Nonlinear Blind Source Separation Using a Radial Basis Function Network

Ying Tan, *Member, IEEE*, Jun Wang, *Senior Member, IEEE*, and Jacek M. Zurada, *Fellow, IEEE*

***Abstract*—This paper proposes a novel neural-network approach to blind source separation in nonlinear mixture. The approach utilizes a radial basis function (RBF) neural-network to approximate the inverse of the nonlinear mixing mapping which is assumed to exist and able to be approximated using an RBF network. A contrast function which consists of the mutual information and partial moments of the outputs of the separation system, is defined to separate the nonlinear mixture. The minimization of the contrast function results in the independence of the outputs with desirable moments such that the original sources are separated properly. Two learning algorithms for the parametric RBF network are developed by using the stochastic gradient descent method and an unsupervised clustering method. By virtue of the RBF neural network, this proposed approach takes advantage of high learning convergence rate of weights in the hidden layer and output layer, natural unsupervised learning characteristics, modular structure, and universal approximation capability. Simulation results are presented to demonstrate the feasibility, robustness, and computability of the proposed method.**

***Index Terms*—Blind source separation, nonlinear mixtures, radial basis function (RBF) neural networks, statistical independence, unsupervised learning.**

I. Introduction

RECENTLY, blind source separation in signal processing has received considerable attention from researchers, due to its numerous promising applications in the areas of communications and speech signal processing [1], [2], medical signal processing including ECG, MEG, and EEG [3], and monitoring [4]. A number of blind separation algorithms have been proposed based on different separation models [5]–[8]. These algorithms play increasingly important roles in many applications. The study of blind signal processing techniques is of both theoretical significance and practical importance.

Blind source separation is to recover unobservable independent sources (or "signals") from multiple observed data masked by linear or nonlinear mixing. Most existing algorithms for linear mixing models stem from the theory of the independent component analysis (ICA) [9]–[11]. ICA is a statistical technique whose goal is to represent a set of random variables as linear functions of statistically independent component variables. Therefore, a solution to blind source separation problem exists and this solution is unique up to some trivial indeterminacies (permutation and scaling) according to the basic ICA theory [9]. Even though the nonlinear mixing model is more realistic and practical, most existing blind separation algorithms developed so far are valid for linear models. For nonlinear mixing models, many difficulties occur and both the linear ICA theory and existing linear demixing algorithms are no longer applicable because of the complexity of nonlinear characteristics. In addition, there is no guarantee for the uniqueness of the solution of nonlinear blind source separation unless additional constraints are imposed on the mixing transformation [12].

So far several authors studied the difficult problem of the nonlinear blind source separation and proposed a few efficient demixing algorithms [12]–[15], [17]–[21]. Deco [13] studied a very particular scenario of volume-conserving nonlinear transforms. Pajunen *et al.*, Herrmann, and Lin [14], [17], [18] proposed model-free methods which used Kohonen's self-organizing map (SOM) to extract independent sources from nonlinear mixture, but suffers from the exponential growth of network complexity and interpolation error in recovering continuous sources. Burel [12] proposed a nonlinear blind source separation algorithm using two-layer perceptrons by the gradient descent method to minimize the mutual information (measure of dependence). Subsequently, Yang *et al.* [19] developed an information backpropagation (BP) algorithm for Burel's model by natural gradient method. In their model cross nonlinearities are included. Taleb *et al.* [20] recently proposed an entropy-based direct algorithm for blind source separation in post nonlinear mixtures.

In addition, the extension of related linear ICA theories to the context of nonlinear mixtures has resulted in the development of nonlinear ICA [14]. The so-called nonlinear ICA is to employ a nonlinear function to transform the nonlinear mixture such that the outputs become statistically independent after the transformation. However, this transform is not unique without some specific constraints on the function of nonlinear mixing. If x and y are two independent random variables, then $f(x)$ and $g(y)$ are also statistically independent regardless of the nonlinear functions f and g. Although there exist many difficulties for this problem, several nonlinear ICA algorithms have been proposed and developed [14], [16]. Most recently, Hyvarinen *et al.* [21] discussed the existence and uniqueness of nonlinear ICA in detail and pointed out that the solution of nonlinear ICA always exists and can become unique up to a rotation provided that the mixing function is constrained to a conformal mapping for

Manuscript received July 4, 1999; revised July 6, 2000.

Y. Tan is with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China.

J. Wang is with the Department of Automation and Computer-Aided Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong.

J. M. Zurada is with the Department of Electrical Engineering, University of Louisville, Louisville, KY 40292 USA.

This work was supported by the Hong Kong Research Grants Council under Grant CUHK4150/97E and in part by the Systems Research Institute, Polish Academy of Science, Warsaw, Poland.

Publisher Item Identifier S 1045-9227(01)00529-X.

a two-dimensional problem together with some other assumptions such as bounded support of the probability density function (pdf).

The purpose of this paper is to investigate a radial-basis function (RBF) neural-network model for blind-style demixing of nonlinear mixtures in the presence of cross nonlinearities. Since the instinct unsupervised learning of the RBF network and blind signal processing are in essence unsupervised learning procedures, therefore the study of the RBF-based separation system seems natural and reasonable. Further, a contrast function proposed in this paper is different from those in the existing methods and algorithms. It consists of the mutual information of the output of the separating system and relevant moment matching, and is expected to guarantee a unique solution for the model. By making use of the fast convergence and universal approximation properties of RBF networks, the new proposed demixing model is expected to outperform the other existing algorithms. In addition, the unsupervised learning algorithms of RBF networks appear naturally suited, and the structure of RBF networks is modular and is easily implemented in hardware. The proposed RBF-based system can overcome several limitations of the existing methods such as highly nonlinear weight update and slow convergence rate. To our knowledge, this is the first time to utilize RBF networks to deal with nonlinear blind source separation problem in literature.

The remainder of this paper is organized as follows: In Section II, a nonlinear mixture model is first presented. In Section III, a concise description of the RBF-based nonlinear separation system is outlined. In Section IV, we define a contrast function on the basis of the principles of independence and moment matching. In Section V, the demixing learning algorithms are derived by minimization of the contrast function with respect to the parameters of the RBF network. The specific steps of the demixing algorithms and an effective performance index are elucidated in Section VI. In Section VII, three simulation results are given to illustrate the correctness and effectiveness of the proposed RBFN-based model for the nonlinear separation. Finally, Section VIII contains concluding remarks.

II. NONLINEAR MIXTURE MODEL

A generic nonlinear mixture model for blind source separation can be described as

$$\mathbf{x}(t) = \mathbf{f}[\mathbf{s}(t)] \tag{1}$$

where

- $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$ vector of observed random variables;
- superscript T transposition operator;
- $\mathbf{s}(t) = [s_1(t), s_2(t), \dots, s_n(t)]^T$ vector of the latent variables called the independent source vector;
- $\mathbf{f}$ unknown multiple-input and multiple-output (MIMO) mapping from R^n to R^n called nonlinear mixing transform (NMT).

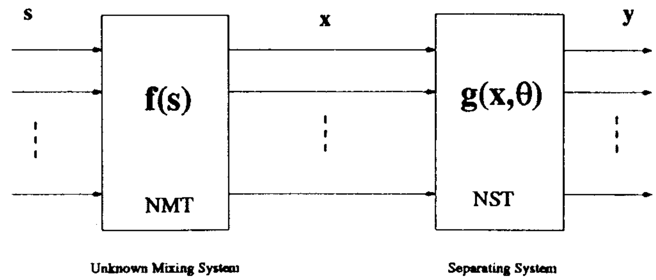

Fig. 1. Nonlinear mixing and separating systems for blind signal separation.

If the mixing function $\mathbf{f}$ is linear, (1) reduces to the linear mixing [28]. In order for the mapping to be invertible we assume that the nonlinear mapping $\mathbf{f}$ is monotone. Even though the dimensions of $\mathbf{x}$ and $\mathbf{s}$ generally need not be equal, we yet make this assumption here for simplicity and convenience. The left part of Fig. 1 shows the general model of blind source separation system described in (1), which contains both channel and cross-channel nonlinearities.

