BLIND SEPARATION OF LINEAR CONVOLUTIVE MIXTURES THROUGH PARALLEL STOCHASTIC OPTIMIZATION

Marc Cohen and Gert Cauwenberghs

Department of Electrical and Computer Engineering
The Johns Hopkins University, Baltimore, MD 21218-2686
E-mail: {marc,gert}@bach.ece.jhu.edu

ABSTRACT

We apply stochastic parallel optimization techniques to on-line blind separation of linear convolutive mixtures of independent time-varying signals. The optimization performs stochastic gradient descent on a scalar measure of statistical independence observed directly on the outputs of the unmixing network, which contains a matrix of finite impulse response (FIR) filters. We derive on-line adaptation rules, and a scalable modular architecture with minimum memory requirements amenable to parallel VLSI implementation. The architecture implements a slight modification of the network adaptation rule, which omits symmetrical non-causal terms in the computation of the stochastic gradient. Simulations indicate near-perfect separation using both versions of the rule, with a minimum phase response resulting from the simplified version.

1. INTRODUCTION

Blind source separation, also known as independent component analysis (ICA), is a research topic of considerable interest because of its wide range of applications. The task is to recover a set of unknown independent signal sources which, propagating through an uncharacterized medium, are mixed when they reach a set of sensors. The mixing in the medium is most typically modeled as linear and instantaneous, while nonlinear convolutive mixing constitutes the most challenging case. In this paper we concentrate on linear convolutive mixing. Our analysis is carried out in the time domain in order to suggest scalable and parallel analog VLSI architectures for performing low power real-time signal separation.

Herault and Jutten [1] first proposed a learning rule for ICA in the case of linear instantaneous mixing. Vittoz and Arreguit [2], Cohen and Andreou [3, 4] and more recently Gharbi and Salam [5] designed analog VLSI chips which implement this algorithm. In recent years, interest in ICA has grown considerably. Of particular interest here, one class of ICA algorithms has been derived by several authors from different information-theoretic and statistical principles, Bell and Sejnowski [6], Cichocki and Unbehauen [7], Amari, Cichocki and Yang [8] to name but a few. In this context, solutions to the task of ICA for linear convolutive mixtures in the frequency-domain have been formulated by Bell and Sejnowski [6], Lambert and Bell [9] and Lee, Bell and Orglmeister [10].

The approach outlined here follows that of Cichocki and Unbehauen [7] for instantaneous linear mixtures, extended to linear convolutive mixtures by drawing upon results from Gorokhov and Loubaton [11].

This work was supported by NSF Career Award MIP-9702346. We thank Pamela Abshire for helpful discussions and carefully verifying the derivation of the learning rules.

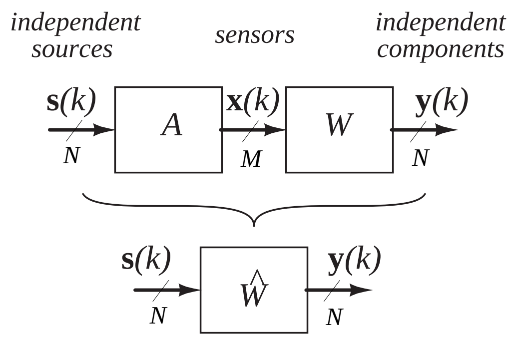

Figure 1: *Problem Statement*

2. PROBLEM STATEMENT

An unknown signal source vector $\mathbf{s}(k)$ passes through an unknown medium A before it is received by an array of sensors $\mathbf{x}(k)$. The task is to recover $\mathbf{s}(k)$ from $\mathbf{x}(k)$, the only assumption being that the components $s_i(k)$ are mutually independent. To this purpose the sensor inputs $\mathbf{x}(k)$ are transformed through W, yielding an output $\mathbf{y}(k)$ which ideally would be $\mathbf{s}(k)$, but which at best is a scaled, permuted, and possibly convolved version of $\mathbf{s}(k)$ in practice (see Figure 1). Without further assumptions on the sources, a solution exists only if the number of independent sensors M matches or exceeds the number of independent sources N.

