State Space Feedforward and Feedback Structures for Blind Source Recovery

Fathi M. Salam and Khurram Waheed

Circuits, Systems, And Neural Networks Laboratory
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48824-1226

ABSTRACT

This paper presents two separate structures for the blind source recovery (BSR) of stochastically independent signal sources. We hypothesize linear state space models for both the mixing environment and the demixing (i.e. recovering) adaptive network. Separate algorithms for adaptive estimation of parameters for the feedforward and feedback recovering networks have been derived. Auxillay conditions for the convergence of these algorithms have also been derived and discussed. Simulation examples have been included to compare the results for both algorithms for an IIR mixing environment. Conclusive remarks about effectiveness of these techniques in various practical problems have also been included.

1. INTRODUCTION

Blind Source Recovery (BSR), or Multi-channel Blind Deconvolution (MCBD), is a practical adaptive filtering problem formulation that combines Blind Source Separation and Blind Source Deconvolution. Recently, Blind Source Recovery (BSR) has been a very active research area in the arena of adaptive signal processing and autonomous (or unsupervised) learning. The BSR problem denotes recovering original sources from environments that may include convolution, transients, and even possible nonlinearity. BSR has several potential application domains including e.g., wireless telecommunication systems, sonar and radar systems, audio and acoustics, image enhancement and biomedical signal processing (EEG/MEG, EOG, EMG, ECG signals).

The state space notion provides a compact representation, which is capable of handling both time delayed and filtered versions of signals in an organized manner [2,3,5,6]. Unlike the transfer function models, the state-space provides an efficient internal description of a system. Moreover, there are various possible equivalent state space realizations for a system, and thus recovery of original sources can be achieved independent from (and even in the absence of) environment *identifiability*, i.e. determining the exact (or a specific function of) parameters of the environment. There exist many adaptive network solutions (representations), which succeed in recovering the original signals even in the absence of precise identifiability [3,5]. The existence of solutions that enable the recovery of original sources have been expressed as *recoverability* [1,3]. The state space model enables much more general description than standard finite/infinite impulse response (FIR/IIR) convolutive filtering and all known filtering (dynamic) models, like AR, MA, ARMA, ARMAX and Gamma filtering can be considered as mere special cases.

Existence and constructions of a theoretical solution to the Blind Source Recovery problem can be easily derived using the state space, given a structure of the environment. The inverse for a state space representation is easily derived subject to the invertibility of the instantaneous relational mixing matrix between input-output – in case this matrix is not square; the condition reduces to the existence of pseudo-inverse of this matrix.

2. ALGORITHMS FOR LINEAR DYNAMIC CASE

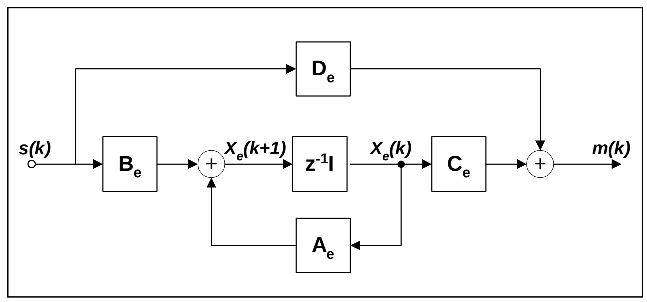

Figure 1: State Space Mixing Environment

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

$$X_e(k+1) = A_e X_e(k) + B_e s(k) \tag{2.1}$$

$$m(k) = C_e X_e(k) + D_e s(k) \tag{2.2}$$

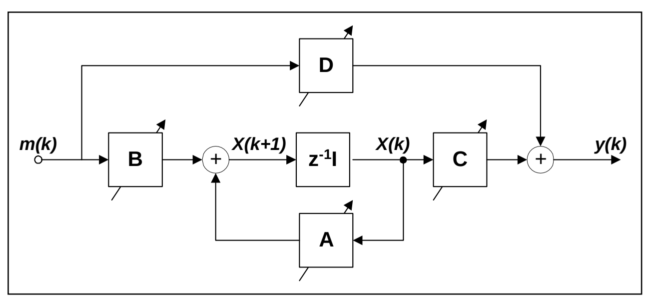

Figure 2: State Space Demixing Network

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

$$X(k+1) = A\,X(k) + B\,m(k) \tag{2.3}$$

$$y(k) = C\,X(k) + D\,m(k) \tag{2.4}$$

The existence of an explicit solution in this case has been earlier shown in papers by Salam et. al. [1]. This existence of solution ensures that the network has the capacity to compensate for the environment and consequently recovers the original signals.