The separating system $\mathbf{g}(\cdot, \theta)$ in the right part of Fig. 1, called nonlinear separation transform (NST), is used to recover the original signals $\mathbf{x}(t)$ from the nonlinear mixture without the knowledge of the source signals $\mathbf{s}(t)$ and the mixing nonlinear function $\mathbf{f}(\cdot)$. Obviously, this problem is not tractable, for general nonlinear mixing system, unless conditions are imposed on the nonlinear function $\mathbf{f}(\cdot)$. At first, the existence of the solution for the NST can be guaranteed. According to related nonlinear ICA theories, the nonlinear ICA problem always has at least one solution. That is, given a random vector $\mathbf{x}$, there is always a function $\mathbf{g}$ so that the components of $\mathbf{y} = [y_1, \dots, y_n]^T$ given by $\mathbf{y} = \mathbf{g}(\mathbf{x})$ are independent as described by the following theorem.

Theorem [14], [21]: Suppose m variables $y_1, \dots, y_m$ are mutually independent and follow a joint uniform distribution in the unit cube $[0,1]^m$. Let x be any random variable and set

$$y_{m+1} = \mathbf{g}(y_1, \dots, y_m, x; p_{y,x}) \tag{2}$$

where function $\mathbf{g}$ is defined as

$$\mathbf{g}(a_1, \dots, a_m, b; p_{y,x}) = P(x \le b \,|\, y_1 = a_1, \dots, y_m = a_m) \tag{3}$$

where $p_{y,x}$ is marginal probability density of $(y_1, \dots, y_m, x)$ and $P(\cdot \,|\, \cdot)$ denotes the conditional probability. Then y_{m+1} is independent from $(y_1, \dots, y_m)$.

By the theorem we can obtain mutually independent variables $\mathbf{y}$ from any variable $\mathbf{x}$, where $\mathbf{y}$ is a uniformly distributed random vector transformed by the mapping $\mathbf{g}$ from $\mathbf{x}$. Unfortunately, this kind of mapping is not unique at all. It is shown in [21] that a unique solution subjected to a rotation can be obtained under the assumptions that the problem is a two-dimensional one, mixing function is a conformal mapping, and the densities of the independent components are known and have bounded support. In order to obtain a unique solution of the model in Fig. 1, we assume that $\mathbf{f}(\cdot)$ is invertible and its inverse $\mathbf{f}^{-1}(\cdot)$ exists and can be uniquely approximated by a parametric RBF network shown in Fig. 2 of the following section.

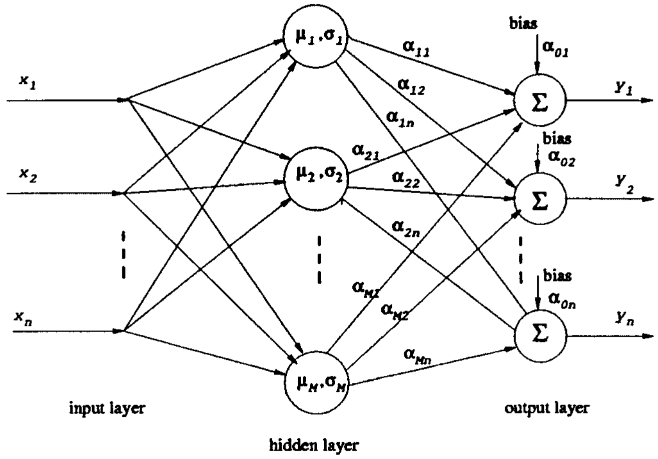

Fig. 2. The RBF networks with n output units.

In addition, we add some constraints on the output; i.e., the moment matching between the outputs of the separating system and sources. According to Fig. 1 or (1), the output of the nonlinear separating system can be written as

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \theta) \tag{4}$$

where θ is a parameter vector to be determined. Substituting (1) into (4), we can obtain

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{f}(\mathbf{s}(t)), \theta) = \mathbf{s}(t) \tag{5}$$

where $g(\cdot, \theta) = f^{-1}(\cdot)$ denotes a parametric fitting function class.

Generally speaking, $g(\cdot, \theta)$ can be altered by varying θ. If we find such $\theta = \hat{\theta}$ that $g(\cdot, \hat{\theta})$ is a good approximation of the inverse of the nonlinear mixing function $f^{-1}(\cdot)$, then a good separation of nonlinear mixture is achieved. However, since given $f(\cdot)$, the only assumption of statistical independence is not sufficient to separate source signals from their nonlinear mixture [21]. Therefore, some further constraints on the mixing function $f(\cdot)$ and statistical properties of the source signals should be imposed such that the source signals can be easily and completely recovered, even though the meaning of "blind" is somewhat different from the linear case due to the required statistical properties of the sources.

III. Nonlinear Separation Based on an RBF Network

A. RBF Neural Networks

Fig. 2 shows an n-input and n-output RBF network model. It consists of three layers; i.e., input layer, hidden layer, and output layer. The neurons in hidden layer are of local response to its input and called RBF neurons while the neurons of the output layer only sum their inputs and are called linear neurons. The RBF network of Fig. 2 is often used to approximate an unknown continuous function $\phi : R^n \rightarrow R^n$ which can be described by the affine mapping

$$\mathbf{u}(\mathbf{x}) = \mathbf{B}K(\mathbf{x}, \mathbf{p}) \tag{6}$$

where $\mathbf{B} = [\alpha_{ij}]$ is a $n \times M$ weight matrix of the output layer, $K(\mathbf{x}, \mathbf{p})$ is the kernel function vector of the RBF network, which consists of the locally receptive functions.

Usually, $K(\mathbf{x}, \mathbf{p})$ takes one of several forms such as $K(r) = r$ (linear), $K(r) = r^3$ (cubic), $K(r) = r^2 \log(r)$ (thin plate spline), $K(r) = \exp(-r^2/2)$ (Gaussian), $K(r) = \sqrt{r^2+1}$ (multiquadric), or $K(r) = 1/\sqrt{r^2+1}$ (inverse multiquadric) where in all cases r is scaled radius $||\mathbf{x} - \mu_i||/\sigma_i$. For later convenience of notation, we let $\mathbf{p} = (\mu_1, \ldots, \mu_M, \sigma_1, \ldots, \sigma_M)$ be the parameter vector of the kernel, and further denote $\theta = (\mathbf{B}, \mathbf{p})$ as the parameter set of the RBF network. For the proposed separating model we choose the conventional Gaussian kernel as the activation function of RBF neurons as it displays several desirable properties from the viewpoint of interpolation and regularization theory [22]. Then the kernel function vector $K(\mathbf{x}, \mathbf{p})$ can be further expressed as

$$\begin{aligned} K(\mathbf{x}, \mathbf{p}) = \big[1, \exp\big(-(\mathbf{x}-\mu_1)^T(\mathbf{x}-\mu_1)/\sigma_1^2\big), \ldots,\\ \exp\big(-(\mathbf{x}-\mu_M)^T(\mathbf{x}-\mu_M)/\sigma_M^2\big)\big]^T. \end{aligned} \tag{7}$$

Here we let the first component of $K(\mathbf{x}, \mathbf{p})$ be one for taking the bias into account.

B. Nonlinear Separation System Based on an RBF Network

It is well known that neural-network training can result in producing weights in undesirable local minima of the criterion function. This problem is particularly serious in recurrent neural networks as well as for multilayer perceptrons with highly nonlinear activation functions because of their highly nonlinear structure, and it gets worse as the network size increases. This difficulty has motivated many researchers to search for a structure where the output dependence on network weights is less nonlinear. The RBF network has a linear dependence on the output layer weights, and the nonlinearity is introduced only by the cost function for training, which helps to address the problem of local minima. Additionally, this network is inherently well suited for blind signal processing because it naturally uses unsupervised learning to cluster the input data.

