SPARSE REPRESENTATION AND BLIND DECONVOLUTION OF DYNAMICAL SYSTEMS

Liqing Zhang and Andrzej Cichocki

Lab. for Advanced Brain Signal Processing
RIKEN Brain Science Institute
Wako shi, Saitama, 351-0198, Japan
{zha,cia}@bsp.brain.riken.go.jp

ABSTRACT

In this paper, we discuss 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 signal separation. We employ two different learning strategies for training the parameters in the two stages. A sparse representation approach is presented based on the independent decomposition. Some properties of the sparse representation approach are discussed. The natural gradient algorithm is used to train the external parameters in the stage of signal separation. The two-stage approach provides a new insight into blind deconvolution in the state-space framework. Finally a computer simulation is given to show the validity and effectiveness of the state-space approach.

1. INTRODUCTION

Blind separation/deconvolution has attracted considerable attention in various areas such as telecommunications, speech recognition, image enhancement and biomedical signal processing [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. The blind 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.

Although there exist a number of models and methods for separating blindly independent sources, there still exist several challenges in generalizing mixtures to dynamic and nonlinear systems, as well as in developing more rigorous and efficient 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 model has been developed for blind source separation and blind deconvolution by Salam et al [11, 12], Zhang et al [13, 14], and Cichocki et al [15]-[16]. In the state space formulation, the parameters of demixing model are divided into two types [17]: the internal parameters and external parameters. The internal parameters are independent of the individual signal separation problem; they are usually trained in an off-line manner, using a set of signal separation problems. In contrast, the external parameters are trained individually for each separation problem. Thus, the internal and external parameters are trained in different ways. The natural gradient algorithm [13],[17] was developed to train the output matrices by minimizing the mutual information. The state space approach was also extended to nonlinear system, and an effective two-stage learning algorithm was presented [15, 17] for training the parameters in demixing models. Furthermore, the Kalman filter was employed to compensate for the model bias and reduce the effect of noise [14, 18].

However, the training strategy for the internal parameters still remains open. It is the main purpose of this paper to develop a learning algorithm for training the internal parameters. The remainder of this paper is organized as follows: We present the general formulation of the blind deconvolution in dynamical environment in section 2. The neural network structure for internal representation is given in section 3. A new learning strategy, the independent decomposition, is proposed and some properties of the learning strategy are discussed in section 4. A computer simulation is given to show the validity of the proposed method.

The state-space description of systems is a new generalized model for blind separation and deconvolution. The main advantage of the state space description for blind deconvolution is that it gives the internal description of a system, which enables us to realize the two-stage approach for blind deconvolution. Recently, some biological experiments indicate that primary visual cortex (area V1) uses a sparse code to efficiently represent natural scenes [19],[20]. This provides an evidence to support that the sparse representation is biologically plausible for training a neural network.

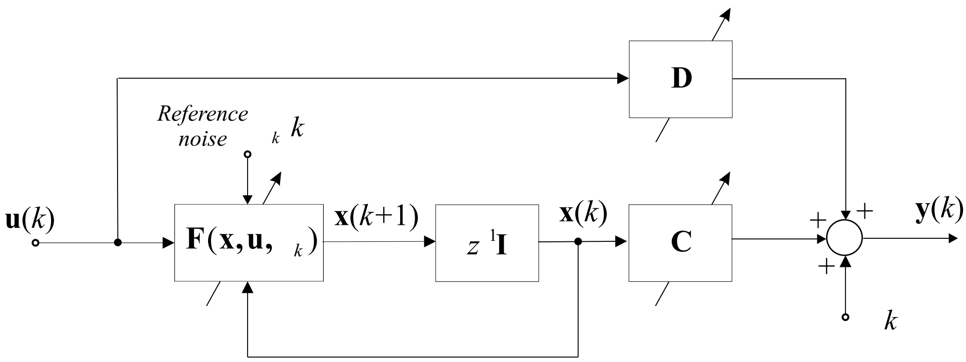

Fig. 1. Illustration of the structure of state space model for blind deconvolution

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

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

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

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

$$\mathbf{x}(k+1) = \boldsymbol{\mathcal{F}}_N(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta}, \boldsymbol{\xi}_P(k)), \quad (3)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \boldsymbol{\theta}(k), \quad (4)$$

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 the source signals $\mathbf{s}(k)$ 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. 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.

