



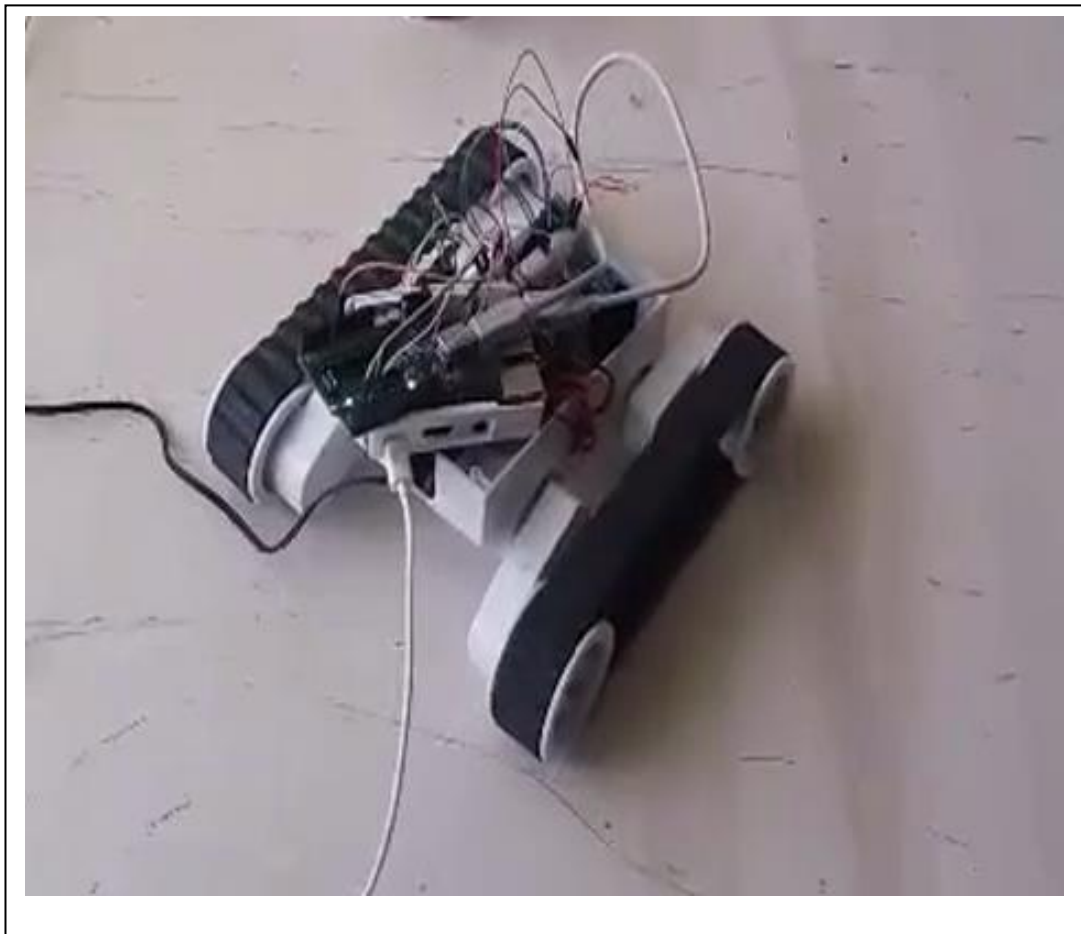
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

Έλεγχος Κίνησης Ρομποτικού Οχήματος μέσω Ηλεκτρο-εγκεφαλικών Σημάτων



**Φοιτητής: Νικόλαος Κοροβέσης
ΑΜ: 6416**

Επιβλέποντες Καθηγητές

**Διονύσης Κανδρής, Αναπληρωτής Καθηγητής
Αλέξανδρος Αλεξανδρίδης, Καθηγητής**

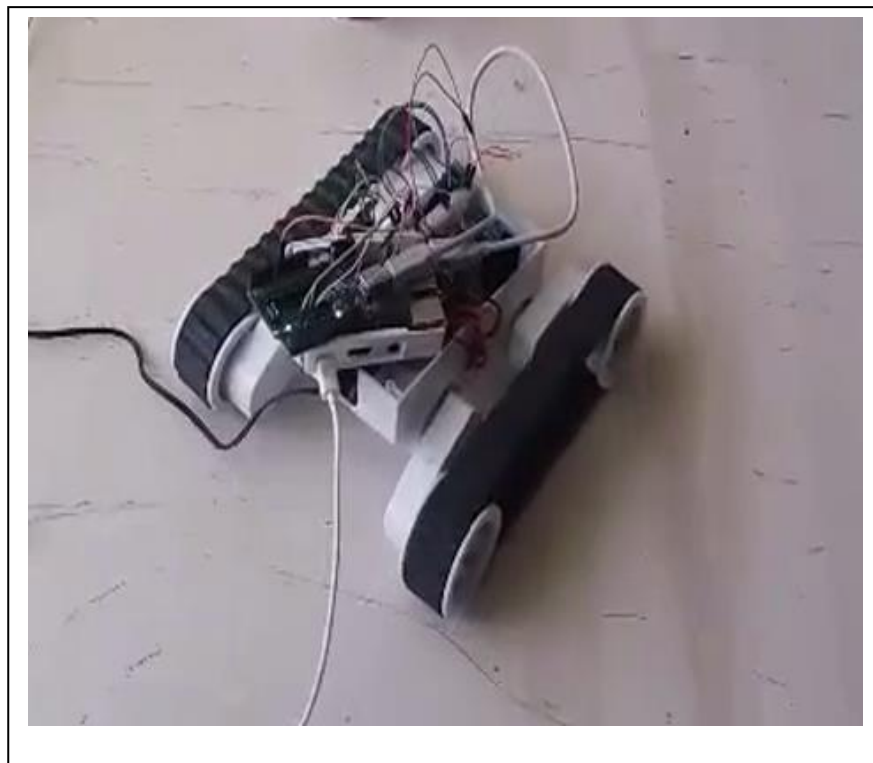
ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΜΑΡΤΙΟΣ 2021



UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Diploma Thesis

Robot Motion Control via EEG Signals




Student: Nikolaos Korovesis
Registration Number: 6416

Supervisors

Dionisis Kandris, Associate Professor
Alex Alexandridis, Professor

ATHENS-EGALEO, MARCH 2021

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Διονύσιος Κανδρής Αναπληρωτής Καθηγητής	Αλέξανδρος Αλεξανδρίδης Καθηγητής	Γρηγόριος Κουλούρας Επίκουρος Καθηγητής
		

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Νικόλαος Κοροβέσης, Απρίλιος 2021

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΠΕΡΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ ΚΑΙ ΛΟΓΟΚΛΟΠΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπόγραφα ότι η παρούσα εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα αποκλειστικά και ότι είμαι ο αποκλειστικός συγγραφέας του κειμένου της.

Η εργασία μου δεν προσβάλλει οποιασδήποτε μορφής δικαιώματα πνευματικής ιδιοκτησίας, προσωπικότητας ή προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής ή λογοκλοπής.

Κάθε βοήθεια που έλαβα για την ολοκλήρωση της εργασίας είναι αναγνωρισμένη και αναφέρεται λεπτομερώς στο κείμενό της. Ειδικότερα, έχω αναφέρει ευδιάκριτα μέσα στο κείμενο και με την κατάλληλη παραπομπή όλες τις πηγές δεδομένων, κώδικα προγραμματισμού Η/Υ, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών που χρησιμοποιήθηκαν, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης, και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Επιπλέον, όλες οι πηγές που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης κατά τα διεθνή πρότυπα.

Τέλος δηλώνω ενυπόγραφα ότι αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της είναι προϊόν λογοκλοπής.

Ημερομηνία 6 Απριλίου 2021

Νικόλαος Κοροβέσης

This work is dedicated to my mother, who taught me resilience and love.

I would like to thank my professors, Dr. Alex Alexandridis and Dr. Dionisis Kandris, for providing me with their guidance and wisdom throughout this challenge. This work would have never been completed without them. Finally, I'd like to thank my family and friends for all their support throughout my student life.

Περίληψη

Σκοπός της διπλωματικής αυτής εργασίας είναι η ανάπτυξη ενός κινητού ρομποτικού συστήματος του οποίου οι κινήσεις ελέγχονται ανάλογα με το άνοιγμα/κλείσιμο των ματιών ενός χειριστή μέσω μιας σύγχρονης και ενδογενούς Διεπαφής Εγκεφάλου-Υπολογιστή, βασισμένης σε Ηλεκτροεγκεφαλογράφημα, η οποία χρησιμοποιεί τα άλφα εγκεφαλικά κύματα. Τα σήματα που καταγράφονται φιλτράρονται ανάλογα έτσι ώστε να εξαχθούν τα κατάλληλα χαρακτηριστικά. Τα χαρακτηριστικά αυτά εισάγονται σε ένα νευρωνικό δίκτυο, το οποίο εκπαιδεύεται ώστε να οδηγήσει το ρομποτικό όχημα επιτυχώς. Τα πειραματικά τεστ τα οποία διεξήχθησαν, απέδειξαν ότι το σύστημα που αναπτύχθηκε είναι ικανό να εκτελέσει τις κινήσεις του ρομποτικού οχήματος σε 4 κατευθύνσεις: ευθεία, όπισθεν, δεξιά και αριστερά, ανάλογα με τα άλφα εγκεφαλικά κύματα του χειριστή του συστήματος, με ακρίβεια της τάξης του 92.1%.

Λέξεις – κλειδιά

Ηλεκτροεγκεφαλογράφημα (ΗΕΓ) , Νευρωνικά Δίκτυα, Διεπαφές Εγκεφάλου-Υπολογιστή, Άλφα Εγκεφαλικά Κύματα, Έλεγχος Ρομπότ

Abstract

The work presented in this thesis concerns the development of a system which performs motion control in a mobile robot in accordance to the eyes' blinking of a human operator via a synchronous and endogenous Electroencephalography-based Brain-Computer Interface, which uses alpha brain waveforms. The received signals are filtered in order to extract suitable features. These features are fed as inputs to a neural network, which is properly trained in order to properly guide the robotic vehicle. Experimental tests executed, proved that the system developed is able to perform movements of the robotic vehicle, under control, in forward, left, backward and right direction according to the alpha brainwaves of its' operator, with an overall accuracy equal to 92.1%.

Keywords

Electroencephalography (EEG), Neural Networks, Brain-Computer Interfaces, Alpha Brainwaves, Robot Control

Table of Contents

List of Tables	9
List of Figures	9
Glossary	11
1 INTRODUCTION	13
1.1 Diploma Thesis Subject.....	13
1.2 Purpose and Goals	13
1.3 Methodology	13
1.4 Innovation	14
1.5 Structure.....	14
2 CHAPTER 1 : Electroencephalography (EEG)	15
2.1 EEG Signals and their Characteristics.....	15
2.2 EEG Recording Principles	17
2.3 EEG Hardware	18
3 CHAPTER 2 : Machine Learning	19
3.1 Data and Feature Extraction	19
3.1.1 Data and Datasets.....	19
3.1.2 Features and Feature Extraction	20
3.2 Neural Networks	20
3.2.1 Perceptron	24
3.3 Multilayer Perceptrons.....	25
3.4 Variations and Applications of Neural Networks	29
4 CHAPTER 3 : Brain-Computer Interfaces	31
4.1 Brainwaves for EEG-BCIs.....	32
4.2 BCI Operation.....	32
4.3 BCI-Based Robot Control	33
5 CHAPTER 4 : Robot Motion Control via EEG Signals	35
5.1 Signal Acquisition	39
5.2 Preprocessing & Feature Extraction	43
5.2.1 Preprocessing.....	43
5.2.2 Feature Extraction	45
5.3 Classification & Translation	46
6 RESULTS & DISCUSSION	49
6.1 Results	49
6.1.1 Evaluation with Offline Data.....	49
6.1.2 Evaluation with Online Data	49
6.2 Discussion	53
7 CONCLUSIONS & FUTURE WORK	54
Bibliography – Citations – Internet Sources	56

List of Tables

Table 1. Binary sequences with corresponding robotic movements	42
Table 2. Hamming distances between robot commands	54

List of Figures

Figure 1. Hans Berger, inventor of the EEG	15
Figure 2. Different types of brainwaves and their waveforms	16
Figure 3. International 10 - 20 system with relative positions distances	17
Figure 4. Typical instrumentation amplifier.....	18
Figure 5. Non-linear model of a neuron	21
Figure 6. Sigmoid activation function	22
Figure 7. Hyperbolic tangent activation function.....	22
Figure 8. Rectified Linear Unit (ReLU) function.....	23
Figure 9. Signal flow graph of a multilayer perceptron	26
Figure 10. Block diagram representing the processes performed in a typical Brain-Computer Interface.....	32
Figure 11. Raspberry Pi 4 (Model B)	35
Figure 12. Arduino UNO.....	36
Figure 13. OpenBCI Ganglion	36
Figure 14. Dagu Rover 5 chassis	37
Figure 15. L298N motor driver module	37
Figure 16. OpenBCI GUI	39
Figure 17. From left to right: Ten20 conductive paste, soap powder, ultrasound gel.....	40
Figure 18. The motor homunculus on the precentral gyrus. The motor cortex can be seen in the lateral view of the cerebral cortex	40
Figure 19. Electrode placement setup	41
Figure 20. Raw EEG signals as recorded from the board	43
Figure 21. EEG signals after notch filtering.....	44
Figure 22. Final EEG signals after both filters were applied	44

Figure 23. Bar chart showing the normalized sum of the FFT amplitudes for each EEG channel...	46
Figure 24. Structure of the neural network built	47
Figure 25 Training and validation model loss	48
Figure 26. Final robot setup	49
Figure 27. Confusion matrix for all issued subject commands	50
Figure 28. Confusion matrix for female subjects	51
Figure 29. Confusion matrix for male subjects	51
Figure 30. Confusion matrix for ages 20 to 28.....	52
Figure 31. Confusion matrix for ages 32 to 40.....	53

Glossary

ALS: Amyotrophic Lateral Sclerosis
ANN: Artificial Neural Network
API: Application Programming Interface
BCI: Brain-Computer Interface
BK: Backpropagation
CMRR: Common-Mode Rejection Ratio
CNN: Convolutional Neural Network
DFT: Discrete Fourier Transform
ECG/EKG: Electrocardiography
EEG: Electroencephalography
EMG: Electromyography
ERD / ERS: Event-Related Desynchronization / Synchronization
ErrP: Error Potential
FFT: Fast Fourier Transform
FIR: Finite Impulse Response
fMRI: functional Magnetic Resonance Imaging
GAN: Generative Adversarial Network
GUI: Graphical User Interface
HDMI: High-Definition Multimedia Interface
IIR: Infinite Impulse Response
LDA: Linear Discriminant Analysis
MEG: Magnetoencephalography
MLP: Multilayer Perceptron
MS: Multiple Sclerosis
NIRS: Near Infrared Spectroscopy
NN: Neural Network
PET: Positron Emission Tomography
RBF Network: Radial Basis Function Network

ReLU: Rectified Linear Unit

RNN: Recurrent Neural Network

SDK: Software Development Kit

SSVEP: Steady-State Visually-Invoked Potential

SVM: Support Vector Machine

TCP: Transmission Control Protocol

USB: Universal Serial Bus

1 INTRODUCTION

1.1 Diploma Thesis Subject

According to the World Health Organization [1], 15% of the world's population lives with some form of disability, of whom 2-4% experience significant difficulties in functioning. This estimate is set to rise due to population aging along with the rapid spread of chronic diseases. It is safe to assume that a very significant portion of those percentages concerns people with physical disabilities. It is therefore of paramount importance for societies to find ways to improve the quality of life of those individuals and whenever possible cure them.

