HARDWARE-ORIENTED MEMORY-LIMITED ONLINE FASTICA ALGORITHM AND HARDWARE ARCHITECTURE FOR SIGNAL SEPARATION

Lan-Da Van, *Senior Member*, *IEEE*, Tsung-Che Lu, and Tzyy-Ping Jung*, *Fellow, IEEE*, Jo-Fu Wang

Dept. Computer Science, National Chiao Tung University, Hsinchu, Taiwan. ldvan@cs.nctu.edu.tw ; tclu@cs.nctu.edu.tw
*Swartz Center for Computational Neuroscience, University of California, San Diego (UCSD), USA. jung@sccn.ucsd.edu

ABSTRACT

This paper presents a hardware-oriented memory-limited online FastICA algorithm and its hardware architecture and implementation for eight-channel electroencephalogram (EEG) signal separation. The online algorithm integrates the data overlapping, garbage detection, channel permutation, and momentum-controlled weight update schemes to stabilize the order of the decomposed source signals across time. This study also realizes the algorithm into a hardware architecture and implementation with a core area of 1.469x1.469 mm^2 in a TSMC 90 nm process. The resulting power dissipation for eight-channel EEG signal separation is 65 mW@100 MHz at 1V.

Index Terms**—** Blind signal separation, EEG, component/ channel switch, fast independent component analysis (FastICA), and hardware implementation.

1. INTRODUCTION

Electroencephalogram (EEG) has been widely applied to the research to understand human cognition and clinical applications. The true neural activities can be obtained by the principal component analysis (PCA) and independent component analysis (ICA) [1-12]. The goal of ICA is to separate the source signals from the mixed signals measured by the scalp EEG. Thus, it is possible to remove eye blink artifacts and noise originated from non-cerebral activities of the EEG signals. That means ICA can separate the useful sources from the artifact-contaminated EEG recordings.

Most of the ICA algorithms use a PC or server to perform offline analysis to separate brain and non-brain source signals. However, using offline analysis, we have to wait until the end of the EEG recording to evaluate task-related EEG dynamics. As a result, we cannot judge the activity immediately while the EEG recording is undergoing. Several ASIC or FPGA implementations of the ICA algorithms have been proposed [13-24] to accelerate the ICA processing. Van *et al.* [18, 22] proposed two FastICA-based hardware architectures and implementations in an ASIC approach for 8 and 2-16 channels, respectively. One of the practical problems with these hardware implementations is that the independent components (ICs) (i.e. the output channels) of the FastICA switch from one data window to the next due to the property of FastICA and the limited memory size in hardware implementation. Thus, it is hard to track the source activities over data windows. This study aims to solve this practical and important component-switching problem to attain more stable component or source activities. We will detail the proposed online FastICA algorithm that integrates the four schemes and its hardware implementation [19]. First, each data window contains overlapped data to stabilize the component/channel order. Second, the standard deviation is used to distinguish whether the data are feasible, namely garbage detection scheme. Third, the channel permutation mechanism is used to discriminate the relation between ICs generated by the original FastICA algorithm from successive data windows. Finally, the momentum-controlled weight matrix is updated. The contributions of this study are as follows. 1) Present the hardware-oriented memory-limited online FastICA algorithm featuring four schemes to resolve the component-switching problem of FastICA, 2) Propose the corresponding hardware architecture and implementation using four computing units (CUs) in a TSMC 90 nm process.

2. PROBLEM OBSERVATION AND PROPOSED ONLINE FASTICA ALGORITHM

In the ICA research field, the signal-separation process is modeled as

$$\mathbf{X} = \mathbf{AS} \tag{1}$$

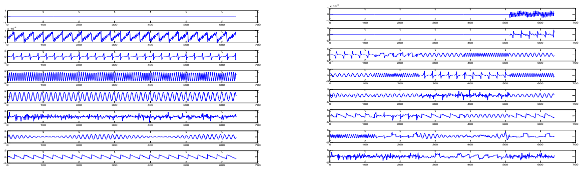
where **X** and **S** denote the *n*x*m* observed signal matrix and the source signal matrix with the vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and the vectors $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$, respectively, and **A** denotes an *n*x*n* mixing matrix with elements a_{ij}. Note that **A** and **S** are both unknown and we have to estimate both **A** and **S** using **X**. FastICA [1, 2, 4] applies the maximum non-gaussianity estimation, so the ICs can be obtained. In other words, after estimating $\mathbf{W}^T$ (i.e., inverse of the matrix **A**), the ICs can be written as

$$\mathbf{S} = \mathbf{W}^T\mathbf{X} \tag{2}$$

The FastICA algorithm consists of two parts. The first is the preprocessing of FastICA used to center and whiten the mixed signals. The second part is the fixed-point iteration algorithm used to obtain the updated weight by the negentropy approximation [1, 2, 4]. Herein, due to the space limitation, the detailed equations will be only shown to explain how to realize them in hardware in Section 3.

