Blind Source Recovery: Algorithms for Static and Dynamic Environments

Fathi M. Salam[1], Gail Erten[2] and Khurram Waheed[1]

[1]Department of ECE, Michigan State University, East Lansing, MI 48824-1226
[2]IC Tech Inc, 4295 Okemos Road, Suite 100, Okemos, MI 48864

Abstract

This paper integrates our contributions in the domain of blind source separation (BSS) and blind source deconvolution (BSD) both in static and dynamic environments. We focus on the use of the state space formulation and the development of a generalized optimization framework, using Kullback-Liebler divergence as the performance measure subject to the constraints of a state space representation. Various special cases are subsequently derived from this general case and are compared with material in recent literature. Some of this reported work has also been implemented in dedicated hardware/software and experimental designs have been compared with their computer simulations.

1. Introduction

The state space approach formulation of signal separation, extraction and recovery was introduced in [5] for linear time-invariant systems. The state notion summarizes weighted past as well as filtered versions of input signals. There are several reasons for this preference. Although transfer function models are equivalent to the state space models when initial conditions are zero, it is difficult to exploit any common features that may be present in dynamic systems. The main advantage of the state space description for blind deconvolution stems from the fact that it not only gives an efficient internal description of a system, but there are various possible equivalent state space realizations for a system, more important being the canonical observable and controllable forms.

The inverse for a state space representation is easily derived subject to the invertibility of the instantaneous relational matrix between input-output, in case the matrix is not square the condition reduces to the existence of pseudo-inverse of this matrix. It is well known how to parameterize some specific classes of models, which are of interest in many applications. In particular, the state space model enables much more general description than standard finite/infinite impulse response (FIR/IIR) convolutive filtering. All known filtering (dynamic) models, like AR, MA, ARMA, ARMAX and Gamma filtering, could also be considered as special cases of flexible state space models.

2. Adaptive Framework for Blind Signal Recovery

In the most general setting, the mixing/convolving environment may be represented by an unknown dynamic process $\bar{\mathbf{H}}$ with inputs being the independent sources $\underline{s}$ and the outputs being the measurements $\underline{\boldsymbol{m}}$. In this extreme case, no structure is assumed about the model of the environment.

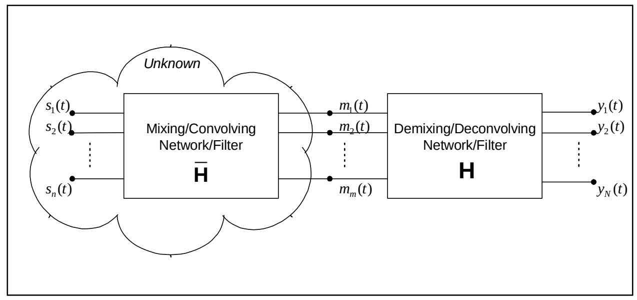

Figure 1: General Blind Source Recovery Framework

The environment can be modeled as a dynamic system with fixed but unknown parameters. The processing network $\mathbf{H}$ must be constructed with the capability to compute the "inverse" (or the "closest to an inverse") of the environment model.
It is possible that an augmented network be constructed so that the inverse of the environment is merely a subsystem of the network with learning. In this case, even if the environment is unstable (due to existence of non-minimum phase zeros), the overall augmented network may represent overall a nonlinear adaptive dynamic system which may converge to the parameters as a stable equilibrium points. Thus achieving the global task of blind identification [8].

3. The Performance Functional

We employ the divergence of a random output vector $\underline{\boldsymbol{y}}$ as our performance measure.

