A Comparison of SOBI, FastICA, JADE and Infomax Algorithms

Guillermo SAHONERO-ALVAREZ
Laboratorio de Ingeniería en Computación, Universidad Católica Boliviana "San Pablo"
La Paz, Bolivia - g.sahonero@acad.ucb.edu.bo

Humberto CALDERON
Laboratorio de Ingeniería en Computación, Universidad Católica Boliviana "San Pablo"
La Paz, Bolivia - hcalderon@ucb.edu.bo

ABSTRACT

The Blind Source Separation (BSS) problem is present in a variety of engineering applications, including the electroencephalographic (EEG) signal separation for artifact removal procedure in Brain Computer Interfaces (BCI). Independent Component Analysis (ICA) is a widely used technique for this purpose, although, it is also well known to have high computational complexity. Thus, when 'real time' operation is the target on BCI systems, hardware solutions and specifically FPGA devices arise as feasible answers to the needs. However, when considering the design constraints and requirements, it is necessary to analyze which ICA algorithm would fit better in hardware. In this work, we evaluate different software implementations of ICA approaches using MATLAB and according to some criteria (running time, allocated memory, accuracy and scalability) targeting four ICA algorithms: Second-Order Blind Identification (SOBI), Hyvarinen's fixed-point algorithm (FastICA), logistic Infomax (Infomax) and Joint Approximation Diagonalization of Eigenmatrices (JADE). The outcomes have shown that SOBI's MATLAB implementation is the best procedure among all the analyzed techniques by drastically overcoming the speed of the others algorithms. Moreover, its correlation grades, corresponding to Pearson and Spearman correlation coefficients respectively, indicate it as one of the more accurate algorithms.

Keywords: Computational Resources Comparison, FPGAs, Hardware Acceleration, Independent Components Analysis, FastICA, Infomax, JADE, SOBI.

1. INTRODUCTION

Nowadays, human-machine interfaces have become more and more significant. Automation of assistive technologies is reaching unexpected boundaries, and it is foreseen that future machines will know what humans want them to do by using just the mind [1–3]. In this scenario, brain computer interfaces (BCI) are emerging as feasible solutions when thoughts and intentions are the only source of input information for a system. BCI's applications, by definition, may include wheel chair control or support movements for paraplegic people guided by brain signals, device's control like vehicles, cell phones, computers, and other applications which can even include gaming [4–8].

However, BCIs are required to provide a response with "no latency" and portability as the actions the user wants to be materialized must be performed on the way [2]. Unfortunately, software solutions cannot be easily considered a definite answer to the requirements, but hardware solutions (FPGA, SoC, ASIC), which can address those needs in a better way, may address better to them instead.

BCIs' system functionality includes a signal separation stage to be performed in order to later classify acquired signals and remove biological artifacts like: heartbeats (cardiac artifacts), eye blink/movements (ocular artifacts), muscle activity and noise from a set of signals extracted with some technique like the electroencephalography (EEG), magneto-encephalography (MEG) or others [9], [10]. Many works state that by applying ICA we can achieve this stage.

As real time human-machine interfaces based on BCI systems require being fast and accurate, the signal separation stage must present these attributes in its implementation, as a primary objective, too. Therefore, both the Academia and the industry are continuously researching on new ways to reach this goal. Furthermore, highly customized hardware accelerators implementation might be the answer to this situation, but hardware design also needs to meet some constraints and requirements related to memory management, time execution, numerical accuracy and even scalability.

ICA can be performed by a variety of algorithms (see [11–16] for more information), from which we have observed that four are commonly used in brain signal separation. These are: Second Order Blind Identification (SOBI)[17], Hyvarinen's fixed point algorithm (FastICA)[18], Infomax [19] and Joint Approximation Diagonalization of Eigenmatrices (JADE)[20]. With the aim of implement one of them on a hardware platform and accelerate the process, an algorithm evaluation is needed in order to determine the appropriate algorithm implementation to be studied and then, implemented. In this sense, we have evaluated these four ICA algorithms according to hardware design criteria and signal separation accuracy and scalability using MATLAB.