The derivation of BSR algorithm is setup as an optimization problem subject to the constraints of multi-variable state space representation and the calculus of variations. Kullback-Lieblar divergence in the mutual information form is used as the performance ("distance") measure.

$$L(y) = -H(y) + \sum_{i=1}^{n} H(y_i) \quad (2.5)$$

where $H(y)$ is the entropy of signal ***y***, given by

$$H(y) = -E\left[\ln\left|p_y(y)\right|\right] = -\int_{y\in Y} p_y(y)\ln\left|p_y(y)\right| dy \quad (2.6)$$

Minimizing the performance functional

$$J_o(w_1, w_2) = \sum_{k=n_o}^{N-1} L^k(y_k) \quad (2.7)$$

subject to (2.3) and (2.4) with the initial conditions X_{k_o}

The augmented cost functional to be optimized becomes

$$J(w_1, w_2) = \sum_{k=n_o}^{N-1} L^k(y_k) + \lambda_{k+1}^T(A\,X_k + B\,m_k - X_{k+1}) \quad (2.8)$$

Define the Hamiltonian as

$$\mathrm{H}^k = L^k(y_k) + \lambda_{k+1}^T(A\,X_k + B\,m_k) \quad (2.9)$$

For the linear time-invariant case the, update laws are given by [3,5]

$$X_{k+1} = \frac{\partial \mathrm{H}^k}{\partial \lambda_{k+1}} = A\,X_k + B\,m_k \quad (2.10)$$

$$\lambda_k = \frac{\partial \mathrm{H}^k}{\partial X_k} = A_k^T \lambda_k + C_k^T \frac{\partial L^k}{\partial y_k} \quad (2.11)$$

$$\Delta A = -\eta \frac{\partial \mathrm{H}^k}{\partial A} = -\eta \lambda_{k+1} X_k^T \quad (2.12)$$

$$\Delta B = -\eta \frac{\partial \mathrm{H}^k}{\partial B} = -\eta \lambda_{k+1} m_k^T \quad (2.13)$$

$$\Delta C = -\eta \frac{\partial \mathrm{H}^k}{\partial C} = -\eta \frac{\partial L^k}{\partial C} = -\eta \varphi(y) X^T \quad (2.14)$$

$$\Delta D = -\eta \frac{\partial \mathrm{H}^k}{\partial D} = -\eta \frac{\partial L^k}{\partial D} = \eta([D]^{-T} - \varphi(y) m^T) \quad (2.15)$$

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 [1] 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 a nonlinearity acting individually on each component of the output vector ***y***, i.e.

$$\varphi(y) = -\frac{\partial p(y) / \partial y}{p(y)} \quad (2.16)$$

The update laws in (2.14) and (2.15) are similar to the gradient descent results [3], 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 MCBD problems. A delay in the recovered signal is acceptable in the BSR problem as long as the delay is fixed for every component recovered.

2.1 Feedforward Structure

We present a formal formulation for deriving the feedforward update laws for the problem using the output equation (2.4) [5,7]. This leads to modified update laws for the matrices *C* and *D*. Further, these new update laws have better convergence performance as compared to (2.14) and (2.15). (Note: the instantaneous time index k has been dropped for convenience)

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

$$\tilde{y} = \begin{bmatrix} y \\ X \end{bmatrix} \qquad \tilde{x} = \begin{bmatrix} m \\ X \end{bmatrix} \qquad \tilde{W} = \begin{bmatrix} D & C \\ 0 & I \end{bmatrix} \quad (2.17)$$

where

$$\tilde{y} = \tilde{W}\,\tilde{x} \quad (2.18)$$

The update law for this augmented parameter matrix $\tilde{W}$ is similar in form to (2.15) or the stochastic gradient law for the static mixing case.

