

University of Southampton Research Repository

Copyright © and Moral Rights for this thesis and, where applicable, any accompanying data are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis and the accompanying data cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content of the thesis and accompanying research data (where applicable) must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holder/s.

When referring to this thesis and any accompanying data, full bibliographic details must be given, e.g.

Thesis: Author (Year of Submission) "Full thesis title", University of Southampton, name of the University Faculty or School or Department, PhD Thesis, pagination.

Data: Author (Year) Title. URI [dataset]

UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Physical Sciences
Electronics and Computer Science

**Real-time hardware architecture for
characterising functional brain connectivity
from mobile electroencephalogram**

by

Rafael Angel Gutierrez Nuno

ORCID: [0000-0002-8226-4725](https://orcid.org/0000-0002-8226-4725)

*A thesis for the degree of
Doctor of Philosophy*

November 2021

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
Electronics and Computer Science

Doctor of Philosophy

**Real-time hardware architecture for characterising functional brain connectivity
from mobile electroencephalogram**

by Rafael Angel Gutierrez Nuno

The usage of transcranial current stimulation has recently gained attention due to its potential as an alternative treatment to mental and neurological disorders, but its usage remains questionable due to the lack of protocols in its implementation. Different studies suggest that functional brain connectivity (FC) can be used to guide the stimulus and provide more accurate treatment. This, combined with wearable electroencephalogram (EEG) systems, opens the possibility for a new treatment where the patient is continuously monitored and the stimulus is correctly applied when needed. However, this approach would require real-time EEG signal processing to characterise the FC, which have only been done in computer software, thus limiting the scope of this treatment.

To address this, a new real-time hardware architecture for characterising FC from mobile EEG is proposed in this work. This architecture performs the characterisation in real-time and can be implemented on a field-programmable gate array (FPGA) as a wearable device instead of a computer. To achieve this performance, every process involved in the calculation was analysed and optimised using different design approaches to select the most efficient implementation to decrease the computational complexity, allowing real-time processing and low energy consumption.

The architecture was synthesised in an FPGA as a proof-of-concept, and it was able to calculate the FC and its characterisation in a total time of 273.10 μ s with a power consumption of 51.84 mW, meeting the timing and energy requirements established for this project. Furthermore, a seizure classifier was created to evaluate the impact in performance using the proposed hardware architecture against its software equivalent. The result showed a maximum difference of 0.15% in the classifier performance, demonstrating that the proposed architecture can be used for the real-time calculation and characterisation of the FC with minimal impact in the final performance.

Contents

List of Figures	xi
List of Tables	xiii
Declaration of Authorship	xv
Acknowledgements	xvii
Abbreviations	xix
1 Introduction	1
1.1 Motivation	2
1.2 Aims and Objectives	4
1.3 Contributions	5
1.4 Thesis Outline	7
2 Background and Literature Review: Electroencephalography, Artifact Removal and Functional Connectivity	9
2.1 The physiology of the brain and electroencephalography	9
2.1.1 Brain activity	9
2.1.2 Generation of the action potential	11
2.1.3 Brain activity monitoring	12
2.1.3.1 Electroencephalography (EEG)	12
2.1.3.2 ElectroCorticoGram (ECoG)	14
2.1.3.3 Local Field Potential (LFP)	14
2.1.3.4 functional Magnetic Resonance Imaging (fMRI)	15
2.1.3.5 functional Near-Infrared Spectroscopy (fNIRS)	15
2.1.3.6 Comparison	16
2.1.4 Wearable EEG systems	18
2.2 Artifacts and their treatment in EEG signals	19
2.2.1 Artifacts in EEG	19
2.2.1.1 Non-Physiological artefacts	19
2.2.1.2 Physiological artefacts	19
2.2.2 Artifact removal algorithms	20
2.2.2.1 Adaptive Filtering (AF)	20
2.2.2.2 Independent Component Analysis (ICA)	21
2.2.2.3 Wavelet Transform (WT)	22
2.2.2.4 Discrete Wavelet Transform (DWT)	22

2.2.2.5	Empirical Mode Decomposition (EMD)	24
2.2.3	Comparison of the algorithms	24
2.2.4	Challenges of artifact separation in wearable EEG	26
2.3	Brain Connectivity Networks	26
2.3.1	Effective connectivity (EC)	27
2.3.2	Functional connectivity (FC)	27
2.3.2.1	General Synchronization Index (GSI)	28
2.3.2.2	Phase locking value (PLV)	29
2.3.2.3	Phase lag index (PLI)	30
2.4	Characterisation of Brain connectivity networks	31
2.5	Applications of Brain connectivity networks in different disease phenotypes	32
3	System specification and design approach	33
3.1	System specification	33
3.1.1	Timing specification	33
3.1.2	Power specification	34
3.1.3	Summary of the system specifications	36
3.2	Our design approach	36
3.2.1	Algorithmic reformulation of computational intensive modules	37
3.2.2	Design flow	38
4	Real-time artifact removal hardware based on hybrid WPT-EMD algorithm for multichannel wearable wireless EEG system	41
4.1	Introduction	41
4.2	Background	42
4.2.1	State of the art artifact removal algorithms	42
4.2.2	Hybrid WPT-EMD Artifact Removal Algorithm	44
4.2.2.1	Wavelet Packet Decomposition (WPT)	44
4.2.2.2	Empirical Mode Decomposition (EMD)	45
4.2.3	Performance parameters	46
4.2.4	Algorithm shortcomings	46
4.3	Algorithmic Formulation and parameter determination for real-time hardware implementation	47
4.3.1	Experimental data	48
4.3.2	WPT	48
4.3.2.1	Window size and Mother Wavelet	48
4.3.2.2	Node removal selection	50
4.3.3	EMD	50
4.3.3.1	Window size	50
4.3.3.2	Interpolation method	51
4.3.3.3	Sifting Iterations	52
4.3.3.4	Maximum Number of IMF	53
4.3.3.5	IMF removal selection	54
4.3.4	Rejection Criteria	54
4.4	Hardware Implementation	55
4.4.1	WPT Architecture	56

4.4.1.1	WPT Decomposition Module	56
4.4.1.2	Variance Module	57
4.4.1.3	WPT Reconstruction Module	57
4.4.2	EMD architecture	58
4.4.2.1	Memory Unit	58
4.4.2.2	Arithmetic Logic Unit (ALU)	59
4.5	Results	60
4.5.1	Validation using a simulated artifact	60
4.5.2	Validation using real artifacts	63
4.5.3	Hardware Resources	66
4.5.4	Architectural Complexity	66
4.5.5	Power Analysis	67
4.5.6	Comparison	68
4.6	Conclusion	69
5	Phase Lag Index hardware implementation for real-time network connectivity formulation	71
5.1	Introduction	71
5.2	Background	72
5.2.1	The Computational flow of PLI calculation	72
5.3	Algorithm complexity and accuracy analysis of PLI components	73
5.3.1	Implementation of the Hilbert transform	73
5.3.1.1	Hilbert transform using Fourier transform	73
5.3.1.2	Hilbert transformer Filter	74
5.3.1.3	Comparison	74
5.3.2	Instantaneous Phase Calculation	76
5.3.2.1	LUT	77
5.3.2.2	Coordinate rotation digital computer (CORDIC)	77
5.3.2.3	Comparison	77
5.3.3	PLI calculation	79
5.4	Hardware Implementation	79
5.4.1	The analytic signal	79
5.4.2	The phase calculation	80
5.4.3	The PLI Calculation	80
5.4.4	PLI module integration	81
5.5	Results	82
5.5.1	Validation	82
5.5.2	Comparison	84
5.6	Conclusion	85
6	Hardware architecture for real-time EEG-based functional brain connectivity parameter extraction	87
6.1	Introduction	87
6.2	Background	88
6.2.1	Network integration parameters	88
6.2.1.1	Degree and Weighted Degree	89
6.2.1.2	Density	89

6.2.1.3	Characteristic Path Length	89
6.2.1.4	Eccentricity, Radius and Diameter	90
6.2.2	Network segregation parameters	90
6.2.2.1	Clustering Coefficient	91
6.2.2.2	Transitivity	91
6.3	Algorithmic optimisation of the computational burden of the parameters	91
6.3.1	Clustering Coefficient Optimisation	92
6.3.2	Transitivity	94
6.3.3	Density	94
6.4	Hardware Architecture	94
6.4.1	PLI Architecture	94
6.4.2	Clustering Coefficient	94
6.4.3	Degree and Weighted degree	95
6.4.4	Density	95
6.4.5	Transitivity	96
6.4.6	Characteristic Path Length	96
6.4.7	Eccentricity, Radius, and Diameter	97
6.5	Validation and Results	98
6.5.1	Validation data	98
6.5.2	Functional Validation	98
6.5.2.1	Error calculation	98
6.5.2.2	Time calculation	99
6.5.3	Hardware savings	100
6.5.4	Synthesis Results	100
6.5.5	Power Consumption	102
6.5.6	Comparison	103
6.6	Conclusions	105
7	Preliminary Validation of a Seizure Classifier	107
7.1	Introduction	107
7.2	Materials and methods	108
7.2.1	Data and pre-processing	108
7.2.2	Classification algorithms	109
7.2.2.1	Random Forest	109
7.2.2.2	LightGBM	110
7.2.2.3	SVM	110
7.2.3	Methodology	111
7.2.3.1	Feature calculation	111
7.2.3.2	Data split for training and testing	112
7.3	Experimental Results	112
7.3.1	Performance metrics	112
7.3.1.1	Random Forest	113
7.3.1.2	LightGBM	114
7.3.1.3	SVM	115
7.3.2	Comparison	117
7.4	Conclusions	118

8	Conclusions and future work	121
8.1	Future Works	121
8.2	Final Thoughts	123
	References	125

List of Figures

1.1	Overall schematic of the proposed neurofeedback wearable system. . . .	3
2.1	The general structure of the brain.	10
2.2	Diagram of the neuron structure and its synapse junction.	10
2.3	Refractory period for a neuron activation.	12
2.4	Electrode placement based on the 10-20 system.	13
2.5	Block diagram of the EEG acquisition.	14
2.6	Absorption spectrum in near-infrared (NIR) window.	16
2.7	Example of a DWT using a level 3 decomposition.	22
2.8	DWT frequency spectrum sub-band splitting.	23
2.9	Signal reconstruction using the IDWT for a level 3 decomposition. . . .	23
2.10	Flowchart of the EMD algorithm.	25
2.11	Functional and Effective connectivity matrix.	27
3.1	Block diagram of the complete neurofeedback system.	37
3.2	Design flow followed in this work, based on the V-model used in agile software development.	39
4.1	Mean ASR comparison using the Dmey wavelet against using different wavelet functions.	49
4.2	Mean ASR Comparison for the EMD using different windows sizes. . . .	52
4.3	Comparison of the mean ASR performance using different number of iterations for the sifting process.	53
4.4	Power spectrum comparison using the proposed rejection criteria. . . .	55
4.5	Architecture of the WPT module composed of the decomposition and reconstruction modules.	56
4.6	General architecture of the EMD module.	59
4.7	Comparison of the RMSE after cleaning the simulated eye blink artifact. .	61
4.8	Comparison between the original and corrupted signal against the cleaned signal.	62
4.9	Comparison in the power spectrum between the signals.	63
4.10	Visual comparison between a corrupted signal (muscle artifacts) and the processed output signal using the algorithm in software.	64
4.11	Comparison of the performance achieved using the algorithm from Bono et al. (2016) against the modified algorithm presented in this work. . . .	65
5.1	Diagram of the computational flow used for the calculation of the PLI index.	73

5.2	Mean RMSE across all patients for the Hilbert transform calculated with the Fourier transform and the Hilbert transformer filter.	75
5.3	RMSE of the Hilbert transform calculated with the Fourier transform and the Hilbert transformer filter.	76
5.4	The Atan2 calculation is used to calculate the angle ϕ of a given point (X,Y) in the Cartesian plane.	76
5.5	Estimation of the number of elements required for the lookup table based on the equation Equation 5.5.	78
5.6	Block diagram of the hardware architecture proposed for the PLI calculation.	79
5.7	Block diagram of the output from the PLI calculation modules. The output from these modules is used to create the final PLI matrix.	82
5.8	RMSE comparison between the four patients and 50 windows.	83
6.1	Block diagram of the general architecture for the PLI and the graph-theoretic measurements calculation.	92
6.2	Block diagram of the hardware architecture proposed for the clustering coefficient.	95
6.3	Block diagram of the hardware architecture proposed for the Density. . .	96
6.4	Block diagram of the hardware architecture proposed for the Transitivity. .	96
6.5	Block diagram of the hardware architecture proposed for the characteristic path length.	97
6.6	Histogram with the relative error results from the functional validation for every graph-theoretic parameter calculated for the 600 windows and the 10 subjects.	99
6.7	Calculation time and the number of NAND gates required for the design with multiple clustering coefficient modules.	105
7.1	Diagram of a random decision forest.	109
7.2	Comparison between the tree grow using level-wise and leaf-wise. . . .	110
7.3	Example of a linear separation using the SVM algorithm.	111
7.4	Block diagram of the process followed in the design of the classifier used for the validation.	112
7.5	Feature importance ranking obtained with the random forest algorithm. .	116
8.1	Overall schematic of the proposed non-invasive wearable closed-loop neuromodulation system.	122

List of Tables

2.1	Frequency bands on EEG signals.	14
2.2	Comparison between the different techniques used for brain activity recording.	16
2.3	Comparison of properties of the different techniques.	25
2.4	Summary of the graph-theoretic parameters and its relationship with brain functions.	31
4.1	Wavelet family functions used for performance comparison.	49
4.2	Synthesis results.	66
4.3	Clock cycles for calculation.	66
4.4	Logic Gates used for every module.	67
4.5	Comparison with similar designs in terms of resources used.	68
4.6	Comparison with similar designs in terms of architectural complexity.	69
5.1	Compilation results summary.	84
5.2	Acceleration between different number of windows.	85
6.1	Correlation between the clustering coefficient and the transitivity obtained with and without the cubic root.	93
6.2	Number of clock cycles required for the calculation of the different parameters.	99
6.3	Comparison between the one-to-one implementation against the optimized proposed architecture in terms of 2-input NAND gates.	100
6.4	Synthesis results for the different modules of the system and the final design.	101
6.5	Architectural complexity estimation using NAND gates.	102
6.6	Power consumption estimation for every module in the system.	103
6.7	Comparison between the design presented in Pal et al. (2017) and the design presented here.	104
7.1	Performance comparison of the classifiers created using random forest, and the calculation of the features using hardware (HW) and software (SW).	114
7.2	Performance comparison of the classifiers created using LightGBM, and the calculation of the features using hardware (HW) and software (SW).	115
7.3	Performance comparison of the classifiers created using SVM, and the calculation of the features using hardware (HW) and software (SW).	117
7.4	Performance comparison with other seizure detection implementation.	118

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:
Nuno and Maharatna (2019)
Nuno et al. (2021)

Signed:.....

Date:.....

Acknowledgements

I would like to thank my supervisory team, starting with Professor Koushik Maharatna for his invaluable guidance and support during my whole candidature. I would also like to thank Professor Neil White for his helpful advice regarding this work. Also, I would like to thank all the incredible people that I have met during my journey; my friends from Granby Grove, Hulse Road, the Mexican Society and my research group. Although I cannot mention each and every one of you by name, all of you have contributed to this work directly or indirectly, and I will always be grateful for having crossed paths with you. Finally, I would like to thank my family who have always supported me, no matter how hard the challenge is or how far away from home I am.

Abbreviations

AF	Adaptive Filtering
ASR	Artifact to Signal Ratio
BOLD	Blood-Oxygen Level Dependent
CORDIC	Coordinate rotation digital computer
CPL	Characteristic Path Length
DWT	DiscreteWavelet Transform
EC	Effective connectivity
ECG	Electrocardiogram
ECoG	ElectroCorticoGram
EEG	Electroencephalogram
EMD	Empirical Mode Decomposition
EMG	Electromyogram
EOG	Electrooculogram
ERP	Event-Related Potential
FC	Functional connectivity
fMRI	functional Magnetic Resonance Imaging
fNIRS	functional Near-Infrared Spectroscopy
GSI	General Synchronization Index
HT	Hilbert Transform
ICA	Independent Component Analysis
IDWT	InverseWavelet Discrete Transform
IMF	Intrinsic Mode Function
LFP	Local Field Potential
LGBM	Light Gradient Boosting Decision Tree
LUT	Lookup Table
PLI	Phase Lag Index
PLV	Phase Locking Value
RF	Random Forest
SVM	Support Vector Machine
TCS	Transcranial Current Stimulation
WPT	Wavelet Packet Transform
WT	Wavelet Transform

Chapter 1

Introduction

Mental and neurological disorders significantly impact the quality of life of those who suffer from them. According to a report from the World Health Organisation (WHO), one in every four people will suffer from a mental or neurological disorder at some point in their life (WHO (2001)). The problem is so significant that some mental and neurological disorders are now classified among the leading causes of overall disease burden in the world (Vos et al. (2015)).

Mental disorders are defined as an atypical way of thinking or behaving that could generate problems or limitations in a person's life. These disorders can be prevented or be treated with therapy or medication. On the other hand, neurological diseases are characterised for affecting any part of the nervous system; it could be the brain, spinal cord, or nerves. These diseases have no cure, and the only option for treatment is medication to mitigate their effects or risky surgical intervention in more severe cases.

The treatment for these disorders represents a significant economic burden on healthcare institutions. For instance, in the United Kingdom (UK), between 2010 and 2011, the National Health Service (NHS) spent approximately 11.9 billion on mental health disorders and 4.3 billion on neurological services. The accumulative amount for both categories is equivalent to about 15% of the total expenditure during that period (Harker (2012)). Based on this estimate and the current trend, the UK is reaching a point where the treatment cost is not sustainable, and new economic alternatives are necessary.

Furthermore, among mental and neurological disorders, the latter has a more significant impact on the patient's life since there is no cure and the only alternative is the usage of medication. However, even medication is not enough to reduce the symptoms in severe cases. For instance, one-third of patients who have epilepsy or schizophrenia are resistant to medications, as are 20-30% of people with a major depression disorder (Johnston et al. (2019); Vita et al. (2019); Schmidt and Löscher

(2005)). This medication resistance clearly shows a limitation in the current treatments, so it is crucial to explore new alternative treatments to overcome this.

A promising alternative treatment regime explored in recent times is Transcranial Current Stimulation (TCS). The fundamental methodology of this technique involves applying a weak electrical stimulus at the scalp to modulate the excitability of the neurons and thereby influencing brain activity. It is an inexpensive and non-invasive technique that has shown potential in the treatment of mental disorders such as depression and psychosis (Yokoi et al. (2018)), and neurological disorders like epilepsy (San-juan et al. (2015)), Alzheimer (Boggio et al. (2009)), and Parkinson's (Benninger et al. (2010)). Additionally, TCS has shown promising results for treating specifically drug-resistant patients with neurological disorders such as major depression (Ferrucci et al. (2009)), epilepsy (Tekturk et al. (2016)) and schizophrenia (Brunelin et al. (2012)). Thus, TCS can be used as a new alternative that could help treat mental and neurological disorders at a lower cost.

However, despite various studies reporting promising results in the usage of TCS as a possible treatment regime for different disorders, there are still significant doubts about its effectiveness and reproducibility, as shown by several studies such as Regner et al. (2018) and Tortella et al. (2015). This is mainly attributed to the lack of underpinning theory for the selection of different stimulation parameters (current dose, stimulation time, type of current to be applied, numbers of electrodes to be used and application point). Consequently, there is no consensus on the stimulation protocol, making its current application arbitrary at best. This arbitrary use of the TCS could be problematic since an inappropriate stimulus application could be counterproductive when treating diseases. Therefore, it is necessary to use a different approach for this technique, finding a more appropriate way to apply the electrical stimulus based on a set of specific criteria.

In recent years, analysis of functional brain connectivity has been more widely used to understand information propagation among different brain areas and thus to qualitatively characterise the interaction mechanisms among different parts of the brain. Subsequently, atypical functional brain connectivity networks have been identified for different diseases phenotypes (Zhang et al. (2018)). Therefore, studies such as the one presented by To et al. (2018) suggest that the characteristics of the functional brain connectivity networks could be adopted as a reliable set of parameters for guiding the application of the TCS.

1.1 Motivation

The possibility of functional brain connectivity-guided neuromodulation, along with the availability of wearable EEG devices, opens up the scope for real-time closed-loop

neuromodulation that can enable the appropriate type of neuromodulation to be applied anytime, anywhere. This would allow a more efficient treatment of several mental and neurological conditions at a lower cost and reduce the current economic burden in treating these disorders. As a specific example, such a system could allow continuous monitoring of functional brain connectivity in an epileptic patient to early detect the onset of an epileptic seizure. It could then apply stimulant currents in real-time with the appropriate dose and spatial-temporal distribution manner breaking the unwanted synchronisation between different brain areas the main cause for an epileptic seizure and thereby stopping the seizure from occurring. The same methodology could be adapted for treating patients with cognitive difficulties by modulating the neural activities in real-time. This approach could bring a step-change in the non-invasive treatment of mental and neurological diseases.

Nevertheless, to enable this closed-loop neuromodulation, it is necessary to calculate and characterise the functional brain connectivity networks in real-time to provide a neurofeedback signal that can then be used to guide the stimulus applied. As discussed by [García-Prieto et al. \(2017\)](#), there are different software alternatives to perform the calculation of the functional brain connectivity, but most of these are done offline after the complete EEG signal is acquired. Even more, the few optimised options for real-time implementation are still software alternatives that need to run on a powerful computer, limiting the usage of neuromodulation inside the clinic. Therefore, to remove this significant constraint, it is necessary to create a new system that is able to perform the calculation and characterisation of the functional brain connectivity networks in real-time without a computer and can be used as a portable device to provide the neurofeedback signal continuously in any place.

The conceptual framework for this neurofeedback system is shown in Figure 1.1. This system acquires and analyses multichannel Electroencephalography (EEG) signals in order to extract the functional brain connectivity networks. Then, a graph-theoretic approach is used to derive the network characteristic parameters and provide a neurofeedback signal that could be used to monitor the patient's brain activity.

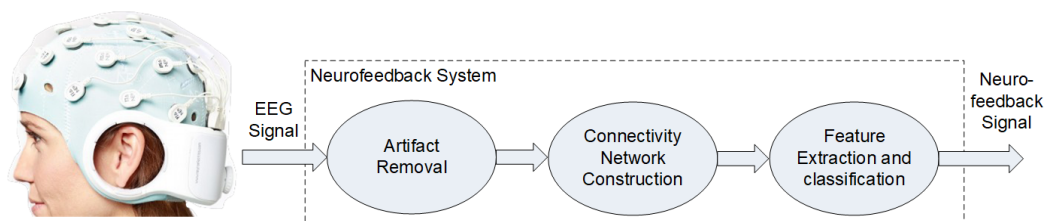


FIGURE 1.1: Overall schematic of the proposed neurofeedback wearable system. The Electroencephalography (EEG) cap used in this illustration is the Starstim EEG system from Neuroelectronics that offers a wearable EEG acquisition system. This device can be complemented with our designs for automated EEG processing in a portable device to monitor the patient on the go.

The first steps in this system are the cleaning of the EEG signal and the construction of the brain connectivity network. Typically, these processes are done entirely offline after the study has been finished. This does not represent a problem in traditional studies, but offline processing is not applicable in the system that we envision. Both of these processes are computationally demanding and require a significant amount of time to be completed using traditional offline processing, which defies the very requirement of such a system. Therefore, there is a significant need for making this whole operation very fast (real-time), and thus it is necessary to implement the whole system in hardware.

However, even if the system is implemented in hardware, it will be rendered useless if it cannot operate in a mobile environment as the principle behind this system is to provide continuous monitoring of the patient's brain activity. Thus, for the practical benefit, such a system should be implemented as a wearable system. In this way, it will be possible to continuously monitor the patient's brain activity and provide the right stimulus if necessary.

Therefore, the hardware implementation will be constrained by two main factors: the area and the energy consumption. The hardware implementation should be small enough to be portable and be used as a wearable system. This guarantees that the patient's comfort and mobility will not be reduced. Furthermore, it is necessary to reduce the energy consumption of the hardware implementation as much as possible since portable systems are powered by a battery with limited energy. Thus, energy-efficient implementations are necessary to guarantee a continuous operation of the system for a prolonged time.

1.2 Aims and Objectives

As previously discussed, the usage of connectivity driven neuromodulation along with the wearable EEG technology open the possibilities to new cost-effective treatments for different mental and neurological disorders that could be used anywhere. This will require a system that is compact and energy efficient to be used as a wearable device, but also be fast enough to perform the calculations in real-time. Therefore, the principal aim of this work is to develop the low energy hardware architecture for the calculation and characterisation of the brain connectivity networks using the graph-theoretic parameters from the EEG signals in real-time, providing a neurofeedback signal that could be used by other systems (i.e. a neuromodulation system).

This thesis's objectives were based on the schematic presented in Fig. 1, focusing on the stages needed to calculate the graph-theoretic parameters. Following a modular

approach, it was decided that every system stage could be considered an independent objective. Thus, the objectives of the thesis are as follows:

- Develop a hardware module to remove the artifacts in EEG signals: EEG signals are highly susceptible to corruption, especially in mobile environments where there exists a greater degree of freedom of physical movements. Removing these artifacts before the signal is processed is essential to avoid a wrong estimation of the brain connectivity networks. Most of the artifact removal algorithms in the literature are performed offline or not designed for mobile environments, which is not consistent with the requirements of the system and therefore, it is necessary to rethink these artifact removal algorithms and perform a proper mapping into hardware to satisfy the real-time and low power requirements.
- Develop a hardware module for functional brain connectivity network construction: After the signal is clean, a proper calculation of the brain connectivity networks can be done. The current approach for the calculation of these networks is entirely offline and therefore it was necessary to adapt this algorithm into a hardware implementation that could perform the calculations in real-time while still having a low power consumption.
- Develop a hardware module for the extraction of quantitative features for network characterisation: Network integration and segregation features can be extracted from brain connectivity networks. These features characterise the brain connectivity networks and provide more information about their behaviour. Both features are highly computation-intensive calculations and therefore they are performed offline. Hence, a different approach should be taken where the fundamental mathematical operations required by the algorithm need to be simplified, allowing real-time processing while having a minimal impact on the accuracy.
- Validate the impact of the calculation of the graph-theoretic parameters in hardware and their viability as a neurofeedback signal: Perform a preliminary validation to evaluate the impact of the graph-theoretic parameters' hardware calculations against the same calculations done offline in software. This analysis will demonstrate that the hardware architecture designed in this work is a viable option that could be used as an alternative for traditional offline software processing.

1.3 Contributions

The major contributions of this thesis are the following:

- Development of a real-time artifact removal module based on hybrid WPT-EMD algorithm for multichannel wearable wireless EEG system. This hybrid algorithm has been previously used to remove artifacts in mobile environments. However, this algorithm is performed completely offline and therefore a redesign was necessary. This algorithm was improved to remove conceptual problems and to enable real-time processing. This latter was achieved by performing an analysis of the different governing parameters of this algorithm to reduce its computational complexity while having a minimal impact on the performance. The algorithm was implemented in an FPGA as a proof of concept showing its ability to remove the artifacts from EEG signals in real-time with similar performance to its software counterpart.
- Development of a module to calculate the functional brain connectivity of an EEG signal using the phase lag index technique. The calculation of functional brain connectivity requires obtaining the analytical signal, which involves extensive computational calculations such as the Hilbert Transform and the arctangent function. Different alternatives were explored to find the most efficient method to perform these calculations, reducing its complexity and allowing real-time processing. The designed hardware architecture was implemented in an FPGA as a proof of concept and was able to calculate the functional brain connectivity networks at least 66 times faster than the software toolbox.
- Development of a module to extract graph-theoretic parameters from functional brain connectivity networks derived from an EEG signal. Similarly to the previous modules, the calculation of the graph-theoretic parameters was optimised to avoid computationally extensive calculations and reduce the number of operations needed, reducing the total calculation time for real-time processing. The algorithm was implemented in an FPGA as a proof of concept and was able to calculate the graph-theoretic parameters in real-time with a minimal error compared to the software toolbox commonly used for offline processing.
- Epileptic seizure detection based on functional brain connectivity and graph connectivity parameters. To validate the impact of the hardware calculations and to show the potential of graph-theoretic parameters as a neurofeedback signal, in this work, we created a seizure classifier using three different machine learning techniques. The classifiers were trained using the graph-theoretic parameters calculated on software and with the modules presented in this work, using the EEG signals from 24 subjects suffering from epileptic seizures. The best classifier achieved a mean sensitivity and specificity above 99% using either the hardware or software calculations of the graph-theoretic parameters. These results demonstrate the minimal impact of the hardware calculations on the final

application and its viability as an alternative to traditional offline software processing.

Some of the contributions mentioned above have been published as follows:

Journal:

- Gutierrez Nuno, Rafael Angel, Chi Hang Raphael Chung, and Koushik Maharatna. "Hardware architecture for real-time EEG-based functional brain connectivity parameter extraction." *Journal of Neural Engineering* (2020).

Conference:

- Gutierrez Nuno, Rafael Angel, and Koushik Maharatna. "A phase lag index hardware calculation for real-time electroencephalography studies." In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 644-647. IEEE, 2019.

1.4 Thesis Outline

In Chapter 2, a background is provided, followed by a literature review of the different methods used to monitor brain activity, including electroencephalography. Furthermore, this literature review also covers the different techniques used for the removal of artifacts in electroencephalography signals and a comparison among them; the different techniques used to study the brain connectivity in EEG signals such as phase lag index, phase-locking value, and Synchronisation likelihood; and finally, the usage of graph-theoretic parameters to characterise the brain connectivity networks such as clustering coefficient, degree, transitivity, among others. Next, a discussion about the system's specifications and the design approach followed in this work is presented in Chapter 3. Then, in Chapter 4, a new algorithm to remove the artifacts in wearable EEG signals is presented. This algorithm implements a combination of WPT and EMD techniques which can remove the particular artifacts found in wearable EEG systems. In Chapter 5, a new hardware architecture to calculate the functional brain connectivity network from EEG signals is proposed. The architecture performs the calculation of the brain connectivity matrices from an EEG signal to measure the synchronisation of the different regions of the brain. Later, in Chapter 6, a new architecture for a hardware calculation of the graph-theoretic parameters from functional brain connectivity matrices is presented. This architecture is combined with the architecture presented in Chapter 5 to create a complete system that performs the functional brain connectivity network estimation and the graph-theoretic parameter

calculation to characterise such networks. Chapter 7 contains an implementation of a seizure detection classifier created as a proof-of-concept and validates the minimal impact of the calculations done with the architecture presented in Chapter 6 compared against the same calculations on software. Finally, conclusions of the work done in this thesis are found in Chapter 8, followed by a brief discussion of future work for this project.

Chapter 2

Background and Literature Review: Electroencephalography, Artifact Removal and Functional Connectivity

2.1 The physiology of the brain and electroencephalography

2.1.1 Brain activity

The human brain is composed by three main components, the brain stem, the cerebellum and the cerebrum (Figure 2.1). The brain stem is the structure that connects the brain with the spinal cord and its main function is to serve as a channel for the signals from the spinal cord to the brain and vice-versa. The cerebellum also is responsible for transmitting information between the spinal cord and the brain but is more focused on the signals responsible for balance, posture and the control of muscles. Finally, the cerebrum is the biggest part of them all and it is composed by neuronal cells (neurons) and non-neuronal cells like the glia which holds the neurons in their place and supply the necessary nutrients for them to function. The cerebrum is the most important structure since it is the one that controls all the different cognitive processes such as thoughts, reasoning, movement, visual processing and others.

Neurons are considered the most fundamental component of the human brain, and their principal function is the transmission of information. The neurons work and communicate with each other using a complicated process involving chemicals and electrical signals. When a neuron is activated by an external stimulus like a signal from the nervous system, it generates an electrochemical pulse that travels through

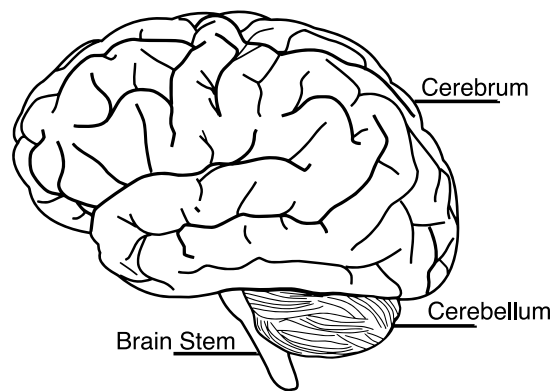


FIGURE 2.1: The general structure of the brain.

the axon connected to other neuron dendrites. The junction between the axon and the other neurons' dendrites is known as a synapse (Figure 2.2), and it is a junction that allows transmission of the electrochemical pulse generated by the activated neuron. The chemicals sent between neurons are called neurotransmitters, and these could be either excitatory types that help activate the receiving neuron or inhibitory type that stops the receiving neuron from activating. These interactions between groups of neurons across the brain are responsible for the different brain dynamics involved in cognitive tasks (Nunez and Srinivasan (2006)).

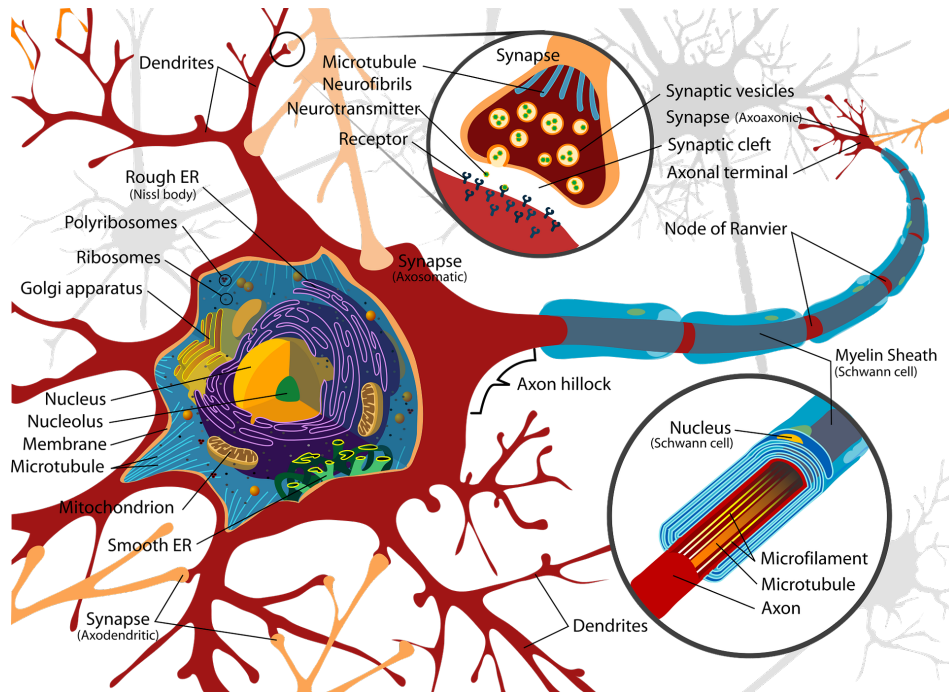


FIGURE 2.2: Diagram of the neuron structure and its synapse junction.

2.1.2 Generation of the action potential

The voltage between the inside and the outside of the cell is known as the membrane potential. The neuron membrane potential in a resting state is usually polarized to around -70 mV (Dario-Becker (2013)). This potential is generated by the difference in the ion concentration between the inside and the outside of the cell, which creates a negative electric charge inside the cell and a more positive electrical charge on the outside. This difference in ion concentration is caused by the so called "sodium-potassium pump", which is continually pumping Na^+ (Sodium) ions out of the cell and moving K^+ (Potassium) ions inside the cell. The pump moves 3 Na^+ ions outside the cell and allows 2 K^+ ions to enter the cell. Therefore, the number of Na^+ ions pumped outside the cell is greater than the number of K^+ ions entering the cell, making the cell more positive outside than inside. This resting potential remains constant until the neuron receives a stimulus.

When a stimulus is applied to the neuron, there is an aperture of the sodium channels in the neuron's membrane, which lets Na^+ ions rush inside the neuron, and depolarises the neuron, increasing its membrane potential. If the stimulus is strong enough, and the membrane potential reaches the excitation voltage threshold of -55 mV, this will trigger the generation of the neuron's action potential. Once this threshold is reached, more Na^+ channels start to open, allowing more ions to go inside the cell. This rush of ions overshoots the membrane potential until it reaches the peak voltage at 30 mV. This peak of voltage is known as the peak action potential.

After the action potential is reached, the sodium channels close, preventing any more sodium ions from entering the cell, and the potassium channels now become open. The K^+ ions now move outside the cell, producing a cell's repolarisation to reach the resting membrane potential again. These potassium channels remain open for more time than needed which causes a hyper-polarization of the neuron, reaching a membrane potential even lower than the resting potential. Here the neuron starts to move Na^+ outside the cell and K^+ ions inside the cell until the resting membrane potential is reached again. Once the resting potential is reached, the neuron can be activated again. This entire process is depicted in Figure 2.3.

The generation of this action potential has been a practical method to monitor the brain activity used in various studies. This action potential allows identifying which particular groups of neurons are activated during a specific cognitive task, providing more information about the different brain regions' interactions.

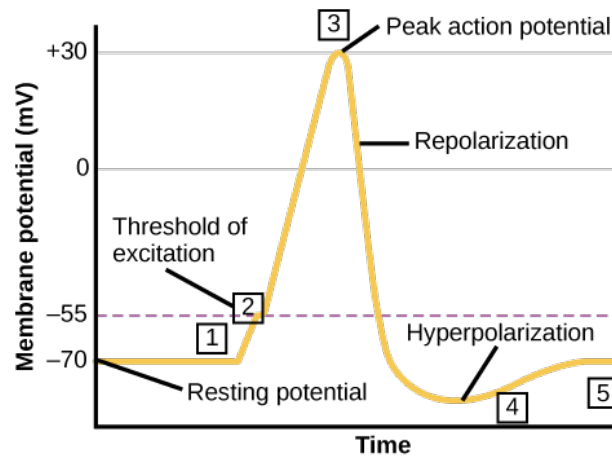


FIGURE 2.3: Refractory period for a neuron activation. Extracted from [Dario-Becker \(2013\)](#).

2.1.3 Brain activity monitoring

In the literature, the two main different physical properties used to record the brain's neuron activity are the electrical potential and the blood's oxygen levels. These two properties are dependent on the neurons' activation, and therefore they could be used to detect if a specific group of neurons in the brain have been activated or not. This subsection presents a discussion on the different techniques that use the principles previously mentioned to monitor brain activity.

2.1.3.1 Electroencephalography (EEG)

The electroencephalogram (EEG) is a technique used to record the electrical activity originated by the action potential of a large group of neurons, using an array of electrodes placed at the patient's scalp. The usage of this technique could be dated to the year 1929 ([La Vaque \(1999\)](#)) and has become a widely used tool in clinical studies for different neurological conditions such as epilepsy, traumas, sleep disorders, strokes and others (([Nunez and Srinivasan, 2006](#))).

The electrodes used for EEG are usually metallic discs (commonly made of Silver/Silver Chloride (Ag/AgCl)) placed in conjunction with a conductive medium that keeps the electrode in place and reduces the impedance between the sensor and the skin, improving the readings. These electrodes may not be the most ideal for the patients since they involve a long setup time, and on few occasions, it even requires that the patient shave the head to make more proper contact with the skin. In literature there are reports of contactless sensors ([Hazrati et al. \(2013\)](#), [Chi and Cauwenberghs \(2010\)](#) and [Chi et al. \(2009\)](#)) that could be used as an alternative. However, these sensors' performance depends on the distance to the scalp, so any movement could affect it, making them more unreliable. A more viable option is to

use soft/dry electrodes that do not require the head's shaving or any conductive medium. A sensor with these characteristics have been presented by [Chen et al. \(2014\)](#) and similar designs can be found in commercial wearable EEG systems.

The electrodes' placement is commonly done following the guidelines of the 10-20 system as presented by [Klem et al. \(1999\)](#), which ensures reproducibility of the results. The 10-20 system uses the nasion, inion, and pre-auricular point to reference the electrodes' placement. The electrodes are placed with a separation distance of 10% between these points of reference and 20% between each electrode. This array is shown in Figure 2.4, resulting in a total of 19 electrodes. Furthermore, there are other electrode placement systems such as the 10-10 or the 10-5, both variations of the 10-20 system. These systems reduce the distance between electrodes from a 20% to a 10% (10-10 system) or 5% (10-5 system). Such systems would allow a more dense electrodes placement, allowing up to 329 electrodes for the 10-5 system ([Jurcak et al. \(2007\)](#)). Although using a 10-10 or a 10-5 system could increase the system's spatial resolution, the large number of electrodes will compromise the comfort of the patient and require a longer setup time. Therefore, these larger arrays of electrodes are often reserved for specific studies, such as electric source imaging for surgical procedures.

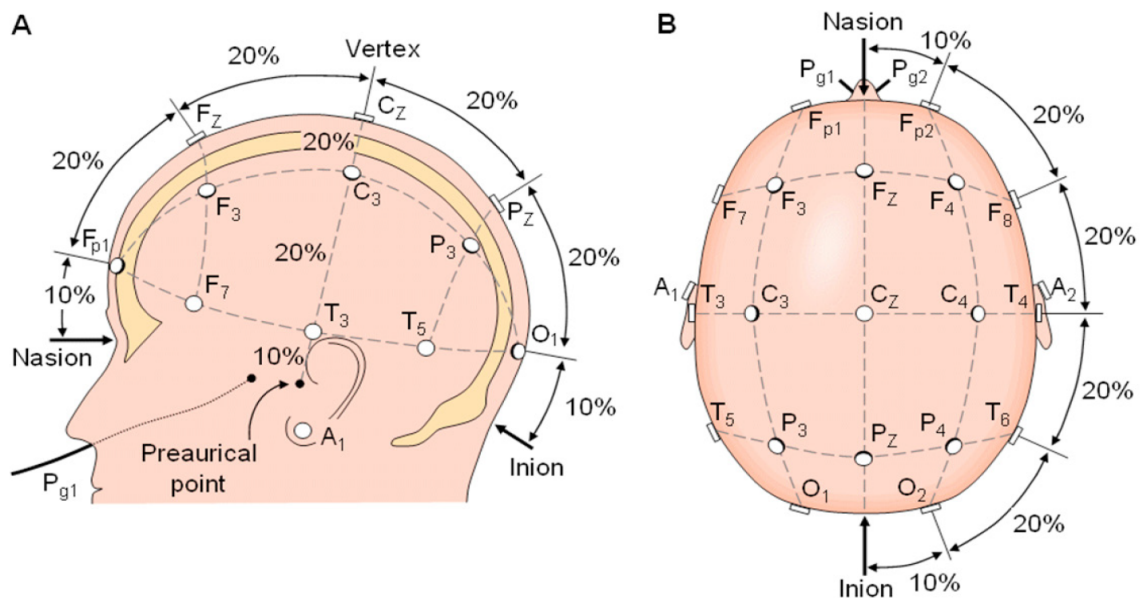


FIGURE 2.4: Electrode placement based on the 10-20 system. Extracted from [Bhavsar et al. \(2018\)](#).