2.1 Stability of the FastICA Algorithm for the Limited Memory Size and Online FastICA Algorithm

Fig. 2.1(a) shows the FastICA processing results of the mixed signals with a data window length of 640. Fig. 2.1(b) shows the simulation results by using the original FastICA algorithm with a window length of 128. As can be seen, the stability of the channel order of FastICA across data windows is not guaranteed, making the EEG interpretation and monitoring difficult. Therefore, we propose an online FastICA, as described in Fig. 2.2 and Sections 2.2-2.5, for hardware-oriented memory-limited implementation.



(a) (b)

Fig. 2.1. (a) Offline FastICA results and (b) referenced online FastICA results.

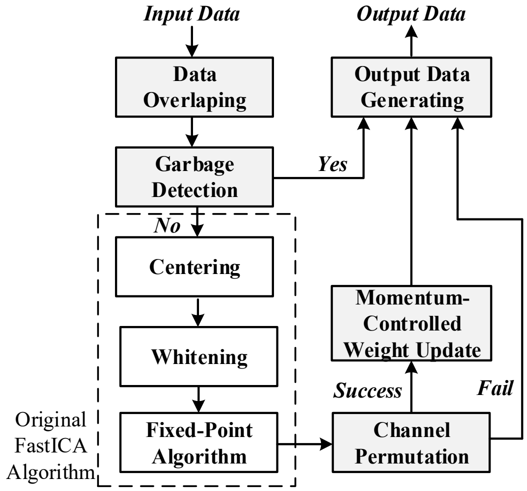
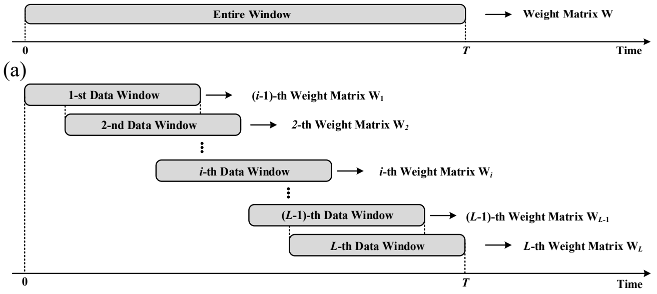


Fig. 2.2. Flowchart of the online FastICA algorithm.

2.2 Data Overlapping

To guarantee a stable IC order across adjacent data windows, we use overlapped data, depicted in Fig. 2.3.



(b)

Fig. 2.3. (a) Offline FastICA processing and (b) online FastICA processing utilizing the overlapped data window.

2.3 Garbage Detection

Similar to other biomedical signal processing algorithms, the ICA algorithm holds the "garbage in, garbage out" (GIGO) rule [25, 26]. If the input data contains useless data, the training result of the FastICA algorithm may result in useless ICs. Therefore, the input data should be examined by the garbage detection procedure for each data window before training FastICA. Eq. (3) is utilized for the garbage detection.

$$\max\{|\bar{x}_i(1)|,\ |\bar{x}_i(2)|,\ \ldots,|\bar{x}_i(m)|\} < k\sigma_{\bar{x}_i} \text{ for } i = 1, 2, \ldots, n, \quad (3)$$

where $\bar{x}$, k and σ denote centering data, a constant and the standard deviation as in (4), respectively.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}} \quad (4)$$

If (3) is not satisfied, the data in the i-th data window will be regarded as garbage and the update procedure for the weight matrix will be terminated in Fig. 2.2. Otherwise, we have to calculate the new weight.