The rest of the paper is organized as follows. Section II outlines the theoretical background. Section III states the Comparison Methodology and Tools used for the main purpose. Section IV presents the Experimental Results and Analysis. Finally, the article is concluded in section V.

2. BACKGROUND

ICA is a statistical technique that solves the BSS problem. This computational method is used to separate a set of linearly mixed multivariate signals and transform it into another set which components are approximately the original signals and independent between them [18], [21].
The core of ICA functionality relies on statistical independence.

The approaches to reach it use entropy measures by assuming non-gaussianity, and even joint diagonalization of correlation or covariance matrices; this latter is generally used when it is assumed to exist time-correlation structure. In fact, as different signals nature assumptions are stated, different algorithms can also be proposed. These signals nature assumptions were studied previously by Cardoso in [22] defining then some manifolds over the ICA techniques work.

Mathematical Model

Classical ICA model is defined by the following expression:

$$\mathrm{x(t) = A\,s(t)} \qquad (1)$$

Where, $\mathrm{x(t)}$ is the received signals vector, A the mixing matrix and $\mathrm{s(t)}$ the original sources vector.
Most ICA algorithms focus on computing an un-mixing matrix W which can be seen as the approximate inverse of A. Therefore, the ICA model turns into:

$$\mathrm{Wx(t) = y(t)} \qquad (2)$$

Being $\mathrm{y(t)}$ the approximate original signals, i.e. $\mathrm{y(t) \approx s(t)}$.
On the other hand, many algorithms, if not all of them, use a pre-processing step known as whitening [17], [18], [20]. In this case, the whitened signals ($\mathrm{z(t)}$) are computed by multiplying the received signals by a whitening matrix B, that is:

$$\mathrm{z(t) = Bx(t)} \qquad (3)$$

The reader must note that all of these expressions can be generalized to a matrix form. Actually, most ICA algorithms implementations perform their operations on a received signals matrix instead of a vector.

Algorithms

SOBI - Second Order Blind Identification: Proposed by Belouchrani et al. in [17], this algorithm relies on second-order statistics to explode the time-correlation structure assumption of the signals. It requires computing the following steps:

1. Whitening
2. Computation of Lagged Correlation Matrices
3. Joint Diagonalization (JD)

The main concept of SOBI is the assumption about the diagonal form of the lagged correlation matrices, which stands according to the following expression:

$$\begin{aligned}\mathrm{R_x(\tau)} &= \mathrm{E\{x(t)x(t+\tau)^T\}} \\ &= \mathrm{A\,R_s(\tau)A^T, \forall\tau}\end{aligned} \qquad (4)$$

Where $\mathrm{R_s}$ is the correlation matrix of the sources signals and $\mathrm{R_x}$, the lagged correlation matrix.
Considering that eq. (4) holds for all values of τ, there exists a unitary matrix $\mathrm{U = BA}$ that jointly diagonalises all the correlation matrices:

$$\mathrm{U^T R_x(\tau) U = R_s(\tau)} \qquad (5)$$

Therefore, the approximate original signals are computed by $\mathrm{U^{\#}Bx(t)}$.

FastICA - Hyvärinen's Fixed Point Algorithm: Hyvärinen's algorithm is often used in 'real time' applications because of the possible parallel implementation [23], [24]. This algorithm converges quickly as it seeks for a component one by one.

FastICA uses kurtosis for the independent components estimation [25]. Whitening is usually performed on data before the execution of the algorithm [18].

The following procedure performs FastICA:

1. Initialize $\mathrm{w_i}$ (in random)
2. $\mathrm{w_i^+ = E\left(\emptyset'\left(w_i^T X\right)\right) w_i - E\left(x\,\emptyset\left(w_i^T X\right)\right)}$
3. $\mathrm{w_i = \frac{w_i^+}{\|w_i^+\|}}$
4. For $\mathrm{i = 1}$, go to step 7. Else, continue with step 5.
5. $\mathrm{w_i^+ = w_i - \sum_{j=1}^{i-1} w_i^T w_j w_j}$
6. $\mathrm{w_i = \frac{w_i^+}{\|w_i^+\|}}$
7. If not converged, go back to step 2. Else go back to step 1 with $\mathrm{i = i + 1}$ until all components are extracted.

