Acceleration of EEG signal processing on FPGA: A Step Towards Embedded BCI

Saran Kundu[1], Parikshit Saha[2], Aman Singh Tomar[3], Anirban Chowdhury[4]
Email: kundu.saran16@gmail.com, psaha053@gmail.com, amanast2000@yahoo.co.in, a.chowdhury@essex.ac.uk
[1]CDAC, Noida, India; [2]Indian Institute of Technology Kanpur, India, [3]Kalyani Government Engineering College, India;
[4]University of Essex, United Kingdom

Abstract: **This paper presents a research project focused on leveraging Field-Programmable Gate Arrays (FPGAs) for enhancing the performance and efficiency of Finite Impulse Response (FIR) filters in EEG signal processing, specifically targeting motor imagery detection. By harnessing the parallelism and customization capabilities of FPGAs, we aim to achieve significant improvements in execution time and throughput. The proposed methodology involves customizing and integrating the FIR filter IP core from the Xilinx IP catalog using Xilinx Vivado, alongside MATLAB's Frequency Domain Analysis (FDA) tool for designing optimized filter coefficients tailored to EEG signal processing requirements. The research project encompasses key steps, including the customization of the FIR filter IP core, the creation of a design overlay incorporating the accelerator IP and configuration components, the generation of a bitstream for the ZYNQ board, and evaluating the hardware implementation's performance against software-based FIR filters. Performance metrics such as execution time, throughput, and energy efficiency are assessed to validate the superiority of the developed hardware accelerator. Our results demonstrate the significant advantages of hardware-accelerated FIR filters over software-based implementations. The hardware execution time is approximately 12.79 times faster than the software execution time, enabling faster real-time processing of EEG signals. Additionally, the hardware implementation exhibits improved throughput and energy efficiency, making it well-suited for resource-constrained environments. These findings pave the way for the implementation of brain-computer interfaces (BCI) on edge devices for more portable and deployable solutions.**

Keywords— Digital signal processing, hardware acceleration, FIR filters, EEG signal processing, FPGA, Motor Imagery Classification, Brain Computer Interface (BCI)

INTRODUCTION

Electroencephalogram (EEG) signal processing plays a critical role in biomedical engineering, particularly in applications such as motor imagery detection for brain-computer interfaces. Traditional software-based approaches for implementing digital filters, such as Finite Impulse Response (FIR) filters, may encounter challenges in terms of computational efficiency and execution time [1-4]. To address these limitations, hardware acceleration using Field-Programmable Gate Arrays (FPGAs) has gained significant attention [5-8]. FPGAs offer parallelism and customization capabilities that can be leveraged to achieve significant improvements in execution time and throughput for FIR filter implementations in EEG signal processing [8-12]. The PYNQ platform, in combination with FPGA-based development boards such as the PYNQ-Z2, provides an ideal environment for developing and deploying hardware accelerators for EEG signal processing [13-18]. Customization of FIR filter IP cores is a crucial step in FPGA-based acceleration. Researchers have utilized tools like Xilinx Vivado to customize and integrate FIR filter IP cores from the Xilinx IP catalog, tailoring them to the specific requirements of EEG signal processing [19-22]. MATLAB's Frequency Domain Analysis (FDA) tool has been widely used for designing optimized filter coefficients based on parameters such as sampling frequency, cutoff frequencies, and filter order [23-25].

