NATURAL GRADIENT APPROACH TO BLIND DECONVOLUTION OF DYNAMICAL SYSTEMS

L. Zhang and A. Cichocki

Laboratory for Open Information Systems
RIKEN Brain Science Institute
Hirosawa 2-1, Wako shi,
Saitama 351-0198, JAPAN
Email: *zha@open.brain.riken.go.jp*

ABSTRACT

In this paper we present a natural gradient approach to blind deconvolution of dynamical systems, described by the state space model. First we formulate blind deconvolution problem in the framework of the state space model. The blind deconvolution is fulfilled in two stages: internal representation and output separation, which give a new insight into blind deconvolution in the state-space framework. The cost function for blind deconvolution is discussed and adaptive natural gradient learning algorithms for updating external parameters are developed by minimizing a certain cost function, which is derived from mutual information of output signals. Stability of the algorithm is also given. Finally computer simulations are given to show the validity and effectiveness of the state-space approach.

1. INTRODUCTION

Blind separation/deconvolution of source signals has been a subject under consideration for more than a decade [2, 6, 8, 10, 11, 16, 19, 20, 22, 25]. 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). 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 [3, 8, 9, 16].

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 [21] is a new generalized model for blind separation and deconvolution. There are several reasons why the state-space models are advantageous for blind deconvolution. 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 deconvolution was discussed by Salam et al [18, 24], Zhang et al [26, 28, 29], and Cichocki et al [12]-[15]. An efficient learning algorithm was developed by Zhang, Cichocki and Amari [26], [28] 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 [29] was also proposed by using the Kalman filter. Cichocki et al extended the state space approach to nonlinear system [14], and an effective two-stage learning algorithm was presented [12] for training the parameters in demixing models. 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 PROBLEM FORMULATION

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

$$\overline{\mathbf{x}}(k+1) = \overline{\boldsymbol{\mathcal{F}}}(\overline{\mathbf{x}}(k), \mathbf{s}(k), \overline{\boldsymbol{\xi}}_P(k)), \quad (1)$$

$$\mathbf{u}(k) = \overline{\mathbf{C}}\overline{\mathbf{x}}(k) + \overline{\mathbf{D}}\mathbf{s}(k) + \overline{\boldsymbol{\theta}}(k). \quad (2)$$

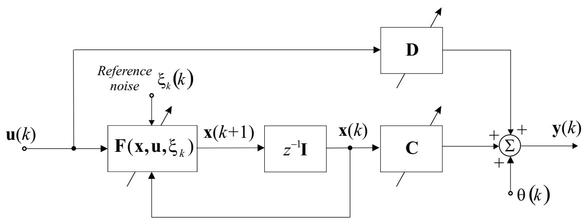

Figure 1: Illustration of the structure of state space model

where $\overline{\boldsymbol{\mathcal{F}}}$ is an unknown nonlinear mapping, $\overline{\mathbf{x}}(k) \in \mathbf{R}^N$ 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. $\overline{\boldsymbol{\xi}}_P(k)$ and $\overline{\boldsymbol{\theta}}(k)$ are the process noises and sensor noises of the mixing system, respectively. The output equation is assumed to be linear. In this paper, we present another dynamic system as a demixing model

$$\begin{aligned} \mathbf{x}(k+1) &= \boldsymbol{\mathcal{F}}_N(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta}, \boldsymbol{\xi}_P(k)), && (3) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \boldsymbol{\theta}(k), && (4) \end{aligned}$$

where $\mathbf{u}(k) \in \mathbf{R}^n$ is the available 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. $\boldsymbol{\xi}_P(k)$ and $\boldsymbol{\theta}(k)$ are the process noises and output noises of the demixing system, respectively. The dimension M of the state vector is the order of the demixing system. See figure 1 for illustration of the structure in the demixing model.

Since the mixing system is completely unknown, we neither know the nonlinear mappings $\overline{\boldsymbol{\mathcal{F}}}$, 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 overestimated, i.e. $M > N$. The overestimation of the order M may produce delays in output signals, but this is acceptable in blind deconvolution. A number of neural network models, such as Radial Based Function, Support Vector Machine and multilayer perceptron, can be used as demixing models. If the mapping $\boldsymbol{\mathcal{F}}_N$ in the state equation is linear, the nonlinear state space model will reduce to the linear generalized multichannel blind deconvolution.