Infomax: This algorithm is based on the maximization of entropy and presents a natural gradient form for the independent components computation. It can be thought as a neural learning method as the mathematical formulation stands for the next expression:

$$\mathrm{W(t+1) = W(t) + \eta(t)(I - f(s)s^T)W(t)} \qquad (6)$$

Being $\mathrm{\eta(t)}$ a learning-rate function and $\mathrm{f(\cdot)}$ a function related to the distribution nature (i.e. super Gaussian or sub Gaussian). It is important to note that the initial value of W is usually a random matrix [25]. More information about the Infomax procedures can be found in [18], [25].

JADE - Joint Approximation Diagonalization of Eigenmatrices: Besides the previous approaches of kurtosis and entropy, JADE, as SOBI does, uses JD and whitening. However, the main difference between both is the set of target matrices on which JD is done. JADE performs it on $\mathrm{\frac{n}{2}(n+1)}$ eigenmatrices that are computed by the fourth order cumulants of whitened signals:

$$\mathrm{[Q^z(e_p e_q^H]_{ij} = Cum(z_i, z_j, z_p, z_q)} \qquad (7)$$

An extended explanation of JADE can be found in [20], [26].

3. COMPARISON METHODOLOGY AND TOOLS

Comparison Criteria
We compare algorithms using criteria based on important requirements and limitations that a future hardware implementation may present. These criteria were:

Running Time (latency): Defined as the amount of time passed from the initiation of the algorithm sequence to the retrieval of the mixing matrix, this benchmark aims to measure the computational load of the algorithm. As the time amount can be variable because of the computer capacity (processor frequency, RAM, etc.), results have been transformed into a percent form showing how "slow" an algorithm is in relation to the faster one.

Allocated Memory: Each algorithm uses specific functions for its implementation; some of them may allocate more memory than others. This indicator shows the approximate amount of allocated memory for the whole algorithm. However, memory management might be optimized in further hardware implementation stage.

Accuracy and Scalability: This parameter aims to measure the quality of the performed process. Although accuracy can be biased by many other factors, e.g. numerical stability, numerical precision, etc., we considered enough to compute the statistical correlation grade between original and approximated signals.

Nevertheless, it is also important to know if the algorithms preserve their accuracy as the number of components to be computed increase. For that purpose, we executed multiple tests in order to find a pattern in the accuracy tendency of the algorithms.

Comparison Procedure and used Tools

Software and Datasets: We used MATLAB as the comparison platform and the following toolboxes, which are standard in the research community, to run the tests:

a. EEG Lab toolbox, which includes the SOBI, Infomax and JADE codes[27].
b. The FastICA package which includes the FastICA code[28].

On the other hand, for running time and allocated memory measurements we used built-in MATLAB functions.

As for the EEG datasets used in this work, they were gotten from P300 EEG available data of the Multimedia Signal Processing Group of the École Polytechnique Féderale de Lausanne[29]. These data is the same used to produce the results of "An efficient P300-based brain-computer interface for disabled subjects"[30]. However, although there are eight data sets corresponding to eight different subjects, we only used data corresponding to the subject one and six; reasons will be exposed later.
As we cannot know the original sources of acquired brain signals and artifacts with fully precision, besides the EEG datasets, we used the free Macaulay Library sound samples in order to compute the accuracy and scalability of the algorithms. From all the available sound packs, we only used three; these are Bird Songs of Florida, Unexpected Voices of the Wild and Voices of Eastern Backyard Birds. From them, the first 32 samples were employed.

Accuracy-Scalability Indicator Computation: For this criterion, the Spearman and Pearson correlation coefficients were used. As the Spearman coefficient indicates the subtle strength of the association between two variables according to an arbitrary monotonic function, the Pearson coefficient points out the grade of linear correlation between those variables[31]. In the first case, the more similar the variables' shapes are, the closer its value to 1 or -1; for the latter one, the more linearly related the variables are, the closer its value to 1 or -1. As we are interested in the signals shape and the linear correlation between them, the signs of such signals are not considered and the absolute value of the coefficients were used instead; this is also a consequence of ICA's sign ambiguity.

