International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:4, No:1, 2010

Floating-Point Scaling for BSS Gain Control

Abdelmalek Fermas, Adel Belouchrani, and Otmane Ait Mohamed

Abstract—In Blind Source Separation (BSS) processing, taking advantage of scaling factor indetermination and based on the floating-point representation, we propose a scaling technique applied to the separation matrix, to avoid the saturation or the weakness in the recovered source signals. This technique performs an Automatic Gain Control (AGC) in an on-line BSS environment. We demonstrate the effectiveness of this technique by using the implementation of a division free BSS algorithm with two input, two output. This technique is computationally cheaper and efficient for a hardware implementation.

Keywords—Automatic Gain Control, Blind Source Separation, Floating-Point Representation, FPGA Implementation.

I. Introduction

DIGITAL Signal Processing (DSP) algorithms are typically some of the most challenging computations. They are often need to be done in real-time, and require a large dynamic range. The requirements for performance and a large dynamic range lead to the use of floating-point number system [1].

Recently, BSS has received attention because of its potential applications such as speech recognition systems, telecommunications and medical signal processing. The problem consists of identifying a system where only output is observed. Source separation may be obtained by first identifying the directional vectors associated to each source and then by projecting the array signal onto the estimated vectors. This is a standard problem in array processing except that in BSS problem, we perform system identification without resorting to the knowledge of the directional vectors. Hence, blind source separation is essentially unaffected by errors in the propagation model or in array calibration.

In VLSI implementation, divisions are more complex to implement than multiplications and require more resources [2]. In this sense, a specific Analytical Second Order Blind Identification (ASOBI) algorithm has been derived considering the temporal coherence properties of the input sources as well as the inherent indeterminacies of the BSS processing. The ASOBI algorithm is division free and more suitable for hardware implementation [3].

The main contribution of this paper is a new technique for solving the recovered source signals errors by taking advantage of the scaling factor indetermination in blind processing and the floating-point representation.

The paper is organized as follows: The ASOBI algorithm is briefly presented in section II. Section III describes the implementation of BSS block processing environment. The solution for the scaling problem by using floating-point representation is presented in section IV. Section V describes the evaluation experiments and shows the results. Finally, we discuss related issues and conclude this paper in Section VI.

A. Fermas is with the Communications Systems Laboratory of Ecole Militaire Polytechnique, Algiers, Algeria e-mail: am.fermas@gmail.com.
A. Belouchrani is with the Electrical Engineering Department of Ecole Nationale Polytechnique, Algiers, Algeria e-mail: adel.belouchrani@enp.edu.dz.
O. Aitmohamed is with the ECE Department, Concordia University, Montreal, Quebec, Canada e-mail: ait@ece.concordia.ca.

II. The ASOBI Algorithm

Consider an array of 2 sensors receiving signals from 2 narrow band sources. The array output denoted $\mathbf{x}(t)$ is a 2×1 random vector, corrupted by additive white noise denoted $\mathbf{n}(t)$ and classically modeled as:

$$\mathbf{x}(t) = \mathbf{y}(t) + \mathbf{n}(t) = \mathbf{H}\mathbf{s}(t) + \mathbf{n}(t) \tag{1}$$

where $\mathbf{s}(t)$ is a 2×1 vector whose p-th component denoted $s_p(t)$ is the signal emitted by the p-th source. The 2×2 matrix:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}$$

is assumed to be full rank but otherwise unknown. The source signals are temporally colored, second order stationary and mutually uncorrelated processes.