The simplified functional if the statistical properties of the output signal are ergodic is given by [9]

$$L(y) = \sum_{y \in Y} p_y(y(k)) \ln \left| \frac{p_y(y(k))}{\prod_{i=1}^{n} p_{y_i}(y_i(k))} \right| \quad (3.1)$$

Further simplification can be achieved using the assumption that as the algorithm approaches convergence the components of the output vector ***y*** will become statistically less dependent, therefore the above functional can be re-written in mutual information form using the entropy i.e.

$$L(y) = -H(y) + \sum_{i=1}^{n} H(y_i) \quad (3.2)$$

where the entropy of signal ***y*** is

$$H(y) = -E\left[\ln\left|p_y(y)\right|\right] = -\int_{y \in Y} p_y(y) \ln\left|p_y(y)\right| dy \quad (3.3)$$

4. Algorithm for Nonlinear Dynamic Case

Assume that the environment can be modeled as the following nonlinear discrete-time dynamic forward model

$$X_e(k+1) = f_e(X_e(k), s(k), h_1) \quad (4.1)$$

$$m(k) = g_e(X_e(k), s(k), h_2) \quad (4.2)$$

where

$s(k)$ - n dimensional vector of original sources

$m(k)$ - m dimensional vector of measurements

$X_e(k)$ - N_e dimensional state vector

h_1 - constant parameter vector (or matrix) of dynamic state equation

h_2 - constant parameter vector (or matrix) of output equation

$f_e(.)$ and $g_e(.)$ - differentiable non-linear functions

Further it is assumed that existence and uniqueness of solutions are satisfied for any given initial conditions $X_e(t_o)$ and sources $s(k)$

The processing network model may be represented by a dynamic feedforward or feedback network. Focussing on feedforward network, we assume the network to be

$$X(k+1) = f(X(k), m(k), w_1) \quad (4.3)$$

$$y(k) = g(X(k), m(k), w_2) \quad (4.4)$$

where

$m(k)$ - m dimensional measurement vector

$y(k)$ - N dimensional output vector

$X(k)$ - L dimensional state vector

w_1 - parameters of the network state equation

w_2 - parameters of the network output equation

$f(.)$ and $g(.)$ - differentiable non-linear functions

Further the assumption of existence and uniqueness of solutions of the differential equations is also assumed for the network model for any given initial conditions $X(t_o)$ and measurement vector $m(k)$.

In order to derive the update law, we formulate the following optimization problem [8], note the notation has been altered a little for convenience.

Minimize

$$J_o(w_1, w_2) = \sum_{k=n_o}^{N-1} L^k(y_k) \quad (4.5)$$

subject to

$$X_{k+1} = f^k(X_k, m_k, w_1) \quad (4.6)$$

$$y_k = g^k(X_k, m_k, w_2) \quad (4.7)$$

with the initial conditions X_{k_o}

The augmented cost functional to be optimized becomes

$$J(w_1, w_2) = \sum_{k=n_o}^{N-1} L^k(y_k) + \lambda_{k+1}^T(f^k(X_k, m_k, w_1) - X_{k+1}) \quad (4.8)$$

Define the Hamiltonian as

$$\mathrm{H}^k = L^k(y_k) + \lambda_{k+1}^T f^k(X, m, w_1) \quad (4.9)$$

Consequently, the *necessary* conditions for optimality are

$$X_{k+1} = \frac{\partial \mathrm{H}^k}{\partial \lambda_{k+1}} = f^k(X, m, w_1) \quad (4.10)$$

$$\lambda_k = \frac{\partial \mathrm{H}^k}{\partial X_k} = (f_{X_k}^k)^T \lambda_{k+1} + \frac{\partial L^k}{\partial X_k} \quad (4.11)$$

and the change in weighting parameters become

$$\Delta w_1 = -\eta \frac{\partial \mathrm{H}^k}{\partial w_1} = -\eta (f_{w_1}^k)^T \lambda_{k+1} \quad (4.12)$$

$$\Delta w_2 = -\eta \frac{\partial \mathrm{H}^k}{\partial w_2} = -\eta \frac{\partial L^k}{\partial w_2} \quad (4.13)$$