However, since ICA cannot obtain the independent components orderly, it was necessary to develop a function to get paired an approximated and an original signal by using both correlation coefficients. In that sense, two signals or variables are said to be a pair if their Spearman or Pearson coefficient is maximum among the set of computed coefficients. On the other hand, in order to determine the scalability properties of the algorithms, we performed 31 tests of each algorithm. Every test had a different increasing number of sources, thus, the test 1 had two components, the test 2 had three and finally, the test 31 had 32 components.

Comparison Procedure: The completely available data in the Multimedia Signal Processing Group of the École Polytechnique Fedérale de Lausanne includes eight subjects, each with four recording sessions, which consists of six images. This makes 24 samples to be analyzed per subject.

As analyzing the eight subjects' data involves processing 192 samples each of 100000 data points and more than 30 channels, we considered enough to process just the data from two subjects, subject 1 and 6. This because of the data signals nature, subjects 1-4 did present a disability and subjects 6-9 did not (subject 5 data were not available online by the time this research has been done). Then, by processing them, we could afford an approximation to different scenarios: when brain signals separation is performed in subjects with some disability and with healthy subjects. Although the aim of this work does not imply the study of brain signals' difference between subjects with and without some disability, we considered interesting to use both kind of datasets as an exploration of our analysis.

Later, we developed a MATLAB function in order to structure the running time and memory tests of each sample. This function returned the time between the beginning and the end of each algorithm and the amount of allocated memory. For the running time tests, the MATLAB function *tic-toc* was used. On the other hand, for the allocated memory tests, a profile was created for every running instance of each algorithm. We must say that no time tracker was implemented inside the profile procedure and MATLAB was the only application running in the foreground.

Finally, for the algorithm's accuracy determination: the 32 sound samples were mixed by a uniformly distributed random matrix. The ICA algorithms processed this mixed data matrix and then, each computed component got paired with one original signal source. The Spearman and Pearson coefficients of such pair were respectively considered with the others to compute the average of the test. We show the results and discuss them in the following section.
We computed the differences between running time of algorithms (shown in the Table II) by using average time and performing:

$$\%\mathrm{Dif} = 100\left(\frac{t_x}{\min(t_{\mathrm{FastICA}}, t_{\mathrm{Infomax}}, t_{\mathrm{SOBI}}, t_{\mathrm{JADE}})} - 1\right) \tag{8}$$

On the other hand, we estimated differences between correlation grade coefficients (shown later in Table III) using:

$$\%\mathrm{Dif} = 100\left|\frac{c_x}{\max(c_{\mathrm{FastICA}}, c_{\mathrm{Infomax}}, c_{\mathrm{SOBI}}, c_{\mathrm{JADE}})} - 1\right| \tag{9}$$

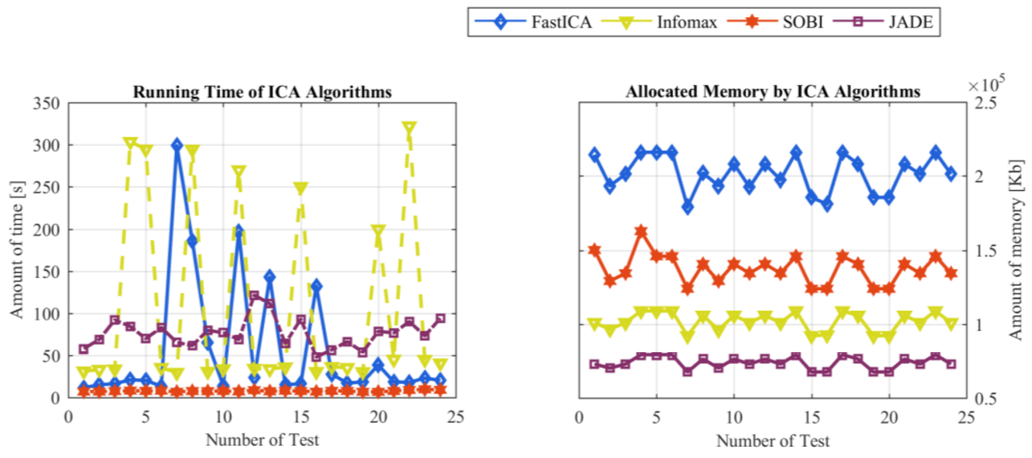

Fig. 1. Running Time and Allocated Memory Results of Subject 1

The computer used for the experiments has the following features:

