

# MIXED-SIGNAL REAL-TIME ADAPTIVE BLIND SOURCE SEPARATION

*Abdullah Celik, Milutin Stanacevic and Gert Cauwenberghs*

Department of Electrical and Computer Engineering  
Johns Hopkins University, Baltimore, MD 21218, USA  
{*acelik,miki,gert*}@*jhu.edu*

## ABSTRACT

A mixed-signal adaptive VLSI architecture for real-time blind separation of linear source mixtures is presented. The architecture is digitally reconfigurable and implements a general class of Independent Component Analysis (ICA) update rules in common outer-product form. In conjunction with *gradient flow*, a technique for converting time-delayed mixtures of traveling wave sources into equivalent linear instantaneous mixtures by observing spatial and temporal derivatives of the field over a miniature array, the ICA architecture allows to separate and localize multiple acoustic sources in the acoustic scene. Experimental results from VLSI implementation of the ICA architecture demonstrate 30 dB separation of two mixtures of two speech signals.

## 1. INTRODUCTION

The human auditory system performs remarkably well in segregating multiple streams of acoustic sources, even in significant presence of noise in the acoustic scene. To do so, it requires to resolve time delays and amplitude differences between sound waves entering both ears, and correlating and grouping these differences across the various source components [1]. Modern hearing aids utilize directional or multiple microphones to emulate some of the functionality of binaural sensing. However, they remain far from adequate in compensating the loss of functionality in impaired ears and their performance degrades significantly in the presence of multiple signal and noise sources in the acoustic scene. To be effective in resolving the signal of interest, both localization and separation of multiple acoustic sources are required.

In previous work we showed that the direction of wave propagation can be estimated obtained by differential spatial sensing of the field on sub-wavelength scale, using *gradient flow* [2, 3]. Mixed-signal VLSI implementation of the method [4] has demonstrated improved performance in terms of power dissipation and bearing resolution over conventional bearing estimation localizers.

---

This work was supported by ONR N00014-99-1-0612 and ONR/DARPA N00014-00-C-0315.

Besides its use in bearing estimation, gradient flow provides an efficient signal representation as a front-end for blind source separation. In the presence of multiple signal sources, gradient flow converts the problem of separating unknown delayed mixtures of independent signal sources, into a simpler problem of separating corresponding instantaneous mixtures of the time-differentiated signals [2]. This formulation is equivalent to the standard problem statement in Independent Component Analysis (ICA) [7]. Gradient flow and ICA combine to yield both separation and localization of multiple independent signal sources.

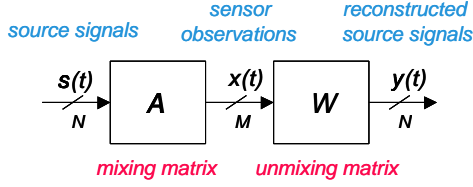
Various analog VLSI implementations of ICA exist in the literature, *e.g.*, [5, 6], and digital implementations using DSP are common practice in the field. We present a general mixed-signal parallel architecture, that can be configured for implementation of various ICA update rules in conjunction with gradient flow. Experimental results from micropower VLSI implementation are included to demonstrate the approach.

## 2. GRADIENT FLOW

Gradient flow [2] is a signal conditioning technique for separating and localizing traveling wave sources by relating spatial and temporal derivatives of the field observed over a miniature sensor array. Observation of first order spatial gradients of the field  $\xi_{10}(t)$  and  $\xi_{01}(t)$  in perpendicular directions in the plane yield linearly mixed observations of the time-differentiated source signals  $\dot{s}_\ell$  each scaled by propagation delays  $\tau_1^\ell$  and  $\tau_2^\ell$  along the gradient directions:

$$\begin{aligned}\xi_{10}(t) &\approx \sum_{\ell} \tau_1^\ell \dot{s}^\ell(t) \\ \xi_{01}(t) &\approx \sum_{\ell} \tau_2^\ell \dot{s}^\ell(t).\end{aligned}\quad (1)$$

In practice, the gradients  $\xi_{10}(t)$  and  $\xi_{01}(t)$  are estimated by finite differences of the field on the sensor grid [2]. Likewise, time differentiation of the observed spatial common mode of the field  $\xi_{00}(t)$  yields further linearly mixed obser-



**Fig. 1.** Standard ICA problem definition. Linear instantaneous mixing, and un-mixing, of independent sources.

variation of the time-differentiated source signals  $\dot{s}^\ell$ :

$$\dot{\xi}_{00}(t) \approx \sum_{\ell} \dot{s}^\ell(t). \quad (2)$$

For a single source  $s(t)$ , estimates of 3-D bearing angles are directly obtained by least squares adaptation of the coefficients  $\tau_1$  and  $\tau_2$  from (1) and (2) [3], efficiently implemented in a micropower mixed-signal VLSI architecture [3]. Multiple concurrent sources  $s^\ell(t)$  ( $\ell = 1, \dots, L$ ) give rise to linear mixtures in the gradient observations  $\xi_{10}$ ,  $\xi_{01}$  and  $\xi_{00}$  that can be blindly separated using independent component analysis (ICA) [7]. The identified source components from the gradient signals then yield the time-differentiated source signals  $\dot{s}^\ell(t)$ , and the un-mixing coefficients yield the corresponding 3-D direction cosines in terms of inter-temporal differences  $\tau_1^\ell$  and  $\tau_2^\ell$  [2]. Therefore, a compact system composed by integration of gradient flow and ICA, can be used to achieve multiple source tracking.