2.4 Channel Permutation

A correlation-based permutation mechanism in Fig. 2.4 is used. The permutation scheme concept is similar to that of [7]. To determine the relationship between the time course of ICs generated by the original FastICA algorithm from the successive data windows, the absolute correlation matrix **R** is calculated. **R** is defined as

$$\mathbf{R} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ r_{n,1} & r_{n,2} & \cdots & r_{n,n} \end{bmatrix} \quad (5)$$

where $r_{p,q}$ denotes the correlation coefficients between the ICs of the overlapped data in Fig. 2.4. Using **R**, the permutation matrix for the i-th data window, **U**, can be acquired as

$$\mathbf{U} = \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ u_{n,1} & u_{n,2} & \cdots & u_{n,n} \end{bmatrix} \quad (6)$$

where

$$u_{i,j} = \begin{cases} 1, & \text{if } |r_{i,j}| = \max\{|r_{i,1}|,\ |r_{i,2}|,\ \ldots,|r_{i,n}|\} \text{ and } r_{i,j} \geq 0 \\ -1, & \text{if } |r_{i,j}| = \max\{|r_{i,1}|,\ |r_{i,2}|,\ \ldots,|r_{i,n}|\} \text{ and } r_{i,j} < 0 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Before applying **U** for permutation, we have to determine whether the relationship belongs to one-to-one mapping or many-to-one mapping. If the relationship belongs to many-to-one mapping, the permutation will fail since there is at least one IC with the (i-1)-th data window mapped to several ICs with the i-th data window. To ensure the one-to-one mapping relationship, Eq. (8) has to be satisfied.

$$\sum_{i=1}^{n} |u_{i,j}| = 1 \quad \text{for } j = 1,\ 2,\ \ldots, n. \quad (8)$$

If failure occurs in the channel permutation in Fig. 2.2, the previous weight will be adopted.

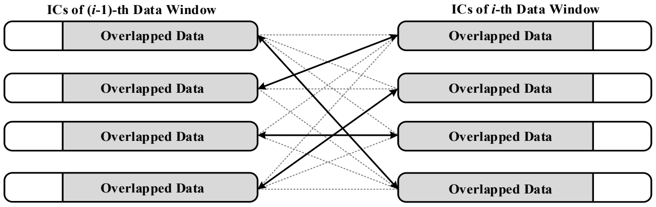


Fig. 2.4. Compute correlations of ICs across data windows.

2.5 Momentum-Controlled Weight Update

Once the channel permutation with success is performed, we add a momentum coefficient, r, to update **W** at each processing. We expect the results of the proposed algorithm will close to the offline results. Finally, the update formula is expressed in (9). We use adjacent **W**'s to update the overall weight matrix **W**.

$$\mathbf{W}_0 = \left[\mathbf{W}_{0,old}/\left\|\mathbf{W}_{0,old}\right\|_\infty\right]\cdot(1-r) + [\mathbf{W}/\|\mathbf{W}\|_\infty]\cdot r \qquad (9)$$

where $\mathbf{W}_0$ is the updated weight matrix at current data window, $\mathbf{W}_{0,old}$ is the weight matrix at previous data window, $\mathbf{W}$ is the weight matrix calculated by the original FastICA algorithm at the current data window, and momentum coefficient r can be set by the user.

3. HARWARE ARCHITECTURE

Fig. 3.1 shows the block diagram of an online FastICA hardware architecture for EEG signal processing. Fig. 3.2 details the computing units (CUs) in Fig. 3.1. The CU uses IEEE-754 floating-point computation and the sample size is 1024. First, the input data are transformed to the floating point through the Fixed-to-Floating Converter and stored in the data memory (DM). Second, to enhance the stability of ICs between adjacent data windows, we use overlapped data stored in the DM across windows, advancing at a step size of 64 data points. Third, data are fetched from the DM to perform centering in (10).

$$\bar{x}(i) = x(i) - E\{x\} = x(i) - \left(\sum_{j=1}^{1024} x(j)\right) >> 10 \qquad (10)$$

Note that the centering results can be reused by garbage detection and centering operation of the original FastICA. Fourth, the standard deviation in (4) of each row can be calculated via CU1. Then, the processed data is written back to DM. If (3) does not meet the threshold, we will ignore this calculation and use the previous weight as the calculated weight. Otherwise, we have to calculate the new weight. Fifth, data are fetched from the DM to calculate covariance in (11) by CUs.

