

## A Comparative Study and Implementation of Blind Source Separation Algorithm using MATLAB and TMS320c6713 DSK

<sup>1</sup>Mohcin Mekhfioui, <sup>2</sup>Rachid Elgouri, <sup>1</sup>Amal Satif, <sup>3</sup>Aziz Benahmed, <sup>3</sup>El-Mehdi Hamzaoui,  
<sup>1</sup>Hadjoudja Abdelkader and <sup>1</sup>Laamari Hlou

<sup>1</sup>Laboratory of Electrical Engineering and Energy Systems, Faculty of Science,

<sup>2</sup>Laboratory of Electrical Engineering and Telecommunications Systems,

Écoles Nationales des Sciences Appliquées, Ibn Tofail University, Kenitra, Morocco

<sup>3</sup>National Centre for Nuclear Energy, Science and Technology, Rabat, Morocco

**Abstract:** The Blind Source Separation (BSS) consists in identifying a set of signals from an unknown source from a set of observed signals. It can be used for image processing, acoustics, telecommunications and biomedical applications. To perform the BSS different methods and properties are carried out. Independent Component Analysis (ICA) is frequently used. In our study, we are running a BSS algorithm on a portable environment the DSK TMS320C6713 card after the validation of this algorithm on MATLAB. The test signals are recorded by two microphones and sent to the DSK TMS320C6713 card via a jack cable. By calculating the Signal to Interference Ratio (SIR) and the Performance Index (PI), we have proved that the algorithm SOBI (Second Order Blind Identification) of Belouchrani is the most efficient to analyze our signal database, so the separation is performed on the DSK TMS320C6713 using the SOBI algorithm to validate the its performance of the in real time.

**Key words:** Blind Source Separation (BSS), ICA, SOBI, TMS320C6713 DSK, independent, environment

---

### INTRODUCTION

The Blind Source Separation (BSS) is a rapidly developing discipline in the signal processing community that offers a solution for separating mixed signals during their propagation. The BSS, therefore, consists in estimating a set of unknown source signals from a set of observed signals that are mixtures of these source signals. Without any additional hypothesis, the problem carried out by the BSS is a wrong problem posed. This is why all BSS methods have assumptions about both sources and mixing.

Source separation techniques have been applied in various scientific and technological fields such as telecommunications, acoustics, imaging and biomedical signal processing.

The first works were published by Herault Jutten and Ans in 1980 (Jutten and Herault, 1991) about mixtures called instant linear. Other work, represent a biological problem that initiated studies on source separation and consists in treating the responses emitted by the muscles subjected to different kinds of excitations (Jutten and Taleb, 2000).

Jutten and Herault (1991) have presented a first mathematical formalism of the BSS problem and an algorithm to reach a solution in 1991. Then the Comon research in 1994 (Comon, 1994) allowed the

mathematical formalization in the simplest case of an instantaneous linear mixture leading to the concept of Independent Component Analysis (ACI).

Previously, few studies are carried out on the real-time implementation of blind source separation. Du and Qi (2004) proposed a parallel ICA FPGA implementation for dimensionality reduction of hyper spectral images by using a RAM location as an interface to communicate with the central processing unit. Bravo *et al.* (2006) implemented the Jacobi method translated into VHDL on the FPGA to solve the eigenvalue and eigenvector problems. In 2008 Shyu *et al.* performed an implementation of Pipelined Fast ICA on FPGA for the real-time separation of blind sources by manually coding the Fast ICA algorithm in Hardware Description Language (HDL) for the same purpose. Min *et al.* (2012) carried out an implementation of the Fast ICA algorithm in a DSP card for blind source separation. Timothy and Joonwan (Hong and Kim, 2015) implemented the JADE algorithm on an FPGA using VHDL furthermore Tornez-Xavier and all Tornez-Xavier *et al.* (2015) worked on FPGA implementation of the ICA algorithm using multiplexing.