$$\Delta\tilde{W} = \eta\left[\tilde{W}^{-T} - \varphi(\tilde{y})\tilde{x}^T\right] \quad (2.19)$$

since

$$\tilde{W}^T = \begin{bmatrix} D^T & 0 \\ C^T & I \end{bmatrix} \quad (2.20)$$

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

$$\tilde{W}^{-T} = \begin{bmatrix} D(D^T D)^{-1} & 0 \\ -C^T D(D^T D)^{-1} & I \end{bmatrix} \quad (2.21)$$

Factoring out the augmented weight term $\tilde{W}^{-T}$, (2.19) can be written as

$$\Delta\tilde{W} = \eta\left[I - \varphi(\tilde{y})\tilde{x}^T\tilde{W}^T\right]\tilde{W}^{-T} \quad (2.22)$$

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

$$\Delta\tilde{W} = \eta\left[I - \varphi(\tilde{y})\tilde{x}^T\tilde{W}^T \right]\tilde{W} \tag{2.23}$$

Using (2.18), we can write (2.23) as

$$\Delta\tilde{W} = \eta\left[I - \varphi(\tilde{y})\tilde{y}^T \right]\tilde{W} \tag{2.24}$$

Writing in terms of the original state space variables the update law (2.24) is given by

$$\Delta\begin{bmatrix} D & C \\ 0 & I_M \end{bmatrix} = \eta\left[I - \varphi\begin{bmatrix} y \\ X \end{bmatrix}\begin{bmatrix} y^T & X^T \end{bmatrix}\right]\begin{bmatrix} D & C \\ 0 & I_M \end{bmatrix} \tag{2.25}$$

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

$$\begin{bmatrix} \Delta D & \Delta C \end{bmatrix} = \eta[D \quad C] - \eta\left[\varphi(y)\, y^T \quad \varphi(y) X^T \right]\begin{bmatrix} D & C \\ 0 & I_M \end{bmatrix} \tag{2.26}$$

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

$$\Delta C(k) = \eta\left((I_N - \varphi(y(k)) y^T(k)) C(k) - \varphi(y(k)) X^T(k) \right) \tag{2.27}$$

$$\Delta D(k) = \eta (I_N - \varphi(y(k)) y^T(k)) D(k) \tag{2.28}$$

The update laws for the natural gradient update derived in [3] are in exact agreement with the update derived above. The update laws in (2.27) and (2.28) are related to the earlier derived update laws (2.14) and (2.15) by the relation

$$\tilde{\nabla} l = \nabla l \begin{bmatrix} I_M + C^T C & C^T D \\ D^T C & D^T D \end{bmatrix} = \nabla l \begin{bmatrix} I_M & 0 \\ C & D \end{bmatrix}^T \begin{bmatrix} I_M & 0 \\ C & D \end{bmatrix} \tag{2.29}$$

where

$$\nabla l = \begin{bmatrix} \dfrac{\partial L^k}{\partial C} & \dfrac{\partial L^k}{\partial D} \end{bmatrix} \tag{2.30}$$

gives the update according to normal stochastic gradient, the conditioning matrix in (2.29) is symmetric and positive definite.

Examining the update of the remaining terms in augmented weight matrix $\tilde{W}$, we have

$$0 = \varphi(X) y^T D \;\Rightarrow\; \varphi(X) y^T = 0 \tag{2.31}$$

Also

$$\Delta I_M = 0 = I_M - \varphi(X) y^T C - \varphi(X) X^T \tag{2.32}$$

Rearranging terms, we have

$$\varphi(X) y^T C + \varphi(X) X^T = I_M \tag{2.33}$$

Using the relation (2.31), the condition (2.33) reduces to

$$\varphi(X) X^T = I_M \tag{2.34}$$

The auxiliary relations (2.31) and (2.34) form supplementary conditions for the convergence of the algorithm and are satisfied in a stochastic sense upon convergence of the algorithm. These conditions are in addition to the stability conditions for the algorithm [3].