$$\mathbf{C_X} = E\{\bar{\mathbf{X}}\bar{\mathbf{X}}^T\} = \frac{1}{1024}\begin{bmatrix} \bar{\mathbf{x}}_1^T\bar{\mathbf{x}}_1 & \bar{\mathbf{x}}_1^T\bar{\mathbf{x}}_2 & \cdots & \bar{\mathbf{x}}_1^T\bar{\mathbf{x}}_8 \\ \bar{\mathbf{x}}_2^T\bar{\mathbf{x}}_1 & \bar{\mathbf{x}}_2^T\bar{\mathbf{x}}_2 & \cdots & \bar{\mathbf{x}}_2^T\bar{\mathbf{x}}_8 \\ \vdots & \vdots & \cdots & \vdots \\ \bar{\mathbf{x}}_8^T\bar{\mathbf{x}}_1 & \bar{\mathbf{x}}_8^T\bar{\mathbf{x}}_2 & \cdots & \bar{\mathbf{x}}_8^T\bar{\mathbf{x}}_8 \end{bmatrix} \qquad (11)$$

Sixth, the eigenvalues and eigenvectors are obtained by the EVD processor detailed in Fig. 4 of [18]. Then, we use the CUs to calculate the whitening data and write back to the DM.

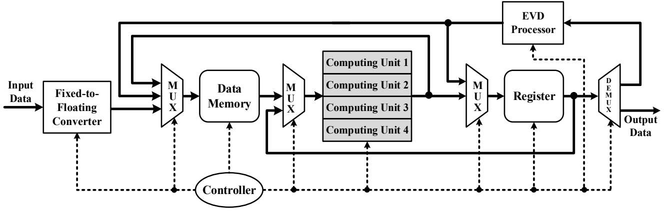


Fig. 3.1. Block diagram of the system architecture.

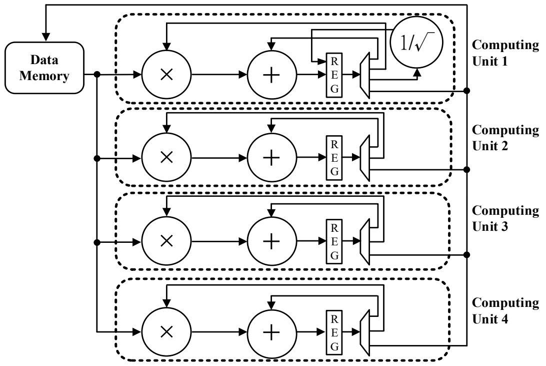


Fig. 3.2. Computing unit (CU) architecture and DM.

Thus, the preprocessing process is completed. Seventh, the whitening data fetched from the DM and weight matrix register are fed to the four CUs to perform fixed-point iteration operation parallel in (12).

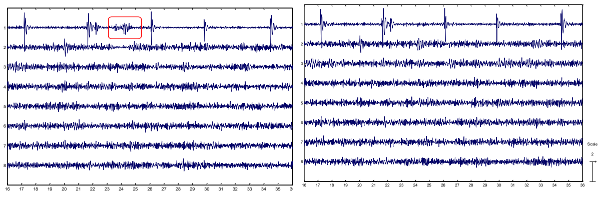
$$\mathbf{w}^+ = E\{\mathbf{z}[g(\mathbf{w}^T\mathbf{z})]^T\} - E\{g'(\mathbf{w}^T\mathbf{z})\}\mathbf{w} = E\{\mathbf{Z}[\tanh(\mathbf{w}^T\mathbf{Z})]^T\} - \{[1024 - \textstyle\sum_{i=1}^{1024}\tanh^2(\mathbf{w}^T\mathbf{z}_i)]/1024\}\,\mathbf{w} \qquad (12)$$

where $\mathbf{z}_i$ is the i-th column vector of the matrix $\mathbf{Z}$ and the thirteen-piecewise linear function approximation is adopted to calculate the hyperbolic tangent. Then, through Gram-Schmidt decorrelation and normalization, the resulting data are written back to the weight matrix register. We use the inner product to check whether the inner product value satisfies a convergence threshold or reaches the maximum iteration. Eighth, we judge the adjacent waveform relation by correlation values in (5)-(8) because FastICA output results might switch across calculations. While considering 8 channels and 4 CUs, the eight channels can be divided into two groups for four CUs to obtain 64 correlation coefficients for the permutation. After that, we update the momentum-controlled weight matrix. Finally, the ICs are obtained via four CUs. Note that the equations (10), (11) and (12) are the same as those in [18] except the sample size.