Table 1: Computer Features

Processor Freq.	Processor Family	MATLAB Version	O.S.	RAM
2.4 GHz	Intel Core i5	R2015a	Window s 7	8GB

4. EXPERIMENTAL RESULTS AND DISCUSSION

Each subject's results have been plotted to carefully observe the behavior of the algorithm given the 24 different samples. We categorized the performed tests in two: subject one and subject six. Nevertheless, we found few differences between them, suggesting that there is no remarkable difference on both subjects' brain signals nature. Results achieved by each algorithm were grouped in only one plot for readability.

By observing the running time results (Fig. 1, 2 and Table 2), it is clear that the faster algorithm in both subject's tests is SOBI. However, it also shows stability and relatively moderate use of memory. We think that this result is a consequence of a good implementation that might guide properly to the HW design. Next to SOBI, we found the second faster algorithm: FastICA, but, there are some instabilities in the time running tendency, which can be a problem while trying to achieve 'real-time' processing. The possibility of a wrong performed implementation is discarded by observing the allocated memory of FastICA, which shows relative stability. We considered the random nature of FastICA initial values as the reason of this instability.

Table 2: Running Time of Algorithms

Algorithm	Average Time	Worst Time	Best Time	%Dif.
FastICA	51.91 [s]	298.68[s]	11.82 [s]	503.87
Infomax	73.73 [s]	321.89 [s]	25.79 [s]	757.70
SOBI	8.60 [s]	12.28 [s]	6.86 [s]	0
JADE	78.20 [s]	162.86[s]	48.26 [s]	809.65

Further, the Infomax algorithm also shows instability in the running time tendency though it has stability in the allocated memory as FastICA. Like in the previous algorithm, we attribute this instability to the random nature of initial values.
Finally, JADE shows interesting properties, as the allocated memory is stable and smaller. However, running time results suggests that the procedure represents a high computational load. We infer that the stability property, which can be observed on JADE and SOBI behavior, relies on the way that both were theoretically formulated; both of them use the Joint Diagonalization as a final step and the whitening procedure as a pre-processing stage. The only difference between them, as we stated before in the Background section, is the nature of the matrices to be diagonalized. As JADE computes the