2.2 Feedback Structure

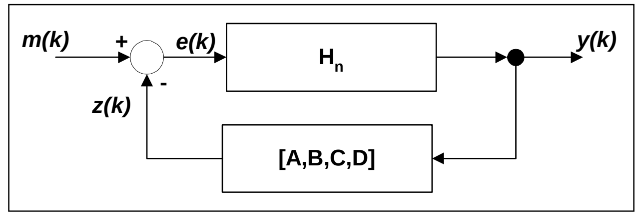

Figure 3: State Space Feedback Demixing Structure

For the feedback case, the network equations are [7]

$$X(k+1) = A\, X(k) + B\, y(k) \tag{2.35}$$

$$z(k) = C\, X(k) + D\, y(k) \tag{2.36}$$

Defining

$$e(k) = m(k) - z(k) \tag{2.37}$$

the output of the feedback structure is given by

$$y(k) = H_n * e(k) \tag{2.38}$$

For simplicity, assuming $H_n = I$, we have

$$y(k) = I * (m(k) - z(k)) = m(k) - z(k) \tag{2.39}$$

rearranging terms, we get

$$(I_N + D)\, y(k) \;+\; C\, X(k) = m(k) \tag{2.40}$$

In matrix form, we can rewrite (2.40) as

$$\begin{bmatrix} I_N + D & C \\ 0 & I_M \end{bmatrix}\begin{bmatrix} y \\ X \end{bmatrix} = \begin{bmatrix} m \\ X \end{bmatrix} \tag{2.41}$$

or

$$\begin{bmatrix} y \\ X \end{bmatrix} = \begin{bmatrix} I_N + D & C \\ 0 & I_M \end{bmatrix}^{-1}\begin{bmatrix} m \\ X \end{bmatrix} \tag{2.42}$$

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

$$\widetilde{X} = \begin{bmatrix} m \\ X \end{bmatrix},\; \widetilde{Y} = \begin{bmatrix} y \\ X \end{bmatrix},\; \text{and } \widetilde{W} = \begin{bmatrix} I_N + D & C \\ 0 & I_M \end{bmatrix} \tag{2.43}$$

we have

$$\widetilde{Y} = \widetilde{W}^{-1}\widetilde{X} = \approx{W}\widetilde{X} \text{ where } \approx{W} = \widetilde{W}^{-1} \tag{2.44}$$

Using the natural gradient, the update law for $\approx{W}$ is

$$\Delta\approx{W} = \eta\left[\approx{W}^{-T} - \varphi(\widetilde{Y})\widetilde{X}^T \right]\approx{W}^T\approx{W} \tag{2.45}$$

$$\Delta\approx{W} = \eta\left[I_{M+N} - \varphi(\widetilde{Y})\widetilde{X}^T\approx{W}^T \right]\approx{W} \tag{2.46}$$

Differentiating $\approx{W}\widetilde{W} = I$, we get

$$\widetilde{\widetilde{W}}^{/}\widetilde{W}+\widetilde{\widetilde{W}}\widetilde{W}^{/}=0 \tag{2.47}$$

rearranging terms

$$\widetilde{\widetilde{W}}^{/}=-\widetilde{\widetilde{W}}\widetilde{W}^{/}\widetilde{W}^{-1}=-\widetilde{W}^{-1}\widetilde{W}^{/}\widetilde{W}^{-1} \tag{2.48}$$

Using 1st order Euler approximation for (2.48), we have

$$\Delta\widetilde{\widetilde{W}}=-\widetilde{W}^{-1}(\Delta\widetilde{W})\widetilde{W}^{-1} \tag{2.49}$$

Also $\widetilde{Y}^{T}=\widetilde{X}^{T}\widetilde{\widetilde{W}}^{T}$ (2.50)