The approach used to measure the electrical signal with the electrodes is similar for most implementations and can be summarized as in the block diagram in Figure 2.5. First, the electrodes are connected to a differential amplifier which amplifies the small electrical signal generated by a group of neurons (usually between $0.5\text{-}100\ \mu\text{V}$). Then, a bandpass filter is used to filter all the frequency components that are not related to brain activity, keeping only the components between 1.5 and 60 Hz. Finally, the signal

is introduced into an Analogue to Digital Converter (ADC) that sends the digital signal to a computer where it can be stored for further processing.

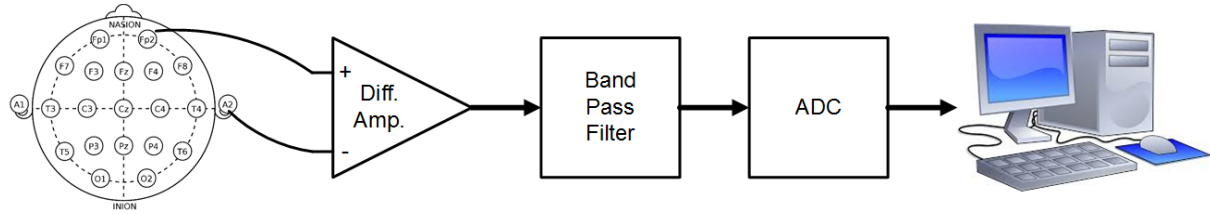


FIGURE 2.5: Block diagram of the EEG acquisition.

Usually, the EEG signals are divided into five frequency bands related to a different cognitive task, as shown in Table 2.1. By analysing the EEG signal's power spectrum, it is possible to identify the dominant frequency band and thus identify the subject's brain state. Furthermore, the same analysis could be used to identify any abnormality in the power spectrum while performing a cognitive task, indicating a possible problem with the patient (Van der Hiele et al. (2007)).

TABLE 2.1: Frequency bands on EEG signals.

Band	Freq. (Hz)	Related cognitive tasks	Possible problems due abnormal levels
Delta	1.5 - 3.5	Deep relaxation and sleep	Poor sleep, low energy due poor sleep, brain injuries, etc.
Theta	3.5 - 7.5	Sleep and emotions	Hyperactivity, poor emotional awareness, depression, etc.
Alpha	7.5 - 12.5	Relaxation	Inability to focus, insomnia, OCD, etc.
Beta	12.5 - 20	Conscious thought and logical thinking	Adrenaline, anxiety, daydreaming, poor cognition, etc.
Gamma	20 - 60	Learning, memory and information processing	Anxiety, stress, depression, learning disabilities, etc.

2.1.3.2 ElectroCorticoGram (ECoG)

This technique measures a group of neurons' electrical activity using electrodes placed in the cortical surface of the brain. This method presents a very high spatial resolution in the order of millimetres and timing resolution in the order of milliseconds (Hill et al. (2012)). However, it is also an invasive method often used only for particular treatments, such as the localisation of the epileptogenic area for patients with epilepsy (Kuruvilla and Flink (2003)).

2.1.3.3 Local Field Potential (LFP)

This technique is used to measure the electrical activity of a small group of neurons in a range of 200~400 μm (Kajikawa and Schroeder (2011)). This measure is done using

wire or micro-machined silicon (Buzsáki (2004)) directly placed in the cortex of the brain, and thus the spatial and temporal resolution is excellent. Nevertheless, this makes it a highly invasive method, limiting its usage.

2.1.3.4 functional Magnetic Resonance Imaging (fMRI)

This technique can be described as a variation of the Magnetic Resonance Imaging (MRI) technique. MRI uses powerful magnets to generate a strong magnetic field (between 0.5-1.5 T) to generate an image of organs in the body. In the absence of a magnetic field, the spin axes of the hydrogen protons (found in the water and fat of the organs) are randomly aligned. When an external magnetic field is applied, the spin axes align according to the magnetic field. After spin axes are aligned, the magnetic field is removed, and the spin axes go to their original position emitting a radio signal. This radio signal is detected with coils placed around the patient, and it is used to construct an image able to discern between different types of tissues (Berger (2002)). In fMRI, the process is similar, but instead of using the spin axes of the hydrogen protons, the properties of the blood in the brain are used. The neurons in the brain require a constant supply of oxygen like every other cell. When the neuron is activated, a greater amount of oxygen is needed to produce the necessary energy. Therefore, it is possible to observe a reduction in the deoxyhemoglobin (HbR) in the blood while the oxyhemoglobin (HbO) increases. The HbO is a diamagnetic material, and therefore its spin axes are not aligned in the presence of a strong magnetic field, while the HbR is a paramagnetic material, and its spin axes do align when a magnetic field is present (Ogawa and Lee (1990)). Then, when a strong magnetic field is applied, only the spin axes of the HbR will be aligned and emit a radio signal when this field is removed. The radio signal emitted will be detected by the coils around the patient's head and used to create an image of which areas of the brain are being activated.

2.1.3.5 functional Near-Infrared Spectroscopy (fNIRS)

This method uses optical sensors that work close to the infrared spectrum of light between 700-900nm (Izzetoglu et al. (2007)), where most of the tissues are transparent. Also, in this spectrum, oxyhemoglobin (HbO) and deoxyhemoglobin (HbR) are excellent absorbers of light, a property that can be used to measure the brain activity. For this technique, an array of photo-transmitters and photo-receivers are placed at the scalp of the patient. The light transmitter uses two different wavelengths in this infra-red spectrum of light. The first is an 805nm wavelength in which the absorption factor of the HbO and HbR are equal, as shown in Figure 2.6. The second wavelength is higher, usually at 850nm, where the HbO absorbs more light than the HbR. By comparing the light scattered at these two wavelengths, it is possible to calculate the concentrations of the HbO and HbR and identify the regions of the brain that are

being activated. This technique's spatial resolution is about 1-10 mm (Quaresima and Ferrari), while the temporal resolution is in the order of seconds.

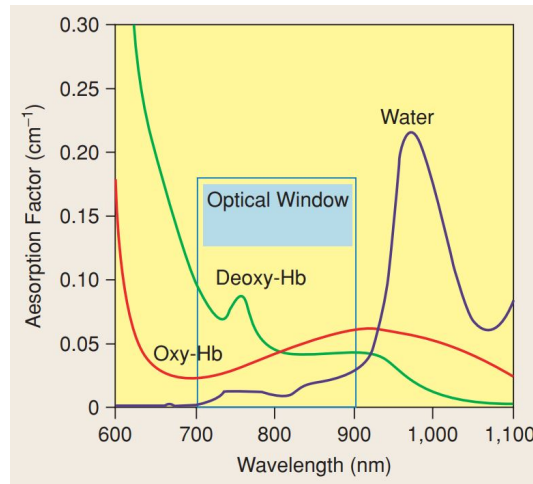


FIGURE 2.6: Absorption spectrum in near-infrared (NIR) window. Extracted from Izzetoglu et al. (2007) ©2007 IEEE.

2.1.3.6 Comparison

After analysing the various alternatives available for brain activity recording, it is necessary to compare each technique's advantages and disadvantages to select the most appropriate technique for the neurofeedback system proposed in chapter 1. A comparison between the electrical potential and the Blood-Oxygen Level Dependent (BOLD) techniques used to record the brain activity are presented in Table 2.2. The points taken into consideration are spatial and time resolution, portability, invasiveness, cost and complexity. These are essential points that need to be considered for the proposed system.

TABLE 2.2: Comparison between the different techniques used for brain activity recording.

Property	Technique	Spatial Res.	Time Res.	Portable	Invasiveness	Cost	Complexity
Electrical potential	EEG	Low	High	Yes	Low	Low	Low
	ECoG	Med.	High	Yes	High.	High	High
	LFP	High	High	No	High	High	High
BOLD	fMRI	High	Low	No	Low	High	High
	fNIRS	Med.	Low	Yes	Low	Low	Low

The spatial resolution required will entirely depend on the implementation. For instance, the high spatial resolution of the LFP technique has been used to identify surgical targets that can be used for deep brain stimulation in the treatment of Parkinson's disease (Thompson et al. (2014)). Similarly, ECoG has also been used in the identification of surgical targets for temporal lobe epilepsy (Hutchings et al.

(2015)). However, this high spatial resolution comes with a trade-off that is their high invasiveness. These methods require a surgical procedure that may not be viable for many studies. Additionally, the high spatial resolution also makes these methods focus on a small brain area. Some studies require a broad coverage of the brain to understand the interactions among the different regions of the brain. For instance, EEG provides a broader coverage at the cost of a lower spatial resolution. Beside this trade-off, EEG is still one of the most popular techniques in the neurological field (Teplan et al. (2002)) due to its low invasiveness and broader coverage.

The temporal resolution is another crucial aspect to consider. The neurofeedback system that we envision needs to provide proper feedback signal at real-time. The only techniques with such time resolution are EEG, ECoG and LFP, which use electrical potential to record brain activity.

Mobility is a crucial aspect of the proposed system since its effectiveness lays in the constant monitoring of the brain to detect when the stimulus is needed. The final device must be portable and unrestrictive, allowing the patient to wear the device and monitor the brain activity while performing their daily life tasks. Thus, techniques such as LFP, ECoG and fMRI are not a viable option, and other methods such as EEG or fNIRS would be a better choice.

Moreover, the use of highly invasive or complex techniques is not ideal for the proposed system. Techniques such as ECoG and LFP require a complex and invasive medical procedure which may not be ideal for many patients. Although less invasive, the fMRI is also a complex technique requiring specialised equipment and properly trained technicians to operate it. Furthermore, the complexity involved in these techniques elevate their total cost and makes them more expensive and less accessible compared with the other techniques. For these reasons, such techniques are not a viable option for the proposed system.

Now that all the techniques have been discussed, it is necessary to select the most appropriate technique for this work. From all the techniques, the first ones to be discarded are techniques related to blood oxygen levels. Both fMRI and fNIRS provide a very high spatial resolution, but they also have a low time resolution which is crucial to provide the stimulus at the right moment. Thus, techniques that use the electrical potential to record brain activity will suit better.

Among the different techniques that use the electrical potential, LFP is a technique that provides an excellent spatial and temporal resolution, but it is complex and highly invasive, and it is not a viable option. ECoG provides a better spatial resolution than EEG, but it is still an invasive method that requires a medical procedure to implant the sensors in the patient's cortex. This procedure is complex, and it is better to be avoided.

Therefore, it was decided that the technique best suited for this work is EEG. This technique covers almost all the aspects that are evaluated here. It is an inexpensive technique with an excellent time resolution. Also, various commercial devices are available at the moment, so it is possible to find a system that can meet the project's requirements. Hence, it was opted to use EEG for the recording of brain activity in this work.

2.1.4 Wearable EEG systems

Nowadays, all the necessary equipment to perform the EEG signal acquisition as depicted in Figure 2.5 could be integrated into a small portable device that the user can wear. These wearable EEG systems have gained attention for their intrinsic advantages compared to limited traditional EEG systems (Sawangjai et al. (2019)). The advances in this area are such that there are many commercial wearable EEG systems in the market. An example of this wearable EEG system is the STARSTIM R20 from Neuroelectronics®, used in the schematic shown in Figure 1.1. The system consists of an elastic cap with all the dry electrodes used to capture the electrical signal. On the rear of the cap, there is a small module that performs the whole process shown in Fig. 2.5. The same module can transmit the EEG signal via Bluetooth to any device. Based on the specifications of the Bluetooth Low Energy (BLE) version 5 protocol, it could be possible to achieve a transfer rate up to 2Mbps (Badihi et al. (2019)) with low energy consumption. Additionally, this system has extra electrodes that can be used to apply an electrical stimulus to the wearer, all controlled by Bluetooth. All of this powered by a battery and enclosed in a small portable device, ideal for the system described in 1.1. Other similar products on the market for wireless EEG systems are the ones developed by OpenBCI that offer a low cost and open-source product, or the products by companies like Neurosky® or Muse™ that offer a more simplistic device with a lower number of channels that have been used in different studies for drowsiness detection (LaRocco et al. (2020)).

Besides all the progress done on EEG systems, there has always been a significant drawback in these systems which is the introduction of artifacts into the EEG signal. The amplitude of the EEG signal is small, making it highly susceptible to corruption from various sources. While this applies to every EEG system, wireless EEG systems are even more susceptible since the artifacts in mobile environments are stronger due to the unrestricted movement of the patient. Therefore it is important to remove those artifacts from the signal before any further analysis is done.

2.2 Artifacts and their treatment in EEG signals

2.2.1 Artifacts in EEG

The introduction of artifacts into the signal is one of the greatest challenges while working with EEG signals. In EEG signals, an artefact is defined as any kind of signal that is not related to the brain activity and hence is not desired. Since the signal from the brain activity is in the order of the μV it is highly susceptible to be corrupted by different kinds of artefacts that could lead to wrong interpretation of signal acquired. The artefacts are classified according to their origin and they are divided into two main groups: Non-Physiological and Physiological artefacts.

2.2.1.1 Non-Physiological artefacts

This category includes all the artifacts that are generated by a source that is not related with any physiological process of the body. Some of the most common examples (Teplan et al. (2002)) of these artifacts are:

- Electrodes detaching
- Dry electrodes with poor conductivity
- Moving wires connected to the electrodes which can affect the connection.
- AC noise from the electrical current

These artifacts can be avoided taking the appropriate precautions while conducting the EEG acquisition. Thus, the removal of these artifacts from EEG signals are seldom found in literature.

2.2.1.2 Physiological artefacts

A physiological artefact is a signal that is produced by any physiological process in the body. Some examples of these artefacts are:

- Cardiac activity: Heartbeat or a pacemaker
- Ocular activity: Eye blinking and eye movement
- Muscle activity: Body movement
- Skin related: Sweating

Some of these artifacts are usually avoided by implementing some restrictions during the EEG acquisition. For instance, the movement of the patient can be restricted to avoid the introduction of muscular artefacts to the signal. However, some physiological artifacts such as cardiac and ocular activity (e.g. eye blinking) cannot be completely avoided. Therefore, it is necessary to develop algorithms that can remove these inevitable artifacts from the EEG signals.

2.2.2 Artifact removal algorithms

One of the usual approaches taken when dealing with corrupted EEG data is to discard the portion of the signal that has been contaminated. Although simple, this approach is not ideal since some important information may be lost. Because of this, there has been a large effort in the creation of different algorithms to deal with the different artifacts in EEG signals.

From all the artifacts mentioned in section 2.2.1, the artifacts related to ocular and muscle activity are the most complicated to remove. Thus, most of the algorithms presented in the literature are focused on these artifacts (Urigüen and Garcia-Zapirain (2015)).

The number of artifact removal algorithms in the literature is vast. Nonetheless, the most used algorithms (based on the reviews from Kandaswamy et al. (2005); Sweeney et al. (2012); Kirkove et al. (2014); Urigüen and Garcia-Zapirain (2015); Rahman et al. (2015), and Jiang et al. (2019)) are based on one of the following techniques: Adaptive filtering, Independent component analysis, Wavelet transform, and Empirical mode decomposition.

2.2.2.1 Adaptive Filtering (AF)

The application of filters to remove specific frequency components in a signal is often used in signal processing. Nevertheless, traditional filtering techniques cannot be used to remove the artifacts in EEG signals properly. The frequency components of the artifacts are widely spread and often overlap the frequencies of the EEG signal (McMenamin et al. (2011)). Thus, filtering the artifacts will also remove the components related to the brain activity.

The adaptive filtering technique overcomes this limitation by using an external signal as a reference to estimate the contribution of the artefacts in the signal. This estimation is used to continually update the coefficients of the filter and avoid over-filtering the signal. The reference signal depends on the artifact, and it is done using an electrode to capture the electrical signal. For instance, it could be an electrooculogram (EOG)

which captures the ocular activity, an electrocardiogram (ECG) for the cardiac activity or an electromyogram (EMG) for muscular activity.

Adaptive filtering is a technique with low computational cost and simple to implement. The main disadvantage of these algorithms is the requirement of a reference signal which may not be available for some applications.

2.2.2.2 Independent Component Analysis (ICA)

The Independent Component Analysis (ICA) is a blind source separation technique that assumes the EEG signal is a linear combination of various statistically independent (orthogonal) sources related to either brain activity or artifacts. Hence, the EEG can be separated into independent components and then reconstructed without the components related to artifacts. This linear combination is represented as in (2.1), where \mathbf{X} corresponds to the acquired EEG signal, \mathbf{A} represents a mixing matrix of the multiple independent components \mathbf{s} .

$$\mathbf{X} = \mathbf{A}\mathbf{s} \quad (2.1)$$

The linear combination of the different independent components to obtain the EEG signal is given by the mixing matrix \mathbf{A} . Therefore, to separate the independent components from the EEG signal, it is necessary to find the inverse of the mixing matrix. Then, when this inverse matrix is multiplied by the EEG signal, the independent components of the signal will be found, as in (2.2). Finding this inverse mixing matrix is a complex process, and thus it is a significant challenge of using this algorithm.

$$\mathbf{A}^{-1}\mathbf{X} = \mathbf{s} \quad (2.2)$$

This algorithm has become one of the most used for artifact removal due to its effectiveness. It has shown promising results removing muscular (Safieddine et al. (2012)) and ocular artifacts (Zhou and Gotman (2009)). The major disadvantages of this algorithm are the high computational cost and its requirement for a large number of sensors. This requirement may not be satisfied in EEG systems with a low number of sensors, such as wearable EEG.

2.2.2.3 Wavelet Transform (WT)

The Wavelet Transform (WT) is a common technique in signal processing used to transform a signal from the time domain into the frequency domain, similar to the Fourier Transform (FT). The analysis in the frequency domain could bring information that was not as evident in the time domain. This is useful with signals that change their frequency over time, such as the artifacts in EEG signals.

The WT is then defined as the inner product between the signal and a wavelet function scaled and translated in time, represented as in (2.3). Here, f represents the signal and Ψ represents the mother wavelet, a represent the scale and b the translation parameter (Urügün and Garcia-Zapirain (2015); Safieddine et al. (2012)).

$$W_{\Psi}f = \langle f, \Psi_{a,b} \rangle \quad (2.3)$$

$$\Psi_{a,b}(t) = |a|^{-1/2} \Psi\left(\frac{t-b}{a}\right) \quad (2.4)$$

2.2.2.4 Discrete Wavelet Transform (DWT)

The Discrete Wavelet Transform (DWT) is a variation of the WT used to transform discrete signals in the time-frequency domain. In the DWT, the transformation is done using a discrete set of the mother wavelet, this being the main difference with the continuous WT. This reduces the computational cost needed to perform the transformation. The DWT is performed by filtering the signal through a series of low and high pass decomposition filters as shown in Figure 2.7 obtaining the detail (high frequency) and approximation (low frequency) coefficients of the decomposition.

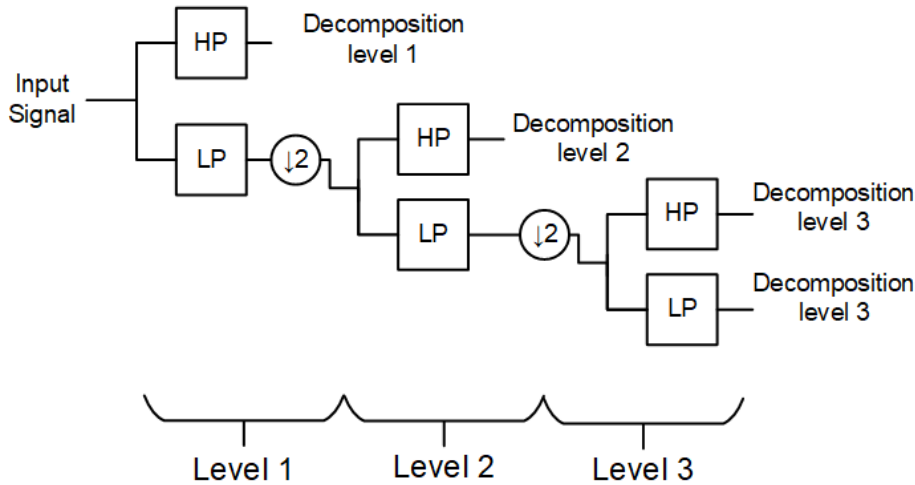


FIGURE 2.7: Example of a DWT using a level 3 decomposition.

This filtering will be equivalent to split the frequency components into two sub-bands from the frequency domain perspective. The filters' output is downsampled and introduced into another high and low pass filter pair, further splitting the sub-bands into smaller bands, as depicted in Figure 2.8. This process is repeated until the desired level of decomposition of the signal is reached. This splitting of the frequency spectrum into smaller sub-bands enables highly selective filtering of the signal, removing only the specific components related to the artefacts.

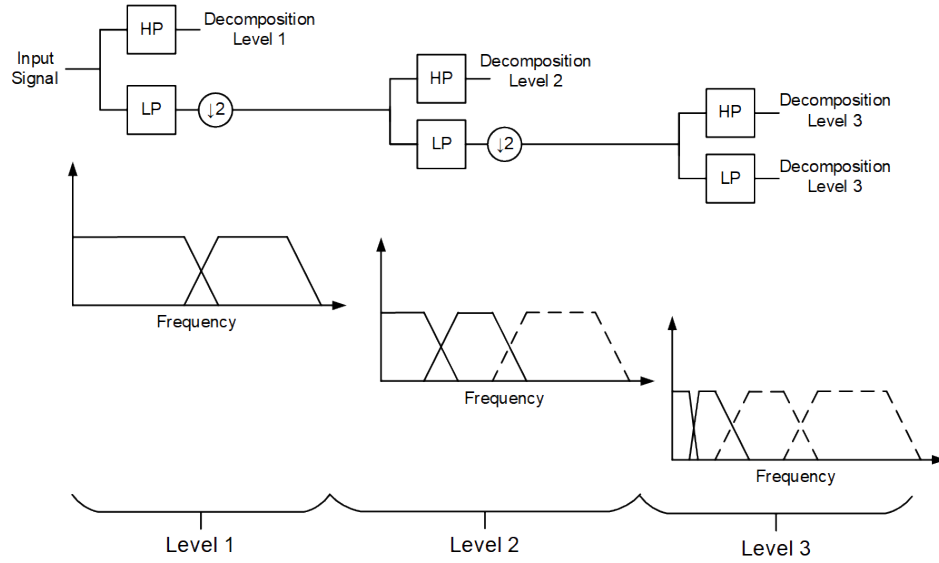


FIGURE 2.8: DWT frequency spectrum sub-band splitting.

After removing the sub-band related to the artifact, the signal is reconstructed by performing the Inverse Wavelet Discrete Transform (IDWT). This process consists of upsampling the signal obtained from the decomposition and introducing them into the reconstruction high and low pass filters, as shown in Figure 2.9. The final result is a reconstructed clean EEG signal without the artifacts components.

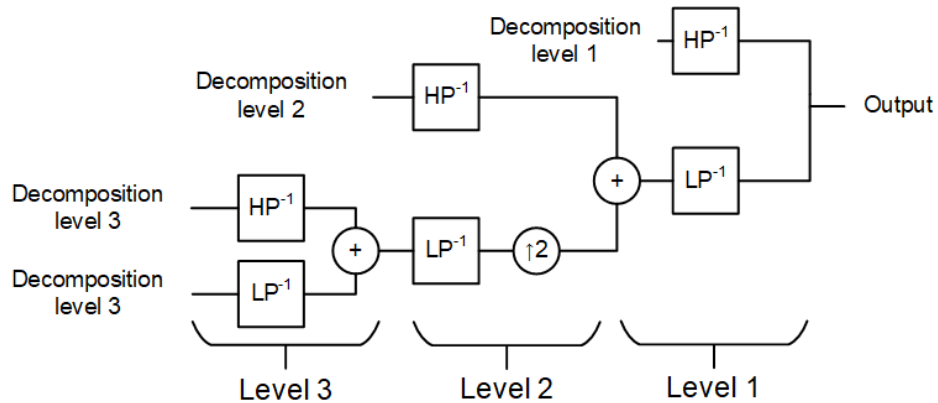


FIGURE 2.9: Signal reconstruction using the IDWT for a level 3 decomposition.

The DWT has been used in the removal of ocular artifacts (Krishnaveni et al. (2006)), but is more often seen in combination with other techniques (e.g. ICA (Mammone

et al. (2011))). The advantage of this technique is its low complexity since it can be implemented using only Finite Impulse Response (FIR) filters.

2.2.2.5 Empirical Mode Decomposition (EMD)

The Hilbert-Huang transform is a method used to decompose the signal into multiple components known as Intrinsic Mode Functions (IMF) (Huang et al. (1998)). The IMFs need to satisfy two main criteria:

- The number of zero-crossings and the total number of extremas (maxima and minima of the signal) must be equal, or different by one.
- The IMF must have a local mean of zero.

This decomposition is represented in (2.5), where x represents the acquired signal, s represents the k number of IMFs and r represents the remainder of the decomposition.

$$x(t) = \sum_{k=1}^K s_k(t) + r(t) \quad (2.5)$$

The IMFs are extracted from the signal by performing the Empirical Mode Decomposition (EMD). The flowchart of the EMD is depicted in 2.10. It consists of a process called sifting, which is repeated until the signal is considered an IMF. Then the IMF is subtracted from the original signal, and the resulting signal is now used in the sifting process. This sifting and subtraction are repeated until no further IMFs can be extracted. The remaining signal is called the residual signal r .

This technique is particularly good with signals that are non-linear and non-stationary such as EEG signals. It has shown promising results in the removal of muscular (Safieddine et al. (2012)) and ocular (Molla et al. (2010)) artifacts. The main disadvantage of this technique is the complexity of the process involved in the extraction of the IMFs.

2.2.3 Comparison of the algorithms

Although all the techniques discussed have shown successful results in removing artifacts from EEG signals, it is important to discuss the advantages and disadvantages of these techniques to select the most appropriate for the application in mind. Here, the following properties are considered crucial for the system described in Chapter 1: 1) extra reference signal, 2) complexity of the technique, 3) the number of

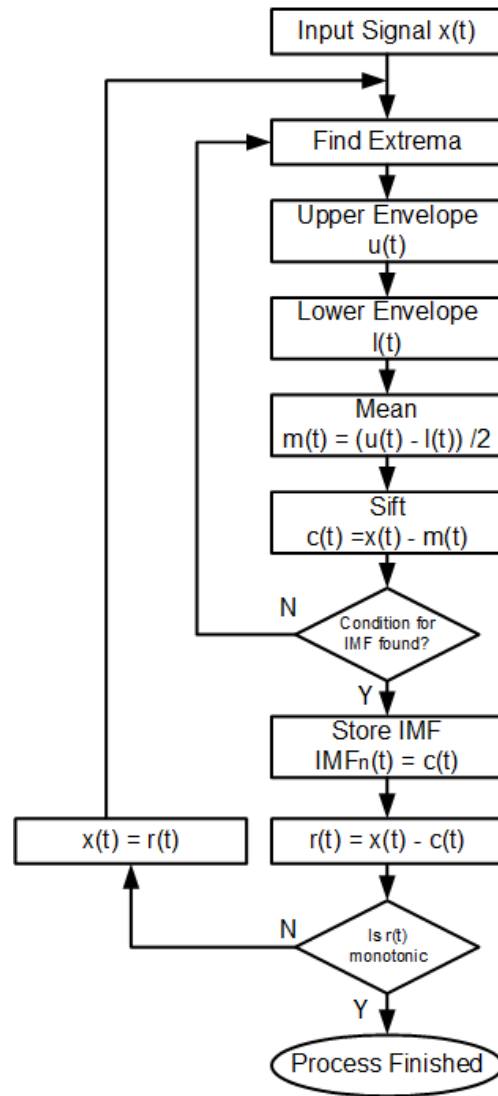


FIGURE 2.10: Flowchart of the EMD algorithm.

channels needed and, 4) Possibility for online application. A comparison of these properties for every technique is presented in Table 2.3.

TABLE 2.3: Comparison of properties of the different techniques.

Algorithm	Extra signal	Complexity	Channels needed	Online application
AF	Yes	Low	Low	Yes
ICA	No	High	High	Yes
EMD	No	High	Low	Yes
DWT	No	Low	Low	Yes

An extra reference signal from an EOG, ECG, or EMG may not be available in many applications (e.g. mobile EEG systems). From all the techniques, adaptive filtering is the only algorithm with such restriction, limiting its usage to applications where these reference signals are available.

The technique's complexity is often overlooked since signal processing is done in a computer with high processing power. Nevertheless, in mobile EEG systems with limited processing power, the whole process will take a longer time to finish. Furthermore, this computational burden will be reflected in the energy consumption of the device, reducing its battery life. Examples of these algorithms are ICA and EMD due to the complex calculations and iterative processing.

Moreover, another property considered was the number of channels required. From all the techniques discussed, only ICA is affected by the number of channels available. Since ICA uses the information available in all the channels to estimate the independent components of the signal, a lower number of channels will reduce its performance.

Finally, a crucial requirement for this work is that the technique can be performed in real-time. All the techniques are suitable for real-time implementation and can be used in this work. Although high complexity techniques could be considered unsuitable for real-time application, they could be appropriate for a real-time implementation with the proper optimisations as discussed in later Chapter 4.

2.2.4 Challenges of artifact separation in wearable EEG

One of the main challenges in wearable EEG systems is the contamination of the signal with artifacts. These systems are commonly used in a more naturalistic environment, making them particularly prone to the introduction of muscle artifacts due to this greater degree of freedom than traditional systems. Besides the ample number of algorithms in the literature, most are not designed to mitigate the impact of such artifacts ([Chen et al. \(2019b\)](#)), and therefore these algorithms are not as effective in a mobile environment as they could be in a more restricted environment ([Kline et al. \(2015\)](#)).

However, there are also algorithms specifically designed to remove artifacts in a mobile environment using techniques such as ICA, adaptive filtering, and combinations of wavelet and empirical mode decomposition. These algorithms are discussed in more detail in Chapter 4.

2.3 Brain Connectivity Networks

Information processing in the human brain is distributed across different regions of the brain. Every region is composed of a group of neurons and is in charge of a specific task. Brain connectivity is a tool used to illustrate how these brain regions interact to perform more complex cognitive tasks. It can be divided into two main categories: effective connectivity and functional connectivity ([Friston \(2011\)](#)).

2.3.1 Effective connectivity (EC)

Effective connectivity considers a dynamic causal interaction between the brain regions, and therefore the response obtained from a particular stimulus could be predicted with a model of the neuronal connections in the brain. This causal interaction shows the influence that a group of neurons exerts on each other. This correlation between neuron groups is represented using the effective connectivity network (Figure 2.11) obtained from EEG signals using different methods such as granger casualty or dynamic causal modelling (Niso et al. (2013)). It is important to mention that effective connectivity finds the causal effects while functional connectivity does not, being this the primary difference amongst them.

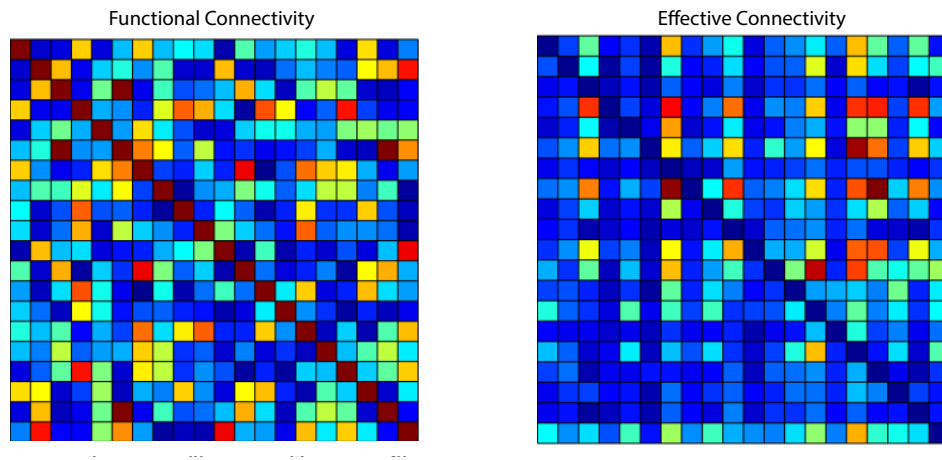


FIGURE 2.11: Functional and Effective connectivity matrix. Every row and column represents a region of the brain, and the intersection between them represents the strength of the connection among them.

2.3.2 Functional connectivity (FC)

Distinctively to effective connectivity, the functional connectivity considers a statistical dependency between the different brain regions even if there is no physical connection among them. By measuring the influences of all the regions in the brain (as opposed to effective connectivity), it is possible to obtain more information about the interactions across all regions. This dependency is represented using a functional connectivity network (Figure 2.11) obtained from EEG signals using methods such as General synchronisation index, phase-locking value and the phase lag index.

In this work, it was decided to use the functional connectivity to analyse EEG signals due to its statistical approach to describing the interactions among different brain regions, instead of the causal approach used in the effective connectivity.

2.3.2.1 General Synchronization Index (GSI)

A method used in the analysis of brain connectivity is the general synchronisation index (GSI). The concept of general synchronisation is an approach used to study the interactions between two dynamic and nonlinear systems in the time domain without further knowledge of the equations that govern it (Wendling et al. (2009)). If there is a synchronisation between these two systems, similar states should appear in both systems at similar times. The synchronisation between the two systems is calculated in the time domain, compared with other methods that are estimated in the frequency domain using the instantaneous phase, such as phase locking value and phase lag index. Different indices are used to measure the general synchronisation, such as the S index, H index, N index and synchronisation likelihood. This latter has become one of the most used due to its genuinely multivariate analysis, making it a more suitable method to analyse EEG signals than the other general synchronisation indices that are bivariate.

The synchronisation likelihood (SL) was proposed by Stam and Van Dijk (2002) as a new index to estimate the synchronisation between two or more time series, as is the case for EEG recordings. To estimate the SL, first, we need to consider the M simultaneous recordings $x_{k,i}$, where k denotes the number of channels for the case of EEG recordings ($k = 1 \dots M$) and i denotes the time instant ($i = 1 \dots N$). Then, an embedded vector $X_{k,i}$ is reconstructed for each channel, using a time delay as in (2.6). Here the τ corresponds to the embedded delay and m to the embedded dimension.

$$X_{k,i} = (x_{k,i}, x_{k,i+\tau}, x_{k,i+2\tau}, \dots, x_{k,i+(m-2)\tau}, x_{k,i+(m-1)\tau}) \quad (2.6)$$

Then, the calculation of the probability $P_{k,i}^\epsilon$ is made. This is the probability that the embedded vectors are closer than a minimum distance of ϵ from each other. This calculation is done for every channel k and instant i as in (2.7). In this equation, the $|\cdot|$ represents the Euclidean distance, and the θ represents the Heaviside step function, which is equal to one when the argument is negative and is equal to zero when the argument is either zero or a positive value. The w_1 is the Theiler correction for autocorrelation effects (Theiler (1986)), while w_2 sharpens the time resolution of the synchronization. These windows have to satisfy the condition $w_1 < w_2 < N$, where N is the number of samples. The ϵ value is found by determining the critical distance where the probability $P_{k,i}^\epsilon = p_{ref}$, where $p_{ref} \ll 1$.

$$P_{k,i}^\epsilon = \frac{1}{2(w_2 - w_1)} \sum_{j=1}^N \theta(\epsilon - |X_{k,i} - X_{k,j}|) \quad (2.7)$$

Next, we determine the discrete pair (i, j) where the embedded vectors are closer than the critical distance ϵ for every channel k and every distance pair that satisfy the condition $w1 < |i - j| < w2$, as in (2.8).

$$H_{i,j} = \sum_{k=1}^M \theta(\epsilon_{k,i} - |X_{k,i} - X_{k,j}|) \quad (2.8)$$

Then, the synchronisation likelihood is calculated for every discrete pair (i, j) and every channel as in (2.9).

$$S_{k,i,j} = \begin{cases} \frac{H_{i,j}-1}{M-1}, & \text{if } |X_{k,i} - X_{k,j}| < \epsilon_{i,k} \\ 0, & \text{if } |X_{k,i} - X_{k,j}| \geq \epsilon_{i,k} \end{cases} \quad (2.9)$$

Finally, the $S_{k,i,j}$ is averaged over all j values to obtain the synchronization likelihood ($S_{k,i}$) for every channel k at every time i as in (2.10).

$$S_{k,i} = \frac{1}{2(w2 - w1)} \sum_{j=1}^N S_{k,i,j} \quad (2.10)$$

Nonetheless, GSI may not be suitable for the analysis of neural signals since these methods are defined for theoretical models of coupled oscillators. These methods are also susceptible to changes in the amplitude of the signals, which is why often other methods based on phase synchronisation are preferred (Sun et al. (2012)).

2.3.2.2 Phase locking value (PLV)

Introduced by Lachaux et al. (1999), this measure calculates the synchronisation between two signals across a number of trials using the phase difference between these signals and their distribution on the unitary circle. The formula to calculate this measurement is defined in Equation 2.11. First, it is necessary to extract the instantaneous phase of the signals across the trial. Then, the difference between these phases is calculated to obtain a complex value for every data in the trial. These complex values are averaged across the trials, and the magnitude of the resulting average will be the PLV.

$$PLV_{i,j} = \left| \frac{1}{N} \sum_{n=1}^N e^{i(\phi_i(n) - \phi_j(n))} \right| \quad (2.11)$$

If the signals are synchronised, then the difference in the phases is low and will result in a PLV value that is close to one. On the other hand, when the signals are not synchronised, the phase difference will lead to a PLV value closer to zero.

2.3.2.3 Phase lag index (PLI)

This measurement was introduced by [Stam et al. \(2007\)](#) as a new way to estimate the phase synchronisation that would be robust against the presence of common sources compared with other measures such as PLV. The PLI calculation assumes that there should be an asymmetric distribution of phases when this distribution is centred in zero. When there is no synchronisation between the signals the distribution will remain in zero, and in the opposite case, when there is a synchronisation between the signals the distribution should be different than zero. Thus, the values for the PLI will be in the range of [0,1], where zero corresponds to the lack of synchronisation and one to a complete synchronisation across the signals.

The PLI is defined in Equation 2.12, where the phase synchronisation is calculated using the sign of the difference in the instantaneous phase (ϕ) between all the possible combinations among the channels. Here, i and j correspond to the channels, and N is the total number of samples in the time window. The sign of the difference in phases is accumulated and averaged across the whole trial. The absolute value of this average corresponds to the PLI.

$$PLI_{i,j} = \left| \frac{1}{N} \sum_{n=1}^N \text{sign}(\phi_i(n) - \phi_j(n)) \right| \quad (2.12)$$

By analysing these connectivity networks it is possible to identify certain features or patterns that could help us to detect an atypical behaviour on the brain. For instance, the study presented by [Douw et al. \(2010\)](#) using these connectivity networks for an early diagnosis of epilepsy. There are also similar studies for the diagnosis of other neurological diseases, like the one presented by [de Haan et al. \(2009\)](#) and [Jamal et al. \(2013\)](#) to diagnose Alzheimer's disease and Autism Spectrum Disorder respectively.

2.4 Characterisation of Brain connectivity networks using Graph-theoretic measures

Although the brain connectivity network provides useful information regarding the different interactions of the brain regions, further analysis is needed to understand the structural and temporal properties of such networks. Complex network analysis addresses this by introducing the concepts of graph theory into analysing these brain connectivity networks. The main goal of this analysis is to quantitatively characterise the functional brain connectivity networks using measures that are easier to compute and provide more insightful information about the underlying dynamic of the brain (Rubinov and Sporns (2010)).

These measures mainly fall into two categories: network segregation features, which identify the creation of groups across the entire network with a stronger interaction among them (i.e. clustering coefficient and transitivity); and network integration features, that capture the ability of the brain to integrate different brain regions (i.e. degree, characteristic path length) (Rubinov and Sporns (2010)). Table 2.4 shows a compilation of the complex network analysis features and their relationship with brain function.

TABLE 2.4: Summary of the graph-theoretic parameters and its relationship with brain functions (Rubinov and Sporns (2010)).

Graph-theoretic parameter	Relationship to brain function
Degree	<i>Interconnection between one region and other regions of the brain.</i>
Degree (weighted)	<i>Similar to the degree but also represent the strength between these regions.</i>
Density	<i>The ratio between the brain regions connected and all the possible connections among all regions.</i>
Clust. coeff.	<i>Measures how the different regions of the brain tend to cluster between them.</i>
Trans.	<i>Similar to the clustering coefficient but it measures the segregation in the whole brain instead of focusing on every region.</i>
Charact. path length	<i>Establish how efficient is the intercommunication among different brain regions</i>
Ecc., Rad. and Dia.	<i>Based on the characteristic path, these parameters establish the best/worst communication path form between the different brain regions.</i>

2.5 Applications of Brain connectivity networks in different disease phenotypes

The analysis of functional brain connectivity networks using complex network analysis has been proven useful in studying different neurological diseases. For example, the studies from [Das and Puthankattil \(2020\)](#) and [Duan et al. \(2020\)](#) analysed the functional brain connectivity networks to find possible patterns that could be used to aid the early detection of patients with Alzheimers disease. Similarly, the studies from [Chen et al. \(2019a\)](#) and [Ahmadlou et al. \(2012\)](#) investigated the overall structure of the brain connectivity network and used it to identify patients that could be diagnosed with attention-deficit hyperactivity disorder (ADHD). Furthermore, the brain connectivity network has also been used in the study of epilepsy. Examples of this are the study from [Staljanssens et al. \(2017\)](#) which focus on the location of the seizure onset zone used in epilepsy surgery, and the study from [Supriya et al. \(2016\)](#) related to the detection of seizures. Due to these excellent results in the study of different phenotypes, it was decided to incorporate the complex network analysis into the proposed connectivity-driven system.