The correlation matrices of $\mathbf{x}(n)$ are given by:

$$\mathbf{R}_{x_1x_1} = h_{11}^2\mathbf{R}_{s_1s_1} + h_{12}^2\mathbf{R}_{s_2s_2} + \sigma^2\mathbf{I} \tag{2}$$
$$\mathbf{R}_{x_2x_2} = h_{21}^2\mathbf{R}_{s_1s_1} + h_{22}^2\mathbf{R}_{s_2s_2} + \sigma^2\mathbf{I} \tag{3}$$
$$\mathbf{R}_{x_1x_2} = h_{11}h_{21}\mathbf{R}_{s_1s_1} + h_{12}h_{22}\mathbf{R}_{s_2s_2} \tag{4}$$

where $\mathbf{x}(n) = [x_1(n)\ x_2(n)]^T$, $\mathbf{I}$ is the $N\times N$ identity matrix, and $\mathbf{R}_{xy}$ is defined as

$$\mathbf{R}_{xy} = E([x(1),\cdots,x(N)]^T[y(1),\cdots,y(N)]) \tag{5}$$

$E(.)$ being the expectation operator and N is some chosen window length which can be a power of 2 so that a division by N becomes a simple bit shifting.

The aim is to calculate the separation matrix $\mathbf{W}$ and then use it for recovering the emitted sources. The solution for this blind identification system is obtained using ASOBI algorithm which requires three processing steps:

A. The Correlation Parametres (F_i and T_i)

Two operators $Off(.)$ and $Tr(.)$ are defined as:

$$Off(\mathbf{M}) = \sum_{i\neq j} M_{ij} \tag{6}$$
$$Tr(\mathbf{M}) = \frac{1}{N}\sum_{i} M_{ii} \tag{7}$$

where $\mathbf{M}$ is any square matrix of dimension $N \times N$ and M_{ij} are the entries of $\mathbf{M}$. By applying these operators to equations (2), (3) and (4), the following set of relations is obtained,

$$\begin{aligned} F_1 &= off(\mathbf{R}_{x_1x_1}) = h_{11}^2 R_1 + h_{12}^2 R_2 && (8)\\ F_2 &= off(\mathbf{R}_{x_2x_2}) = h_{21}^2 R_1 + h_{22}^2 R_2 && (9)\\ F_3 &= off(\mathbf{R}_{x_1x_2}) = h_{11}h_{21} R_1 + h_{12}h_{22} R_2 && (10)\\ T_1 &= tr(\mathbf{R}_{x_1x_1}) = h_{11}^2 + h_{12}^2 + \sigma^2 && (11)\\ T_2 &= tr(\mathbf{R}_{x_2x_2}) = h_{21}^2 + h_{22}^2 + \sigma^2 && (12)\\ T_3 &= tr(\mathbf{R}_{x_1x_2}) = h_{11}h_{21} + h_{12}h_{22} && (13) \end{aligned}$$

where $R_i = off(\mathbf{R}_{s_is_i}),\ i = 1, 2$. In (11), (12) and (13), we use the fact that, under unit-variance assumption, $tr(\mathbf{R}_{s_is_i}) = 1,\ i = 1, 2$.

B. The Mixing Matrix ($\mathbf{H}$*)*

Solving equations (8)-(13) and taking advantage of the inherent indeterminacies of the blind processing, leads to the following simplified solution:

$$\mathbf{H} = \begin{pmatrix} bF_1 - (T_1 - \sigma^2)d_1 & bF_3 - T_3 d_2 \\ bF_3 - T_3 d_1 & bF_2 - (T_2 - \sigma^2)d_2 \end{pmatrix} \quad (14)$$

where $d_1 = a - c$ and $d_2 = a + c$, with

$$\begin{aligned} a &= 2F_3T_3 - (F_1(T_2 - \sigma^2) + (T_1 - \sigma^2)F_2) && (15)\\ b &= 2(T_3^2 - (T_1 - \sigma^2)(T_2 - \sigma^2)) && (16)\\ c^2 &= (F_1(T_2 - \sigma^2) - (T_1 - \sigma^2)F_2))^2 + 4(F_3(T_2 \\ &\quad -\sigma^2) - T_3F_2)(F_3(T_1 - \sigma^2) - T_3F_1). && (17) \end{aligned}$$

Note that the obtained solution does not involve any division operation and reduces in the same time the number of square root operations needed for the channel identification.