Since the local response power of RBF networks offers great classification and approximation capabilities, the Gaussian RBF network is used as a good function approximator in many modeling applications. If we let $\mathbf{S}$ be a compact subset in R^n and $\mathbf{p}(\mathbf{x})$ be a continuous target vector on $\mathbf{S}$, then for any $\epsilon > 0$ there exist M centroids $\mu_i = [\mu_{i1}, \ldots, \mu_{in}]^T$ and an $n \times M$ constant matrix $\mathbf{B}$ such that $\mathbf{r}(\mathbf{x}, \theta) = \mathbf{B} \cdot K(\mathbf{x}, \mathbf{p})$ satisfies $|\mathbf{r}(\mathbf{x}, \theta) - \mathbf{p}(\mathbf{x})| < \epsilon$ for all $\mathbf{x} \in \mathbf{S}$. This approximation ability of RBF networks directly stems from the classic Stone–Weierstrass theorem and is closely related to Parzen's approximation theory. Also, it can be easily derived from the Poggo's regularization formulation [23]. Therefore, the inverse of the nonlinear mixing model can be modeled by using an RBF network. Such architecture is preferred over multilayer perceptrons (MLPs) even though an unsupervised learning algorithm for MLPs was developed in terms of a Bayesian treatment [24], as an RBF network has better capability for functional representation. Since its response is linearly related to its weights, learning in an RBF network is expected to train faster while its local response power offers a good approximation capability. As a result, we can reach

$$\mathbf{y} = \hat{\mathbf{B}}K[\mathbf{f}(\mathbf{s}), \hat{\mathbf{p}}] \propto \mathbf{s} \tag{8}$$

where $\mathbf{g}(\cdot, \hat{\theta}) = \hat{\mathbf{B}}K[\cdot, \hat{\mathbf{p}}]$, $\hat{\mathbf{B}}$ and $\hat{\mathbf{p}}$ are the final estimates of parameters $\mathbf{B}$ and $\mathbf{p}$ of the RBF network such that the inverse of $\mathbf{f}$ is well approximated by the RBF network.

IV. Contrast Function

In order to extract the independent sources from their nonlinear mixtures, we first expect the outputs of the separation system to be mutually statistically independent. For this purpose, we must utilize a measure of independence between random variables. Here, the mutual information is chosen as the measure of independence since it is the best index of this kind so far. Unfortunately, regarding the separation of a nonlinear mixture, independence only is not sufficient to perform blind recovery of the original signals. Therefore, we have to impose some additional constraints on the sources. Here, we are required to possess some knowledge of the moments of the sources except for the independence. So we expect the outputs of the separation system to have some identical moments to those of the sources.

Thus, in order to deal with the nonlinear separation problem effectively, we define a contrast function, which is the objective function for source separation, as

$$C(\theta) = I(\mathbf{y}) + \sum_{i_1 \ldots i_n} c_{i_1 \ldots i_n} \left[M_{i_1 \ldots i_n}(\mathbf{y}, \theta) - M_{i_1 \ldots i_n}(\mathbf{s})\right]^2 \tag{9}$$

where

$I(\mathbf{y})$	mutual information of the outputs of the separation system;
$M_{i_1 \ldots i_n}(\mathbf{y}, \theta)$ and $M_{i_1 \cdots i_n}(\mathbf{s})$	$i_1 \ldots i_n$th moments of $\mathbf{y}$ and $\mathbf{s}$, respectively;
$c_{i_1 \ldots i_n}$	constants which are used to balance the mutual information and the matching of moments.

According to information theory and related the Kullback–Leibler divergence [4], [28], [30], mutual information $I(\mathbf{y})$ in Eq. (9) is expressed as

$$I(\mathbf{y}) = \sum_{i=1}^{n} H(y_i) - H(\mathbf{y}) \tag{10}$$

where

$H(\mathbf{y}) = -E[\log(p_{\mathbf{y}}(\mathbf{y}))]$	joint entropy of random vector $\mathbf{y}$;
$H(y_i) = -E[\log(p_{y_i}(y_i))]$	entropy of random variable y_i, the ith component of $\mathbf{y}$;
$E(\cdot)$	expectation operator.

The $i_1 \ldots i_n$th moment of $\mathbf{y}$ is defined as

$$M_{i_1 \ldots i_n}(\mathbf{y}) = E\left(y_1^{i_1} \ldots y_n^{i_n}\right) - E(y_1^{i_1}) \ldots E\left(y_n^{i_n}\right). \tag{11}$$

It can be seen from (9)–(11) that the contrast function defined in (9) is always nonnegative, and reaches zero if and only if both mutual information is null and a perfect matching of moments between the outputs of the separation system and original sources is achieved. Therefore, independent outputs with the same moments as that of original sources can be found by minimizing the contrast function by adjusting the parameters of the RBF separating system; i.e.,

$$\hat{\theta} = \arg\min_{\theta} \left\{ I(\mathbf{y}) + \sum_{i_1 \ldots i_n} c_{i_1 \ldots i_n} \left[M_{i_1 \ldots i_n}(\mathbf{y}, \theta) - M_{i_1 \ldots i_n}(\mathbf{s})\right]^2 \right\}. \tag{12}$$

In what follows, we will formulate the learning algorithms of the separation system in terms of minimization of the contrast function defined in (9).

V. Unsupervised Learning of the Separating RBF Network

There are two basic methods to train an RBF network in the context of neural networks. One is to jointly optimize all parameters of the network similarly to the training of the MLP. This method usually results in good quality of approximation but also has some drawbacks such as a large amount of computation and a large number of adjustable parameters. Another method is to divide the learning of an RBF network into two steps. The first step is to select all the centers μ in terms of an unsupervised clustering algorithm such as the K-means algorithm proposed by Linde *et al.*(denoted as the LBG algorithm) [25], [26], and choose the radii σ by the k-nearest neighbor rule. The second step is to update the weights $\mathbf{B}$ of the output layer while keeping the μ and σ fixed. The two-step algorithm has fast convergence rate and small computational burden. What follows presents the relevant algorithms of the two methods for minimizing the contrast function of (9).

A. Learning Algorithm I

In order to derive the unsupervised learning algorithm of all the parameters of the separating RBF network, we employ the gradient descent method. First of all, we compute the gradient of the contrast function of (9) with respect to the parameter θ and obtain

$$\frac{\partial C(\theta)}{\partial \theta} = \frac{\partial I(\mathbf{y})}{\partial \theta} + \sum_{i_1 \ldots i_n} 2c_{i_1 \ldots i_n} \cdot \left[M_{i_1 \ldots i_n}(\mathbf{y}, \theta) - M_{i_1 \ldots i_n}(\mathbf{s})\right] \frac{\partial M_{i_1 \ldots i_n}(\mathbf{y}, \theta)}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \theta} \tag{13}$$

where mutual information can be further rewritten as

$$I(\mathbf{y}) = \sum_{i=1}^{n} H(y_i) - E\left\{\log\left|\frac{\partial \mathbf{g}(\mathbf{x}, \theta)}{\partial \mathbf{x}}\right|\right\} - H(\mathbf{x}) \tag{14}$$

where $|\partial \mathbf{g}(\mathbf{x}, \theta)/\partial \mathbf{x}|$ is the determinant of the Jacobian matrix of function $\mathbf{g}(\mathbf{x}, \theta)$ with respect to vector $\mathbf{x}$.

In (14), the computation of $H(y_i)$ needs to use the pdf of y_i which is unknown. There are two methods to tackle the problem.

One method is to approximate it in terms of a series expansion as done by several authors [29]. Another method is to directly estimate the pdf of y_i from the estimated value of y_i during learning by a unsupervised learning algorithm [20]. Because the Gram–Charlier expansion method only needs some moments of y_i and, has less computational amount and complexity and, can be expressed by an explicit formula, but the direct pdf estimation needs more information to estimate the pdf and, is in a form of recursive style. Therefore, here we adopt the former method. By applying the Gram–Charlier expansion suggested by Amari *et al.* [29] to express each marginal pdf of $\mathbf{y}$, the marginal entropy can be approximated as

$$H(y_i) \approx \frac{1}{2}\log(2\pi e) - \frac{\left(k_3^i\right)^2}{2\times 3!} - \frac{\left(k_4^i\right)^2}{2\times 4!} + \frac{3}{8}\left(k_3^i\right)^2 k_4^i + \frac{1}{16}\left(k_4^i\right)^3 \tag{15}$$

where $k_3^i = m_3^i$, $k_4^i = m_4^i - 3$, and $m_k^i = E[(y_i)^k]$, $j = 1, \ldots, n$.

Since $H(\mathbf{x})$ does not contain any parameters of the separating RBF network, it becomes null when taking gradient with respect to the parameters.