5. Algorithm for Linear Dynamic Case

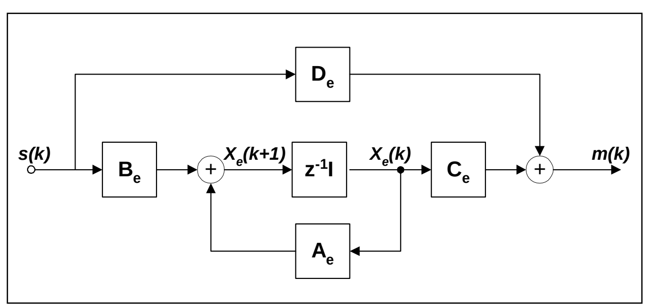

Figure 2: Linear Dynamic Environment Model

In the linear dynamic case, the environment models is assumed to be of the state space form

$$X_e(k+1) = A_e X_e(k) + B_e s(k) \quad (5.1)$$

$$m(k) = C_e X_e(k) + D_e s(k) \quad (5.2)$$

In this case the feedforward separating network will attain the state space form

$$X(k+1) = A\, X(k) + B\, m(k) \quad (5.3)$$

$$y(k) = C\, X(k) + D\, m(k) \quad (5.4)$$

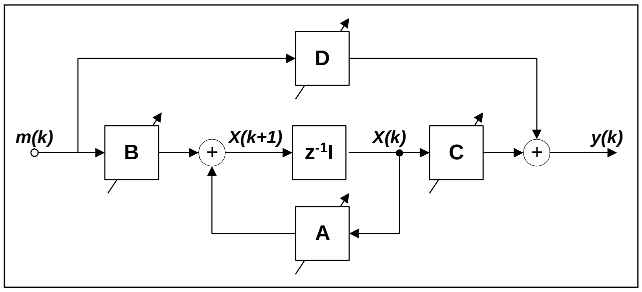

Figure 3: State Space Separating Framework

The existence of an explicit solution in this case has been earlier shown in papers by Salam et. al. [6]. This existence of solution ensures that the network has the capacity to compensate for the environment and consequently recovers the original signals.

For the linear time-invariant case the, the above derived update laws simplifies to

$$X_{k+1} = \frac{\partial H^k}{\partial \lambda_{k+1}} = f^k(X, m, w_1) = A\, X_k + B\, m_k \quad (5.5)$$

$$\lambda_k = \frac{\partial H^k}{\partial X_k} = (f^k_{X_k})^T \lambda_{k+1} + \frac{\partial L^k}{\partial X_k} = A_k^T \lambda_k + C_k^T \frac{\partial L^k}{\partial y_k} \quad (5.6)$$

$$\Delta A = -\eta \frac{\partial H^k}{\partial A} = -\eta (f^k_A)^T \lambda_{k+1} = -\eta \lambda_{k+1} X_k^T \quad (5.7)$$

$$\Delta B = -\eta \frac{\partial H^k}{\partial B} = -\eta (f^k_B)^T \lambda_{k+1} = -\eta \lambda_{k+1} m_k^T \quad (5.8)$$

$$\Delta C = -\eta \frac{\partial H^k}{\partial C} = -\eta \frac{\partial L^k}{\partial C} = -\eta \varphi(y) X^T \quad (5.9)$$

$$\Delta D = -\eta \frac{\partial H^k}{\partial D} = -\eta \frac{\partial L^k}{\partial D} = \eta([D]^{-T} - \varphi(y) m^T) \quad (5.10)$$

The above derived update laws form a comprehensive algorithm and provides the update laws for the states, the co-states and all the parametric matrices in the state space. The invertibility of the state space as discussed in [6] is guaranteed if the matrix D is invertible. In the above derived laws

η - learning rate of the algorithm

$[D]^{-T}$ - represents the transpose of the inverse of the matrix D if it is a square matrix or the transpose of its pseudo-inverse in case it is rectangular in structure.

$\varphi(y)$ - represents an element wise nonlinearity acting individually on each component of the output vector $\underline{y}$, i.e.

$$\varphi(y) = -\frac{\partial p(y) / \partial y}{p(y)} \quad (5.11)$$

The update laws in (5.9) and (5.10) are similar to the gradient descent results [4,8], indicating its optimality. The update law provided above although non-causal, can be easily implemented using some delay and memory storage in a manner similar to the natural gradient implementation for combined BSS/BSD problems. A delay in the recovered signal is acceptable in the blind signal recovery problem as long as the delay is fixed for all the recovered signal.