### 3. INDEPENDENT COMPONENT ANALYSIS

The task of blind source separation (BSS) is to separate and recover independent sources from mixed sensor observations, where both sources and mixing matrix are unknown. Independent component analysis (ICA) offers a principled mathematical approach to solving the BSS problem under the assumption of signal independence across sources [7]. Figure 1 shows the standard problem setting of ICA, in which observations are unknown linear mixtures of the unknown independent sources. ICA minimizes higher-order statistical dependencies between reconstructed signals to estimate both un-mixing matrix and independent signal sources.

#### 3.1. Algorithms

Several approaches exist to solving the linear ICA problem, and the present treatment is limited to static (non-convolutive) ICA algorithms which assume linear instantaneous mixing. Four of such well known algorithms are presented here briefly, in light of the implementation architecture presented in Section 4.1.

**Herault-Jutten (H-J) algorithm [8]** : the first formulation of BSS as inspired by biomimetic principles, the H-J update algorithm is based on a feedback network topology  $\mathbf{y} = -\mathbf{W}\mathbf{y} + \mathbf{x}$  with zero diagonal terms ( $w_{ii} \equiv 0, \forall i$ ). An independence criterion based on nonlinear correlation yields the on-line learning rule for the off-diagonal terms

$$\Delta w_{ij} = \mu f(y_i)g(y_j), \quad i \neq j \quad (3)$$

where  $f(\cdot)$  and  $g(\cdot)$  are appropriately chosen, odd-symmetric functions.

**Bell-Sejnowski (B-S) algorithm [9]** : The derivation of the learning rule (4) is based on the information maximization (*InfoMax*) principle, and maximizes the output entropy of a neural network with nonlinear outputs:

$$\Delta \mathbf{W} = \mu ([\mathbf{W}^T]^{-1} - f(\mathbf{y})\mathbf{x}^T) \quad (4)$$

**Natural Gradient (NG) algorithm [10]** : More robust and uniform convergence is obtained by using Amari's *natural gradient* of the *InfoMax* cost function, leading to the simple learning rule

$$\Delta \mathbf{W} = \mu [\mathbf{I} - f(\mathbf{y})\mathbf{y}^T] \mathbf{W} \quad (5)$$

The convergence of (5) implies  $E\{f_i(y_i)y_i\}=1$  as a constraint on reconstructed signals.

**Cichocki-Unbehauen (C-U) algorithm [11]** : To avoid numerical instability due to non-stationarity in the sources, the C-U algorithm introduces a non-holonomic constraint in the NG learning rule, fixing the diagonal of the unmixing matrix  $\mathbf{W}$ :

$$\Delta \mathbf{W} = \mu [\Lambda - f(\mathbf{y})g(\mathbf{y}^T)] \mathbf{W} \quad (6)$$

where  $\Lambda$  is a diagonal scaling matrix. Convergence of the C-U algorithm implies  $\Lambda_{ii} = E[f(y_i)g(y_i)]$ .

#### 3.2. General Outer-Product Formulation

Efficient implementation in parallel architecture requires a simple form of the update rule, that avoids excessive matrix multiplications and inversions. With simplification, the above ICA update algorithms can be cast in the common, unifying framework of the outer-product rule (3). To map the recurrent H-J network architecture onto a feed-forward form, we apply the following approximation:

$$\mathbf{y} = (\mathbf{I} + \mathbf{W})^{-1}\mathbf{x} \approx (\mathbf{I} - \mathbf{W})\mathbf{x} \quad (7)$$

In other words, we choose to implement the H-J rule with linear feed-forward networks of the type  $\mathbf{y} = \mathbf{W}\mathbf{x}$  with