4. SIMULATION, IMPLEMENTATION, AND COMPARISON RESULTS

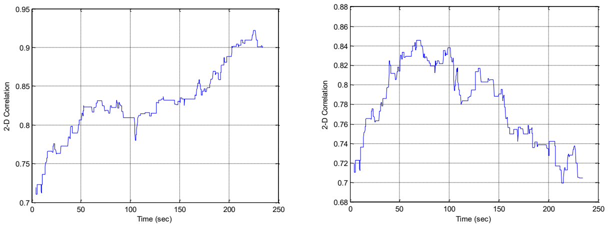
4.1 Online FastICA Algorithm Simulation Result

In the simulation, the real EEG data were used. Fig. 4.1(a) shows the referenced online FastICA results. As can be seen, the first row has a noise (red rectangle). Fig. 4.1(b) shows the proposed memory-limited online FastICA results, where the noise does not exist. The parameters used in this simulation are set as follows: the window size is 1,024, the data overlapping length is 960, the constant k is 4.7, and the momentum coefficient r is 0.05. To show the effect of the garbage detection, Figs. 4.2(a) and 4.2(b) show the 2D correlations between the weight matrices with/without garbage detection, respectively.



(a) (b)

Fig. 4.1. (a) Referenced and (b) proposed online FastICA results with EEG data.



(a) (b)

Fig. 4.2. 2D correlation (a) using garbage detection and (b) without garbage detection.

As can be seen, when time > 210 sec, the 2D correlation is greater than 0.9 with garbage detection. Without garbage detection, the 2D correlation is below 0.76 when time > 210 sec.

4.2 Software Simulation and Post-layout Simulation

The simulation result is used to validate the FastICA implementation using four CUs. Considering the real eight-channel EEG signals in Fig. 4.3, the Matlab and post-layout simulation waveforms, and the corresponding absolute correlation coefficients are shown in Fig. 4.4. The average value of the absolute correlation coefficients is 0.8875. For mixed-signal case, the average value of the absolute correlation coefficients is 0.9615 (herein due to the limited space, only the number is expressed rather than to provide figures). As a result, the proposed architecture can attain the satisfactory blind-source separation with the online stable component/channel order under the limited memory size.

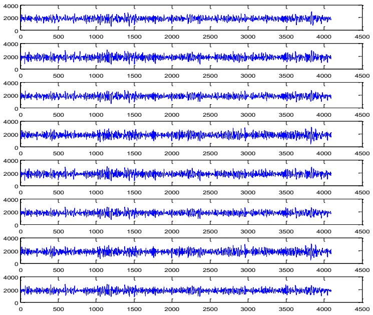
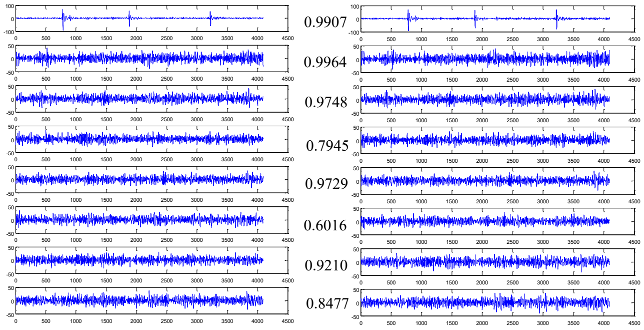


Fig. 4.3. Eight-channel EEG signal.



(a) (b)

Fig. 4.4. Comparison results of online FastICA using (a) Matlab and (b) post-layout simulation for EEG signals.

4.3 Chip Implementation

This subsection presents the chip implementation and layout characteristics. The cell-based design flow with the standard cell library in TSMC 90 nm 1P9M CMOS process is adopted. Artisan Memory Compiler and Synopsys Design Compiler are employed to synthesize the RTL design with the constraint of 10 ns. Cadence SOC Encounter is used to place and route for the proposed architecture. The layout of the proposed chip design is shown in Fig. 4.5, where the information is shown in Table I.