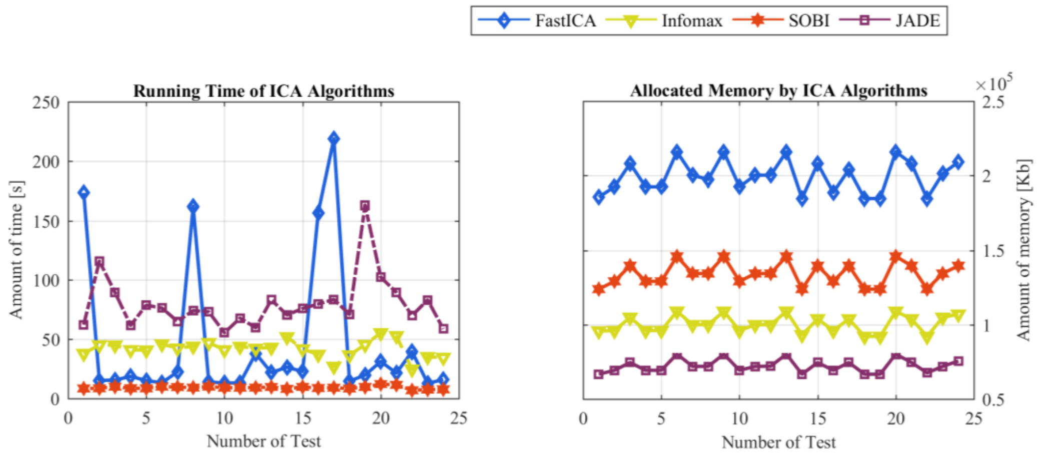

Fig. 2. Running Time and Allocated Memory Results of Subject 6

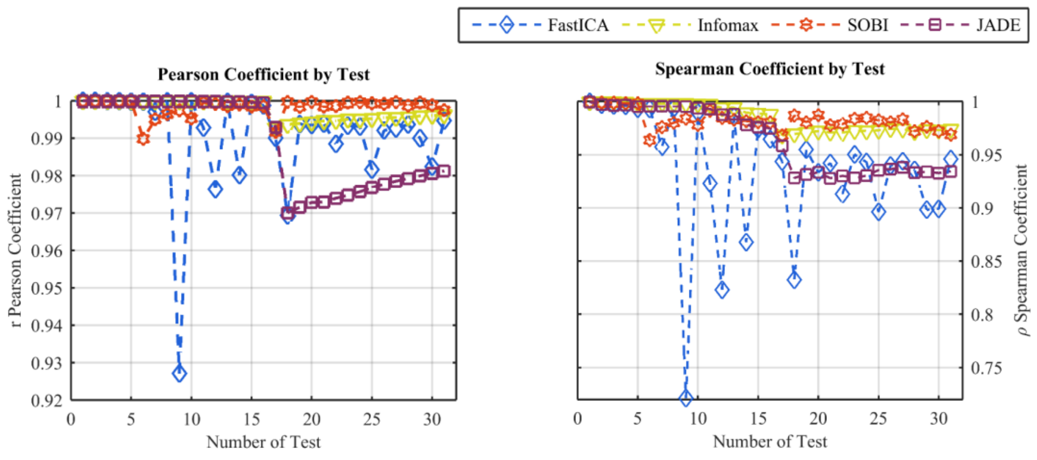

Fig. 3. Spearman and Pearson Coefficients of each Algorithm by Test.

eigenmatrices based on fourth order cumulants, SOBI only computes correlation matrices based on second order statistics. We also think of this feature as the cause of having different results in the running time tests. Such difference may due to the way of computing the eigenmatrices, which is more expensive in processor use than computing simple correlation matrices.

On the general overview of Allocated Memory criteria, we did not appreciate critical differences between the subject 1 and 6 tests. Algorithms have shown an interesting behavior instead. FastICA always allocates more memory than the others and JADE allocates the least amount of memory. SOBI and Infomax remain in the middle of the previous two algorithms.

Referring to the Spearman and Pearson correlation coefficients, all of the studied algorithms have practically shown good performance. Nevertheless, as we can see in Fig. 3 and Table 3, SOBI is the best-rated algorithm considering the Pearson correlation grade and Infomax according to the Spearman correlation.

We noted that FastICA shows some instability related to both tests; the reasons of that behavior must be studied in further works. On the other hand, the algorithm that shows more stability across the number of tests is SOBI; the other techniques show a more pronounced decreasing tendency in both coefficients.

Table 3: Pearson and Spearman Average Coefficients

Algorithm	Pearson Av.	%Dif.	Spearman Av.	%Dif.
FastICA	0.9907	0.7516	0.9369	4.8224
Infomax	0.9975	0.0743	0.9844	0
SOBI	0.9982	0	0.9825	0.1926
JADE	0.9889	0.9355	0.9635	2.1236

5. CONCLUSIONS AND FUTURE WORK

Although FastICA is usually taken as a fast algorithm, we have observed that this may depend on adequate initial values of the procedure. Despite that, a parallel configuration of the technique might be optimal in order to reduce the amount of running time. On the other hand, memory used by FastICA seems to be higher than the other algorithms but this may be outweighed by the achieved acceleration rate if parallelism is implemented.

Furthermore, Infomax results suggest that its implementation may not be a good idea, as parallelism is not easily achieved by its mathematical definition, and because of its running time tendency, which appears to be unpredictable.

We consider that SOBI achieved the best results in the performed tests. Its running time and the allocated memory tendency show stability as critical oscillations are not present in the results. The utilized implementation might be seen as an optimal way to perform SOBI and thus, a guide to implement this algorithm into some hardware platform.

Regardless of the speed disadvantage of JADE, this algorithm allocated less memory than the others and this fact must be explored in future research. An adequate parallelization of JADE algorithm would guide to a faster implementation with a relatively good accuracy and low memory consumption. This is especially desired when embedded systems are foreseen to be constrained in memory.