Meyer-Baese *et al.* (2015) investigated the ability and efficiency of FPGA independent component analysis algorithms to perform real-time separation of blind sources as well as Singh *et al.* (2017) proposed an FPGA

implementation of blind source separation using a new ICA algorithm based on the rotation of the Jacobian matrix and the MATLAB system generator blocks. In 2018, Yiu and Yong Low conduct a real-time signal blindness noise reduction system exploiting the spatial diversity of source mixtures received by different sensors and kurtosis measurement.

In this research, we present a comparison between the different algorithms of the BSS based on the calculation of the Performance Index (PI) and the Signal to Interference Ratio (SIR), the optimal algorithm obtained will be implemented in the DSK TMS320C6713 card to test the its performance in real time.

## MATERIALS AND METHODS

**Blind Source Separation (BSS):** A the main objective of the Blind Source Separation (BSS) is to estimate a set of unknown source signals  $s(t)$  using only the mixed signals obtained by a series of sensors  $x(t)$  also called "observations" Fig. 1(Arahmane *et al.*, 2017).

The term blind denotes the fact that there is no a priori information about the sources or how they have been mixed. In the general case, three conditions are necessary to perform this technique:

- The sources is statistically independent
- The number of sensors is higher or equal to the number of sources
- A mixing matrix between the sources and the sensors

**Mathematical model:** The basic model of the Blind Source Separation (BSS) depends on the assumed mixing model in the general case the equation of the instantaneous linear mixing is written as follows (Pelegrina *et al.*, 2019):

$$x_i(t) = \sum_{j=1}^n a_{ij}s_j(t) + v_i(t) \quad (1)$$

$$i=1, 2, \dots, m \text{ and } j=1, 2, \dots, n$$

Where:

$x(t) = [x_1(t), \dots, x_m(t)]^T$  : The vector containing the observations  
 $a_{ij}$  is the mixing system

$s(t) = [s_1(t), \dots, s_n(t)]^T$  : The vector containing N signals emitted by N unknown sources

$v(t)$  : An additive noise vector, we consider it negligible in the rest of the calculations

The matrix form of Eq. 1 is written as follows:

$$X(t) = HS(t) \quad (2)$$

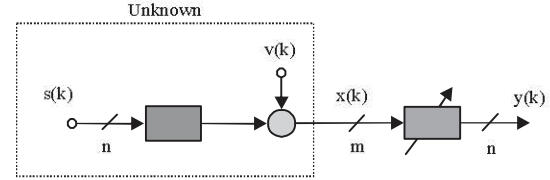


Fig. 1: Principle of the separation of sources (Arahmane *et al.*, 2017)

where, H is a mixture matrix of size  $n \times m$ . It is about finding in the ideal case the matrix W of size  $n \times m$  which inverts the mixture and provides the output vector:

$$y(k) = Wx(k) = WS(k) \approx s(k) \quad (3)$$

The estimated sources are given by the vector  $s(k)$  and their corresponding projections for the different microphones are given by the estimated matrix:  $H = W^{-1}$ .

**Evaluation criteria:** The efficiency of the algorithms of the blind source separation is evaluated by the calculation of the Signal-to-Interference Ratio (SIR) and the separability Performance Index (PI).

**Signal to Interference Report (SIR):** The SIR is calculated to verify the accuracy of latent source separation by an algorithm running the BSS. It is expressed in dB is defined for each source  $s_j$  by (Novey and Adali, 2006):

$$SIR_j = \max_i \left\{ 10 \cdot \log_{10} \left( \frac{g_{ji}^2}{\sum_{k \in [1, N], k \neq j} g_{ik}^2} \right) \right\} \forall j \in [1, N] \quad (4)$$

where, the values  $g_{ij}$  ( $i, j=1, \dots, N$ ) are the coefficients of the matrix of performances G equal to the product of the matrices of mixture A and of separation W,  $G = A * W$ . Knowing that each report  $SIR_j$  quantify the quality of source separation  $s_j$ . We define the SIR which quantifies the performance quality of the BSS algorithm used to separate all sources by:

$$SIR = \frac{1}{N} \sum_{j=1}^N SIR_j \quad (5)$$

More SIR, the higher the separation is good and the BSS method used is efficient.