2.1. 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,

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

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

$$\begin{aligned} \mathbf{x}(k+1) &= \overline{\boldsymbol{\mathcal{F}}}(\mathbf{x}(k), \overline{\mathbf{D}}^{-1}(\mathbf{y}(k) - \overline{\mathbf{C}}\mathbf{x}(k))), && (7) \\ \mathbf{s}(k) &= \overline{\mathbf{D}}^{-1}(\mathbf{u}(k) - \overline{\mathbf{C}}\mathbf{x}(k)). && (8) \end{aligned}$$

This means that if the mixing model is expressed by (5) and (6), we can recover the source signals using the inverse system (7) and (8) if such inverse system exists and it is stable. There is an advantage to the state space model in that we do not need to inverse any nonlinear functions explicitly.

2.2. Internal Representation

The state space description [21] 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.

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 the state space framework, we suggest that the separation of sources should be made in two stages: the first stage 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.

2.3. Canonical Forms

Canonical forms for linear systems are of great importance since they provide a unique state space representation of linear systems [21], [23]. 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.

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

$$\mathbf{H}(z) = \mathbf{P}(z)\mathbf{Q}^{-1}(z), \qquad (9)$$

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.

$$\mathbf{A} = \begin{bmatrix} -\mathcal{Q} & -\mathbf{Q}_N \\ \mathbf{I}_{n(N-1)} & \mathcal{O} \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} \mathbf{I}_n \\ \mathcal{O} \end{bmatrix}, \quad (10)$$

$$\mathbf{C} = (\mathbf{P}_1 \ \mathbf{P}_2 \ \cdots \ \mathbf{P}_N), \ \ \mathbf{D} = \mathbf{P}_0, \quad (11)$$

where $\mathcal{Q} = (\mathbf{Q}_1 \ \mathbf{Q}_2, \ \cdots, \ \mathbf{Q}_{N-1})$ is an $n \times n(N-1)$ matrix, $\mathcal{O}$ 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,

$$\mathbf{AZ} + \mathbf{ZA}^* + \mathbf{BB}^* = \mathbf{0}, \quad (12)$$

$$\mathbf{A}^*\mathbf{Y} + \mathbf{YA} + \mathbf{C}^*\mathbf{C} = \mathbf{0}, \quad (13)$$

are such that $\mathbf{Y} = \mathbf{Z} = diag(\sigma_1, \sigma_2, \cdots, \sigma_n) := \mathbf{\Sigma}_n$, 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. Refer to [23] for the detailed realization of balanced canonical forms. Some existing results and learning algorithms of state space models for blind deconvolution are summarized in the table 1.

3. COST FUNCTION FOR BLIND DECONVOLUTION

Our objective is to train the demixing model such that the output signals are maximally spatially mutually independent and temporarily *i.i.d.*. In this paper we employ the Kullback-Leibler Divergence as a cost function, which defines an asymmetrical measure of two probability functions [1]. Let $p_{\mathbf{y}}(\mathbf{y})$ and $q_{\mathbf{y}}(\mathbf{y})$ denote two different probability density functions. The Kullback-Leibler Divergence between $p_{\mathbf{y}}(\mathbf{y})$ and $q_{\mathbf{y}}(\mathbf{y})$ is defined as as follows:

$$\mathcal{D}(p,q) = \int p_{\mathbf{y}}(y) \log \left(\frac{p_{\mathbf{y}}(\mathbf{y})}{q_{\mathbf{y}}(\mathbf{y})} \right) d\mathbf{y}. \quad (14)$$

The Kullback-Leibler Divergence can be presented in the context of differential geometry as the Riemannian metric in the space of the distributions [1]. Let $p_i(y_i)$ be the i-th marginal probability density function of component y_i, which is defined by

$$p_i(y_i) = \int p_{\mathbf{y}}(\mathbf{y}) d\mathbf{y}^{(i)}, \ i = 1, \cdots, n, \quad (15)$$

