Blind Signal Separation and Recovery in Dynamic Environments

Fathi M. Salam
Circuits, Systems, and Artificial Neural Nets Lab
Michigan State University
East Lansing, MI 48824
Email: salam@ee.msu.edu

Gamze Erten
Innovative Computing Technologies, Inc.
2157 University Park Drive
Okemos, MI 48864
Email: erten@ic-tech.com

Abstract

This work bridges the gap between activities motivated from statistical signal processing, neuromorphic systems, and microelectronic implementation techniques for blind separation and recovery of mixed signals. The composition adopts both discrete-time and continuous-time formulations with a view towards implementations in the digital as well as the analog domains of microelectronic circuits. This paper focuses on the development and formulation of dynamic architectures with adaptive update laws for multi-source blind signal separation/recovery.

1. Introduction

The theoretical results and formulations address the blind separation and recovery of signals in **dynamic** environments [1, 2, 3]. We consider state space dynamic models to represent the mixing environment and consequently the adaptive network used to perform the signal separation and recovery. We employ dynamic models which are easily, and directly, adapted to discrete as well as continuous time channels. The presented environment model and the adaptive network allow for the case when the mixing environment includes (state) feedback and memory. The feedback of the state/output corresponds to Infinite Impulse Response (IIR) filtering in the discrete-time case.

The work begins with performance functions as identified in the literature [4, 5]. The emphasis of our method is in developing the network architecture, and the improved convergent algorithms, with a view towards efficient implementations. An improved approximation of the (nonlinear) mutual information/entropy function is used in order to ensure whitening and also to eliminate the assumption of output unit covariance. The improved expansion produces an odd polynomial in the network outputs which includes a linear term, as well as higher order terms-- all absent from the expansion in [6]. It should be noted however, that the work reported in [6], as well as in [7] have addressed only the static case where the mixing environment is represented by a constant matrix. In [7], a formulation for an FIR filter was also converted into a static matrix mixing problem.

2. Problem Definition

We present a framework that addresses the blind signal separation and recovery (or de-convolution) in dynamic environments. The original work was motivated by the work of Herault and Jutten [8] and Comon [5]. Most of the recent results have focused primary on establishing analytical foundation of the results reported in [4, 5]. Several researchers have used a host of analytical tools that include applied mathematics, statistical signal processing, system theory, dynamical systems and neural networks. The challenge still exists in generalizing the environment to more general dynamic systems.

This paper focuses on extending the environment to include more realistic models beyond a constant matrix, and develops successful update laws. A crucial first step is to include dynamic linear systems of the *state space* which are more general than FIR filters and transfer functions due to the inclusion of feedback and variations in initial conditions. Moreover, these models lend themselves to direct extension to nonlinear models. Another motivation of this work is to enable eventual implementation in analog or mixed mode micro-electronics [9].

The summary gives a brief summary of the results developed. The formulation addresses the feedback *dynamic* structures, where the environment is represented by a suitable realization of a dynamic linear system.

3. Dynamic Architectures

Dynamic models encompass and describe more realistic environments. We have proposed feedforward and feedback architectures of the state space approach [1-3]. Throughout this paper, we shall refer to the mathematical model for signal mixing as the *mixing environment*, while we refer to the mathematical model for the signal recovery as *the (adaptive) network.*

The Feedforward Structure:

Let the n-dimensional source signal vector be s, and the m-dimensional measurement vector be M. We describe the mixing environment by the Linear Time-Invariant (LTI) state space:

$$\dot{\overline{X}} = \overline{A}\,\overline{X} + \overline{B}\,s$$
$$M = \overline{C}\,\overline{X} + \overline{D}\,s$$

The parameter matrices $\overline{A}$, $\overline{B}$, $\overline{C}$ and $\overline{D}$ are of compatible dimensions. This formulation encompasses both continuous-time and discrete-time dynamics The dot on the state $\overline{X}$ means derivative for continuous-time dynamics, it however means "advance" for discrete-time dynamics. The mixing environment is assumed to be (asymptotically) stable, i.e., the matrix $\overline{A}$ has its eigenvalues in the left half complex plane. The (adaptive) network is proposed to be of the form

$$\dot{X} = A\,X + B\,M$$
$$y = C\,X + D\,M,$$

where y is the n-dimensional output, X is the internal state, and the parameter matrices are of compatible dimensions. For simplicity, let us assume that X has the same dimensions as $\overline{X}$. Figure (1) depicts the feedforward form of this framework.