To better illustrate how this analysis could be used in our system, let us consider an application example of a connectivity-driven neuromodulation system in treating epilepsy. Epilepsy is a neurological condition where the patient is susceptible to develop an epileptic seizure. The seizure is produced by the sudden synchronous activation of a large group of neurons ([Rangaprakash and Pradhan \(2014\)](#)). The causes of this condition are complex, but it is often attributed to a variation in particular genes of the patient ([Duncan et al. \(2006\)](#)). It has been established that there is an abnormal change in the functional brain connectivity during these seizures ([Stamoulis et al. \(2010\)](#); [Liao et al. \(2010\)](#); [Zhang et al. \(2009\)](#)) that could be used for its detection. It is possible to continuously estimate the brain connectivity networks and their graph-theoretic parameters to determine if the brain connectivity is unusual. If that is the case, then the electrical stimulus could be applied to bring the brain connectivity to a more natural state. In this way, the stimulus is applied just in time to avoid triggering an epileptic seizure, allowing the patient to live a more pleasant life without the constant fear of having a seizure.

Chapter 3

System specification and design approach

In this chapter, a discussion of the different timing and power specifications for the neuromodulation system is presented. Then, based on this discussion, a list of specifications for the whole system was created as a reference point for every module created in this thesis. Additionally, this chapter introduces the approach followed in the design of all the modules created in this thesis describing the necessary optimisations done to meet the system's specifications, followed by a description of the general design flow used for the design of every module.

3.1 System specification

Before starting with the design of the modules in the system, it is necessary first to define the system's specifications so the modules can be created accordingly. This thesis focused on two specifications crucial for the proposed system that we envision: the timing and power consumption specifications.

3.1.1 Timing specification

In order to provide the correct stimulus in the precise moment that is needed, the neuromodulation system needs to process the signal continuously and update the stimulus accordingly. This behaviour can only be achieved using real-time processing of the signal instead of the commonly used offline approach that requires the complete signal before starting the processing.

Furthermore, the neuromodulation system requires to complete the calculations of the graph-theoretic parameters in the order of milliseconds so the stimulus can be

updated accordingly and match the neural activation rate. The precise time needed for the brain to respond to the stimulus is unknown, and therefore it is difficult to establish a specific value. For instance, some EEG studies focus on the Event-Related Potential (ERP), which are small variations in the potential of the signal generated as a response to a specific event or stimuli such as sensory, cognitive or motor events. The delay in which these signals are generated varies and can go from around 40-70 ms (P50) to 300-600 ms (N400) (Sur and Sinha (2009)). The ranges from these ERPs provide a good starting point but not a definite answer for the processing time needed.

Nonetheless, the study from Baker et al. (2014) was able to identify short-lived states (between 100-200 ms) changes in the functional connectivity representing a unique pattern of brain activity. Since the neurofeedback system also uses functional connectivity, it is expected that any change in the functional connectivity network will be then reflected in a similar period between 100-200 ms. Therefore, it would be necessary to perform the whole process described in Figure 1.1 in less than 200 ms to capture the changes in the functional connectivity and adjust the stimulus accordingly. In addition to the processing time, it is important to consider the acquisition time. The review from Dias et al. (2012) shows that the current technology in EEG acquisition systems can provide fast acquisition of the EEG signal to meet the timing requirements mentioned. For instance, the work presented by Obeid et al. (2004) offers a low-power analogue front end capable of reaching a sampling frequency up to 62.5 KHz per channel. Based on this, we decided to distribute only 10% (20 ms) of the total time budget to the acquisition stage and leave the rest for the processing of the EEG (180 ms), which will require a longer time.

This processing time can only be achieved by processing the signal on the device instead of sending it to an external device like a computer or a mobile phone for its processing. By processing the signal on the device, it is possible to save time (and energy) that otherwise would be wasted transmitting the data to an external device, helping to achieve the timing specification. To put this in perspective, Bluetooth Low Energy (BLE) transceiver chips such as the NRF8001 and the NRF5281 used in commercial wearable devices, have a maximum transmission rate of 1 Mbps and 2 Mbps respectively. Using the device presented by Obeid et al. (2004) as an example, the transmission time of 100 samples will be equal to 19.2 ms for the NRF8001 and 9.6 ms for the NRF5281. This time is about 10% and 5% of the total time allowed for the processing that could be better used in further stages of the EEG signal processing.

3.1.2 Power specification

One of the main advantages of the neurofeedback system is its portability, allowing it to constantly monitor the patient in a nomadic environment while performing any other task.

To guarantee continuous monitoring of the patient, the device needs to operate continuously for at least 18 hours. In this way, the system can be used during the day and charged at night while the patient sleeps. Modern batteries used in portable devices (i.e. mobile phones) can achieve a capacity of 4200 mAh at 3.82V with a small form factor (83.68 x 63.15 x 4.50 mm, and 69 g) (Liang et al. (2019)). This small factor easily integrates it into a portable system without compromising design size or weight, making it ideal for this application. With this battery capacity, the maximum current consumption should be lower than 233 mA/hour to achieve this 18 hours of continuous operation. This represents a power consumption of about 712 mW for the whole system, which involves an acquisition system and signal processing. The review from Xu et al. (2017) showed the different acquisition systems in literature with power consumption between 20 μ W and 7.5 mW per channel. Since the proposed system uses 19 channels, the total power consumption of the acquisition system will be between 380 μ W and 142.5 mW. Consequently, it was decided to allocate only 20% (178 mW) of the total power available to the acquisition system and the remaining (712 mW) to the systems involved in processing the signal.

The total power consumption in an integrated circuit is determined by the static and dynamic power. The static power consumption is caused by the leakage current in the transistors when they are not being switched. This parameter is dependent on the transistor technology used, and therefore no much improvement can be made. On the other hand, dynamic power consumption is caused by the switching activity of the transistors while the circuit is operating. Therefore, it is possible to optimise the design and reduce the total power consumption of the system. The dynamic power consumption can be defined as in Equation 3.1 (Amara et al. (2006)), determined by the supply voltage of the circuit (V_{DD}), the parasitic capacitance of the circuit (C_p), the switching activity (α) and the operation frequency (f).

$$P_d = \alpha C_p V_{DD}^2 f \quad (3.1)$$

Besides the supply voltage, the rest of the parameters can be modified to reduce power consumption. For instance, the parasitic capacitance is directly proportional to the number of transistors, so its reduction will reduce the power. The implementation of computationally intensive arithmetic operations in hardware contributes considerably to the number of transistors of the design. Therefore, it is necessary to reduce or avoid operations such as adders, multipliers, dividers, and square roots to reduce the power consumption on the system.

Furthermore, reducing the complexity of the algorithms implemented on hardware can reduce the system's power consumption. Reducing the algorithms' complexity reduces the number of computations done, which reduces the switching activity of the transistors in the design. Also, reducing the algorithms' complexity helps to reduce

the time required for the processing, making it possible to reduce the operating frequency without affecting the time performance.

Finally, as discussed in the previous section, it is possible to reduce the power consumption by performing the complete processing in the same device instead of transmitting the data to an external device. The study presented by [Maharatna et al. \(2012\)](#) found that better power efficiency can be achieved using processing on edge due to the significant power required to transmit the data. For instance, the energy consumption of the previously mentioned NRF8001 and nRF5281 transceivers is 38.1 mW and 13.8 mW during transmission. Even though this is only for short periods and is relatively lower than the total power budget (less than 5%), it is essential to reduce energy consumption as much as possible to guarantee a prolonged battery life. Therefore, performing the whole processing in the device is preferred to reduce the power consumption.

3.1.3 Summary of the system specifications

Based on the previous discussion about the time and power specifications, it is possible to summarise the specifications in the following list:

- The system should be able to perform real-time processing of the signal.
- The system should perform the whole processing without transmitting the data for external processing.
- The system should perform the complete process in a time no greater than 180 ms.
- The power consumption of the system should be lower than 712 mW. This estimation was done considering a 4200 mAh portable battery and continuous operation for 18 hours.
- The system will reduce to a minimum the number of computationally intensive arithmetic operations (e.g. adders, multipliers, dividers and square roots) that require extensive resources for its implementation in hardware.
- The system should implement an optimised version of the algorithm with lower complexity, reducing the number of computations needed and improving the power consumption.

3.2 Our design approach

The system that we proposed is presented in Figure 3.1. A modular approach was made in this project, creating independent modules in charge of different signal

processing stages. This modularity gives flexibility to the system where the modules can be interchanged with other modules in case that a different functionality is required. For instance, if a different functional brain connectivity algorithm is required, it is possible to change the PLI module for another module that performs the PLV instead. In this way, the whole design is not limited to a particular application and can be modified as needed. Also, in Figure 3.1 it is presented a timeline of the different processes. Except for the artifact removal module that starts as soon as new data is acquired, the other modules perform the calculations sequentially after the previous process finishes. It could be possible to instantiate more modules to speed up the processing of a particular module at the expense of increased energy consumption. In this work, we avoid this approach to save as much energy as possible while still meeting the systems requirements.

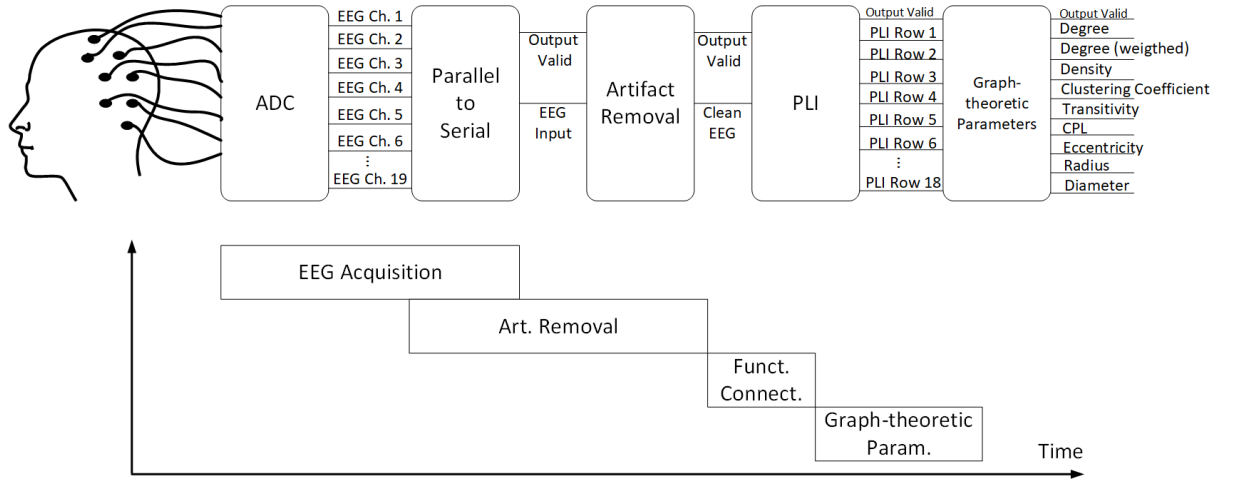


FIGURE 3.1: Block diagram of the complete neurofeedback system.

The methodology followed during the design of all the modules presented in this work focused mainly on reducing the algorithms' complexity and processing time for its hardware implementation. In general, the algorithms found in the literature for processing the EEG signals are not designed considering the computational complexity that such algorithms could have. Since EEG signals are traditionally acquired and later processed offline in a computer, the algorithm's complexity does not represent a problem, nor does the calculation time. However, this is not the case for the system we envision, and therefore, a different approach has to be taken, emphasising the complexity of the algorithms and the processing time so they can be implemented on hardware that can be used in a wearable EEG system.

3.2.1 Algorithmic reformulation of computational intensive modules

Traditional algorithms used in processing EEG signals are not designed considering the complexity that the whole processing may have. In a hardware implementation, this complexity represents a significant problem for the following factors:

- **Resource limitations:** In a hardware implementation, the number of resources is limited and therefore, complex architectures are not implementable. In an FPGA design, the number of arithmetic blocks (adders, multipliers, dividers, etc.) is limited, so that they have to be used more carefully. In the case of an ASIC, the area limitation inherently creates a restriction on the number of resources used. Therefore in both cases, careful usage of the resources need to be done.
- **Processing time:** Traditionally, the EEG signals are acquired and later processed offline. However, the processing needs to be completed as fast as possible in the system that we envision. If the processing time is too long, the stimulus application will be delayed and probably done in a time that was not appropriate. Thus, the complexity needs to be reduced to ensure short processing times.
- **Energy consumption:** Since the device needs to be portable, the energy consumption needs to be low to guarantee a long battery life. Energy consumption is highly related to computational complexity. A complex algorithm will require a more significant number of operations, which will lead to extensive usage of resources that will impact energy consumption. Therefore, a lower complexity will be better to reduce the energy consumption and prolong the battery life in the portable device.

Due to these factors, a reformulation of the algorithms is necessary. This without any compromise in the effectiveness and accuracy of the algorithms.

3.2.2 Design flow

The design flow for all the modules presented in this work is shown in Figure 3.2, and it was based on the traditional V-model often used in software development (Mc Hugh et al. (2013)).

The stages presented here are the following:

1. **Concept and Requirements:** The first step in the process is to create the module concept. The module's main functionality is described in broad terms, providing a better understanding of the module's goal. Then, based on this concept, a series of requirements are proposed. For instance, in this project, the main requirements were low resource usage, low computation time and low energy consumption. These requirements are crucial for the system that we envisioned and were considered throughout the whole project.
2. **Algorithm selection:** After setting the system's main requirements, the next step was to select the algorithm that met the requirements. As discussed in further

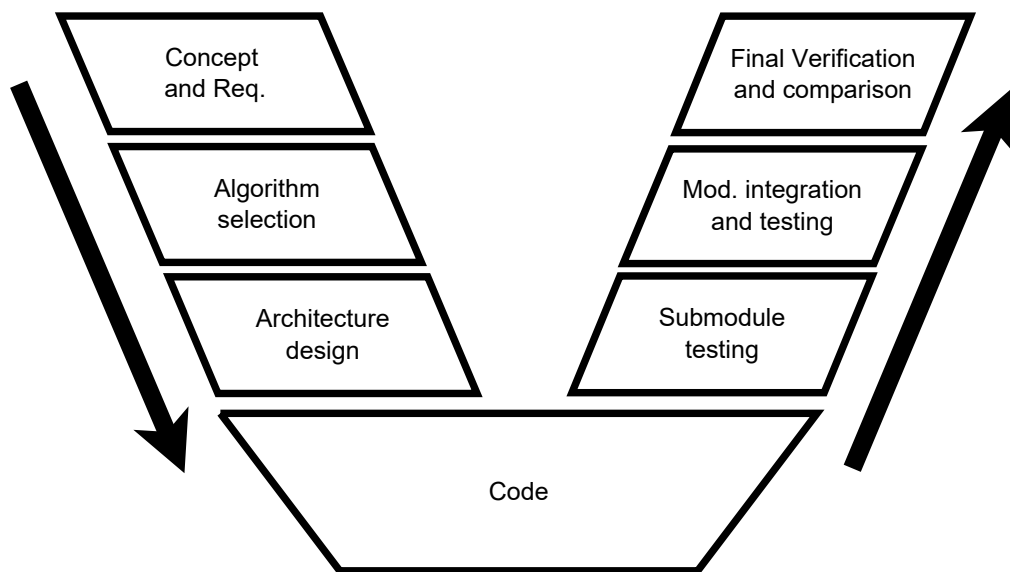


FIGURE 3.2: Design flow followed in this work, based on the V-model used in agile software development. Based on [Mc Hugh et al. \(2013\)](#).

chapters, in some cases, the algorithms did not satisfy the requirements previously mentioned and some modifications were done to fulfil the requirements.

3. **Architecture design:** Once the final algorithm is selected, the architecture of the modules is designed. It is possible to make some more optimisations at this stage to reduce the number of resources used.
4. **Code:** The codification of the architecture is done at this stage. All the modules were coded using System Verilog, a hardware description language (HDL).
5. **Submodule testing:** Every submodule is tested individually to guarantee the correct individual functionality before its integration. This modular approach facilitates debugging since it is easier to isolate any error found.
6. **Module integration and testing:** Once every submodule is tested individually, they are integrated to create the final module. A final test is then done to verify that the integration did not cause any problem.
7. **Final Verification and comparison:** The final module is then functionally validated to verify that it met the requirements and functionality detailed during the first stage. Further analysis is also done to assess the performance impact of this hardware implementation compared to the equivalent software. This analysis guarantees that the modifications done to the algorithm during the process did not significantly impact the algorithm's accuracy. Additionally, the whole design is then synthesised to determine the number of resources used (i.e.

logic elements and memory) and estimate its power consumption. This will allow us to verify if the design can be implemented and meet the specifications in terms of energy consumption. The FPGA used for the synthesis is a development Board DE4 with a Stratix IV GX EP4SGX230K FPGA. This development board is ideal for signal processing applications due to its vast number of logic elements available, a high number of Digital Signal Processing (DSP), and low power consumption. Implementing the system in an FPGA is necessary since it provides a way to perform functional validation of the system and perform any necessary change to meet the requirements established before starting the design of an ASIC, where it is harder to make any modification.

Chapter 4

Real-time artifact removal hardware based on hybrid WPT-EMD algorithm for multichannel wearable wireless EEG system

4.1 Introduction

Electroencephalography (EEG) is a non-invasive clinical tool for monitoring brain activity by measuring the electrical potential at the scalp generated by groups of neurons in the brain. It has been widely applied to study different cognitive processes [Subha et al. \(2010\)](#) and considered a viable tool to aid in diagnosing different neurological disorders such as Epilepsy, Alzheimer's disease, and Autism Spectrum Disorder [Adeli et al. \(2003\)](#); [Stam et al. \(2006\)](#); [Jamal et al. \(2013\)](#) for instance. Furthermore, the recent advent of wireless wearable EEG systems makes it possible to study the brain functionalities during a cognitive task in a mobile environment, obtaining a more natural response than traditional studies in controlled environments. This opens up the possibility to develop new systems that could provide a stimulus to the patient in real-time based on the instantaneous response of the brain in a nomadic environment. An example of such systems is the neurofeedback system described in Chapter 1. This is a wearable device with continuous capture of EEG signals used to formulate functional brain connectivity on-the-fly and compare with templates of the desired "correct" functional brain connectivity. According to the outcomes of this comparison, a control module will regulate the appropriate multi-site Transcranial Current Stimulation (TCS) distribution to match the desired functional brain connectivity.

One of the biggest challenges of this system lies in the effective separation of artifacts while capturing the EEG data, as this is the fundamental pre-processing step before the other computations processes described above can be initiated. EEG is highly susceptible to artifacts from various non-neural sources such as eye blinking, cardiac artifacts, and muscle movements. In the case of wearable EEG in a nomadic environment, the causes for artifacts are heightened due to the patient's greater degree of freedom (Thompson et al. (2008)). Typically, EEG artifact separation is carried out as an offline process where the data is transmitted to either a mobile device or a central computing facility. This scenario is not applicable in our envisaged system owing to the real-time constraint. The system requires to complete all the processing in less than 180 ms, utilising a low amount of energy to enhance battery life and make it sustainable, as described in Chapter 3. These requirements can only be achieved with processing on edge (Maharatna et al. (2012)) instead of external processing, which will increase the processing time and the power consumption due to the transmission of the data.

Therefore, the work presented here focuses on the aspects of artifact separation from the perspective of a wearable system. More specifically, the two principle attributes needed for an artifact separation module in this scenario are: 1) algorithmic accuracy of artifact separation to take care of the different types of artifacts in wearable systems and; 2) compliance of the selected algorithm for real-time implementation satisfying the timing and energy constraints required in our application. This latter attribute calls for an algorithm to architecture mapping optimisation for minimising the number and complexity of arithmetic operations in the algorithm - creating a "lightweight" artifact separation algorithm - without affecting the desired performance. In this work, we propose such a lightweight real-time artifact separation algorithm and show its proof-of-concept implementation in Stratix IV GX Field Programmable Gate Array (FPGA).

4.2 Background

4.2.1 State of the art artifact removal algorithms

Due to the advances in EEG technology, it is now possible to find multiple commercial solutions for wearable EEG systems (Valentin et al. (2018)), offering a new inexpensive and portable solution to monitor brain activity. This opens the possibility to study the brain in a more nomadic environment. However, this new environment also exposes the EEG signal to new and stronger artifacts that can corrupt it, such as muscle artifacts. Therefore, it is necessary to implement algorithms that are designed to mitigate the effects of the artifacts in this particular environment.

There are implementations such as the ones presented by [Gwin et al. \(2010\)](#); [Onikura and Iramina \(2015\)](#); [Daly et al. \(2013\)](#); [Kilicarslan et al. \(2016\)](#); [Lee et al. \(2020\)](#) that were created to mitigate the artifacts in wearable systems using techniques such as ICA and Adaptive filtering. Although the successful results, these approaches have certain limitations that could contrast with some implementations of wearable systems. For instance, the number of electrodes in a wearable system is low, which could reduce the effectiveness of algorithms like ICA that is highly dependent on the number of electrodes in the system ([Troy M et al. \(2012\)](#); [Klug and Gramann \(2020\)](#)). Similarly, algorithms using adaptive filtering methods require an extra signal as a reference to update the coefficients of the filters. This extra signal (i.e. EMG, EOG, accelerometer) may not be available for some systems, decrementing the performance of these algorithms. Therefore, it would be better to opt for other methods not affected by such limitations in wearable systems.

[Bono et al. \(2016\)](#) presented a new algorithm to remove artifacts in wearable EEG systems based on a hybrid approach of Wavelet Packet Transform (WPT) and Empirical Mode Decomposition (EMD). Both methods do not require an extra reference signal and are not dependent on the number of electrodes in the system, making it ideal for wearable EEG systems. Furthermore, this algorithm was tested with real-life artifacts such as eye blinking, head movement, speaking and movement of the hands. The algorithm performed better than using a combination of the WPT and the commonly used ICA. For these reasons, it was decided to implement this algorithm in our system to remove the artifacts in the signal.

However, the selected algorithm was designed for offline cleaning of the signal, which contrasts with the online processing required in the system proposed in Chapter 3. Therefore, it is crucial to modify this algorithm to allow an online cleaning of the EEG signal. To achieve this, it is necessary to reduce the computational complexity of the algorithm so the calculation can be completed as fast as possible ([Cao et al. \(2015\)](#)). This is a significant challenge since it involves a redesign of the algorithm to reduce the complexity without compromising the effectiveness of the algorithm. Additionally, the new design needs to be energy efficient for its implementation in wearable systems. Implementations of artifact removal have been previously done using a microcontroller such as in [Jafari et al. \(2017\)](#) and [Majmudar and Morshed \(2016\)](#). However, these implementations need long processing times and high energy consumption, which is not ideal for this work. These timing and energy constraints involved in wearable systems can only be satisfied with an on-chip implementation, as previously suggested by [Islam et al. \(2016\)](#). Therefore, to achieve the system's timing and energy consumption constraints, it is necessary to modify the algorithm from [Bono et al. \(2016\)](#) to reduce its complexity and create a new hardware architecture for its implementation.

4.2.2 Hybrid WPT-EMD Artifact Removal Algorithm

The algorithm selected for this work is the one proposed by [Bono et al. \(2016\)](#). This algorithm removed the artifacts found in EEG signals from a wearable device with better performance than other techniques such as ICA. Furthermore, this algorithm does not require a high number of electrodes or an extra signal compared to ICA or Adaptive filtering, as previously discussed. These advantages make this algorithm an excellent fit for this work.

The algorithm consists of two main stages; the first stage performs an initial artifact separation using the WPT technique, followed by a further artifact separation at the second stage on the processed signal from the WPT stage using the EMD technique.

4.2.2.1 Wavelet Packet Decomposition (WPT)

The WPT is a variation of the Discrete Wavelet Transformation (DWT) where the signal is recursively decomposed both into low and high (approximation and detail) frequency components using low- and high-pass filter banks respectively, followed by downsampling of the signal. These recursive decompositions are repeated until the desired decomposition level is reached. The coefficients of these filters are generated based on the wavelet used for the decomposition/reconstruction. In [Bono et al. \(2016\)](#), the authors decided to use a Mayer wavelet for the whole WPT.

In this first stage of the hybrid algorithm, the WPT is applied to every EEG channel up to seven-level decomposition generating 128 components (nodes) for each channel. To identify the node related to the artifacts the standard deviation is calculated for every node across all the channels using (4.1), where EGY corresponds to the energy of every node i , C corresponds to the number of channels, and N is the number of samples at the node. The node with the highest standard deviation is considered as the node containing artifacts and is removed from the decomposed set of nodes. The remaining nodes are used for reconstructing the signal using a set of synthesis filters and up samplers in the opposite direction of decomposition. The signal thus reconstructed is used as the input for the EMD processing.

$$\sigma_{EGY} = \sqrt{\frac{1}{C-1} \sum_{c=1}^C (EGY_{c,node_i} - \overline{EGY}_{across\ C,node_i})^2}, \quad (4.1)$$

$$EGY_{c,node_i} = \sum_{n=1}^N node_i^2[n],$$

4.2.2.2 Empirical Mode Decomposition (EMD)

The EMD is a process where the signal is decomposed in a series of different basic oscillating functions known as the Intrinsic Mode Function (IMF) (Huang et al. (1998)). The EMD process is presented in Algorithm 1, and it consists mainly of an iterative process known as Sifting (while loop), which is repeated until a certain condition is met and the resulting signal is now considered as an IMF of the original signal.

Result: IMF, Residue

Residue = Input Signal;

for $n = 1$ to N number of IMFs **do**

 Sift = Residue;

while *Until Sifting is an IMF* **do**

 Calculate the extrema from Sift;

 Calculate the Upper and Lower envelope from extrema points;

 Mean = (Upper envelope + Lower envelope)/2;

 Sift = Sift - Mean;

end

 IMF[n] = Sift;

 Residue = Residue - Sift;

end

Algorithm 1: EMD algorithm.

A commonly used condition to stop is when the standard deviation of the signal is < 0.3 between two consecutive siftings (Huang et al. (2003)). However, in Bono et al. (2016) the condition used is based on two thresholds (θ_1 and θ_2) and the evaluation function ϵ defined in (4.2).

$$\epsilon(t) = \left| \frac{m(t)}{a(t)} \right|; m(t) = \frac{u(t) + l(t)}{2}; a(t) = \frac{u(t) - l(t)}{2} \quad (4.2)$$

The evaluation function is obtained using the mean (**m**) and the mode amplitude (**a**), which are calculated with the upper (**u**) and lower (**l**) envelopes of the sift signal. To consider the sift signal as an IMF the ϵ signal has to meet two conditions:

1. $\epsilon < \theta_1$ for a fraction of $(1-\alpha)$ of the total duration.
2. $\epsilon < \theta_2$ for the rest of the fraction.

The typical values used for these constants are: $\theta_1 \approx 0.05$, $\theta_2 \approx 10\theta_1$ and $\alpha \approx 0.05$.

When the IMF is found, it is subtracted from the input signal, and the process is repeated until the signal cannot be decomposed anymore. This will produce a set of IMFs and a residual signal.

To identify the IMF related to the artifacts, Bono et al. (2016) used the J criterion (J) as defined in (4.3), calculated with the weighted addition ($w = 0.5$) of the standard deviation (σ) and the entropy (H) of every IMF and a clean reference signal (EEG signal acquired in a subject during resting state). The IMF with the highest J is considered to be related to the artifacts.

$$J = w \left(\frac{H_{IMF}}{H_{Resting}} \right) + (1 - w) \left(\frac{\sigma_{IMF}}{\sigma_{Resting}} \right) \quad (4.3)$$

Subsequently, the signal is reconstructed by adding the residual signal and all the IMFs but the artifact IMF.

4.2.3 Performance parameters

To evaluate the performance of the algorithm Bono et al. (2016) used a criterion known as Artifact to Signal Ratio (ASR) (Bono et al. (2014)) as in (4.4).

$$ASR = \frac{P_{Artifact}}{P_{Resting}} = \frac{(P_{Corrupted} - P_{Clean})}{P_{Resting}} \quad (4.4)$$

The ASR compares the power of the signal before ($P_{corrupted}$) and after the artifact removal algorithm (P_{clean}) to evaluate its effectiveness. The $P_{resting}$ value corresponds to the power of the signal in a resting state and it is considered as the closest reference of a clean signal available. Since the $P_{resting}$ signal is constant, the ASR value will be directly proportional to the difference between $P_{corrupted}$ and P_{clean} . If the artifact removal was effective, the power in the clean signal would be lower than the corrupted signal, and therefore the ASR value will be greater than zero. In the opposite case, if the power in the clean signal is higher than the power in the corrupted signal, the difference will be negative, and thus the ASR will be less than zero. Finally, if the power between the corrupted and the clean signal is almost the same, the ASR value will be close to zero, implying that the signal remains almost unchanged. Therefore, the magnitude and the sign of the ASR effectively show how well an algorithm can remove artifacts from a corrupted EEG signal.

4.2.4 Algorithm shortcomings

The algorithm proposed in Bono et al. (2016) was able to achieve a performance improvement of up to 70.4% compared with other techniques used during the study. Despite its good performance, this algorithm is not implementable in a real-time hardware system - the main focus of this work - because of the following reasons:

1. The algorithm can only be applied when the entire EEG signal has been acquired, making it essentially an offline process. In real-life, where EEG signal is acquired continuously, one needs to apply an artifact separation algorithm on a "suitable time window" of EEG signal. Therefore the direct application of Bono et al. (2016) is not possible here.
2. The algorithm in Bono et al. (2016) used the Meyer wavelet for the WPT, which is not appropriate for implementing in a real-time hardware system due to its complexity. Implementing the DWT using this wavelet will require filters with a large number of coefficients (discussed in later sections) which has important implications on the overall hardware resources, complexity, and the subsequent energy consumption. Thus, a more suitable wavelet for the transformation needs to be selected with caution on the impact in terms of accuracy that this could generate.
3. The algorithm Bono et al. (2016) used inherently assumed the ideal scenario of a large number of IMFs during EMD processing. Moreover, it considered a large number of sifting iterations are allowed to reach the desired accuracy set by the user. These conditions are not possible to implement in real-life hardware where the resources and time are restricted, and therefore one needs to impose some upper limits on the number of IMFs and the sifting iterations. It is expected that such a limiting condition will affect the final accuracy of the result, and therefore a trade-off criterion needs to be formulated for the system implementation.
4. The algorithm in Bono et al. (2016) assumed that the EEG signal is "always" corrupted and applied its processing steps accordingly. Although this is a very practical consideration, particularly in a mobile environment, it restricts the generalisability of the algorithm as it will tend to modify the signal even if the EEG is completely clean to start with.

Thus, it is necessary to introduce certain modifications to the algorithm presented in Bono et al. (2016) to make it adoptable in our system - essentially making a new version of the algorithm itself for our specific purpose.

4.3 Algorithmic Formulation and parameter determination for real-time hardware implementation

In this section, we present a thorough exploration of the different governing parameters of the algorithm that could be modified to adapt the algorithm in our system. We analysed the impact of such modifications of these governing parameters in terms of accuracy and hardware complexity and select the optimal configurations of them for our purpose.

4.3.1 Experimental data

For the analysis, we used data generated through the experiment described in [Bono et al. \(2016\)](#). This was obtained using a commercial ENOBIO wireless EEG acquisition cap with a 24-bit Analog-to-digital converter (ADC) with a 19 channel configuration. The signal acquired was sampled at 500 Hz for five healthy subjects (three male and two female, with an average age of 28) and seven different artifacts: eyes blinking, head movement (up and down), chewing, speaking, hand movement, head movement (left and right) and shaking head.

4.3.2 WPT

4.3.2.1 Window size and Mother Wavelet

Two main parameters governing the hardware implementation of the WPT module are the choices of the window size and the mother wavelet. Typically in EEG processing, the entire signal is captured before starting its processing. Such a strategy could not be applied for online real-time processing. Moreover, the accuracy of an algorithm depends upon the choice of mother wavelets. However, a computationally intensive mother wavelet, such as the Mayer wavelet, is detrimental for hardware implementation. Since the WPT is performed using a series of quadrature filters (discussed later in this chapter), the number of resources required will be directly proportional to the number of coefficients of the mother wavelet selected. As shown in Table 4.1, the number of coefficients of the Dmey function can be up to 24-fold higher than other wavelet functions. The higher the number of coefficients, the larger the number of adders and multipliers used in the quadrature filter, which is reflected in the system's computational complexity, area, and energy consumption. Thus, here we explore the algorithm's performance under different window sizes and mother wavelet conditions to select these two parameters for hardware friendly mapping.

To select the best window size, the algorithm was tested with 4 different sizes of non-overlapping windows, starting with the minimum window size required for a seven-level decomposition which is 128 (2^7) samples, followed by 256 (2^8), 512 (2^9) and 1024 (2^{10}) samples. No windowing technique was used in order to reduce the complexity of the algorithm ([Farooq and Datta \(2004\)](#)). These four window sizes have been applied in conjunction to 15 different wavelet functions (presented in Table 4.1 along with the number of required coefficients for each of those functions) to explore optimal window size and mother wavelet for hardware implementation without sacrificing the accuracy of the original algorithm. Here, the ASR defined in (4.4) is used for benchmarking the performance of the algorithm in terms of accuracy. The

ASR calculation was done for the δ (0.5-4 Hz), θ (4-8 Hz), α (8-13 Hz), β (13-30 Hz) and γ (30-42 Hz) frequency bands.

TABLE 4.1: Wavelet family functions used for performance comparison.

Wav. Fam.	Coefs.	Wav. Fam.	Coefs.	Wav. Fam.	Coefs.
Dmey	102	Sym2	4	Bior3.3	8
DB2	4	Sym3	6	Coif1	6
DB3	6	Sym4	8	Coif2	12
DB4	8	Sym5	10	Coif3	18
DB5	10	Bior2.2	6	Coif4	24

Due to a large number of results to compare, instead of comparing ASR for every channel in every frequency band, artifact, patient, window and wavelet, we followed a different approach to make the comparison. First, we find the mean values of the ASR in every channel for Bono et al. (2016) and the modified algorithm. Secondly, we create a box plot with the mean value of the ASR for all the subjects and artifacts (a total of 35 tests); for every window size and wavelet function. This is done for every frequency band as shown in Figure 4.1(A).

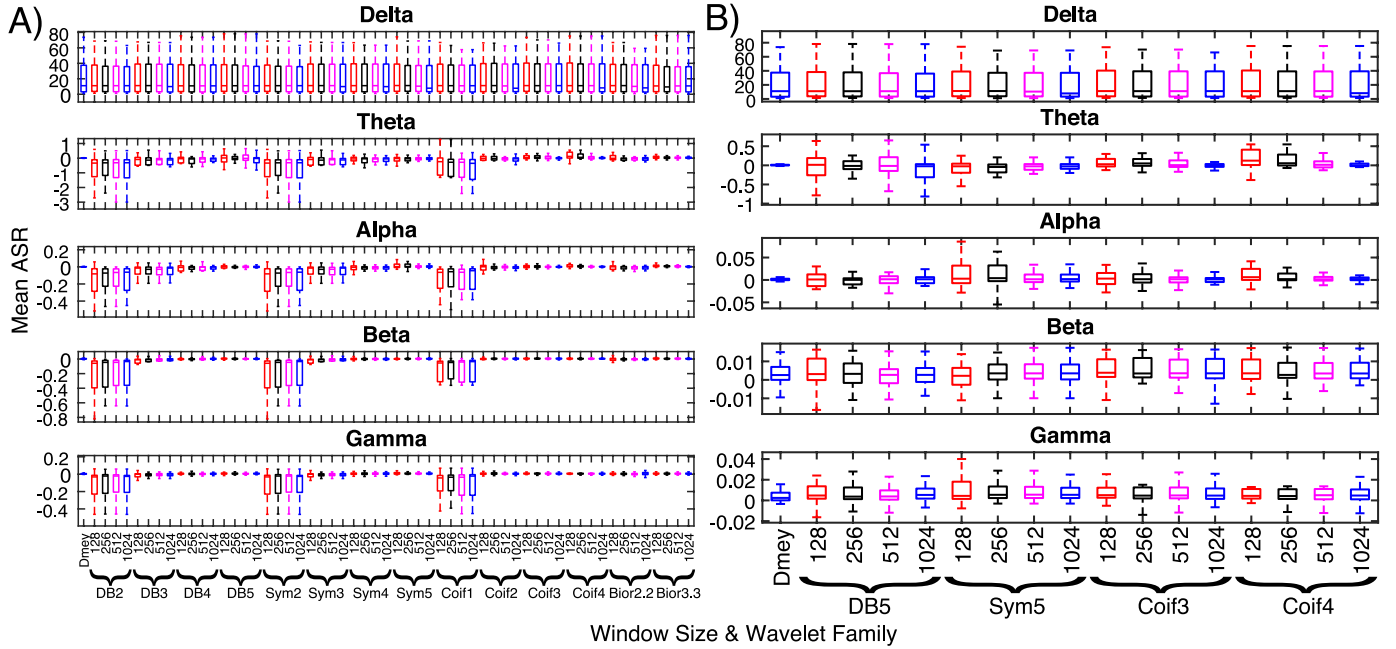


FIGURE 4.1: A) Mean ASR comparison using the Dmey wavelet and the whole signal (as in Bono et al. (2016)) against using different wavelet functions (from Table 4.1) and window sizes. B) Mean ASR Comparison between the Dmey wavelet and the whole signal (as in Bono et al. (2016)) against the wavelet functions with the best performance, using different window sizes.

It is apparent that out of all the wavelet functions, “DB5”, “Sym5”, “Coif3” and “Coif4” show consistent performance similar to the “Dmey” function with a considerably lower number of coefficients. A further comparison between them is shown in Figure 4.1(B). These findings match with the analysis of Wang et al. (2011), where the wavelet function “Coif4” was considered the second-best candidate after

discrete Meyer in the extraction of features from EEG signals using WPT. Between these four functions, the ones requiring the least number of coefficients are “DB5” and “Sym5”. Also, in terms of performance, both functions have a similar performance to the “Dmey” function, which make them suitable for this work. To aid in the selection between these two functions, we consider the work of [Lema-Condo et al. \(2017\)](#) that suggest “DB5” is a suitable function in EEG analysis. Therefore, for this implementation, the “DB5” was selected as the mother wavelet. This function presented a similar performance to the “Dmey” function but required about 10-fold fewer coefficients. This reduction in the number of coefficients will be reflected in the complexity and resource utilisation of the system, reducing the power consumption.

In terms of the window sizes, Figure 4.1(B) shows that increasing the size of the windows help to achieve a more consistent performance as the results diverge less from the median values (smaller box and whiskers) as the window size increases. Moreover, the maximum difference in the median values between a window size of 128 and 1024 was 0.3 for the DB5 wavelet. These differences in the performance between different windows are minimal, and therefore it was decided to use the smallest possible window, namely 128, allowing faster processing of the signal at the expense of a slight decrease in the performance.

4.3.2.2 Node removal selection

To identify the node related to the artifact it is necessary to calculate the standard deviation σ_{EGY} as shown in (4.1) and find the node with the highest value. The main problem of using σ_{EGY} is the square root operation involved in the calculation. Although optimised square root circuits in terms of speed and area have been proposed in [Rahman et al. \(2014\)](#); [Vijeyakumar et al. \(2012\)](#), its usage still represents a cost in terms of resources. The principle behind the node selection is the comparison of σ_{EGY} amongst different nodes to identify and remove the node with the highest value. Therefore, the same comparison could be done using variance instead of standard deviation, which does not need square root computation. Thus, the selection of using the variance for the node removal selection.

4.3.3 EMD

4.3.3.1 Window size

Similar to the WPT, the EMD is usually performed after the whole signal has been acquired, which is again not suitable for real-time processing of the signal and a different approach needs to be taken.

Due to the nature of the algorithm, one of the most important requirements to perform an EMD is that the signal needs to have a certain number of extrema points necessary to perform the interpolations used during the sifting process (Rilling et al. (2003); Huang et al. (1998)). In this case, we need at least two maxima and two minima to perform a linear interpolation (discussed in the next subsection) between the extrema points. Nevertheless, the EEG signals are non-linear and non-stationary, and their characteristics are hard to model. Therefore, it is not possible to predict if the condition previously mentioned will be satisfied within the selected window or not, and also, it is not possible to know how long one needs to wait until this condition is satisfied. Hence, it was decided that the best option would be to implement a fixed window instead of waiting for the minimum number of extrema points. In this way, if the number of points does not suffice, the system will not perform the EMD for that window which will avoid waiting indefinitely and delay the following process in the chain.

Following the same approach used for selecting of the WPT window size in 4.3.2, a comparison of the mean ASR performance with different window sizes is presented in Figure 4.2. In this figure, it can be seen that the performance between the different windows sizes is close to the one achieved by using the whole signal as in Bono et al. (2016), except for the gamma band. This difference in the gamma band is not as significant, and therefore it is considered a small trade-off. In terms of the different window sizes, the performance among them remains similar. Thus it was decided to select the smallest window size of 128 as in the WPT. By having an equal window size as the WPT, it is possible to perform the EMD immediately after the WPT is finished instead of waiting for more samples.

4.3.3.2 Interpolation method

A step in the EMD process is to generate the upper and lower envelopes using the extrema points found in the signal. In Bono et al. (2016), cubic spline interpolation was used to create a continuous and smooth function compared to other interpolation methods. However, cubic spline requires a high computational cost since it is necessary to solve an $[n-1]$ equation system where n is the number of points for every envelope calculated on every sifting iteration. To simplify this, we used linear interpolation in this work as in (4.5), which will reduce the number of arithmetic operations needed in the calculation of the envelopes. Although this method does not generate a function as smooth as the cubic spline interpolation, it still has shown promising results reported in Lu (2007) and later verified in our results, achieving a maximum difference of 0.2 in the mean ASR of the median value from all the artifacts, patients and frequency bands.