Using (2.49) and (2.50), the update law in (2.46) becomes

$$-\widetilde{W}^{-1}(\Delta\widetilde{W})\widetilde{W}^{-1}=\eta\left[I_{M+N}-\varphi(\widetilde{Y})\widetilde{Y}^{T}\right]\widetilde{\widetilde{W}} \tag{2.51}$$

or

$$-\widetilde{W}^{-1}(\Delta\widetilde{W})=\eta\left[I_{M+N}-\varphi(\widetilde{Y})\widetilde{Y}^{T}\right] \qquad \because \widetilde{\widetilde{W}}\widetilde{W}=I \tag{2.52}$$

Arranging terms

$$\begin{aligned}\Delta\widetilde{W}&=-\eta\,\widetilde{W}\left[I_{M+N}-\varphi(\widetilde{Y})\widetilde{Y}^{T}\right]\\&=\eta\,\widetilde{W}\left[\varphi(\widetilde{Y})\widetilde{Y}^{T}-I_{M+N}\right]\end{aligned} \tag{2.53}$$

Inserting values for $\widetilde{W}$ and $\widetilde{Y}$, we have

$$\Delta\begin{bmatrix}I_N+D & C\\ 0 & I_M\end{bmatrix}=\eta\begin{bmatrix}I_N+D & C\\ 0 & I_M\end{bmatrix}\times \begin{bmatrix}\varphi(y)y^T-I_N & \varphi(y)X^T\\ \varphi(X)y^T & \varphi(X)X^T-I_M\end{bmatrix} \tag{2.54}$$

Therefore the update laws for the matrices C and D are

$$\Delta(I_N+D)=\Delta D=\eta\left[(I_N+D)(\varphi(y)y^T-I_N)+C\varphi(X)y^T\right] \tag{2.55}$$

$$\Delta C=\eta[(I_N+D)\varphi(y)X^T+C(\varphi(X)X^T-I_M)] \tag{2.56}$$

Looking at the update of remaining terms in $\widetilde{W}$, we see that

$$0_{M\times N}=\varphi(X)_{M\times 1}y^T{}_{1\times N} \tag{2.57}$$

$$\Delta I_M=0=\varphi(X)X^T-I_M\text{, or} \tag{2.58}$$

$$\varphi(X)X^T=I_M \tag{2.59}$$

Notice similarity of (2.57) and (2.59) to (2.31) and (2.34). These relations form *supplementary conditions* for the convergence of the proposed feedback algorithm. Enforcing these conditions in update laws (2.55) and (2.56) not only simplifies them computationally, but also improves the convergence properties of the proposed algorithm. The final update law for the feedback structure is given by

$$\Delta D=\eta\left[(I_N+D)(\varphi(y)y^T-I_N)\right] \tag{2.60}$$

$$\Delta C=\eta[(I_N+D)\varphi(y)X^T] \tag{2.61}$$

3. SIMULATION EXAMPLES

In this paper we are presenting simulation results for an IIR mixing environment model. Both the mixing and demixing systems are represented by MIMO canonical state space form. The simulations are done for a variety of non-gaussian data distributions, and the results compared.

The environment model used in these simulations is given by

$$\sum_{j=0}^{m-1}A_i m(k-i)=\sum_{i=0}^{n-1}B_i s(k-i)+v(k) \tag{3.1}$$

where

$$\begin{aligned}A_0&=\begin{bmatrix}1.0 & 1.0\\ 1.0 & 1.0\end{bmatrix}, A_1=\begin{bmatrix}0.5 & 0.8\\ 0.8 & -0.3\end{bmatrix}, A_2=\begin{bmatrix}0.06 & 0.4\\ 0.16 & 0.1\end{bmatrix}\\ B_0&=\begin{bmatrix}1.0 & 0.6\\ 0.5 & 1.0\end{bmatrix}, B_1=\begin{bmatrix}-0.5 & 0.5\\ -0.3 & 0.2\end{bmatrix}, B_2=\begin{bmatrix}-0.125 & 0.06\\ -0.1 & 0.04\end{bmatrix}\end{aligned} \tag{3.2}$$

$v(k)$ - Additive Gaussian noise

The theoretical inverse of this IIR mixing environment [1,6] will also be an IIR filter of at least dimension 8.