Focusing on physical disabilities, there are a lot of conditions that can hinder individuals' quality of life. Tetraplegia, Hemiplegia, Parkinson's Disease are some of the more well-known ones. The effects can range from difficulty of movement to loss of movement entirely.

Although research is ongoing, most of these disorders cannot be cured fully, so the focus has shifted more into minimizing the symptoms and providing assistance. Brain-Computer Interfaces, which are designed to not depend on the brain's normal output channels of peripheral nerves and muscles have given rise to many exciting applications and possibilities. By utilizing brain signals directly, they can be used for rehabilitation purposes or to control assistive systems.

While it's indeed a very promising technology, Brain-Computer Interfaces are far from being the panacea for all physical disabilities. Recording and classifying signals directly from the brain is an arduous task and it requires big and complex systems that offer high resolution. In addition, systems have to be simple and user-friendly in order to be applicable to real-world situations.

This diploma thesis concerns the development of a highly reliable synchronous Brain-Computer Interface that performs robot motion control via EEG signals using neural networks as a classifier and alpha brainwaves as the control signal. By being able to move a robot, users of this system could also be able to move any robotic device, such as a robotic wheelchair or arm.

1.2 Purpose and Goals

As mentioned in the last section the purpose of this work concerns the development of a highly reliable Brain-Computer Interface. The main goals of this project are divided as follows:

- Develop a system that is user-friendly so that every person could use it with minimal training.
- The system has to be highly accurate, preferably classifying over 90% of commands correctly.
- With minor changes, the system can be expanded to drive an assistive system, such as a robotic wheelchair.
- Develop a fully functional system, not a simulation.

1.3 Methodology

For this diploma thesis, a top-down approach was followed. The BCI system was broken down into the most basic BCI components and functions, which were then separately implemented. Moreover, the original purpose was to make each program (subsystem) as modular as possible, which made testing and debugging easier.

All different components are implemented as classes in different modules, acting independently. In the end the main script utilizes all those classes and their functions to form the final system. The ease of use in testing and debugging is apparent in the following problem that was encountered:

Originally, the system was developed on a computer running Windows 10. Due to some Bluetooth interfacing issues on this operating system, the connection between the computer and the board was refused. By only running the EEG acquisition module this problem was isolated rapidly and solved. If the system was not modular, this would take a longer time and effort to find isolate.

In order to evaluate the accuracy of this system, it was tested using healthy human subjects executing all available commands after a brief explanation on how the system works. The system's overall accuracy was then derived. Finally, the overall accuracy for different genders and age groups was evaluated.

1.4 Innovation

The innovation of this system lies in how the control signals are implemented. The alpha blocking phenomenon is utilized to form bit sequences which users can manipulate by closing or opening their eyes. While in this work a specific number of commands is used, those commands can be extended with ease by changing the size of each sequence.

In addition, this phenomenon is present in all humans in varying degrees, which makes for a perfect candidate in developing a BCI that is applicable to a wide range of users while retaining its simplicity.

1.5 Structure

The rest of this work is organized as follows: Section 2 introduces Electroencephalography (EEG), delving deeper into EEG signals and their characteristics, the principles of EEG recording and finally, what kind of hardware is used to record EEG signals. Next, in Section 3, machine learning is introduced, starting with data and datasets, features and feature extraction and then moving on the main topic, which is neural networks. Starting with a single perceptron and moving on to Multilayer Perceptrons, the inner workings of neural networks are explained, and finally some variations of neural networks and their main areas of applications are shown. Section 4 introduces Brain-Computer Interfaces (BCIs), focusing on EEG BCIs, the main brainwaves that are used as control signals, how BCIs work and their components and how they are used in controlling assistive systems, focusing on robots. This concludes the introductory chapters.

Next, the main topic is presented and analyzed thoroughly in Section 5, which is Robot Motion Control via EEG Signals. All the inner workings and components of the BCI that was developed are explained in detail, which are signal acquisition, preprocessing, feature extraction, classification and translation.

The results of this work along with the discussion are presented in Section 6. This diploma thesis is concluded in Section 7 and potential future work is laid down. The final pages contain the references that were used.

2 CHAPTER 1 : Electroencephalography (EEG)

Communication within the body of mammals takes place via both electrical and chemical signals. Electrophysiology is the branch of physiology that studies the electrical activities which are associated with bodily parts. The recording of electrophysiological data is performed by placing electrodes at the corresponding areas of interest. By this method, there are numerous systems developed which are able to monitor the electrical activity and corresponding electrophysiological data in various organs such as heart, brain, eyes, muscles, and stomach [2 – 4].

Electroencephalography (EEG) is an electrophysiological method which is used in order to monitor the electrical activity of the brain by placing electrodes on the external surface of the scalp. EEG records variations of voltage caused by the flow of ionic current in the interior of the brain's neurons.

Hans Berger is credited as being the inventor of EEG, to which he gave the name. Based on a small laboratory in Jena, he recorded the first human EEG in 1924. He achieved this feat by using a Siemens double-coil galvanometer, with a sensitivity of 130 $\mu\text{V}/\text{cm}$. Because of doubts of the validity of his work, it wasn't until 1929, 5 years later after his initial discovery that he published his findings under the title "On the Electroencephalogram of Man". His findings were independently confirmed by British scientists Edgar Douglas Adrian and B. H. C. Matthews in 1934. Some very important findings of Berger besides EEG include the discovery of alpha waves, the alpha wave blocking phenomenon, the first EEG recordings of sleep and many more.



Figure 1. Hans Berger, inventor of the EEG

2.1 EEG Signals and their Characteristics

As mentioned above, EEG measures the electrical activity of the brain, otherwise called *brainwaves*. These brainwaves are divided into the following categories, depending on their frequencies [5]:

- **Delta Waves [0 – 4Hz]:** Grey Walter was the first to assign the term "delta waves" to particular types of slow waves recorded in the EEG of humans. The delta term is associated with a frequency range rather than phenomena generating specific electrographic patterns. Until now, there are two primary sources of delta activities: one originating in the thalamus and another one in the cortex. They are prevalent during sleep or anesthesia.

- **Theta Waves [4 – 7Hz]:** First recorded in the hippocampus of epileptic patients by using indwelling electrodes and also using Magnetoencephalography (MEG) in normal subjects. Studies have shown that subjects had an increased frequency and rhythmicity of hippocampal delta while they were writing instead of walking or sitting. Moreover, in another experiment, the amplitude, frequency and rhythmicity of theta waves increased during silent mental activity.

- **Alpha Waves [8 – 13Hz]:** As mentioned above, alpha waves were the first brainwaves to be discovered, thus marking the beginning of EEG. Alpha waves originate from many different areas of the brain, but they are prominent in the occipital lobe, the visual processing center of the mammalian brain. Their amplitude is increased whenever the eyes of an individual are closed during wakeful relaxation. In contrast, the amplitude of alpha waveforms is diminished for the duration of sleep or sleepiness and also when the eyes of an individual are open and mental effort is performed. This phenomenon, which was discovered by Berger is called alpha rhythm blocking, or alpha blocking. Typical amplitudes for alpha waves are $\sim 100\mu\text{V}$.

- **Mu Waves [8 – 13Hz]:** Having the same frequency range as alpha waves, mu waves are prominent in the sensorimotor and motor cortex, areas of the brain which are separated by the central sulcus, otherwise known as the Rolandic fissure. Their amplitude increases during muscular relaxation and decreases during muscular movements or imaginary movements. Their amplitudes are approximately $\sim 10\mu\text{V}$.

- **Beta Waves [14 – 30Hz]:** Seldom exceeding an amplitude of $30\mu\text{V}$, beta waves are encountered mainly over the frontal and central regions of the brain. Beta activity on the central region can be motor activity since it's related to the mu rhythm. Finally, during the entire maturational period, beta activity is relatively more prominent in females than in males.

- **Gamma Waves [30Hz]:** Gamma waves are fast oscillations found during conscious perception. Due to their small amplitudes and high contamination from muscle artifacts their importance was overlooked until recently. Research studies suggest that gamma activity is involved in attention, working memory and long-term memory processes.

In addition to brainwaves, some specific EEG signals occur due to stimuli being presented to the user. They include steady-state visual-evoked potentials (SSVEPs) and various event related potentials (ERPs).

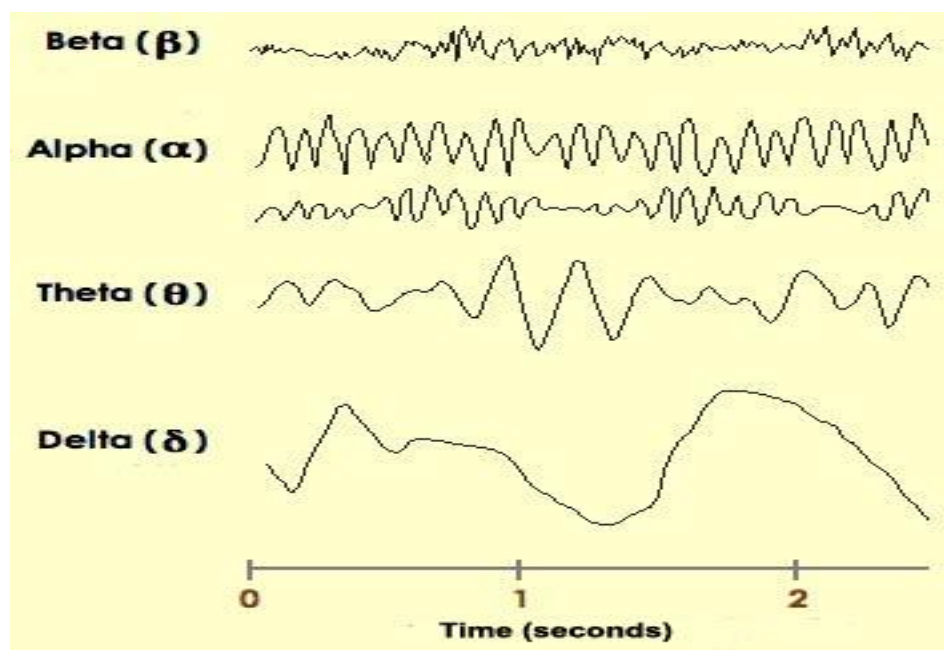


Figure 2. Different types of brainwaves and their waveforms

2.2 EEG Recording Principles

A good starting question about EEG recording is: Where should the electrodes be placed on the scalp? This may seem like a simple question, but there a lot of implications resulting from poor electrode positioning or in different configurations between subjects. The most important drawback is that measurements from different subjects are hard to compare, since humans have different head shapes and sizes. For example, a hydrocephalic subject would have totally inconsistent measurements in relation to a healthy subject. It is therefore apparent that relative positioning has to be utilized.

The answer to the above problems is the 10-20 system or International 10-20 system. Although there are other configurations (10-10 system or configurations based on the application at hand), the 10-20 system is the primary configuration system used in EEG recording. The electrode positions are shown in Figure 3.

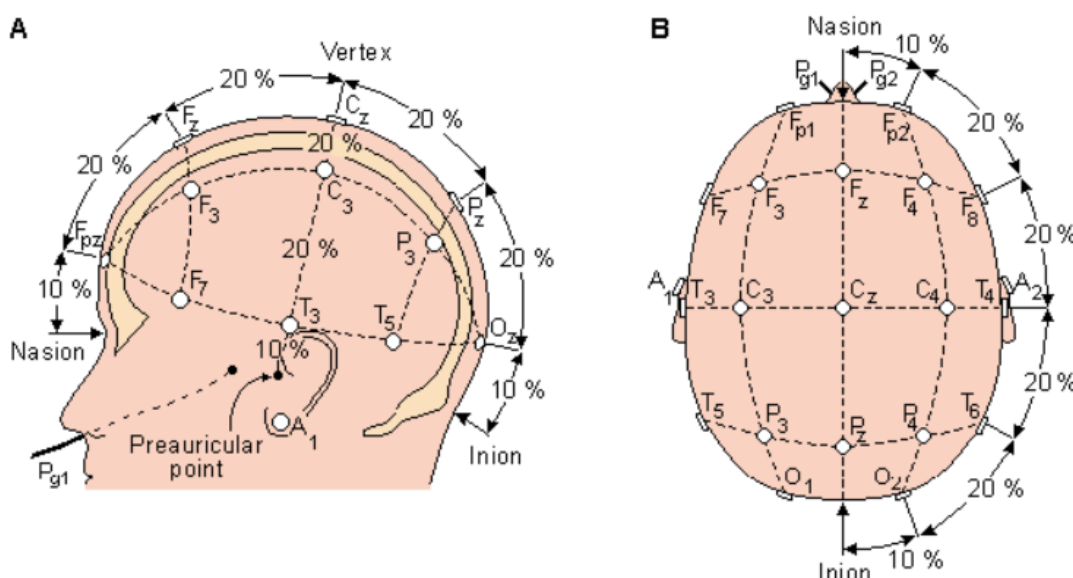


Figure 3. International 10 - 20 system with relative positions distances

The advantage of the 10-20 system lies in that the electrode positions are determined by features present in all humans, namely the inion, nasion and the left and right preauriculars. As seen in the figure, odd numbers refer to the left side of the head, while even numbers refer to the right side. 'A1' and 'A2' refer to the left and right earlobes. The 'z' positions refer to the midline of the head. Finally, 'Fp', 'F', 'T', 'C', 'P' and 'O' refer to the prefrontal, frontal, temporal, central, parietal and occipital areas of the brain, correspondingly [13].

There are certainly a lot of requirements and restrictions in EEG recordings, because of their small amplitudes and susceptibility to noise. For example, within electrified buildings, the 50 or 60 Hz electric field on human bodies can be 10 times larger than the EEG signal itself. Moreover, other biophysical signals, such as ECG and EMG occur in the same frequencies as EEG signals and they are also considered as noise. In addition, the impedance of the scalp diminishes the amplitude of the recorded signal.