Although the used tools (MATLAB, general-purpose computer, ICA toolboxes) might be considered as subjective given the unknown real influence of the operating system and even the way in which algorithms were implemented, we consider that results provide a general overview to determine which algorithm performs better. This is because these commonly used MATLAB ICA toolboxes can be utilized as guides to design in a preliminary way the required hardware accelerators.

In that sense, and despite of the fact that FPGA devices have a lot of room to implement systems with space parallelism [32], a quick overview of the work-chain of SOBI reveals that the algorithm cannot be easily parallelized. We think that these technologies should be used to accelerate inner processes of the algorithm instead. SoC which could afford and optimize the vast linear algebra routines of SOBI might be the best prospect.
Overall, we think of the achieved accuracy of all the algorithms as good. Nevertheless, all of them have shown a decreasing tendency in the coefficients. Future works may study the limits of such behavior in order to determine the outliers of a satisfactory algorithm's performance.

6. REFERENCES

[1] C. Brunner, N. Birbaumer, B. Blankertz, C. Guger, A. Kübler, D. Mattia, del R. M. José, F. Miralles, A. Nijholt, E. Opisso, N. Ramsey, P. Salomon, and M.-P. Gernot R., "**BNCI Horizon 2020: towards a road map for the BCI community**.," in *Brain Computer Interfaces*, C.

Stephanidis and M. Antona, Eds. Springer International Publishing, 2014, pp. 457–486.

[2] L. F. Nicolas-Alonso and J. Gomez-Gil, "**Brain Computer Interfaces, a Review**," *Sensors*, vol. 12, pp. 1277–1279, 2012.

[3] R. Rupp, S. C. Kleih, R. Leeb, J. del R. Millan, A. Kübler, and G. R. Müller-Putz, "**Brain–Computer Interfaces and Assistive Technology**," in *Brain-Computer-Interfaces in their ethical, social and cultural contexts*, 1st ed., vol. 12, Springer Netherlands, 2014, pp. 7–38.

[4] H. H. Alwasiti, I. Aris, and A. Jantan, "**Brain Computer Interface Design and Applications: Challenges and Future**," *World Applied Sciences Journal*, vol. 11, no. 7, pp. 819–825, 2010.

[5] D. P.-O. Bos, B. Reuderink, B. van de Laar, H. Gürkök, C. Mühl, M. Poel, A. Nijholt, and and Dirk Heylen, "**Brain-Computer Interfacing and Games**," in *Brain-Computer Interfaces*, Springer Verlag, 2010, pp. 149–178.

[6] D. Göhring, D. Latotzky, M. Wang, and R. Rojas, "**Semi-Autonomous Car Control Using Brain Computer Interfaces.** Volume 2 Proceedings of the 12th International Conference IAS-12.," in *Intelligent Autonomous Systems 12*, Springer Berlin Heidelberg, 2013, pp. 393–408.

[7] J. R. Wolpaw and N. Birbaumer, "**Brain-computer interfaces for communication and control**," *Clinical Physiology*, vol. 113, no. 6, pp. 767–791, Jun. 2002.

[8] W. Yu-te, Y. Wang, and T.-P. Jung, "**A Cell-Phone Based Brain-Computer Interface for Communication in Daily Life.** International Conference, AICI 2010, Sanya, China. October 23-24, 2010. Proceedings, Part II," in *Artificial Intelligence and Computational Intelligence*, Springer Berlin Heidelberg, 2010, pp. 233–240.

[9] I. R. Keck, V. Fischer, A. M. Tomé, C. G. Puntonet, and E. W. Lang, "**Finding Gold in the Dirt - Biomedical Applications in the Light of ICA**," in *Recent Advances in Biomedical Signal Processing*, J. M. Górriz, E. W. Lang, and J. Ramirez, Eds. Bentham Science Publishers, 2011, pp. 149–156.

[10] B. W. McMenamin, A. J. Shackman, L. L. Greischar, and R. J. Davidson, "**Electromyogenic Artifacts and Electroencephalographic Inferences Revisited**," *Neuroimage*, vol. 54, no. 1, pp. 4–9, Jan. 2011.

