neural network in the demixing model. The mutual information backpropagation is based on the real-time recurrent learning. The algorithm adjusts the synaptic weights of a fully connected recurrent network in real time [60]. In this paper we extend the real-time recurrent learning to the generalized blind deconvolution case.

Consider the demixing model in the following form

$$\begin{aligned} \mathbf{x}(k+1) &= \boldsymbol{\mathcal{F}}_N(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta}) && (112)\\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k)) && (113) \end{aligned}$$

where $\boldsymbol{\mathcal{F}}_N(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta})$ is a certain neural network and $\boldsymbol{\Theta}$ is the training parameters of the network.

Following the same derivation for (55) and (56), we have

$$dl(\mathbf{y}, \mathbf{W}) = -tr(d\mathbf{D}\mathbf{D}^{-1}) + \boldsymbol{\varphi}^T(\mathbf{y})d\mathbf{y} \qquad (114)$$

Taking a differential of $\mathbf{y}$ in equation (112), we have the following relation

$$d\mathbf{y} = d\mathbf{C}\mathbf{x}(k) + d\mathbf{D}\mathbf{u}(k) + \mathbf{C}d\mathbf{x}(k) \qquad (115)$$

If we introduce a new search direction as in (49) and (50), we can derive a learning algorithm for $\mathbf{C}$ and $\mathbf{D}$, which is the same as (55) and (56).

The updating rule for the parameters $\boldsymbol{\Theta}$ is to use the information backpropagation technique. From (114) and (115), we derive the derivative $\frac{\partial l(\mathbf{y},\mathbf{W})}{\partial \boldsymbol{\Theta}}$

$$\frac{\partial l(\mathbf{y}, \mathbf{W})}{\partial \boldsymbol{\Theta}} = \frac{\partial l(\mathbf{y}, \mathbf{W})}{\partial \mathbf{x}(k)} \frac{\partial \mathbf{x}(k)}{\partial \boldsymbol{\Theta}} \qquad (116)$$

where $\frac{\partial \mathbf{x}(k)}{\partial \boldsymbol{\Theta}}$ is recurrently calculated by

$$\frac{\partial \mathbf{x}(k+1)}{\partial \boldsymbol{\Theta}} = \frac{\partial \mathcal{F}(k)}{\partial \mathbf{x}(k)} \frac{\partial \mathbf{x}(k)}{\partial \boldsymbol{\Theta}} + \frac{\partial \mathcal{F}(k)}{\partial \boldsymbol{\Theta}} \qquad (117)$$

where $\mathcal{F}(k) = \boldsymbol{\mathcal{F}}_N(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta})$. The above two equations are only written formally; the precise representation should be in tensor format. The recurrent equation (117) estimates the derivative of state vector $\mathbf{x}(k)$ with respect to the synaptic weights $\boldsymbol{\Theta}$ of the neural network. Using the gradient descent approach, we derive the updating rule for $\boldsymbol{\Theta}$

$$\Delta \boldsymbol{\Theta} = -\eta \varphi'(\mathbf{y}(k))^T \mathbf{C} \frac{\partial \mathbf{x}(k)}{\partial \boldsymbol{\Theta}} \qquad (118)$$

where η is a learning rate.

In order to give an explicit form to the information backpropagation algorithm, one must employ a neural network with general approximation to approximate the nonlinear mapping $\boldsymbol{\mathcal{F}}_N(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta})$. There are a number of neural networks suitable for this purpose, such as the Support Vector Machine, the Radial Based Functions and Multilayer Perceptrons. In this paper we use the Support Vector Machine (SVM) and derive the explicit learning algorithm for training the neural network. A Support Vector Machine for function approximation and pattern recognition utilizes a kernel function to map the data to a Hilbert space, in which the problem becomes linear. The SVM for function approximation is of the following form

$$\boldsymbol{\mathcal{F}}_N(\mathbf{z}) = \sum_{p=1}^{L} \mathbf{V}_p \mathcal{K}(\mathbf{z}, \mathbf{z}_p) + \mathbf{b} \qquad (119)$$

where $\mathbf{z} = [\mathbf{x}^T, \mathbf{u}^T]^T, \mathbf{z}_p \in \mathbf{R}^{N+n}, \mathbf{b} \in \mathbf{R}^N$, and $\mathbf{V}_p = (v_{p,ij}) \in \mathbf{R}^{N\times(N+n)}$. The vector elements $\mathbf{z}_p$, weights $\mathbf{V}_p$ and $\mathbf{b}$ are parameters that are to be determined by the learning process, and the kernel $\mathcal{K}(\mathbf{z}, \mathbf{z}_p)$ is usually chosen in advance, which satisfies Mercer's condition. Refer to [61] for detailed information about the SVM and the kernel function. The recurrent equations for the parameters $\mathbf{V}_p, \mathbf{z}_p$ and $\mathbf{b}$ are expressed by