We thus have the following gradient expression:

$$\frac{\partial I(\mathbf{y})}{\partial \theta} = \sum_{i=1}^{n}\sum_{j=3,4}\frac{\partial H(y_i)}{\partial k_j^i}\frac{\partial k_j^i}{\partial y_i}\frac{\partial y_i}{\partial \theta} - \left|\frac{\partial \mathbf{g}(\mathbf{x},\theta)}{\partial \mathbf{x}}\right|^{-1}\frac{\partial}{\partial \theta}\left|\frac{\partial \mathbf{g}(\mathbf{x},\theta)}{\partial \mathbf{x}}\right|. \tag{16}$$

According to the definition in (11) we can also calculate $j = 1, \ldots, n$

$$\frac{\partial M_{i_1\ldots i_n}(\mathbf{y},\theta)}{\partial y_j} = \frac{\partial E\left(y_1^{i_1}\ldots y_n^{i_n}\right)}{\partial y_j} - \frac{E\left(y_j^{i_j}\right)}{\partial y_j}\Pi_{k\neq j}E\left(y_k^{i_k}\right). \tag{17}$$

Regarding different parameters $\mathbf{B}$, μ, and σ of the parameter set θ of the RBF network, we have the following gradient equations of the separated signal $\mathbf{y}$:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{B}} = K(\mathbf{x},\mathbf{t}) \tag{18}$$

$$\frac{\partial \mathbf{y}}{\partial \mu} = \mathbf{B}\cdot \mathrm{diag}[\mathbf{v_1}\circ K(\mathbf{x},\mathbf{t})] \tag{19}$$

$$\frac{\partial \mathbf{y}}{\partial \sigma} = \mathbf{B}\cdot \mathrm{diag}[\mathbf{v_2}\circ K(\mathbf{x},\mathbf{t})]. \tag{20}$$

where $\mathbf{v_1} = [2(\mathbf{x}-\mu_1)/\sigma_1^2, \ldots, 2(\mathbf{x}-\mu_M)/\sigma_M^2]^T$, $\mathbf{v_2} = [2\|\mathbf{x}-\mu_1\|^2/\sigma_1^3, \ldots, 2\|\mathbf{x}-\mu_M\|^2/\sigma_M^3]^T$, function $\mathrm{diag}[\cdot]$ denotes diagonal matrix, symbol $\circ$ denotes Hadamard product which is the multiplication of corresponding pairs of elements between two vectors.

Finally, from (13), (16)–(20), we can easily calculate the gradients of the constrast function with respect to each parameter of parameter set θ, i.e., $\partial C(\theta)/\partial \mathbf{B}$, $\partial C(\theta)/\partial \mu$ and $\partial C(\theta)/\partial \sigma$, then give the following learning updating formula:

$$\delta \mathbf{B} = -\eta\frac{\partial C(\theta)}{\partial \mathbf{B}}, \quad \delta\mu = -\eta\frac{\partial C(\theta)}{\partial \mu}, \quad \delta\sigma = -\eta\frac{\partial C(\theta)}{\partial \sigma} \tag{21}$$

where $\eta > 0$, $\mu = [\mu_1, \ldots, \mu_M]^T$ and $\sigma = [\sigma_1, \ldots, \sigma_M]^T$; η denotes the positive learning rate; $\delta\mathbf{B}$, $\delta\mu$ and $\delta\sigma$ indicate the adjustments of $\mathbf{B}$, μ and σ, respectively.

B. Learning Algorithm II

Since the algorithm I of the above section needs to jointly optimize all the parameters of the RBF network, and the output of the separating system is a nonlinear function of most parameters of the network such as centers and radii of the RBF neurons, it may suffer from the high complexity and heavy computational burden. We will use a two-step learning algorithm to speed up the learning process of the separation system based on an RBF network.

Blind signal processing is essentially an unsupervised learning process. The selection of the centers and radii of RBF neurons can be done naturally in an unsupervised manner, which makes this structure intrinsically well suited for blind signal processing. As a result, we adopt below a self-organized learning algorithm for selection of the centers and radii of the RBF in the hidden layer, and a stochastic gradient descent of the contrast function for updating the weights in the output layer. For the self-organized selection of the centers of the hidden units, we may use the standard k-means clustering algorithm [26] whose complex-valued version can be found in [27]. This algorithm classifies an input vector $\mathbf{x}$ by assigning it the label most frequently represented among the k-nearest neighbor samples. Specifically, it places the centers of RBF neurons in only those regions of the input space where significant data are present. Let $\mathbf{p}_k(n)$, $k = 1, \ldots, K$ denotes the the centers of RBF neurons at iteration n. Then the best-matching (winning) center $\hat{k}(\mathbf{x})$ at iteration n using the minimum-distance Euclidean criterion can be found as follows:

$$\hat{k}(\mathbf{x}) = \arg\min_k \|\mathbf{x} - \mathbf{p}_k(n)\|, \quad k = 1, \ldots, K. \tag{22}$$

The update rule for the locations of the centers are given by

$$\mathbf{p}_k(n+1) = \begin{cases} \mathbf{p}_k(n) + \eta_s[\mathbf{x} - \mathbf{p}_k(n)], & k = \hat{k}(\mathbf{x}) \\ \mathbf{p}_k(n), & \text{otherwise} \end{cases} \tag{23}$$

where η_s denotes the learning rate in interval $(0, 1)$.

Once the centers and radii are established, we can make use of the minimization of the contrast function to update the weights of the RBF network. The update rule for the weights is the same as that of learning algorithm I, i.e., (18) and (21).

VI. Performance Index and Algorithm Description

A. Performance Index

It is well known that the performance of an algorithm should be measured by some indexes such as efficacy and computation amount. For the proposed model, we would like to achieve the independence of the outputs of the separation system and moment matching between the estimated and original sources. As a result, this performance index should monitor the two aspects. Meanwhile, we also wish to choose an index which not only is to measure the desirable properties but is also easily computable in practice.

From (14), by omitting the unknown $H(\mathbf{x})$, an index to measure the independence of the outputs of the separation system is defined as

$$J_i = \sum_{i=1}^{n} H(y_i) - E\left\{\log\left|\frac{\partial \mathbf{g}(\mathbf{x},\theta)}{\partial \mathbf{x}}\right|\right\}. \tag{24}$$

Even though the index J_i may be negative, the lower the value of J_i is, the more independent the outputs of the separating system is. The smallest negative value of J_i is just equal to the reciprocal of $H(\mathbf{x})$. In a similar manner, according to (14), a performance index measuring moment match between the outputs of the separation system and original sources can also be directly defined as

$$J_m = \sum_{i_1 \ldots i_n} \left[M_{i_1\ldots i_n}(\mathbf{y},\theta) - M_{i_1\ldots i_n}(\mathbf{s})\right]^2. \tag{25}$$

From (25), we know that the value of J_m is nonnegative and will equal zero if and only if the outputs of separating system have the same moments as original sources. In practical implementation, we need not take all of the moments into account, but only a few lower order moments are enough. As a result, finite moment matching up to the kth order

$$J_m^k = \sum_{i_1 \ldots i_n \le k} \left[M_{i_1\ldots i_n}(\mathbf{y},\theta) - M_{i_1\ldots i_n}(\mathbf{s})\right]^2 \tag{26}$$

is done in our experiments of the following section. The maximum value of k is chosen such that the inverse of the mixing nonlinear transform can be uniquely approximated by an RBF network through the minimization of the contrast function. In actual implementation, usually only up to fourth-order moment is enough for this purpose by experiments. We expect both J_i and J_m^k are at their minimia simultaneously, so the two indices can be combined into one overall index as follows:

$$J = J_i + \alpha J_m^k \tag{27}$$

where α is a proportionality constant weighting the two quantities.

It is well known that a recursive iterative algorithm needs a stopping criterion to terminate the iterative process. For this purpose, we define a relative change amount of the overall index $J(\theta(t))$ with respect to parameters θ from instant t to instant $t+1$ as

$$e_r = \frac{|J(\theta(t+1)) - J(\theta(t))|}{|J(\theta(t+1))|}. \tag{28}$$

If e_r is less than a predetermined small positive constant ϵ, the algorithm terminates. What follows summarizes the steps of two learning algorithms of the previous section.

B. Algorithm Description

According to the performance index and the stopping criterion for the iteration process of last subsection as well as the unsupervised learning formula in Section V, we can summarize the specific computational procedures of the learning algorithms as follows:

Note that in both methods we initialize the RBF network randomly even though a good initialization greatly benefits a gradient descent learning algorithm.

The specific separation steps for learning algorithm I (LA-I) can be given.

1) Given initialization parameters $\mathbf{B}$, μ, σ, for the RBF network, choose a small learning rate η and index balance number α as well as the order number k of the moment to be matched.
2) Adjust parameters, $\mathbf{B}$, μ, σ, of the RBF network by using (18)–(21).
3) Compute the performance indexes J_i, J_m^k, J according to (24)–(28) by making use of current model parameters.
4) Verify the termination criterion $(e_r < \epsilon)$ and exit, otherwise go to Step 2).