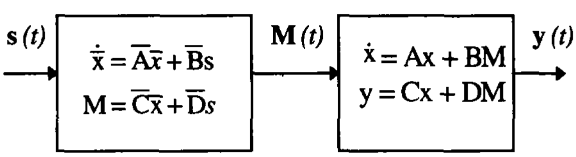

Figure 1. Feedforward Dynamic Structure

The first question is the following: Does there exist parameter matrices A, B, C, and D which would recover the original signals? The answer now follows.

Existence of solutions to the recovery problem:

We state that the (adaptive) dynamic network would be able to counter act the mixing environment, if the network parameters are set at (or attain via an adaptive scheme) the following values:

$$A = A^* = T\,(\overline{A} - \overline{B}\,[D]\,\overline{C})\,T^{-1}$$

$$B = B^* = T\,\overline{B}\,[D]$$

$$C = C^* = -\,[D]\,\overline{C}\,T^{-1}$$

$$D = D^* = [D]$$

where [D] equals

- D^{-1} : the inverse of D, if m = n,
- $(\overline{D^T D})^{-1} D^{-T}$; a pseudo-inverse, if m>n, and
- $D^{-T}(\overline{DD^T})^{-1}$: a pseudo-inverse if m < n.

The matrices A*, B*, and C* can take on a family of values due to the *nonsingular* state-equivalent transformation T. We shall use T to render the network architecture "canonical" or simple from a realization view point. This formulation in effect generalizes the formulations in the literature, which are limited to FIR filters, predominantly for 2-dimensional sources and two measurements, into general n-dimensional sources, and m-dimensional measurements. Note that, this modeling includes the FIR filtering models, and extends to IIR filtering if A is nonzero.

While this feedforward form for the adaptive network is viable, we note a limitation for its applicability, namely, that the parameters of the mixing environment have to be such that the matrix A* is (asymptotically) stable. That is, for a stable mixing environment, the composite matrix of the adaptive network

$$A^* = \overline{A} - \overline{B}\,[D]\,\overline{C}$$

must be (asymptotically) stable, i.e., has its eigenvalues in the left half complex plane. It is apparent that this requirement places a limiting condition on the allowable mixing environments which may exclude certain class of applications!

The Feedback structures:

A more effective architecture is the so-called (output) feedback network architecture, see Figure 2. This architecture leads to less restrictive conditions on the network parameters. Also, because of feedback, it inherits several known attractive properties of feedback systems including, robustness to errors and disturbances, stability, and increased bandwidth. These gains will become apparent from the following equations

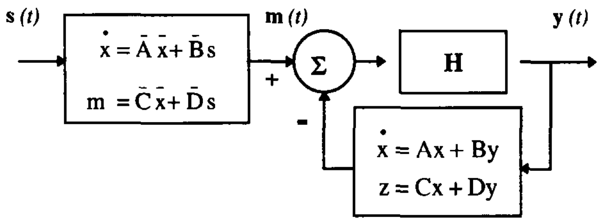

Figure 2. Feedback Dynamic Structure

Existence of solutions to the recovery problem

If y is to converge to a solution proportional (via a permutation matrix P) to s, namely, y= Ps, then, the following parameter matrices of the (adaptive) network will constitute a solution that recovers the original signals:

$$A = A^* = T\,\bar{A}\,T^{-1}$$

$$B = B^* = T\bar{B}P^{-1}$$

$$C = C^* = \bar{C}T^{-1}$$

$$D = D^* = \bar{D}P^{-1} - H$$

In addition to the expected desired properties of having feedback in the architecture of the network, we also achieve simplicity of solutions to the separation/recovery of signals. In this case, it is noted that if the mixing environment is (asymptotically) stable then so would be the solution to the adaptive network. In this case, the architecture is not introducing additional constraints on the network. Note that H in the forward path of the network may in general represent a matrix in the simplest case, or a transfer function of a dynamic model. Furthermore, in the event that m = n, H may be chosen to be the identity matrix.

The elements of the procedure and its advantages are now apparent. Further generalizations of the procedures for developing the architectures can also account for non-minimum phase mixing environments. These steps are straightforward application of the above procedure and hence will not be elaborated upon.

An important generalization is to include nonlinearity as part of the architecture-- explicitly. One model is to include nonlinearity as a static mapping of the measurement variable M(t). In this event, the adaptive network needs to include a compensating nonlinearity at its input stage. Thus, the input must include an "inverse-type" nonlinearity to counter act the measurement prior to further processing. This type of mixing environment is encountered in wireless applications that include satellite platforms.