In the linear case, A is modeled as a linear matrix operator, and a linear matrix operator W performs the unmixing

$$\mathbf{y} = W\,\mathbf{x} = \hat{W}\,\mathbf{s} \tag{1}$$

where the transformed operator $\hat{W} = W\,A$ refers the outputs to the sources. It is convenient to view ICA in the reference frame $\hat{W}$, independent of A. Ideally, $\hat{W} = P\,I$ where P is an arbitrary scaling and permutation operator and I is the identity operator. We start by reviewing the instantaneous case for which A and W are ordinary matrices.

3. LINEAR INSTANTANEOUS MIXTURES

Cichocki and Unbehauen [7] formulated ICA in terms of a scalar measure of output signal independence to be minimized,

$$\begin{aligned}\mathcal{F} &= \frac{1}{4}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\mathrm{E}\left[y_i(k)\,y_j(k)\right]-\lambda\,\delta_{ij}\right)^2 \\ &= \frac{1}{4}\left\|\mathrm{E}\left[\mathbf{y}(k)\,\mathbf{y}^T(k)\right]-\Lambda\right\|_F \end{aligned} \quad (2)$$

where $\| \, \|_F$ is the Frobenius norm of a matrix, $\mathrm{E}[\ ]$ the expectation operator, and $\Lambda = \lambda I$ where λ is a normalization constant. Combining (1) and (2) and noticing that $\mathrm{E}\left[\mathbf{y}(k)\mathbf{y}^T(k)\right] = \hat{W}\,\mathrm{E}\left[\mathbf{s}(k)\,\mathbf{s}^T(k)\right]\,\hat{W}^T = \hat{W}\,\hat{W}^T$, (2) simplifies to

$$\mathcal{F} = \frac{1}{4}\left\|\hat{W}\,\hat{W}^T-\Lambda\right\|_F \ . \quad (3)$$

Gradient descent of (3) in the reference frame produces an update

$$\Delta\hat{W} = -\mu\nabla_{\hat{W}}\,\mathcal{F} = \mu\left(\Lambda-\hat{W}\,\hat{W}^T\right)\,\hat{W} \quad (4)$$

which (assuming $\frac{dA}{dt}\approx 0$) transforms into an adaptation rule for the weights W

$$\Delta W = \mu\left(\Lambda-\mathrm{E}\left[\mathbf{y}(k)\,\mathbf{y}^T(k)\right]\right)\,W \quad (5)$$

where μ is the adaptation rate constant. A stochastic on-line weight adaptatation rule is obtained by removing the expectation operator in (5) [7]. Finally, antisymmetric nonlinear functions $f(\,)$ and $g(\,)$ are applied component-wise to $\mathbf{y}$ to ensure output signal independence beyond second-order statistics (removal of higher order cummulants) [1, 7], yielding the general update rule

$$\Delta W = \mu\left(\Lambda-f(\mathbf{y}(k))\,g(\mathbf{y}^T(k))\right)\,W \quad (6)$$

of which the ICA algorithms in [6, 8] and others are special cases.

4. LINEAR CONVOLUTIVE MIXTURES

We consider the case where the mixing operator A is linear and convolutive, characterized by a matrix of FIR filters

$$\mathbf{x}(k) = \sum_{p=0}^{K-1} A(p)\,\mathbf{s}(k-p) \quad (7)$$

where the coefficients $A_{ij}(p)$ denote the response of sensor i at time p to an impulse in source j at time 0.

Gorokhov and Loubaton [11] show that linear FIR convolutive mixtures of the type in (7) can, in principle, be perfectly unmixed and deconvolved by an FIR convolutive matrix operator W of appropriate filter length, provided that strictly $M > N$. The length required of each FIR filter in W to invert A perfectly is $L \geq N\,K-1$. Since A is essentially unknown, a lower bound on L needs to be estimated.