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

$$\mathcal{D}(p,q) = \int p_{\mathbf{y}}(y) \log \left(\frac{p_{\mathbf{y}}(\mathbf{y})}{\prod_{i=1}^n p_i(y_i)} \right) d\mathbf{y}, \quad (16)$$

or we rewrite it into the mutual information form

$$l(\mathbf{W}) = -H(\mathbf{y}, \mathbf{W}) + \sum_{i=1}^n H(y_i, \mathbf{W}), \quad (17)$$

where $\mathbf{W} = [\mathbf{\Theta}, \ \mathbf{C}, \ \mathbf{D}]$ is the set of parameters in the demixing model, and $H(\mathbf{y}, \mathbf{W}) = -\int p(\mathbf{y}, \mathbf{W}) \log p(\mathbf{y}, \mathbf{W}) d\mathbf{y}$, $H(y_i, \mathbf{W}) = -\int p_i(y_i) \log p_i(y_i) dy_i$. 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 $l(\mathbf{W}) = 0$. 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 $p_i(y_i)$. In fact, if the matrix $\mathbf{D}$ is nonsingular, we can express the entropy $H(\mathbf{y}, \mathbf{W})$ as Gradient

$$H(\mathbf{y}, \mathbf{W}) = -\log |det(\mathbf{D})| + const. \quad (18)$$

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

$$l(\mathbf{y}, \mathbf{W}) = -\log |det(\mathbf{D})| - \sum_{i=1}^n \log q(y_i), \quad (19)$$

where $q(y_i)$ is an estimation of the true probability density function of source signals.

4. NATURAL LEARNING ALGORITHM

In this section, we develop a learning algorithm to update the external parameters $\mathbf{W} = [\mathbf{C}, \mathbf{D}]$, given the parameters $\mathbf{\Theta}$ in the demixing model. In order to obtain an improved learning performance, we define a new search direction, which is related to the natural gradient, developed by Amari [1]. The natural gradient search scheme proposed by Amari [2], [1] 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 $l(\mathbf{y}, \mathbf{W})$.

For simplicity we suppose that the matrix $\mathbf{D}$ in the demixing model (4) is a nonsingular $n \times n$ matrix. Following Amari's derivation for the natural gradient method [5, 26], we can calculate the total differential as

$$dl(\mathbf{y}, \mathbf{W}) = -tr(d\mathbf{D}\mathbf{D}^{-1}) + \boldsymbol{\varphi}^T(\mathbf{y}) d\mathbf{y}, \quad (20)$$

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

$$\varphi_i(y_i) = -\frac{d \log q_i(y_i)}{dy_i} = -\frac{q_i'(y_i)}{q_i(y_i)}, \quad (21)$$

Table 1: List of State Space Models and Algorithms for Blind Deconvolution

Paper	Model	Algorithm
Zhang and Cichocki (1998) [28]	Linear State Space Model	$\Delta \mathbf{C} = -\eta \boldsymbol{\varphi}(\mathbf{y})\mathbf{x}^T$ $\Delta \mathbf{D} = \eta(\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{u}^T\mathbf{D}^T)\mathbf{D}$
Zhang and Cichocki (1998) [29]	Linear State Space Model	$\Delta \mathbf{C} = \eta((\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T)\mathbf{C} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{x}^T)$ $\Delta \mathbf{D} = \eta(\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T)\mathbf{D}$ *with Kalman Filter*
Cichocki and Zhang (1998) [14]	Nonlinear State space Model	$\Delta \mathbf{C} = \eta((\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T)\mathbf{C} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{x}^T)$ $\Delta \mathbf{D} = \eta(\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T)\mathbf{D}$
Cichocki and Zhang (1998) [12]	Nonlinear State space Model	Two stage Approach
Cichocki and Zhang (1999) [13]	Nonlinear State space Model	$\Delta \mathbf{C} = \eta((\mathbf{I} - \langle \boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T \rangle)\mathbf{C} - \langle \boldsymbol{\varphi}(\mathbf{y})\mathbf{x}^T \rangle \Lambda)$ $\Delta \mathbf{D} = \eta(\mathbf{I} - \langle \boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T \rangle)\mathbf{D}$
Erten and Salam (1999) [18]	Nonlinear State space Model	$\Delta \mathbf{C} = \eta(\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{x}^T)\mathbf{C}$ $\Delta \mathbf{D} = \eta(\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T)\mathbf{D}$
Salam and Erten (1999) [24]	Nonlinear State space Model	Lagrange Multiplier Approach
Zhang and Cichocki (*this paper*)	Nonlinear State Space Model	$\Delta[\mathbf{C}\ \mathbf{D}] = -\eta \nabla l \begin{bmatrix} \mathbf{I} + \mathbf{C}^T\mathbf{C} & \mathbf{C}^T\mathbf{D} \\ \mathbf{D}^T\mathbf{C} & \mathbf{D}^T\mathbf{D} \end{bmatrix}$