The unknown parameter $\boldsymbol{\Theta}$ in equation (3) is referred to as the internal parameter, and $[\mathbf{C}\ \mathbf{D}]$ in equation (4) as the external parameter, respectively. The learning strategy for the internal and external parameter is different. We should employ the blind deconvolution algorithm [14, 17] for the external parameter $[\mathbf{C}\ \mathbf{D}]$. However, in this paper, the learning algorithm for internal parameter $\boldsymbol{\Theta}$ will use a different strategy: the sparse representation.

3. NEURAL NETWORK FOR INTERNAL REPRESENTATION

In this section, we introduce the architecture of neural network for the internal representation of the dynamical system. The neural network in this paper is topologically similar to the multi-layer perceptron (MLP). The mechanism of learning for the network in this paper is essentially different from the one in MLP. The neural network is designed as a hierarchical competitive-cooperative network, which consists of N layers, say $N = 4$. In general, the forward connections to a neuron in one layer are topologically linked to related region of the preceding layer. For simplicity, we impose extra constraints on the connections. Each neuron receives connections from L neurons in the preceding layer, where L is a given number. We choose L=12 in the simulations. Figure 2 illustrates the general architecture of the neural network used in this paper.

The input to the neural network is denoted by $\mathbf{u}(k)$, and the neural state (the potential) of p-th layer is denoted by $\mathbf{x}^p(k)$. The number of neurons at p-th layer is denoted by M_p. Assume that v_i^p is the output of $j-$th neuron at $p-$th layer , given by

$$v_i^p(k) = \varphi(x_i^p(k)). \quad (5)$$

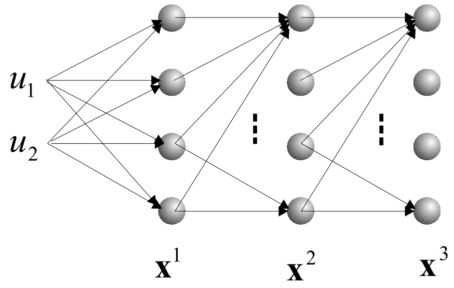

Fig. 2. The neural network architecture for internal representation

In general, the potential $x_i^p(k)$ can be described by a recurrent neural network

$$x_i^p(k+1) \quad = \quad \lambda_i^p x_i^p(k) + u_i^p(k) + \epsilon_i(k) \qquad (6)$$

where λ_i^p is a certain decay factor of i-th neuron at layer p, ϵ_i is the noise term and $u_i^p = \sum_{j=1}{}^{M_p} \mathbf{W}_{ij}^p v_j^{p-1}$ is the input to the i-th neuron at $p-$th layer. $\mathbf{W}_{ij}^p$ is the connection weight between the output of neuron j at $(p-1)$-th layer and the input of neuron i at p-th layer. If $p = 1, y_j^0 = u_j$, where $\mathbf{u} = (u_1, \cdots, u_n)^T$ is the input applied to the network. We can also rewrite the neural network in a compact form

$$\mathbf{v}^p(k) = \boldsymbol{\varphi}(\mathbf{x}^p(k)) \qquad (7)$$

The state vector of p-th layer can be described by

$$\mathbf{x}^p(k+1) = \boldsymbol{\Lambda}^p \mathbf{x}^p(k) + \mathbf{u}^p(k) + \boldsymbol{\epsilon}^p(k) \qquad (8)$$

where $\boldsymbol{\Lambda}^p = diag(\lambda_1^p, \cdots, \lambda_{M_p}^p)$ and $\mathbf{u}^p = \mathbf{W}^p \mathbf{v}^{p-1}$.

The connection weight $\mathbf{W}^p, p = 1, \cdots, N$ are the parameters to be determined during training. The set of the connection $\{ \mathbf{W}^p, p = 1, \cdots, N\}$ is denoted by $\boldsymbol{\Theta}$ as in equation (3). The learning strategy in this paper is essentially different from the error backpropagation approach. Our purpose is to train the neural network such that the internal states have maximally sparse representation. Thus the source signals can be linearly expressed by the external parameters.

4. INDEPENDENT DECOMPOSITION AND SPARSE REPRESENTATION

In this section, we should present a sparse representation approach for training the internal parameter. According to recent findings [19], primary visual cortex (area V1) uses a sparse code to efficiently represent natural scenes. Another finding is that retinal ganglion neurons act largely independently to encode information [20]. This provides an evidence to support that the sparse representation is biologically plausible for training a neural network.