We state the optimization criterion for the convolutive case in terms of the Frobenius norm

$$\mathcal{F} = \frac{1}{4}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{l=-\infty}^{\infty}\left(\mathrm{E}\left[y_i(k)y_j^T(k-l)\right]-\lambda\,\delta_{ij}\delta_{l0}\right)^2, \quad (8)$$

which not only attempts to separate, but also to deconvolve (whiten) the outputs (*i.e.,* produce impulse autocorrelation). In what follows, we assume for simplicity of the derivation that the sources are white to start with,

$$\mathrm{E}\left[s_n(k-r)s_m(k-s)\right] = \delta_{nm}\delta_{rs}\,, \quad (9)$$

which transforms (8) into

$$\mathcal{F} = \frac{1}{4}\sum_{i,j,l}\left(\sum_{n,p}\hat{W}_{in}(p)\hat{W}_{jn}(p-l)-\lambda\delta_{ij}\delta_{l0}\right)^2. \quad (10)$$

As before, gradient descent of $\mathcal{F}$ in (10) and elimination of the common A operator on both sides yields an update rule

$$\Delta W(q) = \mu\left(\lambda W(q)-\sum_r \mathrm{E}[\mathbf{y}(k-r)\mathbf{y}^T(k-q)]W(r)\right) \quad (11)$$

Removing the expectation operator and applying the antisymmetric nonlinear functions $f(\,)$ and $g(\,)$ component-wise to $\mathbf{y}$ in (11) results in the following on-line weight update rule for convolutive ICA

$$\Delta W(q) = \mu\left(\lambda W(q)-f(\mathbf{y}(k))\,\mathbf{z}^T(k,q)\right) \quad (12)$$

where the vector $\mathbf{z}(k,q)$ is constructed as

$$\mathbf{z}(k,q) = \sum_{r=0}^{L-1} W^T(r)\,g(\mathbf{y}(k-(q-r))) \quad (13)$$

The convolutive ICA algorithm derived in [9] and demonstrated in [10] reduces to a special case of (12) transformed to the frequency domain.

5. SIMULATIONS

To simplify analysis, simulation results reported here are referenced to the sources, in terms of $\hat{W}$. The results apply to arbitrary operators A as long as $M > N$, A is full rank, and the length L of the unmixing filters in W warrants invertibility according to the above conditions.

We chose to simulate a system $\hat{W}$ with two source inputs and two outputs, $N = 2$ and impulse length $L = 6$. $s_1(k)$ and $s_2(k)$ are uniform white noise signals $\in [-1, 1]$. The weights $\hat{W}(q)$, $q = 0 \ldots L-1$, are initialized with uniform random weights $\in [-1, 1]$. For the functions f and g, we chose the identity map $f(\mathbf{y}(t)) \equiv \mathbf{y}(t)$ and the signum function $g(\mathbf{y}^T(t)) \equiv \mathrm{sgn}(\mathbf{y}^T(t))$ mainly because they are simple to implement. Other functions could be investigated such as the phase preserving functions of Cardoso *et al* [12].

Figure 2 show the trajectories of all $2 \times 2 \times 6$ weights in $\hat{W}$ over time, implementing (12) and (13). As expected for $\lambda = 1$, all but two weights converge to zero and the remaining two weights converge to ± 2. Figure 3 shows the impulse responses of the 2×2 filters each of length 6. It is clear that $y_1(k)$ corresponds to $s_2(k-3)$ and $y_2(k)$ to $s_1(k-1)$, which is one of many valid solutions to this unmixing/deconvolution task.

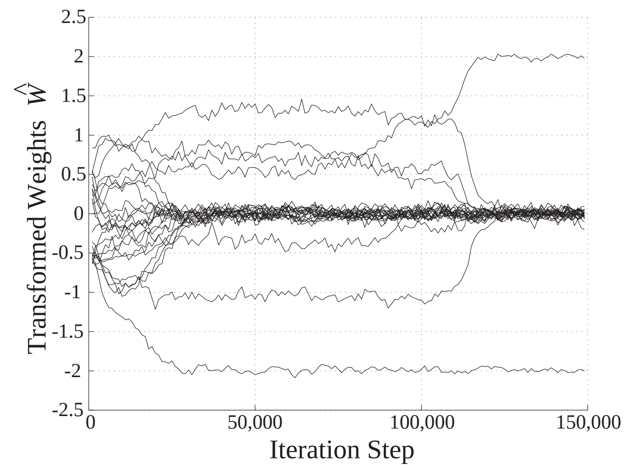

Figure 2: *Trajectory of the coefficients* $\hat{W}_{ij}(q)$ *over time* k *for the full update rule*