With the dynamic architecture defined in the proper way, ensuring that a solution to the blind signal separation does exist, we now move to the next step of defining the proper adaptive procedure/algorithm which would enable the network to converge to one of its possible solutions. In this way, we are ensuring that if convergence takes place, the solution, the adaptive network would converge to, does exist and it is stable. Consequently, after convergence, the network will retain the variable for signal processing/recovery.

4. Performance Measure/Function

The mutual information of a random vector **y** is a measure of independence among its components and is defined as [4, 5]:

$$I(\mathbf{y}) = \int f_{\mathbf{y}}(\mathbf{u}) \ln \frac{f_{\mathbf{y}}(\mathbf{u})}{\prod_i f_{y_i}(u_i)} d\mathbf{u} \qquad (1)$$

where $f_y(\mathbf{y})$ is the probability density function (pdf) of the random vector **y**. The functional $I(\mathbf{y})$ is always non-negative and is zero if and only if the components of the random vector **y** are statistically independent. This important measure defines the *degree of dependence* among the components of the signal vector. Therefore, it represents an appropriate functional for characterizing (the degree of) statistical independence. $I(\mathbf{y})$ can be expressed in terms of the entropy

$$I(\mathbf{y}) = -H(\mathbf{y}) + \sum_i H(y_i) \qquad (2)$$

where $H(\mathbf{y}) := -E[\ln f_{\mathbf{y}}]$, is the **entropy** of **y**, and E[.] denotes the expected value.

5. Update Law for The Feedforward architecture

The update law is now developed for dynamic environments to recover the original signals following the procedures in [2, 3]. The environment here is modeled as a linear dynamical system. Consequently, the network will also be modeled as a linear dynamical system.

The Feedforward Case:

The network is a feedforward dynamical system as in Figure 1. In this case, one defines the performance index

$$J(\mathbf{x},\mathbf{w}) = \int_0^T \mathcal{L}\,(t,\mathbf{x},\dot{\mathbf{x}},\lambda,\mathbf{w})dt \qquad (3)$$

where $\mathcal{L}$ is the Lagrangian and is defined as

$$\mathcal{L}(t,\mathbf{x},\dot{\mathbf{x}},\lambda,\mathbf{w}) = \phi\,(t,\mathbf{x},w) + \lambda^{T}(\dot{\mathbf{x}} - A\mathbf{x} - B\mathbf{e}) \qquad (4)$$

where $\lambda(t)$ is the adjoint state equation defined by

$$\dot{\lambda} = -A^T\lambda + \frac{\partial \phi}{\partial \mathbf{x}} \qquad (5)$$

The functional ϕ may represent a scaled version of our measure of dependence I (y), **w** is a vector constructed of the rows of the parameter matrices C and D. Note that a canonical realization may be used so that B is constant. The matrix A, in the canonical representation, may have only N-parameters, where N is the dimension of the state vector X. The parameters, A, C, and D, represented generically by $\mathbf{w}p$, will be updated using the general gradient descent form:

$$\dot{\mathbf{w}p} = -\eta\frac{\partial \mathcal{L}}{\partial\, \mathbf{w}p} \qquad (6)$$

Therefore, using the performance index defined in Equation (2) , the matrices C and D are updated according to

$$\dot{D} = \eta(I - f_a(\mathbf{y})\mathbf{y}^T)D \qquad (7)$$

$$\dot{C} = \gamma(I - f_a(\mathbf{y})\mathbf{x}^T)C \qquad (8)$$

where $f_a(.)$ is given by a variety of nonlinear *expansive* odd-functions which include hyperbolic sine, and the inverse of a sigmiodal function. In the specific computation/approximation performed in [2, 3], the function is given as

$$f_a(y) = \frac{71}{12}y^{15} - \frac{355}{12}y^{13} + \frac{190}{3}y^{11} - \frac{4033}{24}y^9 + \frac{941}{3}y^7 + \frac{47}{8}y^5 + y^3 + y \qquad (9)$$

The essential features in using equation (9) are summarized as follows: (1) it is analytically derived and justified, (2) it includes a linear term in y and thus enables the performance of second order statistics necessary for signal whitening, (3) it contains higher order terms which emanate from the 4th order cumulant statistics in the output signal y, and (4) it does not make the assumption that the output signal has unity covariance. To date, to out knowledge, the function of equation (9) represents the only function used in the literature with the above characteristics. This function, therefore, exceeds the limitations of the other analytically derived function reported in [6].

Computer simulations confirm that the algorithm converges if the function defined in (9) is used. Example of computer simulations were reported in [2, 3]. A variety of extensive simulations are included in Gharbi and Salam [3].