The demixing network is setup to be in proposed feedforward and feedback state space structures. In both the cases, the state propagation matrices *A* and *B* are kept fixed while matrices output matrices *C* and *D* are adaptively updated. The convergence properties are discussed below. The benchmark employed for comparison is the multi-channel intersymbol interference (MISI), which is defined as

$$MISI_k=\sum_{i=1}^{N}\frac{\left|\sum_j\sum_p\left|G_{pij}\right|-\max_{p,j}\left|G_{pij}\right|\right|}{\max_{p,j}\left|G_{pij}\right|}+\sum_{j=1}^{N}\frac{\left|\sum_i\sum_p\left|G_{pij}\right|-\max_{p,i}\left|G_{pij}\right|\right|}{\max_{p,i}\left|G_{pij}\right|} \tag{3.3}$$

where *G(z)* – Global Transfer Function, given by

$$G(z)=H(z)*\bar{H}(z) \tag{3.4}$$

and

$\bar{H}(z)=[A_e,B_e,C_e,D_e]$ –Transfer Function of Environment

$H(z)=[A,B,C,D]$ –Transfer Function of Network,

We have used an element-wise acting nonlinearity which depends on the batch kurtosis of the output of the state space network as it converges. The algorithm is able to converge efficiently for both super-gaussian and sub-gaussian source densities using this function. The observed convergence rate is comparable to the cases where an implicit non-linearity depending on the original source distributions is employed. The nonlinearity, however, fails to give fast convergence results for sources with densities close to Gaussian. $\varphi(y)=y+\kappa_4(y)\tanh(\beta y)$ (3.5)

where

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

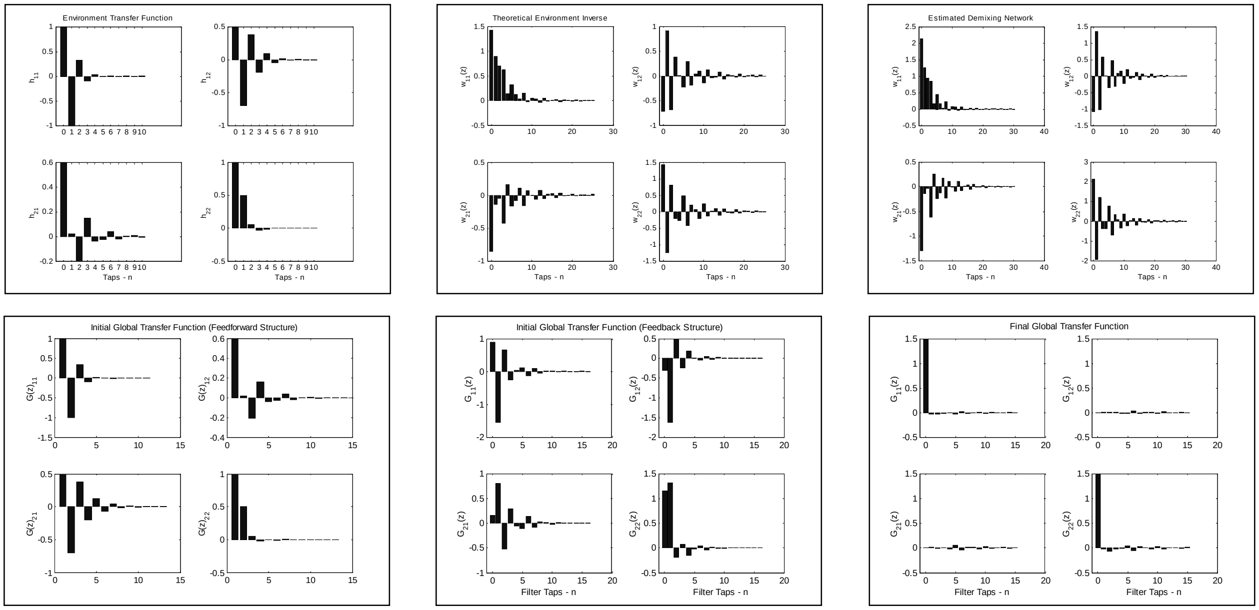

Figure 4: Representative Simulation Results using the proposed algorithms

For both simulations, the network is assumed to be in MIMO controller form. The matrices A and B have the structure

$$A = \begin{bmatrix} -Q_1 & -Q_2 & \cdots & -Q_{M-1} & -Q_M \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix}, \; B = \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (3.6)$$

where the state space network/filter can be represented by

$$H(z) = [P(z)][Q(z)]^{-1} \qquad (3.7)$$

and

$$P(z) = \sum_{i=0}^{n} P_i z^{-i}, Q(z) = \sum_{i=0}^{n} Q_i z^{-i} \text{ with } Q_0 = I_N \qquad (3.8)$$