$$\frac{\partial \mathbf{x}(k+1)}{\partial v_{p,ij}} = \frac{\partial \boldsymbol{\mathcal{F}}_N(k)}{\partial \mathbf{x}^T(k)} \frac{\partial \mathbf{x}(k)}{\partial v_{p,ij}} + I_{ij} \boldsymbol{\mathcal{K}}_p \qquad (120)$$

$$\frac{\partial \mathbf{x}(k+1)}{\partial z_{p,i}} = \frac{\partial \boldsymbol{\mathcal{F}}_N(k)}{\partial \mathbf{x}^T(k)} \frac{\partial \mathbf{x}(k)}{\partial z_{p,i}} + \mathbf{V}_p \frac{\partial \boldsymbol{\mathcal{K}}_p}{\partial z_{p,i}} \qquad (121)$$

$$\frac{\partial \mathbf{x}(k+1)}{\partial b_i} = \frac{\partial \boldsymbol{\mathcal{F}}_N(k)}{\partial \mathbf{x}^T(k)} \frac{\partial \mathbf{x}(k)}{\partial b_i} + \mathbf{I}_i \qquad (122)$$

where $\frac{\partial \boldsymbol{\mathcal{F}}_N(k)}{\partial \mathbf{x}^T(k)} = \sum_{p=1}^{L} \mathbf{V}_p \frac{\partial \mathcal{K}(\mathbf{z},\mathbf{z}_p)}{\partial \mathbf{x}^T(k)}$, $\boldsymbol{\mathcal{K}}_p = \mathcal{K}(\mathbf{z}, \mathbf{z}_p)$, and I_{ij} is an $N\times(N+n)$-matrix with all zero elements except the (i,j)-th element equal to 1. $\mathbf{I}_i$ is a $N-$dimensional vector with all zero elements except the i-element equal to 1. Therefore, the updating rule for $\mathbf{V}_p, \mathbf{z}_p$ and $\mathbf{b}$ in the SVM is

$$\begin{aligned} \Delta v_{p,ij} &= -\eta \varphi'(\mathbf{y}(k))^T \mathbf{C} \frac{\partial \mathbf{x}(k)}{\partial v_{p,ij}} && (123)\\ \Delta z_{p,i} &= -\eta \varphi'(\mathbf{y}(k))^T \mathbf{C} \frac{\partial \mathbf{x}(k)}{\partial z_{p,i}} && (124)\\ \Delta b_i &= -\eta \varphi'(\mathbf{y}(k))^T \mathbf{C} \frac{\partial \mathbf{x}(k)}{\partial b_i} && (125) \end{aligned}$$

Because of the approximation of nonlinear function by SVM, it unavoidably produces a model bias, which leads to a model error. In order to compensate for the model bias, we can also introduce the state estimator−Extended Kalman Filter approach to estimate the state vector of the demixing model. A detailed analysis of the Extended Kalman Filter for blind deconvolution will be presented in a separate paper.

11. Computer Simulations

In this section we present a number of computer simulations to demonstrate the validity and effectiveness of the natural gradient algorithm, the information backpropagation algorithm and the Kalman filter for blind deconvolution. Comparisons between several basic separation algorithms are also given.

To evaluate the performance of the proposed learning algorithms, we employ the multichannel intersymbol interference [33], denoted by M_{ISI}, as a criteria,

$$M_{ISI} = \sum_{i=1}^{n} \frac{|\sum_j \sum_p |\mathbf{G}_{pij}| - \max_{p,j} |\mathbf{G}_{pij}|}{\max_{p,j} |\mathbf{G}_{pij}|} + \sum_{j=1}^{n} \frac{|\sum_i \sum_p |\mathbf{G}_{pij}| - \max_{p,i} |\mathbf{G}_{pij}|}{\max_{p,i} |\mathbf{G}_{pij}|} \quad (126)$$

It is easy to show that $M_{ISI} = 0$ if and only if $\mathbf{G}(z)$ is of the form (36). In order to avoid the effect of a single numerical trial on evaluating the performance of algorithms, we use the ensemble average approach, that is, in each trial we obtain a time sequence of M_{ISI}, and then we take average of the ISI performance to evaluate the performance of algorithms.

The learning rate is another important factor in implementing the natural gradient algorithm. The strategy in this paper is to update the learning rate by $\eta(k+1) = \max\{0.9\eta(k), 10^{-4}\}$; for each 200 iterations, the initial value $\eta(0) = 10^{-2}$.

11.1 The Natural Gradient Algorithm vs. the Ordinary Gradient Algorithm

A large number of computer simulations have been performed to compare the learning performance of the natural gradient algorithm (55) and (56) with the ordinary gradient algorithm (44) and (45). In this group of simulations, we assume that the internal parameters in the demixing model are predetermined and represented in the controller canonical form (14).