**Performance Index of separability (PI):** The Performance Index of the separability (PI) is given by the following equation (Laassiri *et al.*, 2017):

$$PI = \frac{1}{n(n-1)} \sum_{i=1}^n \left\{ \left( \sum_{k=1}^n \frac{|g_{ik}|}{\max_j |g_{ik}|} - 1 \right) + \left( \sum_{k=1}^n \frac{|g_{ik}|}{\max_j |g_{ik}|} - 1 \right) \right\} \quad (6)$$

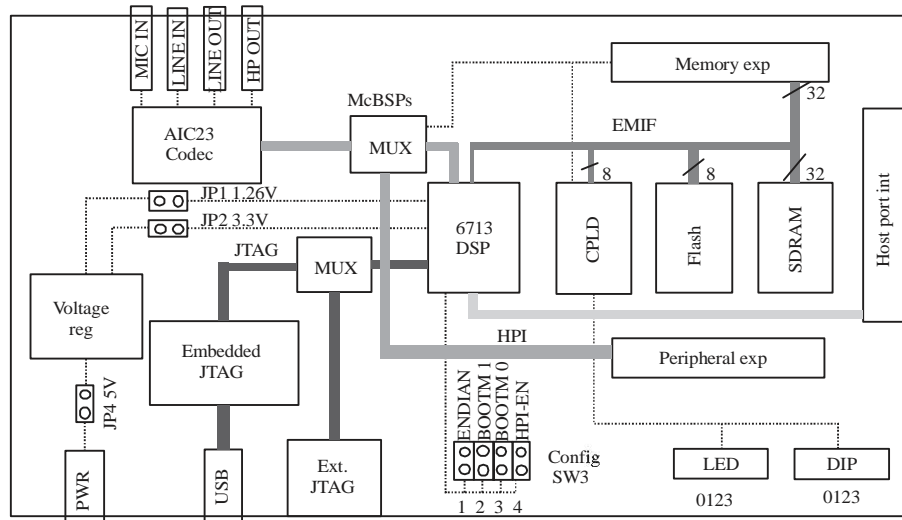


Fig. 2: TMS320C6713DSK physical layout (Texas Instruments, 2006)

where,  $g_{ij}$  is the  $(i, j)$  element of the global matrix of the system  $G = WH$  and  $\max_i(g_{ij})$  presents the maximum value among the elements in the  $i$ th row vector of  $G$ . Similarly the term  $\max_j(g_{ij})$  corresponds to the maximum value among the elements in the  $j$ th column vector of  $G$ .

When the perfect separation is obtained, the performance index PI is zero. In practice, a PI value of about  $10^{-2}$  indicates a relatively good performance in terms of separation (Arahmane *et al.*, 2018).

**The DSK TMS320C6713 board:** The TMS320C6000 platform of the DSP digital signal processors is part of the TMS320 family of TEXAS INSTRUMENTS. It includes the TMS320C62x fixed arithmetic and TMS320C67x floating arithmetic processors. The TMS320C6713 is considered the most efficient member of the C67x category, its VLIW architecture allows it to process 8 instructions in parallel by 8 functional units. The discussion in this section will focus on the TMS320C6713 processor. The architecture and associated devices will also be discussed (Texas Instruments, 2006).

The DSK evaluation is a high-performance board with the required hardware and software support tools for real time signal processing. It is a complete DSP system. This card is equipped with the TMS320C6713 floating point processor and a 32-bit TLV320AIC23 (AIC23) stereo codec for input and output Fig. 2. A key feature includes:

- Single voltage power supply (+5V)
- Two McASPs
- TMS320C6713 DSP operating at 225 MHz
- AIC23 codec stereo
- 4 DIP switches and 4 user LEDs

- Software board configuration through registers implemented in CPLD
- 512KB non-volatile flash memory and 16 MB SDRAM
- 512M-byte total addressable external memory space
- JTAG emulator with USB host interface or external emulator

The TMS320C67x DSP generation is supported by the express DSP IT reference development toolkit, including a highly optimized C/C++ Compiler, Code Composer Studio Integrated Development Environment (CCSIDE), JTAG-based emulation and real-time debugging and the DSP/BIOS kernel. CCS offers solid central functions with user-friendly configuration and graphical visualization tools for system design. C/C++ programming for the application is respected, linked and executed by the CCS (Texas Instruments, 2006).