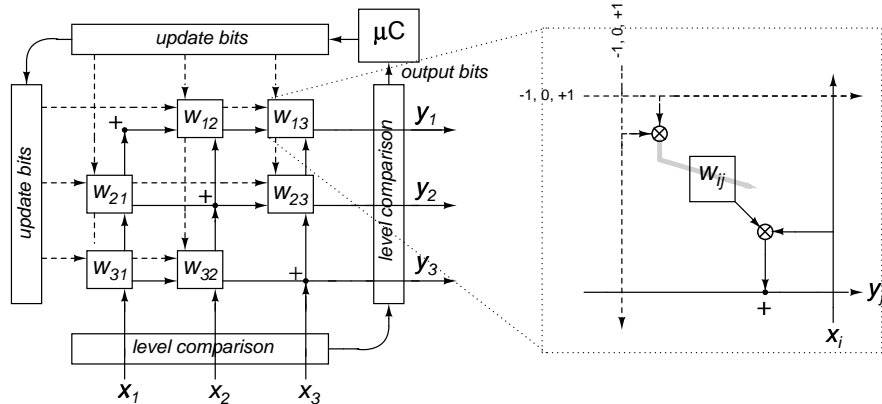


Fig. 2. Parallel implementation of ICA. General outer-product architecture.

fixed diagonal terms  $w_{ii} \equiv 1$ , and with off-diagonal terms adapting according to

$$\Delta w_{ij} = -\mu f(y_i)g(y_j), \quad i \neq j \quad (8)$$

Equivalently, the implemented update rule can be seen as the gradient of *InfoMax* (4) multiplied by  $\mathbf{W}^T$ , rather than the natural gradient multiplication factor  $\mathbf{W}^T \mathbf{W}$ . To obtain the full natural gradient in outer-product form, it is necessary to include a back-propagation path in the network architecture, and thus additional silicon resources, to implement the vector contribution  $\mathbf{y}^T \mathbf{W}$  [12].

Interestingly, in the special case of a  $2 \times 2$  network (2 sources and 2 observations) the the update rule (8) reduces to non-holonomic (zero-diagonal) form of the natural gradient rule (5), or equivalently the C-U rule (6), without a backward signal path in the architecture. This equivalence follows from the condition  $\lambda_1 = f(y_1)g(y_1)$  and  $\lambda_2 = f(y_2)g(y_2)$  which can be met by dynamically adapting  $\Lambda$  accordingly.

## 4. IMPLEMENTATION

### 4.1. Architecture

Level comparison provides implementation of discrete approximations of any scalar function  $f(y)$  and  $g(y)$  appearing in different learning rules. Since speech signals are approximately Laplacian distributed, the nonlinear scalar function  $f(y)$  is approximated by  $\text{sign}(y)$  and implemented using single bit quantization. Conversely, a linear function  $g(y) \equiv y$  in the learning rule is approximated by a 3-level staircase function  $(-1, 0, +1)$  using 2-bit quantization. The quantization of the  $f$  and  $g$  terms in the update rule (8) simplifies the implementation to that of discrete counting operations.

The functional block diagram of a  $3 \times 3$  outer-product incremental ICA architecture, supporting a quantized form

of the general update rule (8), is shown in Figure 2. Unmixing coefficients are stored digitally in each cell of the architecture. The update is performed locally by once or repeatedly incrementing, decrementing or holding the current value of counter based on the learning rule served by the micro-controller. The 8 most significant bits of the 14-bit counter holding and updating the coefficients are presented to a multiplying D/A capacitor array [4] to linearly unmix the separated signal. The remaining 6 bits in the coefficient registers provide flexibility in programming the update rate to tailor convergence.

### 4.2. VLSI Implementation

A prototype  $3 \times 3$  mixed-signal ICA processor was designed, fabricated, and tested. The differential analog input channels directly interface with gradient output signals from a previously developed gradient flow processor for acoustic localization [4], to extend its functionality to joint separation and localization of up to three acoustic sources. The mixed-signal architecture is implemented using fully differential switched-capacitor sampled-data circuits. Correlated double sampling performs common mode offset rejection and  $1/f$  noise reduction. An external micro-controller provides flexibility in the implementation of different learning rules. The ICA architecture is integrated on a single  $3mm \times 3mm$  chip fabricated in  $0.5 \mu m$  3M2P CMOS technology. Details of circuit implementation and characterization of the chip will be presented elsewhere.