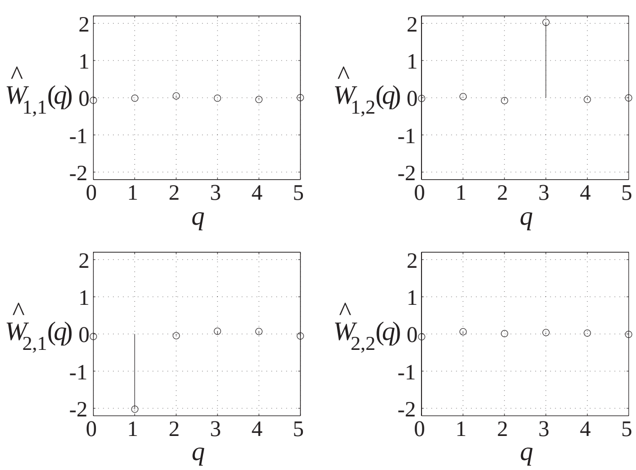

Figure 3: *Impulse response of each filter after convergence*

6. ARCHITECTURE

6.1. Triangularization of the outer-product updates

The form of the update rule in (12) is attractive for on-line parallel VLSI implementation, as it operates in real-time on the outer-product of two instantaneous vectors, for which elegant analog and mixed-mode VLSI adapation architectures exist [13]. The remaining difficulty lies in the construction of vector $\mathbf{z}(k,q)$, outside of the matrix of cells. One problem with the form of (13) for on-line implementation is that it is non-causal, incurring future contributions from elements $W(r)$ to $W(q)$ whenever $r > q$. This problem could be alleviated by shifting the time axis and feeding signals through delay lines, which adds to the implementation complexity. Rather, we propose to simplify the construction of $\mathbf{z}(k,q)$ in (13) by omitting all terms for which $r > q$,

$$\mathbf{z}'(k,q) = \sum_{r=0}^{q} W^T(r) g(\mathbf{y}(k-(q-r))) \,. \tag{14}$$

The effect of this simplification is an asymmetry in the outer-

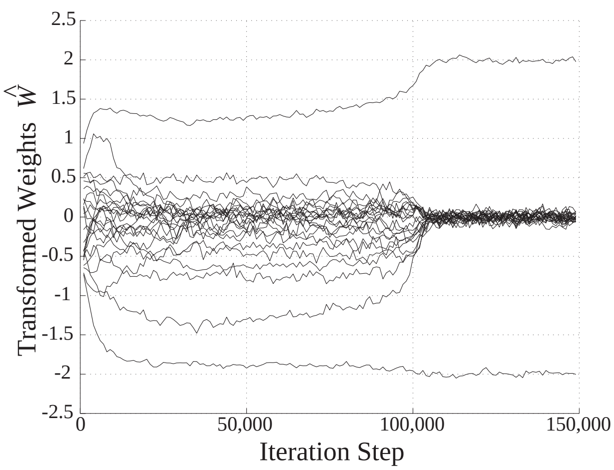

Figure 4: *Trajectory of the coefficients* $\hat{W}_{ij}(q)$ *over time* k *for the triangularized update rule*

product component of the update rule, reducing it to a triangular matrix in the time dimension (q,r). This asymmetry in time does not affect the solution reached at convergence, since all off-diagonal terms are zero when the gradient is zero. Simulations repeating the above experiments from identical initial conditions, shown in Figure 4, indicate that the convergence properties of the triangularized version of the update rule are qualitatively similar to that of the full version. Yet, the triangularized version seems to favor solutions with lower indices q for the non-zero weights in $\hat{W}(q)$, providing a minimum delay in the phase response of the filters, which clearly is a desirable property. The minimum phase response is a direct consequence of the asymmetry in q and r, and can be understood in terms of the correspondingly modified form of $\mathcal{F}$ in (8).