The mixing model used for computer simulations is the multichannel ARMA model

$$\mathbf{u}(k) + \sum_{i=1}^{N} \mathbf{A}_i \mathbf{u}(k-i) = \sum_{i=0}^{N} \mathbf{B}_i \mathbf{s}(k-i) + \mathbf{v}(k) \quad (127)$$

where $\mathbf{u}, \mathbf{s}$ and $\mathbf{v} \in \mathbf{R}^3$. The matrices $\mathbf{A}_i \in \mathbf{R}^{3\times3}$ and $\mathbf{B}_i \in \mathbf{R}^{3\times3}$ are randomly chosen such that the mixing system is stable and minimum phase. The source signals $\mathbf{s}$ are randomly generated i.i.d signals uniformly distributed in the range (-1,1), and $\mathbf{v}$ are the Gaussian noises with zero mean and a covariance matrix $0.1\mathbf{I}$. The nonlinear activation function is chosen to be $\boldsymbol{\varphi}_i(y_i) = y_i^3$ for any i.

Example 1. We employ an AR model of order $N = 10$ as a mixing system, which can be exactly inverted by a FIR filter. A large number of simulations show that the natural gradient learning algorithm can easily and quickly recover source signals in the sense of $\mathbf{W}(z)\mathbf{H}(z) = \mathbf{P\Lambda}$.

Figure 2 illustrates 100 trial ensemble average M_{ISI} performances of the natural gradient learning algorithm and the ordinary gradient learning algorithm. It is observed that the natural gradient algorithm usually needs less than 3000 iterations to obtain satisfactory results, while the ordinary gradient algorithm needs more than 20000 iterations to obtain satisfactory results, since there is a long plateau in the ordinary gradient learning.

11.2 Information Backpropagation

In the previous subsection we assumed that the internal parameters are designed as fixed matrices. However, in many practical applications, we need to train both the internal parameters and external parameters. If we know nothing about the demixing model, a general solution is to choose an initial value for $\mathbf{A}$ and $\mathbf{B}$ in the controller canonical form, then use the information backpropagation to update the internal parameters during training.

Example 2. We assume that the mixing system is an ARMA model of order 10, which is stable and minimum phase. The transfer function

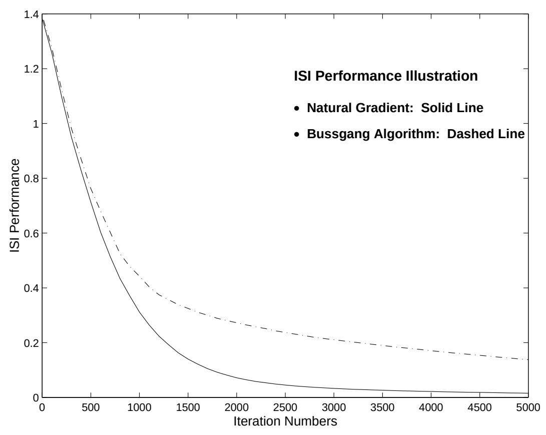

Fig. 2 M_{ISI} performance of the natural gradient algorithm for Example 1

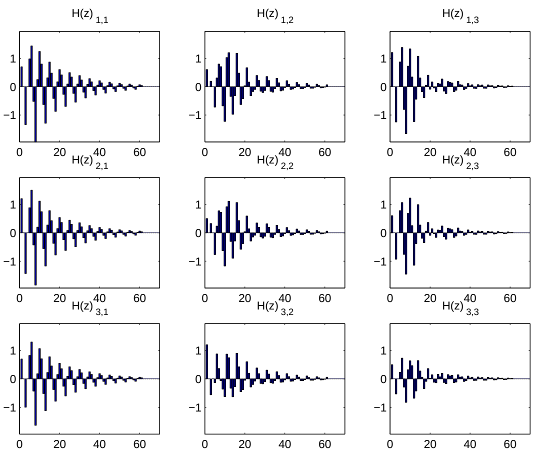

Fig. 3 The coefficients of $\mathbf{H}(z)$ of the mixing system for Example 2

of the mixing system is plotted in Fig. 3. The demixing system is chosen to be a state space system of order 40, and the initial values for $\mathbf{A}$ and $\mathbf{B}$ are in the form (108). From the simulation we see that if we do not update $\mathbf{A}$ and $\mathbf{B}$, the outcome of the recovered signal cannot be perfect. After training $\mathbf{A}$ by using algorithm (103), we can recover source signals quite well. Fig. 4 plots the global transfer function $\mathbf{G}(z) = \mathbf{W}(z)\mathbf{H}(z)$ up to order 60.