The update laws for the matrix $\boldsymbol{D}$ can be further elaborated as

$$\Delta D = \eta([D]^{-T} - \varphi(y) m^T) = \eta(I - \varphi(y)(Dm)^T)[D]^{-T} \quad (5.12)$$

where

I – Identity matrix of the dimensions of $[D]$

and defining the component of output due to the instantaneous mixture input as

$$y_m = D\, m \quad (5.13)$$

Further using a positive definite matrix $D^T D$ post-multiplication of (5.13), we obtain the following modified update law with reduced computational cost requiring no matrix inversion.

$$\Delta D = \eta([D]^{-T} - \varphi(y) m^T) D^T D = \eta(I - \varphi(y) y_m^T) D \quad (5.14)$$

6. Natural Gradient Algorithm

Formal formulation for deriving the update laws for the problem can be done using the output equation (5.4)

(note the instantaneous time index k has been dropped for convenience)

Defining vectors $\tilde{y}$ and $\tilde{x}$, and the matrix $\tilde{W}$ as

$$\tilde{y} = \begin{bmatrix} y \\ X \end{bmatrix} \qquad \tilde{x} = \begin{bmatrix} m \\ X \end{bmatrix} \qquad \tilde{W} = \begin{bmatrix} D & C \\ 0 & I \end{bmatrix} \quad (6.1)$$

where

$$\tilde{y} = \tilde{W}\, \tilde{x} \quad (6.2)$$

The update law for this augmented parameter matrix $\tilde{W}$ is similar in form to (5.10) or the stochastic gradient law derived for the static mixing case

$$\Delta \tilde{W} = \eta \left[\tilde{W}^{-T} - \varphi(\tilde{y}) \tilde{x}^T \right] \quad (6.3)$$

we have

$$\tilde{W}^T = \begin{bmatrix} D^T & 0 \\ C^T & I \end{bmatrix} \tag{6.4}$$

Consequently for the general case where ***D*** may not be square, its inverse (assuming the pseudo-inverse for ***D*** to exist), we get

$$\tilde{W}^{-T} = \begin{bmatrix} D(D^T D)^{-1} & 0 \\ -C^T D(D^T D)^{-1} & I \end{bmatrix} \tag{6.5}$$

Factoring out the augmented weight term $\tilde{W}^{-T}$, (6.3) can be written as

$$\Delta\tilde{W} = \eta\left[I - \varphi(\tilde{y})\tilde{x}^T\tilde{W}^T\right]\tilde{W}^{-T} \tag{6.6}$$

Post-multiplying by the matrix $\tilde{W}^T\tilde{W}$, the update law becomes

$$\Delta\tilde{W} = \eta\left[I - \varphi(\tilde{y})\tilde{x}^T\tilde{W}^T\right]\tilde{W} \tag{6.7}$$

Writing in terms of the original state space variables the update law (6.7) is given by

$$\Delta\begin{bmatrix} D & C \\ 0 & I \end{bmatrix} = \eta\left[I - \varphi\begin{bmatrix} y \\ X \end{bmatrix}\begin{bmatrix} m^T & X^T \end{bmatrix}\begin{bmatrix} D^T & 0 \\ C^T & I \end{bmatrix}\right]\begin{bmatrix} D & C \\ 0 & I \end{bmatrix} \tag{6.8}$$

considering the update laws for matrices ***C*** and ***D*** only

$$\begin{bmatrix} \Delta D & \Delta C \end{bmatrix} = \eta[D \quad C] - \eta\varphi(y)\times \begin{bmatrix} m^T D^T + X^T C^T & X^T \end{bmatrix}\begin{bmatrix} D & C \\ 0 & I \end{bmatrix} \tag{6.9}$$

observe that

$$y^T = m^T D^T + X^T C^T \tag{6.10}$$

therefore the final instantaneous update laws for the matrices C and D are

$$\Delta C(k) = \eta\left((I - \varphi(y(k))y^T(k))C(k) - \varphi(y(k))X^T(k)\right) \tag{6.11}$$

$$\Delta D(k) = \eta(I - \varphi(y(k))y^T(k))D(k) \tag{6.12}$$

The update laws for the natural gradient update derived in [4] are in exact agreement with the update derived above. The update laws in (6.11) and (6.12) are related to the earlier derived update laws (5.9) and (5.10) by the relation

$$\tilde{\nabla}l = \nabla l\begin{bmatrix} I + C^T C & C^T D \\ D^T C & D^T D \end{bmatrix} = \nabla l\begin{bmatrix} I & 0 \\ C & D \end{bmatrix}^T\begin{bmatrix} I & 0 \\ C & D \end{bmatrix} \tag{6.13}$$

where

∇l - gives the update according to normal stochastic gradient, the conditioning matrix in (6.13) is symmetric and positive definite.