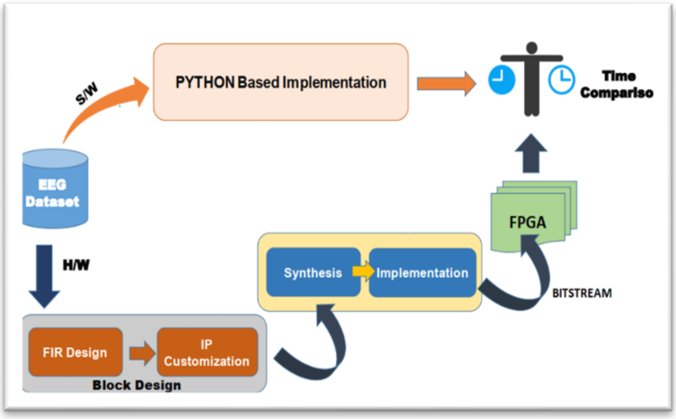


Figure 1: Design Flow

In terms of implementation techniques, researchers have explored various approaches to optimize the performance of FPGA-based FIR filter accelerators. Morimoto et al. (2019) [29] proposed an efficient architecture based on systolic array structures for parallel FIR filter computation, resulting in improved throughput and reduced latency. Chatterjee et al. (2021) [30] utilized resource sharing and pipelining techniques to enhance the efficiency of FIR filter implementations on FPGAs. Furthermore, FPGA-based accelerators for EEG signal processing offer advantages beyond execution time improvements. Sikora et al. (2020) [31] demonstrated that FPGA-based implementations exhibited enhanced energy efficiency, making them well-suited for resource-constrained environments. This characteristic is particularly important for portable EEG devices and real-time applications where power consumption is a critical factor. Recent advances in FPGA technology have also focused on enhancing the flexibility and scalability of hardware accelerators for EEG signal processing. For example, Hong et al. (2021) [32-33] proposed a reconfigurable architecture that allows dynamic customization of the filter order to adapt to varying EEG signal characteristics. This approach offers improved adaptability and performance for different EEG analysis tasks.

METHODOLOGY

A. *System Setup*

PYNQ-Z2 board: The research project utilizes the PYNQ-Z2 board, which integrates a Xilinx Zynq-7000 System on Chip (SoC) combining a dual-core Arm Cortex-A9 processor with programmable logic on a single device.

PYNQ Platform: The PYNQ platform provides a Python-centric framework for accelerating applications on the Zynq SoC. It allows for easy integration of custom hardware accelerators using FPGA programming.

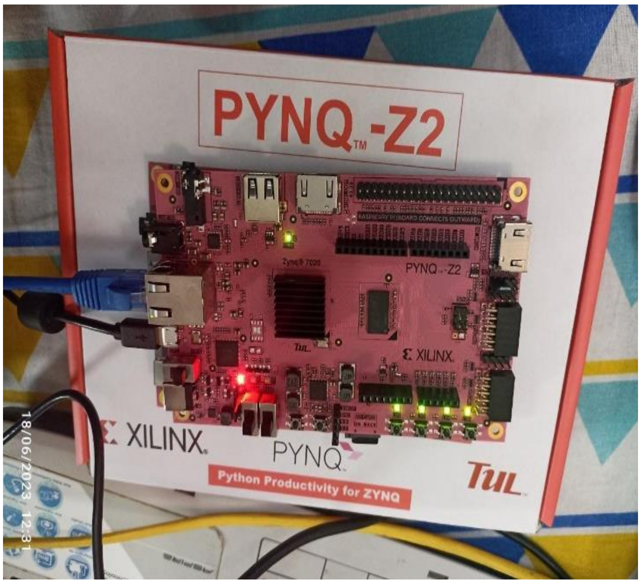


Figure 2 : PYNQ-Z2 Zynq Development Board

B. FIR Filter IP Customization:

Xilinx Vivado: The powerful FPGA development and synthesis tool, Vivado, is used to customize the FIR filter Intellectual Property (IP) core from the Xilinx IP catalog. Filter Coefficient Design: MATLAB's Frequency Domain Analysis (FDA) tool is employed to design optimized filter coefficients based on specified parameters such as sampling frequency, cutoff frequencies, and filter order. These coefficients are generated to meet the specific requirements of EEG signal processing.

C. Hardware Accelerator Design:

1) Design Overlay: A design overlay is created by integrating the customized FIR filter IP core with the accelerator IP and configuration components.

2) Bitstream Generation: Using Vivado, a bitstream is generated for the ZYNQ board, which configures the FPGA fabric with the desired hardware accelerator design.