3.1 Simulation for Feedforward Case

The matrix C is initialized with small random normally distributed elements with a variance of 0.1. The matrix D is initialized to be non-singular, dominantly diagonal (see Fig. 4). The simulation results are shown in Fig. 5 after 50,000 instantaneous updates of both the matrices. For sub-gaussian distribution, the learning rate is decreased exponentially with increasing iterations to achieve both fast convergence and good steady state value of the benchmark. For the gamma-distributed sources, a relatively smaller learning rate was required to achieve convergence.

3.2 Simulation for Feedback Case

For the feedback case, the matrix C is initialized with small random normally distributed elements with a variance of 0.1. The matrix D is initialized to be non-singular and dominantly anti-diagonal (see initial global solution in Fig. 4). The simulation results presented below in Fig. 6 show results after 60,000 instantaneous updates of both the matrices. These simulations also employ exponentially decaying learning rates, where relatively smaller learning rate were required for convergence in case of gamma distributed sources.

4. CONCLUSIONS

This paper presents the simulated results for an IIR mixing/filtering environment using our proposed feedforward and feedback algorithms for BSR. We setup the problem in a state-space framework and used an adaptive non-linearity dependent purely on the statistics of the network output. We showed that the algorithms are able to converge for a variety of source distributions. Using the feedforward approach the update of output matrices C and D is equivalent to learning blindly about the zeros of the demixing network, similarly in the feedback case, this update accounts for tuning the poles of the demixing network. Therefore, a combination of the two proposed approaches will be capable to do both. This is currently being investigated and the results will be supplied in future publications.

Relatively low-peaked gamma distributed sources, resembling speech-like statistics, were chosen in above simulations. The algorithm required relatively smaller learning rates for convergence in this case as compared to uniform and bimodal distributions, which account more for communication problems.

5. REFERENCES

[1] Salam F. M. "An adaptive blind separation of independent signals." *Proc. of Int'l Symp. on Circuits and Systems*, 1: pp. 431-1434, May 1993.

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

[3] Salam F. M. and Erten G.; “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

[4] Lee T. W., Girolami M., Bell A. J. and Sejnowski T.: “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

[5] Salam F. M., Erten G. and Waheed K.: “Blind Source Recovery: Algorithms for Static and Dynamic Environments”; *Int’l Joint Conf. on Neural Networks*, Washington, D.C., July 2001. (accepted)

[6] Waheed K. and Salam F. M.: “Blind Source Recovery: Some Implementation and Performance Issues”; *44th IEEE Midwest Symp. On Circuits and Systems*; August 14-17 2001, Dayton Ohio. (accepted)

[7] Waheed K. and Salam F.M.: “Blind Source Recovery: A State Space Framwork”; *Technical Memo No. ECE-CSANN’001, Dept of ECE, Michigan State University*, June 2001.