C. The Separation Matrix ($\mathbf{W}$*)*

Now, we need to calculate the weights $\mathbf{W}$ of the separation filter to achieve our task of source signal recovery.

Taking into account the inherent indeterminacies of BSS, the zero forcing solution which maximizes the signal to interference at the output of the filter is given by

$$\mathbf{WH} = \mathbf{PD}$$

where $\mathbf{P}$ and $\mathbf{D}$ are a permutation matrix and a diagonal matrix, respectively. The solution is given by

$$\mathbf{W} = \begin{bmatrix} h_{22} & -h_{12} \\ -h_{21} & h_{11} \end{bmatrix} \quad (18)$$

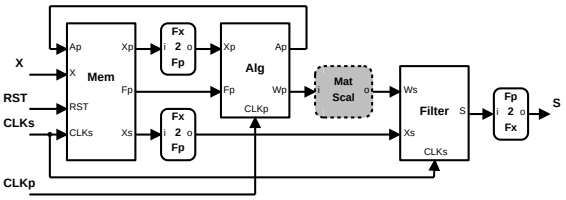

Fig. 1. Block diagram of BSS block processing

III. The BSS Environment

To support the implementation of a BSS block processing of algorithm, we propose the architecture of Fig. 1.

Figure 2 displays the three-stage pipeline composing the BSS block processing implementation (*Acq* = Acquisition of a frame of mixture samples, *Est* = Estimation of separation matrix, *Sep* = Separation of sources). The vertical and the horizontal axis represents successive frames and time respectively. So in the gray column, the earliest frame is in separation stage, the middle frame is in estimation operation and the latest frame is undergoing acquisition.

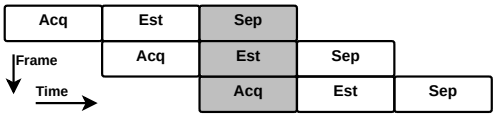

Fig. 2. Frame scheduling on the BSS block processing

The BSS Environment contains the following main blocks:

A. The Memory System (Mem)

In block processing algorithms, we need a block of samples available during each processing period. The i^{th} frame is stored in one of its two memory sub-blocks (Fig. 3). The $(i-1)^{th}$ frame is interfaced with the *Alg* as a read only memory (**Ap**: Address, **Xp**: Data) to compute the separation matrix $\mathbf{Wp}$ at the processing clock speed **CLKp**. The $(i-2)^{th}$ frame is outputted at port **Xs** as a stream to the *Filter* block.

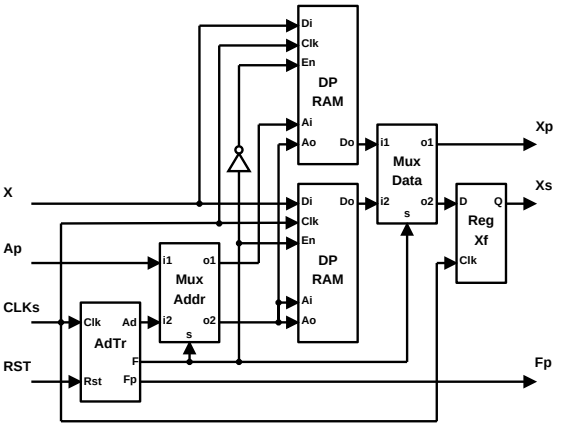

Fig. 3. Memory system block diagram

B. The Fx2Fp and Fp2Fx Blocks

Implement fix to floating-point conversion and vice versa, respectively. For the implementation, the library *FPLibrary* of parameterizable arithmetic operators for real numbers has been used [4].

C. The BSS Algorithm (Alg)

The top level block diagram of the ASOBI algorithm implementation (Fig. 4) has been modeled into three main parts namely Correlation Matrix (*CM*), Mixing Matrix (*HM*) and Separation Matrix (*WM*) as presented in [5]. This block can seek each vector of mixed signals on the data port **Xp** which corresponds to the address presented at port **Ap** from the memory system. A high level of the frame synchronization pulse **Fp** transmitted from the *Mem* block to the *Alg* block, allows the initialization of the algorithm. The output of this block is the estimated separation matrix **Wp**.