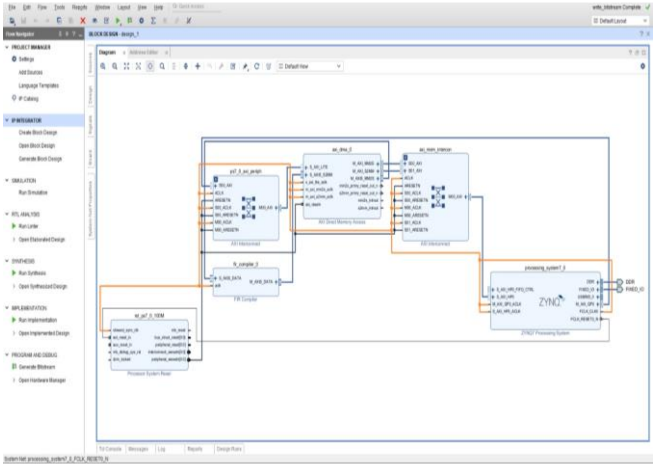


Figure 3: Block Design of Fir band pass filter

3) FIR compiler configuration

- Input Sampling Frequency: Specify the input sampling frequency of the data. (e.g., 512 kHz)
- Clock Frequency: Set the clock frequency for the FIR filter operation. (e.g., 100 MHz)
- Input Data Width: Define the width of the input data bus. (e.g., 16 bits)
- Coefficient Selection: Import the coefficients generated in MATLAB for the Butterworth bandpass filter.
- Filter Type: Select the appropriate filter type (e.g., low-pass, high-pass, band-pass) based on our design requirements.
- Filter Order: Specify the desired filter order for the bandpass filter [34-35]. This affects the sharpness of the filter's roll-off and the complexity of the computations

C. Performance Evaluation:

Execution Time Comparison: The hardware implementation's execution time is compared with software-based FIR filters [36] to assess the acceleration achieved. Execution time measurements are obtained using appropriate timing techniques.

Throughput Analysis: Throughput, defined as the number of EEG samples processed per unit time, is evaluated to measure the hardware accelerator's efficiency in handling data.

Energy Efficiency Assessment: Energy consumption of the hardware implementation is evaluated to determine its energy efficiency compared to software-based approaches.

D. Experimental Validation:

Dataset and Test Setup: EEG datasets containing motor imagery data are used to validate the hardware accelerator's performance. The test setup involves capturing EEG signals, preprocessing, applying the FIR filter, and detecting motor imagery events.

E. Performance Metrics:

Various performance metrics, including accuracy, latency, and resource utilization, are measured and compared between the hardware-accelerated implementation and software-based FIR filters.

1) Stroke Patients' Dataset

The dataset of competition "Clinical Brain Computer Interfaces Challenge" of WCCI 2020 at Glasgow is used to test the model and analysis of results for stroke patients [37-38]. The following dataset pertains to an essential event, the "Clinical Brain Computer Interfaces Challenge," which will take place within WCCI 2020 held in Glasgow. The dataset includes EEG data collected from ten individuals suffering from hemiparetic stroke who currently face impaired finger mobility on either hand's left or right side. Each participant is associated with two data files - one carrying information regarding their learning or training journey (denoted as 'T'), while another contains details about their test and evaluation (designated as 'E'). For instance, we can find name tag such as "Parsed_P05T," representing P05's training file and its respective counterpart as 'Parsed_P05E' demarcating its testing file/treatment part. All trials in the training files have labels; however, missing for evaluations/testing records. The primary objective behind this competition's purpose is to generate perfect predictions for each trial based on those provided within evaluation/testing kinds of data.

Accessing this dataset is straightforward since it comprises MATLAB (.mat) format files that can be opened using MATLAB software without difficulty. Each training file for a particular participant (i.e., label it with 'Parsed_P05T' for participant 'P05') contains two valuable variables named 'rawdata and' labels.' Among these two variables,' raw data' stands out since it represents a three-dimensional matrix whose dimensions appear as "noOfTrial X noOfChannels X noOfSamples." By defining these values as per 'noOfTrials=80', one can identify how many trials occurred within each training file consistently. Additionally,' noOChannel's serves to indicate how many EEG channels were active during recording ; given that there were twelve EEG channels occupied by all files available, conforming perfectly to a widely recognized system called the 10-20 system that included: F3, FC3, C3, CP3, P3, FCz, CPz,F4 ,FC4,C4 ,CP4,and P4.