Similarly, the unsupervised learning steps for learning algorithm II (LA-II) can be given.

1) Given the number of RBF neurons M, learning rates η_s and choose a small learning rate η and index balance number α as well as the order number k to be matched.
2) Utilize k-means clustering algorithm (22) and (23) to determine the centers μ and radii σ of RBF neurons.
3) Adjust weight parameter $\mathbf{B}$ by using (18) and (21).
4) Compute the performance indexes J_i, J_m^k, J according to (24) and (28) by making use of current model parameters.
5) Verify the termination criterion $(e_r < \epsilon)$ and exit, otherwise go to Step 2).

VII. Simulation Results

The model and algorithms proposed in this paper have been implemented on a SUN Sparc workstation. A number of simulations have been performed to fully evaluate the validity and performance of the algorithms for stationary source signals with complex nonlinear mixture. This section reports the performance by presenting three illustrative examples.

Example 1: Consider a two-channel nonlinear mixture with a cubic nonlinearity

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \mathbf{A_2} \begin{bmatrix} (\cdot)^3 \\ (\cdot)^3 \end{bmatrix} \mathbf{A_1} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \tag{29}$$

where mixing matrices $\mathbf{A_1}$ and $\mathbf{A_2}$ are nonsingular and given as

$$\mathbf{A_1} = \begin{pmatrix} 0.25 & 0.86 \\ -0.86 & 0.25 \end{pmatrix}, \quad \mathbf{A_2} = \begin{pmatrix} 0.5 & 0.9 \\ -0.9 & 0.5 \end{pmatrix}.$$

The schematic diagram of the mixing model expressed in (29) is shown in Fig. 3.

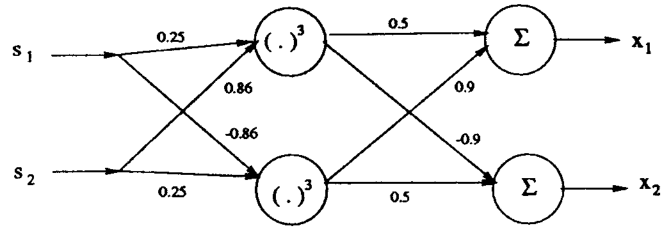

Fig. 3. Two-channel cubic nonlinear mixing model.

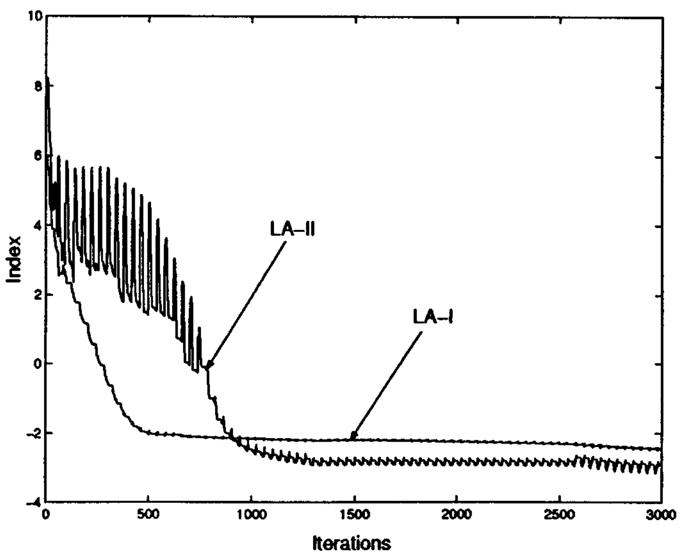

Fig. 4. Learning curves of two algorithms LA-I and LA-II.

Obviously, the inverse of the mixing model in Fig. 3 exists and can be expressed as

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \mathbf{A_1^{-1}} \begin{bmatrix} \mathrm{sgn}(\cdot)(\cdot)^{1/3} \\ \mathrm{sgn}(\cdot)(\cdot)^{1/3} \end{bmatrix} \mathbf{A_2^{-1}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

where sgn function is to take the sign of the argument.

It is different from the channel nonlinear models there exists cross-channel nonlinearity in the model due to the postmatrix mixing process in Fig. 3. The source vector $\mathbf{s}(t)$ consists of a sinusoidal signal and an amplitude-modulated signal; i.e., $\mathbf{s}(t) = [0.5 * [1 + \sin(6\pi t)] \cos(100\pi t), \sin(20\pi t)]^T$.

An RBF network shown in Fig. 2 is used to separate this nonlinear mixture. In this experiment we choose six hidden neurons with Gaussian kernel function. The output neurons are linear summation neurons. The moment matching is taken up to third order. An example of the learning curves for LA-I and LA-II is shown in Fig. 4. The learning curve of LA-I is smooth and it converges faster than that of LA-II which exhibits some oscillations during the convergence. But LA-II approaches a lower index than LA-I. It can be seen from the figure that the convergence rate of the algorithms is very fast. The value of the performance index after convergence of the learning algorithm is very small so that the separated signals obtained by the proposed model are seen to be mutually independent.

Fig. 5 shows the two source signals $\mathbf{s}(t)$ and the input signals $\mathbf{x}(t)$ of the separating system of Fig. 1, i.e., the mixture of the sources. Figs. 6 and 7 show the signals separated by LA-I and LA-II, respectively. The overall learning process for LA-II only

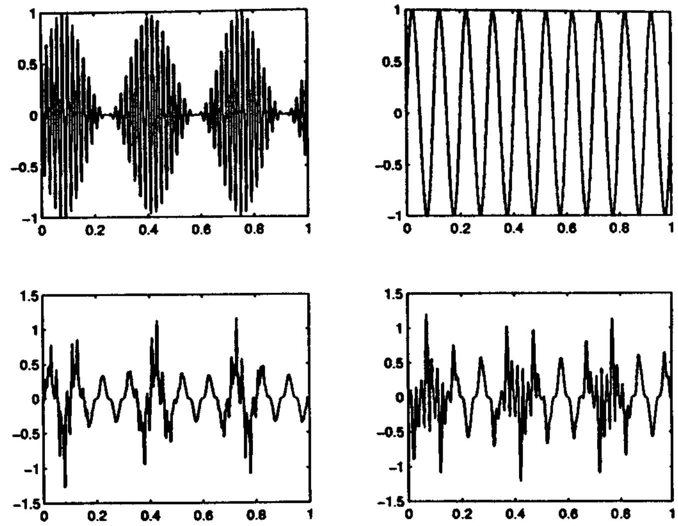

Fig. 5. Two source signals (above) and their nonlinear mixtures (below).

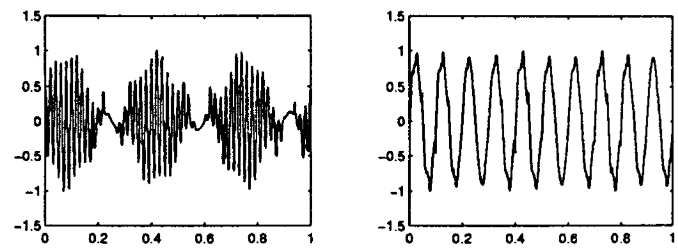

Fig. 6. Separated signals of the RNF neural-network approach by using LA-I.

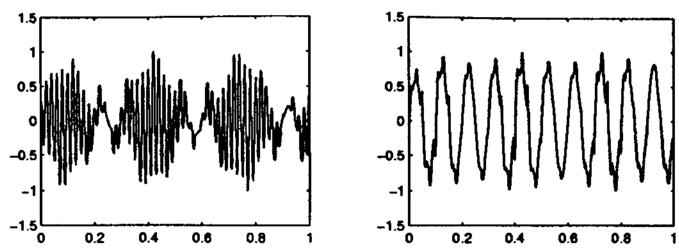

Fig. 7. Separated signals of the RNF neural-network approach by using LA-II.

take 96.2 s at a Sun Ultra 1 workstation, but LA-I takes much more time than LA-II. This is because LA-I needs to compute the gradients of all parameters of network while LA-II requires only the gradient of the output weights. It is also noted from the figure that J is smaller for LA-II than that for LA-I. This is because the local minima of the optimization process of the LA-I degrades the steady-state convergence performance of LA-I. For convenient comparison, we also have plotted the separating results of the linear demixing algorithm proposed in [31] for the same problem of Example 1 into Fig. 8. It can be seen that the linear demixing algorithm fails to separate the nonlinear mixture but the proposed model and algorithms can give a clear separation of this nonlinear mixture. We also used the other specific nonlinear algorithms such as the SOM algorithm [17], but no useful result has been obtained.