From computer simulations we see that the overestimation of system order N essentially do not affect the outcome of the learning algorithm, but it only increases the computing cost.

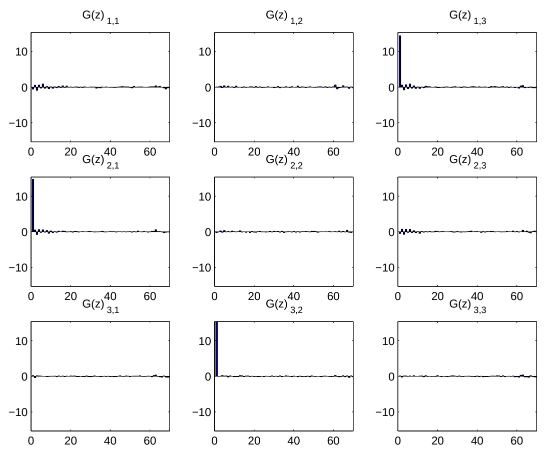

Fig. 4 The coefficients of $\mathbf{G}(z)$ after 3000 iterations for Example 2

11.3 Kalman filter implementation

Because the approximation of internal parameters will unavoidably produce a model bias, we employ the Kalman filter to compensate for the model bias and reduce the effect of noise. Several numerical simulations have been performed to demonstrate the performance of the Kalman filter. Here we give only one illustrative example.

Example 3. The transfer function of the mixing system is plotted in Fig. 5. It is assumed to be unknown for the algorithm.

Assume that source signals are i.i.d quadrature amplitude modulated (QAM). The Gaussian noise represented by $\mathbf{v}$ was zero mean with a covariance matrix $0.1\mathbf{I}$. The initial values for matrices $\mathbf{A}$ and $\mathbf{B}$ in the state equation are chosen to be ones in canonical controller form and the initial value for matrix $\mathbf{C}$ is set to a zero matrix or given randomly in the range (-1,1), and $\mathbf{D} = \mathbf{I}_3 \in \mathbf{R}^{3\times 3}$.

We use the natural gradient algorithm (GD) to train the output matrices $\mathbf{C}$ and $\mathbf{D}$, use the information backpropagation (IB) algorithm (103) to estimate the state matrix $\mathbf{A}$ and employ the Kalman filter (KF) to estimate the state vector $\mathbf{x}(k)$ of the system as well. Figures 6 and 7 show the sensor signal constellation, output constellation of demixing system by using the natural gradient algorithm, information backpropagation and Kalman filter, respectively.

It is worth noting that the output signals con-

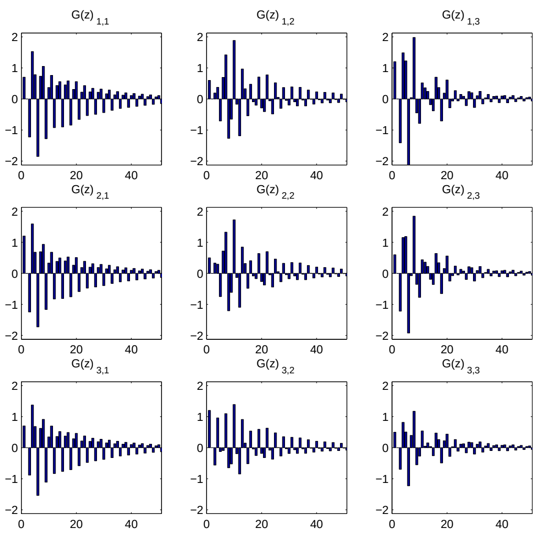

Fig. 5 The coefficients of $\mathbf{H}(z)$ of the mixing system for Example 3

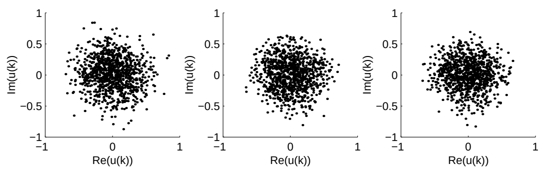

Fig. 6 Sensor signal constellation for Example 3

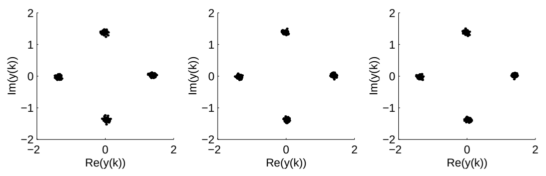

Fig. 7 Output constellation for Example 3

verge to the characteristic QAM constellation, up to an amplitude and phase rotation factors ambiguities.