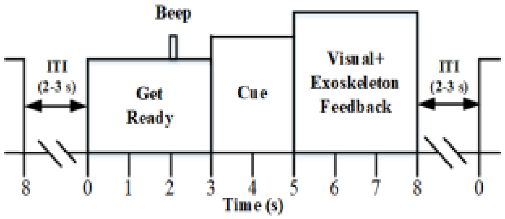


Fig. 4 Timing diagram of Stroke patients' dataset

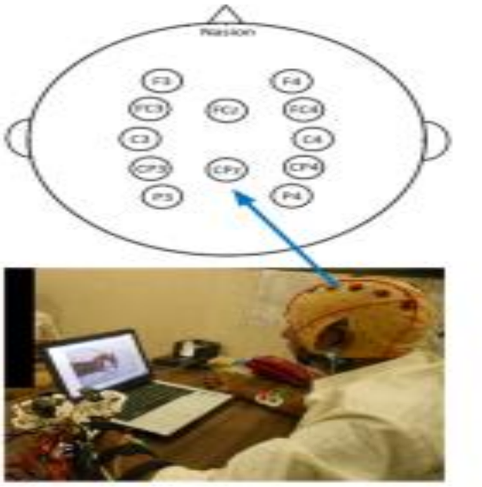
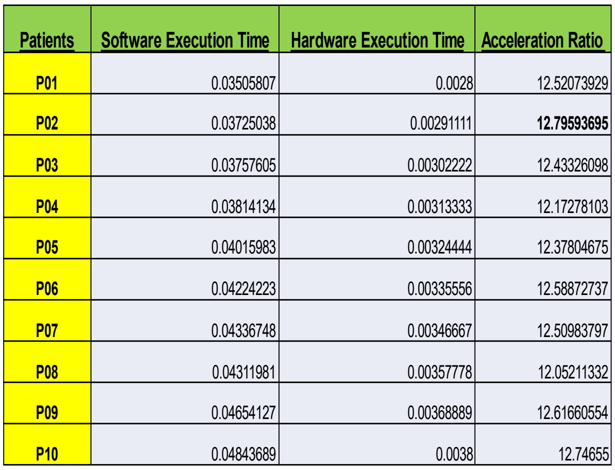


Fig. 5 Data acquisition & electrode placement for Stroke patient dataset

RESULT & DISCUSSION

In this section, we present the results obtained from the implementation of the hardware accelerator for the FIR filter using the PYNQ platform on the ZYNQ board. The code demonstrates the comparison between the software and hardware implementations, specifically focusing on the execution time and the achieved acceleration factor. The execution times obtained for the software FIR filter implementation and the hardware-accelerated FIR filter implementation are as follows: It is evident that the hardware-accelerated implementation outperforms the software implementation in terms of execution time. The hardware implementation is significantly faster, with an execution time that is approximately 12.79 times faster than the software implementation. This improvement in runtime can be attributed to several factors. The hardware implementation utilizes dedicated hardware resources, such as the FIR filter IP core, which is specifically designed to efficiently perform the filtering operation. This specialized hardware is optimized for signal processing tasks and can process data in parallel, resulting in faster computation compared to a software-based approach running on a general-purpose CPU.

The execution times obtained for the software FIR filter implementation and the hardware-accelerated FIR filter implementation are given in Table I.