Example 2: Consider a more general nonlinear mixing system with four source signals and suppose the nonlinear

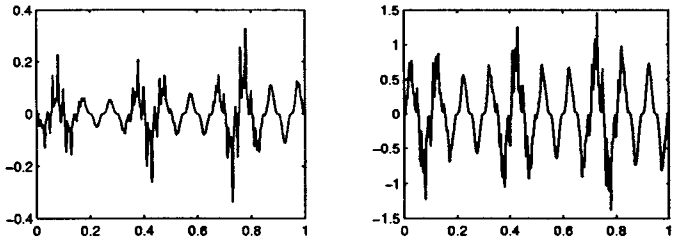

Fig. 8. Separated signals of adaptive algorithm for linear mixture case.

mixing function $\mathbf{f}(\mathbf{s})$ is implemented by using a three-layer neural network with four input neurons, four hidden neurons, and four output neurons. The activation function of hidden neurons is assumed to be sigmoidal function. The output neurons are all linear neurons. Thus the nonlinear mixture can be produced by

$$\mathbf{x}(t) = \mathbf{V} \cdot \mathrm{sgm}[\mathbf{W}\mathbf{s}(t)] \quad (30)$$

where function $\mathrm{sgm}[\cdot]$ denotes bipolar sigmoid activation function which can be $\tanh(\mathbf{s})$ or $(1 - \exp(-x))/(1 + \exp(-x))$ or other monotonically increasing continuous functions. The mixing matrices $\mathbf{W}$ and $\mathbf{V}$ are nonsingular and their elements are randomly chosen in $[0, 1]$ in this example as

$$\mathbf{W} = \begin{pmatrix} 0.2844 & 0.5828 & 0.4329 & 0.5298 \\ 0.4692 & 0.4235 & 0.2259 & 0.6405 \\ 0.0648 & 0.5155 & 0.5798 & 0.2091 \\ 0.9883 & 0.3340 & 0.7604 & 0.3798 \end{pmatrix},$$

$$\mathbf{V} = \begin{pmatrix} 0.7349 & 0.1556 & 0.4902 & 0.4507 \\ 0.6873 & 0.1911 & 0.8159 & 0.4122 \\ 0.3461 & 0.4225 & 0.4608 & 0.9016 \\ 0.1660 & 0.8560 & 0.4574 & 0.0056 \end{pmatrix}.$$

We further assume that the source vector $\mathbf{s}(t)$ consists of a binary signal, a sinusoid, a saw-toothed wave (ramp) and a high-frequency carrier, i.e., $\mathbf{s}(t) = [\mathrm{sgn}[\cos(160\pi t)], \sin(20\pi t), \mathrm{ramp}(t, \tau), \sin(800\pi t)]^T$, where function $\mathrm{ramp}(t, \tau)$ denotes a periodic saw-tooth wave with period $\tau = 0.1667$.

We adopt an RBF network with three layers as shown in Fig. 2. The input and output layers have four neurons, respectively, and the hidden layer has eight neurons. Up to fourth-order moment matching is used; i.e., $k = 4$. The index balance constant $\alpha = 1$.

Figs. 9 and 10 show, respectively, the four source signals and their nonlinear mixture as in (30). Figs. 11 and 12 show the separated signals of this proposed RBF-based separation model by LA-I and LA-II, respectively, for this problem. Fig. 13 depicts the performance index curves during the learning phase of the proposed algorithms LA-I and LA-II. From the learning curves we can clearly observe a fast convergence of the learning process and a successful separation of the multiple-channel nonlinear mixtures. From these figures, we have also observed the similar phenomena about the convergence rate and steady-state accuracy as that in Example 1 for LA-I and LA-II. Similar to Example 1, we have also studied the separation results of other common linear and nonlinear demixing algorithms for this problem. Fig. 14 shows the separation results

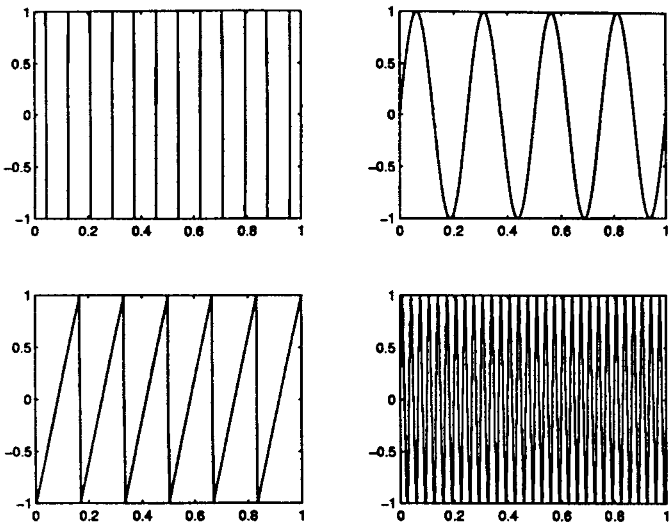

Fig. 9. Four source signals: a binary, a sinusoid, a ramp and a carrier.

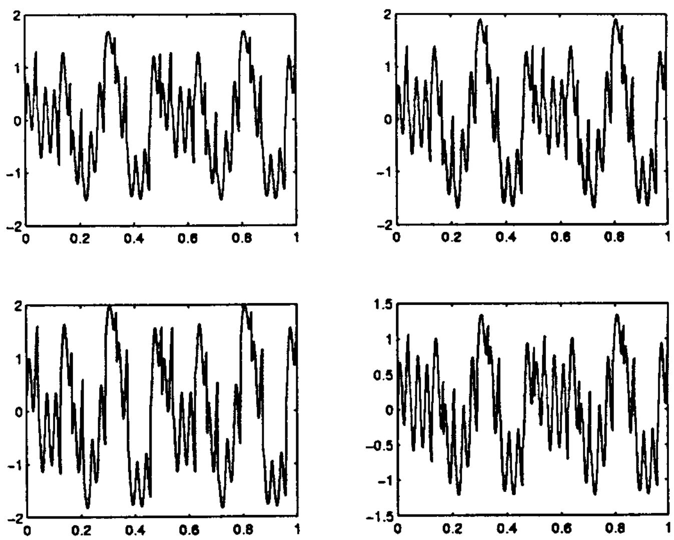

Fig. 10. Mixtures of four source signals by a nonlinear mixing model in (35).

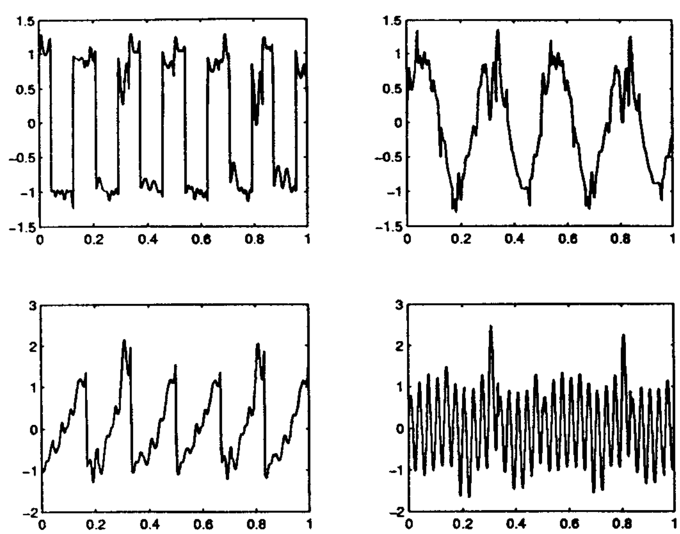

Fig. 11. Separated signals of the proposed model by using LA-I.