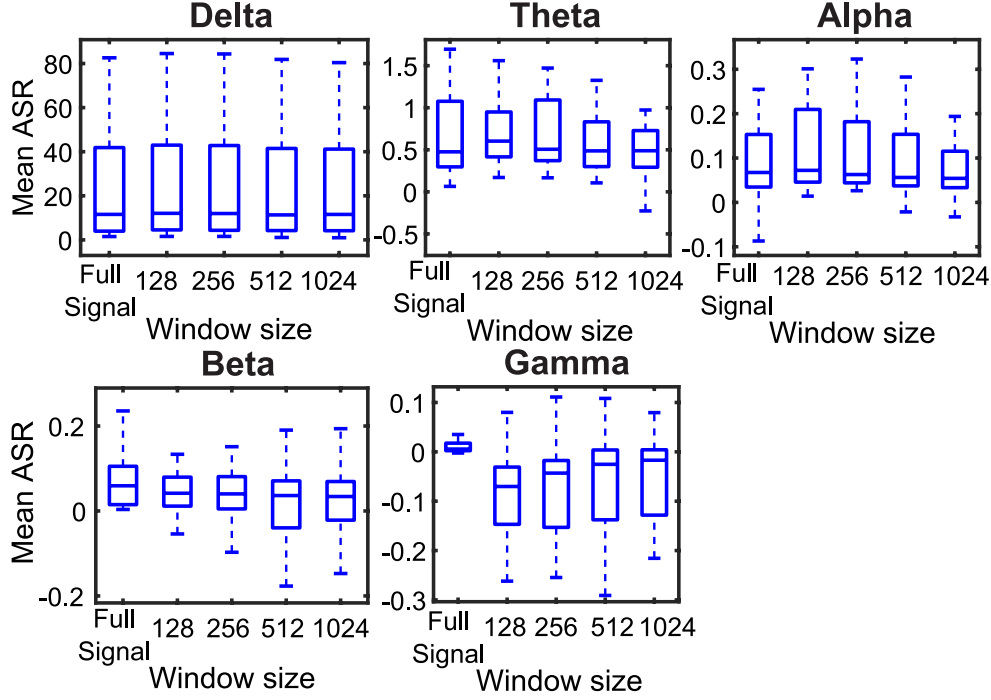


FIGURE 4.2: Mean ASR Comparison for the EMD, using the complete signal as in [Bono et al. \(2016\)](#) and different windows sizes.

$$\gamma = \gamma_o + (\gamma_1 + \gamma_0) \frac{X - X_0}{X_1 - X_0} \quad (4.5)$$

4.3.3.3 Sifting Iterations

The condition to stop the sifting is important not only because many iterations could lead to a wrong IMF decomposition but also require a significant amount of processing time. Thus, we decided to use a fixed number of iterations instead of using the stopping condition defined in (4.2).

To avoid any over sifting of the signal, the number of iterations of the sifting process is suggested to be in the range of $3 \leq n \leq 5$ provided the same number of iterations is applied to every block of the signal ([Huang et al. \(2003\)](#); [Rilling et al. \(2003\)](#)). To verify the validity of this assumption, we compared the mean ASR performance of the algorithm using a different number of iterations, as depicted in Figure 4.3. Here, it can be seen that the performance of the algorithm fluctuates at a low number of iterations and becomes almost constant at a higher number of iterations, particularly above four. Based on this observation, we decided to use four as the fixed iteration number, which will let us achieve good performance without requiring extensive computation, or uncertainty in processing time.

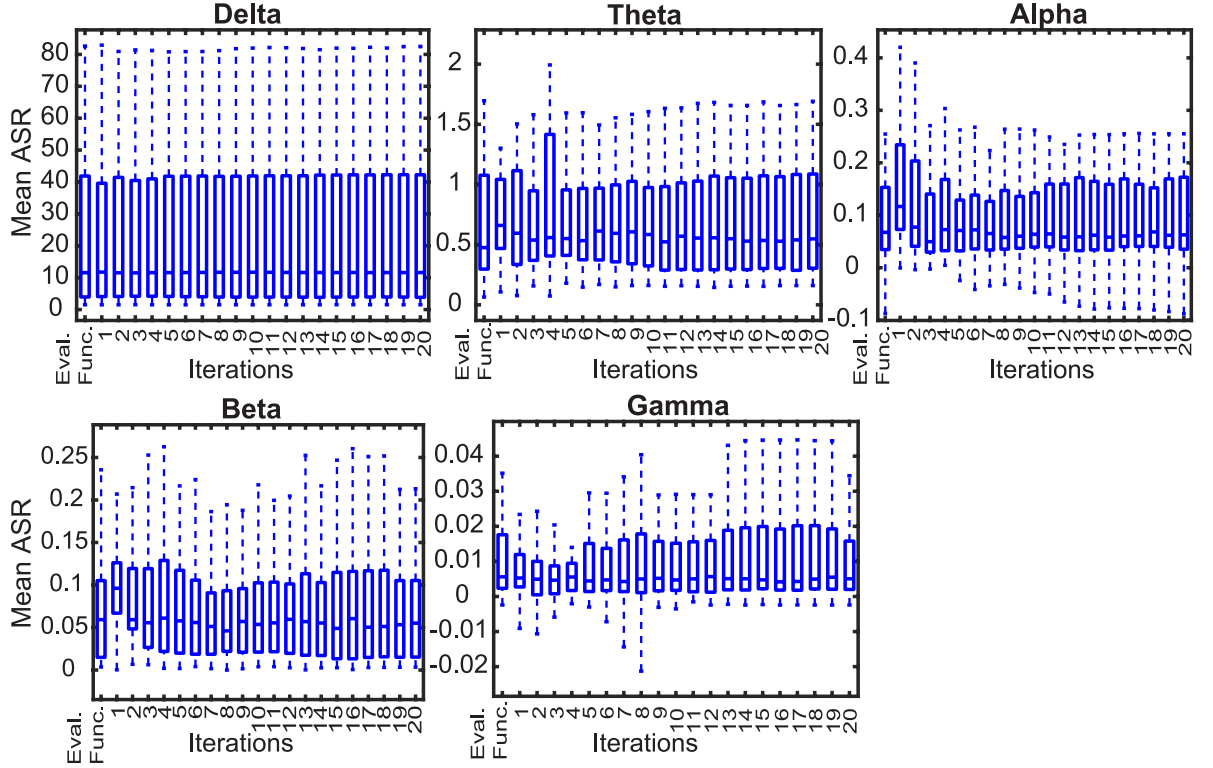


FIGURE 4.3: Comparison of the mean ASR performance in the $\delta, \theta, \alpha, \beta, \gamma$ frequency bands using the evaluation function (Equation 4.2) and different number of iterations for the sifting process.

4.3.3.4 Maximum Number of IMF

The maximum number of IMFs to be extracted from the signal will significantly impact the processing time of the system since it is necessary to repeat the sifting process four times (the adopted number of fixed iterations) for every IMF extraction. The number of IMFs that can be extracted from the signal is unknown, and it is entirely dependent upon the characteristics of the signal. Thus it is not possible to know in advance how many IMFs are going to be extracted in each window. The number of IMFs used for the EMD algorithm variates in every implementation. However, based on different studies, it was found that the number of IMFs typically used are in the range between six to ten (Abdulhay et al. (2020); Balocchi et al. (2003); Nandhini et al.; Lala and Karmakar (2020)). Hence, it was decided to set a limit to extract only 10 IMFs per window, a limit that we consider to be a proper decomposition that also meets the timing constraints of the system. It is worth mentioning that this limit is just used to prevent further processing in the case that the signal could be decomposed in even more than 10 IMFs, considering this as the worst case (largest possible computation) limit. On the other hand, no lower limit of iterations has been set.

4.3.3.5 IMF removal selection

The selection of the IMFs to be removed from the signal is made according to the J criterion as in (4.3), where the IMF with the highest J is removed from the signal. Nevertheless, while testing the algorithm it was noticed that for all the subjects and the corresponding artifacts the standard deviation has a significantly greater weight than the entropy making the J biased towards the second term of (4.3) ($\sigma_{IMF}/\sigma_{Resting}$). Thus, it is possible to simplify (4.3) using only its second term. Additionally, since the entropy is neglected, it is not necessary to perform the normalisation of the standard deviation of the signal against the standard deviation in the resting state, and therefore the J criterion effectively becomes the same as the standard deviation of the IMFs. The computational complexity of the J criterion could be reduced even further by using variance instead of standard deviation according to the argument presented in subsection 4.3.2.2.

4.3.4 Rejection Criteria

As mentioned in the background section, the algorithm from [Bono et al. \(2016\)](#) assumed that EEG data is always corrupted as it would be in a practical scenario for a mobile environment, where many elements could contribute to the contamination of the signal, and therefore a clean EEG signal is highly unlikely. Nevertheless, there is still a possibility that the EEG signal acquired is clean, and therefore its filtering is not necessary. This scenario needs to be considered, and consequently, it was decided to implement a rejection criterion in WPT and EMD to select if the node/IMF needs to be cleaned or not.

As discussed in 4.3.2.2 and 4.3.3.5, the signal is cleaned by removing the node/IMF with the highest variance for every window of the processed signal, whether or not the signal is corrupted. Hence, it is necessary to create a rejection criterion to decide if the signal needs to be filtered or not. Here, we used a threshold value as a rejection criteria to identify if the variance of the node/IMF corresponds to a contaminated signal or not. The threshold value was found by analysing an EEG signal of a person in a resting state with closed eyes (obtained from [Bono et al. \(2016\)](#)) and finding the maximum values of the variance for the nodes and IMFs extracted from the signal. These maximum values of variance are then used as a threshold value. Then, if the variance of a node/IMF is lower than the threshold set, the signal is considered clean, and the respective node/IMF is not removed, preserving the original signal.

Using the same resting-state EEG signal, in Figure 4.4 we calculated the power spectrum of channel FP2 to show the effects of the rejection criteria. The algorithm from [Bono et al. \(2016\)](#) does not have any rejection criteria or any other way to determine if the signal is clean or not; hence the signal is filtered all the time even

when not required. On the other hand, the new approach with the rejection criteria allows us to conserve the original signal and avoid any modification to the signal since it is considered to be already clean.

It is worth mentioning that the same threshold was used for all the patients. The results show that even though using this generalised approach, good results can be achieved.

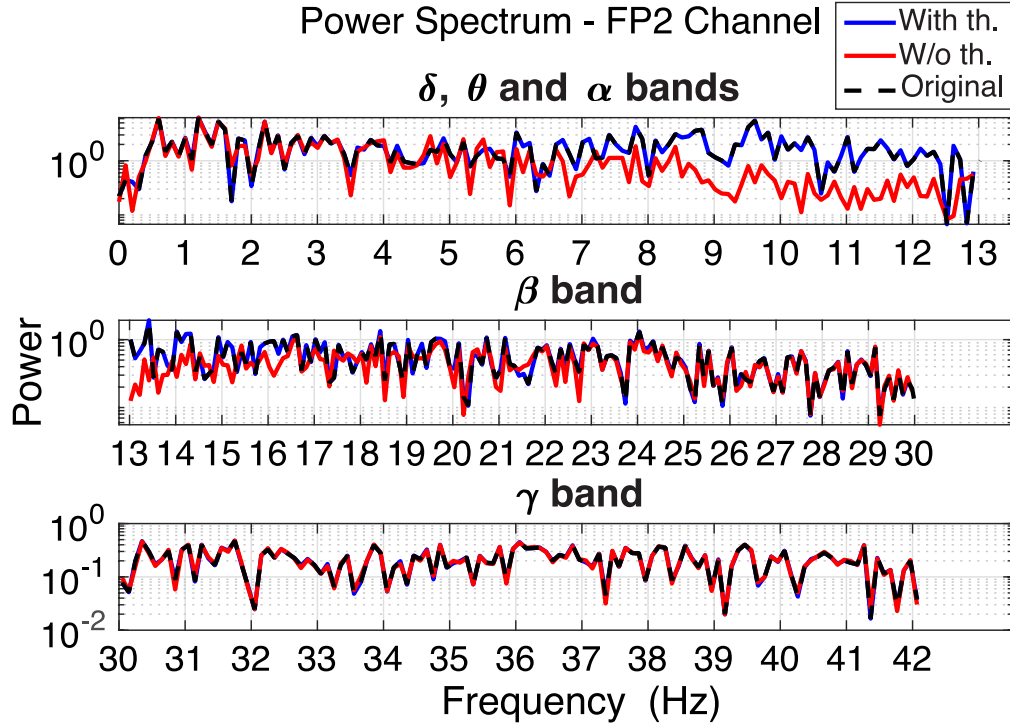


FIGURE 4.4: Power spectrum comparison using the proposed rejection criteria and without it (as in [Bono et al. \(2016\)](#)) after processing a clean signal.

4.4 Hardware Implementation

Once the new algorithm was designed based on the exploration done in section 4.3, we proceeded to perform the hardware mapping of the new algorithm. The authors of [Maharatna et al. \(2012\)](#) discussed this issue extensively in the general scenario of monitoring and concluded that instead of transmitting the data continuously to a central facility, better energy performance could be obtained by implementing the system locally at the device. Thus, such self-sustained wearable and highly demanding computational systems are best realised using a dedicated Application-Specific Integrated Circuit (ASIC) design. Therefore, here we propose a hardware architecture for a 19-channel system, a typical number of electrodes for a mobile EEG system.

4.4.1 WPT Architecture

The architecture of the WPT module is presented in Figure 4.5 and is composed of 3 main modules: WPT decomposition, Variance, and Reconstruction.

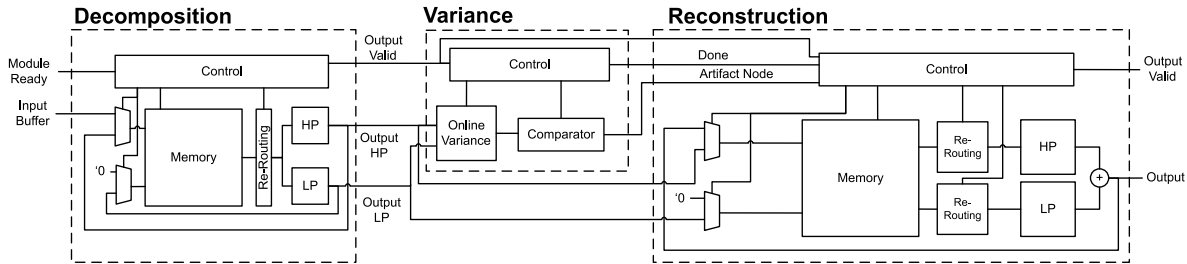


FIGURE 4.5: Architecture of the WPT module composed of the decomposition and reconstruction modules on the left and right sides.

4.4.1.1 WPT Decomposition Module

This module performs a seven-level decomposition of the input signal for the 19 channels of the EEG as described in 4.2.2.1. The quadrature filters (high- and low-pass) used for the decomposition have been implemented using the direct form realisation, with the respective decomposition coefficients of a “DB5” wavelet function. These filters were implemented using shift-and-add logic as the coefficients are fixed and pre-computed, eliminating the requirement of multipliers and reducing the required hardware.

Also, we implement only a pair of Low- and High-pass filters and schedule their usage according to their operations during different decomposition levels, a similar approach as in [Ejaz et al. \(2013\)](#), which allows us to reuse the filters and minimise hardware resources. The scheduling of the decomposition is performed in a simplistic manner. Due to the down-sampling of the signal, to receive one value of the next stage, it is necessary to have at least two values in the signal, then the decomposition is performed when two new values are available from the previous level. For instance, if there are four new values at the input, the first-level decomposition will generate two new values, then these two new values will be used for the next level of decomposition, and this process repeats for the 128 samples until the seventh level of decomposition is reached.

Typically in filters implemented with the direct form, the previous input values of the filter are stored using registers. However, since we are reusing the filters for the decomposition of multiple channels and multiple decomposition levels, it was decided to store the previous input values of the filter in RAM since it is less scarce than registers and can load the values in parallel as needed. The data was stored in RAM as in a cyclic buffer, and extra logic was added between the memory and the

filters (Re-routing module) to properly load the previous input values of the filter as needed during the decomposition.

The total memory needed in this module for each of the EEG channels is equal to $WL(2^K - 1)$, where W is the word length, L is the number of coefficients for the filter, and K is the maximum decomposition level of the signal. Here, we used 32-bit word length and ten coefficients for the filter, requiring 40.64 Kb of memory per channel.

Even though every module has been optimised to use as few elements as possible, it is impossible to instantiate one WPT module for each of the channels due to the amount of hardware requirement, especially in our FPGA implementation platform. Hence, it was decided to use only one WPT module and schedule the decomposition of the 19 channels by repeated use of this module. Although it could be possible to instantiate a couple more modules and parallelise the workload to improve the speed of the system, we stick to the decision of using only one and save as many resources as possible from the viewpoint of low energy consumption, at the cost of a slower computation. Even though this decision, our analysis (further in this chapter) shows that this arrangement still satisfies the real-time computation requirement.

4.4.1.2 Variance Module

At the output of the seven-level decomposition, a total of 128 nodes for every channel are obtained, and to find the node related to the artifacts it is necessary to calculate the variance. Since the nodes of the decomposition are calculated serially in our architecture, the usage of parallelism here would not help to improve the performance of the system. Therefore, we perform a single pass calculation of the variance as proposed in [Chan et al. \(1983\)](#) allowing online variance computation and thus finishing the entire calculation as soon as the last value is received. Once the variance is calculated for every node across all the channels, this module compares the variance values and finds the node with the highest variance of all, which is considered to be the node related to the artifacts.

4.4.1.3 WPT Reconstruction Module

As shown in Figure 4.5, the reconstruction module is connected to the output of the decomposition module. In this way, the output nodes corresponding to the last level of decomposition can be stored directly in the reconstruction memory as the calculation is done. When the decomposition module finishes calculating all the output nodes, the reconstruction module waits until the variance module finds the node with the highest variance to remove it. Removing the artifact node is as simple as overwriting the value of this node in all the channels with zero in the reconstruction memory.

As in the decomposition module, the reconstruction module needs to store the the previous input values of the filter for every node of the reconstruction, therefore RAM is also used here. Nevertheless, this module must store the last level of decomposition, requiring extra storage of 128 nodes with ten values each, doubling the necessary memory to store the values compared with the decomposition module. However, instead of performing a down-sampling as in the decomposition, the decomposition performs an up-sampling. Then, a zero value is introduced for every new value obtained by the filter, resulting in half of the stored values being equal to zero at any time. Knowing this in advance, it is not necessary to store those zero values, which save half of the space needed in the memory, allowing to use of the same memory as in the decomposition module.

The reconstruction is performed using a Low- and High-pass filter followed by an addition to combine the output signals from them. Then, the result is stored in the RAM, and the process is repeated until the reconstruction is completed. The coefficients of the filter are the reconstruction coefficients of the “DB5” wavelet function implemented using shift-and-add logic as in the decomposition filter implementation.

The scheduling to perform the reconstruction of the signal is different compared to the one implemented for the decomposition. The reconstruction starts from the last level(seven) and performs the respective filtering of every node in that level, and this is then repeated for the previous level until the first level is reached. From this point, every output is followed by an up-sampling so the filtering is repeated with a zero as the new input, starting from the first level then going to the next level. Afterwards, the output of this level is used for the previous levels and the process is repeated until the first level is reached. This sequence is repeated starting from the second level, then the third level and so on until the last level is performed.

4.4.2 EMD architecture

The EMD architecture is presented in Figure 4.6. It is composed of two main modules: the memory unit and the arithmetic logic unit.

4.4.2.1 Memory Unit

The memory unit stores all the signals involved in the calculation process of the EMD, such as the upper and lower envelopes and their mean, the intermediate signal obtained during the sifting iterations, the extracted IMFs and the residual signal, among others. Every block in the Memory unit (except for the variance that uses registers to store the values) uses a total of WS memory elements, where W is equal to

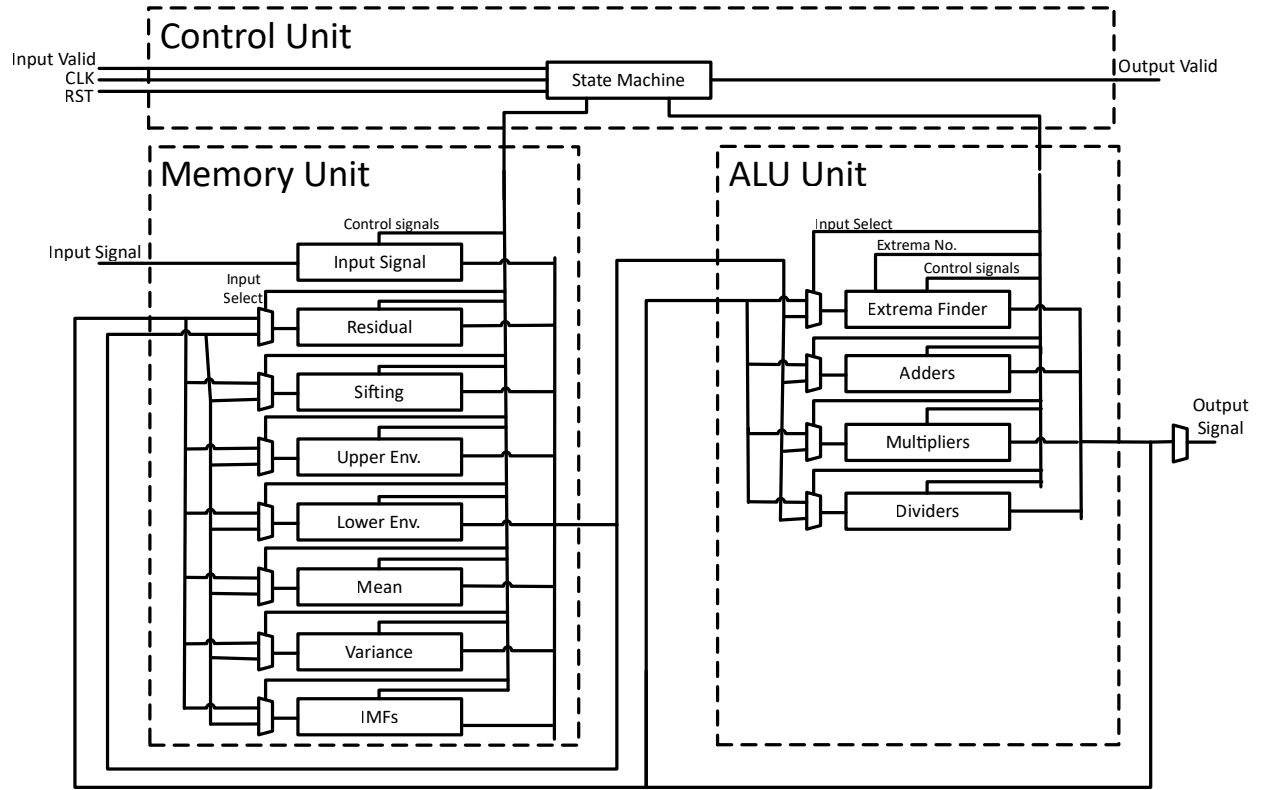


FIGURE 4.6: General architecture of the EMD module. The memory unit stores all the data used during the processing and the ALU unit performs the calculations of every stage in the process.

the word length and S equal to the window size. Using 32 bits and a 128 window size, every block requires 4096 memory elements resulting in a total memory of 81.92 Kb.

4.4.2.2 Arithmetic Logic Unit (ALU)

The ALU unit comprises adders, multipliers, and dividers that are reused across the process. Those elements are limited in FPGAs and reusing them helps to save resources in general. Additionally, the ALU contains a module used to find the extrema points by comparing every point in the signal with its neighbours and determining if it is a maximum/minimum point (Cormen et al. (2009)). All the components of the memory unit and the ALU are connected, and the input to every module is controlled by the control unit. This allows the connection of the inputs and outputs of the different modules depending on the type of operation to be performed or the elements needed. For instance, while performing the interpolation we need the adders and multipliers to be connected to perform the proper operation, while the connection may change to perform a different operation like the mean calculation, for example.

Due to the nature of the EMD processing, the algorithm is performed step by step instead of a parallel calculation. The module also processes one channel at a time and it is necessary to wait until the process is finished before starting another one. It could be possible to instantiate more than one EMD module to process multiple channels simultaneously and reduce the processing time at the cost of increasing the resources used and hence the energy consumption. This study opted to use only one instance of the EMD module since the time required to process the 19 channels is still less than the 180 ms established in our requirements, this based on the 500 Hz sampling rate used in this study.

4.5 Results

To verify the correct behaviour of the proposed architecture for the new algorithm, we performed a series of functional validations using EEG signals contaminated with simulated artifacts and with real artifacts. The cleaning performance from these validations was compared against the performance of the algorithm presented in [Bono et al. \(2016\)](#). This comparison allows us to demonstrate how our new algorithm manages to achieve a better performance than the original without its shortcomings. Furthermore, as a proof-of-concept, the architecture of the new algorithm was implemented on an FPGA, and an analysis of its resource usage and power consumption was made. Finally, a comparison with other systems in the literature was made. The data used for the validation is the same data collected by [Bono et al. \(2016\)](#), described in the previous subsection 4.3.1. For the hardware design, a fixed point representation(S(21,10)) of the data was used.

4.5.1 Validation using a simulated artifact

For the validation process, we created an EEG signal corrupted with a simulated blinking eye artifact. This is a common and well-defined artifact in EEG. Using a simulated artifact allows us to have a reference signal and evaluate the performance of the system to clean the signal. The simulated corrupted EEG signal is defined as in (4.6), where $EEG_{Orig.}$ corresponds to a resting state EEG signal obtained from [Bono et al. \(2016\)](#), **Artifact** is the simulated artifact signal, and λ is a factor used to simulate different levels of contamination in the signal, going from zero (original signal) until 15 which is a heavily contaminated signal.

$$EEG_{Corrup.} = EEG_{Orig.} + \lambda(Artifact) \quad (4.6)$$

The artifact signal was created using the positive half-cycle of a 1 Hz sine wave to resemble a blinking eye artifact with a peak at $25\mu V$. This signal was added to the

channels "FP1", "FP2", "F3", "F4", and "FZ". These channels correspond to the electrodes closer to the eyes and therefore would be more affected by a blinking eye artifact. Finally, random noise was introduced to all channels by combining 50 different sine waves with a random frequency (0.1-125 Hz), random phase ($0-2\pi$), and random amplitude ($0-5\mu V$), a similar approach to Yeung et al. (2004).

The simulated corrupted EEG was then cleaned using the design from Bono et al. (2016) and the proposed hardware architecture. In Figure 4.7 we present the Root Mean Square Error (RMSE) between the original signal and the cleaned signal. Here, it can be seen that when the λ factor is equal to zero, the RMSE is close to zero for the proposed system since the signal was not modified, compared to the design from Bono et al. (2016), which is modifying the signal even though it is considered clean. Furthermore, it can be seen that as λ increases, the contamination of the EEG is greater and causes a decrease in the performance of the algorithm, which is reflected by the increase of the RMSE.

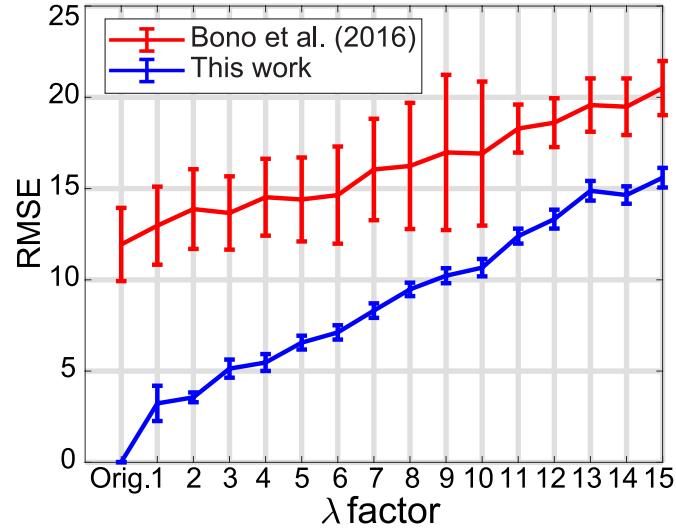


FIGURE 4.7: Comparison of the RMSE after cleaning the simulated eye blink artifact. The line connects the median value of the RMSE across the 19 channels. The error bars represent one standard deviation from the RMSE values across the 19 channels.

In Figure 4.8, we presented the original signal, the heavily contaminated ($\lambda=10$) simulated corrupted signal, and the clean signal using the modified algorithm and the implementation in Bono et al. (2016). In this figure, it can be seen that the blinking artifacts were satisfactorily removed from the corrupted signal using the modified algorithm proposed here, while the algorithm from Bono et al. (2016) was unable to remove them entirely from the signal. Also in this figure, the modified algorithm manages to maintain the morphology of the signal compared to Bono et al. (2016), which could be related to the introduction of the threshold to avoid over-cleaning the signal.

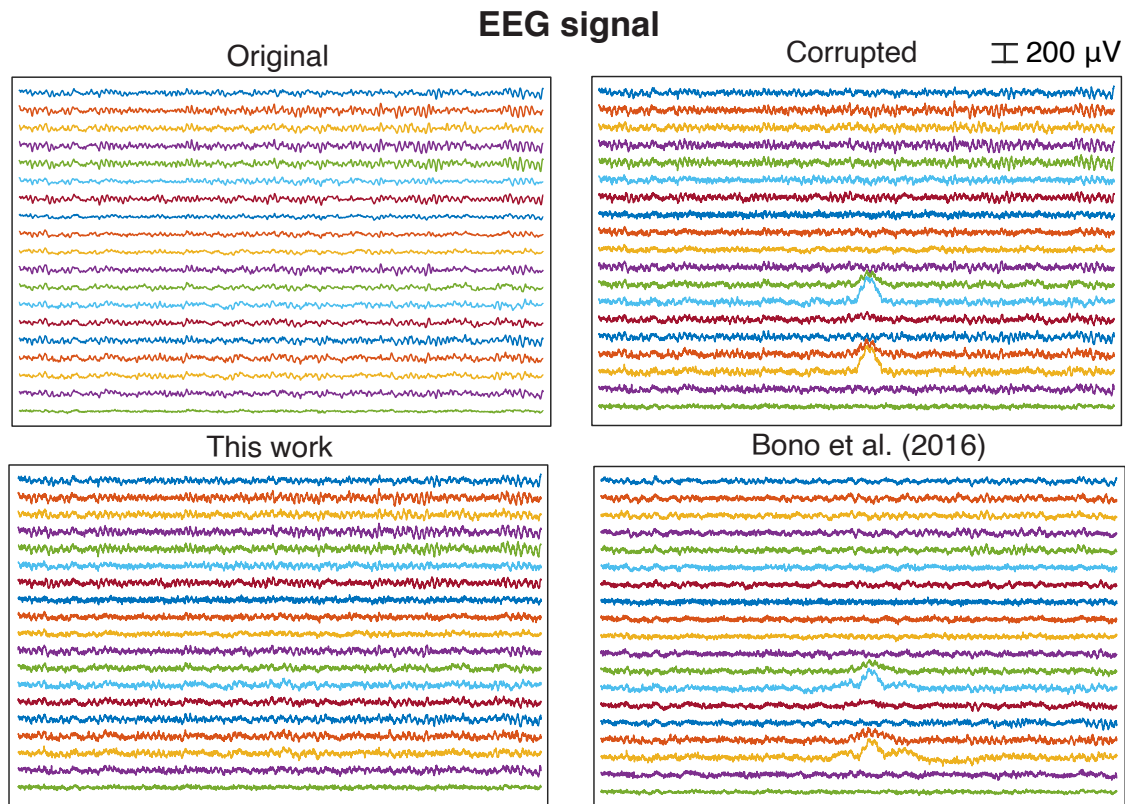


FIGURE 4.8: Comparison between the original and corrupted signal against the cleaned signal. The graphs show a resting state clean signal, a corrupted signal composed of the clean signal and simulated artifacts, and underneath the output signal after performing our proposed algorithm and then finally using the algorithm presented by Bono et al. (2016).

Furthermore, in Figure 4.9, we presented the power spectrum of the signal in the FP2 channel, one of the most affected by the simulated blinking eye artifact. We can see that the proposed system can filter the significant eye blinking artifact concentrated in the lower part of the delta band in the power spectrum. Moreover, it can be seen the cleaned signal possesses a closer spectrum to the original signal (delta band) than the implementation on Bono et al. (2016). Additionally, around 11 Hz, it can be seen that the modified algorithm avoided performing an over cleaning of the signal due to its

thresholding while the algorithm from Bono et al. (2016) cleaned the signal even though it was not needed.

This analysis shows that the modified algorithm can remove the synthetic eye-blinking artifacts introduced into the signal with a similar or better performance than Bono et al. (2016).

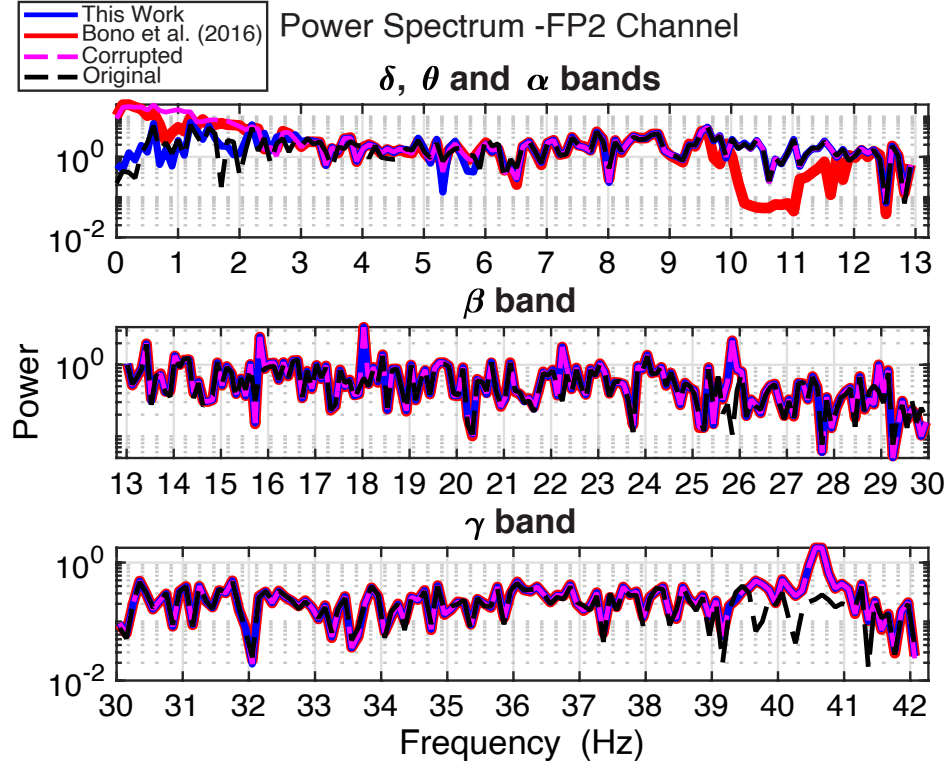


FIGURE 4.9: The graphs show a comparison in the power spectrum between the signals shown in Figure 4.8 across the different frequency bands.

4.5.2 Validation using real artifacts

Now that the modified algorithm has been tested with synthetic artifacts, it is time to validate its performance with more realistic artifacts. The input used for this validation was the same data from Section 4.3, which was obtained from five subjects generating seven artifacts: eyes blinking, head movement (up and down), chewing, speaking, hand movement, head movement (left and right) and shaking head.

The performance was evaluated by visual inspection comparing the clean signal produced by the software implementation from Bono et al. (2016) against the one produced by the hardware implementation proposed, as in Figure 4.10. The signal used here corresponds to the “O1” electrode of an EEG signal corrupted with muscle artifacts due to head shaking. It can be seen that both algorithms were able to remove the significant spikes in the EEG signal related to the muscle artifacts. However, there

are segments in the signal where the performance of the hardware implementation is not as expected. For instance, although the visual inspection shows that the signal is cleaned without affecting the shape of the original signal (Figure 4.10, window 1 and 3), some parts of the cleaned signal exhibit 'roughness' compared to the software output (Figure 4.10, window 2). This is caused by the adopted linear interpolation method instead of the cubic spline used originally. Furthermore, it is possible to observe that some parts of the signal present a sharp change in the signal (Figure 4.10, window 3). This is caused due to the discontinuities introduced when windowing of the signal. It may be worth exploring the possibility of using a windowing technique to see if these effects can be reduced, or this is an unavoidable limitation of the algorithm. These observations allude to the practical limitation of a hardware implementation of the system.

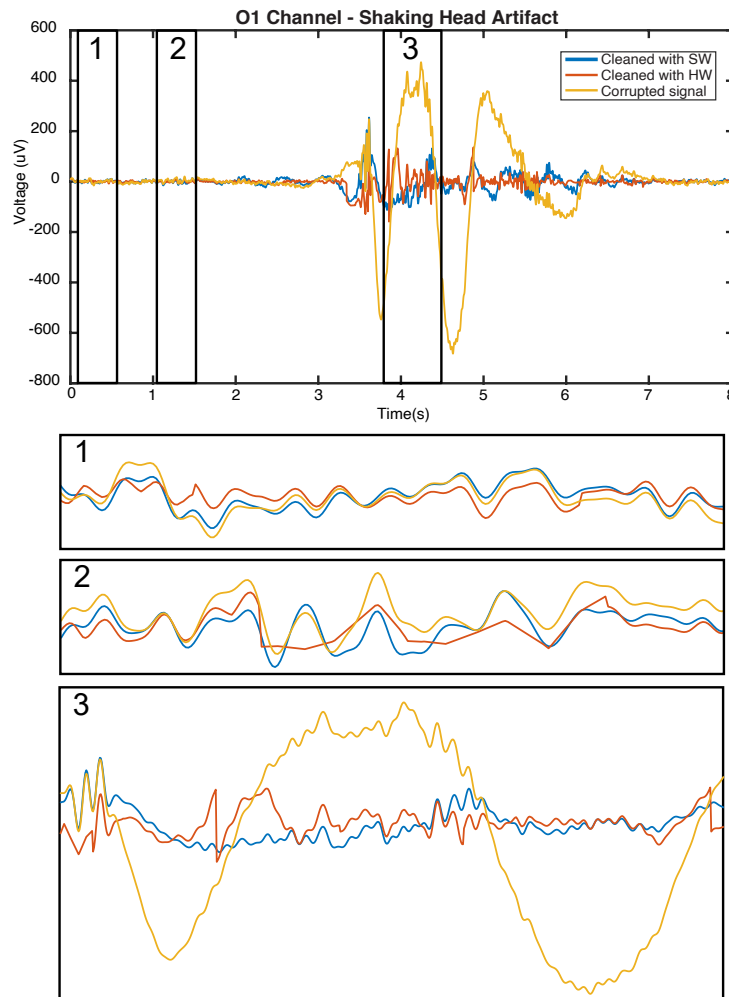


FIGURE 4.10: Visual comparison between a corrupted signal (muscle artifacts) and the processed output signal using the algorithm in software (Bono et al. (2016)) and hardware (this work). Window 1 shows how the signal is cleaned while keeping its shape. Windows 2 and 3 show the effects of the interpolation and the windowing of the signal respectively.

Additionally, the hardware implementation was simulated and benchmarked for each subject and artifact using the mean ASR, defined in section 4.3. It is worth noting that the performance of the ASR is dependent on the frequency band. Usually, the artifacts introduced to the signal could be contained in a specific frequency band, and its removal will reduce the power in that particular frequency band. This removal will then lead to a positive increase in ASR value. On the contrary, if no artifact was removed from the signal in a particular frequency band, the power will remain the same, and the ASR value will be close to zero.

The mean ASR was then calculated for every frequency band using the hardware implementation and the software implementation from [Bono et al. \(2016\)](#). The results obtained are shown in Figure 4.11. From these results, it can be concluded that the artifacts introduced in the signal are mainly contained in the delta band, which leads to higher ASR values compared with other frequency bands. This could also be seen in Figure 4.10, where the high amplitude oscillating signal seems to have a frequency between 0.5 and 1.5 Hz. Nevertheless, the ASR in the other bands are values greater than zero, showing that there was some cleaning of the signals for these frequencies too, although not as significant as in the delta band. Furthermore, the results in figure Figure 4.11 show that the performance in terms of ASR between the hardware implementation and the software implementation from [Bono et al. \(2016\)](#) are similar. The main difference in the results is in the gamma band, where it can be seen that the mean ASR obtained with this work is slightly higher than the work from [Bono et al. \(2016\)](#) which was close to zero, which means that this frequency band was barely modified. These results validate that the hardware system is working correctly and that the modifications done to the algorithm do not have a significant impact on its final performance.

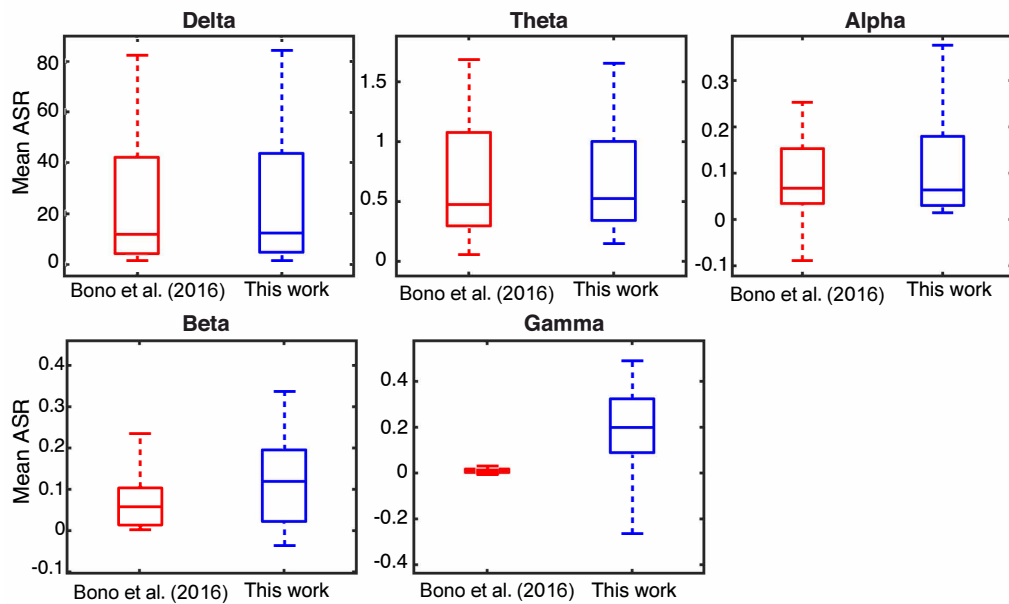


FIGURE 4.11: Comparison of the performance achieved using the algorithm from [Bono et al. \(2016\)](#) against the modified algorithm presented in this work.

4.5.3 Hardware Resources

After validating the functionality of the architecture in simulations, the whole system was synthesised using Quartus Prime software for a Stratix IV GX EP4SGX230K FPGA, as previously described in Chapter 3. This development board is ideal for signal processing applications due to its vast number of logic elements available, a high number of Digital Signal Processing (DSP), and low power consumption. The compilation results are shown in Table 4.2. The synthesis results demonstrate that our design requires only 28% and 11% of the total logic elements and memory respectively, and only 10% DSP blocks which are more limited in FPGAs.

TABLE 4.2: Synthesis results.