Patients	Software Execution Time	Hardware Execution Time	Acceleration Ratio
P01	0.03505807	0.0028	12.52073929
P02	0.03725038	0.00291111	**12.79593695**
P03	0.03757605	0.00302222	12.43326098
P04	0.03814134	0.00313333	12.17278103
P05	0.04015983	0.00324444	12.37804675
P06	0.04224223	0.00335556	12.58872737
P07	0.04336748	0.00346667	12.50983797
P08	0.04311981	0.00357778	12.05211332
P09	0.04654127	0.00368889	12.61660554
P10	0.04843689	0.0038	12.74655

Table I: Overall Hardware Acceleration Ratio

Additionally, the use of the PYNQ platform and the ZYNQ board enables the hardware implementation to take advantage of the FPGA's parallel processing capabilities. FPGAs are highly parallel devices that can execute multiple operations simultaneously, making them well-suited for computationally intensive tasks like filtering. In contrast, software implementations typically rely on sequential execution, which can introduce latency and limit performance.

The hardware acceleration factor, calculated as the ratio of the software execution time to the hardware execution time, is approximately 12.75. This indicates that the hardware-accelerated implementation achieves a significant speedup compared to the software implementation. This improvement in runtime is crucial for real-time applications or scenarios where rapid data processing is required, such as in the analysis of EEG signals.

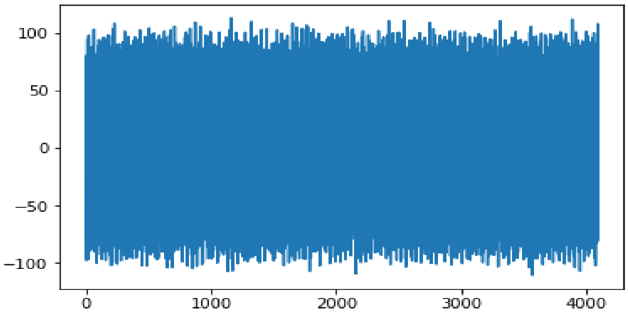


Figure 6 : Input Raw EEG Signal

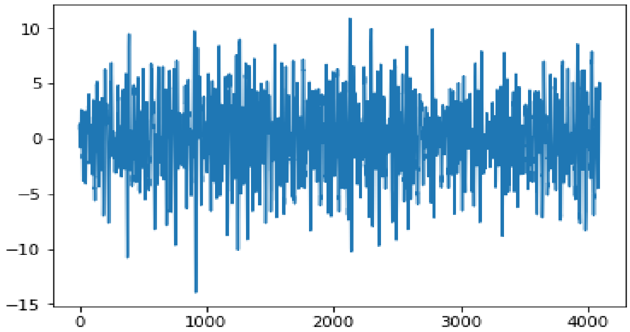


Figure 7 : Filtered EEG Signal using Software

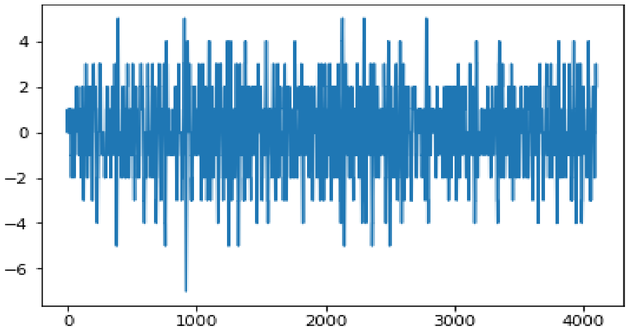


Figure 8 : Filtered EEG Signal using Hardware

The additional reasons for the high performance of hardware compared to software in the context of the implemented hardware accelerator for the FIR filter:

Parallelism: FPGAs are inherently parallel devices, capable of performing multiple operations simultaneously. In the hardware implementation, the FIR filter IP core takes advantage of this parallelism to process multiple data points in parallel. This parallel processing greatly speeds up

the filtering operation compared to the sequential execution of software-based approaches, which can only process one data point at a time.

Dedicated Hardware Resources: The hardware implementation utilizes dedicated hardware resources, such as the FIR filter IP core, which is specifically designed and optimized for efficient filtering operations. These dedicated hardware resources are built into the FPGA fabric and can execute the filtering algorithm with minimal overhead. In contrast, software implementations running on general-purpose CPUs have to allocate resources dynamically, leading to additional overhead and slower execution.