Filtering plays a very significant role in overcoming many of the obstacles mentioned above. For the 50/60 Hz noise, a notch filter can be used to suppress that frequency. Moreover, bandpass filtering is standard in EEG applications in order to drop unwanted frequencies. Standard filtering settings for routine EEG are: 1 Hz for the low frequency filter and 50-70 for the high frequency filter. For the reduction of the scalp impedance, conducting gel or paste is placed between the scalp and each electrode.

2.3 EEG Hardware

As mentioned above, EEG signals are acquired through the placement of electrodes on the head. But what about the hardware that is used to acquire this tiny signal (microvolt scale) in a noisy environment? The integral components of the EEG hardware are the amplifier and the filter.

The amplifiers used in electrophysiologic signals are instrumentation amplifiers. A typical instrumentation amplifier setup is shown in Figure 4. Their advantage over conventional amplifiers is the rejection of signals that are identical in both inputs, which are called common-mode signals and the amplification of differential-mode signals. Their usefulness will be demonstrated in the following example.

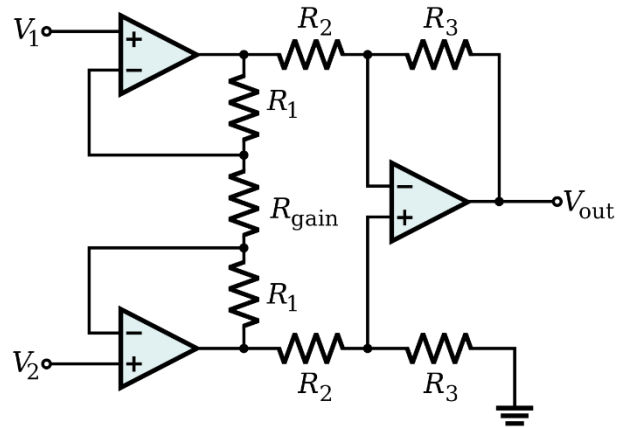


Figure 4. Typical instrumentation amplifier

Suppose we acquire a signal between the O1 and A2 points in the 10-20 system. It is well known that the earlobe is the most electrically ‘neutral’

part of the head. This means that the only signal that will be present in the earlobe will be the noise from outer sources. Of course, this noise will be also present in the O1 region, so this common-mode signal will be rejected by the amplifier, resulting in the suppression of noise. Plus, the true EEG signal will be amplified. The relationship of the common-mode and the differential-mode gains is called common mode rejection ratio (CMRR). The higher this value is, the lower will the common-mode signal be present in the output of the amplifier. It can be defined as:

$$CMRR = 20 \log_{10} \left(\frac{A_d}{|A_{cm}|} \right) dB$$

Where A_d is the differential-mode gain and A_{cm} is the common-mode gain. Finally, filtering the signal is necessary to keep only the wanted frequencies; in this case EEG frequencies [6].

With digital EEG, after the signal is acquired, it is converted to a digital signal by an ADC (Analog to Digital Converter) and is stored or displayed in order to be processed or evaluated further.

3 CHAPTER 2 : Machine Learning

Traditional computer programs have their logic and function dictated a priori by well-defined commands, which are created by the programmers who made them. While this approach is satisfactory for most cases, the advancement of modern technology has paved the way for new applications where computers can operate in innovative ways by utilizing algorithms that mimic the functions or behaviors of complex biological systems.

Machine learning is concerned with the question of how to construct computer programs that automatically improve with experience. By using sample data from the desired application, a model is built which makes predictions without being explicitly programmed to do so. Its advantage over conventional methods lies in the fact that some problems don't have a specific solution or it is very hard to define one. Examples of such applications include computer vision, email classification, speech recognition, natural language processing and many others. The process in which a model is trained based on sample data is called learning.

There are mainly three categories of learning that take place when training a machine learning model.

- **Supervised learning** is the process in which the model is learning a function that maps the input to an output based on example input-output pairs. The aim of this method is for the model to be able to generalize from the training data to unseen data. Supervised training is used for classification and regression problems.
- **Unsupervised learning**, in contrast to supervised learning, doesn't use labeled data to train the model. Instead, provision is made for a task-independent measure of the quality of representation that the model is required to learn and the free parameters of the network are optimized with respect to that measure.
- **Reinforcement Learning**, in which the learning of an input-output mapping is performed through continued interaction with the environment in order to minimize a scalar index of performance.

3.1 Data and Feature Extraction

3.1.1 Data and Datasets

In order to expand on feature extraction techniques, we first have to define shortly what data is, how we can collect data and talk about datasets. **Data** represent information. One person's height or weight can be defined as data for that person. Data can be stored in many formats, but since the rise of computing, most of the data in our world is stored in digital format. A collection of data is called a dataset. In the height and weight example, a collection of these characteristics from many people could be regarded as a **dataset**. Datasets are big tables of data where every row represents an instance or observation of an event, and every column different information about that instance. In the height and weight example every row would represent a person and there would be a total of 2 columns for weight and height.

3.1.2 Features and Feature Extraction

In machine learning terminology, every row of a dataset could be described as a *feature vector* and its constituent elements are called *features*. Most of the times these features are used to train classifiers so as to be able to make future predictions based on the previous observations contained in the dataset. In the weight and height example, a model could be used to determine a person's sex based on these features. In order to achieve this, there would need to be another column in the dataset which is called the *target* column. Targets refer to what the actual class of the feature vector is. A trained model should be able to define the targets correctly based on the input it would get.

It is important to note that in most of the times in real life, the useful information that is stored in the dataset is not readily available and it has to be extracted from it. Moreover, some data that is stored in the dataset may not be useful for our specific goal or even be redundant. The process of receiving the desired information for our task is called feature extraction. *Feature extraction* involves the use of algorithms to transform the data into some useful format, or retain only the useful characteristics for our designated task. Since the main purpose of this work is EEG signal classification, there will be a focus on feature extraction techniques for EEG signals. Some of the most widely used techniques will be mentioned later in this work.

3.2 Neural Networks [7]

In its most general form, an Artificial Neural Network (ANN), or simply Neural Network (NN) is a machine that is designed to model the way in which the brain performs a particular task or function of interest; the network is usually implemented by using electronic components or is simulated in software on a digital computer. Neural networks perform useful computations through a process of learning. To achieve good performance, neural networks employ a massive interconnection of simple computing cells referred to as “neurons” or “processing units”. Thus, a definition of a neural network viewed as an adaptive machine can be the following:

A neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a *learning* process.
2. Interneuron connection strengths, known as *synaptic weights*, are used to store the acquired knowledge.

The fundamental processing unit of a neural network is the neuron, inspired by the biological neurons that make up the complex networks in our brains. Below, in Figure 5, a non-linear model of a neuron is shown. This model was introduced by Warren McCulloch and Walter Pitts in 1943.

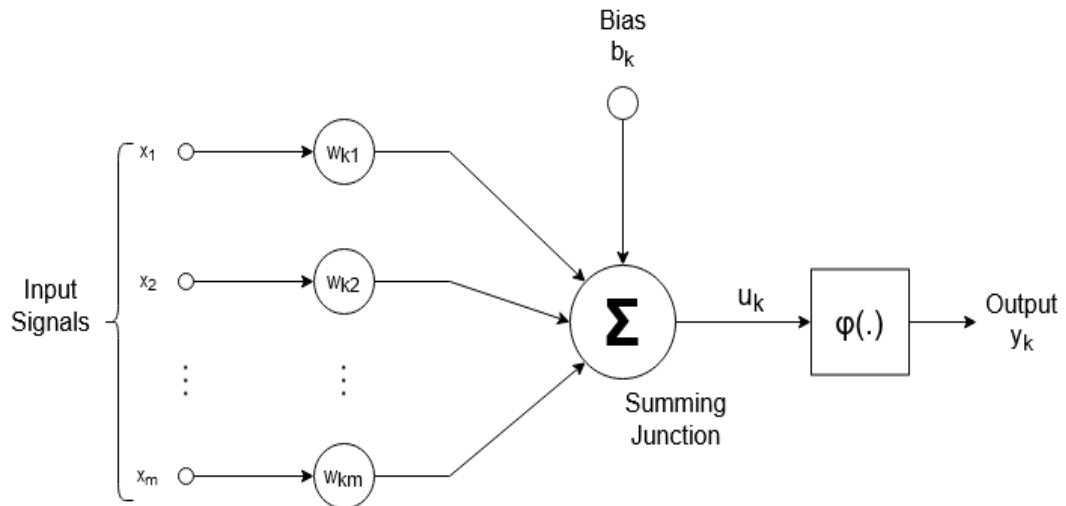


Figure 5. Non-linear model of a neuron

As it's evident from the picture above, the fundamental building blocks of this neuronal model are the following:

- **Synapses**, which are characterized by their respective synaptic weights. A signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj} . Unlike the weight of a synapse in the brain, weights in artificial neurons take positive and negative values.
- **Adder**, which sums the input signals, weighted by the respective synaptic strengths of the neuron.
- **Activation function**, which limits the amplitude at the output of a neuron. There are many activation functions used in neural networks, some of which are:
 - **Sigmoid function**, which limits the output in the range $[0, 1]$. The most common example of sigmoid functions is the logistic function, which is characterized from the equation, and is depicted in Figure 6 below:

$$s(x) = \frac{1}{1 + e^{-x}}$$

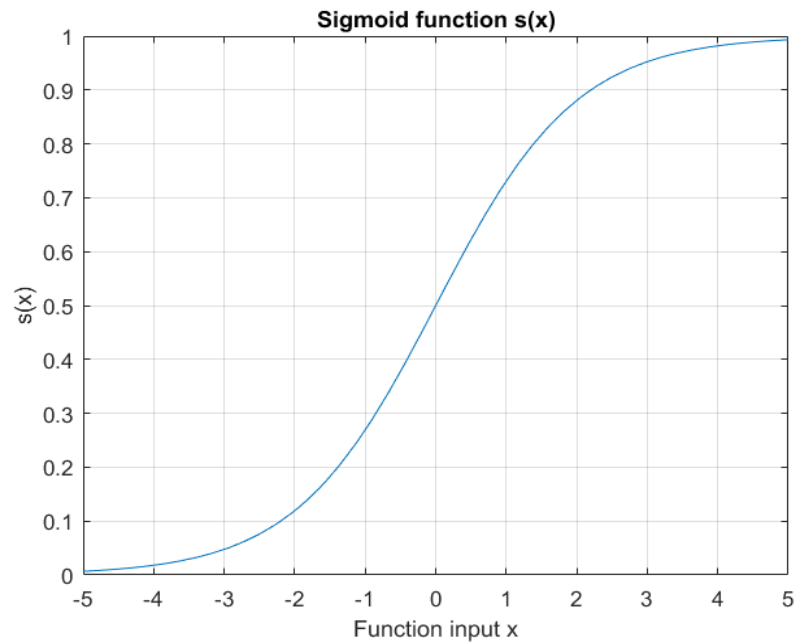


Figure 6. Sigmoid activation function

- **Hyperbolic tangent function**, which limits the output in the range $[-1,1]$. Its advantage is that negative inputs will be mapped strongly negative and that zero inputs will be mapped near zero. It's mainly used in classification between two classes. Figure 7 shows the graph of the hyperbolic tangent function:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

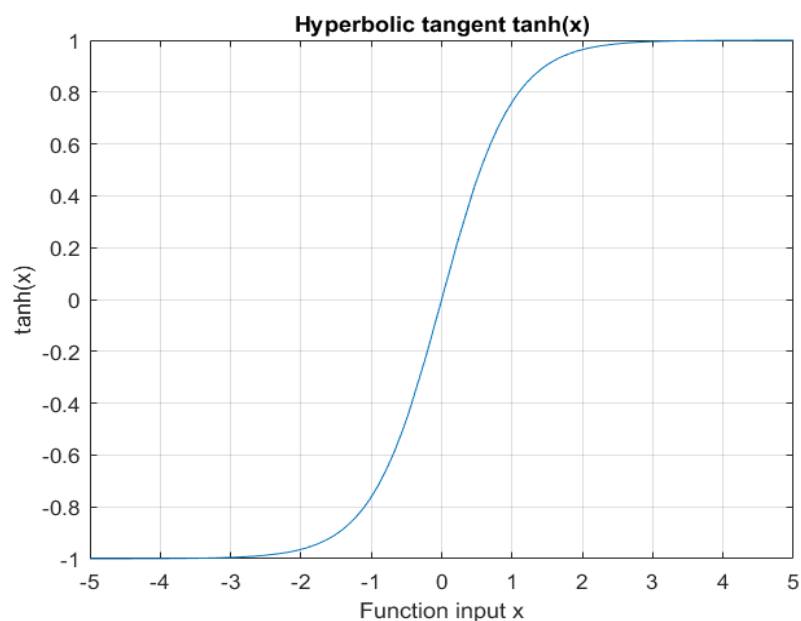


Figure 7. Hyperbolic tangent activation function

- **Rectified Linear Unit (ReLU) function**, which bounds the input only for negative numbers. The range of the function output is $[0, \infty]$. The ReLU is one of the most used activation functions in the world at the moment, primarily in convolutional neural networks and deep learning. The equation and graphical depiction for the ReLU are shown below (Figure 8).

$$ReLU(x) = \max(0, x)$$

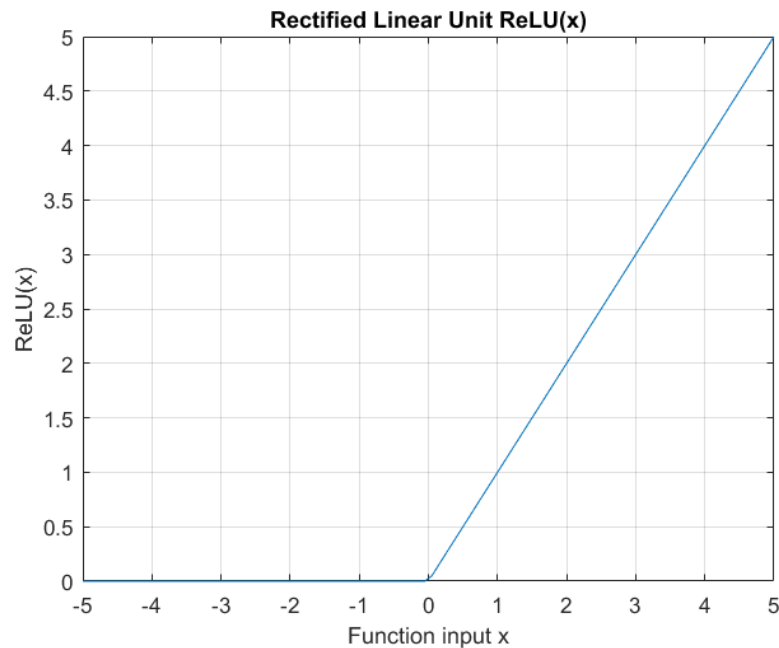


Figure 8. Rectified Linear Unit (ReLU) function

- **Bias**, which is denoted by b_k . The bias has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively.