	Logic Elem.	RAM bits	DPS Blocks	Freq. (MHz)
WPT(Dec)	7,139	772,160	0	19.66
WPT(Rec)	6,851	775,200	0	51.54
Variance	16,694	0	47	4.74
EMD	19,600	81,920	78	3.23
Final	50,445	1,629,280	124	3.32

The synthesised design shows a maximum operating frequency of 3.32 MHz in the worst-case scenario. In the system we used, the EEG signal is sampled at 500Hz. To see whether the system satisfies the real-time processing requirement, we calculate the number of clock cycles required to perform the entire computation, and the breakdown of the number of required clock cycles for each of the modules is shown in Table 4.3. The total number of clock cycles for the entire 19-channel system is 569,298 (a total of 171.4 ms with a clock at 3.32MHz), which under the constraint of window size of 128 has to be completed within a time equal to $128/(\text{sampling rate})$. Therefore, using the 3.32 MHz as the maximum frequency, it is possible to process a signal up to ≈ 741 samples per second while still satisfying the real-time computation requirement.

TABLE 4.3: Clock cycles for calculation.

	General formula
WPT(Dec)	$(K + 1)2^K$
WPT(Rec)	$K(2^{(K+1)}) + Ch + 1$
EMD	$W[\#IMF(1 + 5 * \#Sift) + 12] + 10$

4.5.4 Architectural Complexity

The architectural complexity of the whole system was evaluated by using a 2-input NAND gate equivalent for every module in the system. The rough estimation of the

resources needed in terms of basic logic gates is presented in Table 4.4. The calculation was done by estimating the logic blocks needed for every module such as adders, multipliers, dividers, and multiplexers in terms of their logic gate equivalent. This complexity analysis provides a rough estimation of the number of the logic gates required if the design is implemented on an ASIC. This allows a fundamental estimation of different design elements, such as the area and power, based on the parameters of the ASIC technology selected for the implementation.

TABLE 4.4: Logic Gates used for every module.

	Dec.	Rec.	Var.	EMD	Total
NAND Gates	60,272	72,410	236,446	180,088	549,216

4.5.5 Power Analysis

Power consumption is another important characteristic to consider in wearable systems. To analyse the power consumption of this design, we used the available software provided by the FPGA manufacturer. The dynamic power reported by the tool was equal to 11.28 mW, using a clock frequency of 3.32 MHz and 0.9V as the VCC voltage of the FPGA. It is worth mentioning that this power consumption can be reduced even further if the clock frequency or the supply voltage is reduced. However, in this design, we decided not to follow this approach since this is the first stage of further signal processing, and it has to be performed in the lowest possible time.

In this design, we believe that it is better to perform the whole processing on edge instead of sending the information to another device (i.e. cellphone) and perform the processing there. By doing the processing in the device, we avoid using energy for the data transmission and allow faster processing of the signal.

To put this into perspective, using the previously discussed NRF8001 and NRF5281 BLE transceiver chips, the power consumption during transmission is higher than our device, reaching up to three times higher in the case of NRF8001. Additionally, the transmission time using these BLE transceivers for a 32-bit word, 128 samples and 19 channels at the maximum transmission rate (1 Mbps and 2 Mbps for the nRF8001 and nRF5281, respectively) will be equal to 77.824 ms for the nRF8001 and 38.912 ms for the nRF5281. This time is equivalent to 45% and 22.70% of the total time (171.4 ms) required to perform the complete algorithm on the FPGA implementation. This estimation corresponds only to the transmission of the results since the input signal can be transmitted during the sampling. Also, the estimation considers excellent communication without retransmission of the packages, which would increase the processing time and the power consumption.

Therefore, the system we proposed here is a viable option to remove artifacts requiring low power consumption and low calculation time, ideally for applications requiring fast signal processing, such as wearable EEG systems.

4.5.6 Comparison

In the literature, few architectures are proposed for the WPT and the EMD separately, but there is no implementation yet that combines these two algorithms and performs the cleaning of the signals. Therefore, it is not possible to directly compare our architecture with anything available in the literature. However, it is possible to compare the individual modules of the system with other implementations. This comparison is presented in Table 4.5.

TABLE 4.5: Comparison with similar designs in terms of resources used.

Design	Process	Logic Elem.	RAM bits	Freq. (MHz)	Word size
Chilo and Lindblad (2008)	DWT	506	3,200	50	18
Mahmoud et al. (2008)	WPT	5,980	N/A	0.55	8
Mei-hua et al. (2006)	DWT	1,835	10,172	29	16
This work	WPT (Dec)	7,139	772,160	19.66	32
This work	WPT (Rec)	6,851	775,200	51.54	32
Chen et al. (2016)	EMD	3,355	N/A	50	32
Wang et al. (2010)	EMD	27,969	N/A	N/A	32
Hong and Bao (2012)	EMD	28,037	0	12.5	14
This work	EMD	19,600	81,920	3.23	32
This work	WPT-EMD art. rem.	50,445	1,629,280	3.32	32

With respect to the Wavelet decomposition, we can see that the implementations presented by Chilo and Lindblad (2008), Mahmoud et al. (2008) and Mei-hua et al. (2006) present a lower number of logic elements and memory elements than our implementation. Nonetheless, these designs use a smaller word size compared with our system that uses 32 bits. Thus, if the same word size were used, the number of logic elements needed would be closer to our implementation. In terms of the memory, if we considered that the design from Mei-hua et al. (2006) needs almost double the memory because of the word size, and double that because DWT only process half the elements than WPT, and finally multiply that for the number of channels used, that implementation may end up using even more memory than ours. Only the implementation from Chilo and Lindblad (2008) requires fewer logic elements and less memory than our design, even with the considerations mentioned above.

For EMD, the proposed architecture uses lower resources than the implementation presented by Wang et al. (2010) and Hong and Bao (2012). The only implementation

that uses fewer resources is the one reported by [Chen et al. \(2016\)](#). Nevertheless, this implementation only does the EMD without filtering the signal compared to our module that also performs the filtering of the artifacts. To find the IMF related to the artifact, the proposed EMD module calculates the variance of every IMF in the same way that the Variance module used in the WPT. If we consider that this extra processing will need the same number of logic elements as the Variance module, and we subtract them from the reported by the synthesis of the EMD; the result will be that our proposed implementations require fewer logic elements than the one reported by [Chen et al. \(2016\)](#). About the memory, only [Hong and Bao \(2012\)](#) reported its usage and it requires less than our implementation.

Moreover, another comparison has been made with other artifact removal systems in terms of NAND gates, which is presented in Table 4.6. The proposed architecture requires a lower number of NAND gates from the different implementations, except for the design proposed by [Acharyya et al. \(2018\)](#). Nonetheless, the primary process done on the design presented by [Acharyya et al. \(2018\)](#) is related to the DWT of the signal, using the Haar wavelet (with only two coefficients). Therefore, by comparing only the NAND gates of the logic related to the WPT in the proposed architecture, the number of gates will be lower than the design from [Acharyya et al. \(2018\)](#).

TABLE 4.6: Comparison with similar designs in terms of architectural complexity.

	This Work	Acharyya et al. (2018)	Daly et al. (2014)	Hu et al. (2015)
NAND Gates	549,216	285,870*	5,931,641*	23,726,566*

*Approximation done using the analysis from [Acharyya et al. \(2018\)](#)

4.6 Conclusion

This work proposes a new algorithm for the separation of artifacts in wearable EEG systems. The functional validation using simulated and real artifacts demonstrated that the algorithm effectively removed the artifacts in those signals. The algorithm was mapped into a hardware architecture and implemented in an FPGA as a proof-of-concept. This implementation achieves a 3.32 MHz operation frequency, which results in a processing time of 171.4ms, satisfying the real-time operational constraint of our system. According to the application, the operation frequency can be reduced to decrease the system's power consumption at the expense of increasing the processing time. However, for this application, it was decided to use the maximum frequency to allow further processing of the signal while still meeting the 180 ms requirement. Furthermore, the architectural complexity analysis showed that the design would require about 549K NAND gates, representing low resource utilisation. This estimation indicates that the proposed design could be implemented on an ASIC,

improving the design energy efficiency and occupy a smaller hardware area.
Therefore, the proposed algorithm will be an appropriate fit for our envisaged system.

Chapter 5

Phase Lag Index hardware implementation for real-time network connectivity formulation

5.1 Introduction

Functional brain connectivity from non-invasive Electroencephalography (EEG) signals, in recent years has been used to extract information flow pathways in the brain ([Friston \(2011\)](#)) for different neurological conditions such as Alzheimer disease ([Engels et al. \(2015\)](#)) and Parkinson's disease [Stoffers et al. \(2008\)](#). Several methods have been proposed in the literature for the construction of the functional connectivity network, such as Phase Locking Value (PLV), Phase Lag Index (PLI) and Global Synchronization Index (GSI). Among them, PLI has emerged as an effective technique inherently robust to the confounding effects of volume conduction found in the other methods. Its effectiveness has been shown in several studies, such as [Kasakawa et al. \(2016\)](#) and [Yu et al. \(2016\)](#).

Although functional brain connectivity provides useful clinical information pertinent to different disease conditions, its application in treatment is rather restricted. In a typical application scenario, the functional brain connectivity is calculated offline after the study is done, owing to its computational complexity. However, since it alludes to information flow in the brain, we envision that functional brain connectivity may assist in providing the basis of real-time neurofeedback that could be used in the treatment of different neurological conditions. Therefore, it is essential to overcome the offline processing restriction that acts as a roadblock to exploring new applications of functional brain connectivity that require real-time information about the network.

To alleviate this roadblock, we propose a new hardware architecture to construct the functional brain connectivity network using PLI. This architecture was created in a modular way that would allow for scalability of the system to N number of channels, making it suitable for different studies using different electrode arrays. In this case, a 16 channel system with a word size of 32 bits was synthesised as a proof of concept. The system was tested on simulations, where the results demonstrate a decrease in the PLI calculation time against the MATLAB software used for offline processing, reaching a maximum acceleration of two orders of magnitude.

5.2 Background

5.2.1 The Computational flow of PLI calculation

The calculation of the PLI can be done using the Equation 2.12 previously discussed in Chapter 2. First, it is necessary to estimate the instantaneous phase for every sample in the window. The estimation of this instantaneous phase can be done using the analytical signal as defined in Equation 5.1. The analytical signal ($y(t)$) is a complex signal with a real component ($x(n)$) corresponding to the original signal and an imaginary component ($\tilde{x}(n)$) corresponding to the Hilbert Transform (HT).

$$y(t) = x(t) + i\tilde{x}(t) = A(t)e^{i\phi(t)}; \quad (5.1)$$

The HT of a signal is equivalent to the same input signal but with a phase-shifting of 90° and it is defined as the convolution of the original signal $x(t)$ and the function $-(1/\pi t)$, as shown in Equation 5.2 (Cizek (1970)).

$$\tilde{x}(t) = -\frac{1}{\pi} \int_{-\infty}^{\infty} x(\tau) \frac{d\tau}{\tau - t} = -\frac{1}{\pi t} * x(t) \quad (5.2)$$

Once the analytical signal is calculated, it is then possible to obtain the instantaneous amplitude ($A(t)$) and the instantaneous phase ($\phi(t)$) of the signal as in Equation 5.3, using the Arctangent function.

$$A(t) = \sqrt{\tilde{x}(t)^2 + x(t)^2}; \quad \phi(t) = \arctan \frac{\tilde{x}(t)}{x(t)} \quad (5.3)$$

Consequently, the complete process followed to calculate the PLI value is depicted in Figure 5.1. The process depicted here involves computationally complex calculations such as calculating the Hilbert transform and the Arctangent function. The complexity of these calculations could lead to high hardware resource usage and higher energy

consumption, which is not ideal for a wearable system such as the one we envision. Therefore, it was necessary to analyse these calculations and evaluate the different options available to optimise them and reduce their complexity.

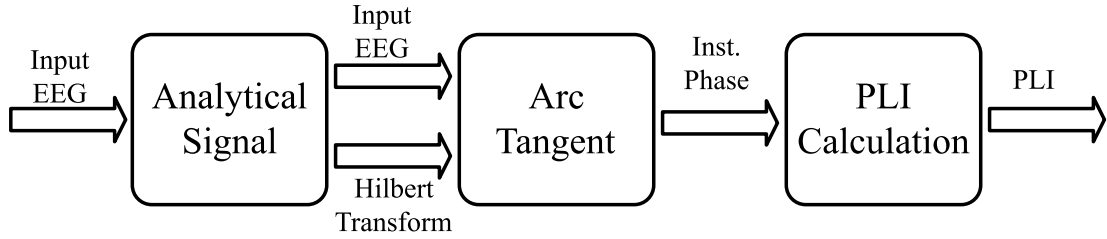


FIGURE 5.1: Diagram of the computational flow used for the calculation of the PLI index.

5.3 Algorithm complexity and accuracy analysis of PLI components

5.3.1 Implementation of the Hilbert transform

The analytical signal is a complex signal where the real component corresponds to the input signal, and its imaginary component corresponds to the Hilbert transform (HT) of the input signal, as in Equation 5.1. Two main approaches can be used to calculate the Hilbert transform of the signal: the Fourier transform and a Hilbert transformer filter.

5.3.1.1 Hilbert transform using Fourier transform

The standard approach to obtain the analytical signal (which includes the HT of the signal) is based on the Fourier transform (FT) of the signal (Luo et al. (2009)). The analytical signal can be obtained by performing the FT of the function defined in Equation 5.2. From this transformation, we found that the analytical signal can be defined as in Equation 5.4, where $X(f)$ and $Y(f)$ correspond to the FT of the input signal and the analytical signal, respectively (Marple (1999)). After defining $Y(f)$ as in Equation 5.4, this signal is converted back to the time domain using the inverse FT (IFT), obtaining the analytical signal finally as in 5.1.

$$Y(f) = \begin{cases} 2X(f), & \text{if } f > 0 \\ X(0), & \text{if } f = 0 \\ 0, & \text{Otherwise} \end{cases} \quad (5.4)$$

This is the standard method used for the calculation of the analytical signal due to its simplicity. The highest complexity of this method lies in the calculation of the FT and the IFT.

5.3.1.2 Hilbert transformer Filter

The alternative method to calculate the HT of a signal is using a Hilbert transformer filter. The principle behind this method is that the HT of a signal (Equation 5.2) can be obtained using a 90° phase shifter filter, with a linear time-invariant frequency response and unity gain. This approach has been used in ultrasound applications such as in [Qiu et al. \(2010\)](#) and [Assef et al. \(2019\)](#), where it has shown good performance with low hardware resource utilisation.

The Hilbert transformer's advantages are its low complexity and the lower hardware resource usage to perform the HT of the signal.

5.3.1.3 Comparison

The usage of a Hilbert transformer is an alternative that is worth exploring since the complexity and hardware usage could be less than the standard approach of using the FT. For this reason, here, we present an analysis of the impact that using a Hilbert transformer will have on the accuracy of calculation of the PLI. For the analysis, we used real EEG data from four different patients. The data used for this test was generated by [Shoeb \(2009\)](#) and is available online.

The first step in the Hilbert transformer filter design is the selection of the number of coefficients for the filter. Different filters were created using the MATLAB filter design assistant, varying the number of coefficients and using different design methods (least squares and equiripple). Then, the HT was calculated for every filter and compared with the HT generated using the FT. The Root Mean Square Error (RMSE) between these two HTs was calculated for every filter design and every patient. The mean RMSE across all the patients is presented in Figure 5.2. In this figure, it can be seen that the error decreases as the number of coefficients increases. Nonetheless, the RMSE curve is asymptotic, and the error does not decrease even if the number of coefficients keeps increasing. Based on this behaviour, we decided to use the equiripple method due to its better performance, and a total of 218 coefficient since it is the minimum number of coefficients needed to achieve the lowest possible error.

It is worth mentioning that in Figure 5.2, even the lowest RMSE achieved is still significant and may have a significant impact in the calculation of the PLI. To evaluate the impact introduced by the Hilbert transformer, the PLI was calculated using the HT obtained using the FT and the Hilbert transformer filter. The RMSE between both PLI

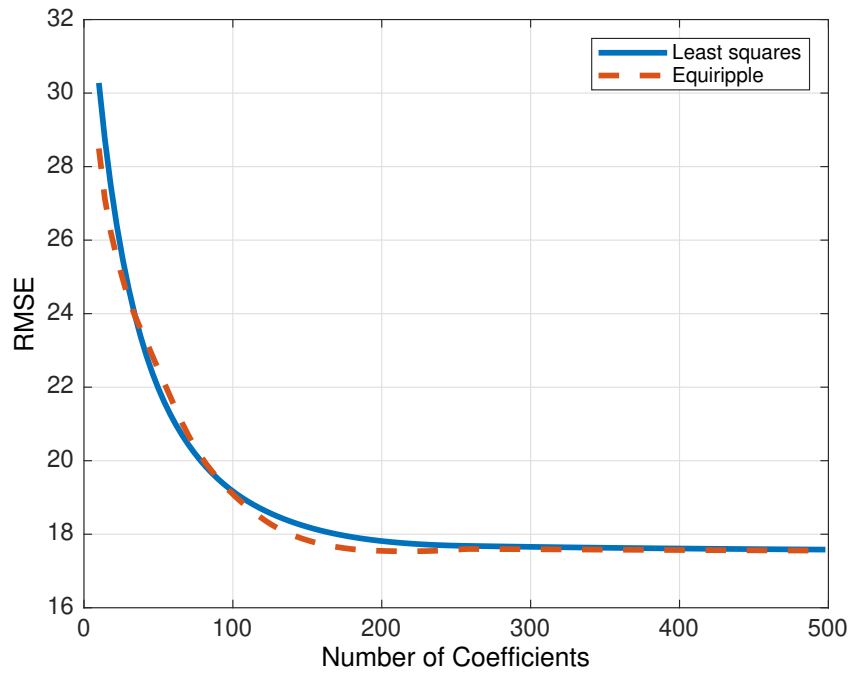


FIGURE 5.2: Mean RMSE across all patients for the Hilbert transform calculated with the Fourier transform and the Hilbert transformer filter.

calculations was done for 50 different windows from every patient, with a window length of 128 samples. The results are shown in Figure 5.3. This figure shows that the RMSE value varies among every window calculation, with a mean RMSE value between 0.09 and 0.24. Considering the maximum value for the PLI is one, then the error introduced in the calculation will be in the range of 9% and 24%, which is not ideal.

The analysis done here shown that even if the usage of Hilbert transformers for the HT could be a promising alternative to reduce the complexity and hardware resources usage, the error introduced into the signal will significantly impact the calculation of further stages. Therefore, it was not to use the Hilbert transformer for the HT and instead use the FT for the HT, as it is commonly done.

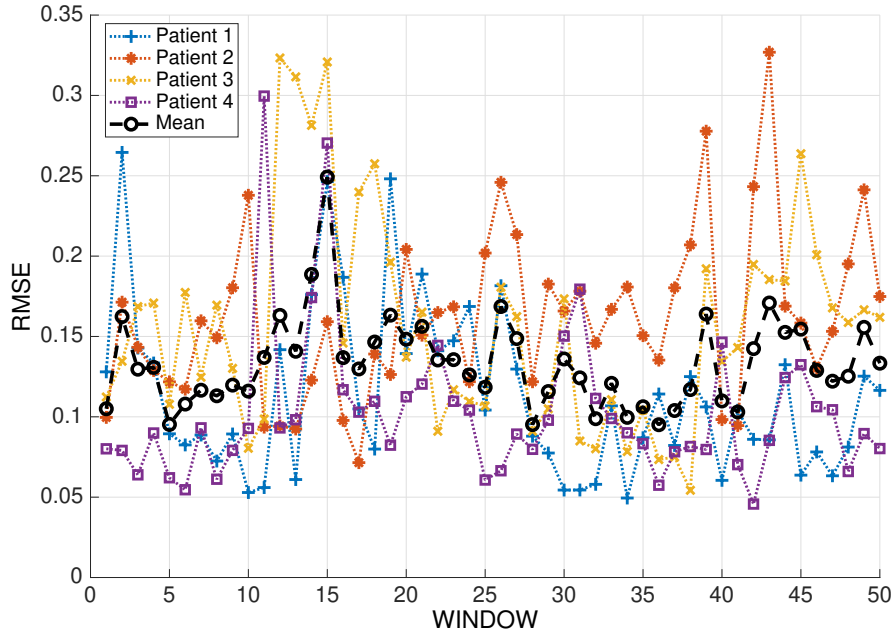


FIGURE 5.3: RMSE of the Hilbert transform calculated with the Fourier transform and the Hilbert transformer filter.

5.3.2 Instantaneous Phase Calculation

The instantaneous phase can be obtained by using the analytical signal as in Equation 5.3. In this equation, it is necessary to calculate the angle in the euclidean plane for the point represented by the HT and the input signal, as in Figure 5.4. This angle can be obtained using the arctangent function for two points, also know as Atan2. Implementing this function in hardware can be done using either a lookup table or the COordinate Rotation DIgital Computer (CORDIC) algorithm.

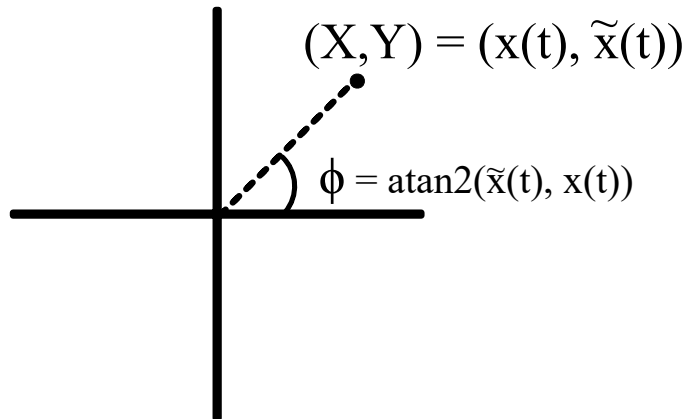


FIGURE 5.4: The Atan2 calculation is used to calculate the angle ϕ of a given point (X, Y) in the Cartesian plane.

5.3.2.1 LUT

The most straightforward approach to implement any function in hardware is to store all the possible output values of the function in a Lookup Table (LUT). In this way, the input value will be used to find the address of the corresponding output value in the table. The usage of a LUT is often preferred since it requires low calculation time and minimal hardware resources, since all data is stored in memory, and the only logic necessary is to map the input value to an address in the table.

However, the size of the memory becomes a problem as the word length of the data increases. The number of entries in a LUT can be determined using Equation 5.5, where N corresponds to the input data's word length. Then, the total size of the memory will be equal to the number of entries in the LUT multiplied by the desired word length of the output.

$$\text{Table entries} = 2^{(N * \#Inputs)} ; \quad (5.5)$$

This is a low complexity method with the major trade-off that it may require a significant number of memory elements for its implementation, accordingly to the word size selected.

5.3.2.2 Coordinate rotation digital computer (CORDIC)

The other approach that could be used for the calculation of the Atan2 function is using a CORDIC. The CORDIC is a computing technique introduced by [Volder \(1959\)](#), and it is used widely in hardware designs to calculate different mathematical functions such as trigonometric, hyperbolic, linear and logarithmic functions ([Andraka \(1998\)](#)). The CORDIC calculation consists of an iterative process that uses a series of rotations to find the angle of the input vector. The CORDIC calculation is done in vectoring mode, where the vector coordinates are used to calculate its magnitude and angle.

This is a method with low complexity with the only trade-off of being an iterative process that could add extra delay in the calculations.

5.3.2.3 Comparison

Between the two methods discussed, LUT usage seems to be more promising due to its low hardware resources usage, requiring mostly memory that is broadly available in the system. However, it is necessary to see if the memory in the FPGA will be enough for such implementation.

In Figure 5.5, the number of entries for a LUT was estimated for different word lengths using Equation 5.5. This figure shows that the total number of entries in the LUT grows exponentially as the word length increases. For this particular reason, the usage of LUT is not recommended when using a large word length (Torres et al. (2017)). In this system, the word length selected is 32 bits, which means that the LUT's total memory will be equal to 5.9030×10^{20} bits. This is significantly larger than the 14.625×10^6 bits available in the Stratix IV GX EP4SGX230K FPGA.

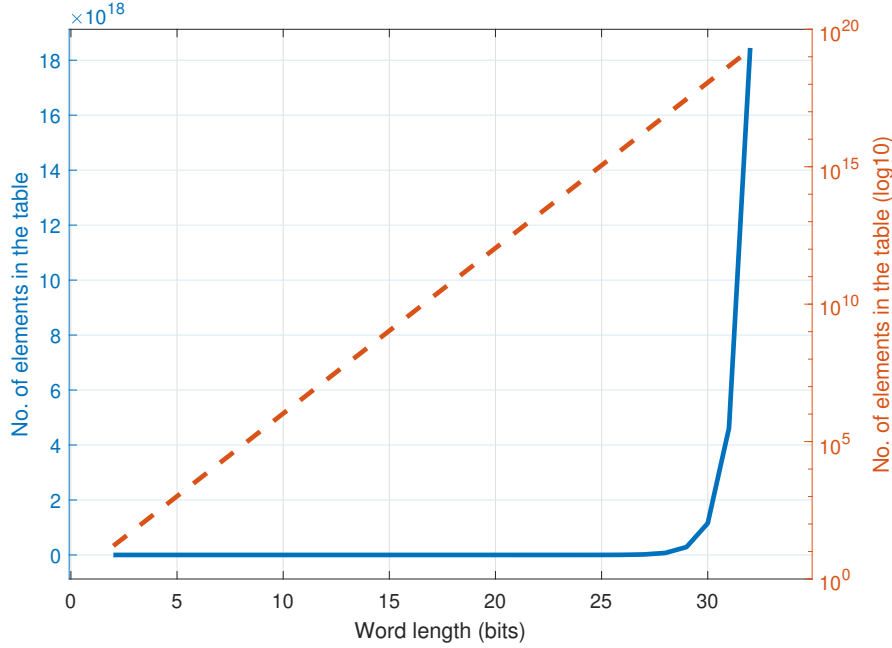


FIGURE 5.5: Estimation of the number of elements required for the lookup table based on the equation Equation 5.5.

However, the memory size could be reduced using different optimisations. For instance, it is possible to rotate the point into the first quadrant of the plane and calculate the angle for that new point. Then, the calculated angle is compensated by adding/subtracting $\pi/2$ or π due to the rotation of the point. Doing the calculation in this way reduces memory size by four since only the values for the first quadrant are necessary. Moreover, it is also possible to remove entries from the table and estimate the missing entries using an interpolation method, a similar approach to Lee et al. (2008). However, even using interpolation to reduce the number of entries in the table, it would be necessary to reduce the size of the table by a factor of 2^{23} .

Even after all these optimisations, the LUT still requires a significant amount of memory for the word length used in the system, and therefore its implementation is not possible for this work. Therefore, it was decided to use CORDIC to perform the calculation instead. The only disadvantage of this method is the delay due to its iterative processing, but this can be overcome using parallel CORDICs in the calculations (discussed in the next section).

5.3.3 PLI calculation

The PLI calculation can be obtained as in Equation 2.12 once the instantaneous phase has been calculated. The only optimisation done for the PLI calculation was the “ $1/K$ ” division after the sign values have been accumulated. The whole division can be avoided if, instead of dividing the accumulated value by 128 (window size), the accumulated value is considered an unsigned 8-bit fixed-point number with a factor of $1/128$ (or $1/2^7$).

5.4 Hardware Implementation

The proposed architecture for the complete system is shown in Figure 5.6, and it consists of three main modules: Analytical Signal, Phase Calculation, and PLI calculation.

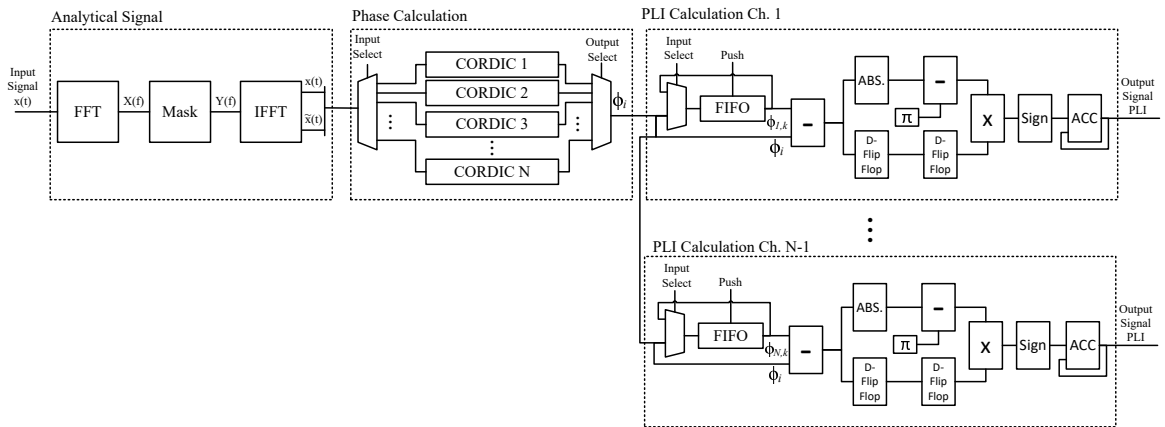


FIGURE 5.6: Block diagram of the hardware architecture proposed for the PLI calculation. Modified from Nuno and Maharatna (2019).

5.4.1 The analytic signal

This module generates the analytical signal used to calculate the instantaneous amplitude and phase as in Equation 5.1. As previously discussed, there are different approaches to calculate the analytical signal, but it was decided to calculate the analytical signal using the FT instead of a Hilbert transformer filter.

The process followed for the calculation is then first to perform the FT of the signal, then apply the mask operation defined in Equation 5.4, and finally perform the Inverse FT. The FT was done using the Fast Fourier Transform (FFT) algorithm, using the Intellectual Property (IP) core offered by Intel®. Although different implementations of the FFT are available in the literature, the advantage of using an IP core from the FPGA vendor is that the design has been optimised for the FPGA used,

which could lead to a smaller design due to the better usage of the resources. Moreover, the IP core has been thoroughly tested, which guarantees a reliable system and reduces the developing time required for the design and testing. The mask operation from Equation 5.4 can be done by doing a bit shifting to the left (equivalent to a multiplication by 2) of the values where the " $f > 0$ ", passing through the value where " $f = 0$ " and make the rest of the values equal to zero. This can be implemented with simple logic and will not require a high number of hardware resources. Finally, the IFT was performed using the same IP core from Intel[®] which also can perform the Inverse Fast Fourier Transform (IFFT).

5.4.2 The phase calculation

The calculation of the instantaneous phase of the signal was done using the CORDIC algorithm to calculate the arctangent function, based on analysis done in 5.3.2. The main disadvantage of the CORDIC is the iterative process involved in the calculation. The CORDIC requires a fixed number of cycles to complete the calculation, creating a bottleneck in the processing when new data is available. For this reason, it was decided to use an array of CORDIC modules. This array parallelise the whole process, allowing a continuous input and output instead of waiting for the CORDIC to finish before starting a new calculation. Also, this array is essential since the latency introduced by the CORDIC will be introduced for every sample in the signal, resulting in a significant cumulative delay for the whole process. The number of CORDIC used in this array will be equal to the number of iterations needed to perform the calculation. This number of iterations depends on the accuracy required for the system. The analysis from [Kota and Cavallaro \(1993\)](#) suggests that it would be necessary to perform N iterations for an N -bit wide number. However, we found that performing a total of 15 iterations provides a good resolution with a maximum absolute error of 2^{-13} . In addition to the 15 iterations, an initial extra iteration was used in order to extend the CORDIC range of operation to $\pm\pi$, as suggested by [Meher et al. \(2009\)](#).

5.4.3 The PLI Calculation

Once the CORDIC calculate the phase, the output is sent to the PLI calculation module so the calculation can be started. At the input of this module, there is a First In, First Out (FIFO) memory with a size equal to the window size. The function of FIFO memory is to store the phase values of every channel since these values are reused to calculate the PLI for every channel combination.

First, when the phase calculation of the first channel is done, the new phase values are stored in FIFO memory. After every phase values of the window have been stored in

the FIFO, the first phase value of the window from the first channel will be available at the output of the FIFO, and at the same time, the new phase value for the following channel will be ready at the input of the module. It is then when the calculation of the PLI between these two channels can start. The output of the FIFO is sent back to the input so the values can be reused and this process can be repeated when the phase values of the next channel are available.

To calculate the PLI, we subtract the value of π to the absolute value of the phase difference, followed by a multiplication between this subtraction and the initial phase difference. These steps were used by Niso et al. (2013), and they are necessary since the phase difference is in the range of $[-2\pi, 2\pi]$ and Equation 2.12 requires this difference to be in the range of $[-\pi, \pi]$. This procedure ensures that the phase difference is in the correct range and the sign is calculated correctly. After this adjustment to the phase difference, the sign of the value is introduced to an accumulator. This accumulator adds the values of the sign of the phase differences across the whole window. Finally, when the 128 values from the window have been added, the output of the accumulator will be 8 bits values with a fixed point of 7 bits. In this way, we avoid performing the division of the accumulated value and the total number of samples.

Contrary to the other modules reused in the system, this module needs to be instantiated multiple times to perform a continuous calculation for all the channels. The number of PLI Calculation modules needed will be equal to the number of channels in the system minus one.

5.4.4 PLI module integration

Once all the modules were designed individually, they were combined to create the final architecture presented in Figure 5.6. Every module here were designed as synchronous modules, where the communication between modules is done using an "Input and Output Valid" signal at every module input and output, respectively. In this way, every module sends an "Output Valid" signal to the next module in the chain, which is received as an "Input valid" signal that indicates that new data is available and the process can start.

First, the signal is introduced to the system one channel at a time, in windows of 128 samples. The submodules for the FFT and the IFFT are able to process the signal in a continuous burst, so one window can be introduced to the system after the other without any interruption. The analytical signal module will start the process automatically every time a new window is received. When the processing finishes, the analytical module output a new point every clock cycle.

Then, in the Phase calculation module, the analytical signal from the previous stage is received one by one and process them as soon as they are received. When a new point is received, the CORDIC available starts the calculation of the phase. Due to the sequential nature of the CORDIC calculation, every CORDIC takes 16 cycles. Since the analytical signal outputs new data every cycle after its own process is finished, the new point calculation is done by the next available CORDIC, avoiding a bottleneck with the analytical signal module and allow a continuous processing.

Finally, every PLI calculation module stores the phase values of the corresponding channel into the FIFO memory and the process described in subsection 5.4.3 is done. Then, every PLI calculation module will produce the PLI value of the corresponding channel and the rest of the channels, creating the upper diagonal of the PLI matrix as in Figure 5.7.

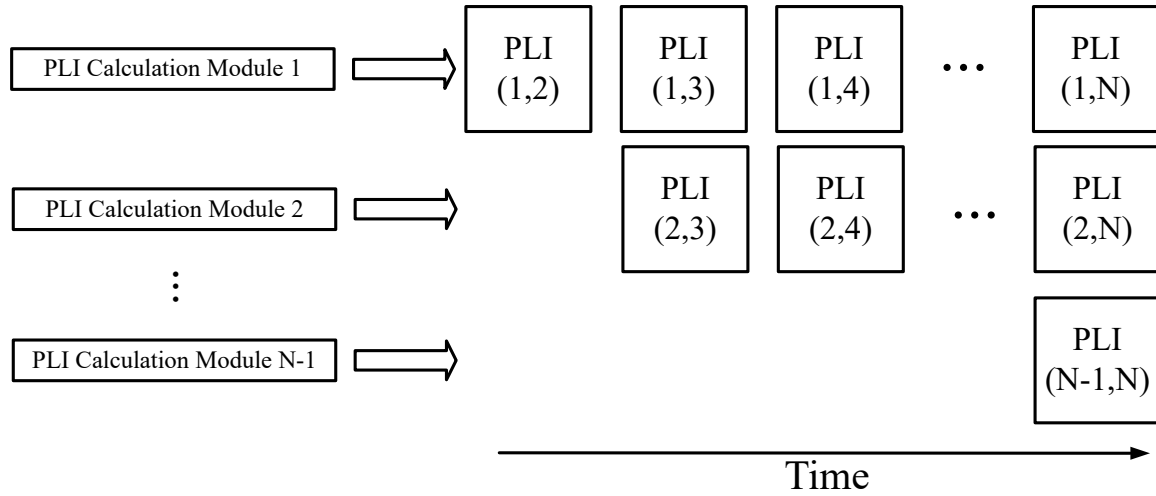


FIGURE 5.7: Block diagram of the output from the PLI calculation modules. The output from these modules is used to create the final PLI matrix.

5.5 Results

5.5.1 Validation

To validate the results, a new MATLAB script was created to calculate the PLI and compare the output of this with the hardware output. The script is based on the libraries of The Neurophysiological Biomarker Toolbox (NBT) (<http://www.nbtwiki.net/>) [Hardstone et al. \(2012\)](#), and the HERMES toolbox (<http://hermes.ctb.upm.es/>) [Niso et al. \(2013\)](#) and the main modification was the calculation of the PLI in windows instead of using the complete signal.

Afterwards, the complete design was tested using ModelSim software for simulations. The input signal of the system is 32 bits (15 fractional bits) and 16 channel EEG data,

with a sampling frequency of 256 samples per second from four different patients (same data used in Section 5.3). The EEG is published online by [Shoeb \(2009\)](#).

The EEG signal is introduced to the system, one channel at a time, in windows of 128 samples (half a second). The PLI calculation was done for every one of these windows. In total, 50 windows per patient were used for the testing of the system. The results obtained from the simulation were compared with the MATLAB script that performs the calculation of the PLI. Figure 5.8 presents the RMSE between the simulations and the MATLAB results, showing that the maximum RMSE achieved was 2.39×10^{-3} for patient one at the first window of the test. Furthermore, it is possible to see the error change between patients, but the average of the RMSE is in the range of 0 to 1.46×10^{-3} .

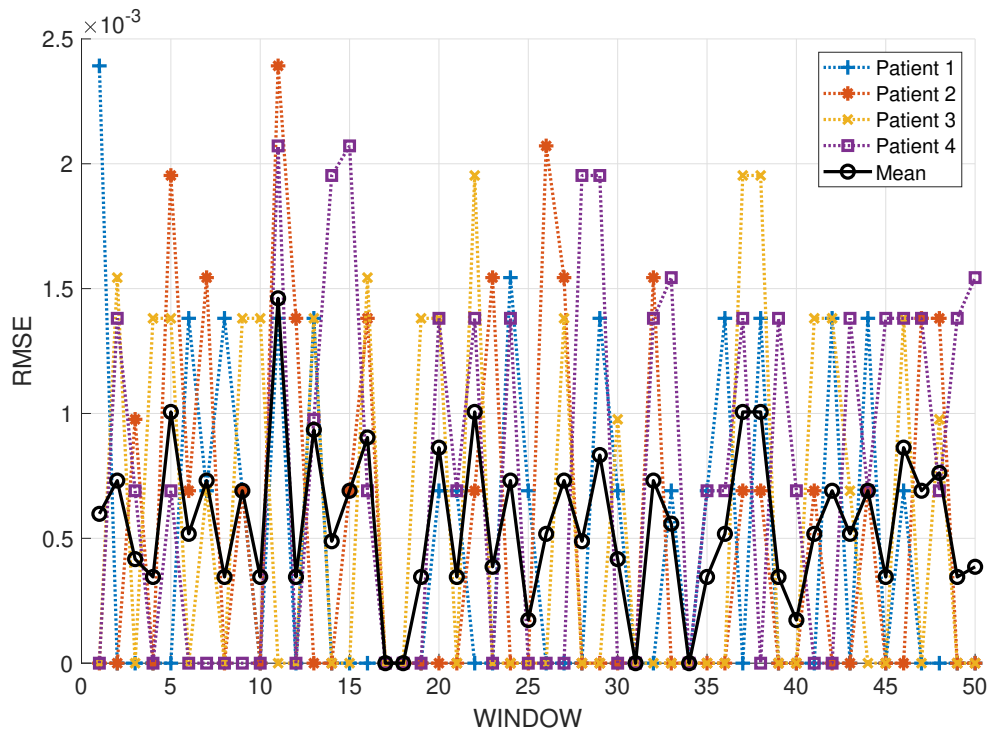


FIGURE 5.8: RMSE comparison between the four patients and 50 windows.

All the modules were synthesised for the Stratix IV GX EP4SGX230K FPGA as described in Chapter 3. A summary of the compilation reports obtained for every module is presented in Table 5.1. This table shows the maximum operating frequency and the usage of the resources in the FPGA in terms of logic elements (combinational Adaptive LUTs (ALUTs), memory ALUTs and dedicated registers), memory blocks and Digital Signal Processing (DSP) blocks. From all the modules, the module that calculates the analytical signal is the module with the larger number of elements used. Nonetheless, this module only needs to be instantiated once independently of the number of channels. On the other hand, some modules need to be instantiated more than once, such as the CORDIC in the Angle Calculation module (depending on the

number of iterations) and the PLI Calculation module (depending on the number of channels).

TABLE 5.1: Compilation results summary.

	Comb. ALUT	Mem. ALUT	Reg.	Mem. blocks (bits)	DSP blocks	Freq. (MHz)
Analytical Signal	8392	20	16139	36888	192	188.61
CORDIC	371	0	102	0	0	268.6
FIFO	1	0	4096	0	0	1066.1
PLI	184	0	223	0	0	541.13
Ctrl. Logic	13	0	8	0	0	677.51
16 Ch. System	16964	20	82642	36888	192	188.32

Moreover, a 16 channels system was synthesised, and its resource usage is also reported in Table 5.1. The logic elements used here correspond to 51% of the total available on the board, while the memory and the DSP blocks represent a total of <1% and 15% respectively, fitting under the demonstrating the feasibility of implementing the design on an FPGA. The maximum frequency achieved by the system is 188.32 MHz, which is close to the speed of the Analytical Signal module. This suggests that this module may limit the speed of the system. These results show that the device can be implemented satisfactorily with the available resources in the FPGA, particularly the lowly available DSP blocks.

5.5.2 Comparison

The only similar implementation to this design is the one presented by [Pal et al. \(2017\)](#), but this architecture does not perform the phase calculation, and thus the comparison will not be proper. Therefore, it was decided to compare the time required to perform the calculation of the PLI in MATLAB and the proposed architecture. The MATLAB script was executed on a computer with an Intel® i7-6700 processor and 32 GB of RAM. The time required for the processing was calculated for a different number of windows and presented in Table 5.2, starting from one window (0.5 seconds) followed by two windows (1 second), 120 windows (1 minute), 1200 windows (10 minutes) and 7200 windows (1 hour). The results shown that the maximum acceleration achieved was about 124 times for one window, and the minimum achieved was about 66 times for 120, 1200 and 7200 windows. Hence, it is possible to use this architecture to accelerate the PLI calculation and achieve real-time processing of the signal, even for large window sizes.

TABLE 5.2: Acceleration between different number of windows.