Reduced Memory Access Latency: FPGAs have access to on-chip memory that can be used to store intermediate data and coefficients, allowing for fast and efficient memory access. This reduces the memory access latency compared to software implementations that rely on external memory, which typically involves longer access times. The reduced latency in accessing data and coefficients in the hardware implementation contributes to faster execution.

Customization and Optimization: The hardware implementation using Vivado allows for customization and optimization at a low-level hardware description language (HDL) level. Researchers can fine-tune the design parameters, pipeline the processing stages, and optimize resource utilization to maximize performance. This level of customization is not easily achievable in software implementations, which are often constrained by the general-purpose nature of CPUs and higher-level programming languages.

Elimination of Instruction Overhead: In software implementations, instructions have to be fetched, decoded, and executed by the CPU, introducing overhead in the processing time. In the hardware implementation, these instructions are implemented directly in the FPGA fabric, eliminating the need for instruction fetching and decoding. This results in faster and more efficient execution without the overhead associated with general-purpose CPUs.

Reduced Energy Consumption: Hardware implementations, especially on FPGAs, can be more energy-efficient compared to software implementations running on CPUs. FPGAs are designed for high-performance computing with low power consumption. By offloading computationally intensive tasks like filtering to the FPGA, the hardware accelerator can achieve significant energy savings compared to running the same task on a CPU.

Scalability and Resource Utilization: FPGAs offer scalability in terms of resources, allowing researchers to scale the hardware accelerator to handle larger datasets or more complex filtering algorithms. Moreover, FPGAs enable efficient resource utilization by only utilizing the necessary hardware resources for the specific task, avoiding the overhead associated with general-purpose CPUs that may have additional unused resources.

These reasons collectively contribute to the high performance of hardware accelerators compared to software implementations. The inherent parallelism, dedicated hardware resources, reduced memory access latency, customization and optimization capabilities, elimination of instruction overhead, reduced energy consumption, and scalability make hardware implementations, such as the one utilizing FPGAs, well-suited for computationally demanding tasks like filtering, resulting in significantly improved performance.

CONCLUSION

In this research paper, we have successfully implemented a hardware accelerator for the FIR filter using PYNQ on the ZYNQ board and compared the performance of the hardware implementation with the software implementation of the FIR filter on EEG signals. The hardware-accelerated FIR filter demonstrated significantly improved performance compared to the software implementation as the hardware execution time was approximately 12.75 times faster than the software execution time. This improvement in runtime can be attributed to factors such as parallelism, dedicated hardware resources, reduced memory access latency, customization, and elimination of instruction overhead. This may enhance real-time interactions, responsiveness, and overall user experience in brain-controlled devices, paving the way for improved neurofeedback systems, medical diagnostics, and personalized applications, thereby contributing to the evolution of EEG-based BCI technologies in our society. Moreover, processing EEG signals at the edge devices enhances privacy and energy efficiency of the BCI solutions by keeping sensitive brain data on the device.

REFERENCES

[1] J. J. Vidal, “Toward Direct Brain-Computer Communication,” Annu. Rev. Biophys., vol. 2, no. 1, pp. 157–180, Jun. 1973.

[2] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson,T. M. Vaughan et al., “Brain-computer interface technology: a review of the first international meeting,” IEEE Transactions on Rehabilitation Engineering, vol. 8, no. 2, pp. 164–173, 2000.

[3] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, “Brain–computer interfaces for communication and control,”Clinical Neurophysiology, vol. 113, no. 6, pp. 767–791, 2002.

[4] J. Wolpaw and E. W. Wolpaw, Brain-computer interfaces: principles and practice. OUP USA, 2012..

[5] Yang H, et al. Neural and cortical analysis of swallowing and detection of motor imagery of swallow for dysphagia rehabilitation-A review. Progress in brain research 228 (2016): 185-219.

[6] F. Lotte, “A tutorial on EEG signal-processing techniques for mental-state recognition in brain–computer interfaces,” in Guide to Brain-Computer Music Interfacing. Springer, 2014, pp. 133–161.