Mathematically, the neuron k can be successfully described by the pair of equations shown below:

$$u_k = \sum_{j=1}^m w_{kj} x_j$$

$$y_k = \varphi(u_k + b_k)$$

where x_1, x_2, \dots, x_m are the input signals, $w_{k1}, w_{k2}, \dots, w_{km}$ are the respective synaptic weights of neuron k , u_k is the output of the summer and y_k is the output signal of the neuron.

3.2.1 Perceptron

The most famous algorithm for training a neuron is called the *perceptron*. It was proposed in 1958 by Frank Rosenblatt, an American psychologist. It is based on the McCulloch – Pitts model, which was described above. A summary of its convergence algorithm is shown below:

- $x(n) = \text{input vector } (m + 1) \times 1 = [+1, x_1(n), x_2(n), \dots, x_m(n)]^T$
- $w(n) = \text{weight vector } (m + 1) \times 1 = [b, w_1(n), w_2(n), \dots, x_m(n)]^T$
- $b = \text{bias}$
- $y(n) = \text{real output (quantized)}$
- $d(n) = \text{desired output}$
- $\eta = \text{learning parameter, } \eta \in (0, 1]$

1. Initialization. We start by initializing the weights to a value; 0 is most commonly used. Afterwards, we compute the following for a step $n=1, 2, \dots$

2. Activation. In the n time step, activate the perceptron by applying the input vector $x(n)$ and the desired output $d(n)$.

3. Calculation of the real output. Compute the real output of the perceptron according to:

$$y(n) = \text{sgn}[w^T(n) \cdot x(n)]$$

where $\text{sgn}(\cdot)$ is the sign function.

4. Weight vector update. Update the weight vector of the perceptron according to the following:

$$w(n + 1) = w(n) + \eta \cdot [d(n) - y(n)] \cdot x(n)$$

where

$$d(n) = \begin{cases} +1 & \text{if } x(n) \text{ belongs to class 1} \\ -1 & \text{if } x(n) \text{ belongs to class 2} \end{cases}$$

5. Continuation. Increment the timestep n by 1 and go to step 2.

It is very important to note that a perceptron can **only** classify linearly separable patterns successfully.

We can introduce a cost function to the above algorithm which allows for search based on a gradient vector. Specifically, we define the perceptron cost function as follows:

$$J(w) = \sum_{x \in \mathfrak{X}} (-w^T \cdot x)$$

where X is the total of samples x which are classified wrongly by a perceptron which uses w as its weight vector. If all samples are classified correctly, then the set X is an empty set. The main characteristic of this cost function is that it is differentiable in respect to the weight vector. Consequently, differentiating the function $J(w)$ with respect to w , we get the gradient vector:

$$\nabla J(w) = \sum_{x \in \tilde{x}} (-x)$$

where the gradient

$$\nabla = \left[\frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \dots, \frac{\partial}{\partial w_m} \right]^T$$

In the steepest descent method, the weight adjustment in every timestep is applied in the opposite direction with respect to the gradient vector $\nabla J(w)$. Consequently, the algorithm assumes the form:

$$w(n+1) = w(n) - \eta(n) \cdot \nabla J(w) = w(n) + \eta(n) \sum_{x \in \tilde{x}} x$$

The above algorithm can be described as a batch algorithm because for each timestep, a group of falsely classified samples is used for the calculation of the adjustment.

3.3 Multilayer Perceptrons

After having laid the foundations of the perceptron, it's time to move on to more complex structures of networks, called Multilayer Perceptrons (MLPs). MLPs have 3 basic characteristics:

1. Every neuron model contains a non-linear, differentiable activation function.
2. The network has one or more layers which remain 'hidden' from the input and output nodes.
3. The network displays high connectivity, the extent of which is determined from the synaptic weights of the network.

Due to the high complexity that arises from these characteristics, theoretical analysis of the network can prove to be an arduous task. This in turn leads to an increased difficulty in the learning process of the network.

As a historical note, it is important to state that the shift from a single layer perceptron to a multilayer perceptron had been an ongoing problem for AI research scientists in the 20th century since their inception from Rosenblatt. The most important factor that stifled the advancement of multilayer perceptrons was the inability of scientists to figure out a way to train those networks reliably. Moreover, the book *Perceptrons* by Marvin Minsky and Seymour Papert, published in 1969 dealt a devastating blow to the flourishing AI research at the time, by criticizing the capabilities of neural networks and thus, interest and funding in AI research rapidly declined until the early 1990s. This period of time came to be known as the AI winter, gaining its name from the term Nuclear winter.

One of the major breakthroughs that led to the rise of popularity of neural networks was the development of the backpropagation algorithm during the 1980s. Backpropagation involves two phases, the feedforward phase and the backpropagation phase. In order to understand these two phases, we have to introduce two types of signals that occur into an MLP.

- **Function signals**, which are fed on the network input, and propagate through the network, producing an output.
- **Error signals**, which originate from an output neuron and propagate backwards, layer by layer.

Every hidden or output neuron of an MLP is designed to form two computations:

- The computation of the function signal which appears on the output of each neuron, which is expressed as a linear non-continuous function of the input signal and the synaptic weights related with that neuron.
- The computation of an approximation of the gradient vector, which is needed for the backpropagation phase.

Below, the mathematical expressions for the analysis of the BK algorithm will be formulated. The basis for this will be the signal flow graph of a Multilayer Perceptron depicted on Figure 9 below.

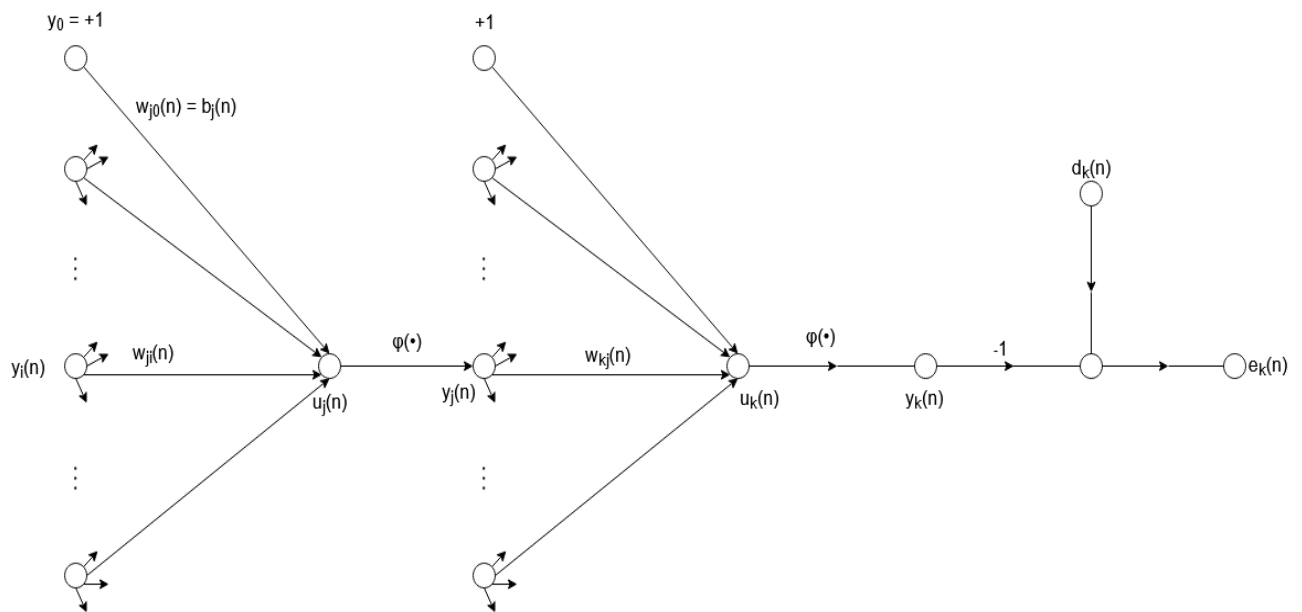


Figure 9. Signal flow graph of a multilayer perceptron

Here, the neuron j is fed by a set of function signals which originate from a neuron layer to its left. Consequently, the local field produced at the input of the activation function relative to the neuron j is:

$$u_j = \sum_{i=0}^m w_{ji}(n) y_{ji}(n)$$

where m is the total number of inputs (not including the bias) which are applied to neuron j . The function signal at the output of neuron j is:

$$y_j(n) = \varphi_j(u_j(n))$$

The BK algorithm performs a correction on the synaptic weight $w_{ji}(n)$, $\Delta w_{ji}(n)$, which is proportional to the partial derivative (using the chain rule):

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial u_j(n)} \frac{\partial u_j(n)}{\partial w_{ji}(n)}$$

$\mathcal{E}(n)$ is called **total instantaneous error energy** and is basically a cost function, which means that by minimizing it, the output error will be minimized too, leading to increased performance of the network. $\mathcal{E}(n)$ is defined as:

$$\mathcal{E}(n) = \frac{1}{2} \sum_{k \in C} e_k^2$$

The set C includes all the neurons in the output layer. The partial derivative $\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$ represents a sensitivity factor, which determines the search direction in the weight space for the weight w_{ji} . In simpler terms, we want to adjust the weights in a way that minimizes the cost function. Next, the error $e_j(n)$ is defined as:

$$e_j(n) = d_j(n) - y_j(n)$$

Thus

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1$$

The function signal $y_j(n)$ on the output of a neuron j is defined as:

$$y_j(n) = \varphi_j(u_j(n))$$

Where $\varphi_j(n)$ is the activation function and $u_j(n)$ is the local field which is fed in the activation function, defined as:

$$u_j(n) = \sum_{i=0}^m w_{ji}(n) \cdot y_i(n)$$

This formula is almost identical to the single layer perceptron, but instead of $x(n)$, $y(n)$ is used to note that the output of the previous layer acts as an input to the next layer. By using the above, we know that:

$$\frac{\partial y_j(n)}{\partial u_j(n)} = \varphi'_j(u_j(n))$$

and

$$\frac{\partial u_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

Finally, by taking all the above formulas into account, we can derive the following:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -e_j(n) \cdot \varphi'_j(u_j(n)) \cdot y_i(n) \quad (1)$$

The correction $\Delta w_{ji}(n)$ which is applied into $w_{ji}(n)$ is defined by the delta rule:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} \quad (2)$$

Where η is the learning rate of the algorithm. The negative sign is used to indicate a gradient descent into the weight space (i.e., a direction which will minimize the cost function). The local gradient can be defined as:

$$\delta_j(n) = \frac{\partial \mathcal{E}(n)}{\partial u_j(n)} = e_j(n) \cdot \varphi'_j(u_j(n)) \quad (3)$$

Through formulas (1) and (3), formula (2) can be rewritten as:

$$\Delta w_{ji}(n) = \eta \cdot \delta_j(n) \cdot y_i(n)$$

Depending if the neuron is a hidden or an output neuron, we get the following:

$$\delta_j(n) = e_j(n) \cdot \varphi'_j(u_j(n)), \text{ if neuron } j \text{ is an output neuron}$$

$$\delta_j(n) = \varphi'_j(u_j(n)) \sum_k \delta_k(n) w_{kj}(n), \text{ if neuron } j \text{ is a hidden neuron}$$

Now that the above relationships were established, the two phases of the backpropagation algorithm can be explained.

- In the **feedforward phase**, the synaptic weights do not change throughout the whole network and the function signals of the network are calculated for every neuron layer. When the output layer is reached, the error is calculated based on the desired and the real output of the network.

- The **backpropagation phase** starts at the output layer, sending the error signals to the left in all of the network layers and layer by layer, calculating the local gradient recursively. During this phase, the synaptic weights are also adjusted accordingly.

Training can be terminated when one or more stopping conditions are met. Below an overview of the backpropagation algorithm will be listed.

- 1. Initialization.** The synaptic weights are chosen from a uniform distribution
- 2. Training examples presentation.** An epoch of training examples is shown to the network.
- 3. Forward computation.** The local fields and function signals of the network are computed, layer by layer. Then, the error is calculated at the output layer for each neuron.
- 4. Backward computation.** The local gradients of the network are computed, which are defined as:

$$\delta_j^{(l)}(n) \begin{cases} e_j^{(L)}(n) \cdot \varphi_j'(u_j^{(L)}(n)), & \text{for the neuron } j \text{ in the output layer } L \\ \varphi_j'(u_j^{(l)}(n)) \cdot \sum_k \delta_k^{(l+1)}(n) \cdot w_{kj}^{(l+1)}(n), & \text{for the neuron } j \text{ in the hidden layer } l \end{cases}$$

the weights are adjusted according to the delta rule

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \eta \cdot \delta_j^{(l)}(n) \cdot y_i^{(l-1)}(n)$$
- 5. Iteration.** Repeat steps 3 and 4, presenting new training examples to the network, until the stopping criterion is satisfied.

3.4 Variations and Applications of Neural Networks

Since research on neural networks is ongoing, a plethora of variations have been developed over the years, which target different applications.

Radial Basis Function (RBF) Networks, which in their simplest forms consist of three layers, the input layer the output layer and the hidden layers. The activation functions of the hidden layers are Gaussian functions. RBF networks are distinguished from other neural networks due to their universal approximation and faster learning speed. They are mainly used for function approximation and systems control.

Recurrent Neural Networks (RNNs), are a type of neural networks which use sequential data. Their main difference over feedforward neural networks is that the outputs of neurons are fed back to the neurons of the previous layer. They are mainly used in the fields of Natural Language Processing (NLP) and speech recognition.

Convolutional Neural Networks (CNNs), feature the convolution and the pooling layers. The convolution uses filters that perform convolution operations as it is scanning the input. The resulting output is called the feature map. The pooling layer is a downsampling operation, typically applied after a convolution layer, which does some spatial invariance. They are mainly used for image classification and object recognition.

Generative Adversarial Networks (GANs), are composed of a generative and a discriminative model, where the generative model aims at generating the most truthful output that will be fed into the discriminative which aims at differentiating the generated and true image. They are mainly used in image and music generation and synthesis and text-to-image translation.

4 CHAPTER 3 : Brain-Computer Interfaces

A Brain-Computer Interface (BCI) is a system that enables communication between brain and machines. A BCI, in order to perform its purposes, records brain signals, interprets them and produces corresponding commands to a connected machine [8]. BCI technology is used in various applications, such as security and authentication, education, neuromarketing and advertisement, games and entertainment and several medical applications, such as cognitive neuroscience, brain-related prevention and diagnosis of health problems, rehabilitation and restoration [9 – 12].