Taking a differential of $\mathbf{y}$ in equation (4), we have the following relation

$$d\mathbf{y} = d\mathbf{C}\mathbf{x}(k) + d\mathbf{D}\mathbf{u}(k). \quad (22)$$

In this paper we derive the extended natural gradient by introducing a new search direction. From linear output equation (4), we have

$$\mathbf{u}(k) = \mathbf{D}^{-1}(\mathbf{y}(k) - \mathbf{C}\mathbf{x}(k)). \quad (23)$$

Substituting (23) into (22), we obtain

$$d\mathbf{y} = (d\mathbf{C} - d\mathbf{D}\mathbf{D}^{-1}\mathbf{C})\mathbf{x} + d\mathbf{D}\mathbf{D}^{-1}\mathbf{y}. \quad (24)$$

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

$$d\mathbf{X}_1 = d\mathbf{C} - d\mathbf{D}\mathbf{D}^{-1}\mathbf{C}, \quad (25)$$
$$d\mathbf{X}_2 = d\mathbf{D}\mathbf{D}^{-1}. \quad (26)$$

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

$$\frac{\partial l(\mathbf{y}, \mathbf{W})}{\partial \mathbf{X}_1} = \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{x}^T(k), \quad (27)$$
$$\frac{\partial l(\mathbf{y}, \mathbf{W})}{\partial \mathbf{X}_2} = \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k) - \mathbf{I}. \quad (28)$$

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

$$\Delta \mathbf{X}_1(k) = -\eta \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{x}^T(k), \quad (29)$$
$$\Delta \mathbf{X}_2(k) = -\eta(\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k) - \mathbf{I}). \quad (30)$$

where η is a learning rate. From (25) and (26), we obtain a novel learning algorithm to update matrices $\mathbf{C}$ and $\mathbf{D}$ as

$$\Delta \mathbf{C}(k) = \eta\left((\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T)\mathbf{C} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{x}^T\right), \quad (31)$$
$$\Delta \mathbf{D}(k) = \eta\left(\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y})\mathbf{y}^T\right)\mathbf{D}. \quad (32)$$

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

$$\tilde{\nabla} l = \nabla l \begin{bmatrix} \mathbf{I} + \mathbf{C}^T\mathbf{C} & \mathbf{C}^T\mathbf{D} \\ \mathbf{D}^T\mathbf{C} & \mathbf{D}^T\mathbf{D} \end{bmatrix}. \quad (33)$$

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

$$[\Delta \mathbf{C}\ \Delta \mathbf{D}] = -\eta(k)\tilde{\nabla} l(\mathbf{y}, \mathbf{W}). \quad (34)$$

It is easy to see that the preconditioning matrix

$$\begin{bmatrix} \mathbf{I} + \mathbf{C}^T\mathbf{C} & \mathbf{C}^T\mathbf{D} \\ \mathbf{D}^T\mathbf{C} & \mathbf{D}^T\mathbf{D} \end{bmatrix}.$$

is symmetric positive definite, and this expression is the extension of Amari's natural gradient to the state space model. From (31) and (32), we see that the natural gradient learning algorithm [6] is covered as a special case of the learning algorithm when the mixture is simplified to an instantaneous case.

The algorithm includes an unknown score function $\boldsymbol{\varphi}(\mathbf{y})$. The optimal one is given by equation (21) 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 $\varphi_i(y_i) = y_i^3$ [4, 17]. A question will be raised as to whether the learning algorithm will converge to a true solution if the estimated score functions are used. The theory of the semi-parametric model for blind separation/deconvolution ([3], [7], [27]) 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 [4].