## RESULTS AND DISCUSSION

### Software simulation

**Signal model:** The base signal consists of two audio signals of sizes 2500 samples for each, the signals were recorded by a microphone, the first signal contains the words of a person “voice.mat” and the second contains a musical instrument called “instrument.mat”. These signals were combined linearly by a mixing matrix  $A$  to produce the input signals and (in this article, we use the same mixing matrix to compare the different algorithms).

$$A = \begin{bmatrix} 0.35 & 0.86 \\ 0.63 & 0.42 \end{bmatrix}$$



**Simulation results:** In our study, we used the TOOLBOX ICALAB (Version 3.0) grouping several algorithms of the blind source separation BSS that, we tested (20 algorithms tested) to determine the algorithm that gives the higher performance. However, tests have shown that only 9 algorithms are stable:

- SOBI: Second Order Blind Identification (Wang *et al.*, 2017)
- SAD: Sequential Approximate Diagonalization (Li and Zhang, 2007)
- FOBI: Fourth Order Blind Identification (Nordhausen and Virta, 2019)
- JADE: Joint Approximate Diagonalization of Eigen matrices (Cardoso and Souloumiac, 1993)
- SYM-WHITE: Symmetric Pre-Whitening algorithm
- AMUSE: Algorithm for Multiple Unknown Source Extraction (Tong *et al.*, 1991)
- POWERICA: Power iteration for ICA (Ding, 2006)
- FPICA: Fixed-Point ICA (Hyvärinen and Oja, 1997)
- UNICA: Unbiased quasi Newton algorithm for ICA

The results obtained by calculating the Signal-to-Interference (SIR) and the Performance Index of separability (PI) of these algorithms are presented in Table 1.

These results show that the SOBI algorithm (Second Order Blind Identification) from Belouchrani present a separation technique that suits more our case. Fig. 3 shows the origin signals ( $s_1$  and  $s_2$ ), the multiplexed signals ( $x_1$  and  $x_2$ ) and the output signals ( $y_1$  and  $y_2$ ) for the SOBI algorithm.

**About SOBI-Second Order Blind Identification:** The SOBI algorithm (Wang *et al.*, 2017) is based on the correlation property of the source signals which is realized by the multi-delay inter covariance matrix. Three calculation steps are required:

- Whitening
- Determining the unitary factor
- Joint Diagonalization (JD)

The main concept of the SOBI algorithm is the hypothesis on the diagonal shape of the staggered correlation matrices which is obtained by the following expression:

$$R_s(\tau) = E\{x(t)x(t+\tau)^T\} = AR_s(\tau)A^T, \forall \tau \quad (7)$$

where,  $R_s$  is the staggered correlation matrix  $R_s$ , the correlation matrix of the source signals. Whereas Eq. 7 is applicable to all  $\tau$  values, there is a unit matrix  $U = BA$  that jointly diagonalizes all correlation matrices:

$$U^T R_s(\tau) U = R_s(\tau) \quad (8)$$

Accordingly, the estimated original signals are calculated by  $U^* Bx(t)$ .

#### Implementation and experimental results

**Implementation in the TMS320C6713 DSK board:** In this part, we implemented the SOBI algorithm in a DSK TMS320C6713 board. To test the implementation, we first used two microphones to produce the input signals  $x_1$  and  $x_2$ , a speaker and an oscilloscope to view the output signals Fig. 4.

Table 1: The performance index obtained for the algorithms BSS tested

Algorithm	PI	SIR
SOBI	0.001	61.63
SAD	0.002	55.61
AMUSE	0.004	48.22
POWERICA	0.008	41.10
FPICA	0.011	39.27
JADE	0.025	34.71
FOBI	0.035	29.12
UNICA	0.079	23.58
SYM-WHITE	0.095	21.05