A BCI provides an interconnection platform that supports the full duplex communication between the brain and an external device. According to the way that BCIs use to set up the brain-device interconnection, they are classified as non-invasive or invasive. Non-invasive BCIs use electrodes placed on the scalp. They are easy and safe to use, low-cost, portable, and offer a relatively high temporal resolution. Invasive BCIs use electrodes implanted in the interior of the scalp. Comparatively to non-invasive BCIs, they offer higher values of amplitude, spatial resolution and resistance to noise. However, they require neurosurgery operations and they are both unsafe and expensive. Furthermore, scar tissue decreases the quality of signals received. Practically, non-invasive BCIs offer are used more often.

There are various non-invasive methodologies used in BCI technology, such as Positron Emission Tomography (PET), functional Magnetic Resonance Imaging (fMRI) and Near-Infrared Spectroscopy (NIRS), which study changes made in blood flow, magnetoencephalography (MEG), which monitors the magnetic action of the brain, and EEG, which records the electric activity of the brain. Both NIRS and fMRI BCIs offer high spatial and temporal resolution, but poor temporal resolution. Moreover, MEG and PET BCIs offer high spatial and temporal resolution. However, PET BCIs require the inoculation of a radioactive constituent into the bloodstream. Furthermore, both fMRI and MEG methods rely on the use of equipment which is not only costly, but also huge. EEG BCIs are by far the most popular type, because, despite their relatively poor spatial resolution, they have high temporal resolution, low-cost and easy installation [9].

Moreover, BCIs are classified as either exogenous or endogenous, according to the nature of input signals. Exogenous BCIs analyze the brain activity created due to external stimuli. They are easy to set up and offer high bit rates, but they need the continuous response of the user to outward incitements which may be either tiring, or even unfeasible. Endogenous BCIs use self-regulation of brainwaves without external stimuli. They provide lower data transfer rates but they can be operated via free self-control even by users with sensory organs affected or suffering from motor neuron diseases [13].

Similarly, BCI systems are classified, according to the method used for input data processing, as synchronous or asynchronous. Synchronous BCIs analyze the brain signals only after a specific prompt and during predefined time intervals. Thus, the overall process is better organized and the user is free to make any kind of movements, which would produce artifacts, when brain signals are not observed. They also require minimal training and have stable performance and high accuracy. Asynchronous BCIs inspect brain signals successively, thus letting the user act at free will. Therefore, they offer more natural human-machine interaction. However, they are more complex in design and evaluation and require extensive training. Moreover, their performance may vary between users, and their accuracy is not very high [13].

4.1 Brainwaves for EEG-BCIs

The most commonly used types of brain waveforms to develop EEG-based BCIs are P300, SSVEP, ErrP, ERD/ERS and alpha brainwaves [14].

P300 is an event-related positive potential deflection which is caused by the reaction to a desired external stimulus of visual, auditory, or tactile modality. P300 waveforms are typically measured, with a latency of roughly 250 to 500 ms between stimulus and response, by using electrodes located over the parietal lobe of the scalp.

Steady state visually evoked potentials (SSVEP) are brain waveforms of exogenous type that are generated as responses to visual stimulation at specific frequencies ranging from 3.5 Hz to 75 Hz. Considering that SSVEP signals often have their highest values at medial occipital electrode sites, they are supposed to originate mostly from the primary visual cortex.

Event-related desynchronization and event-related synchronization (ERD/ERS) waves are endogenous brain signals, which are generated when performing mental tasks, such as motor imagery or mental arithmetic. They can be measured at different cortical locations.

Error-related potential (ErrP) waveforms are brain signals which are activated every time that a subject identifies the commitment of an error which has been made either by himself/herself or by another individual during various choice tasks. Waves of this kind can be captured by applying electrodes on various brain regions including the anterior cingulate cortex, anterior insula, inferior parietal lobe, and intraparietal sulcus, as well as other regions of the cortex, subcortex and cerebellum.

Alpha brainwaves are brain signals which have their amplitude increased whenever the eyes of an individual are closed during wakeful relaxation. In contrast, the amplitude of alpha waveforms is diminished for the duration of sleepiness and sleep and also when having eyes opened while mental effort is performed. This phenomenon is usually referred to as alpha rhythm blocking. Alpha brain waveforms can be monitored by applying a number of electrodes on both sides of the posterior segments of the scalp, where the occipital lobe, which is the center of visual processing activities in the brain, is positioned.

4.2 BCI Operation

The operation of a typical BCI system is based on the sequential execution of a number of procedures, which namely are signal acquisition, preprocessing, feature extraction, classification, translation, and feedback to operator [13, 14], as shown in Figure 10.

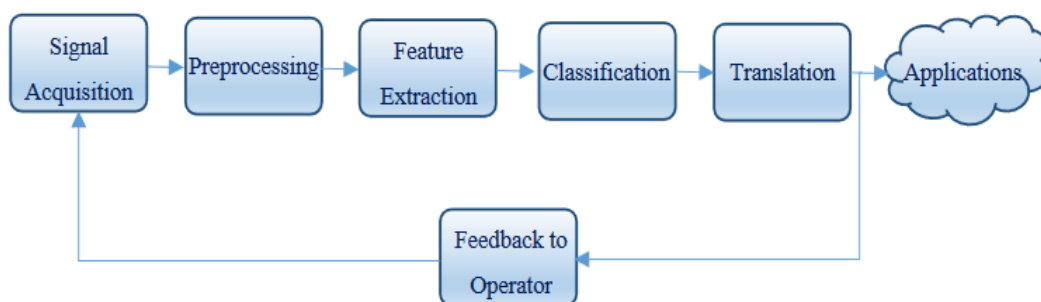


Figure 10. Block diagram representing the processes performed in a typical Brain-Computer Interface

In EEG-BCIs, **signal acquisition** is performed by using electrodes which are positioned along the scalp of the user. Normally, the settlement of electrodes on the scalp is performed in compliance to the International 10-20 system. According to this system, electrodes are located on the scalp at 10% and 20% of a measured distance from the reference spots including nasion, inion, left and right preauricular, as mentioned earlier [13].

Preprocessing is the procedure which is carried out in order to reduce the noise from the signal and apply some filtering and other methods in order to remove artifacts which are caused by endogenous sources, such as motions of eyes, muscles and heart, and exogenous sources, such as power-line coupling and impedance mismatch [15]. Preprocessing is usually performed by using low-pass, high-pass, band-pass or notch filtering. However, the use of such filters may eliminate useful elements of EEG signals having the same frequency band as artifacts [16].

In **feature extraction**, specific features of the signals in time domain or/and frequency domain that can expressively differentiate specific classes are extracted and positioned into a feature vector in order to enable the classification phase which follows. Autoregressive (AR), Hjorth and EEG signal power are commonly used feature extraction techniques [17].

During the **classification** phase, a properly built algorithm is used. This algorithm distinguishes between classes which correspond to various brain activity patterns by deciding to which of these classes every feature vector suits best. Neural networks (NNs) are widely used as classifiers in BCIs because they provide the ability to approximate nonlinear decision boundaries [18, 19]. Alternatively, linear discriminant analysis (LDA), support vector machines (SVM), and statistical classifiers may be used [20]. The advantage of LDA is that it is a simple-to-use probabilistic approach based on Bayes' Rule. On the other hand, NNs have the advantage of being able to approximate nonlinear decision boundaries. In cases where a small amount of training data is available, the use of SVM is a very good choice. Finally, statistical classifiers have the ability to represent the uncertainty that is inherent in brain signals.

During the **translation** phase, the extracted signal features are converted into particular commands to the device(s) under control, through the use of dedicated translation algorithms. Specifically, these algorithms have the ability not only to adapt to the continuing variations of the signal features, but also to ensure that the complete device control range is covered by the specific signal features from the user.

Finally, in the **feedback** to operator phase, the final outcome of the overall operation of the BCI system is transferred back to the system operator, so that the performance of the system can be evaluated.

4.3 BCI-Based Robot Control

An EEG-based brain-controlled robot is a robot that uses an EEG-based BCI to receive control commands from its human operator. EEG-based brain-controlled mobile robots can support the movement of both elderly people and people who are severely disabled with destructive neuromuscular disorders, such as amyotrophic lateral sclerosis (ALS), multiple sclerosis (MS), or strokes.

There are two main classes of EEG-based brain-controlled assistive robots which namely are **brain-controlled manipulators** and **brain-controlled** mobile robots. Similarly, assistive mobile robots are classified in two categories according to their mode of operation [14].

The first category consists of assistive mobile robots which operate under *direct BCI control*. Robots of this kind are controlled exclusively via the commands that their users send to the robots controlled via BCI modules, without any additional assistance by robot intelligence elements. For this reason, they are less expensive and complex to develop and their users keep the absolute motion control.

On the other hand, the overall performance of these brain-controlled mobile robots mainly depends on the performance of the BCIs, which in many cases may have inadequate speed of response and accuracy. Furthermore, the demand for continuous production of motor control commands by the users may be extremely tiring for them.

The initial example of a robot of this kind was presented in [21] where the left and right turning movements of a robotic wheelchair were directly controlled by corresponding motion commands translated from user signals.

Similarly, in [22] a brain-controlled mobile robot was able to perform forward, left, and right motions by using a BCI based on motor imagery.

Moreover, in [23] the motion control of a wheelchair is performed via a BCI, which captures alpha brainwaves. Specifically, a set of icons corresponding to predefined commands are sequentially displayed on a screen and the user is able to select the desired command by closing his/her eyes as soon as its corresponding icon appears on the display unit.

The second category consists of assistive mobile robots which operate under shared control. In the robots of this category the control is performed by combining a BCI system along with an intelligent controller, such as an autonomous navigation system. Due to their enhanced intelligence, robots of this type are safer and less tiring for their users and more accurate in interpreting and executing their commands. On the other hand, their development is of higher cost and computational complexity.

A typical example of shared control in assistive mobile robots is proposed in [24]. In this system, the operator, by using a SSVEP BCI system, has the ability to send commands in order to move a robotic wheelchair in four directions (forwards, backwards, left and right), while an autonomous navigation system executes the delivered commands.

Similarly, in [25], by using a P300 BCI, the operator uses a list of predefined locations in order to select the desired location and sends this selection to an autonomous navigation system, which guides a robotic wheelchair to the selected location. The limitation of the specific system is that it is able to be operated only in a known environment.

Likewise, in [26] shared control is used. Specifically, the combined use of a P300 BCI alongside with an autonomous navigation system is proposed in order to perform the motion control of a robotic wheelchair in an environment which is unknown. Moreover, the user has the ability to make the wheelchair turn either left or right by focusing correspondingly on one of two relative icons at a predefined visual display.

In [27] three mental tasks, which namely are the imagination of right- or left-hand movements and the generation of words beginning with the same random letter, were used in a BCI system applied to a robotic wheelchair. The system developed, which interacts with the user by using a PDA screen and speakers, is able to guide the robotic wheelchair both in known and unknown environments.

5 CHAPTER 4 : Robot Motion Control via EEG Signals

As was mentioned in the introduction, the purpose of this work is the development of a system which performs motion control in a mobile robot in accordance to the eyes' blinking of a human operator via a synchronous and endogenous Electroencephalography-based Brain-Computer Interface, which uses alpha brain waveforms. In order to implement this system, a variety of software and hardware technologies were utilized. It is therefore important to give a brief description of all those tools and their main uses before moving on to the detailed description of this work. The technologies will be divided into *hardware* and *software* parts. Beginning with hardware, the following were used:

Hardware

- **Raspberry Pi**, is a small, yet very powerful single board computer that can also interact with its environment through the use of GPIO pins. Over the past years its popularity has risen immensely, being used in many maker projects, such as home automation, networking, homemade weather forecasting stations. It is also used in education and for few commercial purposes. It includes its own operating system called Raspberry Pi OS but can also run many other operating systems.



Figure 11. Raspberry Pi 4 (Model B)

- **Arduino UNO**, is a single board microcontroller which allows for rapid prototyping and ease of use. It's easy to program by using the Arduino language, which is similar to C, but with many built-in functions that automate a lot of procedures. The Arduino UNO is based on the ATmega328P microcontroller.

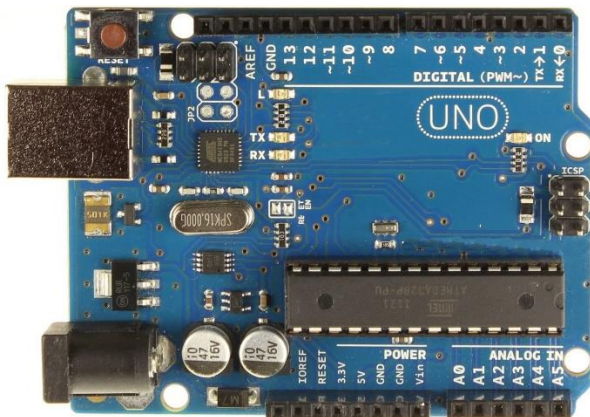


Figure 12. Arduino UNO

- **OpenBCI Ganglion**, is a high quality, affordable biosensing device [28]. It has 4 high-impedance differential inputs, a driven ground, a positive and a negative voltage supply. The inputs can be used as individual differential inputs for measuring EMG or ECG, or they can be individually connected to a reference electrode for measuring EEG. Data is sampled at 200Hz on each of the 4 channels. Gold-plated cup electrodes were also bought, which are used for EEG recording. A conductive gel has to be used in order to lower the impedance between the electrodes and the scalp. In this work, a Raspberry Pi 3 model B was used initially for the implementation, but was later replaced with then newest Raspberry Pi 4 model.

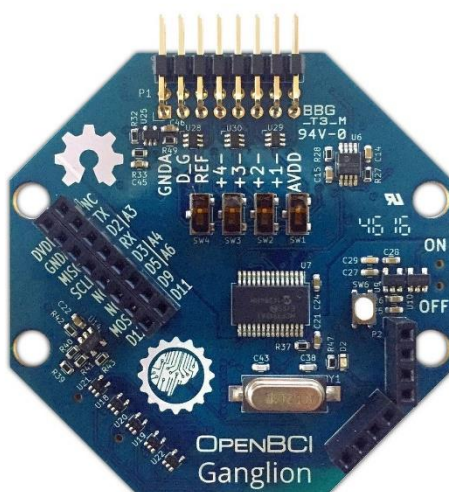


Figure 13. OpenBCI Ganglion

- **Dagu Rover 5 Chassis**, manufactured by Pololu Electronics, is a durable robot platform with caterpillar treads that let it drive over many types of surfaces and uneven terrain. A robot or motor controller is required to drive the two built-in DC motors. A unique feature of this chassis is the ability to adjust its height by changing the angles at which the gearbox assemblies are mounted on the body. The DC motors' free run current is 210 mA and their stall current is 2.4 A at 7.2 V. The maximum speed is 25 cm/s at 7.2 V.