4.4 Comparison and Evaluation Results

Table I shows a comparison among this work and existing FastICA implementations. This table compares whether online stable channel order in hardware, the number of channels, sample size, speed, power consumption, gate counts, computation time, process technology, and implementation approaches. It is noted that the proposed FastICA architecture achieves high channel-order stability. That means the components are stable across time.

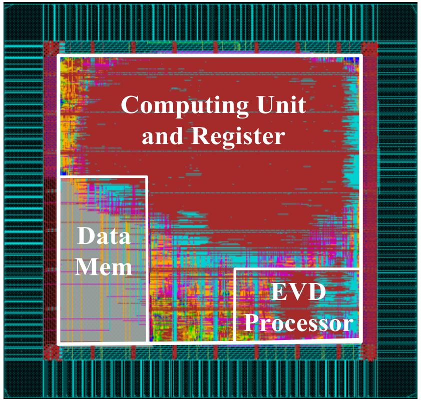


Fig. 4.5. Layout of hardware-oriented memory-limited online FastICA implementation.

5. CONCLUSIONS

This work proposed and presented a solution to the component/channel-switching problem due to the limited memory size constraint of the ASIC-based approach. The solution combines data overlapping, garbage detection, channel permutation, and momentum-controlled weight update schemes. The result from our simulation study demonstrated the proposed method is very effective to stabilize the component/channel order either in ASIC hardware or pure software simulations.

TABLE I: COMPARISON RESULTS AMONG VARIOUS FASTICA IMPLEMENTATIONS

	Shyu [16]	Van [18]	Roh [20]	Yang [21]	Van [22]	C.-C. [23]	Bhardwaj [24]	This Work
Online Stable Channel Order in Hardware	NO	NO	NO	NO	NO	NO	NO	YES
# of Channels/Weight Vectors (WVs)	2	8	16	8	2-16	2	6	8
Sample Size	3000	256	512	256	512	64000	1024	1024
Speed (MHz)	50	100	20	11	100	100	240	100
Power Dissipation (mW)	NA	16.35	4.45	0.0816	16.35	4200	0.5703	65.0
Gates (kilo)	NA	272	NA	69.2	401	NA	NA	840
Computation Time (ms)	~3	290 (max)	Variable	84.2 (max)	1850 (max)	68.1	NA	150 (max)
Process Technology (nm)	NA	90	130	90	90	NA	90	90
Implementation Approach	FPGA	ASIC	ASIC	ASIC	ASIC	FPGA+ARM	ASIC	ASIC

6. REFERENCES

[1] A. Hyvärinen and E. Oja, “A fast fixed-point algorithm for independent component analysis,” *Neural Computation*, vol. 9, pp. 1483–1492, 1997.

[2] A. Hyvärinen, “Fast and robust fixed-point algorithms for independent component analysis,” *IEEE Trans. Neural Networks*, vol. 10, pp. 626–634, May [19]

[3] 9.

[4] T. W. Lee, *Independent Component Analysis: Theory and Applications*. Boston, MA: Kluwer, 1998.

[5] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis.* New York: Wiley, 2001. Vigario, “Extraction of ocular artifacts from EEG using independent component analysis,” *Electroencephalogr. Clin. Neurophysiol*., vol. 103, pp. 395–404, 1997.

[6] R. Vigario, J. Sarela, V. Jousmaki, M. Hamalainen, and E. Oja, “Independent component approach to the analysis of EEG and MEG recordings,” *IEEE Trans. Biomed. Eng.,* vol. 47, pp. 589–593, May 2000.

[7] L. Vigon, M. R. Saatchi, J. E. W. Mayhew, and R. Fernandes, “Quantitative evaluation of techniques for ocular artefact filtering of EEG waveforms,” *IEE Proceedings-Science, Measurement and Technology*, vol. 147, no. 5, pp. 219-228, 2000.

[8] G. Wang, N. N. Rao, Z. L. Zhang, Q. Mo, and P. Wang, “An extended online Fast-ICA algorithm,” In *Proc. International Symposium on Neural Networks*, pp. 1109-1114, May 2006, Springer, Berlin, Heidelberg.

[9] A. Kachenoura, L. Albera, L. Senhadji, and P. Comon, “ICA: A potential tool for BCI systems,” *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 57–68, Jan. 2008.

[10] R. Vigario, and E. Oja, “BSS and ICA in neuroinformatics: from current practices to open challenges,” *IEEE Reviews in Biomedical Engineering*, vol. 1, pp. 50-61, 2008.