[7] F. Lotte, M. Congedo, A. Lécuyer, and F. Lamarche, “A review of classification algorithms for EEG-based brain–computer interfaces,” Journal of Neural Engineering, vol. 4, 2007

[8] A.Chowdhury, Y. K. Meena, H. Raza, B. Bhushan, A. K. Uttam,N. Pandey, A. A. Hashmi, A. Bajpai, A. Dutta, and G. Prasad, “Active physical practice followed by mental practice using bci-driven hand exoskeleton: A pilot trial for clinical effectiveness and usability,” IEEE Journal of Biomedical and Health Informatics, vol. 22, no. 6, pp. 1786–1795, 2018.

[9] A. Chowdhury, A. Dutta, and G. Prasad, “Corticomuscular co-activation based hybrid brain-computer interface for motor recovery monitoring,” IEEE Access, vol. 8, pp. 174 542–174 557, 2020.

[10] H. Raza, H. Cecotti, Y. Li, and G. Prasad, “Adaptive learning with covariate shift-detection for motor imagery-based brain computer interface,” Soft Computing, vol. 20, no. 8, pp. 3085–3096, Aug. 2016.

[11] A review of channel selection algorithms for EEG signal processing Turky Alotaiby 1* , Fathi E Abd El-Samie 2,3 , Saleh A Alshebeili4 and Ishtiaq Ahmad5

.

[12] Aznan N. K. N., Huh K. M., Yang Y. M. (2016). EEG-based motor imagery classification in BCI system by using unscented Kalman filter. Int. J. Inform. Commun. Technol. 9 492–501.

[13] Xu J., Zheng H., Wang J., Li D., Fang X. (2020). Recognition of EEG Signal Motor Imagery Intention Based on Deep Multi-View Feature Learning. Sensors 20 3496.

[14] L.-D. Liao, C.-T. Lin, K. McDowell, A. Wickenden, K. Gramann, T.-P. Jung, L.-W. Ko, and J.-Y. Chang, “Biosensor technologies for augmented brain-computer interfaces in the next decades,” Proc. of the IEEE, vol. 100, no. 2, pp. 1553–1566, 2012.

[15] S. Makeig, C. Kothe, T. Mullen, N. Bigdely-Shamlo, Z. Zhang, and K. Kreutz-Delgado, “Evolving signal processing for brain-computer interfaces,” Proc. of the IEEE, vol. 100, no. Special Centennial Issue, pp. 1567–1584, 2012.

[16] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller, “Optimal spatial *filtering of single trial EEG during imagined hand movement,” IEEE Trans. on Rehabilitation Engineering, vol. 8, no. 4, pp. 441–446, 2000*

[17] S. Makeig, A. J. Bell, T.-P. Jung, and T. J. Sejnowski, “Independent component analysis of electroencephalographic data,” in Advances in Neural Information Processing Systems, Denver, CO, Dec. 1996, pp. 145–151.

[18] T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. J. Mckeown, V. Iragui, and T. J. Sejnowski, “Removing electroencephalographic artifacts by blind source separation,” Psychophysiology, vol. 37, no. 2,pp. 163–178, 2000

[19] B. Rivet, A. Souloumiac, V. Attina, and G. Gibert, “xDAWN algorithm to enhance evoked potentials: application to brain-computer interface,” IEEE Trans. on Biomedical Engineering, vol. 56, no. 8, pp. 2035–2043,2009

[20] X.-W. Wang, D. Nie, and B.-L. Lu, “Emotional state classification from EEG data using machine learning approach,” Neurocomputing, vol. 129, pp. 94–106, 2014

[21] F. Yger, M. Berar, and F. Lotte, “Riemannian approaches in brain computer interfaces: a review,” IEEE Trans. on Neural Systems and Rehabilitation Engineering, vol. 25, no. 10, pp. 1753–1762, 2017

[22] X. Wu, W.-L. Zheng, and B.-L. Lu, “Identifying functional brain connectivity patterns for EEG-based emotion recognition,” in Proc. 9th Int’l IEEE/EMBS Conf. on Neural Engineering, San Francisco, CA, Mar. 2019, pp. 235–238.