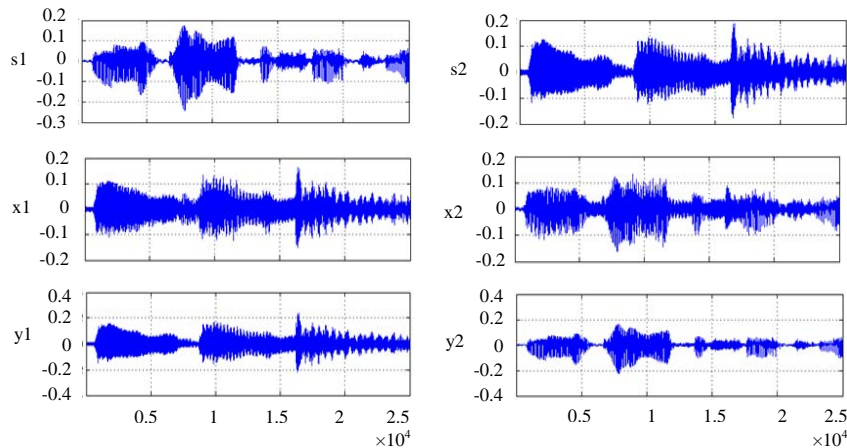


Fig. 3: Simulation results of the SOBI algorithm

The separation algorithm is designed in MATLAB/Simulink and is implemented on the TMS320C6713 board, all implementation steps start by generating the



Fig. 4: Hardware setup for system separation

Simulink Model for the system using the Simulink blocks in MATLAB until the file download '\*.out' on the DSP card, shown in Fig. 5.

Figure 6 shows a design of blind source separation by the real-time SOBI algorithm using the TMS320C6713 card under MATLAB/Simulink.

In addition, we designed a reliable module-based S function to evaluate the performance of the BSS technique by calculating the Signal to Interference (SIR) and separability Performance Index (PI) values.

**Experimental results:** In real application, the SOBI algorithm has been implemented and tested with two microphones and a speaker, we used the switches on the DSK TMS320C6713 board to choose the output to be shown on the oscilloscope or listened on the speaker. DSK TMS320C6713 board to choose the output to be shown on the oscilloscope or listened on the speaker.

Using our workstation Fig. 7, we can separate the input signals in real time using the SOBI algorithm and TMS320C6713 card. To observe the different signals