Window(s)	Time (s)		
	Software (average)	Hardware	Acceleration
1	1.78×10^{-3}	14.30×10^{-6}	124
2	2.82×10^{-3}	28.61×10^{-6}	98.5
120	117.50×10^{-3}	1717.02×10^{-6}	68.4
1200	1139.35×10^{-3}	17170.21×10^{-6}	66.4
7200	6884.30×10^{-3}	103021.27×10^{-6}	66.8

5.6 Conclusion

In this work, a new hardware architecture for the PLI calculation was presented. Here, it was decided to synthesise a 16 channel system, although the system could be extended to an N number of channels accordingly to the number of electrodes used in the study. This system achieved a frequency of 188.32 MHz, requiring only 51% of the logic resources available, which leaves room for increasing the number of channels.

The design was tested using simulation and comparing the results with a MATLAB script used to obtain the PLI values. The RMSE between MATLAB and the hardware simulations was between 0 and 1.46×10^{-3} on average for all the different patients. Furthermore, the simulations demonstrated that with this design, it is possible to perform the PLI calculation in a much shorter time than using a software-based solution, achieving an acceleration of 66 times minimum for a number of windows ≥ 120 and a maximum of 124 times for a single window. The processing time for the calculation is significantly lower than the time established in the requirements, and thus it is possible to reduce the operating frequency to reduce the power consumption. However, in this design, it was decided to keep the maximum speed to keep the processing time at the minimum and provide some leeway to the other modules involved in the signal processing.

All the improvements done in the design helped reduce the algorithm's complexity and calculation time, which is essential to enable the usage of these connectivity measurements in real-time applications compared to the common offline approach such as in neurofeedback systems.

Chapter 6

Hardware architecture for real-time EEG-based functional brain connectivity parameter extraction

6.1 Introduction

Electroencephalogram (EEG) is a non-invasive modality for monitoring brain activities and has been widely applied in clinical neuroscience. Recently, using EEG in conjunction with techniques such as functional brain connectivity (FC) and graph-theoretic characterisation of these connectivity networks showed promising results in the detection/diagnosis of different cognitive disorders such as epilepsy (Douw et al. (2010); Sargolzaei et al. (2015)), Alzheimer's (Stam et al. (2003); Yu et al. (2016)), Parkinson's (Stoffers et al. (2008); Utianski et al. (2016)), and autism (Jamal et al. (2014); Ahmadi and Adeli (2017)). The characterisation of the functional connectivity networks using graph-theoretic parameters could be also beneficial in neurofeedback systems that require continuous monitoring of the brain to adjust its response accordingly. The constant analysis will provide information regarding the patient's brain activity in real-time and will aid to identify any atypical behaviour in the brain signal, allowing the system to respond accordingly. A neurofeedback system with these characteristics opens the possibilities of using brain connectivity as a clinical tool in treating cognitive disorders, monitoring brain activity in real-time and providing the necessary treatment at the precise moment that is needed.

To create such a neurofeedback system, it would be necessary to perform the complete functional brain connectivity analysis and its graph-theoretic characterisation in real-time. However, similarly to other methods, the EEG signal processing to obtain the functional connectivity and its characterisation using graph-theoretic parameters is usually performed offline due to the computational complexity involved in the

process. Therefore, to incorporate this analysis, it is necessary to design a device that can complete the overall process in real-time while meeting the requirements previously discussed in Chapter 3. Such a self-sustained and highly computation-demanding wearable system such as the one we envision is best realised using dedicated Application Specific Integrated Circuit (ASIC) design.

To achieve this, in this work we propose a novel system hardware architecture for a 19-channel EEG system that has two main modules, namely, computation of Phase Lag Index (PLI) matrix resulting into the formulation of FC and the extraction of graph-theoretic parameters from these FCs. This architecture is based on the work presented in Chapter 5 and [Pal et al. \(2017\)](#). However, those two works only presented the two modules required for implementing the entire system, namely, the PLI computation and graph-theoretic parameters extraction, respectively. In this work, we present a complete and integrated solution to calculate the FC measurements from EEG signals in real-time. To our knowledge, this is the very first attempt to implement such a system in real-time hardware.

The main novelty lies in the proposed approximations in the mathematical definitions of the graph-theoretic parameters, making them "lightweight", i.e., reducing the numbers of computationally intensive arithmetic operations such as divisions, multiplications, and square/cubic rooting that are fundamental in the conventional definitions of these parameters. This optimal algorithm-to-architecture mapping methodology made the overall design amenable for hardware implementation and allowed the processing of the brain connectivity networks and its graph-theoretic characterisation in real-time as required in neurofeedback systems. Also, we showed that such approximations have minimal impact on the final performance of the system by comparing the hardware outputs with the traditional software implementation of these parameters ([Hardstone et al. \(2012\)](#); [Rubinov and Sporns \(2010\)](#)).

Furthermore, this design can act as an emulation alternative in real-time to the current MATLAB simulation toolboxes (NBT ([Hardstone et al. \(2012\)](#)), BCT ([Rubinov and Sporns \(2010\)](#)) and HERMES ([Niso et al. \(2013\)](#))) which performs the calculation in an offline manner, enabling faster exploration of different cognitive phenomena.

6.2 Background

6.2.1 Network integration parameters

The Network integration parameters are used to describe the capacity of the brain to integrate different regions of the brain involved in the processing of information. These parameters use the concept of paths to describe how the information travels between the different brain regions and how efficient these networks are. Examples of

these parameters are the degree, density, characteristic path length, eccentricity, radius and diameter.

6.2.1.1 Degree and Weighted Degree

The node degree is a measurement used to represent the interconnection of one node with the other nodes in the network. This provides information about how closely interconnected the nodes in the network are. The node degree can be estimated using Equation 6.1, counting the number of edges (a_{ij}) greater than a certain threshold (i.e. zero) for a particular node i . Furthermore, a weighted degree (k_i^w) could also be calculated but in this case, the weights (w) of the edges are accumulated for a particular node. The weighted degree not only establishes that there is a connection between two nodes but also provides information about how strong this connection is.

$$k_i = \sum_{j=1}^N a_{ij}; k_i^w = \sum_{j=1}^N w_{ij} \quad (6.1)$$

6.2.1.2 Density

The density can be obtained using Equation 6.2, which counts the number of edges (w) greater than a threshold in the upper triangular elements of the matrix and divided by the total possible number of connections in the network. This ratio provides information on how dense the connection between the different brain regions is.

$$D = 2 \sum_{j,i \in N, j > i} upper(w_{ij} > threshold) / (N^2 - N) \quad (6.2)$$

6.2.1.3 Characteristic Path Length

The Characteristic Path Length (CPL) is a measurement of integration that describes the interconnection among the different nodes and how efficient the communication between them is. The CPL is defined in Equation 6.3. The shortest distance (d_{ij}) between two nodes can be obtained using Equation 6.4, where $f(w)$ is a function to convert the edge values into distance values. A commonly used function is the inverse value of the weight.

$$CPL = \sum_{i \in N} \frac{\sum_{j \in N, j \neq i} d_{ij}}{N(N-1)} \quad (6.3)$$

$$d_{ij} = \text{ShortestPath}_{ij}(f(w)) \quad (6.4)$$

Once the edge values are converted into distances, the shortest distance between the two nodes is found. The most common algorithm used to find the shortest distance between two nodes is Dijkstra's algorithm (DA) presented in [Dijkstra et al. \(1959\)](#). The steps to perform the DA are the following:

1. Create a set of unvisited nodes, which includes all nodes except node i .
2. Set the tentative distance d between the node i and every other node j to the edge length between them, i.e. $d_{i,j} = l_{i,j}$. Throughout the algorithm, $d_{i,j}$ will hold the shortest path length from i to j through visited nodes. Currently, only node i is visited, so the initial value of $d_{i,j}$ is $l_{i,j}$.
3. In the set of unvisited nodes, find the node k with the shortest tentative distance.
4. For every node j , update the tentative distance as follows:
 $d_{i,j}(\text{new}) = \text{minimum}(d_{i,j}, d_{i,k} + l_{k,j})$. The existing shortest path from i to j through visited nodes has a path length of $d_{i,j}$. If a path through node k (which has length $d_{i,k} + l_{k,j}$) is shorter, this will become the new shortest path between the nodes.
5. Remove node k from the set of unvisited nodes. Repeat steps 3-5 until the set of unvisited nodes is empty.
6. After visiting all nodes, $d_{i,j}$ now holds the shortest path length from i to j through visited nodes.

6.2.1.4 Eccentricity, Radius and Diameter

The eccentricity is the maximum distance between two nodes in the network. In addition to this measurement, there is also the radius and diameter of a network, which corresponds to the minimum and maximum distance respectively for the whole graph. The distance between the nodes is calculated using Equation 6.4 as in the characteristic path length.

6.2.2 Network segregation parameters

Between the different brain regions, it is possible to identify specific regions with a more robust interaction among them, forming groups of brain regions strongly connected between them. These groups are known as clusters, and their existence in the network suggests segregated processing of the information. Network segregation

parameters are used to quantify the segregation of the network due to the presence of these clusters. Examples of network segregation parameters are the clustering coefficient and the transitivity.

6.2.2.1 Clustering Coefficient

The clustering coefficient is used to measure the cliquishness between a group of nodes in the network, which in turn determines how clustered the network is (Watts and Strogatz (1998)). The calculation of the clustering coefficient is done as in Equation 6.5, where k_i represents the degree of the node and t_i the geometric mean of the node, which is calculated using the weights (w) for all the possible triangles around a specific node i .

$$t_i = \frac{1}{2} \sum_{j,h \in N} \sqrt[3]{w_{ij}w_{jh}w_{ih}}; CC_i = \sum_{i \in N} \frac{2t_i}{k_i(k_i - 1)} \quad (6.5)$$

Also, it is possible to obtain the mean clustering coefficient as in Equation 6.6, which gives a more general idea of how clustered the whole network is.

$$CC_{mean} = \frac{1}{N} \sum_{i \in N} CC_i \quad (6.6)$$

6.2.2.2 Transitivity

Transitivity is a variation of the mean clustering coefficient measurement used to understand the segregation of the whole network. The transitivity is calculated using Equation 6.7, similar to the clustering coefficient but in this case, the values of the numerator and denominator of every node are accumulated before performing the division. This normalisation helps the transitivity be less susceptible to nodes with a lower degree (Rubinov and Sporns (2010)) and therefore its advantage over the mean clustering coefficient.

$$T = \frac{\sum_{i \in N} 2t_i}{\sum_{i \in N} k_i(k_i - 1)} \quad (6.7)$$

6.3 Algorithmic optimisation of the computational burden of the parameters

As described in the previous section, each of the graph-theoretic parameters requires complex arithmetic computations such as computing cubic root, divisions and

multiplications. Typically, such operations require significant arithmetic resources and therefore have a direct impact on the energy consumption of the architecture as energy consumption is directly proportional to the arithmetic complexity. Therefore, to make the system efficient, one may need to approximate the mathematical definitions of graph-theoretic parameters so that they can be amenably mapped to a hardware platform while reducing the overall computational complexity - ideally avoiding dividers and cubic root computations and using a minimal number of multiplications - without sacrificing accuracy. This section describes the approximations in the definitions of the graph-theoretic parameters done in this work for an efficient algorithm to architectural mapping leading to optimisation of arithmetic resources.

From the Equations 6.5 and 6.7 in Section 6.2 it can be seen that transitivity and clustering coefficient all require computation of the geometric mean of triangles in the matrix and the degree. Therefore, we have decided to consider the clustering coefficient computation as a core computational unit that, apart from computing clustering coefficients, also outputs the geometric mean of all the triangles t_i and the degree. These values are then shared with the transitivity and density to perform the respective calculations shown in Figure 6.1. In this way, the values are reused and redundant hardware in the system is eliminated.

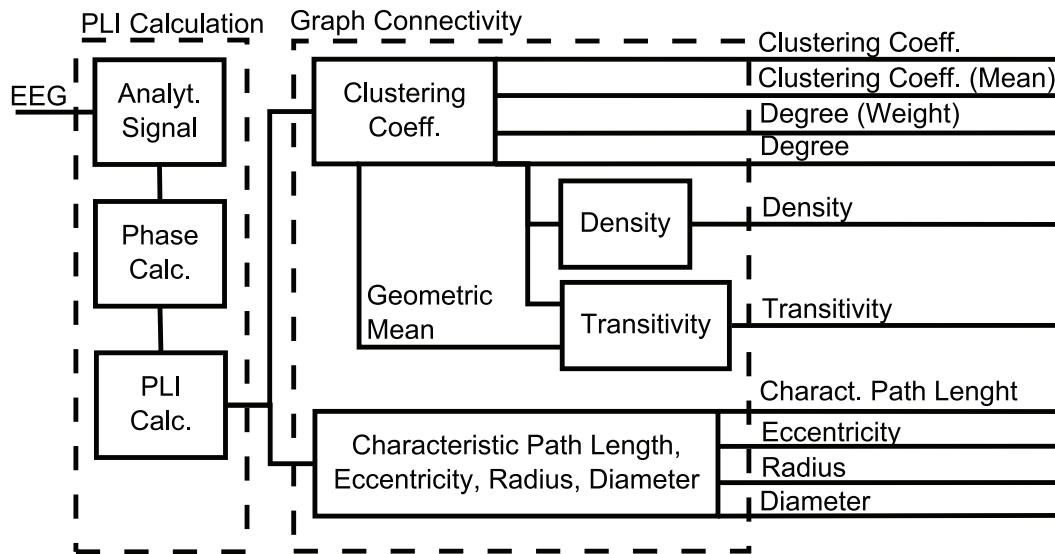


FIGURE 6.1: Block diagram of the general architecture for the PLI and the graph-theoretic measurements calculation. Extracted from Nuno et al. (2021).

6.3.1 Clustering Coefficient Optimisation

The most computationally expensive operation needed in the clustering coefficient calculation is the computation of the geometric mean of all the triangles t_i . This is an iterative process that involves the multiplication of the weights for every possible triangle permutation as in Equation 6.5, followed by cubic root evaluation. As can be

seen in Pineiro et al. (2008); Bruguera et al. (2011), such computation needs significant arithmetic resources. To avoid that, we opt to take a different approach by completely avoiding evaluation of cubic root and simply using the multiplication between the weights as the numerator in Equation 6.8, as in essence the cubic root will only contribute in terms of a scaling factor and not in terms of the actual nature of the t_i and hence for the value of clustering coefficients.

$$t_i = \frac{1}{2} \sum_{j,h \in N} w_{ij}w_{jh}w_{ih}; CC_i = \sum_{i \in N} \frac{2t_i}{k_i(k_i - 1)} \quad (6.8)$$

To verify the impact of such modification in the definition of t_i , we carried out a correlation analysis of the clustering coefficient calculated using Equation 6.5 and Equation 6.8, withdrawing five-minute real EEG signals from 10 different subjects using data obtained from Shoeb (2009). The data was recorded by 23-channels at 256 Hz with 16-bit resolution. The results are shown in Table 6.1. The average correlation between these signals is 0.923, which indicates that even though the signal scale is different, the morphology of the signals remains very close to the original and therefore could be used in the characterisation of FC matrices.

TABLE 6.1: Correlation between the clustering coefficient and the transitivity obtained with and without the cubic root.

Subject	Correlation	
	Clustering Coefficient	Transitivity
1	0.898	0.928
2	0.920	0.948
3	0.918	0.936
4	0.905	0.923
5	0.942	0.959
6	0.937	0.960
7	0.914	0.932
8	0.901	0.915
9	0.943	0.948
10	0.953	0.957
Mean	0.923	0.941

To reduce the complexity even further, we utilise the symmetry properties of the PLI matrix. This leads to the computation involving only the elements of the upper triangle of the PLI matrix in the geometric mean calculation of t_i , resulting in a reduction of computational complexity by half. This optimised geometric mean of t_i was then used to calculate clustering coefficients as well as the degree k_i (which was also used in the calculation of the clustering coefficient).

6.3.2 Transitivity

Since the transitivity calculation is similar to the clustering coefficient, we reuse the geometric mean and the degree for deriving transitivity. To validate that the modifications done to the clustering coefficient do not affect the calculations of the transitivity, the same correlation analysis has been done and its results are shown in Table 6.1. The average correlation is 0.9405, which demonstrates that not using the cubic root on the triangle calculation does not have a substantial impact on the transitivity either.

6.3.3 Density

Exploiting the symmetric properties of PLI, the summation of the binary degree k_i is equal to two times the summation of the non-zero values in the upper triangular elements of the matrix. Thus, it is possible to calculate the density as in Equation 6.9 with the already calculated degree, reducing the arithmetic resources required.

$$D = \frac{\sum_{i \in N} k_i}{N(N-1)} \quad (6.9)$$

6.4 Hardware Architecture

The system is composed of two main core modules: the PLI calculation and the graph connectivity module as shown in Figure 6.1. The first one calculates the PLI matrix from an EEG signal, while the graph connectivity module uses that matrix to extract the graph-theoretic parameters discussed in Section 6.2.

6.4.1 PLI Architecture

The architecture used for the PLI calculation is the one presented in Chapter 5, composed of three main modules: Analytical signal, Phase calculation and PLI calculation modules. However, this architecture was extended to 19-channels in this work instead of 16-channels as it was previously presented. Due to its modular design, the only modification required for this increase in the number of channels was the inclusion of three more PLI calculations modules to the design.

6.4.2 Clustering Coefficient

As explained in section 6.3, the clustering coefficient module is considered as a core module that calculates not only the clustering coefficient but also the geometric mean

of the triangles and the degree (binary and weighted). The block diagram of this module is presented in Figure 6.2, consisting of three basic processing blocks: geometric mean, degree and weighted degree. The geometric mean t_i was calculated first as in Equation 6.8, multiplying and accumulating the weights of all the possible permutations for every node i in the matrix. Simultaneously, the degree (binary and weighted) is calculated in the respective blocks (explained in a later subsection). Once the geometric mean of all the permutations in node i have been done, the values are divided by $k_i(k_i - 1)$. The same process is repeated for every node and the results are accumulated until the final clustering coefficient value is obtained.

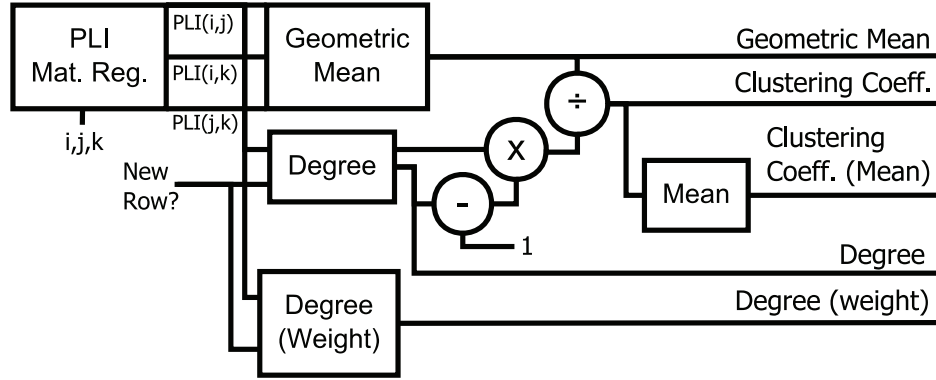


FIGURE 6.2: Block diagram of the hardware architecture proposed for the clustering coefficient. Extracted from Nuno et al. (2021).

6.4.3 Degree and Weighted degree

The degree and the weighted degree are calculated as in Equation 6.1, accumulating the number of edges greater than zero and the weights of the edges respectively for every node in the matrix. Both degrees are calculated in the clustering coefficient module. By doing all triangles permutations for every node, we guarantee that all the edges of this node will be used at least once, therefore it is possible to reuse these weight values in the calculation of the degree. Doing this is not necessary to include an extra copy of the PLI values and the extra logic to perform the degree calculation in a separate process. The only disadvantage of this is that the degree calculation would take longer than it would have done using a separate process, but this does not have a critical impact on the performance since the degree result is still available before the other calculations, e.g. the clustering coefficient.

6.4.4 Density

As mentioned before, the density module reuses the previously computed values of the degree in the clustering coefficient module and thus consists only of an accumulator and divider, as shown in Figure 6.3. The density is calculated as in

Equation 6.9, accumulating the degree values of every node followed by a division by the total number of elements outside the main diagonal. Since the total number of elements is constant, we precompute the inverse of it and multiply this inverted value with the output of the accumulator to perform the division operation. This approach reduces the total arithmetic complexity as opposed to using an arithmetically complex full divider.

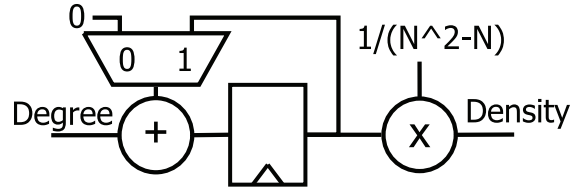


FIGURE 6.3: Block diagram of the hardware architecture proposed for the Density. Extracted from Nuno et al. (2021).

6.4.5 Transitivity

The transitivity calculation is almost identical to the clustering coefficient with the exception that the numerator and denominator are accumulated before performing the division. Thus, the triangle and the degree values from the clustering coefficient calculation are reused here and the only extra hardware needed is an accumulator for the numerator, a multiplier and accumulator for the denominator and the final divider to perform the calculation. The architecture for the transitivity is shown in Figure 6.4.

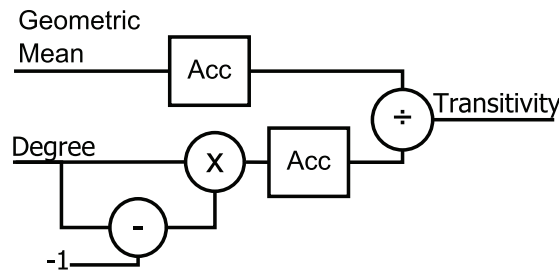


FIGURE 6.4: Block diagram of the hardware architecture proposed for the Transitivity. Extracted from Nuno et al. (2021).

6.4.6 Characteristic Path Length

To perform the CPL calculation, it is necessary to map the weights of every edge in the matrix to an equivalent distance. Here, we decided to perform the mapping as in Equation 6.10. With this mapping, the weight values closer to one which represents a strong connection on the brain connectivity matrix will have a small distance value. On the other hand, values close to zero will have a greater distance value representing a weak connection between the nodes.

$$f_{ij}(w_{ij}) = 1 - w_{ij} \quad (6.10)$$

Once the mapping is done, the shortest path is found using Dijkstras algorithm (DA) discussed in Section 6.2. The Shortest Path Finding Unit (SPFU) in Figure 6.5 (a) performs the distance mapping shown in Equation 6.10 and compares the actual minimum distance from the node against the particular node distance plus the current node distance, as it is done in step 4 in Section 6.2. The Minimum finding unit finds the node with the minimum distance that will be used as the new node (step 5) and the current node distance is updated with this node distance. The process is repeated until the distance from the initially selected node and the rest of the nodes is found. Then, the whole process is repeated with a different initial node and the minimum distance from that node with respect of the other is found. When all the minimum distances from all the nodes with respect to the other nodes is completed, these values are accumulated and later divided by ' $N^2 - N$ ' as in Equation 6.3, obtaining the final CPL value. This division is done by multiplying the value with the precomputed inverse value of ' $N^2 - N$ '.

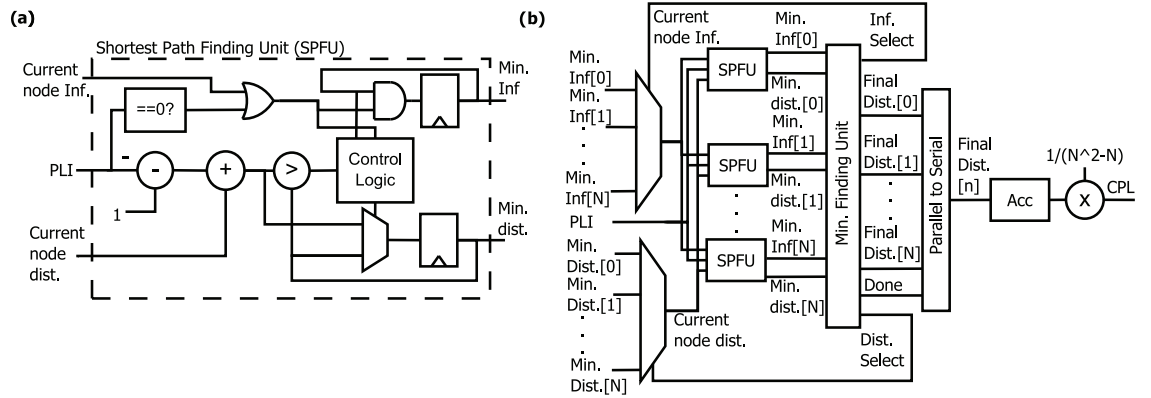


FIGURE 6.5: (a) Block diagram of the hardware architecture proposed for the Shortest Path Finding Unit (SPFU). (b) Block diagram of the hardware architecture proposed for the characteristic path length. Extracted from Nuno et al. (2021).

6.4.7 Eccentricity, Radius, and Diameter

To calculate these three measurements, it is necessary to calculate first the shortest distance between nodes. This is already done during the calculation of the CPL, then these values are stored and reused here. Then, the calculation is just as simple as finding the maximum values for every node (eccentricity) and finding the maximum and minimum of the whole network (diameter and radius respectively). This is done by checking node distance and using a comparator to determine if the new value becomes a maximum/minimum against the previous value examined.

6.5 Validation and Results

To verify the proper behavior of the complete system, all the modules were connected together, and a functional test was performed using simulations.

6.5.1 Validation data

The validation was done using a five-minute, 19-channel EEG signal recorded at 256 Hz from 10 different subjects. These records were obtained from the database of [Shoeb \(2009\)](#). These EEG signals are the same used in the parameter optimisation for Section 6.3.

6.5.2 Functional Validation

The proposed system was simulated using the ModelSim software and the output of this system was compared with the output obtained using the toolbox from [Rubinov and Sporns \(2010\)](#). The window size used in the PLI calculation was 128 samples. Since the EEG used was sampled at 256 Hz, it is possible to calculate two windows for every second of the EEG. Therefore, a five-minute EEG signal will generate a total of 600 PLI matrices for every subject. From every PLI matrix, the graph-theoretic parameters mentioned in Section 6.2 are calculated, followed by calculating the relative error between the hardware architecture and the MATLAB toolbox. The same process was repeated for every one of the 10 subjects.

6.5.2.1 Error calculation

The results of the relative error are shown in Figure 6.6 as a histogram for every parameter calculated across all the windows and all the subjects. Based on the distribution fit of the histogram fit in this figure, it can be observed that the centre of the curve is located close to zero, which indicates that for most of the values obtained the relative error is concentrated in this area. The degree is the one with the maximum relative error range that goes from -7.97% to 7.82% (mean= -0.076%, $3\sigma = 7.9\%$). Despite this wide range, most of the errors obtained were concentrated in a range of about -2% to 2%. Also, the RMSE value was calculated for all the parameters. The maximum RMSE found among all the parameters is 0.123 for the degree, a small value that also proves the correct behaviour of the system.

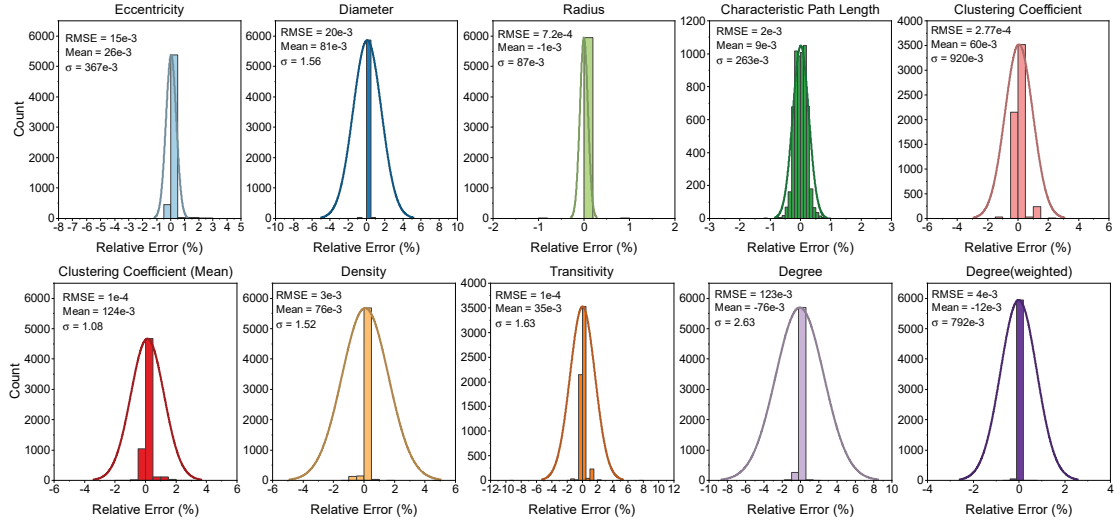


FIGURE 6.6: Histogram with the relative error results from the functional validation for every graph-theoretic parameter calculated for the 600 windows and the 10 subjects. The straight line shows the histogram fit of the relative error in order to appreciate better the distribution of the error. Inside every plot was added the mean RMSE, mean and standard deviation value for all parameters. Extracted from: [Nuno et al. \(2021\)](#).

6.5.2.2 Time calculation

The number of clock cycles required to perform the calculation of every connectivity measurement can be found in Table 6.2, where N corresponds to the number of Channels and the $\binom{N-1}{2}$ is a function to obtain all the possible combinations for $N - 1$ channels for two positions.

TABLE 6.2: Number of clock cycles required for the calculation of the different parameters.

	Clock Cycles
Degree	$N \binom{N-1}{2} + 1$
Degree (weighted)	$N \binom{N-1}{2} + 1$
Clustering Coefficient	$N \binom{N-1}{2} + 7$
Transitivity	$N \binom{N-1}{2} + 6$
Density	$N \binom{N-1}{2} + 2$
Characteristic Path Length	$N^2 + N + 1$
Eccentricity, Radius, Diameter	$N^2 + N + 1$

In Table 6.2, the measurement with the longer calculation time was the clustering coefficient. The reason for this is the extensive iterative processing done to calculate the geometric mean of all the possible triangles for a specific node. This triangle calculation could be performed in parallel if desired, instantiating as many modules as the number of nodes in the system. Nevertheless, without parallelisation, the design is still fast enough for real-time implementation. Thus an increase in the speed is not

necessary and will only increase the resources needed and the power consumption of the system. The total calculation time on MATLAB is 3.74ms, which is about 28 times higher than ours (131us). It is worth mentioning that the time required to perform the degree is almost as long as the Clustering coefficient because it is calculated during the same process. Indeed, the degree is a simple measurement that could be obtained faster than the actual time, but that implies creating another instance of the PLI matrix and extra logic to perform the calculation independently from the clustering coefficient. Since there is no benefit in calculating the degree faster, in this work it was opted to keep the degree calculation in conjunction with the clustering coefficient.

6.5.3 Hardware savings

In this subsection, we compare the hardware savings achieved with our approximations compared to a one-to-one implementation of the algorithm. This is shown in Table 6.3 in terms of 2-input NAND gates. In this table, we converted all the adders, multipliers, dividers, and comparators complexities in terms of 2-input NAND gates, providing an easier way to compare these elements since each of them may require a different number of logic gates for its implementation. It is important to mention that the optimised degree is considered zero on this table since it is implemented inside the Clustering coefficient module. After all the optimisations the total saving comes to around 25%, which translates to an equal reduction of area and power consumption of the final system compared to a one-to-one implementation.

TABLE 6.3: Comparison between the one-to-one implementation against the optimized proposed architecture in terms of 2-input NAND gates.

Module	NAND gates		Savings (%)
	Not Opt.	Opt.	
Degree	150	0	100
Degree (weigh.)	135	0	100
Density	2,002	1,296	35.26
Clust. coeff.	68,863	59,472	13.63
Trans.	69,013	43,081	37.57
Charact. path len.	15,548	11,676	24.90
Ecc., Rad. and Dia.	315	315	0
Total	156,026	115,840	25.75

6.5.4 Synthesis Results

After the functional test was performed and the system was performing as expected, the next step is to perform the synthesis of the system on an FPGA. The FPGA used

for this system is the Stratix IV GX EP4SGX230K. The synthesis results for every module are shown in Table 6.4.

TABLE 6.4: Synthesis results for the different modules of the system and the final design.

	Comb. ALUT	Reg.	Mem. Blocks (bits)	DSP Blocks	Freq. (MHz)
<i>PLI</i>					
PLI calculation system	22,398	95,406	40,408	192	215.05
<i>Graph-theoretic Parameters</i>					
Clust. coeff.	3,892	2,105	0	9	23.74
Trans.	1,873	86	0	0	564.33
Density	65	15	0	0	800
Charact. path length	4,517	2,002	0	4	61.39
Ecc., Rad. and Dia.	200	546	0	0	200.36
Graph-theoretic parameters system	11,835	6,060	0	17	22.16
<i>PLI + Graph-theoretic parameters</i>					
Final design	33,679	101,205	40,408	209	22.16

The module with the highest resource usage was the PLI calculation, using 61% of the available resources on the board. In comparison to this, the proposed architecture for the calculation of the graph-theoretic parameters requires 9% of the total resources available at the board. Nonetheless, the calculation of the graph-theoretic parameters is slower than the PLI calculation, achieving a maximum frequency of 22.16 MHz. This operation frequency is still high enough to perform the whole process in the order of microseconds, leaving some room for other processes that would require a longer computation time, such as artifact removal.

However, it is a well-known fact that although FPGA implementations provide design flexibility, they are inherently not optimised for area, speed and power compared to specialised ASIC implementation. To get a better understanding of hardware complexity, we performed an estimation of the logic gates required for the design by converting all the adders, multipliers, dividers, and comparators complexities in terms of 2-input NAND gates. We did not implement the FFT module here but implemented the full PLI and graph-theoretic parameter extraction circuits. The reason for not implementing FFT is that there are several optimized implementations of it already proposed over the decades and any could be adopted for the current purpose depending upon the area and power budget. Therefore, we selected the implementation presented in [Yu and Yen \(2014\)](#) due to its low area and power consumption for this estimation. The result is shown in Table 6.5 in terms of 2-input NAND gate equivalent.

TABLE 6.5: Architectural complexity estimation using NAND gates.

Module	No. of NAND
Analytical sig.	409,374
Phase Calc.	92,160
PLI Calc.	318,960
PLI Unit (19 Ch. system)	820,494
Degree	0
Degree (weighted)	0
Density	1,296
Clust. coeff.	59,472
Trans.	43,081
Charact. path length	11,676
Ecc., Rad. and Dia.	315
Graph-theoretic parameters system	115,840
PLI + Graph-theoretic parameters	936,334

6.5.5 Power Consumption

One important point for this design was to create a low power system. To demonstrate the low power capabilities of the design, a power estimation was obtained for an FPGA and an ASIC implementation.

For the FPGA estimation, the software provided by the manufacturer was used. This software guarantees a high level of accuracy when the software is provided with enough switching activity from the simulations. The switching activity was generated running the simulations previously mentioned in section 6.5.1. The EEG data is provided at the same sample rate at which it was acquired, and the frequency used in the system is 22.16 MHz which is the maximum frequency achieved. The dynamic power consumption of the system in this particular FPGA at a VCC voltage of 0.9V is 51.84mW. This total power consumption is important if this design is meant to be implemented as a portable continuous monitoring system for instance. Based on this power consumption and the current battery capacities in portable devices like mobile phones, with a capacity of up to 4200 mAh (Liang et al. (2019)), the proposed system could be energised continuously for about three days (almost 73 hours). It could be integrated with other systems for a further, more advanced processing of the signal as a stand-alone system. Despite this promising result, however, as mentioned earlier, it is a well-known fact that FPGAs are inherently not as efficient in terms of power consumption as an ASIC implementation where a design can be optimised from the point of view of power consumption. Therefore, we expect the system to show even better power performance when implemented in ASIC.

To estimate the power consumption on an ASIC, we used the Synopsis Design Compiler and a 90nm TSMC CMOS technology. In this estimation, we used the power reported in [Yu and Yen \(2014\)](#), an FFT module implemented in the same CMOS technology. Our results are presented in Table 6.6. Based on the estimations, the power consumption of the design presented (just processing) here will be around 39.37 mW, representing a power saving of around 24% from the 51.84 mW estimated for the FPGA implementation. This estimation was done without any other techniques such as clock/power gating. Furthermore, it is important to note that FPGA uses a CMOS technology of 40nm, less than half the size of the library used for the estimations. Thus, if a similar technology to the FPGA were to be used for the ASIC implementation, further power reduction can be expected.

TABLE 6.6: Power consumption estimation for every module in the system.

Module	Power (mW)
Analytical sig.	6.932
Phase Calc.	2.522
PLI Calc.	28.800
PLI Unit (19 Ch. system)	38.256
Degree	0
Degree (weighted)	0
Density	0.009
Clust. coeff.	0.787
Trans.	0.148
Charact. path length	0.021
Ecc., Rad. and Dia.	0.149
Graph-theoretic parameters system	1.116
PLI + Graph-theoretic parameters	39.372

6.5.6 Comparison

In literature, only the authors from [Pal et al. \(2017\)](#) have presented an architecture that performs a calculation of the graph connectivity measurements. The comparison between this and the proposed system is presented in Table 6.7.

There are two major differences between these two systems. The first one is the inclusion of the module to calculate the PLI calculation in the proposed system. In [Pal et al. \(2017\)](#), the authors did not perform the calculation of the instantaneous phase used for the calculation of the PLI because it was assumed that these phases were given as an input. In the proposed system, the input will be an EEG signal instead of phase values, and therefore includes the PLI calculation that was avoided in [Pal et al. \(2017\)](#). The second major difference is the way the graph-theoretic parameters are

TABLE 6.7: Comparison between the design presented in Pal et al. (2017) and the design presented here.

	Error		Adders		Mult. & Div.		Clock Cycles	
	Pal et al. (2017)	This work	Pal et al. (2017)	This work	Pal et al. (2017)	This work	Pal et al. (2017)	This work
Degree	0	$O(2^{-7})$	8	0	0	0	88	169
Degree (weighted)	0	$O(2^{-9})$	64	0	0	0	88	169
Density	$O(2^{-8})$	$O(2^{-9})$	102	1	0	1	142	170
Clustering Coefficient	0	$O(2^{-14})$	896	5	1,024	5	87	175
Transitivity	$O(2^{-10})$	$O(2^{-15})$	903	3	1,024	2	88	174
Characteristic path length	$O(2^{-9})$	$O(2^{-9})$	21	3	0	1	157	73
Eccentricity, Radius, Diameter	0	$O(2^{-9})$	37	0	0	0	168	73

calculated. The whole architecture was redesigned with the main goal to reduce the number of arithmetic operations in the calculation. This not only allows to reduce the size of the final system but also decrease the power consumption of the system. By reducing the number of arithmetic operations used, the total energy of the system is decreased as well. The reduction in resources used can be estimated by comparing the number of logic 2-input NAND gates that both architectures use. The number of gates reported in Pal et al. (2017) is equal to 1,116K, while in this work the estimate for the complete system came to around 937K (Table 6.5), about 16% less. Nevertheless, the architecture of Pal et al. (2017) does not include the 19 channel PLI calculation and it is only doing the graph-theoretic parameter calculation. Thus, if we compare only the graph-theoretic parameter calculation of this work against Pal et al. (2017), the 2-input NAND gates required for this architecture will be around 116K, representing a reduction of about 89% in the total 2-input NAND gates used. Additionally, even though it is not possible to compare the power between both architectures since the CMOS technology is different (130 nm) from ours (90 nm), it is expected that the reduction on logic elements previously discussed will have a similar impact on the total power consumption of the system. Even though there was a reduction in the number of arithmetic elements used for the operations, the proposed system is still able to perform the calculation at a similar time to that of the system presented in Pal et al. (2017). The difference in the calculation time between these two is seven clock cycles only. At the maximum frequency achieved, the total time required to calculate the graph-theoretic measurements in this design is equal to 7.89us compared to the 6.7us reported in Pal et al. (2017). It could be possible to use more Clustering coefficient modules (module with the longest computation time) to speed up the calculations and reach a similar time than Pal et al. (2017), but this will increase the total NAND gates used in the design. This trade-off is presented in Figure 6.7. Here, it can be seen that the area increases exponentially as the time is reduced. Therefore, this design opted to use only one Clustering coefficient and reduce to a minimum the number of logic elements and the energy consumption, even though the calculation time is longer.

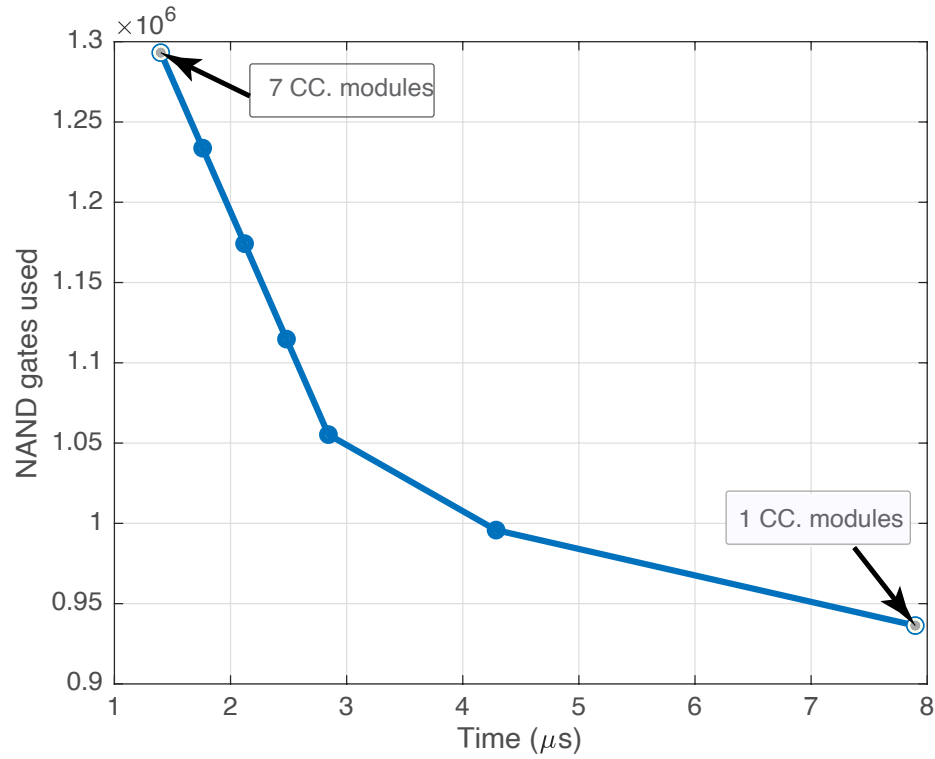


FIGURE 6.7: Calculation time and the number of NAND gates required for the design using between one and seven Clustering Coefficient (CC) modules to accelerate the calculations.