7. Special Case: Filtering Structures

One of the advantage of using the state space models is that signal processing filter structures form special cases where the constituent matrices of the dynamic state equation takes specific forms.

7.1 IIR Filtering

Consider the case where the matrices ***A*** and ***B*** are in the controllable canonical form. In this case ***A*** and ***B*** are conveniently represented as

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & \cdots & A_{1L} \\ I & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \text{ and } B = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{7.1}$$

where

A – matrix of dimension Lm $\times$ Lm.

A_{1j} - Block sub-matrix of dimension m $\times$ m, may be simplified to a diagonal matrix

I – Identity matrix of dimension m $\times$ m

0 – Zero matrix of dimension m $\times$ m

B – matrix of dimension Lm $\times$ m

the state matrix is given by

$$X_k = X(k) = \begin{bmatrix} X_1(k) \\ X_2(k) \\ \vdots \\ X_L(k) \end{bmatrix} \tag{7.2}$$

where

$X(k)$ - is Lm $\times$ m dimensional state vector, and each

$X_j(k)$ - is an m $\times$ m dimensional state vector

For this model the state model reduces to the following set of equations representing an IIR filtering structure.

$$\begin{aligned} X_1(k+1) &= \sum_{j=1}^{L} A_{1j}X_j(k) + m(k) \\ X_2(k+1) &= X_1(k) \\ &\vdots \\ X_L(k+1) &= X_{L-1}(k) \\ y(k) &= \sum_{j=1}^{L} C_j X_j(k) + Dm(k) \end{aligned} \tag{7.3}$$

In the IIR filtering case, there is no update law required for *B*, while for the matrix *A* we need to update only the first block row and therefore the update law reduces to

$$\Delta A_{1j} = -\eta\frac{\partial \mathrm{H}^k}{\partial A_{1j}} = -\eta(f_{A_{1j}}^k)^T\lambda_{k+1} = -\eta\lambda_1 X_j^T(k) \tag{7.4}$$

The special structure in matrix *A* also effects the update law for the co-state equations (propagating in future time) which reduces to

$$\lambda_1(k) = \lambda_2(k+1) + C_1^T \frac{\partial L^k}{\partial y_k}(k)$$
$$\lambda_2(k) = \lambda_3(k+1) + C_2^T \frac{\partial L^k}{\partial y_k}(k) \qquad (7.5)$$
$$\vdots$$
$$\lambda_L(k) = C_L^T \frac{\partial L^k}{\partial y_k}(k)$$

solving specifically for time k and then using time shift for time k+1, we obtain the recursive form for the update as

$$\lambda_1(k) = \sum_{j=1}^{L} C_j^T \frac{\partial L^k}{\partial y_k}(k+j-1)$$
$$\lambda_1(k+1) = \sum_{j=1}^{L} C_j^T \frac{\partial L^k}{\partial y_k}(k+j) \qquad (7.6)$$

which is in exact accordance with the natural gradient algorithm derived for blind source separation/ deconvolution by Amari et. al. [3,4] and can be implemented in a similar fashion by using the usual time delayed version of the algorithm and buffer storage memory.

Using (7.6), we can further simplify (7.4) for the update of the block sub-matrices in *A* as

$$\Delta A_{1j} = -\eta \sum_{j=1}^{L} C_j^T \frac{\partial L^k}{\partial y_k}(k+j) X_j^T \qquad (7.7)$$

7.2 FIR Filtering

In case, the first row block sub-matrices A_{1j} are zero, then the above filtering model (7.3) reduces to the FIR filtering structure. In this case the state space model reduces to

$$X_1(k+1) = m(k)$$
$$X_2(k+1) = X_1(k)$$
$$\vdots \qquad (7.8)$$
$$X_L(k+1) = X_{L-1}(k)$$
$$y(k) = \sum_{j=1}^{L} C_j X_j(k) + Dm(k)$$

where the state dynamic equations reduce merely to the delays of the mixture inputs i.e.