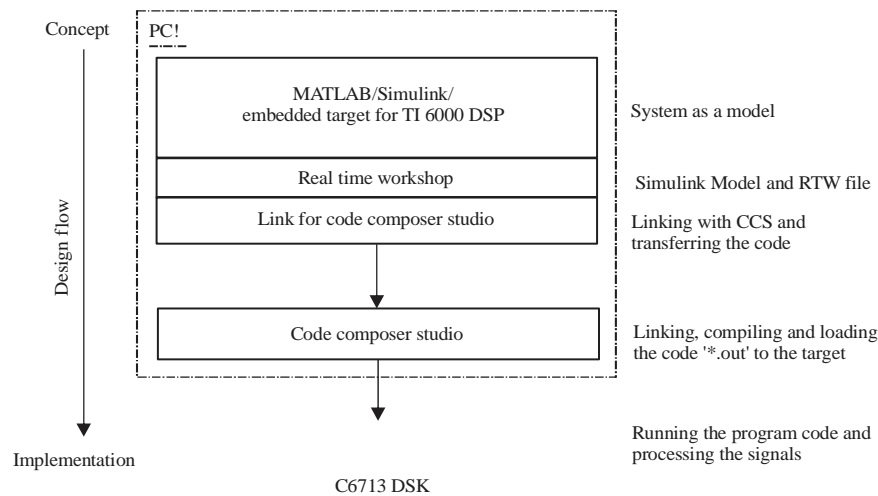


Fig. 5: Flow diagram of the procedures of implementation the model on C6713 DSK

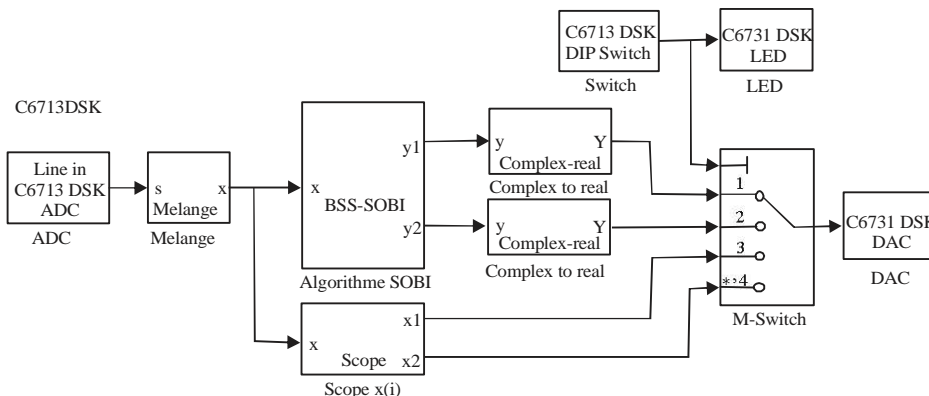


Fig. 6: Model real-time separation of tow audio signal



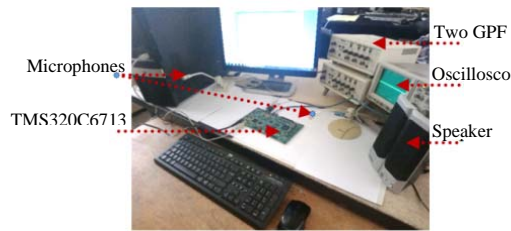


Fig. 7: Work workstation setup

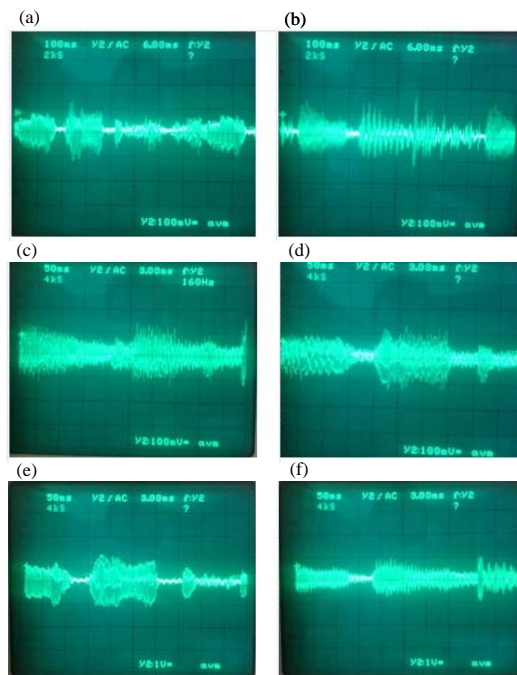


Fig. 8: Result of Real time implementation using the SOBI algorithm for an audio signal (a) Source signal  $s_1$ , (b) Source signal  $s_2$ , (c) Mixing signal  $x_1$ , (d) Mixing signal  $x_2$ , (e) Estimated signal  $y_1$  and (f) Estimated signal  $y_2$

(source signals, mixing signals and estimated signals), we should change the binary value of the switches of the card (Fig. 8).

To see the separation results clearly, we replaced the two microphones with two generators (GPF) to generate two basic signals ( $s_1$  and  $s_2$ ) of the same shape and different frequency (500 and 2500 Hz). These two signals will be mixed with the matrix  $A$  in the DSK board to produce the two mix signals  $x_1$  and  $x_2$ . The different input and output signals of the SOBI algorithm implemented in the TMS320C6713 board are presented in Fig. 9.

The results of the implementation verify that the behavior and graphics of our implemented model are similar to the results obtained by simulation on Simulink.