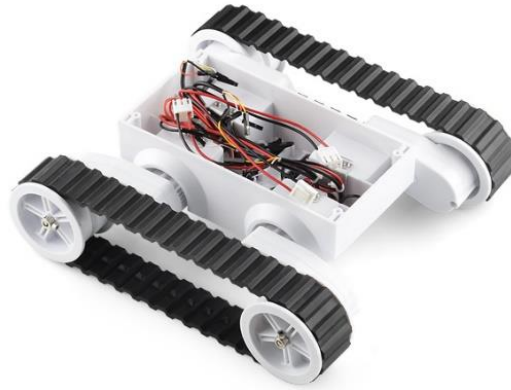


Figure 14. Dagu Rover 5 chassis

- **L298N Motor Driver**, is an H-bridge Module, which allows for control of speed and direction of two DC motors, or the control of a bipolar stepper motor. It can be used with motors of voltages between 5V and 35V. The maximum current per channel is 2A, the peak current is 3A, and a 5V regulator is included if the motor power is more than 7V.

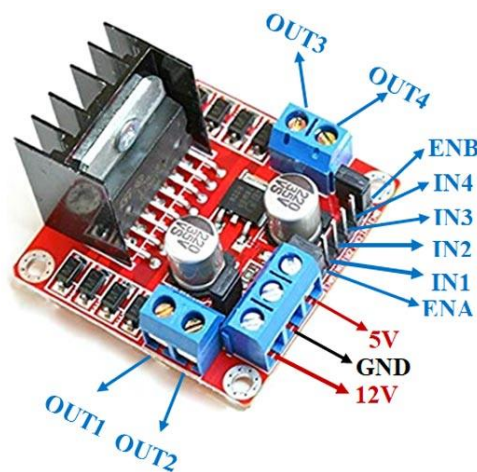


Figure 15. L298N motor driver module

- **RJ-45 Cable & HDMI Cable & USB Cable**, which are used for the connections. RJ-45 cables are used primarily in Ethernet networking. HDMI is a proprietary audio/video interface for transmitting uncompressed video and compressed or uncompressed digital audio data from an HDMI-compliant source device. Universal Serial Bus (USB) is an industry standard that establishes specifications for cables and connectors and protocols for connection, communication and power supply between computers, peripherals and other computers.

Software

- **MATLAB**, is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with other programs written in other languages. Through the use of toolboxes, the capabilities of MATLAB can be extended, providing methods for specific scientific fields, such as computer vision, machine learning, signal processing and more. These methods allow for rapid prototyping of projects. In this work, it was used for prototyping purposes and for creating datasets. This was made possible by analyzing the signals using **EEGLAB**, a toolbox for EEG signal processing and analysis.

- **Python**, is an interpreted, high-level and general-purpose programming language. It emphasizes code readability and simplicity. It is very easy to learn compared to other programming languages, but also very powerful and popular. This has led to the creation of many libraries for Python, which have extended its capabilities beyond its general scope. The libraries that were used in this work include: **NumPy**, which is used for numeric computing and utilizes matrices, allowing for very fast computations, **Matplotlib**, which is used for data visualization and plotting, **SciPy**, which is used for scientific computing and contains many modules for tasks common in science and engineering.

- **TensorFlow & Keras**. TensorFlow is an open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. It was originally developed by researchers and engineers working on the Google Brain team. It provides Python and C++ APIs. Keras is a deep learning API written in Python, running on top of TensorFlow. It was developed with a focus on enabling fast experimentation.

- **Node.js**, is an open-source, cross-platform, JavaScript runtime environment. It executes JavaScript outside of a browser. It lets developers use JavaScript to write command line tools and for server-side scripting.

- **OpenBCI GUI**, which is OpenBCI's software tool for visualizing, recording and streaming data from the OpenBCI boards. Data can be displayed in live-time, played back, saved to the computer in .txt format. It was developed with the Processing language, which is in turn based on Java. The OpenBCI GUI environment is shown in the figure below.



Figure 16. OpenBCI GUI

The implementation of this project will be broken down into parts in accordance with the block diagram that represents the processes that take place in a typical Brain-Computer Interface. The reasoning behind every technology that was used will be mentioned as well as setbacks that were encountered.

5.1 Signal Acquisition

In order to acquire the necessary EEG signal, the OpenBCI Ganglion was used along with the gold cup electrodes. As for the conductive paste, some was provided by the department of neurology of the General Oncological Hospital of Kifisia “Agioli Anargyroi”. The quantity was enough for performing initial tests but it was soon depleted. For this purpose, a custom paste was made by combining some soap powder with ultrasound gel. The paste that was created outperformed the commercial paste but would dry relatively fast. That wasn’t a problem though, because once the electrodes were placed on the scalp with the paste, the input impedance wouldn’t change despite the dryness of the paste. Moreover, the paste would make the electrodes stick on the scalp better than the commercial paste. The products that were used are shown in the figure below.

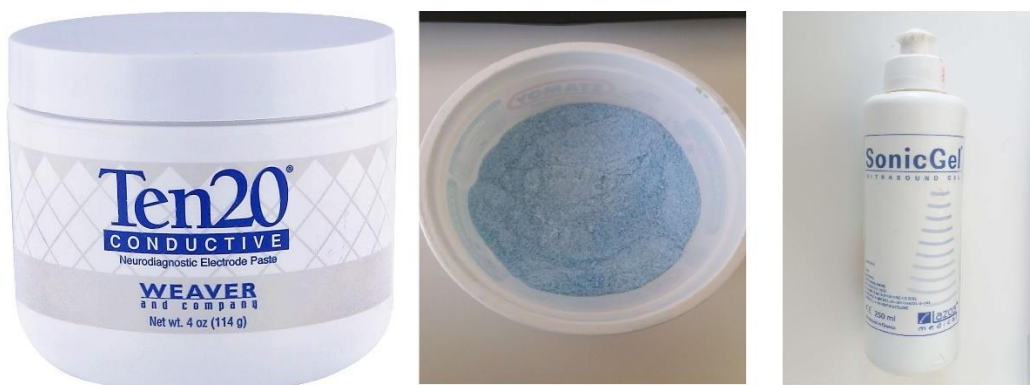


Figure 17. From left to right: Ten20 conductive paste, soap powder, ultrasound gel

Since the signal that was recorded was an EEG signal, all the inputs were individually connected to a reference electrode. Before mentioning the final configuration that was used, it is important to mention the project’s initial purpose, which was to move the robot using Mu waves, which, as it was mentioned in the introduction, originate from the motor cortex. The central idea to this is that when a subject moved his/her right arm, the robot would turn right, and when the left arm was moved, the robot would turn left. Finally, the robot would move forward when the subject would move his/her legs. For this implementation, the following configuration was adopted based on the figure below:

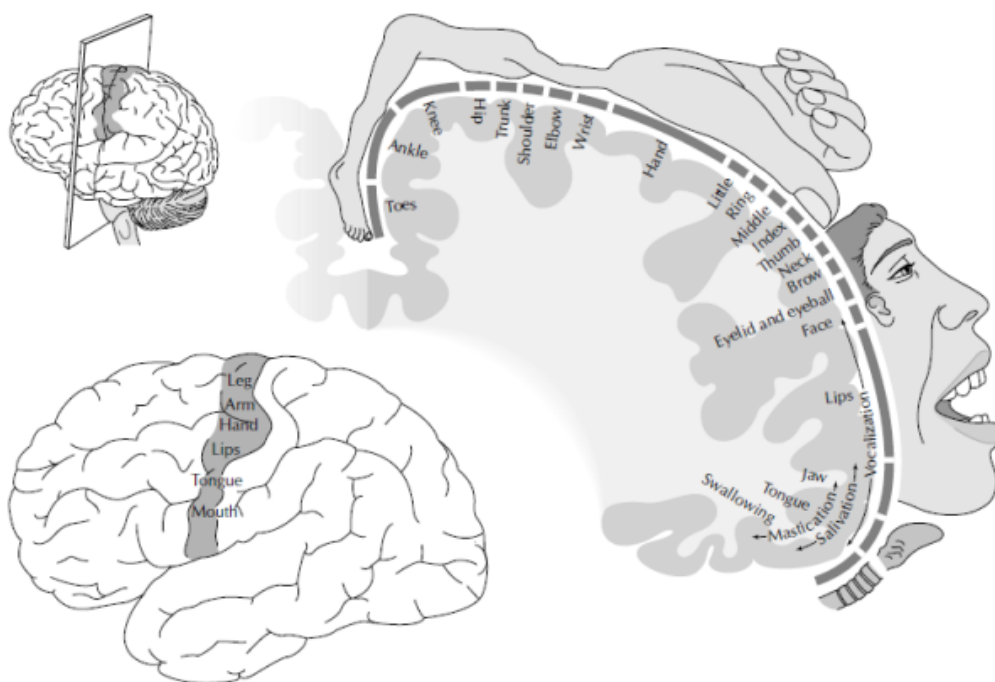


Figure 18. The motor homunculus on the precentral gyrus. The motor cortex can be seen in the lateral view of the cerebral cortex

- An electrode was placed in the ‘Cz’ region for recording Mu waves from leg movements.
- An electrode was placed in the ‘C4’ region for recording Mu waves from left arm movements.
- An electrode was placed in the ‘C3’ region for capturing Mu waves from right arm movements.

It is important to note that for the arms’ movements, these positions were chosen because the hemispheres of the cerebrum and the thalamus represent mainly the contralateral side of the body.

After extensive trials and signal processing, it was concluded that due to equipment limitations, Mu waves couldn’t be captured reliably, so alpha waves were chosen instead to be the brain control signal. As it was mentioned before, alpha waves are the prominent EEG wave pattern in awake adults while having eyes closed in the frequency range of 8-13 Hz. Generally, EEG-BCIs based on rhythms like alpha waveforms are less sensitive to artifacts than other types due to the fact that signal monitoring is limited in thin frequency bands. For this reason, a high signal-to-noise ratio (SNR) is achieved [15]. Thus, the electrode configuration changed in the following way:

- An electrode was placed in the ‘O1’ region.
- An electrode was placed in the ‘O2’ region.

In both configurations, the reference electrode was placed on the left earlobe (A1), while the ground electrode was placed on the right earlobe (A2). The specific positions were chosen because, although alpha rhythms are also generated in other parts of the brain, they are considered to exhibit greater amplitude in the posterior part of the brain, specifically at derivations ‘O1’ and ‘O2’ [29]. As it was mentioned before, the amplitude of alpha brainwaves diminishes when subjects open their eyes. This phenomenon is called alpha blocking. By taking advantage of this, subjects can form n-bit binary sequences by opening or closing their eyes in 2-second intervals. Each bit interval is designated by an acoustic cue.



Figure 19. Electrode placement setup

Moreover, since this is a synchronous BCI, a button has to be pressed for the recording procedure to start. Increased alpha activity (eyes closed) corresponds to a binary ‘1’, while decreased binary activity (eyes open) corresponds to a binary ‘0’. As a proof of concept, 4-bit binary sequences were selected to demonstrate the effectiveness of this system. In total, 4 control signals were designated for 4 robotic movements as it can be seen in Table 1.

Table 1. Binary sequences with corresponding robotic movements

Binary Sequence	Robotic Movement
'1010'	Forward
'0101'	Reverse
'1100'	Left
'0011'	Right

Following from the electrode placement and the control signal definition, the signals had to be recorded and stored in a file in order to create a dataset. For this purpose, the OpenBCI GUI and MATLAB were used. Firstly, by using the OpenBCI GUI, the signals were visualized and tests were performed to see if the changes in eyes opening/closing would cause a significant effect that would be able to be recorded reliably. EEGLAB was used extensively with its many tools and functions for EEG analysis. After this was confirmed, the control signals that were recorded were saved in a .txt file. It is important to mention that this was the first phase of the dataset creation since the signals had to be processed further to make this happen.

The procedure mentioned in the paragraph above concerns the testing phase of the signal acquisition, which concerns the data collection in order to create the dataset. However, for the actual implementation, a different setup was used. First of all, it is important to note that the OpenBCI Ganglion transmits its data to a computer through the means of a Bluetooth connection. While the OpenBCI GUI automatically connects to the board and streams data easily, it is not suitable for a custom application. Thus, a script had to be created in order to stream the data from the board in real-time. OpenBCI offers many different SDKs in order to stream the data from the board directly at present, but at the time this project was developed, the main SDKs available were OpenBCI Python and OpenBCI Node.js.

The Python library was chosen first, since the original goal of this project was to be developed solely using Python. Sadly, this was not possible because of the many problems this library had, making it unsuitable. It has since been deprecated and replaced by another library, BrainFlow, which offers bindings for C++, Python and Java.

Since the Python library was not usable, the only remaining choice was the Node.js library, which proved to be quite versatile. The script that was created would get the data from the board and encode them as JSON objects. Afterwards, with the use of the nanomsg binding for Node.js, the script could transmit the data through a TCP connection. This ensured that the data would be streamed reliably and more importantly, that many different scripts could get the data at the same time, which means that many different objects can be controlled with the same control signals. The Node.js script acts as the server in the TCP connection.

On the other side, a Python script acting as the client in the TCP establishes a connection between the two scripts. As soon as a connection is established, the user can start the recording process with the push of a button or conclude the program execution. As soon as the recording process begins, an acoustic cue is played every 2 seconds in order to inform the user of the start of

each recording interval. The user can then close his/her eyes accordingly, depending on what move they want to execute. Since the data is received from the two channels, ‘O1’ and ‘O2’, they are stored in 2 different NumPy arrays. Moreover, the timestamps of those samples are stored in a NumPy array. Each array contains 1604 elements. The reason for this is the sampling frequency of the board, which is 200 Hz. This means that every 2-second interval accounts for 401 samples. So, for the whole 8-seconds of recording, 1604 samples are stored.

Plotting the following signals in raw form from both regions ‘O1’ and ‘O2’ using matplotlib, we get the figure below. This signal corresponds to a ‘left’ movement. It is evident that by not filtering the signal, a conclusion on the sequence that was executed cannot be reached. Following from the signal acquisition, preprocessing and feature extraction of the acquired signals are the next steps for this project.