First, we present an independent decomposition method of random variables, which is related to independent component analysis. Different from the model for blind source separation, we do not assume that the sensor signals, denoted by $\mathbf{u}(k)$ are generated by a linear instantaneous model. The purpose of independent decomposition is to train a linear transform $\mathbf{W}^p$ such that

$$\mathbf{u}^p(k) = \mathbf{W}^p \mathbf{v}^{p-1}(k) \qquad (9)$$

are maximally mutually independent. To this end, we can employ ICA learning algorithms, such as the natural gradient algorithm [5] and equivariant algorithm [8], to train the transform $\mathbf{W}^p$. In this paper, we use the natural gradient algorithm, which is described by

$$\mathbf{W}^p(k+1) = \mathbf{W}^p(k) - \eta(k)[\mathbf{I} - \boldsymbol{\varphi}(\mathbf{u}^p)(\mathbf{u}^p)^T]\mathbf{W}^p. \quad (10)$$

where $\boldsymbol{\varphi}(\mathbf{u}^p) = [\varphi_1(u_1^p), \ldots, \varphi_m(u_m^p)]^T$ is a vector of certain activation functions. The matrix $\mathbf{W}^p$ is not necessary to be square. If it is rectangular, we should replace algorithm (10) with the natural gradient approach for over- and under-complete mixture [13] to train the matrix $\mathbf{W}^p$. The activation functions, which are closely related to the distributions of the output signals, are crucial to the independent decomposition. Different choice may lead to a different transform matrix $\mathbf{W}$. In order to make the output signals as sparse as possible, we should choose the activation functions derived from certain sparse probability density functions. For example, we can choose $\varphi(u) = tanh(u)$, assuming the pdf $p(u) = \frac{c}{e^u + e^{-u}}$, where c is the normalization factor. This is a heaver-tailed distribution for the latent variable than the Gaussian distribution. Other distribution models, such as the Laplacian, can also be used here.

Now, we are going to show that the independent decomposition will give a spatially sparse coding strategy. For simplicity, We assume that the internal system is described by a multi-layer perceptron, i.e. equilibrium state of equation (6). Supposed that $u_i^p(k)$ is the neuron potential of the $i-$th neuron in the network. The neuron fires if the value $|u_i^p(k)|$ exceeds a certain threshold, say $\theta > 0$. As we know, the algorithm (10) will lead to each of the output signals as a temporally sparse signal, which means that only a few $|u_i^p(k)|$ in time sequence will exceed the threshold θ. On the other hand, the independency between two different neurons $u_i^p(k)$ and $u_j^p(k)$ will prevent them from firing at the same time. Thus, given time k, only a few neurons fire, according to the above analysis. Therefore, we infer that the independent decomposition will produce a sparse representation in neural networks.

If the input stream is very complex, the neuron responses in the first layer may not be mutually independent, even after sufficiently learning. Similarly, if we consider the neuron responses in first layer as the input to the second layer, we can develop a learning strategy to adjust the connections

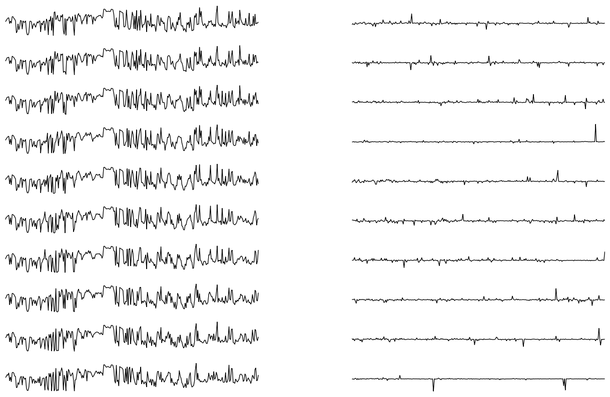

Fig. 3. a) The sensor signals $\mathbf{u}(k)$; b) The output $\mathbf{v}(k)$ after independent decomposition

between the first and the second layers, by making the neurons responses in the second layer maximally mutually independent. For the same reason, we can present learning algorithms for train the connections between other layers.

It should noted that the independent decomposition is closely related to the redundancy reduction [21]. The principle of redundancy reduction states that a useful goal of sensory coding is to transform the input in such a manner that reduces the redundancy due to statistical dependencies among elements of the input stream. The independent decomposition provides a way to realize the redundancy reduction for signal representation. Figure 3 shows clearly that there are plenty of redundancy in the input data (here the input data are ten components of a video stream) and after independent decomposition, the output signals are temporally sparse and maximally spatially independent.

4.1. Cooperative and Competitive Mechanism

The independent decomposition results in cooperation and competition between the neurons in the same layer in the following sense: If certain feature (an independent component) is represented by a neuron, then the independent decomposition algorithm will force other neurons not to represent the same feature, but other features (other independent components). This implies a competitive mechanism in learning. On the other hand, if certain feature is not represented in the neural network, the independent decomposition rule will push some neuron to represent the feature. This attributes to the cooperative mechanism in learning.