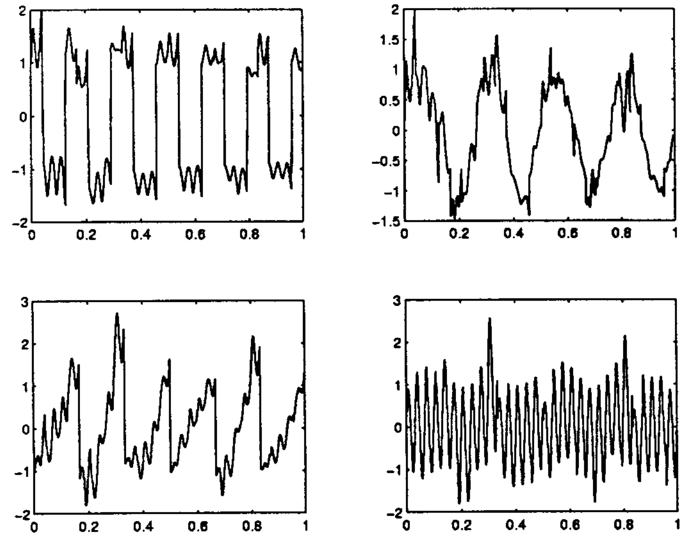

Fig. 12. Separated signals of the proposed model by using LA-II.

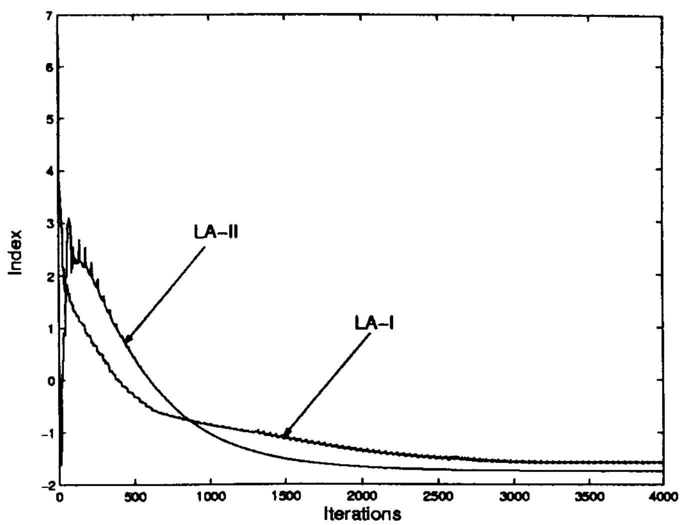

Fig. 13. Index curves of the learning processes of the LA-I and LA-II.

of the adaptive algorithm of linear mixture case [31]. The final comparison results confirm the learning algorithm again.

Example 3: To examine the robustness of the proposed method with respect to the noise interference consider a nonlinear mixture corrupted by noise. We reexamine the simulation in Example 2 with the same parameters in the presence of noise. The proposed model is again simulated for the nonlinear mixture corrupted by additive white noise of uniformly distributed on $[-a, a]$. We define the signal to noise ratio (SNR) as $\mathrm{SNR} = 20\log(A_i/a)$, where A_i denotes the amplitude of the ith signal.

For SNR to be 10 dB, the sources and the corrupted mixtures are shown on Figs. 15 and 16, respectively. The resulting separated signals at the output by LA-II after the learning process of the algorithms are depicted on Fig. 17. Since the learning process and the separation results of the present model by using LA-I are similar to that in LA-II in the presence of noise, so we only illustrate the separation results of LA-II here. From these figures, a successful separation process is also observed in spite

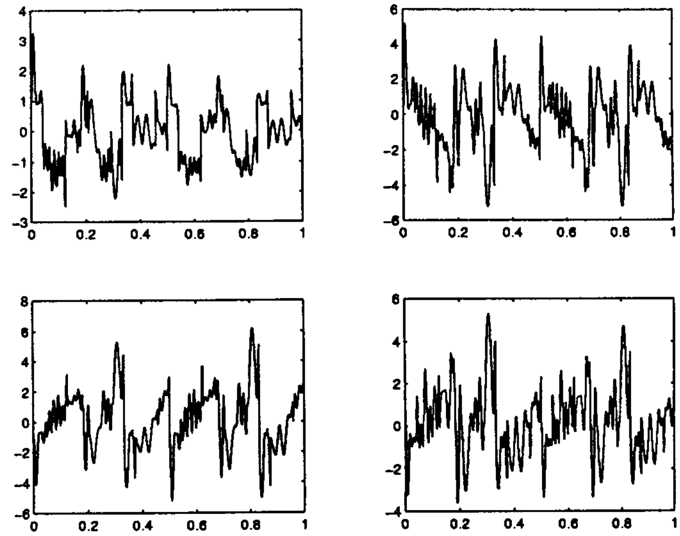

Fig. 14. Separated signals of the adaptive algorithm for linear mixture case.

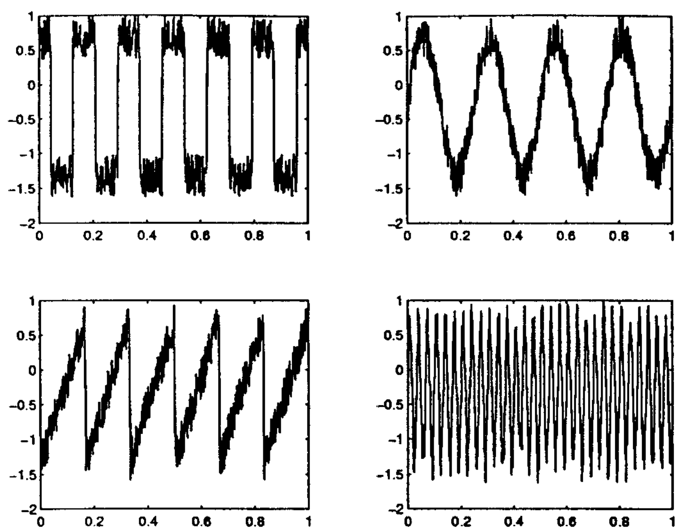

Fig. 15. Four source signals corrupted by noise.

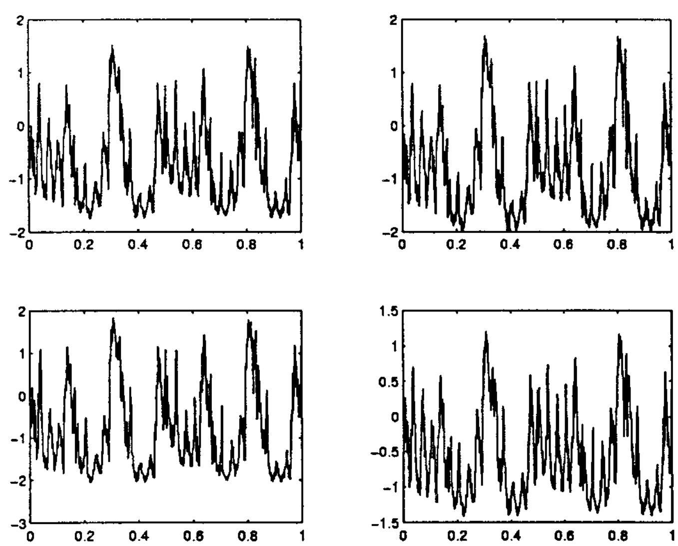

Fig. 16. Mixtures of noisy source signals.

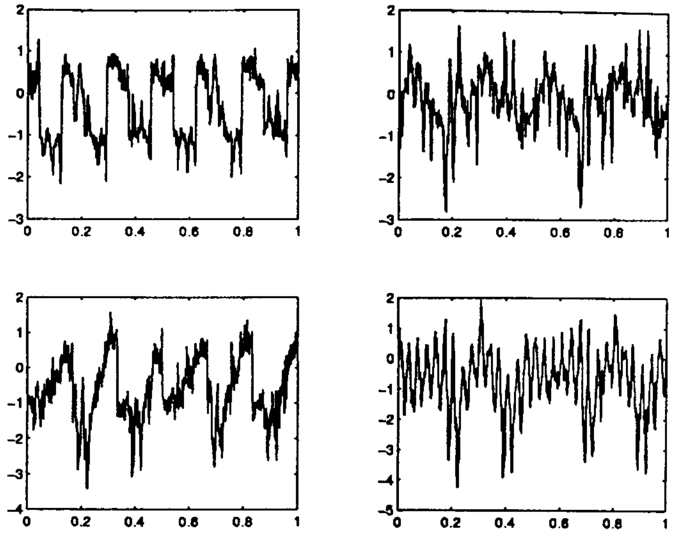

Fig. 17. Separated signals of the nonlinear demixing system in noisy case.

of the presence of nonlinearity and noise. This experimental result confirms the robustness of the algorithm versus noise corruption of the mixtures. The separated signals also contain additive noise because the present algorithm is only a signal separation algorithm and not a noise suppression algorithm. Further noise suppression would require more information of the statistical characteristics of the noise. To our knowledge, there exists a number of noise reduction methods and algorithms in the signal processing literature. One can choose an appropriate method according to the given type on noise if the noise suppression of the separated signals is needed.