6.6 Conclusions

In this work, a new architecture was presented that is capable of extracting the PLI and the graph connectivity measurements from an EEG signal. For validation, a 19-channel version of the architecture was synthesised. The synthesised design requires a total of 71% of the total resources of the FPGA and can operate at a maximum frequency of 22.16 MHz. The maximum operating frequency can be tuned according to the application and the required processing time, helping to reduce the device's power consumption. However, in this work, it was decided to use the maximum frequency to reduce the calculation time and use it in other processes that may be more demanding (e.g. artifact removal). Also, the unused logic elements in the FPGA could be used then to extend the number of channels or include further processing of the graph connectivity measurements, according to the application needs. The power consumption of the system was equal to 51.84mW. The validation results show that the relative error percentage is low for all the measurements, with the largest error being the degree within the range of $\pm 7.9\%$. Additionally, the total time required for the calculation of all the graph connectivity measurements, at a maximum frequency achieved, was equal to 131 μs in our design, accelerating the calculation by at least one order of magnitude compared with the time required to

perform the same calculation on MATLAB (3.74ms), showing that the design can perform the real-time calculation of the functional brain connectivity and its graph-theoretic characterisation. Also, the results obtained show that the design presented here meets the design considerations discussed in Chapter 3, showing its feasibility to be used in systems that require real-time processing of the EEG signal such as neurofeedback systems.

Chapter 7

Preliminary Validation of a Seizure Classifier

7.1 Introduction

The usage of machine learning for the analysis of EEG signals has been used in different applications, from the emotional state and depression ([Wang et al. \(2014\)](#); [Hosseinfard et al. \(2013\)](#)) to the control of brain-computer interfaces ([Müller et al. \(2008\)](#)). Another major application that has been explored is in the clinical diagnosis for the detection of epileptic seizures ([Jaiswal and Banka \(2018\)](#)). Seizure detection not only serves as a clinical tool to avoid time-consuming visual inspections of large EEG data sets but it could also be used in the creation of portable devices that can accurately detect the seizure onset in a nomadic environment. Such a device can detect the seizure onset and quickly provide immediate therapy to the patient, reducing the risk of physical damage and improving the patient's quality of life. Such a system is one of the many possible scenarios in which the neuromodulation system described in Chapter 1 can be used. In this particular scenario, the neurofeedback system will continuously monitor the signal from the brain and discern if the patient is having a seizure or not and whether or not the application of the electrical stimulus is necessary to aid the patient.

To create such a system, it is still necessary to develop a system that can detect the seizure onset. A common approach in the literature for seizure detection is to use functional brain connectivity (FC) to identify the seizure onset. Typically, the seizure starts in a specific region of the brain and then it starts to propagate to the different areas of the brain generating the activation of multiple regions in the brain ([Lemieux et al. \(2011\)](#)). These changes in brain activity can be observed using functional brain connectivity, making it an excellent method for seizure detection algorithms ([Akbarian and Erfanian \(2020\)](#)). Examples of this are the studies from [Myers et al.](#)

(2016) and [Detti et al. \(2020\)](#), that have explored this possibility by implementing a seizure detection using FC measurements in the frequency domain as Phase-Locking Value (PLV) and Phase lag Index (PLI), respectively. Similarly, other approaches have explored the usage of graph theory measurements to characterise the brain connectivity networks created using techniques such as visibility graph, horizontal visibility graph, and difference visibility graph ([Artameeyanant et al. \(2017\)](#); [Wang et al. \(2017\)](#)).

However, few studies combine phase-based FC networks and the many different graph-theoretic parameters to detect seizures in the literature. To our knowledge, the only studies doing this are the ones presented by [Dhulekar et al. \(2015\)](#) and [Akbarian and Erfanian \(2017\)](#). In both of these studies, seizure detection is done using graph-theoretic parameters (e.g. clustering coefficient, transitivity, characteristic path length) extracted from FC networks created using PLI. However, the results from [Dhulekar et al. \(2015\)](#) were not as satisfactory as expected, whilst the study from [Akbarian and Erfanian \(2017\)](#) was done using only rodents.

To address this gap in the literature, in this chapter, we present a new design for a seizure detection classifier that uses the graph-theoretic parameters (introduced in Section 6.2) extracted from an FC network created using PLI. This study will allow us to explore the potential of combining these two approaches to characterise the patient's brain activity and use it to detect the seizure onset.

Moreover, in this chapter, we also use the classifier as a way to evaluate the total impact in the performance due to all calculations done in hardware for the graph-theoretic parameters calculated with the hardware architecture from Chapter 6, compared to a complete software approach as is typically done. This comparison provides a more realistic evaluation of how the error introduced by the hardware calculations is propagated through the system and its impact on the system's final performance. Also, this provides a quantitative analysis to evaluate the potential trade-off in the performance when using a hardware approach in such calculations.

7.2 Materials and methods

7.2.1 Data and pre-processing

The data used here was the "CHB-MIT" database that contains EEG recordings acquired from patients of the Childrens Hospital Boston. This database was produced by [Shoeb \(2009\)](#) and can be found online. This database contains the anonymised recordings from 24 patients with an age range between 1.5-22. The data was recorded using the 10-20 electrode placement and 23 channels at 256 Hz with 16-bit resolution.

The number of channels was reduced to 19 to match the number of channels used in the modules created in previous chapters. The only pre-processing done to the EEG signal was a band-pass filter with a cut-off frequency between 8-13 Hz, corresponding to the alpha band from the EEG signal. The decision to use the alpha band was based on the works from [Detti et al. \(2018\)](#) and [Lopes et al. \(2019\)](#), in which better results were obtained by using this particular frequency band.

7.2.2 Classification algorithms

The classifier was created using three different supervised machine learning algorithms: Random forest, LightGBM and Support vector machine. These algorithms were selected due to their low complexity and excellent performance with a low number of data available.

7.2.2.1 Random Forest

Random forest (RF) is a popular algorithm introduced by [Breiman \(2001\)](#), and it consists of the creation of multiple decision trees trained using randomly selected vector of samples and variables. Every decision tree contributes to the final decision, and the most popular class is selected as the output as in Figure 7.1. This algorithm is widely popular due to its good classification performance and speed in the classification process ([Belgiu and Drăguț \(2016\)](#)).

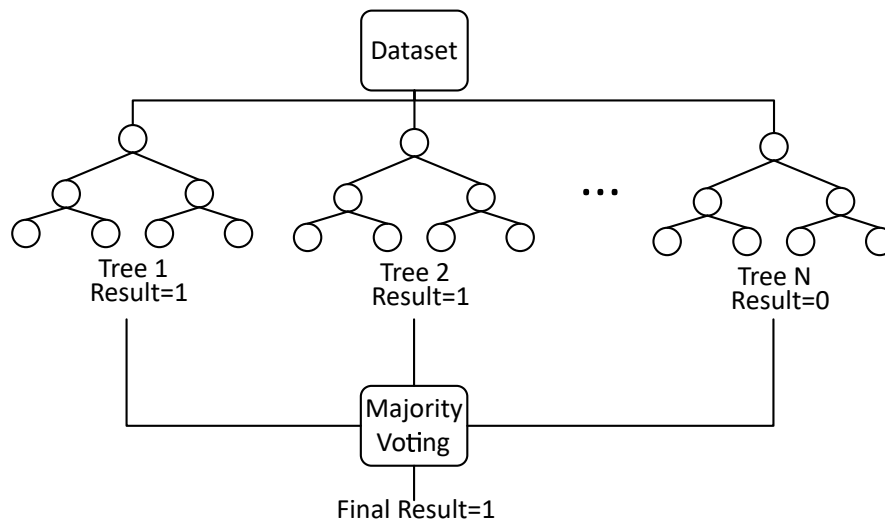


FIGURE 7.1: Diagram of a random decision forest. The classification done by multiple decision trees is combined and the final classification is done based on the majority of the results.

7.2.2.2 LightGBM

The LightGBM algorithm (LGBM) is a variation of the Gradient Boosting Decision Tree (GBDT) algorithm, introduced by [Ke et al. \(2017\)](#). This new algorithm introduces two new techniques to reduce the algorithm's computational complexity: gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB). The GOSS technique helps to significantly reduce the data size by excluding from the training the data with a low gradient and use only the data instances with a more significant gradient. Thus, in the LGBM algorithm, the trees are grown leaf-wise instead of level-wise, choosing the leaf with the maximum delta loss ([Shaker et al. \(2020\)](#)), as shown in Figure 7.2. The EFB technique reduces the number of data used by combining mutually exclusive features, improving the speed of the training and the prediction. Combining these two techniques results in optimising the tree splitting that accelerates the training process and improves the accuracy of the classification ([Ivacu \(2021\)](#)).

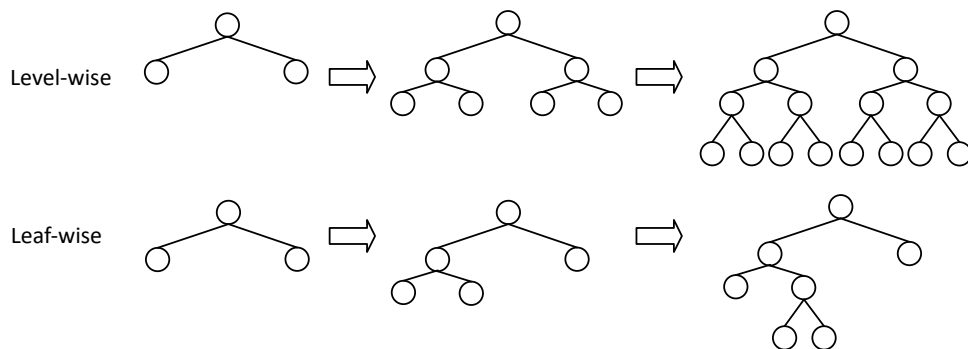


FIGURE 7.2: Comparison between the tree grow using level-wise and leaf-wise.

7.2.2.3 SVM

Firstly introduced by [Vapnik \(1995\)](#), the Support Vector Machine (SVM) is an algorithm widely used for classification problems in different areas such as image recognition, speech recognition, text categorisation, among others ([Pradhan \(2012\)](#)).

The concept behind SVM is to find a function that can separate the data from two or more distinct classes as in Figure 7.3. This function is known as hyperplane, and it is constructed by the SVM algorithm by maximising the distance between the hyper-plane and the support vectors formed by the closest points to the hyperplane. This process is widely used due to its easy training process and its good trade-off between complexity and error ([Chen et al. \(2017\)](#)).

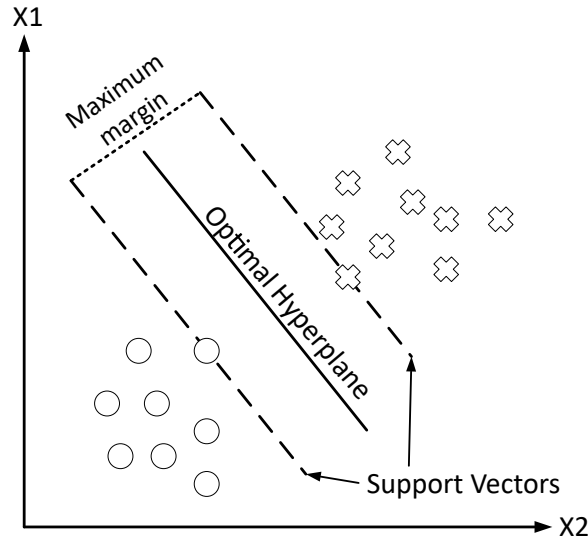


FIGURE 7.3: Example of a linear separation using the SVM algorithm. The optimal hyperplane corresponds to the plane with the maximum margin between itself and the support vectors formed by the closest points.

7.2.3 Methodology

The methodology followed for the design of the classifier is summarised in the diagram of Figure 7.4. Here, the EEG signal is used to extract the graph-theoretic parameters in software and hardware, so it is possible to compare the performance of these two approaches. These parameters are split into a training and testing data set used as features for the classifier. Then, two independent classifiers are trained using one of the machine learning algorithms previously discussed. After the training, both classifiers are tested, and their predictions are used to evaluate their performance. The exact process is repeated for every patient and every machine learning algorithm.

Following this methodology allows us to identify which algorithm provides the best performance and also quantify the total error introduced to the classification due to all the optimisations done for the hardware calculations.

7.2.3.1 Feature calculation

As depicted in the diagram from Figure 7.4, the calculation of the graph-theoretic parameters is done using a software and a hardware approach. The former was done using the different toolboxes available for MATLAB (NBT for the PLI and BCT for the graph-theoretic parameters), while the latter was done using the architecture presented in Chapter 6.

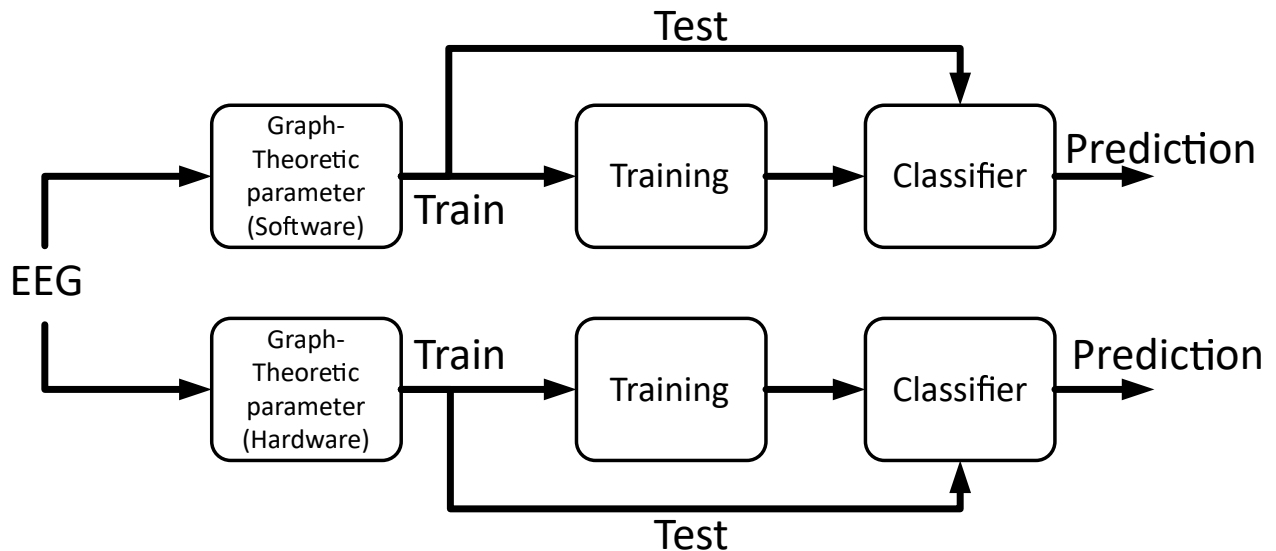


FIGURE 7.4: Block diagram of the process followed in the design of the classifier used for the validation.

7.2.3.2 Data split for training and testing

The splitting of the data for the training and the testing was done using a stratified five-fold cross-validation. A stratified cross-validation approach is preferred for data that suffers from class imbalance. Here, the data for the class corresponding to the seizure is significantly smaller than the other class. Therefore, a stratified approach will be better suited, preserving the original ratio between the classes among the folds. The splitting of the data was the same for the hardware and software features. The data was split using the same indices for both approaches to ensure that any mismatch in performance is due to optimizations in the hardware calculation of the features and not the splitting of the data.

7.3 Experimental Results

To assess the performance between the hardware and software calculations, we created different classifiers using the machine learning algorithms mentioned above. We produced two versions of them, one using the hardware calculations and the other using the software calculation, as depicted in Figure 7.4.

7.3.1 Performance metrics

The performance of the classifiers created in this work was evaluated using three different performance metrics: accuracy, sensitivity, and specificity. These metrics are defined in Equations 7.1 - 7.3, and they can be calculated in terms of the number of

true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) predictions obtained during the testing of the classifier.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (7.1)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (7.2)$$

$$Specificity = \frac{TN}{TN + FP} \quad (7.3)$$

7.3.1.1 Random Forest

The first pair of classifiers was created using the RF algorithm. For this algorithm, we utilise all the graph-theoretic parameters for the training and testing of the classifier. The same hyper-parameters were used for both classifiers to avoid differences in the performance due to these parameters. The performance of the algorithm is presented in Table 7.1.

The results show that the classifier created with the RF algorithm can perform the classification task properly. The average performance using the software calculations of the graph-theoretic parameters was 95.70%, 95.17% and 95.70% for the accuracy, sensitivity, and specificity. On the other hand, the average performance using the hardware calculations was 95.71%, 95.57% and 95.70% for the accuracy, sensitivity and specificity, respectively. The best performance across all patients was obtained using the hardware calculations of the graph-theoretic parameters, achieving maximum accuracy and specificity of 99.35% for patient 9, and a maximum sensitivity of 100% for patient 21. These results verify that the performance obtained using the hardware calculations are equal to or even better than using the software calculations.

TABLE 7.1: Performance comparison of the classifiers created using random forest, and the calculation of the features using hardware (HW) and software (SW).

	Accuracy		Sensitivity		Specificity	
	Sw	Hw	Sw	Hw	Sw	Hw
1	96.56%	95.94%	97.74%	98.11%	96.55%	95.93%
2	97.71%	97.89%	98.06%	98.06%	97.71%	97.89%
3	95.05%	94.79%	97.10%	96.68%	95.05%	94.78%
4	97.82%	98.12%	97.36%	96.92%	97.82%	98.12%
5	92.56%	92.81%	95.52%	95.22%	92.54%	92.80%
6	97.61%	97.47%	92.39%	95.65%	97.61%	97.47%
7	98.12%	97.78%	97.44%	97.44%	98.12%	97.78%
8	94.29%	94.36%	95.10%	96.55%	94.28%	94.33%
9	99.33%	99.35%	93.98%	93.98%	99.34%	99.35%
10	94.58%	93.74%	91.79%	94.03%	94.58%	93.74%
11	97.22%	96.74%	98.14%	98.14%	97.21%	96.73%
12	92.61%	93.33%	94.21%	93.90%	92.60%	93.32%
13	92.69%	93.59%	96.97%	96.97%	92.64%	93.55%
14	94.56%	95.00%	96.04%	97.03%	94.55%	95.00%
15	93.27%	94.02%	96.82%	96.90%	93.22%	93.98%
16	95.84%	95.19%	80.49%	80.49%	95.85%	95.20%
17	95.65%	95.91%	96.59%	97.16%	95.64%	95.90%
18	98.42%	98.15%	93.16%	92.63%	98.43%	98.17%
19	94.57%	95.82%	96.48%	97.18%	94.56%	95.81%
20	95.00%	95.16%	95.45%	95.45%	95.00%	95.16%
21	97.37%	96.96%	98.32%	100.00%	97.37%	96.96%
22	99.30%	99.19%	94.26%	93.44%	99.31%	99.20%
23	92.72%	92.59%	94.09%	95.67%	92.72%	92.58%
24	93.98%	93.09%	96.68%	96.01%	93.97%	93.07%
Mean	95.70%	95.71%	95.17%	95.57%	95.70%	95.70%

7.3.1.2 LightGBM

The next pair of classifiers was created using the LGBM algorithm. Here, all the graph-theoretic parameters were used to train and test the classifiers, and the same hyper-parameters were used for both classifiers. The performance of the algorithm is presented in Table 7.2.

The results obtained with this algorithm were the most satisfactory. The average performance using the software calculations was 99.66% for accuracy and specificity, and 99.35% for sensitivity. The average performance using the hardware calculations were the same, except for the sensitivity that came to 99.20%. The best performance across all patients was obtained with patient 19, achieving a 100% for all the performance metrics for software and hardware calculations. These results show that the performance for both calculations is close, and therefore the trade-off between the software and hardware calculations are minimal.

TABLE 7.2: Performance comparison of the classifiers created using LightGBM, and the calculation of the features using hardware (HW) and software (SW).

	Accuracy		Sensitivity		Specificity	
	Sw	Hw	Sw	Hw	Sw	Hw
1	99.96%	99.96%	99.25%	99.25%	99.96%	99.96%
2	99.95%	99.93%	98.06%	97.09%	99.95%	99.93%
3	99.68%	99.69%	99.17%	99.17%	99.68%	99.69%
4	99.69%	99.71%	100.00%	100.00%	99.69%	99.71%
5	99.86%	99.82%	100.00%	99.40%	99.86%	99.82%
6	99.74%	99.77%	100.00%	100.00%	99.74%	99.77%
7	99.78%	99.77%	100.00%	100.00%	99.78%	99.77%
8	99.39%	99.48%	99.82%	99.46%	99.39%	99.48%
9	99.76%	99.78%	98.19%	98.80%	99.76%	99.78%
10	99.72%	99.65%	98.13%	96.64%	99.73%	99.66%
11	99.87%	99.89%	100.00%	100.00%	99.87%	99.89%
12	99.47%	99.45%	99.09%	99.39%	99.47%	99.45%
13	99.65%	99.63%	100.00%	99.62%	99.64%	99.63%
14	99.62%	99.63%	100.00%	99.01%	99.62%	99.63%
15	99.32%	99.27%	99.75%	99.58%	99.32%	99.27%
16	99.04%	99.15%	95.12%	97.56%	99.04%	99.15%
17	99.78%	99.76%	100.00%	100.00%	99.78%	99.76%
18	99.15%	99.02%	97.89%	96.84%	99.15%	99.03%
19	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
20	99.23%	99.16%	100.00%	100.00%	99.23%	99.16%
21	99.88%	99.90%	100.00%	100.00%	99.88%	99.90%
22	99.67%	99.67%	100.00%	100.00%	99.67%	99.67%
23	99.98%	99.98%	100.00%	99.61%	99.98%	99.98%
24	99.67%	99.67%	100.00%	99.34%	99.67%	99.67%
Mean	99.66%	99.66%	99.35%	99.20%	99.66%	99.66%

7.3.1.3 SVM

The final pair of classifiers was created using the SVM algorithm. Contrary to the approach done in the previous algorithms, in the SVM algorithm, it was decided not to use all the graph-theoretic parameters as features since this will significantly impact the complexity of the algorithm. The complexity involved in the SVM algorithm can increase rapidly as the size of the data used increases, being able to reach a complexity of order $O(n^3)$ (Blachnik (2015)). To reduce the algorithm's complexity, the dimensionality of the data was reduced using the RF algorithm, keeping the most significant features. The RF algorithm can be used to rank the importance of the features as in Figure 7.5. The figure shows the feature importance, and it can be seen that the first 15 features are the most relevant in the data, with cumulative feature importance of about 0.99. Thus, it is possible to reduce the complexity of the SVM algorithm by using only these 15 features. This analysis was done for every patient since the importance of the features variate among all the patients. The same hyper-parameters were used for both classifiers.

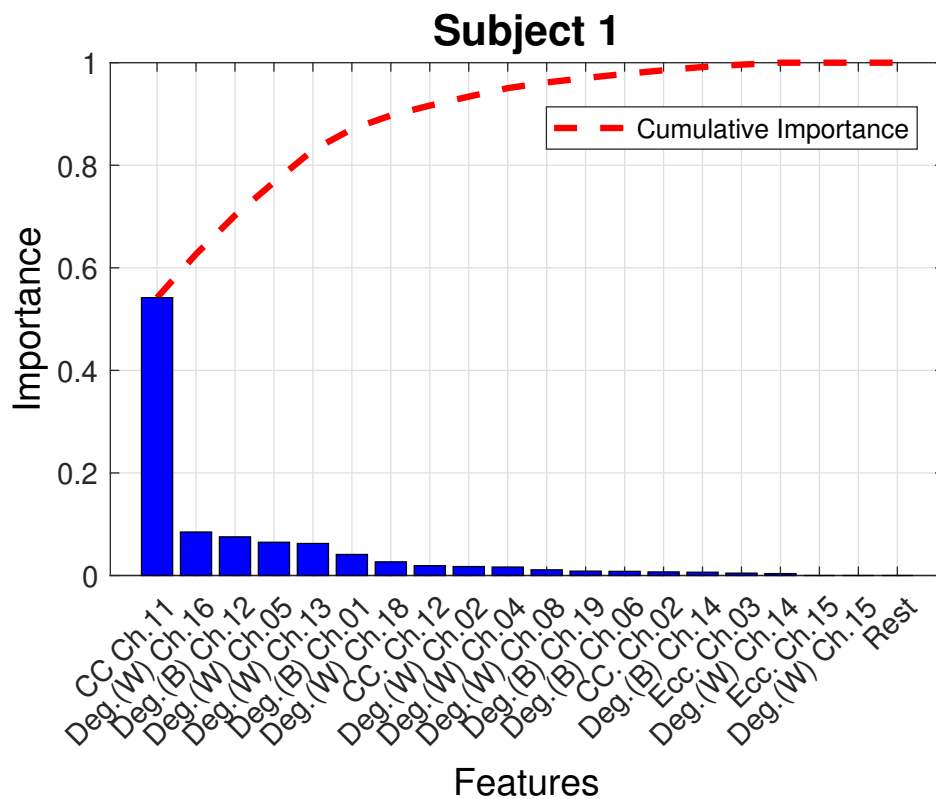


FIGURE 7.5: Feature importance ranking obtained with the random forest algorithm.

The performance results for the classifiers created using the SVM algorithm are shown in Table 7.3. The performance of these classifiers was good, achieving an average performance of 95.91%, 99.53% and 95.9% for the accuracy, sensitivity and specificity with the software calculations. The average performance for the hardware calculation was very similar, with a slight difference of 0.06% for the accuracy and specificity, and 0.01% for the sensitivity. Across patients, the best performance was obtained for patient 22 with 99.72% in the accuracy and specificity, and 100% sensitivity. The hardware performance for this patient was close, with a difference of 0.01% in accuracy and specificity. This difference shows how close the performance is from both calculations and the trivial trade-off that it represents for using the graph-theoretic parameters calculated in hardware instead of software.

TABLE 7.3: Performance comparison of the classifiers created using SVM, and the calculation of the features using hardware (HW) and software (SW).

	Accuracy		Sensitivity		Specificity	
	Sw	Hw	Sw	Hw	Sw	Hw
1	98.96%	98.86%	100.00%	100.00%	98.95%	98.86%
2	97.79%	97.69%	100.00%	100.00%	97.79%	97.69%
3	98.89%	98.86%	100.00%	100.00%	98.88%	98.85%
4	97.04%	97.11%	100.00%	100.00%	97.04%	97.11%
5	98.04%	98.61%	100.00%	100.00%	98.04%	98.61%
6	97.87%	98.03%	100.00%	100.00%	97.87%	98.03%
7	96.80%	96.98%	100.00%	100.00%	96.80%	96.98%
8	96.32%	96.38%	100.00%	100.00%	96.27%	96.33%
9	99.40%	99.39%	100.00%	100.00%	99.40%	99.39%
10	97.03%	97.55%	98.51%	98.51%	97.03%	97.55%
11	98.54%	98.38%	100.00%	100.00%	98.53%	98.37%
12	92.97%	92.99%	99.70%	99.70%	92.92%	92.94%
13	92.34%	92.42%	98.86%	100.00%	92.27%	92.34%
14	96.78%	96.85%	100.00%	100.00%	96.77%	96.84%
15	97.71%	97.75%	98.66%	98.66%	97.70%	97.74%
16	85.47%	85.68%	100.00%	100.00%	85.45%	85.66%
17	92.34%	92.19%	100.00%	100.00%	92.31%	92.16%
18	98.05%	98.13%	100.00%	100.00%	98.05%	98.13%
19	95.63%	96.20%	100.00%	100.00%	95.62%	96.19%
20	94.49%	94.47%	100.00%	100.00%	94.47%	94.45%
21	97.62%	97.54%	96.64%	98.32%	97.62%	97.54%
22	99.72%	99.71%	100.00%	100.00%	99.72%	99.71%
23	89.25%	88.83%	96.46%	93.70%	89.22%	88.81%
24	92.85%	92.75%	100.00%	100.00%	92.81%	92.70%
Mean	95.91%	95.97%	99.53%	99.54%	95.90%	95.96%

7.3.2 Comparison

To further assess the performance of the seizure detection classifiers from this work, we compared the performance with other seizure detection implementation in the literature. Although there are many seizure detection studies in the literature, in this comparison, we decided to use studies that have used the same database that we used (described in Section 7.2.1). The comparison is shown in Table 7.4.

From Table 7.4, we found that the worst performance in terms of sensitivity corresponds to [Khan et al. \(2012\)](#) with an 83.6%. However, the same study was able to obtain the best performance in terms of specificity from all the studies. This difference in the performance shows a common problem found in classifications systems, where the specificity can be increased at the cost of decreasing the sensitivity and vice-versa ([Sun et al. \(2020\)](#)). In our design, we decided to follow a more balanced approach where the sensitivity and specificity are high but remain similar to each other. For instance, the classifier with the best overall performance was presented in this work

TABLE 7.4: Performance comparison with other seizure detection implementation.

Study	Method	Sensitivity	Specificity
Zhou et al. (2018)	CNN	96.90%	98.10%
Fergus et al. (2015)	KNNC	93.00%	98.00%
Khan et al. (2012)	LDA	83.60%	100.00%
Nasehi and Pourghassem (2013)	IPSONN	98.00%	-
Shoeb and Guttag (2010)	SVM	96.00%	-
Elhosary et al. (2019)	SVM	96.70%	90.34%
Wang et al. (2018)	SVM	98.40%	-
Yoo (2017)	D^2A -SVM	95.70%	98.00%
This work (HW features)	RF	95.57%	95.70%
This work (HW features)	LGBM	99.20%	99.66%
This work (HW features)	SVM	99.54%	95.96%

created with the LGBM algorithm. Even though the performance of this classifier was the second-best in sensitivity and specificity, the performance is similar in both metrics contrary to the work from Khan et al. (2012) and the SVM classifier from this work (best performances in specificity and sensitivity respectively), with a higher margin in performance between the metrics.

Moreover, the classifier created with the RF algorithm presented the worst performance from all the designs in this work. However, its performance was better than the one from Fergus et al. (2015) and Khan et al. (2012) in terms of sensitivity and better than Elhosary et al. (2019) in terms of specificity. The classifier created with the SVM algorithm presented the best performance in the sensitivity and outperformed the work from Elhosary et al. (2019) in terms of specificity.

This comparison shows that the classifiers presented in this work have a similar or better performance than other implementations in the literature. From all the classifiers used for the comparison, it can be said that the one created using the LGBM algorithm is the best due to its high results ($\geq 99.2\%$) in both sensitivity and specificity.

7.4 Conclusions

In this work, the design of new classifiers for seizure detection using the graph-theoretic parameters were presented as a proof-of-concept application for the neurofeedback system presented in previous chapters. A comparison of the classifier's performance using these calculations showed that the final trade-off between using a software approach against a hardware approach is minimal, with a maximum difference of 0.15% in all the performance metrics used. The classifiers made using the graph-theoretic parameters calculated on hardware and the LGBM algorithm achieved

the best overall performance with an average sensitivity of 99.35% and specificity and accuracy of 99.66%. Furthermore, the designs presented in this work show a similar or better performance than other seizure detection classifiers in the literature.

The results obtained in this work are encouraging. They show that it is possible to utilize the graph-theoretic parameters in the detection of seizure and that the calculation of these parameters in hardware did not significantly impact the final performance of the seizure detector. Therefore, the graph-theoretic parameters can be used as a neurofeedback signal incorporated into any neuromodulation system.

Chapter 8

Conclusions and future work

The main aim of this thesis was the development of a low energy hardware architecture for the calculation of the graph-theoretic parameters from the EEG signals in real-time. This aim was accomplished in Chapter 6 where a new hardware architecture for calculating the graph-theoretic parameters was introduced. This architecture was implemented in an FPGA device that allows the whole calculation of the functional brain connectivity from the EEG signals and its characterisation in a small and portable device and removes the necessity of a computer to perform the same processing. The implementation performed the complete calculation in 273.10 μ s, with a power consumption of 51.84 mW, meeting the requirements established for this work of 180 ms and 712mW. Additionally, the preliminary validation presented in Chapter 7 demonstrated that the graph-theoretic parameters calculated using the hardware architecture from this work have a minimal impact (0.15% maximum) on the final performance compared to a software solution. Also, this seizure detection classifier serves as a proof-of-concept for how the graph-theoretic parameters could be used as a neurofeedback signal in the treatment of neurological diseases such as epilepsy and many other diseases.

8.1 Future Works

The different modules presented in this thesis provide a new alternative to the traditional offline processing of the EEG signal. Moreover, this work has the potential to be extended and create a new non-invasive wearable closed-loop neuromodulation system, similar to the one discussed by [Schestatsky et al. \(2013\)](#), that could be beneficial in the treatment of different neurological disorders ([Krook-Magnuson et al. \(2015\)](#)). The concept envisioned by our research group for such a system is presented in Figure 8.1. The work presented here corresponds to the first part of this system related to processing the EEG signal to create the neurofeedback signal. The second

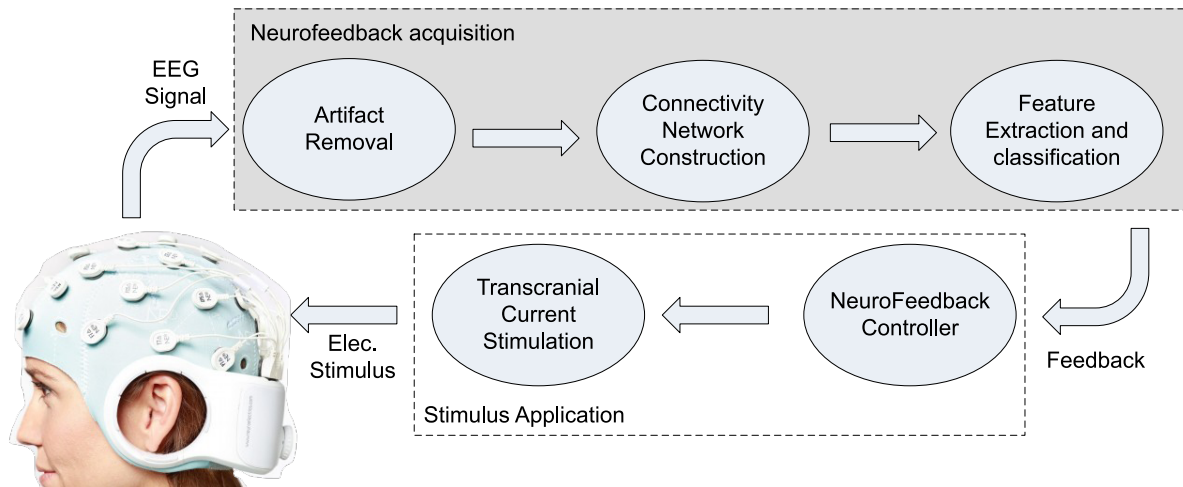


FIGURE 8.1: Overall schematic of the proposed non-invasive wearable closed-loop neuromodulation system.

part of the system related to the application of the stimulus remains unexplored, and further work needs to be done. Therefore, the following future work is proposed to complete the envisioned closed-loop neuromodulation system:

- Design of the electrical stimulus controller: It is necessary to design a new controller that can determine if the patient requires the stimulation based on the characteristics of the brain activity calculated by the hardware architecture presented in this work. A great example of how this can be achieved is the seizure classifier presented in this work. The patient will only require the electrical stimulus when the seizure is happening to compensate for the brain activity and stop the seizure. The classifier can determine when the seizure is happening or not, thus enabling the stimulus's application. Similarly, this can be extended to other disorders such as depression (Veer et al. (2010)) and attention-deficit/hyperactivity disorder (Tomasí and Volkow (2012)) where atypical brain activity can be identified.
- Creation of the protocol for the electrical stimulus application: As previously mentioned, there is not a clear consensus on how the electrical stimulus needs to be applied. Thus, it is necessary to keep exploring this area and identify how the stimulus needs to be adjusted based on the characteristics of the brain connectivity networks, similar to the work done by Yaquib et al. (2021) but extended to neurological diseases. Also, it is crucial to identify the specific areas of the brain that need to be stimulated according to the disorder since it will differ from one to another (Herrera-Melendez et al. (2020)). This focused stimulation will also involve the study of new stimulation techniques such as the one from Park et al. (2011), that properly focus the stimulus to the required area and avoid propagation to other areas that do not require it.

- System integration and prototyping: After developing the stimulus controller, it is possible to create a prototype of the whole closed-loop neuromodulation system previously described and start testing the device with actual patients. The designs presented here can be implemented in an ASIC to help reduce the device's size and energy consumption to create a more efficient device.

8.2 Final Thoughts

The work presented here opens the possibility to create a new kind of treatment for different mental and neurological disorders. The aforementioned closed-loop neuromodulation system could improve the efficiency of the TCS treatment and offer a new inexpensive tool that could help to bring the treatment to more people, particularly to low and middle-income countries where the burden of these disorders is more significant (Feigin et al. (2020)). For instance, the pilot trial from Alonzo et al. (2019) for a home-administered TCS showed positive results in the treatment of depression, demonstrating the feasibility of this technique as a more economical treatment that any person could easily use.

Furthermore, the usage of TCS has increased in the later years to a point where more and more companies started to invest in this area. Thus nowadays, it is possible to find commercial devices that aim to bring TCS to a broader public. For instance, the companies Flow Neuroscience and PlatoScience Medical offer commercial devices that anyone can use to treat depression. To give some perspective of the potential for these devices, the market research report presented by Markets and MarketsTM estimates that the epilepsy monitoring devices market is projected to reach USD 615 million by 2026 (Markets and Markets (2021)). Similarly, a study from Expert Market Research estimates that the global non-invasive neuromodulation market will grow to around USD 21 billion by 2026 (Research (2021)). These estimations show the huge market gap for neuromodulation systems at the moment and how projects like the one presented in this work can be further developed to provide a new device that can fit into this market that is still at an early stage.

References

- Enas Abdulhay, Maha Alafeef, Loai Alzghoul, Miral Al Momani, Rabah Al Abdi, N Arunkumar, Roberto Munoz, and Victor Hugo C de Albuquerque. Computer-aided autism diagnosis via second-order difference plot area applied to eeg empirical mode decomposition. *Neural Computing and Applications*, 32(15): 10947–10956, 2020.
- Amit Acharyya, Pranit N Jadhav, Valentina Bono, Koushik Maharatna, and Ganesh R Naik. Low-complexity hardware design methodology for reliable and automated removal of ocular and muscular artifact from eeg. *Computer methods and programs in biomedicine*, 158:123–133, 2018.
- Hojjat Adeli, Ziqin Zhou, and Nahid Dadmehr. Analysis of eeg records in an epileptic patient using wavelet transform. *Journal of neuroscience methods*, 123(1):69–87, 2003.
- Mehran Ahmadlou and Hojjat Adeli. Complexity of weighted graph: A new technique to investigate structural complexity of brain activities with applications to aging and autism. *Neuroscience letters*, 650:103–108, 2017.
- Mehran Ahmadlou, Hojjat Adeli, and Amir Adeli. Graph theoretical analysis of organization of functional brain networks in adhd. *Clinical EEG and neuroscience*, 43(1):5–13, 2012.
- Behnaz Akbarian and Abbas Erfanian. Automatic detection of ptz-induced seizures based on functional brain connectivity network in rats. In *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 576–579. IEEE, 2017.
- Behnaz Akbarian and Abbas Erfanian. A framework for seizure detection using effective connectivity, graph theory, and multi-level modular network. *Biomedical Signal Processing and Control*, 59:101878, 2020.
- Angelo Alonzo, Joanna Fong, Nicola Ball, Donel Martin, Nicholas Chand, and Colleen Loo. Pilot trial of home-administered transcranial direct current stimulation for the treatment of depression. *Journal of affective disorders*, 252:475–483, 2019.
- Amara Amara, Frederic Amiel, and Thomas Ea. Fpga vs. asic for low power applications. *Microelectronics journal*, 37(8):669–677, 2006.

- Ray Andraka. A survey of cordic algorithms for fpga based computers. In *Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays*, pages 191–200, 1998.
- Patcharin Artameeyanant, Sivarit Sultornsanee, and Kosin Chamnongthai. Electroencephalography-based feature extraction using complex network for automated epileptic seizure detection. *Expert Systems*, 34(3):e12211, 2017.
- Amauri Amorin Assef, Jonathan de Oliveira, Joaquim Miguel Maia, and Eduardo Tavares Costa. Fpga implementation and evaluation of an approximate hilbert transform-based envelope detector for ultrasound imaging using the dsp builder development tool. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2813–2816. IEEE, 2019.
- Behnam Badihi, Fayeze Ghavimi, and Riku Jäntti. On the system-level performance evaluation of bluetooth 5 in iot: Open office case study. In *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, pages 485–489. IEEE, 2019.
- Adam P Baker, Matthew J Brookes, Iead A Rezek, Stephen M Smith, Timothy Behrens, Penny J Probert Smith, and Mark Woolrich. Fast transient networks in spontaneous human brain activity. *Elife*, 3:e01867, 2014.
- R Balocchi, D Menicucci, and M Varanini. Empirical mode decomposition to approach the problem of detecting sources from a reduced number of mixtures. In *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439)*, volume 3, pages 2443–2446. IEEE, 2003.
- Mariana Belgiu and Lucian Drăguț. Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, 114:24–31, 2016.
- David H Benninger, Mikhail Lomarev, Grisel Lopez, Eric M Wassermann, Xiaobai Li, Elaine Considine, and Mark Hallett. Transcranial direct current stimulation for the treatment of parkinson’s disease. *Journal of Neurology, Neurosurgery & Psychiatry*, 81(10):1105–1111, 2010.
- Abi Berger. How does it work?: Magnetic resonance imaging. *BMJ: British Medical Journal*, 324(7328):35, 2002.
- Ronakben Bhavsar, Na Helian, Yi Sun, Neil Davey, Tony Steffert, and David Mayor. Efficient methods for calculating sample entropy in time series data analysis. *Procedia computer science*, 145:97–104, 2018.
- Marcin Blachnik. Reducing time complexity of svm model by lvq data compression. In *International Conference on Artificial Intelligence and Soft Computing*, pages 687–695. Springer, 2015.