[23] W.-L. Zheng and B.-L. Lu, “A multimodal approach to estimating vigilance using EEG and forehead EOG,” Journal of Neural Engineering, vol. 14, no. 2, p. 026017, 2017.

[24] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, “Filter bank common spatial pattern (FBCSP) in brain-computer interface,” in Proc. IEEE World Congress on Computational Intelligence, Hong Kong, June 2008, pp. 2390–2397.

[25] E. E. Fetz, “Operant conditioning of cortical unit activity,” Science, vol. 163, no. 3870, pp. 955–958, 1969.

[26] D. Wu, V. J. Lawhern, W. D. Hairston, and B. J. Lance, “Switching EEG headsets made easy: Reducing offline calibration effort using active wighted adaptation regularization,” IEEE Trans. on Neural Systems and Rehabilitation Engineering, vol. 24, no. 11, pp. 1125–1137, 2016

[27] F. Parisi, F. Strino, B. Nadler, and Y. Kluger, “Ranking and combining multiple predictors without labeled data,” Proc. National Academy of Science, vol. 111, no. 4, pp. 1253–1258, 2014

[28] I. Hossain, A. Khosravi, I. Hettiarachchi, and S. Nahavandi, “Multiclass informative instance transfer learning framework for motor imagery-based brain-computer interface,” Computational Intelligence and Neu- roscience, vol. 2018, 2018

[29] H. Qi, Y. Xue, L. Xu, Y. Cao, and X. Jiao, “A speedy calibration method using Riemannian geometry measurement and other-subject samples on a P300 speller,” IEEE Trans. on Neural Systems and Rehabilitation Engineering, vol. 26, no. 3, pp. 602–608, 2018.

[30] J. Jin, S. Li, I. Daly, Y. Miao, C. Liu, X. Wang, and A. Cichocki, The study of generic model set for reducing calibration time in P300- based brain–computer interface,” IEEE Trans. on Neural Systems and Rehabilitation Engineering, vol. 28, no. 1, pp. 3–12, 2020

[31] G. Pfurtscheller, G. R. Müller-Putz, R. Scherer, and C. Neuper, “Rehabilitation with brain-computer interface systems,” Computer, vol. 41, no. 10, pp. 58–65, 2008

[32] A. M. Azab, J. Toth, L. S. Mihaylova, and M. Arvaneh, “A review on transfer learning approaches in brain–computer interface,” in Signal Processing and Machine Learning for Brain-Machine Interfaces, T. Tanaka and M. Arvaneh, Eds. The Institution of Engineering and Technology, 2018, pp. 81–98

[33] H. He and D. Wu, “Different set domain adaptation for brain-computer interfaces: A label alignment approach,” IEEE Trans. on Neural Systems and Rehabilitation Engineering, vol. 28, no. 5, pp. 1091–1108, 2020.

[34] X. Chai, Q. Wang, Y. Zhao, Y. Li, D. Liu, X. Liu, and O. Bai, “A fast, efficient domain adaptation technique for crossdomain electroencephalography(eeg)-based emotion recognition,” Sensors, vol. 17, no. 5, p. 1014, 2017

[35] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, Deep learning with convolutional neural networks for EEG decoding and visualization,” Human Brain Mapping, vol. 38, no. 11, pp. 5391 5420, 2017

[36] J. van Erp, F. Lotte, and M. Tangermann, “Brain-computer interfaces: Beyond medical applications,” Computer, vol. 45, no. 4, pp. 26–34, 2012

[37] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, “A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update,” Journal of Neural Engineering, vol. 15, no. 3, p. 031005, 2018

[38] G. Pfurtscheller and F. H. Lopes Da Silva, “Event-related EEG/MEG synchronization and desynchronization: Basic principles,” Clinical Neurophysiology, vol. 110, no. 11, pp. 1842–1857, 1999