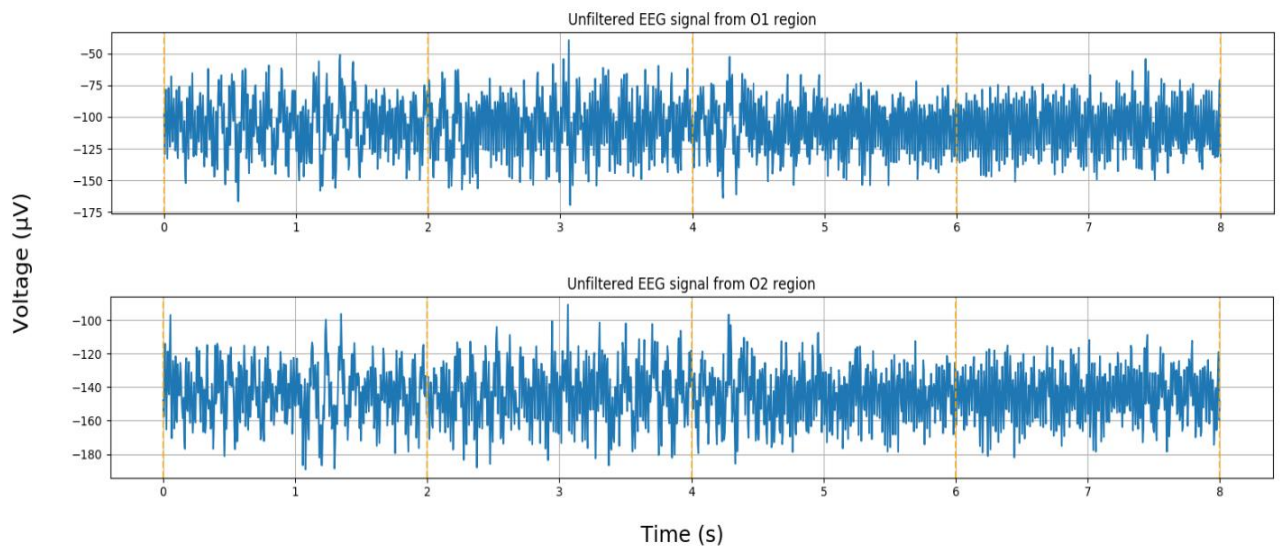


Figure 20. Raw EEG signals as recorded from the board

5.2 Preprocessing & Feature Extraction

5.2.1 Preprocessing

After the signals were recorded and stored in the NumPy arrays, the preprocessing phase began. In order to extract the desired alpha brainwaves from the EEG signals, filtering was applied. Specifically, 2 types of filters were utilized, one to remove the mains frequency component, which is 50 Hz and one to drop frequencies outside of the alpha range.

In order to remove the mains frequency, a notch filter was used. Notch filters, also known as band-stop filters, reject a very narrow frequency band and leave the rest of the spectrum almost unchanged. In order to define a notch filter, some parameters are needed. These are the cutoff frequency f_c and the quality factor Q . The following formula connects these two variables.

$$Q = \frac{f_c}{BW}$$

where BW is the bandwidth of the stopband. It is therefore necessary for the bandwidth to be very low in order to have a good notch filter. Since the cutoff frequency in this case is 50Hz, the quality factor was set to 35, thus giving a bandwidth approximately equal to 1.43 Hz, which is satisfactory for this application. The resulting signals from both regions of the brain are shown in the figure below for the movement ‘left’.

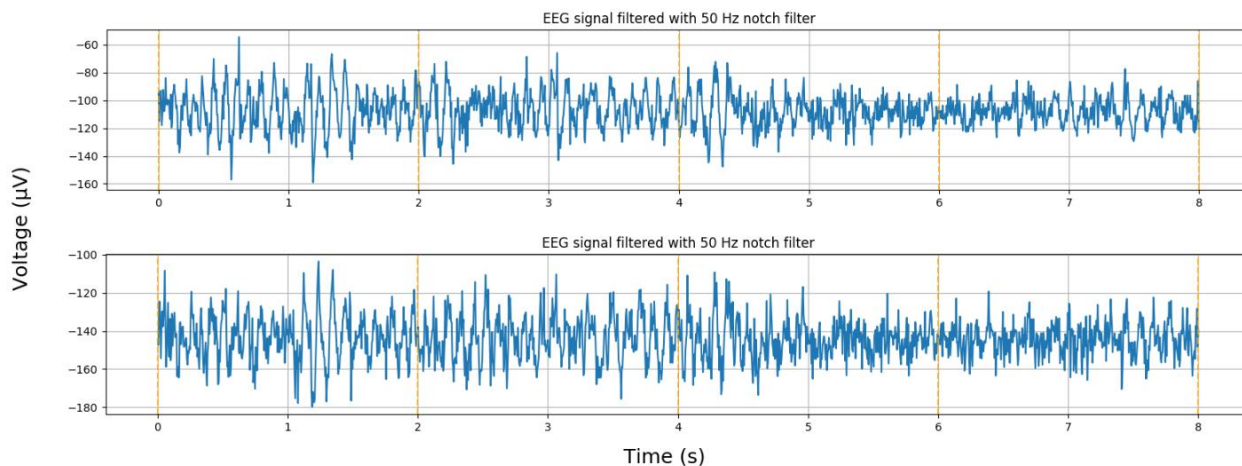


Figure 21. EEG signals after notch filtering

In order to drop the frequencies outside the alpha range, a band-pass filter was utilized. The required parameters to define this filter were the passband and stopband edge frequencies, the maximum loss in the passband and the minimum attenuation in the stopband. The passband of the filter was in the range of 5 – 15 Hz. The stopbands on both sides started 1 Hz away on each side. The maximum loss in the passband was set to 0.1 dB and the minimum attenuation in the stopband 30 dB. The final signals after both filters were applied are show in the figure below for the movement ‘left’.

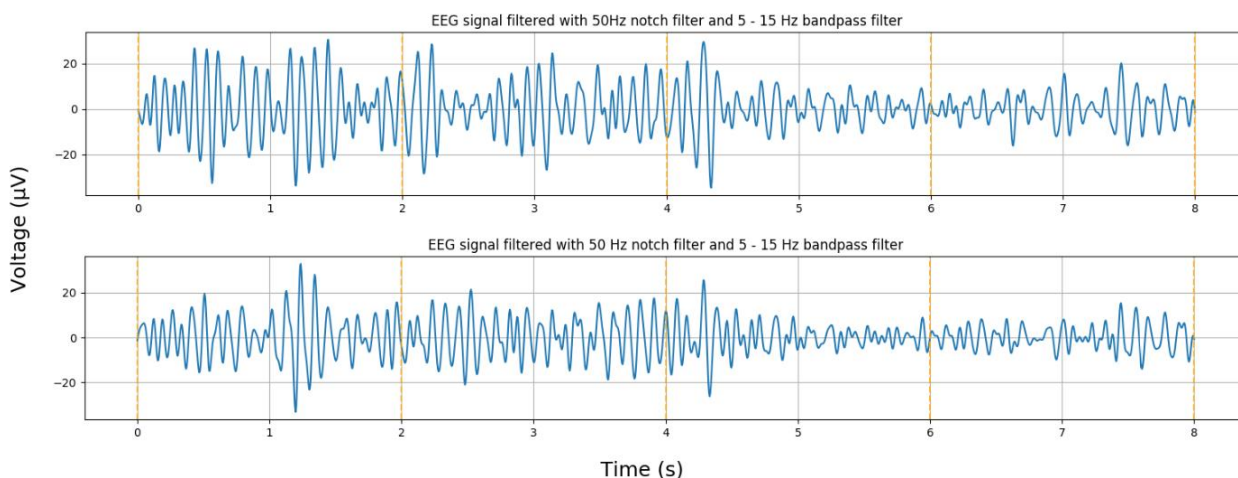


Figure 22. Final EEG signals after both filters were applied

The SciPy Python library was used to create the two filters and filter the signal. For the notch filter, the `scipy.signal.iirnotch()` function was used. For the bandpass filter, the `scipy.signal.iirdesign()` function was used. It is evident from the function names that both filters are *Infinite Impulse Response (IIR)* filters. The reasoning behind this choice was that IIR filters work faster than *Finite Impulse Response (FIR)* filters and they require less memory space.

In addition, the above SciPy functions require the frequencies to be normalized before being passed as inputs. The values are normalized from 0 to 1, where 1 is the Nyquist frequency ($\frac{F_s}{2}$). Since the sampling frequency of the system is 200 Hz, this is achieved by dividing all frequencies with the number 100.

Signals from both electrodes were filtered with the notch filter and the bandpass filter consecutively. Then, the filtered signals were stored again in NumPy arrays for the feature extraction phase. From the figures above, it is easy to conclude that the signals correspond to a ‘left’ movement.

5.2.2 Feature Extraction

Since alpha wave blocking is the reduction of alpha waves’ amplitude, this change can be measured by transforming the EEG signal from the time domain to the frequency domain. This is achieved by computing the Discrete Fourier Transform (DFT) of the signal using the Fast Fourier Transform (FFT) algorithm. The DFT of a signal is defined as follows:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{j \cdot 2\pi \cdot k \cdot n}{N}}$$

where:

X_k is the frequency coefficient (in complex form),

$\frac{k}{N}$ corresponds to the frequency (cycles per samples),

n corresponds to the time unit,

x_n is the input sequence.

The DFT is computed using the `scipy.fftpack.fft()` function, which returns an array with the complex coefficients. Furthermore, the signal side spectrum magnitudes are calculated and stored. Then, the magnitudes for the alpha frequency range are then preserved and finally, summed. This process is repeated 4 times for each individual control signal; this is because control signals comprise of 4 2-second recording intervals.

Min-Max normalization is used to scale the features in the range of [0, 1], which are then saved as a dataset. The resulting feature vector consists of 8 amplitude sums, 4 for each channel (O1, O2). A total of 256 feature vectors are contained within the dataset. A visualization of an example feature vector for the movement ‘left’ is depicted in the figure below, where there are 8 different values, 2 for each bit. It is fairly easy to distinguish each individual bit value; in this case ‘1100’.

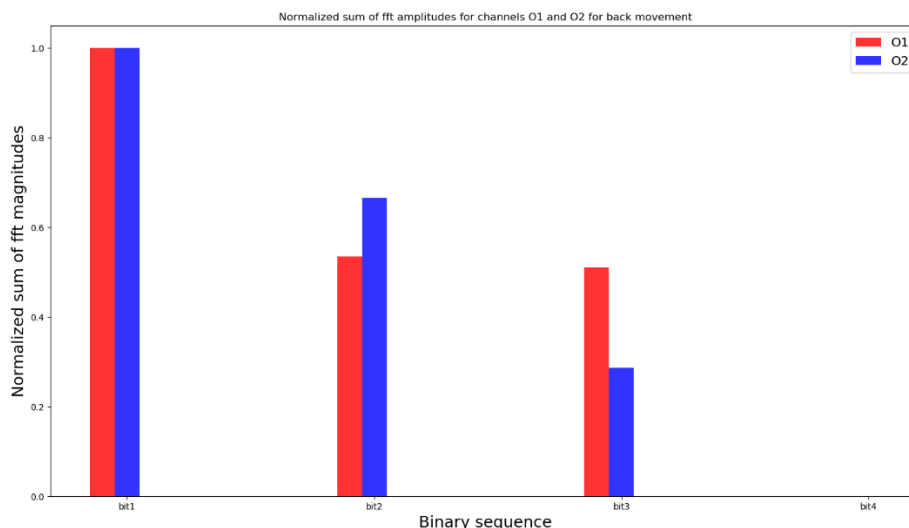


Figure 23. Bar chart showing the normalized sum of the FFT amplitudes for each EEG channel

The exact same procedure described above is also followed for the robot movement but instead of being stored in the dataset, the feature vector is directly fed into the classifier input, which is the next phase of this work.

5.3 Classification & Translation

The classifier utilized for this work is a Multilayer Perceptron (MLP) neural network. The selection was made because MLP neural networks constitute a very popular machine learning technique and there is an abundance of successful applications of MLP neural networks in EEG signal classification and BCI research [30, 31].

The classifier built consists of an input layer with 8 neurons, since the feature vector contains 8 amplitude sums, 4 for each channel. Furthermore, there are 4 neurons in the output layer because there are 4 available classes (forward, reverse, left and right). Moreover, there are 2 hidden layers, each one consisting of 100 neurons.

The number of layers and neurons was determined by a trial-and-error procedure. Specifically, 1 – 3 hidden layers were considered. In addition, for each layer, the number of neurons examined was 20 – 200 with a step of 20. In total, 175 different network configurations were considered. It was concluded that a 2-hidden layer network with 100 neurons in each layer achieved the desired performance in terms of classification accuracy. A graphical depiction of the classifier built is illustrated in the figure below.

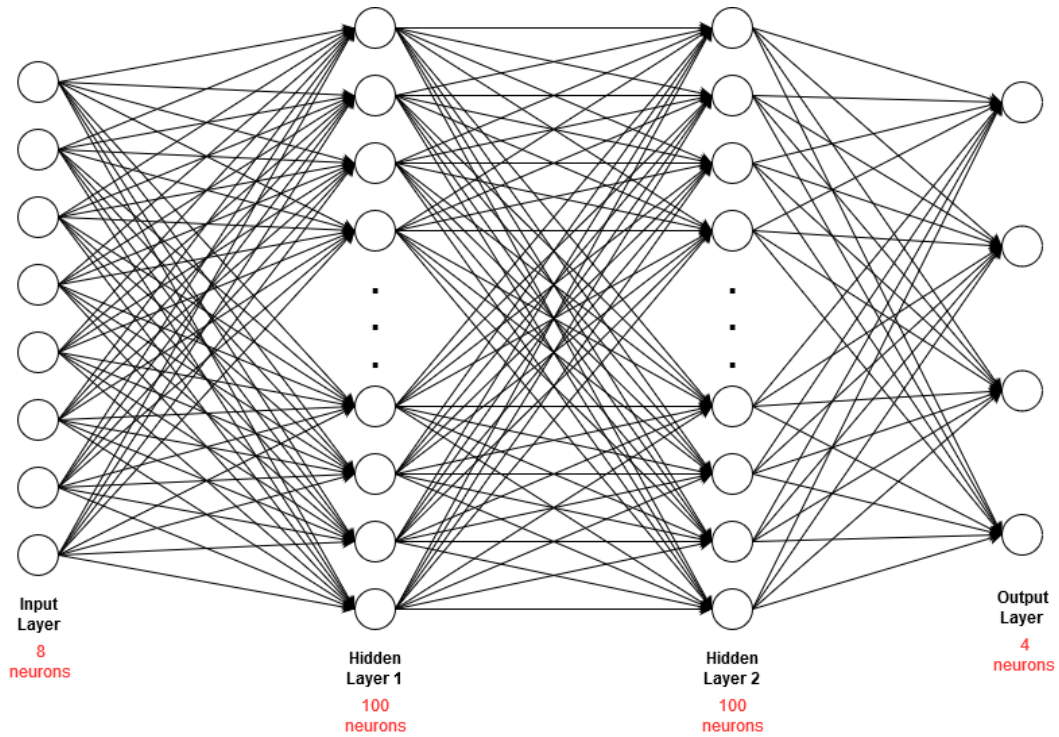


Figure 24. Structure of the neural network built

The activation function for the hidden layers is the Rectified Linear Unit (ReLU). The advantages of ReLU include increased training speed and less suffering from the vanishing gradient problem [31]. As for the output layers, the sigmoid function was used, which bounds the output of each layer in the range of [0, 1]. This means that each neuron in the output layer produces probabilities of the input being on of the 4 commands. The command with the highest probability is selected.