6.2. Scalable, modular parallel architecture

The resulting architecture for VLSI implementation is illustrated in Figure 5, in the simplified case for a matrix of $N = 2$ inputs and $M = 3$ outputs, and $L = 3$ taps per matrix element FIR filter.

Sensor inputs $x_j(k)$ are presented at the bottom of the system, where they are fed into delay lines to generate the delayed versions $x_j(k-q)$ projected across columns in the array. Outputs $y_i(k)$ are extracted along rows from the right. The components $z'_j(k,q)$ are generated on top and projected along columns. Each dashed box represents a unit cell performing the weight multiplication $W_{ij}(q)$ and the weight update $\Delta W_{ij}(q)$.

An enlarged view of the unit cell is shown in Figure 5(b). The delayed input (voltage) $x_j(k-q)$ is multiplied by the weight $W_{ij}(q)$, and accumulated along the output (current or charge) summing line y_i^{sum}. The accumulated output

$$y_i(k) = \sum_j \sum_q W_{ij}(q)\, x_j(k-q) \tag{15}$$

is fed back along each row to generate the outer-product updates. To construct $z'_j(k,q)$, each cell also multiplies $g(y_i(k))$ with $W_{ij}(q)$ and accumulates the result onto the $\gamma_j^{sum}(k,q)$ (current or charge) summing line to construct

$$\gamma_j(k,q) = \sum_i W_{ij}(q)\, g(y_i(k)) \tag{16}$$

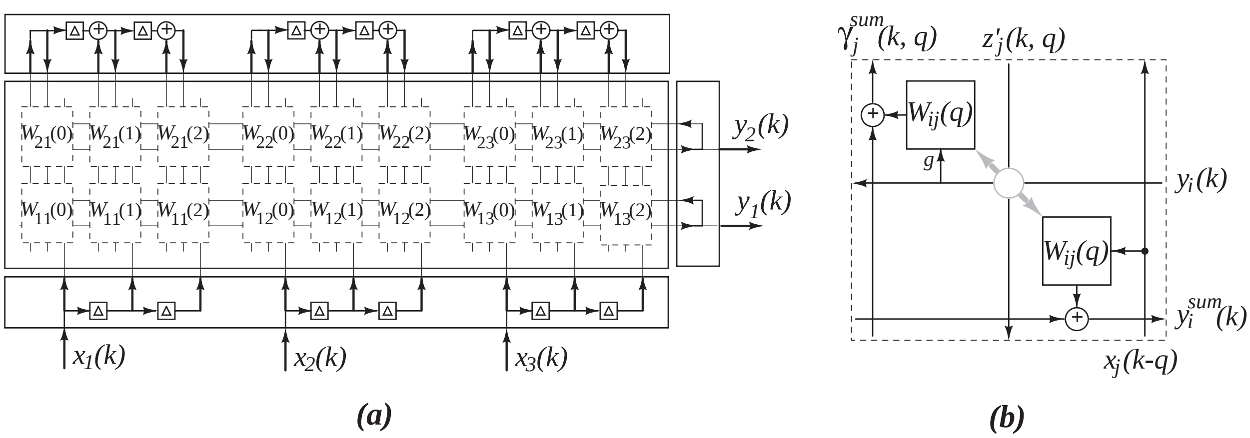

Figure 5: *Parallel architecture:* **(a)** *example system block diagram for* $N = 2$, $M = 3$, *and* $L = 3$, **(b)** *unit cell diagram.*

which is incrementally shifted and accumulated (along a bucket-brigade line [14]) on the top of the array to generate the $\mathbf{z}'(k, q)$ vector

$$z'_j(k, q) = \sum_{r=0}^{q} \gamma_j(k - (q - r), r) \qquad (17)$$

which is functionally equivalent to (14).