- Paulo S Boggio, Lais P Khoury, Debora CS Martins, Oscar EMS Martins, EC De Macedo, and Felipe Fregni. Temporal cortex direct current stimulation enhances performance on a visual recognition memory task in alzheimer disease. *Journal of Neurology, Neurosurgery & Psychiatry*, 80(4):444–447, 2009.
- Valentina Bono, Wasifa Jamal, Saptarshi Das, and Koushik Maharatna. Artifact reduction in multichannel pervasive eeg using hybrid wpt-ica and wpt-emd signal decomposition techniques. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5864–5868. IEEE, 2014.
- Valentina Bono, Saptarshi Das, Wasifa Jamal, and Koushik Maharatna. Hybrid wavelet and emd/ica approach for artifact suppression in pervasive eeg. *Journal of neuroscience methods*, 267:89–107, 2016.
- Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- Javier D Bruguera et al. Composite iterative algorithm and architecture for q-th root calculation. In *2011 IEEE 20th Symposium on Computer Arithmetic*, pages 52–61. IEEE, 2011.
- Jerome Brunelin, Marine Mondino, Leila Gassab, Frederic Haesebaert, Lofti Gaha, Marie-Françoise Suaud-Chagny, Mohamed Saoud, Anwar Mechri, and Emmanuel Poulet. Examining transcranial direct-current stimulation (tdcs) as a treatment for hallucinations in schizophrenia. *American Journal of Psychiatry*, 169(7):719–724, 2012.
- György Buzsáki. Large-scale recording of neuronal ensembles, May 2004. ISSN 10976256. URL <http://www.nature.com/articles/nn1233>.
- Kai Cao, Ying Guo, and Steven W Su. A review of motion related eeg artifact removal techniques. In *2015 9th International Conference on Sensing Technology (ICST)*, pages 600–604. IEEE, 2015.
- Tony F Chan, Gene H Golub, and Randall J LeVeque. Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37(3): 242–247, 1983.
- He Chen, Yan Song, and Xiaoli Li. A deep learning framework for identifying children with adhd using an eeg-based brain network. *Neurocomputing*, 356:83–96, 2019a.
- Pei-Yin Chen, Yen-Chen Lai, and Ju-Yang Zheng. Hardware design and implementation for empirical mode decomposition. *IEEE Transactions on Industrial Electronics*, 63(6):3686–3694, 2016.
- Wei Chen, Hamid Reza Pourghasemi, Aiding Kornejady, and Ning Zhang. Landslide spatial modeling: Introducing new ensembles of ann, maxent, and svm machine learning techniques. *Geoderma*, 305:314–327, 2017.

- Xun Chen, Xueyuan Xu, Aiping Liu, Soojin Lee, Xiang Chen, Xu Zhang, Martin J McKeown, and Z Jane Wang. Removal of muscle artifacts from the eeg: a review and recommendations. *IEEE Sensors Journal*, 19(14):5353–5368, 2019b.
- Yun-Hsuan Chen, Maaïke Op De Beeck, Luc Vanderheyden, Evelien Carrette, Vojkan Mihajlović, Kris Vanstreels, Bernard Grundlehner, Stefanie Gadeyne, Paul Boon, and Chris Van Hoof. Soft, comfortable polymer dry electrodes for high quality ecg and eeg recording. *Sensors*, 14(12):23758–23780, 2014.
- Yu M Chi and Gert Cauwenberghs. Wireless non-contact eeg/ecg electrodes for body sensor networks. In *Body Sensor Networks (BSN), 2010 International Conference on*, pages 297–301. IEEE, 2010.
- Yu M Chi, Stephen R Deiss, and Gert Cauwenberghs. Non-contact low power eeg/ecg electrode for high density wearable biopotential sensor networks. In *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, pages 246–250. IEEE, 2009.
- José Chilo and Thomas Lindblad. Hardware implementation of 1d wavelet transform on an fpga for infrasound signal classification. *IEEE Transactions on Nuclear Science*, 55(1):9–13, 2008.
- Vaclav Cizek. Discrete hilbert transform. *IEEE Transactions on Audio and Electroacoustics*, 18(4):340–343, 1970.
- Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Minimum and maximum. In *Introduction to algorithms*, pages 214–215. MIT press, Cambridge, USA, 3 edition, 2009.
- Ian Daly, Martin Billinger, Reinhold Scherer, and Gernot Müller-Putz. On the automated removal of artifacts related to head movement from the eeg. *IEEE Transactions on neural systems and rehabilitation engineering*, 21(3):427–434, 2013.
- Ian Daly, Reinhold Scherer, Martin Billinger, and Gernot Müller-Putz. Force: Fully online and automated artifact removal for brain-computer interfacing. *IEEE transactions on neural systems and rehabilitation engineering*, 23(5):725–736, 2014.
- Juville Dario-Becker. *Biology: Mixed Majors, Part 1*–. OpenStax College, Rice University, 2013.
- Surya Das and Subha D Puthankattil. Complex network analysis of mci-ad eeg signals under cognitive and resting state. *Brain research*, 1735:146743, 2020.
- Willem de Haan, Yolande AL Pijnenburg, Rob LM Strijers, Yolande van der Made, Wiesje M van der Flier, Philip Scheltens, and Cornelis J Stam. Functional neural network analysis in frontotemporal dementia and alzheimer’s disease using eeg and graph theory. *BMC neuroscience*, 10(1):101, 2009.

- Paolo Detti, Garazi Zabalo Manrique de Lara, Renato Bruni, Marco Pranzo, Francesco Sarnari, and Giampaolo Vatti. A patient-specific approach for short-term epileptic seizures prediction through the analysis of eeg synchronization. *IEEE Transactions on Biomedical Engineering*, 66(6):1494–1504, 2018.
- Paolo Detti, Giampaolo Vatti, and Garazi Zabalo Manrique de Lara. Eeg synchronization analysis for seizure prediction: A study on data of noninvasive recordings. *Processes*, 8(7):846, 2020.
- Nimit Dhulekar, Srinivas Nambirajan, Basak Oztan, and Bülent Yener. Seizure prediction by graph mining, transfer learning, and transformation learning. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 32–52. Springer, 2015.
- Nuno Sérgio Dias, João Paulo Carmo, Paulo Mateus Mendes, and José Higinio Correia. Wireless instrumentation system based on dry electrodes for acquiring eeg signals. *Medical engineering & physics*, 34(7):972–981, 2012.
- Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- Linda Douw, Marjolein De Groot, Edwin Van Dellen, Jan J Heimans, Hanneke E Ronner, Cornelis J Stam, and Jaap C Reijneveld. functional connectivity is a sensitive predictor of epilepsy diagnosis after the first seizure. *PLoS One*, 5(5):e10839, 2010.
- Feng Duan, Zihao Huang, Zhe Sun, Yu Zhang, Qibin Zhao, Andrzej Cichocki, Zhenglu Yang, and Jordi Solé-Casals. Topological network analysis of early alzheimers disease based on resting-state eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(10):2164–2172, 2020.
- John S Duncan, Josemir W Sander, Sanjay M Sisodiya, and Matthew C Walker. Adult epilepsy. *The Lancet*, 367(9516):1087–1100, 2006.
- Waleed Ejaz, Mahin Atiq, Hyung Kim, et al. Recursive pyramid algorithm-based discrete wavelet transform for reactive power measurement in smart meters. *Energies*, 6(9):4721–4738, 2013.
- Heba Elhosary, Michael H Zakhari, Mohamed A Elgammal, Mohamed A Abd El Ghany, Khaled N Salama, and Hassan Mostafa. Low-power hardware implementation of a support vector machine training and classification for neural seizure detection. *IEEE transactions on biomedical circuits and systems*, 13(6):1324–1337, 2019.
- Marjolein MA Engels, Cornelis J Stam, Wiesje M van der Flier, Philip Scheltens, Hanneke de Waal, and Elisabeth CW van Straaten. Declining functional connectivity and changing hub locations in alzheimers disease: an eeg study. *BMC neurology*, 15(1):145, 2015.

- O Farooq and S Datta. Wavelet based robust sub-band features for phoneme recognition. *IEE Proceedings-Vision, Image and Signal Processing*, 151(3):187–193, 2004.
- Valery L Feigin, Theo Vos, Emma Nichols, Mayowa O Owolabi, William M Carroll, Martin Dichgans, Günther Deuschl, Priya Parmar, Michael Brainin, and Christopher Murray. The global burden of neurological disorders: translating evidence into policy. *The Lancet Neurology*, 19(3):255–265, 2020.
- Paul Fergus, David Hignett, Abir Hussain, Dhiya Al-Jumeily, and Khaled Abdel-Aziz. Automatic epileptic seizure detection using scalp eeg and advanced artificial intelligence techniques. *BioMed research international*, 2015, 2015.
- R Ferrucci, M Bortolomasi, M Vergari, L Tadini, B Salvoro, M Giacomuzzi, S Barbieri, and A Priori. Transcranial direct current stimulation in severe, drug-resistant major depression. *Journal of affective disorders*, 118(1-3):215–219, 2009.
- Karl J Friston. Functional and effective connectivity: a review. *Brain connectivity*, 1(1): 13–36, 2011.
- Juan García-Prieto, Ricardo Bajo, and Ernesto Pereda. Efficient computation of functional brain networks: towards real-time functional connectivity. *Frontiers in neuroinformatics*, 11:8, 2017.
- Joseph T Gwin, Klaus Gramann, Scott Makeig, and Daniel P Ferris. Removal of movement artifact from high-density eeg recorded during walking and running. *Journal of neurophysiology*, 103(6):3526–3534, 2010.
- Richard Hardstone, Simon-Shlomo Poil, Giuseppina Schiavone, Rick Jansen, Vadim V Nikulin, Huibert D Mansvelder, and Klaus Linkenkaer-Hansen. Detrended fluctuation analysis: a scale-free view on neuronal oscillations. *Frontiers in physiology*, 3:450, 2012.
- Rachel Harker. Nhs funding and expenditure sn/sg/724. *House of Commons Library*.(April 03, 2012), 2012.
- Mehrnaz Kh Hazrati, Haliza Mat Husin, and Ulrich G Hofmann. Wireless brain signal recordings based on capacitive electrodes. In *Intelligent Signal Processing (WISP), 2013 IEEE 8th International Symposium on*, pages 8–13. IEEE, 2013.
- Ana-Lucia Herrera-Melendez, Malek Bajbouj, and Sabine Aust. Application of transcranial direct current stimulation in psychiatry. *Neuropsychobiology*, 79(6): 372–383, 2020.
- N Jeremy Hill, Disha Gupta, Peter Brunner, Aysegul Gunduz, Matthew A Adamo, Anthony Ritaccio, and Gerwin Schalk. Recording human electrocorticographic (ecog) signals for neuroscientific research and real-time functional cortical mapping. *Journal of Visualized Experiments*, (64), June 2012. ISSN 1940-087X. . URL

- <http://www.ncbi.nlm.nih.gov/pubmed/22782131><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3471287><http://www.jove.com/video/3993/>.
- Ying-Yi Hong and Yu-Qing Bao. Fpga implementation for real-time empirical mode decomposition. *IEEE Transactions on Instrumentation and Measurement*, 61(12): 3175–3184, 2012.
- Behshad Hosseinifard, Mohammad Hassan Moradi, and Reza Rostami. Classifying depression patients and normal subjects using machine learning techniques and nonlinear features from eeg signal. *Computer methods and programs in biomedicine*, 109(3):339–345, 2013.
- Jing Hu, Chun-sheng Wang, Min Wu, Yu-xiao Du, Yong He, and Jinhua She. Removal of eeg and emg artifacts from eeg using combination of functional link neural network and adaptive neural fuzzy inference system. *Neurocomputing*, 151:278–287, 2015.
- Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. In *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences*, volume 454, pages 903–995. The Royal Society, 1998.
- Norden E Huang, Man-Li C Wu, Steven R Long, Samuel SP Shen, Wendong Qu, Per Gloersen, and Kuang L Fan. A confidence limit for the empirical mode decomposition and hilbert spectral analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 459(2037):2317–2345, 2003.
- Frances Hutchings, Cheol E Han, Simon S Keller, Bernd Weber, Peter N Taylor, and Marcus Kaiser. Predicting surgery targets in temporal lobe epilepsy through structural connectome based simulations. *PLoS computational biology*, 11(12): e1004642, 2015.
- Md Kafiul Islam, Amir Rastegarnia, and Zhi Yang. Methods for artifact detection and removal from scalp eeg: A review. *Neurophysiologie Clinique/Clinical Neurophysiology*, 46(4-5):287–305, 2016.
- Codru-Florin Ivacu. Option pricing using machine learning. *Expert Systems with Applications*, 163:113799, 2021.
- Meltem Izzetoglu, Scott C Bunce, Kurtulus Izzetoglu, Banu Onaral, and Kambiz Pourrezaei. Functional brain imaging using near-infrared technology. *IEEE Engineering in Medicine and Biology Magazine*, 26(4):38, 2007.

- Ali Jafari, Sunil Gandhi, Sri Harsha Konuru, W David Hairston, Tim Oates, and Tinoosh Mohsenin. An eeg artifact identification embedded system using ica and multi-instance learning. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4. IEEE, 2017.
- Abeg Kumar Jaiswal and Haider Banka. Epileptic seizure detection in eeg signal using machine learning techniques. *Australasian physical & engineering sciences in medicine*, 41(1):81–94, 2018.
- Wasifa Jamal, Saptarshi Das, Koushik Maharatna, Doga Kuyucu, Federico Sicca, Lucia Billeci, Fabio Apicella, and Filippo Muratori. Using brain connectivity measure of eeg synchrostates for discriminating typical and autism spectrum disorder. In *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*, pages 1402–1405. IEEE, 2013.
- Wasifa Jamal, Saptarshi Das, Ioana-Anastasia Oprescu, Koushik Maharatna, Fabio Apicella, and Federico Sicca. Classification of autism spectrum disorder using supervised learning of brain connectivity measures extracted from synchrostates. *Journal of neural engineering*, 11(4):046019, 2014.
- Xiao Jiang, Gui-Bin Bian, and Zean Tian. Removal of artifacts from eeg signals: a review. *Sensors*, 19(5):987, 2019.
- Karissa M Johnston, Lauren C Powell, Ian M Anderson, Shelagh Szabo, and Stephanie Cline. The burden of treatment-resistant depression: A systematic review of the economic and quality of life literature. *Journal of affective disorders*, 242:195–210, 2019.
- Valer Jurcak, Daisuke Tsuzuki, and Ippeita Dan. 10/20, 10/10, and 10/5 systems revisited: their validity as relative head-surface-based positioning systems. *Neuroimage*, 34(4):1600–1611, 2007.
- Yoshinao Kajikawa and Charles E. Schroeder. How local is the local field potential? *Neuron*, 72(5):847–858, December 2011. ISSN 08966273. . URL <http://www.sciencedirect.com/science/article/pii/S089662731100883X>.
- A Kandaswamy, V Krishnaveni, S Jayaraman, N Malmurugan, and K Ramadoss. Removal of ocular artifacts from eega survey. *IETE journal of research*, 51(2):121–130, 2005.
- Shinya Kasakawa, Teruya Yamanishi, Tetsuya Takahashi, Kanji Ueno, Mitsuru Kikuchi, and Haruhiko Nishimura. Approaches of phase lag index to eeg signals in alzheimers disease from complex network analysis. In *Innovation in Medicine and Healthcare 2015*, pages 459–468. Springer, 2016.
- Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017.

- Yusuf Uzzaman Khan, Nidal Rafiuddin, and Omar Farooq. Automated seizure detection in scalp eeg using multiple wavelet scales. In *2012 IEEE international conference on signal processing, computing and control*, pages 1–5. IEEE, 2012.
- Atilla Kilicarslan, Robert G Grossman, and Jose Luis Contreras-Vidal. A robust adaptive denoising framework for real-time artifact removal in scalp eeg measurements. *Journal of neural engineering*, 13(2):026013, 2016.
- Murielle Kirkove, Clémentine François, and Jacques Verly. Comparative evaluation of existing and new methods for correcting ocular artifacts in electroencephalographic recordings. *Signal Processing*, 98:102–120, 2014.
- George H Klem, Hans Otto Lüders, HH Jasper, C Elger, et al. The ten-twenty electrode system of the international federation. *Electroencephalogr Clin Neurophysiol*, 52(3):3–6, 1999.
- Julia E Kline, Helen J Huang, Kristine L Snyder, and Daniel P Ferris. Isolating gait-related movement artifacts in electroencephalography during human walking. *Journal of neural engineering*, 12(4):046022, 2015.
- Marius Klug and Klaus Gramann. Identifying key factors for improving ica-based decomposition of eeg data in mobile and stationary experiments. *European Journal of Neuroscience*, 2020.
- Kishore Kota and Joseph R. Cavallaro. Numerical accuracy and hardware tradeoffs for cordic arithmetic for special-purpose processors. *IEEE Transactions on Computers*, 42(7):769–779, 1993.
- V Krishnaveni, S Jayaraman, L Anitha, and K Ramadoss. Removal of ocular artifacts from eeg using adaptive thresholding of wavelet coefficients. *Journal of neural engineering*, 3(4):338, 2006.
- Esther Krook-Magnuson, Jennifer N Gelinas, Ivan Soltesz, and György Buzsáki. Neuroelectronics and biooptics: closed-loop technologies in neurological disorders. *JAMA neurology*, 72(7):823–829, 2015.
- Abraham Kuruvilla and Roland Flink. Intraoperative electrocorticography in epilepsy surgery: Useful or not? *Seizure*, 12(8):577–584, December 2003. ISSN 10591311. . URL <http://www.ncbi.nlm.nih.gov/pubmed/14630497>.
- TJ La Vaque. The history of eeg hans berger: psychophysiolgist. a historical vignette. *Journal of Neurotherapy*, 3(2):1–9, 1999.
- Jean-Philippe Lachaux, Eugenio Rodriguez, Jacques Martinerie, and Francisco J Varela. Measuring phase synchrony in brain signals. *Human brain mapping*, 8(4):194–208, 1999.

- Himadri Lala and Subrata Karmakar. Classification of arc fault between broken conductor and high-impedance surface: an empirical mode decomposition and stockwell transform-based approach. *IET Generation, Transmission & Distribution*, 14 (22):5277–5286, 2020.
- John LaRocco, Minh Dong Le, and Dong-Guk Paeng. A systemic review of available low-cost eeg headsets used for drowsiness detection. *Frontiers in neuroinformatics*, 14, 2020.
- Dong-U Lee, Ray Cheung, Wayne Luk, and John Villasenor. Hardware implementation trade-offs of polynomial approximations and interpolations. *IEEE Transactions on computers*, 57(5):686–701, 2008.
- Young-Eun Lee, No-Sang Kwak, and Seong-Whan Lee. A real-time movement artifact removal method for ambulatory brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2020.
- Efrén L Lema-Condo, Freddy L Bueno-Palomeque, Susana E Castro-Villalobos, Esteban F Ordoñez-Morales, and Luis J Serpa-Andrade. Comparison of wavelet transform symlets (2-10) and daubechies (2-10) for an electroencephalographic signal analysis. In *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pages 1–4. IEEE, 2017.
- Louis Lemieux, Jean Daunizeau, and Matthew Walker. Concepts of connectivity and human epileptic activity. *Frontiers in systems neuroscience*, 5:12, 2011.
- Yeru Liang, Chen-Zi Zhao, Hong Yuan, Yuan Chen, Weicai Zhang, Jia-Qi Huang, Dingshan Yu, Yingliang Liu, Maria-Magdalena Titirici, Yu-Lun Chueh, et al. A review of rechargeable batteries for portable electronic devices. *InfoMat*, 1(1):6–32, 2019.
- Wei Liao, Zhiqiang Zhang, Zhengyong Pan, Dante Mantini, Jurong Ding, Xujun Duan, Cheng Luo, Guangming Lu, and Huaifu Chen. Altered functional connectivity and small-world in mesial temporal lobe epilepsy. *PloS one*, 5(1):e8525, 2010.
- Marinho A Lopes, Suejen Perani, Siti N Yaakub, Mark P Richardson, Marc Goodfellow, and John R Terry. Revealing epilepsy type using a computational analysis of interictal eeg. *Scientific reports*, 9(1):1–10, 2019.
- Louis Yu Lu. Fast intrinsic mode decomposition of time series data with sawtooth transform. *arXiv preprint arXiv:0710.3170*, 2007.
- Huageng Luo, Xingjie Fang, and Bugra Ertas. Hilbert transform and its engineering applications. *AIAA journal*, 47(4):923–932, 2009.
- Koushik Maharatna, Evangelos B Mazomenos, John Morgan, and Silvio Bonfiglio. Towards the development of next-generation remote healthcare system: Some

- practical considerations. In *2012 IEEE International Symposium on Circuits and Systems*, pages 1–4. IEEE, 2012.
- Mohamed I Mahmoud, Moawad IM Dessouky, Salah Deyab, and Fatma H Elfouly. Signal denoising by wavelet packet transform on fpga technology. *Ubiquitous Computing and Communication journal*, 3:54–58, 2008.
- Charvi A Majmudar and Bashir I Morshed. Autonomous oa removal in real-time from single channel eeg data on a wearable device using a hybrid algebraic-wavelet algorithm. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(1):1–16, 2016.
- Nadia Mammone, Fabio La Foresta, and Francesco Carlo Morabito. Automatic artifact rejection from multichannel scalp eeg by wavelet ica. *IEEE Sensors Journal*, 12(3): 533–542, 2011.
- Markets and Markets. Epilepsy monitoring devices market - global forecast to 2026 — marketsandmarkets, 2021. URL https://www.marketsandmarkets.com/Market-Reports/epilepsy-monitoring-devices-market-62758189.html?utm_source=prnewswire&utm_medium=referral&utm_campaign=paidp.
- Lawrence Marple. Computing the discrete-time” analytic” signal via fft. *IEEE Transactions on signal processing*, 47(9):2600–2603, 1999.
- Martin Mc Hugh, Oisín Cawley, Fergal McCaffcry, Ita Richardson, and Xiaofeng Wang. An agile v-model for medical device software development to overcome the challenges with plan-driven software development lifecycles. In *2013 5th International Workshop on Software Engineering in Health Care (SEHC)*, pages 12–19. IEEE, 2013.
- Brenton W McMenamin, Alexander J Shackman, Lawrence L Greischar, and Richard J Davidson. Electromyogenic artifacts and electroencephalographic inferences revisited. *Neuroimage*, 54(1):4–9, 2011.
- Pramod K Meher, Javier Valls, Tso-Bing Juang, K Sridharan, and Koushik Maharatna. 50 years of cordic: Algorithms, architectures, and applications. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(9):1893–1907, 2009.
- Xu Mei-hua, Chen Zhang-jin, Ran Feng, and Cheng Yu-lan. Architecture research and vlsi implementation for discrete wavelet packet transform. In *Conference on High Density Microsystem Design and Packaging and Component Failure Analysis, 2006. HDP’06.*, pages 1–4. IEEE, 2006.
- Md KI Molla, Toshihisa Tanaka, Tomasz M Rutkowski, and Andrzej Cichocki. Separation of eeg artifacts from eeg signals using bivariate emd. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 562–565. IEEE, 2010.

- Klaus-Robert Müller, Michael Tangermann, Guido Dornhege, Matthias Krauledat, Gabriel Curio, and Benjamin Blankertz. Machine learning for real-time single-trial eeg-analysis: from brain-computer interfacing to mental state monitoring. *Journal of neuroscience methods*, 167(1):82–90, 2008.
- Mark H Myers, Akshay Padmanabha, Gahangir Hossain, Amy L de Jongh Curry, and Charles D Blaha. Seizure prediction and detection via phase and amplitude lock values. *Frontiers in human neuroscience*, 10:80, 2016.
- A Nandhini, KP Bharath, and Rajesh Kumar M Mahalti Mohammed Sohail. Denoising of speech signal using empirical mode decomposition and kalman filter.
- Saadat Nasehi and Hossein Pourghassem. Patient-specific epileptic seizure onset detection algorithm based on spectral features and ipsonn classifier. In *2013 international conference on communication systems and network technologies*, pages 186–190. IEEE, 2013.
- Guiomar Niso, Ricardo Bruña, Ernesto Pereda, Ricardo Gutiérrez, Ricardo Bajo, Fernando Maestú, and Francisco Del-Pozo. Hermes: towards an integrated toolbox to characterize functional and effective brain connectivity. *Neuroinformatics*, 11(4): 405–434, 2013.
- P.L. Nunez and R. Srinivasan. *Electric Fields of the Brain: The Neurophysics of EEG*. Oxford University Press, 2006. ISBN 9780195050387. URL https://books.google.co.uk/books?id=fUv54as56_8C.
- Rafael Angel Gutierrez Nuno and Koushik Maharatna. A phase lag index hardware calculation for real-time electroencephalography studies. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 644–647. IEEE, 2019.
- Rafael Angel Gutierrez Nuno, Chi Hang Raphael Chung, and Koushik Maharatna. Hardware architecture for real-time eeg-based functional brain connectivity parameter extraction. *Journal of Neural Engineering*, 18(3):036012, 2021.
- Iyad Obeid, Miguel AL Nicolelis, and Patrick D Wolf. A low power multichannel analog front end for portable neural signal recordings. *Journal of Neuroscience Methods*, 133(1-2):27–32, 2004.
- Seiji Ogawa and Tso-Ming Lee. Magnetic resonance imaging of blood vessels at high fields: in vivo and in vitro measurements and image simulation. *Magnetic resonance in medicine*, 16(1):9–18, 1990.
- Kazuki Onikura and Keiji Iramina. Evaluation of a head movement artifact removal method for eeg considering real-time processing. In *2015 8th Biomedical Engineering International Conference (BMEiCON)*, pages 1–4. IEEE, 2015.

- Chandrajit Pal, Dwaipayan Biswas, Koushik Maharatna, and Amlan Chakrabarti. Architecture for complex network measures of brain connectivity. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4. IEEE, 2017.
- Ji-Hye Park, Seung Bong Hong, Do-Won Kim, Minah Suh, and Chang-Hwan Im. A novel array-type transcranial direct current stimulation (tdcs) system for accurate focusing on targeted brain areas. *IEEE Transactions on Magnetics*, 47(5):882–885, 2011.
- Alex Pineiro, Javier D Bruguera, Fabrizio Lamberti, and Paolo Montuschi. A radix-2 digit-by-digit architecture for cube root. *IEEE Transactions on Computers*, 57(4): 562–566, 2008.
- Ashis Pradhan. Support vector machine-a survey. *International Journal of Emerging Technology and Advanced Engineering*, 2(8):82–85, 2012.
- Weibao Qiu, Yanyan Yu, and Lei Sun. A programmable, cost-effective, real-time high frequency ultrasound imaging board based on high-speed fpga. In *2010 IEEE International Ultrasonics Symposium*, pages 1976–1979. IEEE, 2010.
- Valentina Quaresima and Marco Ferrari. Functional near-infrared spectroscopy (fnirs) for assessing cerebral cortex function during human behavior in natural/social situations: A concise review. *Organizational Research Methods*, page 1094428116658959.
- Atul Rahman et al. New efficient hardware design methodology for modified non-restoring square root algorithm. In *2014 International Conference on Informatics, Electronics & Vision (ICIEV)*, pages 1–6. IEEE, 2014.
- Faridah Abd Rahman, Mohd Fauzi Othman, and Nurul Aimi Shaharuddin. A review on the current state of artifact removal methods for electroencephalogram signals. In *Control Conference (ASCC), 2015 10th Asian*, pages 1–6. IEEE, 2015.
- D Rangaprakash and N Pradhan. Study of phase synchronization in multichannel seizure eeg using nonlinear recurrence measure. *Biomedical Signal Processing and Control*, 11:114–122, 2014.
- Gabriela G Regner, Patrícia Pereira, Douglas T Leffa, Carla de Oliveira, Rafael Vercelino, Felipe Fregni, and Iraci LS Torres. Preclinical to clinical translation of studies of transcranial direct-current stimulation in the treatment of epilepsy: a systematic review. *Frontiers in neuroscience*, 12:189, 2018.
- Expert Market Research, 2021. URL <https://www.expertmarketresearch.com/reports/noninvasive-neuromodulation-market>.
- Gabriel Rilling, Patrick Flandrin, Paulo Goncalves, et al. On empirical mode decomposition and its algorithms. 3(3):8–11, 2003.

- Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52(3):1059–1069, 2010.
- Doha Safieddine, Amar Kachenoura, Laurent Albera, Gwénaél Birot, Ahmad Karfoul, Anca Pasnicu, Arnaud Biraben, Fabrice Wendling, Lotfi Senhadji, and Isabelle Merlet. Removal of muscle artifact from eeg data: comparison between stochastic (ica and cca) and deterministic (emd and wavelet-based) approaches. *EURASIP Journal on Advances in Signal Processing*, 2012(1):127, 2012.
- Daniel San-juan, León Morales-Quezada, Adolfo Josué Orozco Garduño, Mario Alonso-Vanegas, Maricarmen Fernández González-Aragón, Dulce Anabel Espinoza López, Rafael Vázquez Gregorio, David J Ansel, and Felipe Fregni. Transcranial direct current stimulation in epilepsy. *Brain stimulation*, 8(3):455–464, 2015.
- Saman Sargolzaei, Mercedes Cabrerizo, Mohammed Goryawala, Anas Salah Eddin, and Malek Adjouadi. Scalp eeg brain functional connectivity networks in pediatric epilepsy. *Computers in biology and medicine*, 56:158–166, 2015.
- Phattarapong Sawangjai, Supanida Hompoonsup, Pitshaporn Leelaarporn, Supavit Kongwudhikunakorn, and Theerawit Wilaiprasitporn. Consumer grade eeg measuring sensors as research tools: A review. *IEEE Sensors Journal*, 20(8):3996–4024, 2019.
- Pedro Schestatsky, Leon Morales-Quezada, and Felipe Fregni. Simultaneous eeg monitoring during transcranial direct current stimulation. *Journal of visualized experiments: JoVE*, (76), 2013.
- Dieter Schmidt and Wolfgang Löscher. Drug resistance in epilepsy: putative neurobiologic and clinical mechanisms. *Epilepsia*, 46(6):858–877, 2005.
- Bilal Shaker, Myeong-Sang Yu, Jin Sook Song, Sunjoo Ahn, Jae Yong Ryu, Kwang-Seok Oh, and Dokyun Na. Lightbbb: computational prediction model of blood–brain-barrier penetration based on lightgbm. *Bioinformatics*, 2020.
- Ali H Shoeb and John V Guttag. Application of machine learning to epileptic seizure detection. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 975–982, 2010.
- Ali Hossam Shoeb. *Application of machine learning to epileptic seizure onset detection and treatment*. PhD thesis, Massachusetts Institute of Technology, 2009.
- Willeke Staljanse, Gregor Strobbe, Roel Van Herten, Gwénaél Birot, Markus Gschwind, Margitta Seeck, Stefaan Vandenbergh, Serge Vulliémont, and Pieter Van Mierlo. Seizure onset zone localization from ictal high-density eeg in refractory focal epilepsy. *Brain topography*, 30(2):257–271, 2017.

- CJ Stam, Y Van Der Made, YAL Pijnenburg, and PH Scheltens. Eeg synchronization in mild cognitive impairment and alzheimer's disease. *Acta Neurologica Scandinavica*, 108(2):90–96, 2003.
- Cornelis J Stam and BW Van Dijk. Synchronization likelihood: an unbiased measure of generalized synchronization in multivariate data sets. *Physica D: Nonlinear Phenomena*, 163(3-4):236–251, 2002.
- Cornelis J Stam, BF Jones, G Nolte, M Breakspear, and Ph Scheltens. Small-world networks and functional connectivity in alzheimer's disease. *Cerebral cortex*, 17(1): 92–99, 2006.
- Cornelis J Stam, Guido Nolte, and Andreas Daffertshofer. Phase lag index: assessment of functional connectivity from multi channel eeg and meg with diminished bias from common sources. *Human brain mapping*, 28(11):1178–1193, 2007.
- Catherine Stamoulis, Lawrence J Gruber, and Bernard S Chang. Network dynamics of the epileptic brain at rest. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 150–153. IEEE, 2010.
- Diederick Stoffers, JLW Bosboom, JB Deijen, E Ch Wolters, Cornelis J Stam, and Henk W Berendse. Increased cortico-cortical functional connectivity in early-stage parkinson's disease: an meg study. *Neuroimage*, 41(2):212–222, 2008.
- D Puthankattil Subha, Paul K Joseph, Rajendra Acharya, and Choo Min Lim. Eeg signal analysis: a survey. *Journal of medical systems*, 34(2):195–212, 2010.
- Junfeng Sun, Zhijun Li, and Shanbao Tong. Inferring functional neural connectivity with phase synchronization analysis: a review of methodology. *Computational and mathematical methods in medicine*, 2012, 2012.
- Wenzhe Sun, Jan-Dirk Schmöcker, and Toshiyuki Nakamura. On the tradeoff between sensitivity and specificity in bus bunching prediction. *Journal of Intelligent Transportation Systems*, pages 1–17, 2020.
- Supriya Supriya, Siuly Siuly, Hua Wang, Jinli Cao, and Yanchun Zhang. Weighted visibility graph with complex network features in the detection of epilepsy. *IEEE access*, 4:6554–6566, 2016.
- Shravani Sur and VK Sinha. Event-related potential: An overview. *Industrial psychiatry journal*, 18(1):70, 2009.
- Kevin T Sweeney, Tomás E Ward, and Seán F McLoone. Artifact removal in physiological signalspractices and possibilities. *IEEE transactions on information technology in biomedicine*, 16(3):488–500, 2012.

- Pinar Tekturk, Ezgi Tuna Erdogan, Adnan Kurt, Ebru Nur Vanli-Yavuz, Esme Ekizoglu, Ece Kocagoncu, Zeynep Kucuk, Serkan Aksu, Nerses Bebek, Zuhall Yapici, et al. The effect of transcranial direct current stimulation on seizure frequency of patients with mesial temporal lobe epilepsy with hippocampal sclerosis. *Clinical neurology and neurosurgery*, 149:27–32, 2016.
- Michal Teplan et al. Fundamentals of eeg measurement. *Measurement science review*, 2(2):1–11, 2002.
- James Theiler. Spurious dimension from correlation algorithms applied to limited time-series data. *Physical review A*, 34(3):2427, 1986.
- John A Thompson, David Lanctin, Nuri Firat Ince, and Aviva Abosch. Clinical implications of local field potentials for understanding and treating movement disorders. *Stereotactic and functional neurosurgery*, 92(4):251–263, 2014.
- Trevor Thompson, Tony Steffert, Tomas Ros, Joseph Leach, and John Gruzelier. Eeg applications for sport and performance. *Methods*, 45(4):279–288, 2008.
- Wing Ting To, Dirk De Ridder, John Hart Jr, and Sven Vanneste. Changing brain networks through non-invasive neuromodulation. *Frontiers in human neuroscience*, 12:128, 2018.
- Dardo Tomasi and Nora D Volkow. Abnormal functional connectivity in children with attention-deficit/hyperactivity disorder. *Biological psychiatry*, 71(5):443–450, 2012.
- Vicente Torres, Javier Valls, and Richard Lyons. Fast-and low-complexity atan2 (a, b) approximation [tips and tricks]. *IEEE Signal Processing Magazine*, 34(6):164–169, 2017.
- Gabriel Tortella, Roberta Casati, Luana VM Aparicio, Antonio Mantovani, Natasha Senço, Giordano D’Urso, Jerome Brunelin, Fabiana Guarienti, Priscila Mara Lorencini Selingardi, Débora Muszkat, et al. Transcranial direct current stimulation in psychiatric disorders. *World journal of psychiatry*, 5(1):88, 2015.
- Lau Troy M, Gwin Joseph T, and Ferris Daniel P. How many electrodes are really needed for eeg-based mobile brain imaging? *Journal of Behavioral and Brain Science*, 2012, 2012.
- Jose Antonio Urigüen and Begoña Garcia-Zapirain. Eeg artifact removal state-of-the-art and guidelines. *Journal of Neural Engineering*, 12(3):031001, 2015. ISSN 1741-2560. . URL <http://stacks.iop.org/1741-2552/12/i=3/a=031001?key=crossref.ac94ab4f39506d3939e0e43f9ef3972e>.
- Rene L Utianski, John N Caviness, Elisabeth CW van Straaten, Thomas G Beach, Brittany N Dugger, Holly A Shill, Erika D Driver-Dunckley, Marwan N Sabbagh, Shyamal Mehta, Charles H Adler, et al. Graph theory network function in

- parkinsons disease assessed with electroencephalography. *Clinical Neurophysiology*, 127(5):2228–2236, 2016.
- Olivier Valentin, Mikaël Ducharme, Gabrielle Crétot-Richert, Hami Monsarrat-Chanon, Guilhem Viallet, Aidin Delnavaz, and Jérémie Voix. Validation and benchmarking of a wearable eeg acquisition platform for real-world applications. *IEEE transactions on biomedical circuits and systems*, 13(1):103–111, 2018.
- K Van der Hiele, AA Vein, A Van Der Welle, J Van Der Grond, RGJ Westendorp, ELEM Bollen, MA Van Buchem, JG Van Dijk, and HAM Middelkoop. Eeg and mri correlates of mild cognitive impairment and alzheimer’s disease. *Neurobiology of aging*, 28(9):1322–1329, 2007.
- Vladimir N Vapnik. Constructing learning algorithms. In *The nature of statistical learning theory*, pages 119–166. Springer, 1995.
- Ilya M Veer, Christian Beckmann, Marie-Jose Van Tol, Luca Ferrarini, Julien Milles, Dick Veltman, André Aleman, Mark A Van Buchem, Nic JA Van Der Wee, and Serge AR Rombouts. Whole brain resting-state analysis reveals decreased functional connectivity in major depression. *Frontiers in systems neuroscience*, 4:41, 2010.
- KN Vijeyakumar, V Sumathy, P Vasakipriya, and A Dinesh Babu. Fpga implementation of low power high speed square root circuits. pages 1–5, 2012.
- Antonio Vita, Alessandra Minelli, Stefano Barlati, Giacomo Deste, Edoardo Giacomuzzi, Paolo Valsecchi, Cesare Turrina, and Massimo Gennarelli. Treatment-resistant schizophrenia: genetic and neuroimaging correlates. *Frontiers in Pharmacology*, 10:402, 2019.
- Jack E Volder. The cordic trigonometric computing technique. *IRE Transactions on electronic computers*, (3):330–334, 1959.
- Theo Vos, Ryan M Barber, Brad Bell, Amelia Bertozzi-Villa, Stan Biryukov, Ian Bolliger, Fiona Charlson, Adrian Davis, Louisa Degenhardt, Daniel Dicker, et al. Global, regional, and national incidence, prevalence, and years lived with disability for 301 acute and chronic diseases and injuries in 188 countries, 1990–2013: a systematic analysis for the global burden of disease study 2013. *The Lancet*, 386(9995):743–800, 2015.
- Deng Wang, Duoqian Miao, and Chen Xie. Best basis-based wavelet packet entropy feature extraction and hierarchical eeg classification for epileptic detection. *Expert Systems with Applications*, 38(11):14314–14320, 2011.
- Hongda Wang, Weiwei Shi, and Chiu-Sing Choy. Hardware design of real time epileptic seizure detection based on stft and svm. *IEEE Access*, 6:67277–67290, 2018.

- Lei Wang, Mang I Vai, Peng Un Mak, and Chion In Ieong. Hardware-accelerated implementation of emd. *IEEE Biomed. Eng. Inf*, pages 912–915, 2010.
- Lei Wang, Xi Long, Johan BAM Arends, and Ronald M Aarts. Eeg analysis of seizure patterns using visibility graphs for detection of generalized seizures. *Journal of neuroscience methods*, 290:85–94, 2017.
- Xiao-Wei Wang, Dan Nie, and Bao-Liang Lu. Emotional state classification from eeg data using machine learning approach. *Neurocomputing*, 129:94–106, 2014.
- Duncan J Watts and Steven H Strogatz. Collective dynamics of small-world networks. *nature*, 393(6684):440, 1998.
- Fabrice Wendling, Karim Ansari-Asl, Fabrice Bartolomei, and Lotfi Senhadji. From eeg signals to brain connectivity: a model-based evaluation of interdependence measures. *Journal of neuroscience methods*, 183(1):9–18, 2009.
- WHO. Mental disorders affect one in four people, October 2001. URL http://www.who.int/whr/2001/media_centre/press_release/en/.
- Jiawei Xu, Srinjoy Mitra, Chris Van Hoof, Refet Firat Yazicioglu, and Kofi AA Makinwa. Active electrodes for wearable eeg acquisition: Review and electronics design methodology. *IEEE reviews in biomedical engineering*, 10:187–198, 2017.
- M Atif Yaqub, Keum-Shik Hong, Amad Zafar, and Chang-Seok Kim. Control of transcranial direct current stimulation duration by assessing functional connectivity of near-infrared spectroscopy signals. *International Journal of Neural Systems*, page 2150050, 2021.
- Nick Yeung, Rafal Bogacz, Clay B Holroyd, and Jonathan D Cohen. Detection of synchronized oscillations in the electroencephalogram: an evaluation of methods. *Psychophysiology*, 41(6):822–832, 2004.
- Yuma Yokoi, Zui Narita, and Tomiki Sumiyoshi. Transcranial direct current stimulation in depression and psychosis: a systematic review. *Clinical EEG and neuroscience*, 49(2):93–102, 2018.
- Jerald Yoo. On-chip epilepsy detection: Where machine learning meets patient-specific healthcare. In *2017 International SoC Design Conference (ISOCC)*, pages 146–147. IEEE, 2017.
- Chu Yu and Mao-Hsu Yen. Area-efficient 128-to 2048/1536-point pipeline fft processor for lte and mobile wimax systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(9):1793–1800, 2014.
- Meichen Yu, Alida A Gouw, Arjan Hillebrand, Betty M Tijms, Cornelis Jan Stam, Elisabeth CW van Straaten, and Yolande AL Pijnenburg. Different functional

- connectivity and network topology in behavioral variant of frontotemporal dementia and alzheimer's disease: an eeg study. *Neurobiology of aging*, 42:150–162, 2016.
- Daoqiang Zhang, Jiashuang Huang, Biao Jie, Junqiang Du, Liyang Tu, and Mingxia Liu. Ordinal pattern: A new descriptor for brain connectivity networks. *IEEE transactions on medical imaging*, 37(7):1711–1722, 2018.
- Zhiqiang Zhang, Guangming Lu, Yuan Zhong, Qifu Tan, Wei Liao, Zhili Chen, Jixin Shi, and Yijun Liu. Impaired perceptual networks in temporal lobe epilepsy revealed by resting fmri. *Journal of neurology*, 256(10):1705–1713, 2009.
- Mengni Zhou, Cheng Tian, Rui Cao, Bin Wang, Yan Niu, Ting Hu, Hao Guo, and Jie Xiang. Epileptic seizure detection based on eeg signals and cnn. *Frontiers in neuroinformatics*, 12:95, 2018.
- Weidong Zhou and Jean Gotman. Automatic removal of eye movement artifacts from the eeg using ica and the dipole model. *Progress in Natural Science*, 19(9):1165–1170, 2009.