[11] M. Crespo-Garcia, M. Atienza, and J. L. Cantero, "**Muscle Artifact Removal from Human Sleep EEG by Using Independent Component Analysis**," *Annals of Biomedical Engineering*, vol. 36, no. 3, pp. 467–475, Mar. 2008.

[12] A. Delorme, J. Plamer, R. Oostenveld, J. Onton, and S. Makeig., "**Comparing results of algorithms implementing blind source separation of EEG data**.," *Swartz Foundation and NIH Grant*, 2007.

[13] A. Delorme, T. Sejnowski, and S. Makeig, "**Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis**," *Neuroimage*, vol. 34, no. 4, pp. 1443–1449, 2007.

[14] A. Kachenoura, L. Albera, L. Senhadji, and P. Comon, "**ICA: a potential tool for BCI systems**," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 57–68, 2006.

[15] V. Matic, W. Deburchgraeve, and S. Van Huffel, "**Comparison of ICA algorithms for ECG artifact removal from EEG signals**," in *Proc. of the 4th Annual symposium of the IEEE-EMBS Benelux Chapter.(IEEE-EMBS)*, 2009, pp. 137–140.

[16] M. Naeem, C. Brunner, and G. Pfurtscheller, "**Dimensionality Reduction and Channel Selection of Motor Imagery Electroencephalographic Data**," *Computational Intelligence and Neuroscience*, vol. 2009, p. 8, 2009.

[17] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "**A Blind Source Separation Technique Using Second-Order Statistics**," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.

[18] A. Hyvärinen and E. Oja, "**Independent Component Analysis: Algorithms and Applications**," *Neural Networks*, vol. 13, no. 4–5, pp. 411–430, 1999.

[19] S. Amari, A. Cichocki, and H. H. Yang, "**A New Learning Algorithm for Blind Signal Separation**," in *Advances in Neural Information Processing Systems*, 1996, pp. 757–763.

[20] J.-F. Cardoso and A. Souloumiac, "**Blind Beamforming for non Gaussian Signals**," *IEEE Proceedings-F*, vol. 140, pp. 362–370, 1993.

[21] J. V. Stone, "**Independent component analysis: an introduction**," *TRENDS in Cognitive Sciences*, vol. 6, no. 2, pp. 59–64, 2002.

[22] J.-F. Cardoso, "**The Three Easy Routes to Independent Component Analysis; Contrasts and Geometry**," *Proc. ICA 2001*, Dec. 2001.

[23] S. K. Behera, "**FastICA for Blind Source Separation and Its Implementation**," National Institute of Technology Rourkela - Department of Electronics and Communication Engineering, 2009.

[24] A.-L. Taha, "**FPGA Implementation of Blind Source Separation using FastICA,**" The University of Windsor - Department of Electrical and Computer Engineering, 2010.

[25] D. Langlois, S. Chartier, and D. Gosselin, "**An Introduction to Independent Component Analysis: InfoMax and FastICA algorithms**," *Tutorials in Quantitative Methods for Psychology*, vol. 6, no. 1, pp. 31–38, 2010.

[26] J.-F. Cardoso, "**High-order Contrasts for Independent Component Analysis**," *Neural Comput.*, vol. 11, no. 1, pp. 157–192, Jan. 1999.

[27] A. Delorme and S. Makeig, "**EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics**," *Journal of Neuroscience Methods*, vol. 134, pp. 9–21, 2004.

[28] D. of Information and C. S.-A. University, "**The FastICA package for MATLAB** - Web Page: research.ics.aalto.fi/ica/fastica/." Jun-2015.

[29] M. S. P. Group, "**École Polytechnique Federale de Lausanne - MMSPG. BCI Datasets**. Web-Page: mmspg.epfl.ch/BCI_datasets." Jun-2015.

[30] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, and K. Diserens, "**An efficient P300-based brain–computer interface for disabled subjects**," *Journal of Neuroscience methods*, vol. 167, no. 1, pp. 115–125, 2008.

[31] J. Hauke and T. Kossowski, "**Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data**," *Quaestiones Geographicae*, vol. 30, no. 2, pp. 87–93, 2011.

[32] D. R. H. Calderón, "**Arithmetic Soft-Core Accelerators**," Delft University of Technology, Delft, Netherlands, 2007.