12. Conclusion

In this paper we have presented a general framework of the state space approach for multichannel blind deconvolution/separation. The state space model allows us to separate blind deconvolution in two steps: supervised learning for internal parameters and unsupervised learning for external parameters. Adaptive learning algorithms for updating external parameters are developed by minimizing the suitable cost function, which is derived from mutual information of output signals. The information backpropagation approach technique is developed for training the internal parameters as an alternative method. An state estimator based on the Kalman filter is also presented in order to amend the model bias and reduce the effect of noise. Finally we give suggestions for how to extend the information backpropagation to the nonlinear case. Computer simulations are given to demonstrate the validity and effectiveness of the state-space approach.

13. Appendix

13.1 Inverse of Linear State Space Model

Suppose that $\overline{\mathbf{D}}$ satisfies $rank(\overline{\mathbf{D}}) = n$, and $\overline{\mathbf{D}}^{\dagger}$ is the generalized inverse of $\overline{\mathbf{D}}$, in the sense of a Penrose generalized pseudo-inverse. Let

$$\mathbf{D} = \overline{\mathbf{D}}^{\dagger},\ \mathbf{A} - \overline{\mathbf{A}} = \mathbf{B}\overline{\mathbf{C}}$$

$$\mathbf{B} = \overline{\mathbf{B}}\mathbf{D},\ \mathbf{C} = -\mathbf{D}\overline{\mathbf{C}}$$

then the global system can be described as

$$\mathbf{G}(z) = \mathbf{W}(z)\mathbf{H}(z) = \mathbf{I} \qquad (128)$$

The state transform does not change the transfer functions, for any nonsingular transform $\mathbf{T}$, if the following relation holds

$$\begin{aligned}
\mathbf{A} &= \mathbf{T}(\overline{\mathbf{A}} - \overline{\mathbf{B}}\overline{\mathbf{D}}^{\dagger}\overline{\mathbf{C}})\mathbf{T}^{-1}\\
\mathbf{B} &= \mathbf{T}\overline{\mathbf{B}}\overline{\mathbf{D}}^{\dagger}\\
\mathbf{C} &= -\overline{\mathbf{D}}^{\dagger}\overline{\mathbf{C}}\mathbf{T}^{-1}\\
\mathbf{D} &= \overline{\mathbf{D}}^{\dagger}
\end{aligned}$$

Therefore, source signals can be recovered by linear state space demixing model (10) and (11).

13.2 Derivation of Cost Function

We consider n observations $\{u_i(k)\}$ and n output signals $\{y_i(k)\}$ with length L.

$$\begin{aligned}
\mathcal{U}(k) &= [\mathbf{u}^T(1), \mathbf{u}^T(2), \ldots, \mathbf{u}^T(L)]^T\\
\mathcal{Y}(k) &= [\mathbf{y}^T(1), \mathbf{y}^T(2), \ldots, \mathbf{y}^T(L)]^T
\end{aligned}$$

where $\mathbf{u}(k) = [u_1(k), \cdots, u_n(k)]^T$ and $\mathbf{y}(k) = [y_1(k), \cdots, y_n(k)]$. The task of blind deconvolution is to train a state space demixing model such that the joint probability density of $\mathcal{Y}$ is factorized as follows:

$$p(\mathcal{Y}) = \prod_{i=1}^{n} \prod_{k=1}^{L} p_i(y_i(k)) \tag{129}$$

where $\{p_i(\cdot)\}$ is the probability density of source signals. In order to measure the mutual independence of output signals, we employ the Kullback-Leibler divergence as a criterion, which is an asymmetric measure of distance between two different probability distributions,

$$KL(\mathbf{W}(z)) = \frac{1}{L} \int p(\mathcal{Y}) \log \frac{p(\mathcal{Y})}{\prod_{i=1}^{n} \prod_{k=1}^{L} q_i(y_i(k))} \tag{130}$$

where we replace $p_i(\cdot)$ by certain approximate density functions $q_i(\cdot)$ for estimated sources, since we do not know the true probability distributions $p_i(\cdot)$ of original source signals.

Provided that initial conditions are set to $\mathbf{x}(1) = \mathbf{0}$, we have the following relation

$$\mathcal{Y} = \mathcal{W}\mathcal{U} \tag{131}$$

where $\mathcal{W}$ is given by

$$\mathcal{W} = \begin{bmatrix} \mathbf{H}_0 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_1 & \mathbf{H}_0 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{H}_{L-2} & \mathbf{H}_{L-3} & \ddots & \mathbf{H}_0 & \mathbf{0} \\ \mathbf{H}_{L-1} & \mathbf{H}_{L-2} & \cdots & \mathbf{H}_1 & \mathbf{H}_0 \end{bmatrix} \tag{132}$$

where $\mathbf{H}_i$ for $i = 0, 1, \cdots$ are the Markov parameters defined by $\mathbf{H}_0 = \mathbf{D}$, $\mathbf{H}_i = \mathbf{C}\mathbf{A}^{i-1}\mathbf{B}$, $i = 0, 1, \cdots$. According to the property of the probability density function, we derive the following relation between $p(\mathcal{U})$ and $p(\mathcal{Y})$:

$$p(\mathcal{Y}) = \frac{p(\mathcal{U})}{|\det \mathbf{H}_0^L|} \tag{133}$$

Using the relation (130), we derive the cost function $l(\mathbf{W}(z))$ as follows

$$l(\mathbf{W}(z)) = -\log|\det \mathbf{H}_0| - \sum_{i=1}^{n} \frac{1}{L} \sum_{k=1}^{L} \log q_i(y_i(k)) \tag{134}$$