The Feedback Case

The (output) feedback architecture of Figure 2 may be simplified in realization with the following (canonical) state-space representation:

The environment:

$$\dot{\overline{X}}_i = \overline{A}_i\,\overline{X}_i + \overline{B}_i\,\overline{S}, \quad 1 \le i \le L$$

$$M = \sum_{i=1}^{L}\overline{C}_i\,\overline{X}_i + \overline{D}\,\overline{S}$$

The network:

$$\dot{X}_i = A_i X_i + B_i\, y, \quad 1 \le i \le L$$

$$Z = \sum_{i=1}^{L} C_i X_i + D\,y$$

$$y = M - Z$$

where each X_i represents a state vector of the environment of the same dimension as the source signals, and each X_i represents a state of the network of the same dimension as the output signal. For simplicity, we assumed the same number, L, of the state vectors in both environment and network.

Now, using the performance index defined in Equation (2) , the matrices C_i and D are updated according to

$$\dot{D} = \eta D(-I + f_a(\mathbf{y})\mathbf{y}^T) \qquad (10)$$

$$\dot{C}_i = \gamma C_i(-I + f_a(\mathbf{y})\mathbf{x}^T) \qquad (11)$$

A simpler update law which was verified to work in certain cases may be satisfactory in special applications:

$$\dot{D} = \eta f_a(\mathbf{y})\mathbf{y}^T \qquad (12)$$

$$\dot{C}_i = \gamma f_a(\mathbf{y})\mathbf{x}_i^T \qquad (13)$$

Computer simulations reported in [2, 3] were also performed to demonstrate the performance of equations (12-13).

It should be clear that the states may, in the simple FIR filtering, represent simple delays of the sources, while the states in the network represent delays in the fed back output signals. However, this view is a simple consideration of the delays of the signal that occur in real physical applications. The framework, therefore, is more general since it may consider arbitrary delays including those of IIR filtering and continuous-time physical effects.

6. Implementation of the architectures and update laws

In [9], a direct hardware implementation of a practical extension of the HJ network to a first-order dynamic network in reported with experimental results. Direct implementations represent an avenue of effective implementation of the architectures and algorithms for the fastest execution of the recovery network.

Another paradigm includes DSP architectures. For a DSP based emulation of the signal separation algorithm families discussed here, it will be up to the tradeoffs in a particular application to identify the best processor architecture and numerical representations, e.g., floating or fixed point. To achieve a highly integrated solution (e.g., one chip) will require embedding a DSP core either from a pre-designed device or designed from standard silicon cell libraries.

The compiler front-end to the DSP assembler and linker forms a direct bridge from a high level language coded algorithm simulation environment to DSP emulation. In addition, a similar direct link exists between many computing environments and the DSP emulation environments, for example, C/C++ library and compilers for various processors.

Programmable logic can be an integral part of the related development process. A programmable DSP core (a DSP processor that is designed for integration into a custom chip) can be integrated with custom logic to differentiate a system and reduce system cost, space, and power consumption.

7. Conclusion

Proposed dynamic architectures, which ensure existence of solutions are augmented with effective update laws for the problem of blind separation/recovery of independent sources. Computer simulations demonstrated successful performance of these algorithms. More effectively, the feedback architecture lends itself to stable, robust performing adaptive networks that enable attaining solutions for a broader class of mixing environments. A description of the DSP realizations developed by IC Tech, Inc. is also outlined.

8. References

[1] F. Salam, An adaptive network for blind separation of independent signals, Proc. Of IEEE Symposium on Circuits and Systems, Chicago, Vol. 1: pp. 431-434, May 1993.

[2] A. Gharbi and F. Salam, Algorithms for blind signal separation and recovery in static and dynamics environments, IEEE Symposium on Circuits and Systems, Hong Kong, pp. 713-716, June 1997.

[3] A. Gharbi and F. Salam, Algorithms blind separation of signals for the static and dynamic environments, Technical report, MSU/S/No. 104, January 1997.

[4] S.Kullback. Topics in statistical information theory. *Springer Verlag*, New York, 1987.

[5] P. Comon. Independent component analysis: A new concept ? *Signal Processing*, 36(3):287-314, 1994.

[6] S. Amari, A. Cichocki, and H. Yang. A new learning algorithm for blind signal separation. *MIT Press* (in press), 1996.

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

[8] A. Bell and T. Sejnowski. An in formation maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129-1159, 1995.

[9] A. Gharbi and F. Salam, Implementation and Experimental results of a chip for the separation of mixed signals, Journal of Circuits, Systems, and Computers, Vol. 6(2), pp. 115-139, April 1996.