[11] B. Sallberg, N. Grbic, and I. Claesson, “Complex-valued independent component analysis for online blind speech extraction,” *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1624-1632, Nov. 2008.

[12] S. H. Hsu, T. R. Mullen, T. P. Jung, and G. Cauwenberghs, “Real-time adaptive EEG source separation using online recursive independent component analysis,” *IEEE Trans. Neural Systems and Rehabilitation Engineering*, vol. 24, no. 3, pp. 309–319, Mar. 2016.

[13] C. M. Kim, H. M. Park, T. Kim, Y. K. Choi, and S. Y. Lee, “FPGA implementation of ICA algorithm for blind signal separation and adaptive noise canceling,” *IEEE Trans. Neural Networks*, vol. 14, no. 5, pp. 1038–1046, Sep. 2003.

[14] C. Charoensak and F. Sattar, “A single-chip FPGA design for real-time ICA-based blind source separation algorithm,” in *Proc. IEEE Int. Symp. on Circuits and Systems*, vol. 6, pp. 5822-5825, May 2005.

[15] H. Du, H. Qi, “A reconfigurable FPGA system for parallel independent component analysis,” *EURASIP Journal on Embedded Systems*, vol. 2006, 12 pages, 2006.

[16] K. K. Shyu, M. H. Lee, Y. T. Wu, and P. L. Lee, “Implementation of pipelined FastICA on FPGA for real-time blind source separation,” *IEEE Trans. Neural Networks*, vol. 19, Jun. 2008.

[17] W. C. Huang, S. H. Hung, L. D. Van and C. T. Lin,” FPGA Implementation of 4-channel ICA for on-line EEG signal separation.” In *Proc. IEEE BioCAS.*, pp. 65-68, Nov. 2008

[18] L. D. Van, D. Y. Wu, C. S. Chen, "Energy-efficient FastICA implementation for biomedical signal separation", *IEEE Trans. on Neural Networks*, vol. 22, no. 11, pp. 1809-1823, Nov. 2011.

[19] Jo-Fu Wang, *Design and Implementation of a Hardware-oriented Online FastICA Algorithm*, Master Thesis, National Chiao Tung University, 2013. (Advisor: Lan-Da Van)

[20] T. Roh, K. Song, K., H. Cho, D. Shin, and H. J. Yoo,, “A wearable neuro-feedback system with EEG-based mental status monitoring and transcranial electrical stimulation,” *IEEE Trans. Biomedical Circuits and Systems*, vol. 8, no. 6, pp. 755-764, Dec. 2014.

[21] C. H. Yang, Y. H. Shih, H. Chiueh, “An 81.6uW FastICA processor for epileptic seizure detection,” *IEEE Trans. Biomedical Circuits and Systems*, vol. 9, no. 1, pp. 60-71, Feb. 2015.

[22] L. D. Van, P. Y. Huang, and T. C. Lu, “Cost-effective and variable-channel FastICA hardware architecture and implementation for EEG signal processing,” *Journal of Signal Processing Systems*, vol. 82, issue 1, pp. 91-113, Jan. 2016.

[23] F. Carrizosa-Corral, A. Vazquez-Cervantes, J. R. Montes, T. Hernandez-Diaz, J. C. Solano Vargas, L. Barriga-Rodriguez, J. A. Soto-Cajiga, H. Jimenez-Hernandez, “FPGA-SoC implementation of an ICA-based background subtraction method,” *International Journal of Circuit Theory and Applications*, vol. 46, pp. 1703-1722, Apr. 2018.

[24] S. Bhardwaj, S. Raghuraman, and A. Acharyya, “Simplex FastICA: An accelerated and low complex architecture design methodology for *n*D FastICA,” *IEEE Trans. Very Large Scale Integration (VLSI) Systems,* to appeared in 2019.

[25] J. Onton, M. Westerfield, J. Townsend, and S. Makeig, “Imaging human EEG dynamics using independent component analysis,” *Neuroscience and Biobehavioral Reviews*, vol. 30, no. 6, 2006.

[26] S. Hoffmann, “Independent component analysis of ocular artifacts,” Ph.D. dissertation, Faculty of Psychology, Ruhr-University Bochum, Germany, 2009.