Note that $p(\mathcal{U})$ was not included in (134) because it does not depend on the set of parameters $\{\mathbf{W}\}$.

References

[1] S. Amari: Natural gradient works efficiently in learning, *Neural Computation*, Vol.10, pp.251–276, 1998.

[2] S. Amari and A. Cichocki: Adaptive blind signal processing– neural network approaches, *Proceedings of the IEEE*,Vol. 86, No. 10, pp.2026–2048, 1998.

[3] C. Jutten and J. Herault: Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture, *Signal Processing*, Vol. 24, pp.1–10, 1991.

[4] C. Jutten, H. Nguyen Thi, E. Dijkstra, E. Vittoz, and J. Caelen: Blind separation of sources, an algorithm for separation of convolutive mixtures, *Proc. of Int. Workshop on High Order Statistics*, pp. 273–276, Chamrousse, France, 1991.

[5] P. Comon, C. Jutten, and J. Herault: Blind separation of sources, Part II: Problem statement, *Signal Processing*, Vol. 24, pp.11–20, 1991.

[6] P. Comon: Independent component analysis: a new concept? *Signal Processing*, Vol. 36, pp.287–314, 1994.

[7] A. Cichocki, R. Unbehauen, and E. Rummert: Robust learning algorithm for blind separation of signals, *Electronics Letters*, Vol. 30, No. 17, pp.1386–1387, 1994.

[8] A. Bell and T. Sejnowski: An information maximization approach to blind separation and blind deconvolution, *Neural Computation*, Vol.7, pp.1129–1159, 1995.

[9] S. Amari, A. Cichocki, and H. Yang: A new learning algorithm for blind signal separation, In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems 8 (NIPS'95)*, pp. 757–763, Cambridge, MA, 1996. The MIT Press.

[10] A. Cichocki and R. Unbehauen: Robust neural networks with on-line learning for blind identification and blind separation of sources, *IEEE Trans Circuits and Systems I : Fundamentals Theory and Applications*, Vol. 43, No. 11, pp.894–906, 1996.

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

[12] Y. Hua: Fast maximum likelihood for blind identification of multiple FIR channels, *IEEE Trans. Signal Processing*, Vol. 44, pp.661–672, 1996.

[13] H. Yang and S. Amari: Adaptive on-line learning algorithms for blind separation: Maximum entropy and minimal mutual information, *Neural Comput.*, Vol. 9, pp.1457–1482, 1997.

[14] J.-F. Cardoso: Blind signal separation: Statistical principles, *Proceedings. of the IEEE*, Vol. 86, No. 10, pp.2009–2025, 1998.

[15] R. W. Lucky: Techniques for adaptive equalization of digital communication systems, *Bell Sys. Tech. J.,*, Vol. 45, pp.255–286, 1966.

[16] D.N. Godard: Self-recovering equalization and carrier tracking in two-dimensional data communica-

tion systems, *IEEE trans. Comm.*, Vol. COM-28, pp.1867–1875, 1980.

[17] A. Benveniste and M. Goursat: Blind equalizers, *IEEE Trans. Comm.*, Vol. 32, pp.871–883, 1984.

[18] A. Cichocki and W. Kasprzak: Local adaptive learning algorithms for blind separation of natural images, *Neural Network World*, Vol. 6, No. 4, pp.515–523, 1996.

[19] W. Kasprzak and A. Cichocki: Hidden image separation from incomplete image mixtures by independent component analysis, *Proc. of 13th Int. Conf. on Pattern Recognition (ICPR'96)*, pp. 394–398. IEEE Computer Society Press, 1996.

[20] A. Gorokhov, P. Loubation, and E. Moulines: Second order blind equalization in multiple input multiple output FIR systems: A weighted least squares approach, *Proc. IEEE int. Conf. Acoust., Speech, Signal Processing*, pp. 2415–2418, Atlanta, May 1996.

[21] S. Makeig, A. Bell, T.-P. Jung, and T. Sejnowski: Independent component analysis in electroencephalographic data, In M. M. et al., editor, *Advances in Neural Information Processing Ssytems 8*, Vol. 8, pp. 145–151, Cambridge, MA, 1996. MIT Press.