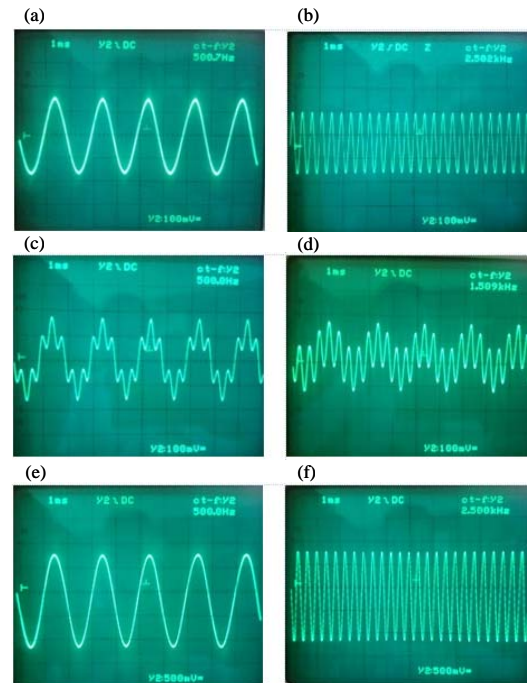


Fig. 9: Result of Real time implementation using the SOBI algorithm for a sinusoidal signal (a) Source signal  $s_1$ , (b) Source signal  $s_2$ , (c) Mixing signal  $x_1$ , (d) Mixing signal  $x_2$ , (e) Estimated signal  $y_1$  and (f) Estimated signal  $y_2$

We used the TMS320C6713 map to separate two linearly mixed signals by an  $A$  matrix, using the best-performance algorithm obtained by calculating the Performance Index of separability (PI) and the Signal-to-Interference Ratio (SIR). This study aims to reproduce the performance of the implementation of the Second Order Blind Identification (SOBI) algorithm as expected in MATLAB. The experiment carried out obtained in this study is still in the onset and provide a background to be developed in future works concerning the performance of the SOBI algorithm in real time which is a promising area of research.

## CONCLUSION

In this research, we presented a comparison of the blind separation of sources using different algorithms of the ICALAB tool on two audio signals picked up by two microphones and multiplied by a linear matrix  $A$ , based on the calculation of the Performance Index of separability (PI) and Signal to Interference (SIR) we found that the SOBI algorithm (Second order Blind Identification) from Belouchrani and al is most efficient to analyze our database. To validate this performance in practice and in real time, we implemented this algorithm

in the TMS320C6713 board, using two microphones and a speaker, we obtained accurate results similar to the results of MATLAB.

This research has thus gathered overall information about selecting the optimal algorithm to analyze any database in real time on a signal processing board and provided a platform for future research in the field of blind source separation in real time implementation.