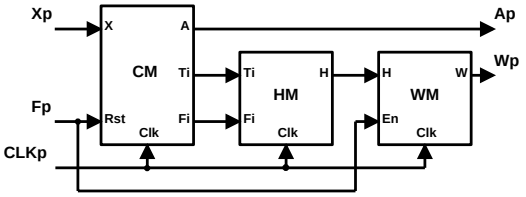

Fig. 4. ASOBI implementation block diagram

D. The Matrix Scaling (MatScal)

This block reduce the estimated separation matrix **Wp** dynamics through substraction operations, applying the proposed technique presented in section IV. In result, the scaled matrix **Ws** is used in the next block to recover the source signals avoiding a saturation or a weakness at output.

E. The Separation Filter (Filter)

This block performs the separation itself using the weighting vectors in the scaled separation matrix **Ws** and the samples of the mixture signals **Xs** to recover the estimated sources **S**:

$$\mathbf{S} = \mathbf{WsXs} \tag{19}$$

It include the sources number of beam former (Fig. 5).

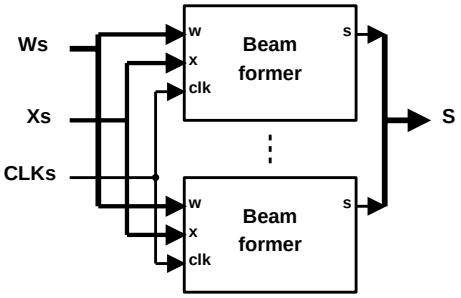

Fig. 5. Separation filter block diagram

IV. The Floating-Point Scaling

In blind context, complete identification of the mixture matrix $\mathbf{H}$ is impossible as shown by the following relation:

$$\mathbf{x}(t) = \mathbf{Hs}(t) + \mathbf{n}(t) = \sum_{p=1}^{2} \frac{\mathbf{h_p}}{\alpha_p} \alpha_p s_p(t) + \mathbf{n}(t) \tag{20}$$

where $\alpha_p \in \mathbb{R}$ and $\mathbf{h_p}$ denotes the p-th column of $\mathbf{H}$. Hence, the exchange of a fixed scalar factor between a source signal and the corresponding column of $\mathbf{H}$ leaves the observations unaffected [3].

In the same way, the exchange of a fixed scalar factor between a mixture signal and the corresponding line of the separation matrix $\mathbf{Wp}$ doesn't affect the estimated source.

When the estimated separation matrix $\mathbf{Wp}$ can be written as: $\mathbf{Wp} = \mathbf{Ws}2^{\alpha}$ with $|\alpha|$ a large scaler, so the estimated sources can be either saturated ($\alpha > 0$) or weakened ($\alpha < 0$).

To overcome this scaling problem, the floating-point representation is used.

In general, a floating-point number F presented in Fig 6, can be expressed as follows:

$$F = (-1)^s 1.f 2^{e-b} \tag{21}$$

Where s is the sign bit, f is the fraction and e is the biased exponent.

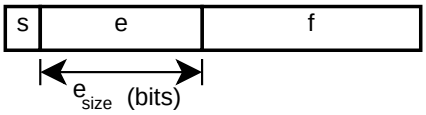

Fig. 6. Floating-Point number representation

The actual exponent is the value of the exponent field minus the bias. The value of bias b depends on the size of exponent e_{size} as in equation (22).

$$b = 2^{e_{size}-1} - 1 \tag{22}$$

The idea is: for each source associated to the estimated separation matrix $\mathbf{Wp}$ line i:

- keep all signs s_{ij} and fractions f_{ij} fields unchanged (the same orientation of the directional vectors):

$$f'_{ij} = f_{ij}, \tag{23}$$
$$s'_{ij} = s_{ij}; \tag{24}$$

- reduce the dynamic of the biased exponent e_{ij} (change the directional vectors amplitude):

$$e'_{ij} = e_{ij} - d_i, \tag{25}$$

where $d_i = e_{i1} - b$, represents the dynamic relative to the reference e_{i1}.