$$X_1(k) = m(k-1)$$
$$X_2(k) = m(k-2)$$
$$\vdots \qquad (7.9)$$
$$X_L(k) = m(k-L)$$

In this case only matrices *C* and *D* need to be updated as both matrices A and B contain only block identity and zero matrices and are absorbed in (7.9)

The doubly finite FIR filter approximations are handled by using time delayed versions of the algorithm and the requirement of buffered storage, with the advantage of being able to converge at stable solutions.

The problem of the filter approximation length and resulting delay can also be handled in the frequency domain implementations of the algorithm where 1024 or higher tap double sided FFT equivalent frequency domain FIR filter is the converged solution. The result can be converted to the time domain equivalent by appropriately using the inverse FFT followed by chopping or windowing techniques to contain maximum possible filter energy and minimizing algorithm delay and hence the buffering memory requirement [1].

8. Effect of non-linearity

As defined above in (5.11) the optimal nonlinearity for the update laws depends on the probability density function of the outputs which upon convergence is similar to the probability density of the independent sources. Most of the literature on BSS/BSD has focussed on use of specific non-linearities chosen based on the known distribution/density of sources. This is equivalent to converting a blind problem to a semi-blind problem where some information about the sources is assumed to be known.

We have used the non-linearity derived for static cases in [2] and have used it successfully for the combined BSS/BSD or filtering/mixing environment. The non-linearity is given as

$$\varphi(y) = y + \kappa_4(y)\tanh(\beta y) \text{ where } \begin{cases} \kappa_4(y) \geq 1 : \text{super-gaussian} \\ \kappa_4(y) \leq -1 : \text{sub-gaussian} \end{cases} \qquad (8.1)$$

where

$\kappa_4(y)$ - batch kurtosis of the output signals

Using this nonlinearity, the algorithm is able to converge for a wide class of source densities [2] including certain mixtures of different densities. The nonlinearity, however, does not always give good results for sources with densities close to Gaussian.

9. Simulations

In this section, we present various simulation results for a number of source densities. The convergence performance of the algorithms are presented using the mutichannel intersymbol interference [4] which is defined as

$$ISI_k = \sum_{i=1}^{N} \frac{\left|\sum_j \sum_p \left|G_{pij}\right| - \max_{p,j}\left|G_{pij}\right|\right|}{\max_{p,j}\left|G_{pij}\right|} + \sum_{j=1}^{N} \frac{\left|\sum_i \sum_p \left|G_{pij}\right| - \max_{p,i}\left|G_{pij}\right|\right|}{\max_{p,i}\left|G_{pij}\right|} \quad (9.1)$$

where $G(z)$ – Global Transfer Function

$$G(z) = W(z) * H(z) \quad (9.2)$$

and

$H(z)$ –Transfer Function of the Environment

$W(z)$ –Transfer Function of the network, given by

$$W = C(zI - A)^{-1} B + D \quad (9.3)$$

In this paper we present the results for the non-minimum phase FIR filtering environment with state space representation equivalent to the FIR model

$$m(k) = \sum_{i=0}^{n-1} H_i s(k-i) + v(k) \quad (9.4)$$

where

$$H_0 = \begin{bmatrix} 1 & 0.5 \\ -0.2 & 0.2 \end{bmatrix}, H_1 = \begin{bmatrix} 1 & -0.3 \\ 0.4 & 1 \end{bmatrix}, H_2 = \begin{bmatrix} -0.75 & 0.2 \\ 0.7 & 0 \end{bmatrix} \quad (9.5)$$

$v(k)$ - additive noise

The problem is formulated in the state space framework as explained earlier. The theoretical inverse of this FIR filter will be an unstable IIR filter with both poles and zeros present outside the unit circle. However, as remarked earlier we set up the problem for a doubly finite FIR filter inverse with 31 taps. The algorithm is applied to the gamma, uniform, bimodal and a combination of the above and is able to converge in all cases with the worst ISI of around -16dB in the case of uniform distributed data.