## REFERENCES

- Arahmane, H., E.M. Hamzaoui and R.C. El Moursli, 2018. Blind nonnegative matrix factorization algorithms to estimate the neutron flux of fission chamber detector: Application to neutron-gamma discrimination. *J. Neutron Res.*, 20: 35-42.
- Arahmane, H., E.M. Hamzaoui, E. Moursli and R. Cherkaoui, 2017. Neutron flux monitoring based on blind source separation algorithms in Moroccan TRIGA MARK II reactor. *Sci. Technol. Nucl. Install.*, Vol. 2017,
- Bravo, I., P. Jiménez, M. Mazo, J.L. Lázaro and A. Gardel, 2006. Implementation in FPGAs of Jacobi method to solve the eigenvalue and eigenvector problem. *Proceedings of the 2006 International Conference on Field Programmable Logic and Applications*, August 28-30, 2006, IEEE, Madrid, Spain, pp: 1-4.
- Cardoso, J.F. and A. Souloumiac, 1993. Blind beamforming for non gaussian signals. *IEE Proc. F (Radar Signal Process.)*, 140: 362-370.
- Comon, P., 1994. Independent component analysis: A new concept? *Signal Process.*, 36: 287-314.
- Ding, S., 2006. A power iteration algorithm for ICA based on diagonalizations of non-linearized covariance matrix. *Proceedings of the 1st International Conference on Innovative Computing, Information and Control-Volume I (ICICIC'06)* Vol. 2, August 30-September 1, 2006, IEEE, Beijing, China, pp: 730-733.
- Du, H. and H. Qi, 2004. An FPGA implementation of parallel ICA for dimensionality reduction in hyperspectral images. *Proceedings of the 2004 IEEE International Geoscience and Remote Sensing Symposium (IGARSS '04)* Vol. 5, September 20-24, 2004, IEEE, Anchorage, Alaska, pp: 3257-3260.
- Herault, J. and C. Jutten, 1986. Space or time adaptive signal processing by neural network models. *AIP Conf. Proc.*, 151: 206-211.
- Hong, T.S. and J. Kim, 2015. FPGA implementation of JADE ICA algorithm. *Proceedings of the 2015 International SoC Design Conference (ISOCC'15)*, November 2-5, 2015, IEEE, Gyeongju, South Korea, pp: 31-32.
- Hyvarinen, A. and E. Oja, 1997. A fast fixed-point algorithm for independent component analysis. *Neural Comput.*, 9: 1483-1492.
- Jutten, C. and A. Taleb, 2000. Source separation: From dusk till dawn. *Proceedings of the 2nd International Workshop on Independent Component Analysis and Blind Source Separation (ICA' 2000)*, June 19-22, 2000, ICA, Helsinki, Finland, pp: 15-26.
- Jutten, C. and J. Herault, 1991. Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture. *Signal Proc.*, 24: 1-10.
- Laassiri, M., E.M. Hamzaoui and R.C. El Moursli, 2017. Application of nonnegative tensor factorization for neutron-gamma discrimination of Monte Carlo simulated fission chamber's output signals. *Results Phys.*, 7: 1422-1426.
- Li, X.L. and X.D. Zhang, 2006. Sequential blind extraction adopting second-order statistics. *IEEE. Signal Process. Lett.*, 14: 58-61.
- Meyer-Baese, U., C. Odom, G. Botella and A. Meyer-Baese, 2015. Independent component analysis algorithm FPGA design to perform real-time blind source separation. *Proceedings of the Conference on Independent Component Analyses, Compressive Sampling, Large Data Analyses (LDA), Neural Networks, Biosystems and Nanoengineering XIII*, Vol. 9496, May 20, 2015, International Society for Optics and Photonics, Baltimore, Maryland, p: 1-12.
- Min, Z., Z. Mu and M. Wenjie, 2012. Implementation of FastICA on DSP for blind source separation. *Procedia Eng.*, 29: 4228-4233.
- Nordhausen, K. and J. Virta, 2019. An overview of properties and extensions of FOBI. *Knowl. Based Syst.*, 173: 113-116.
- Novoy, M. and T. Adali, 2006. Adaptable nonlinearity for complex maximization of nongaussianity and a fixed-point algorithm. *Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, September 6-8, 2006, IEEE, Arlington, Virginia, USA., pp: 79-84.
- Pelegina, G.D., R. Attux and L.T. Duarte, 2019. Application of multi-objective optimization to blind source separation. *Expert Syst. Appl.*, 131: 60-70.
- Shyu, K.K., M.H. Lee, Y.T. Wu and P.L. Lee, 2008. Implementation of pipelined FastICA on FPGA for real-time blind source separation. *IEEE Trans. Neural Networks*, 19: 958-970.

- Singh, V., V.K. Somani and J. Manikandan, 2017. FPGA implementation of blind source separation using a novel ICA algorithm. Proceedings of the 2017 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia'17), October 5-7, 2017, IEEE, Bangalore, India, pp: 67-71.
- Texas Instruments, 2006. TMS320C67x/C67x+ DSP CPU and instruction set reference guide, November 2006. Literature Number: SPRU733A, Texas Instruments, Dallas, Texas, USA.
- Tong, L., R.W. Liu, V.C. Soon and Y.F. Huang, 1991. Indeterminacy and identifiability of blind identification. IEEE. Trans. Circuits Syst., 38: 499-509.
- Tornez-Xavier, G.M., L.M. Flores-Nava, F. Gomez-Castaneda and J.A. Moreno-Cadenas, 2015. FPGA implementation of the ICA algorithm using multiplexing. Proceedings of the 2015 12<sup>th</sup> International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE'15), October 28-30, 2015, IEEE, Mexico City, Mexico, pp: 1-4.
- Wang, C., J. Wang and T. Zhang, 2017. Operational modal analysis for slow linear time-varying structures based on moving window second order blind identification. Signal Proc., 133: 169-186.
- Yiu, K.F.C. and S.Y. Low, 2018. On a real-time blind signal separation noise reduction system. Int. J. Reconfigurable Comput., Vol. 2018, No. 4. 10.1155/2018/3721756