5. STABILITY OF LEARNING ALGORITHM

In this section we discuss the stability of the natural gradient algorithm (31) and (32). Since the algorithm is derived from (29) and (30), we only need to discuss the stability of (29) and (30). Consider its learning rule

$$\Delta \mathbf{X}_1(k) = -\eta \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{x}^T(k), \quad (35)$$
$$\Delta \mathbf{X}_2(k) = -\eta(\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k) - \mathbf{I}). \quad (36)$$

The equilibrium points of the dynamical system satisfy

$$E[\boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{x}^T(k)] = 0, \quad (37)$$
$$E\left[\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(k))\mathbf{y}^T(k)\right] = 0. \quad (38)$$

Clearly, the true solution $\mathbf{C}$ and $\mathbf{D}$ is the solution of (37) and (38). 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 (31) and (32), 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 (35) and (36) under certain conditions on the source signals.

Consider the average version of the learning algorithm

$$\Delta \mathbf{X}_1(k) = \eta \mathbf{F}_1(\mathbf{X}), \quad (39)$$
$$\Delta \mathbf{X}_2(k) = \eta \mathbf{F}_2(\mathbf{X}), \quad (40)$$

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

$$\Delta\delta\mathbf{X}_1(k) = \eta\left(\frac{\partial \mathbf{F}_1}{\partial \mathbf{X}_1}\delta\mathbf{X}_1 + \frac{\partial \mathbf{F}_1}{\partial \mathbf{X}_2}\delta\mathbf{X}_2\right), \quad (41)$$
$$\Delta\delta\mathbf{X}_2(k) = \eta\left(\frac{\partial \mathbf{F}_2}{\partial \mathbf{X}_1}\delta\mathbf{X}_1 + \frac{\partial \mathbf{F}_2}{\partial \mathbf{X}_2}\delta\mathbf{X}_2\right). \quad (42)$$

From the definition of $\mathbf{X}_1$ and $\mathbf{X}_2$, we have

$$\delta\mathbf{y}(k) = \delta\mathbf{X}_1\mathbf{x}(k) + \delta\mathbf{X}_2\mathbf{y}(k). \quad (43)$$

Using the i.i.d. property of $\mathbf{y}(k)$ and (43), we derive the variational equation of $\mathbf{X}_1$

$$\Delta\delta\mathbf{X}_{1,i} = -\eta E\left[\varphi(y_i)\right] E[\mathbf{x}\mathbf{x}^T]\delta\mathbf{X}_{1,i}, \quad (44)$$

for $i = 1, \cdots, n$, where $\mathbf{X}_{1,i}$ is the vector of the i-th row of matrix $\mathbf{X}_1$. We take the following notation

$$\sigma_i^2 = E[y_i^2], \ \kappa_i = E[\varphi_i'(y_i)], \ m_i = E[y_i^2\varphi_i(y_i)]. \quad (45)$$

where $\varphi_i' = \frac{d\varphi_i}{dy_i}$. We can easily derive the stability conditions for (44). If $\kappa_i > 0$ 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 (44), we derive the variational equation of $\mathbf{X}_2$

$$\Delta\delta X_{2,ij} = -\eta\left(\kappa_i\sigma_j^2\delta X_{2,ij} + \delta X_{2,ji}\right), \quad (46)$$
$$\Delta\delta X_{2,ji} = -\eta\left(\kappa_j\sigma_i^2\delta X_{2,ji} + \delta X_{2,ij}\right), \quad (47)$$

for $i \neq j$, and $i, j = 1, \cdots, n$, and

$$\Delta\delta X_{2,ii} = -\eta(m_i + 1)\delta X_{2,ii}, \quad (48)$$

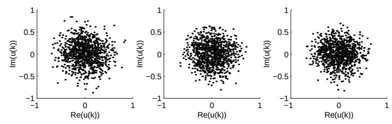

Figure 2: Sensor signal constellations

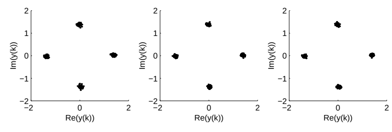

Figure 3: Output constellations

for $i = 1, \cdots, n$. Therefore, the stability conditions for (46)-(48) are summarized as

$$m_i + 1 > 0, \ \kappa_i > 0, \ \kappa_i\kappa_j\sigma_i^2\sigma_j^2 > 1, \quad (49)$$

for $i, j = 1, \cdots, n$. Stability conditions have similar form to the one given by Amari et al [4]. In summary we have the following theorem