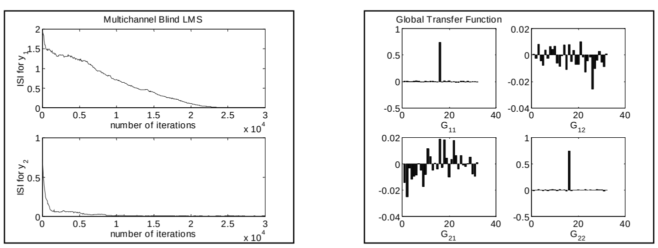

Figure 4: Results for gamma distributed data

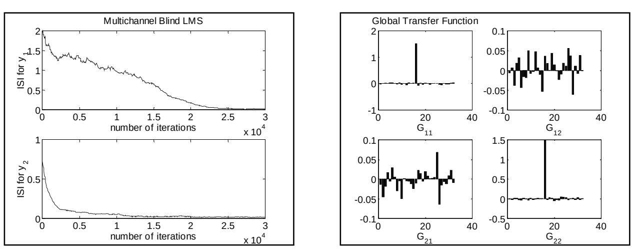

Figure 5: Results for uniform distributed sources

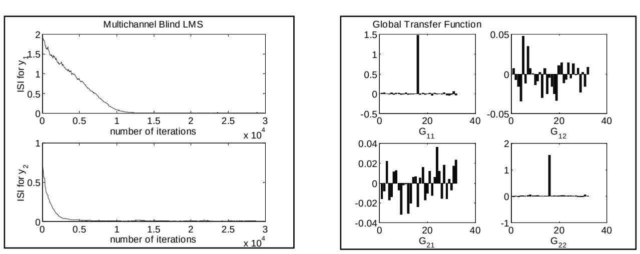

Figure 6: Results for bimodal distributed sources

The algorithms work blindly without any assumption about the distribution structure of the sources but the convergence speed is still comparable to the cases where an optimal non-linearity is chosen for a certain distribution structure of the sources.

10. References

[1] T. W. Lee, A. J. Bell, and R. Orglmeister, “Blind Source Separation of Real World Signals ", *Proc. of IEEE Int’l Conf. on Neural Networks*, June 1997, Houston, pp 2129-2135.

[2] T. W. Lee, M. Girolami, A. J. Bell and T. Sejnowski: “A Unifying Information-Theoretic Framework for Independent Component Analysis””, *Int’l Journal of Computers and Mathematic with applications*, Vol 31 (11), pp. 1-21, March 2000

[3] S. Amari, S. Douglas, A. Cichocki, and H. Yang: “Multichannel blind deconvolution and equalization using the natural gradient.” in *Proc. of IEEE Wkshp on Signal Processing: Adv. in Wireless Comm.*, pp. 101–104, Paris, France, 1997.

[4] L. Zhang and A. Cichocki: “Blind deconvolution of dynamical systems: A state space approach”, *Journal of Signal Processing*, Vol. 4, No. 2, Mar. 2000, pp. 111-130.

[5] F. M. A. Salam. “An adaptive network for blind separation of independent signals.” *Proc. of Int’l Symp. on Circuits and Systems*, 1: pp. 431-1434, May 1993.

[6] A. B. Gharbi and F. M. Salam. “Implementation and experimental results of a chip for the separation of mixed and filtered signals.”; *Journal of Circuits, Systems and Computers*, 6(2): pp. 115-139, April 1996.

[7] Ammar B. A. Gharbi and Fathi Salam: “Algorithms for Blind Signal Separation and Recovery in Static and Dynamic Environments.” *IEEE Proc. Int’l Symp. on Circuits & Systems 1997*, June 9-12, Hong Kong.

[8] F. M. Salam and G. Erten; “The State Space Framework for Blind Dynamic Signal Extraction and Recovery”; *Proc. of Int’l Symp. on Circuits and Systems, 1999*, Volume: 5, 1999; pp. 66 –69

[9] G. Erten, F. M. Salam; “Voice Extraction by on-line signal separation and recovery”, *IEEE Trans. On Circuits and Systems II; Analog and Digital Signal Processing*, 1999, Volume: 46, Issue 7; pp. 915-922