It is clear from equation (25), that the new exponent field of the reference e'_{i1} will be:

$$e'_{i1} = e_{i1} - d_i = e_{i1} - (e_{i1} - b) = b, \tag{26}$$

so the reference element actual exponent of the scaled separation matrix will be equal to zero. The actual exponent for the others elements is reduced with the same amount d_i and will be near to zero for each line i. Hence, the result separation matrix dynamics is decreased and adapted to a correct source recovery.

The hardware implementation of this technique is presented in Fig. 7, where we can notice that its need only one subtraction operation for each matrix element.

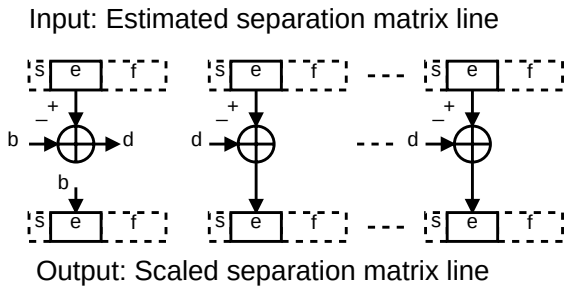

Fig. 7. Floating-point based scaling technique implementation

V. The Implementation Results

We first present a sample run of the proposed hardware implementation, consisting of two speech signals (Fig. 8.a), which they are mixed (Fig. 8.b) by the following matrix,

$$\mathbf{H} = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 0.8 \end{bmatrix}. \tag{27}$$

It appears from figure 8.c that without using a scaling technique, the recovered signals are saturated in this case. A correct estimation of the source signals is provided (8.d), using this new scaling technique.

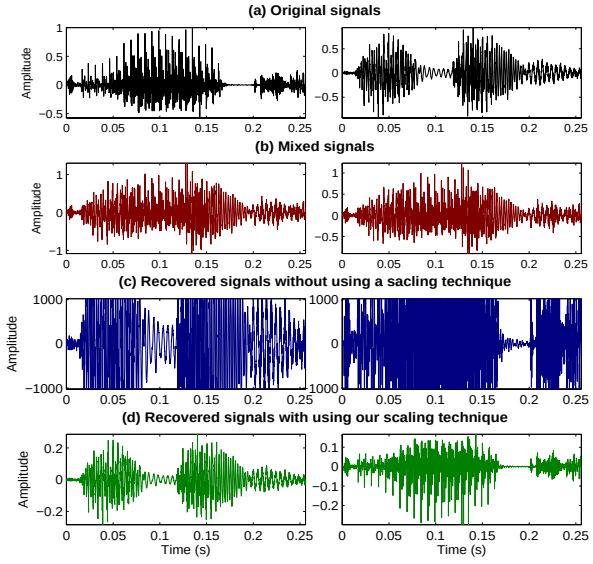

Fig. 8. Speech signals separation example

Also, we have used an FPGA Virtex-5 to assess the performances of the implementation and to show the effect of the word length and the sample size on the resource utilization, the maximum working frequency and the separation quality. This quality is characterized in terms of signal rejection ratio as discussed in [3]. We recall that lower is this ratio better is the separation quality.

TABLE I
FPGA IMPLEMENTATION RESULTS

Word length (bits)	Sample size	Number of slices		Maximum frequency (MHz)	Rejection ratio (dB)
		Logic	DSP48Es		
24	128	12716	26	34.00	-12.60
	256	12892	26	34.00	-16.53
	512	13176	26	33.98	-32.10
32	128	16269	52	26.93	-13.76
	256	16465	52	26.93	-19.27
	512	16770	52	26.93	-41.42

From Table I, we can see that the working frequency isn't affected by the sample size but only by the word length. This due to the fact that the operation critical path is affected by the word length. Furthermore, one can observe that when the word length and/or the sample size increase the rejection ratio decreases which means that we have a good separation.