Theorem 1 *If the covariance matrix $E[\mathbf{x}\mathbf{x}^T]$ is positive definite and the conditions (49) 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. If the mixing system is controllable, then the matrix

$$\left[\mathbf{B}\ \mathbf{A}\mathbf{B}\ \cdots\ \mathbf{A}^{N-1}\mathbf{B}\right] \quad (50)$$

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

6. COMPUTER SIMULATIONS

A number of computer simulations have been performed to demonstrate the validity and effectiveness of the natural gradient algorithm for generalized blind deconvolution. Due to the limited space, we only give an illustrative example. The mixing model used for computer simulation is the 3-channel ARMA model, of which the parameters 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. Figures 2 and 3 plot the sensor signal constellations and output constellations of the demixing model.

7. CONCLUSION

In this paper we have presented a natural gradient algorithm of the state space approach for multichannel blind deconvolution. The state space formulation allows us to separate blind deconvolution in two steps: internal information representation and unsupervised learning for external parameters. The internal parameters can be trained either by supervised learning or by information backpropagation approach. An state estimator based on the Kalman filter can also be used in order to amend the model bias and reduce the effect of noise. An illustrative simulation is given to demonstrate the validity and effectiveness of the state-space approach.

8. REFERENCES

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

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

[3] S. Amari and J.-F. Cardoso. Blind source separation–semiparametric statistical approach. *IEEE Trans. Signal Processing*, 45:2692–2700, Nov. 1997.

[4] S. Amari, T. Chen, and A. Cichocki. Stability analysis of adaptive blind source separation. *Neural Networks*, 10:1345–1351, 1997.

[5] S. Amari and A. Cichocki. Adaptive blind signal processing– neural network approaches. *Proceedings of the IEEE*, 86(10):2026–2048, 1998.

[6] 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)*, pages 757–763, Cambridge, MA, 1996. The MIT Press.

[7] S. Amari and M. Kawanabe. Information geometry of estimating functions in semiparametric statistical models. *Bernoulli*, 3(1):29–54, 1997.

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

[9] J.-F. Cardoso. Blind signal separation: Statistical principles. *Proceedings of the IEEE*, 86(10):2009–2025, 1998.

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

[11] A. Cichocki, R. Unbehauen, and E. Rummert. Robust learning algorithm for blind separation of signals. *Electronics Letters*, 30(17):1386–1387, 1994.

[12] A. Cichocki and L. Zhang. Two-stage blind deconvolution using state-space models (invited). In *Proceedings of ICONIP'98* , pages 729–732, Kitakyushu, Japan, Oct. 21-23 1998.

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

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

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

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

[17] S. Douglas, A. Cichocki, and S. Amari. Multichannel blind separation and deconvolution of sources with arbitrary distributions. In *Proceeding of NNSP'97* , pages 436–445, Florida, US, September 1997.

[18] G. Erten and F. Salam. Voice extraction by on-line signal separation and recovery. *IEEE Trans. Circuits and Systems II*, 46(7):915–922, 1999.

[19] A. Hyvarinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

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

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

[22] J. Karhunen, P. Pajunen, and E. Oja. The nonlinear PCA criterion in blind source separation: relations with other approaches. *Neurocomputing*, 22:5–20, 1998.

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

[24] F. Salam and G. Erten. The state space framework for blind dynamic signal extraction and recovery. In *Proc. of ISCAS'99*, pages V–66 –69, Orlando, Florida, May 30-June 2 1999.

[25] J. Tugnait and B. Huang. Multistep linear predictors-based blind identification and equalization of multiple-input multiple-output channels. *IEEE Trans. Signal Processing*, 48:26–38, 2000.

[26] L. Zhang, S. Amari, and A. Cichocki. Natural gradient approach to blind separation of over- and undercomplete mixtures. In *Proc. of ICA'99* , pages 455–460, Aussois, France, 1999.

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

[28] L. Zhang and A. Cichocki. Blind deconvolution/ equalization using state-space models. In *Proceeding of NNSP'98* , pages 123–131, Cambridge, UK, August 31-September 2 1998.

[29] 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*, volume 11, pages 648–654. MIT press, Cambridge, MA, 1999.