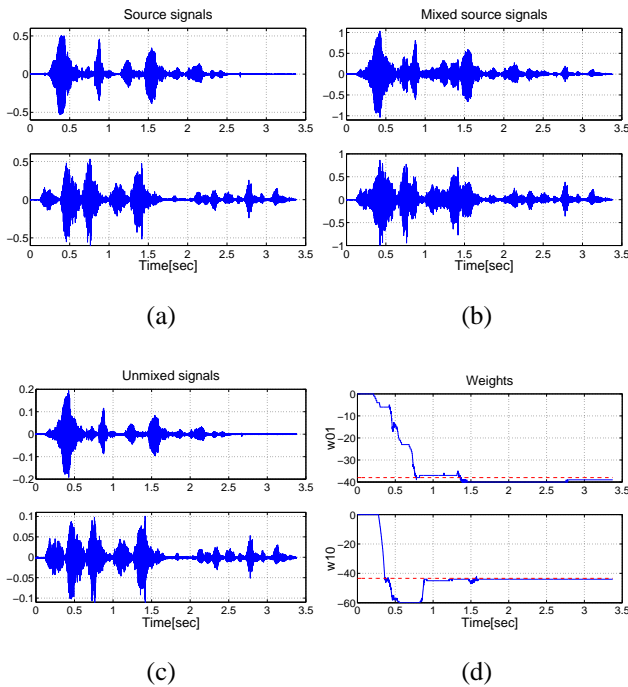
## 5. EXPERIMENTAL RESULTS

Initial experiments on the current prototype chip consisted of applying synthetic mixtures of speech signals as inputs. The original speech signals shown in Figure 3 (a) were

mixed using the linear matrix:

$$\begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$$

The synthetic mixture shown in Figure 3-(b) was applied to the ICA processor by using a 12 bit D/A converter. The on-line estimated signals, acquired from the output channels of the chip using 12 bit A/D conversion, are shown in Figure 3 (c). The estimates converge towards the original (unseen) sources in Figure 3 (a) over time, with 30 dB separation at convergence. Figure 3 (d) illustrates the dynamics of the recorded off-diagonal matrix elements  $w_{12}$  and  $w_{21}$  converging towards their ideal values (dashed lines).



**Fig. 3.** Experimental blind separation of two speech sources by the VLSI chip. (a): Original speech signals; (b): Mixture of speech signals; (c): Recovered signals obtained from chip; (d): ICA weights recorded on-line from chip.

## 6. CONCLUSION

A mixed-signal parallel VLSI architecture for implementation of ICA in general outer-product form was presented. Initial experimental results from a prototype micropower chip demonstrated 30 dB real-time separation of mixed speech signals presented to the chip. Further experimentation is directed towards integrating the ICA VLSI architecture with previously developed circuits implementing gradi-

ent flow [4], for real-time adaptive acoustic source separation and localization, using miniature acoustic arrays. The ICA architecture is directly amenable to integration with sensor arrays for small, compact, battery-operated “smart” sensor applications in hearing aids, personal digital assistants, and surveillance networks.

## 7. REFERENCES

- [1] A.S. Bregman, *Auditory Scene Analysis, The Perceptual Organization of Sound*, Cambridge MA: MIT Press, 1990.
- [2] G. Cauwenberghs, M. Stanacevic, and G. Zweig, “Blind Broadband Source Localization and Separation in Miniature Sensor Arrays,” *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS’2001)*, Sydney, Australia, May 6-9, 2001.
- [3] M. Stanacevic and G. Cauwenberghs, “Mixed-Signal Gradient Flow Bearing Estimation”, *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS’2003)*, Bangkok Thailand, May 25-28, 2003.
- [4] M. Stanacevic and G. Cauwenberghs, “Micropower Mixed-Signal Acoustic Localizer,” *Proc. IEEE Eur. Solid State Circuits Conf. (ESSCIRC 2003)*, Estoril Portugal, Sept. 16-18, 2003.
- [5] Cohen, M.H., Andreou, A.G. “Analog CMOS Integration and Experimentation with an Autoadaptive Independent Component Analyzer,” *IEEE Trans. Circuits and Systems II*, vol 42 (2), pp 65-77, Feb. 1995.
- [6] Gharbi, A.B.A., Salam, F.M.A. “Implementation and Test Results of a Chip for the Separation of Mixed Signals,” *Proc. Int. Symp. Circuits and Systems (ISCAS’95)*, May 1995.
- [7] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, New York: John Wiley, 2002.
- [8] C. Jutten and J. Herault, “Blind Separation of Sources .I. an Adaptive Algorithm Based on Neuromimetic Architecture,” *Signal Proc.*, vol. 24 (1), pp 1-10, 1991.
- [9] A.J. Bell and T.J Sejnowski. “An Information-Maximization Approach to Blind Separation and Blind Deconvolution,” *Neural Computation*, vol. 7, pp 1129-1159, 1995.
- [10] A. Cichocki, R. Unbehauen, L. Moszczanski, and E. Rummert. “A New On-Line Adaptive Learning Algorithm for Blind Separation of Sources,” *Int. Symp. Artificial Neural Networks ISANN-94*, pp 406-411, Taiwan, Dec. 1994.
- [11] A. Cichocki, R. Unbehauen, and E. Rummert. “Robust Learning Algorithm for Blind Separation of Signals,” *Electronics Letters*, vol. 30 (17), pp 1386-1387.
- [12] M. Cohen and G. Cauwenberghs, “Blind Separation of Linear Convulsive Mixtures through Parallel Stochastic Optimization,” *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS’98)*, Monterey CA, vol. 3, pp. 17-20, 1998.