5. NATURAL LEARNING ALGORITHM

In this section, we employ the natural gradient 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 [22]. The natural gradient search scheme 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 $l(\mathbf{y}, \mathbf{W})$. The relation between the natural gradient and the ordinary gradient can be defined by [17]

$$\tilde{\nabla} l = \nabla l \begin{bmatrix} \mathbf{I} + \mathbf{C}^T\mathbf{C} & \mathbf{C}^T\mathbf{D} \\ \mathbf{D}^T\mathbf{C} & \mathbf{D}^T\mathbf{D} \end{bmatrix}. \tag{11}$$

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 natural gradient algorithm can be written in the following form

$$[\Delta\mathbf{C}\ \Delta\mathbf{D}] = -\eta(k)\tilde{\nabla} l(\mathbf{y}, \mathbf{W}). \tag{12}$$

Explicitly, we obtain an learning algorithm to update matrices $\mathbf{C}$ and $\mathbf{D}$ as

$$\Delta\mathbf{C}(k) = \eta\left((\mathbf{I} - \boldsymbol{\psi}(\mathbf{y})\mathbf{y}^T)\mathbf{C} - \boldsymbol{\psi}(\mathbf{y})\mathbf{x}^T\right), \tag{13}$$

$$\Delta\mathbf{D}(k) = \eta\left(\mathbf{I} - \boldsymbol{\psi}(\mathbf{y})\mathbf{y}^T\right)\mathbf{D}, \tag{14}$$

where $\boldsymbol{\psi}(\mathbf{y})$ is the vector of activation functions. 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.

The algorithm includes an unknown score function $\boldsymbol{\varphi}(\mathbf{y})$. The optimal one is given by $\psi_i(y_i) = -\frac{p_i'(y_i)}{p_i(y_i)}$, 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 $\psi_i(y_i) = tanh(y_i)$. Respectively, if it is sub-Gaussian, one can choose $\psi_i(y_i) = y_i^3$.

6. STATE ESTIMATOR – THE KALMAN FILTER

The Kalman filter is a powerful approach for estimating the state vector in state-space models. Therefore, it can be used to compensate for the model bias and to reduce the effect of noise. Consider the demixing model with noise terms,

$$\mathbf{x}(k+1) = \boldsymbol{\mathcal{F}}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{\Theta}, \mathbf{w}(k)) \tag{15}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \boldsymbol{\theta}(k) + \mathbf{v}(k) \tag{16}$$

where the random variables $\mathbf{w}(k)$ and $\mathbf{v}(k)$ represent the process and measurement noise. To estimate a state vector of the nonlinear system, we begin with the variational equation

$$\delta\mathbf{x}(k+1) \approx +\mathbf{A}_k\delta\mathbf{x}(k) + \mathbf{W}_k\mathbf{w}(k), \tag{17}$$

where $\mathbf{A}_k$ is the Jacobian matrix of partial derivatives of $\mathcal{F}$ with respect to $\mathbf{x}$, $\mathbf{A}_{k,ij} = \frac{\partial \mathcal{F}_i}{\partial x_j}(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta}, \mathbf{0})$, and $\mathbf{W}_k$ is the Jacobian matrix of partial derivatives of $\mathcal{F}$ with respect to $\mathbf{w}$, $\mathbf{W}_{k,ij} = \frac{\partial \mathcal{F}_i}{\partial w_j}(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\Theta}, \mathbf{0})$.

Suppose that the random variables of $\mathbf{w}$ and $\mathbf{v}$ have the following probability density functions

$$p(w(k)) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k), \; p(v(k)) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k). \qquad (18)$$

where $\mathbf{Q}_k$ and $\mathbf{R}_k$ are covariance matrices of $\mathbf{w}$ and $\mathbf{v}$. Given theses approximations, the Kalman filter equation used to the state $\hat{\mathbf{x}}$ is

$$\hat{\mathbf{x}} = \mathbf{x}(k) + \mathbf{K}_k \mathbf{r}(k), \qquad (19)$$

where the matrix $\mathbf{K}_k$ is called the Kalman gain. $\mathbf{r}(k)$ 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 [23] 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

$$\mathbf{r}(k) = \Delta\mathbf{y}(k) = \Delta\mathbf{C}\mathbf{x}(k) + \Delta\mathbf{D}\mathbf{u}(k), \qquad (20)$$

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 $\mathbf{K}$. Refer to [23] for the detailed updating algorithm.