[22] K. Torkkola: Blind separation of convolved sources based on information maximization, *Proc. of the 1996 IEEE Workshop Neural Networks for Signal Processing 6 (NNSP96)*, pp. 423–432, New York, NY, 1996. IEEE Press.

[23] A. Cichocki, S. Amari, and J. Cao: Neural network models for blind separation of time delayed and convolved signals, *IEICE Trans. Fundamentals*, Vol. E80-A, No. 9, pp.1595–1603, Sept. 1997.

[24] T. W. Lee, A. Bell, , and R. Lambert: Blind separation of delayed and convolved sources, *Advances in Neural Information Processing Systems, Vol. 9*, pp. 758–764, Boston, MA, 1996. MIT Press.

[25] S. Haykin: *Blind Deconvolution*, Prentice-Hall, New Jersey, 1994.

[26] S. Amari and J.-F. Cardoso: Blind source separation– semiparametric statistical approach, *IEEE Trans. Signal Processing*, Vol. 45, No. 11, pp.2692–2700, Nov. 1997.

[27] S. Bellini: Bussgang Techniques for blind equalization, *Proc. of IEEE Global Telecommunications Conference,* pp. 1634-1640, Houston, TX, 1986.

[28] S. Amari, S. Douglas, A. Cichocki, and H. Yang: Multichannel blind deconvolution and equalization using the natural gradient, *Proc. IEEE Workshop on Signal Processing Adv. in Wireless Communications*, pp. 101–104, Paris, France, April 1997.

[29] S. Amari, S. Douglas, A. Cichocki, and H. Yang: Novel on-line algorithms for blind deconvolution using natural gradient approach, *Proc. 11th IFAC Symposium on System Identification, SYSID'97*, pp. 1057–1062, Kitakyushu, Japan, July, 8-11 1997.

[30] S. Amari, S. Douglas, and A. Cichocki: Multichannel blind deconvolution and source separation using the natural gradient, *IEEE Trans. Signal Processing*, to appear.

[31] L. Zhang, S. Amari, and A. Cichocki: Natural gradient approach to blind separation of over- and undercomplete mixtures, *Proc. of Independent Component Analysis and Signal Separation(ICA'99)*, pp. 455–460, Aussois, France, 1999.

[32] Y. Inouye and T. Habe: Multichannel blind equalization using second- and fourth-order cumulants, *Proc. IEEE Signal Processing Workshop on Higher-Order Statistics*, pp. 96–100, 1995.

[33] Y. Inouye and S. Ohno: Adaptive algorithms for implementing the single-stage criterion for multichannel blind deconvolution, *Proc. of 5th Int. Conf. on Neural Information Processing, ICONIP'98*, pp. 733–736, Kitakyushu, Japan, October, 21-23 1998.

[34] J. Tugnait: Identification and deconvolution of multichannel linear non-gaussian processes using higher order statistics and inverse filter criteria, *IEEE Trans. Signal Processing*, Vol. 45, pp.658–672, 1997.

[35] T. Kailath: *Linear Systems*, Prentice Hall, Englewood Cliffs, NJ, 1980.

[36] O. L. R. Jacobs: *Introduction to Control Theory*, Oxford University Press, 1993.

[37] R. J. Ober: Balanced canonical forms, In S. Bittanti and G. Picci, editors, *Identification, Adaptation, Learning*, NATO ASI Series, pp. 120–183. Springer, 1996.

[38] F. Salam: An adaptive network for blind separation of independent signals, *Proc. '93 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 431–434, 1993.

[39] A. Gharbi and F. Salam: Algorithm for blind signal separation and recovery in static and dynamics environments, *Proc. IEEE Symposium on Circuits and Systems*, pp. 713–716, Hong Kong, 1997.

[40] F. Salam and G. Erten: The state space framework for blind dynamic signal extraction and recovery, *Proc of '99 Int. Symposium on Circuits and Systems, ISCAS'99*, pp. V–66 –69, Orlando, Florida, May 30-June 2 1999.

[41] L. Zhang and A. Cichocki: Blind deconvolution/equalization using state-space models, *Proc. of 1998 Int'l IEEE Workshop on Neural Networks for Signal Processing (NNSP'98)*, pp. 123–131, Cambridge, UK, August 31-September 2 1998.

[42] L. Zhang and A. Cichocki: Information backpropagation learning algorithm for blind dynamic separation, *Proc. of IASTED International Conference Signal and Image Processing (SIP'98)*, pp. 1–5, Las Vegas, October. 28-31 1998.

[43] L. Zhang and A. Cichocki: Blind separation of filtered source using state-space approach, In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems*, Vol. 11, pp. 648–654. MIT press, Cambridge, MA, 1999.