VIII. Concluding Remarks

In this paper, a blind signal separation approach based on an RBF network is developed for the separation of nonlinearly mixed sources by defining a novel contrast function. The novelty of this new approach is in the use of the RBF architecture as well as in the novel contrast function. This contrast function consists of mutual information and cumulants matching and results in diminishing the indeterminacies caused by nonlinearity. Two learning algorithms for the parameters of the RBF separating system are developed with use of the stochastic gradient descent method and unsupervised clustering method. Due to the local response feature of RBF networks, this proposed model is characterized by fast learning convergence rate of weights, natural unsupervised learning characteristics, modular network structure. All of these properties make it be an effective candidate for fast and real-time multichannel signal separation. A number of experiments have been carried out to verify the claims in this paper. All simulation results show that this kind of separating model with two proposed learning algorithms is characterized by fast convergence and strong denonlinearity capability. In the simulations we have occasionally encountered a few phantom solutions, which are probably caused by the finite order moment matching and local minima of the contrast function. Therefore, our future investigations will focus on the uniqueness condition of inverse of the nonlinear mixing transform in addition to the independence of the source signals.

References

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

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

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

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

[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] J. Herault and C. Jutten, "Space or time adaptive signal processing by neural network models," in *Proc. AIP Conf.*, J. S. Denker, Ed., Snowbird, UT, 1986, pp. 206–211.

[7] S. Li and T. J. Sejnowski, "Adaptive separation of mixed sound sources with delays by a beamforming Herault-Jutten network," *IEEE J. Oceanic Eng.*, vol. 20, pp. 73–79, 1995.

[8] U. A. Lindgren and H. Broman, "Source separation using a criterion based on second-order statistics," *IEEE Trans. Signal Processing*, vol. 46, pp. 1837–1850, 1998.

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

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

[11] A. Hyvärinen, "New approximations of differential entropy for independent component analysis and projection pursuit," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1999, vol. 10, pp. 273–279.

[12] G. Burel, "Blind separation of sources: A nonlinear neural algorithm," *Neural Networks*, vol. 5, pp. 937–947, 1992.

[13] G. Deco and W. Brauer, "Nonlinear higher-order statistical decorrelation by volume-conserving neural architectures," *Neural Networks*, vol. 8, pp. 525–535, 1995.

[14] P. Pajunen, A. Hyvarinen, and J. Karhunen, "Nonliner blind source separation by self-organizing maps," in *Progress in Neural Information Processing: Proc. ICONIP'96*, vol. 2. New York, 1996, pp. 1207–1210.

[15] P. Pajunen, "Blind source separation using algorithmic information theory," in *Neurocomput.*: Elsevier, 1998, vol. 22, pp. 35–48.

[16] T.-W Lee, B. Koehler, and B. U. Orglmeister, "Blind source separation of nonlinear mixing model," in *Proc. 1997 IEEE Workshop, Neural Networks for Signal Processing VII*, Oct. 1997, pp. 405–415.

[17] M. Herrmann and H. H. Yang, "Perspectives and limitations of self-organizing maps in blind separation of source signals," in *Progress in Neural Information Processing: Proc. ICONIP'96*, 1996, pp. 1211–1216.

[18] J. K. Lin, D. G. Grier, and J. D. Cowan, "Source separation and density estimation by faithful equivariant SOM," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1997, vol. 9.

[19] H. H. Yang, S. Amari, and A. Cichocki, "Information backpropagation for blind separation of sources from nonlinear mixture," in *Proc. IEEE ICNN*, TX, 1997, pp. 2141–2146.

[20] A. Taleb, C. Jutten, and S. Olympieff, "Source separation in post nonlinear mixtures: An entropy-based algorithm," in *Proc. ESANN'98*, 1998, pp. 2089–2092.

[21] A. Hyvarinen and P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results," *Neural Networks*, vol. 12, pp. 429–439, 1999.

[22] S. Haykin, *Neural Networks, A Comprehensive Foundation*: Prentice-Hall, 1994.

[23] T. Poggio and F. Girosi, "A Theory of Networks for Approximation and Learning," MIT, Tech. Rep. AI 1140, 1989.

[24] H. Lappalainen and X. Giannakopoulos, "Multilayer perceptrons as nonlinear generative models for unsupervised learning: A Bayesian treatment," in *Proc. Int. Conf. Artificial Neural Networks (ICANN'99)*, Edinburgh, U.K., 1999, pp. 19–24.

[25] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95, 1980.

[26] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, pp. 281–294, 1989.

[27] S. Chen, S. McLaughlin, and B. Mulgrew, "Complex-valued radial-basis function network, part I: Network architecture and learning algorithm," *Signal Processing*, vol. 35, pp. 19–31, 1994.

[28] S. Amari and A. Cichocki, "Adaptive blind signal processing—Neural network approach," *Proc. IEEE*, vol. 86, pp. 2026–2048, 1998.

[29] S. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separtion," in *Advances in Neural Information Processing Systems*, vol. 8, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds. Cambridge, MA, 1996, pp. 757–763.

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

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

Ying Tan (M'98) was born in Yingshan County, Sichuan Province, China, on September 5, 1964. He received the B.S. degree in electronic engineering from University of Science and Technology of China, Hefei, in 1985, the M.S. degree in electronic engineering from Xidian University, Xi'an, China, in 1988, and the Ph.D. degree in electronic engineering from Southeast University, Nanjing, in 1997.

He was a Teaching Assistant Professor from 1988 to 1989, then a Lecturer since 1990. From 1997 to 1999, he was a Postdoctoral Research Fellow then an Associate Professor in the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China. He visited the Chinese University of Hong Kong in 1999. His research interests include neural-network theory and its applications, computational intelligence, signal and information processing, pattern recognition, and statistical signal analysis and processing. He has published more than 60 academic papers in refereed journals and conferences in these areas.

Dr. Tan has received a number of academic awards and research achievement awards from his universities and country due to his outstanding contributions and distinguished works. He was recently awarded the 1998 Wong Kuan-cheng Postdoctoral Fellowship of Chinese Academy of Science for his outstanding research contributions. He is currently a member of the IEEE Signal Processing and Communications Societies, and a member of the IEE Signal Processing Society, and a Senior Member of China Institute of Electronics.

Jun Wang (S'89–M'90–SM'93) received the B.S. degree in electrical engineering and the M.S. degree in systems engineering from Dalian Institute of Technology (now renamed as Dalian University of Technology), Dalian, China. He received the Ph.D. degree in systems engineering from Case Western Reserve University, Cleveland, OH.

He is an Associate Professor in the Department of Mechanical and Automation Engineering at the Chinese University of Hong Kong. Prior to this position, he was an Associate Professor at the University of North Dakota, Grand Forks. He is the author or coauthor of more than 60 journal papers, ten book chapters, three edited books, and numerous conference papers. His current research interests include theory and methodology of neural networks and their engineering applications.

Jacek M. Zurada (M'82–SM'83–F'96) received the Ph.D. degree with honors from the Technical University of Gdansk, Poland, in 1975.

He is the Samuel T. Fife Alumni Professor of Electrical and Computer Engineering at the University of Louisville, Louisville, KY. He has authored or coauthored more than 160 papers in the area of neural networks and circuits and systems for signal processing. He authored the text *Introduction to Artificial Neural Systems* (New York: PWS, 1992), coedited the book *Computational Intelligence: Imitating Life* (Piscataway, NJ: IEEE Press, 1994) with R. J. Marks and C. J. Robinson, and the book *Knowledge-Based Neurocomputing* (Cambridge, MA: MIT Press, 2000) with I. Cloete. He has been invited speaker on learning algorithms, associative memory, neural networks for optimization, and hybrid techniques for rule extraction at many national and international conferences.

Dr. Zurada has been the Editor-in-Chief of IEEE TRANSACTIONS ON NEURAL NETWORKS since 1998. He was Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS from 1992 to 1997 and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, Part I from 1997 to 1999 and Part II from 1993 to 1997. He was Guest Coeditor of the Special Issue on Soft Computing of Neurocomputing 1996 and Guest Coeditor of the Special Issue on Neural Networks and Hybrid Intelligent Models of IEEE TRANSACTIONS ON NEURAL NETWORKS. He has received a number of awards for distinction in research and teaching and he is an IEEE Distinguished Speaker.