Finally, $z'_j(k, q)$ is projected down each column. Locally in the unit cell, it is multiplied by $f(y_i(k))$, and the result subtracted from $\lambda\, W_{ij}(q)$ and scaled by the learning rate-constant μ to produce a weight update $\Delta\, W_{ij}(q)$ consistent with the triangularized version of (12).

Other arrangements exist to implement the adaptive rule in a parallel architecture embedded with the matrix of coefficients, possibly avoiding the triangularization of the updates. The advantage of using this architecture is that all computations within the array are instantaneous, which avoids excessive wiring or memory storage. All memory-intensive operations are performed at the periphery of the array, and the computational complexity of the unit cell is minimized.

7. CONCLUSIONS

We derived an on-line learning rule for ICA in the case of linear convolutive mixtures of independent signal sources. The performance criterion in the gradient descent optimization is based directly on a scalar measure of statistical independence defined directly on the outputs of the unmixing network. This formulation is independent of the specifics of the network, and can be extended to nonlinearly convoluted mixtures. In the case studied, both mixing and unmixing transformations are represented by a matrix of filters with finite impulse response. With minor modifications, the derived adapation rule for the coefficient updates was cast in a scalable parallel architecture embedded in the matrix of unmixing filters. All operations are forward time, and no memory is required internally in the array of cells. Simulations confirmed the validity of the simplified rule, yielding solutions with minimum phase delay response.

8. REFERENCES

[1] Herault, J. and Jutten, J., "Space or time adaptive signal processing by neural network models," in Denker, J. S. (ed), *Neural Networks for Computing: AIP conference proceedings 151*, American Institute of Physics, New York, 1986.

[2] Vittoz, E. and Arreguit, X., "CMOS Integration of Herault-Jutten cells for separation of sources," *Proc. Workshop on Analog VLSI and Neural Systems*, 1989.

[3] Cohen, M. and Andreou, A., "Current-mode subthreshold MOS implementation of Herault-Jutten autoadaptive network," *IEEE J. Solid State Circuits*, **27**, 5, pp 714 – 727, 1992.

[4] Cohen, M. and Andreou, A., "Analog CMOS Integration and Experimentation with an Autoadaptive Independent Component Analyzer." *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, **42**, 2, pp. 65 – 77, 1995.

[5] Gharbi, A. and Salam, F., "Implementation and Experimantal results of a chip for the separation of mixed and filtered signals," *Journal of Circuits, Systems and Computers*, **6**, 2, pp 115 – 138, 1996.

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

[7] Cichocki, A. and Unbehauen, R., "Robust neural networks with on-line learning for blind identification and blind separation of sources," *IEEE Trans. Circuits and Systems –I: Fundamental Theory and Applications*, **43**, 11, pp 894 – 906, 1996.

[8] Amari, S., Cichocki, A. and Yang, H., "A new learning algorithm for blind signal Separation," Neural Information Processing Systems, **8**, 1996.

[9] Lambert, R. and Bell, A., "Blind separation of multiple speakers in a multipath environment," *Proceedings of ICASSP*, Munich, 1997.

[10] Lee, T-W., Bell, A. and Orglmeister, R., "Blind source separation of real world signals," *Proceedings of International Conference on Neural Networks*,(Houston), 1997.

[11] Gorokhov, A. and Loubaton, P., "Subspace-based techniques for blind separation of convolutive mixtures with temporally correlated sources," *IEEE Trans. Circuits and Systems –I: Fundamental Theory and Applications*, **44**, 9, pp 813 – 820, 1997.

[12] Cardoso, J-F., Belouchrani, A. and Laheld, B., "A new composite criterion for adaptive and iterative blind source separation," *Proceedings of ICASSP*, 1994.

[13] Cauwenberghs, G., "Neuromorphic Learning VLSI Systems: A Survey," in T.S. Lande, Ed., *Neuromorphic Systems Engineering*, Norwell MA: Kluwer Academic, in print 1998.

[14] Edwards, R.T., Cauwenberghs, G. and Pineda, F.J., "A Mixed-Signal Correlator for Acoustic Transient Classification," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'97)*, Hong Kong, June 1997, vol. I, pp 621-624.