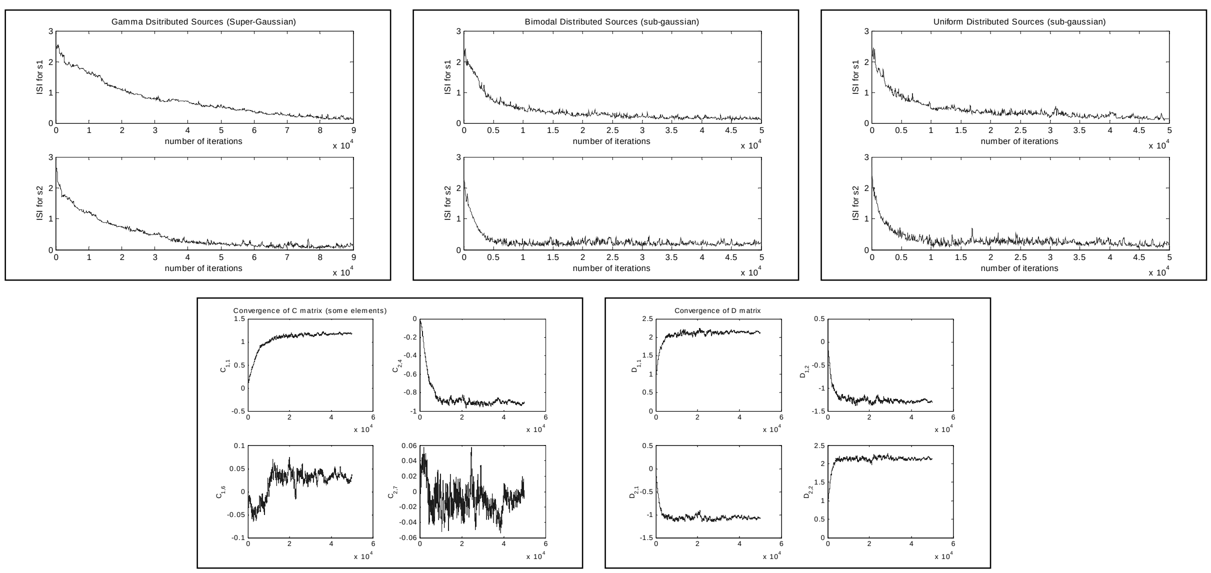

Figure 5: Results for Feedforward Formulation

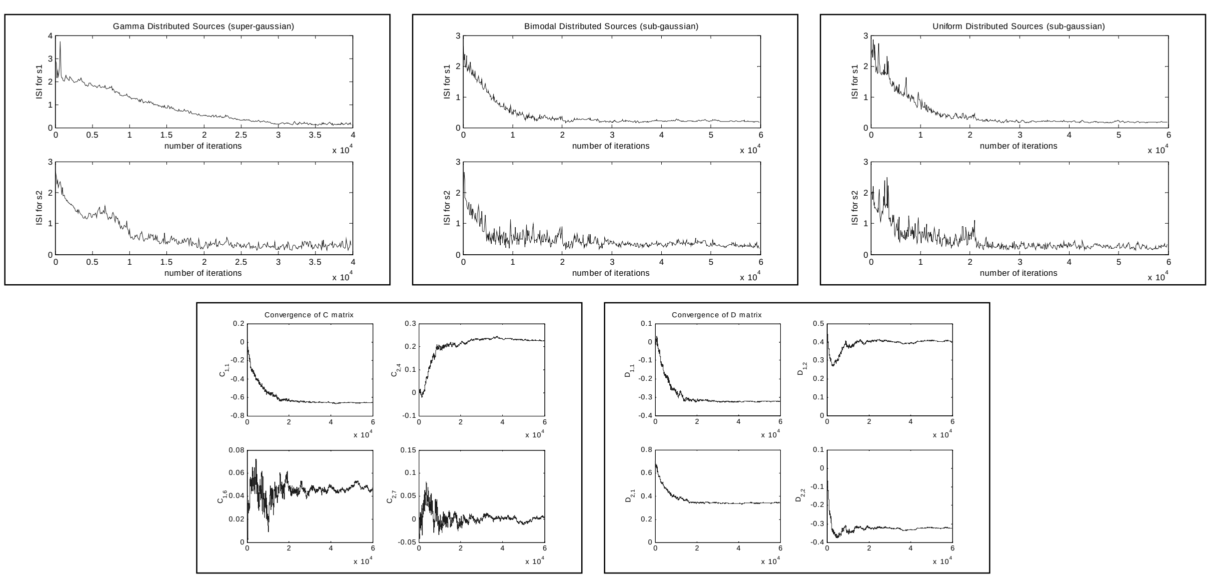

Figure 6: Results for Feedback Formulation