[44] A. Cichocki, L. Zhang, and S. Amari: Semi-blind and state-space approaches to nonlinear dynamic independent component analysis, *Proc. NOLTA'98*, pp. 291–294, 1998.

[45] A. Cichocki and L. Zhang: Two-stage blind deconvolution using state-space models (invited), *Proc.*

of the Fifth International Conference on Neural Information Processing(ICONIP'98), pp. 729–732, Kitakyushu, Japan, Oct. 21-23 1998.

[46] A. Cichocki and L. Zhang: Adaptive multichannel blind deconvolution using state-space models, *Proc of '99 IEEE Workshop on Higher-Order Statistics*, pp. 296–299, Caesarea, Israel, June 14-16 1999.

[47] A. Cichocki, L. Zhang, and T. Rutkowski: Blind separation and filtering using state space models, *Proc of '99 Int. Symposium on Circuits and Systems, ISCAS'99*, pp. V–78–81, Orlando, Florida, May 30-June 2 1999.

[48] L. Zhang, A. Cichocki, and S. Amari: Multichannel blind deconvolution of nonminimum phase systems using information backpropagation, *Proc. of the Fifth International Conference on Neural Information Processing (ICONIP'99)*, pp.210-216, Perth, Australia, Nov. 16-20, 1999.

[49] S. Amari: *Differential–geometrical methods in statistics, Lecture Notes in Statistics*, Vol. 28, Springer, Berlin, 1985.

[50] S. Amari and H. Nagaoka: *Methods of Information Geometry*, AMS and Oxford University Press, 1999.

[51] P. Comon: Independent component analysis, *Proc. Int. Workshop onHigher-order Statistics*, pp. 111–120, Chamrouse, France, 1991.

[52] A. Stuart and J. Ord: *Kendall's Advanced Theory of Statistics*, Edward Arnold, 1994.

[53] P. Comon: Contrasts for multichannel blind deconvolution, *Signal Processing Letters*, Vol. 3, No. 7, pp.209–211, 1996.

[54] S. Amari, T. Chen, and A. Cichocki: Stability analysis of adaptive blind source separation, *Neural Networks*, Vol. 10, No. 8, pp.1345–1351, 1997.

[55] S. Douglas, A. Cichocki, and S. Amari: Multichannel blind separation and deconvolution of sources with arbitrary distributions, *Proc. of IEEE Workshop on Neural Networks for Signal Processing(NNSP'97)*, pp. 436–445, Florida, US, September 1997.

[56] S. Amari and M. Kawanabe: Information geometry of estimating functions in semiparametric statistical models, *Bernoulli*, Vol. 3, No. 1, pp.29–54, 1997.

[57] S. Amari and M. Kawanabe: Estimating functions in semiparametric statistical models, In I. V. Basawa, V. Godambe, and R. Taylor, editors, *Estimating Functions*, Vol. 32 of *Monograph Series*, pp. 65–81. IMS, 1998.

[58] L. Zhang, S. Amari, and A. Cichocki: Semiparametric model and superefficiency in blind deconvolution, *Submitted to Signal Processing*, 1999.

[59] M. S. Grewal and A. P. Andrews: *Kalman Filtering: Theory and Practice*, Prentice Hall, 1993.

[60] R. Williams and D. Zipser: A learning algorithm for continually running fully recurrent neural networks, *Neural Computation*, Vol. 1, No. 2, pp.270–280, 1989.

[61] V. Vapnik: *Statistical Learning Theory*, John Wiley & Sons, 1998.

Liqing Zhang received the B.S. degree in Mathematics from Hangzhou University, and the Ph.D. degree in Computer Science from Zhongshan University, China, in 1983 and 1988 respectively. In 1988, he joined the Department of Automation at South China University of Technology, as a lecturer. From 1990 to 1995, he was an Associate Professor, in 1995 he was promoted to a full Professor in the College of Electronic and Information Engineering at South China University of Technology. He is currently working in the laboratory for Open Information Systems, RIKEN Brain Science Institute (Japan) as a Frontier Researcher. His research interests include Learning Theory, Statistical Signal Processing, Mathematical Theory for Neural Networks and Intelligent Systems. He is author or coauthor of more than 60 scientific papers in journals and conference proceedings.

Andrzej Cichocki received the M.Sc. (with honors), Ph.D., and Habilitate Doctorate (Dr.Sc.) degrees, all in electrical engineering, 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 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 (Teubner-Wiley,1993). He spent at University Erlangen-Nuernberg (Germany) a few years as Alexander Humboldt Research Fellow and Guest Professor. He is currently working as a head of the laboratory for Open Information Systems in the Brain Science Institute RIKEN (Japan), in the Brain-Style Information Processing Group directed by by Professor Shun-ichi Amari. His current research interests include neural networks, signal and image processing of biomedical data and nonlinear dynamic systems theory.