7. COMPUTER SIMULATION

A number of computer simulations have been performed to demonstrate the validity and effectiveness of the two-stage approach for generalized blind deconvolution. Due to the limited space, we only give an illustrative example. The mixing model in this simulation is the 3-channel nonlinear neural network model, The source signals $\mathbf{s}$ are randomly generated binary signals and $\mathbf{v}$ are the Gaussian noises with zero mean and a covariance matrix $0.1\mathbf{I}$. The demixing model consists of two parts: the dynamical system and output system. The dynamical system is described by neural network (5) and (6) and the output system by equation (4).

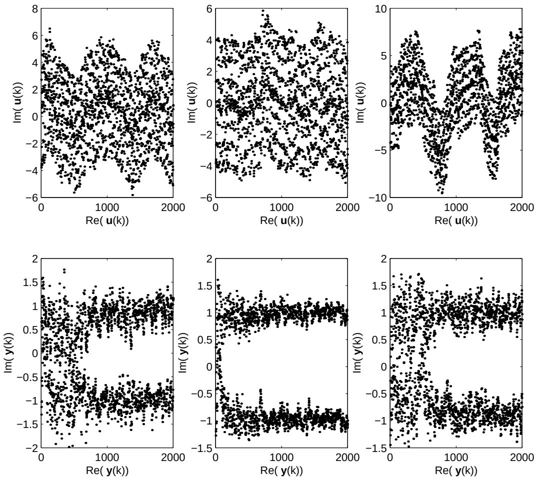

Fig. 4. First Row: Sensor signal constellations, Second Row: Reconstructed signal constellations

The internal system is trained by the independent decomposition approach, and the output system by the blind deconvolution algorithm (13) and (14), respectively, where the nonlinear activation function is chosen to be $\varphi_i(y_i) = y_i^3$ for any i. Figure 4 plots the sensor signal constellations and output constellations of the demixing model. From this simulation, we see that the two-stage approach can recover the binary sources from the dynamic nonlinear mixture.

8. CONCLUSION

In this paper, we have presented a two-stage approach to blind deconvolution of dynamical system, described by the state space model. The state space formulation allows us to separate blind deconvolution in two steps: internal information representation and unsupervised learning for external parameters. In this paper, we use the independent decomposition method for the internal information representation, which give a sparse representation of the neural network. The independent decomposition also provides a way to realize cooperative and competitive learning. An illustrative simulation is given to demonstrate the validity and effectiveness of the state-space approach.

9. REFERENCES

[1] 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.

[2] L. Tong, R.W. Liu, V.C. Soon, and Y.F. Huang, “Indeterminacy and identifiability of blind identification,”

IEEE Trans. Circuits, Syst., vol. 38, no. 5, pp. 499–509, May 1991.

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

[4] 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.

[5] S. Amari, A. Cichocki, and H.H. Yang, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems 8 (NIPS*95)*, G. Tesauro, D.S. Touretzky, and T.K. Leen, Eds., Cambridge, MA, 1996, The MIT Press, pp. 757–763.

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

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

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

[9] A. Hyvarinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Computation*, vol. 9, no. 7, pp. 1483–1492, 1997.

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

[11] G. Erten and F. Salam, "Voice extraction by on-line signal separation and recovery," *IEEE Trans. Circuits and Systems II*, vol. 46, no. 7, pp. 915–922, 1999.

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

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

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

[15] A. Cichocki and L. Zhang, "Two-stage blind deconvolution using state-space models (invited)," in *Proceedings of the Fifth International Conference on Neural Information Processing(ICONIP'98)*, Kitakyushu, Japan, Oct. 21-23 1998, pp. 729–732.

[16] 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*, Orlando, Florida, May 30-June 2 1999, pp. V–78–81.

[17] L. Zhang and A. Cichocki, "Natural gradient approach to blind deconvolution of dynamical systems," in *Proceeding of ICA'2000*, Helsinki, Finland, 2000, pp. 27–32.

[18] L. Zhang, A. Cichocki, and S. Amari, "Kalman filter and state-space approach to blind deconvolution," in *Proceeding of NNSP'2000*, Sydney, Australia, 2000, p. aceppted.

[19] W. E. Vinje and J. L. Gallant, "Sparse coding and decorrelation in primary visual cortex during natural vision," *Science*, vol. 287, pp. 1273–1276, 2000.

[20] S. Nirenberg, S. M. Carciert, A. L. Jacobs, and P. E. Latham, "Retinal ganglion cells act largely as independent encoders," *Nature*, vol. 411, pp. 698–701, 2001.

[21] H. B. Barlow, "Unsupervised learning," *Neural Computation*, vol. 1, pp. 295–311, 1989.

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

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