VI. Conclusion

We have designed and implemented a BSS block processing environment needed for the hardware implementation in real-time applications of related algorithms such as ASOBI.

A scaling technique based on floating-point representation is proposed and implemented to solve the separation matrix scaling problem. We have overcome this problem and obtained a correct source separation.

In blind context, this dynamic reduction technique perform an Automatic Gain Control in Multiple-Input Multiple-Output systems as BSS.

From hardware complexity point of view, this scaling technique can be achieved in one clock cycle and requires low resources cost. We keep the entire architecture of BSS environment division free as the ASOBI algorithm implementation.

References

[1] B. Fagin and C. Renard, "Field Programmable Gate Arrays and Floating Point Arithmetic," in *IEEE Trans. on VLSI Systems*, Vol. 2, Sept. 1994, pp. 365-367.

[2] J.-P. Deschamps and G. Sutter, "Finite Field Division Implementation," in *15th Int. Conf. on Field Programmable Logic and Applications*, 2005.

[3] A. Belouchrani, E. Bourennane and K. Abed-Meraim, "A closed form Solution for the Blind Separation of Two Sources from Two Sensors," in *14th European Signal Processing Conf.*, 2006.

[4] J. Detrey and F. de Dinechin, "A VHDL Library of LNS Operators," in *37th Asilomar Conf. on Signals, Systems and Computers*, 2003.

[5] A. Fermas, A. Belouchrani and O. Aitmohmed, "Hardware Implementation of Free Division Block-based BSS Algorithm," in *7th IEEE Int. NEWCAS Conf.*, 2009.

Abdelmalek Fermas received the State Engineering degree in Electrical Engineering in 2000 and the M.Sc. degree in Communications Systems in 2005 from the Ecole Militaire Polytechnique (EMP) of Algiers, Algeria, where, he is currently a Ph.D. student. His research interests include Communications Systems and VLSI Implementation for Signal Processing.

Adel Belouchrani received the State Engineering degree in 1991 from the Ecole Nationale Polytechnique (ENP), Algiers, Algeria and the M.Sc. degree in Signal Processing from the Institut National Polytechnique de Grenoble (INPG), France, in 1992, and the Ph.D. degree in the field of signal and image processing from Ecole Nationale Superieure des Telecommunications (ENST), Paris, France, in 1995.

He was a Visiting Scholar at the Electrical Engineering and Computer Sciences Department of University of California at Berkeley, USA, from 1995 to 1996, working on fast adaptive blind equalization and carrier phase tracking. He was with the Department of Electrical and Computer Engineering of Villanova University, Pennsylvania, USA, as research associate from 1996 to 1997. During this period, he was also a consultant to Comcast Inc. During February 1997, he was a Visiting Scientist at the Laboratory for Artificial Brain System, Riken, JAPAN. From August 1997 to October 1997, he was with Alcatel ETCA, Belgium, working on Very high speed Digital Subscriber Line (VDSL).

He is currently with the Electrical Engineering Department of the Ecole Nationale Polytechnique, Algiers, Algeria, as Full Professor. His research interests are in the areas of Statistical Signal Processing, Array Signal Processing, Blind Source Separation, Multi-Channel Deconvolution, Digital Communications, Wireless Communications, Time Frequency Representations, Application of Spatial Time Frequency Distributions to Array Signal Processing, VLSI Implementation for Signal Processing, and Biomedical Signal Processing.

Otmane Ait Mohamed completed his Ph.D. in 1996 in the University Henri Pioncare, France. He later joined the University of Montreal as a post doctoral fellow then a research associate within the LASSO group, focusing his research on the problem of non-termination in the reachability analysis using Multiway Decision Graph.

Prior to joining Concordia University, he spent three years with Nortel Networks, Ottawa, working on hardware verification. He was has also worked at Cistel Technology Inc and lectured at the University of Montreal.

His research interests include Hardware Verification and Body Sensor Networks.