The loss function used to measure the prediction error of the network during training is binary cross-entropy [32], which is widely used in binary classification problems. It is defined as:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N [y_n \cdot \log \hat{y}_n + (1 - y_n) \cdot \log(1 - \hat{y}_n)]$$

where N is the number of samples, y_n is the target output, and \hat{y}_n is the predicted output. The training batch size is 16 and 40% of the total data are used for validation. The number of epochs is 100. Finally, the optimization algorithm used to minimize the prediction error by adjusting the weight of each neuron is Adam, using the default hyperparameter values, as described in [33]. All models were trained in TensorFlow [34], using the Keras API [35]. In the figure below, the neural network model training and validation loss is displayed. It can be distinguished that training could take place for a smaller number of epochs, since the loss is already at an acceptable value at around 25 epochs.

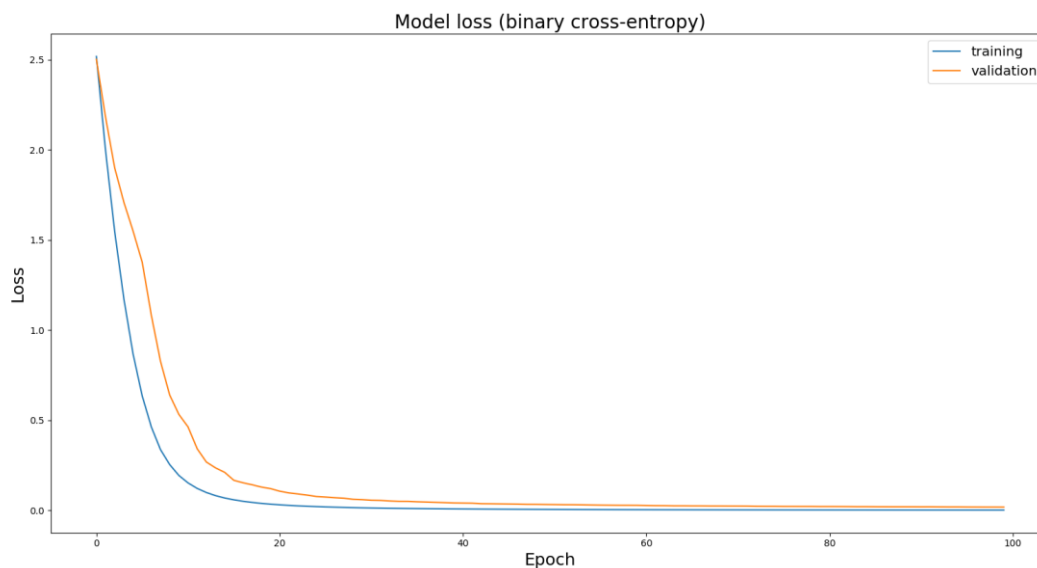


Figure 25 Training and validation model loss

After confirming the accuracy of this model with the tests, it is time to move on to the actual implementation. The model's architecture, weights and all other necessary information are stored in HDF5 format. The NumPy arrays that hold the features are joined together and their shape is changed so as to be compatible with the neural network's inputs. The network then classifies the features accordingly and outputs its prediction. This prediction is used to decide which movement will be executed.

After the prediction is made, the computer connects to a Raspberry Pi via TCP and transmits the command. The decoded command is received by the Raspberry Pi which in turn, is connected to and Arduino UNO through a serial connection (USB). The command is sent to the Arduino, which moves the motors on the robot according to the predicted movement. Since the current required to drive the motors is quite high compared to the current that the Arduino can supply, an L298N motor driver is used. Each motor is activated for 2 seconds, which is the duration of each movement. The final setup is shown on the figure below.

The Raspberry Pi could have been omitted entirely if the Arduino UNO was equipped with a Wi-Fi module such as the ESP8266, lowering the overall cost of implementation of this work. Nevertheless, with a Raspberry Pi on the robot, the range of options in future versions of this project can be widely extended, since the Raspberry Pi is a very powerful microcomputer. An example of this would be mounting a camera on the robot.

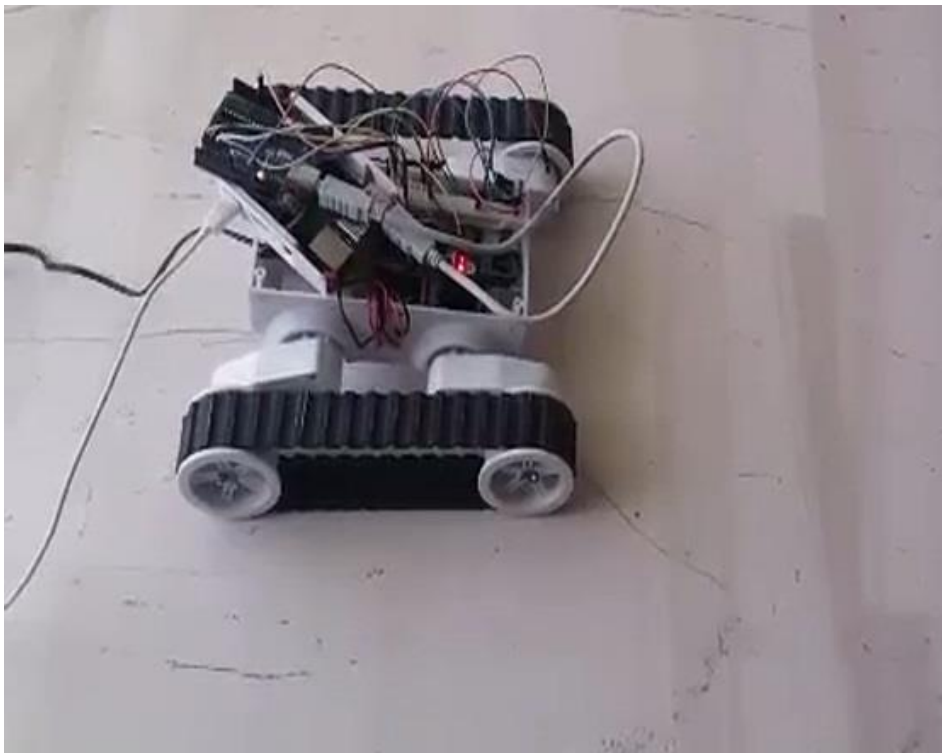


Figure 26. Final robot setup

6 RESULTS & DISCUSSION

The performance of the developed system was evaluated by using both offline and online data which were gathered through a series of experimental tests performed on 12 healthy subjects.

6.1 Results

6.1.1 Evaluation with Offline Data

For the offline evaluation, the system was tested by using prerecorded data gathered from the same subjects used for recording the training data. Specifically, a small testing dataset of 50 feature vectors representing different movements was used. The neural network classified all of the movements correctly.

6.1.2 Evaluation with Online Data

After evaluating the system on offline data, a real-time performance analysis was carried out by using six female and six male subjects aged 20 to 28, and two female and two male subjects aged 32 to 40 years. The specific subjects were different from those that were used for the classifier training and offline evaluation. For this purpose, an experimental process was carried out. The subjects were instructed to move the robot in the following order: forward, reverse, left and right consecutively.

Each one of the 12 subjects was briefed shortly on how the BCI works and how to issue each movement command to the robot. A small number of trial runs were performed for the subjects to

get acquainted with the procedure. In total, 40 experimental tests were carried out. The total number of commands issued was 480.

The results of the experimental procedure showed that the lowest classification accuracy achieved among the subjects was 85% while the highest one was 97.5%. The overall accuracy for all commands was 92.1%. The confusion matrix for the total number of commands considered for classification is illustrated in the figure below, where green diagonal cells correspond to commands that are successfully classified, the red cells correspond to incorrectly classified commands, the gray column on the right displays the precision and false recovery rate of the classifier, the gray row in the bottom expresses the recall and the false negative rate of the classifier, and the blue cell displays the overall accuracy.

Output Class	forward	113 23.5%	1 0.2%	7 1.5%	2 0.4%	91.9% 8.1%
	reverse	2 0.4%	109 22.7%	0 0.0%	9 1.9%	90.8% 9.2%
	left	5 1.0%	0 0.0%	113 23.5%	2 0.4%	94.2% 5.8%
	right	0 0.0%	10 2.1%	0 0.0%	107 22.3%	91.5% 8.5%
		94.2% 5.8%	90.8% 9.2%	94.2% 5.8%	89.2% 10.8%	92.1% 7.9%
		forward	reverse	left	right	Target Class

Figure 27. Confusion matrix for all issued subject commands

Next, for analysis purposes, the experimental results were studied according to the gender and the age of the subjects that participated in the experimental procedure.

Specifically, the results were first grouped and analyzed separately for each gender. The confusion matrices for the female subjects and the male subjects are depicted in the two figures below respectively, where it is shown that the female subjects had a 1.6% higher classification accuracy compared to the male subjects (92.9% to 91.3%).

Output Class	forward	56 23.3%	1 0.4%	4 1.7%	0 0.0%	91.8% 8.2%
	reverse	1 0.4%	55 22.9%	0 0.0%	3 1.3%	93.2% 6.8%
	left	3 1.3%	0 0.0%	56 23.3%	1 0.4%	93.3% 6.7%
	right	0 0.0%	4 1.7%	0 0.0%	56 23.3%	93.3% 6.7%
		93.3% 6.7%	91.7% 8.3%	93.3% 6.7%	93.3% 6.7%	92.9% 7.1%
	forward	reverse	left	right		
	Target Class					

Figure 28. Confusion matrix for female subjects

Output Class	forward	57 23.8%	0 0.0%	3 1.3%	2 0.8%	91.9% 8.1%
	reverse	1 0.4%	54 22.5%	0 0.0%	6 2.5%	88.5% 11.5%
	left	2 0.8%	0 0.0%	57 23.8%	1 0.4%	95.0% 5.0%
	right	0 0.0%	6 2.5%	0 0.0%	51 21.3%	89.5% 10.5%
		95.0% 5.0%	90.0% 10.0%	95.0% 5.0%	85.0% 15.0%	91.3% 8.8%
	forward	reverse	left	right		
	Target Class					

Figure 29. Confusion matrix for male subjects

Next, the experimental results were grouped and analyzed according to the age of the subjects. The first group contains the results that refer to the eight subjects aged between 20 and 28 years and the second one the results derived by the four subjects aged between 32 and 40 years. The confusion matrices for the group 20 – 28 and the group 32 – 40 are depicted in the figures below, where it is shown that these two groups have almost the same precision accuracy (92.2% for the subjects aged 20 to 28 and 91.9% for the subjects aged 32 to 40).

forward	74 23.1%	1 0.3%	4 1.3%	1 0.3%	92.5% 7.5%
reverse	2 0.6%	72 22.5%	0 0.0%	5 1.6%	91.1% 8.9%
left	4 1.3%	0 0.0%	76 23.8%	1 0.3%	93.8% 6.2%
right	0 0.0%	7 2.2%	0 0.0%	73 22.8%	91.3% 8.8%
	92.5% 7.5%	90.0% 10.0%	95.0% 5.0%	91.3% 8.8%	92.2% 7.8%
	forward	reverse	left	right	

Figure 30. Confusion matrix for ages 20 to 28

forward	39 24.4%	0 0.0%	3 1.9%	1 0.6%	90.7% 9.3%
reverse	0 0.0%	37 23.1%	0 0.0%	4 2.5%	90.2% 9.8%
left	1 0.6%	0 0.0%	37 23.1%	1 0.6%	94.9% 5.1%
right	0 0.0%	3 1.9%	0 0.0%	34 21.3%	91.9% 8.1%
	97.5% 2.5%	92.5% 7.5%	92.5% 7.5%	85.0% 15.0%	91.9% 8.1%
	forward	reverse	left	right	
	Target Class				

Figure 31. Confusion matrix for ages 32 to 40

6.2 Discussion

The overall accuracy of 92.1% achieved by the system is considered to be rather satisfactory, especially given the fact that this rate is the result of real-time evaluation. It is also important to note that different subjects than the ones used for training were employed for this evaluation, a fact which attests to the robustness of the proposed method.

Better insight into the results can be gained by looking at the confusion matrix for all issued subject commands. It can be seen that the proposed approach not only achieves a satisfactory overall success rate, but also provides good performance per each individual movement.

Further analysis of inter-class performance shows that in 8.3% of the cases a ‘reverse’ command was issued, it was misclassified as a ‘right’ command. Moreover, the command ‘left’ was misclassified as a ‘forward’ command at a rate of 5.8% and the ‘right’ command as a ‘reverse’ command at a rate of 7.5%. This can be attributed to the fact that there is a short time delay until alpha waves amplitudes increase or decrease upon eye closing and opening, respectively. Therefore, these amplitudes are calculated into the next bit value, which can lead to errors.

A good indicator of the probability of a command being classified wrongly is the Hamming distance between each command, shown in the table below. Therefore, the ‘forward’ and ‘reverse’ commands are more likely to be misinterpreted into ‘left’ or ‘right’ commands and vice versa. Representing each command with more than four bits would increase the Hamming distance and, as a result, the system accuracy, but it would also increase the overall recording time since the duration of every bit recording is two seconds.

Table 2. Hamming distances between robot commands

Command	'1010'	'0101'	'1100'	'0011'	
Forward	'1010'	0	4	2	2
Reverse	'0101'	4	0	2	2
Left	'1100'	2	2	0	4
Right	'0011'	2	2	4	0

The categorization of the experimental results performed according to the age of the subjects showed that the deviation in the classification accuracy of the age groups is negligible, probably because of the relatively small age difference between the two groups.

However, female subjects in the experimental procedure followed, achieved relatively higher classification accuracy than the male ones. This can be attributed to the fact that women in general exhibit greater alpha amplitudes than men [36, 37].

On the other hand, although the performance of the proposed system was found to be successful, it is true that all the participants during the experiments made in this research work were healthy. Therefore, in real-life conditions the effectiveness of experimental systems, like the one developed in this work, is questionable because it strongly depends on the health conditions of their users who are supposed not only to be disabled persons but also having disability of various levels.

Moreover, the achievement of successful performance of a mobile robot within the territory of a controlled laboratory environment does not guarantee its effectiveness in real-world applications where the conditions are variable and fuzzy.

Furthermore, the BCI systems that are based on a single signal may not be applicable to all users. Therefore, hybrid schemes which make combined use of various types of brain signals can be a more complex yet even more effective alternative.

7 CONCLUSIONS & FUTURE WORK

The work presented here, concerns the development of a control system which guides the motion of a mobile robot via a synchronous and endogenous EEG-based BCI, which uses the alpha brain waveforms of a human operator.

Experiments made, with the involvement of 12 subjects who had minimum training, proved that the system developed is able to guide the robotic vehicle under control in forward, left, backward and right directions according to the eyes' blinking of its human operator. The accuracy achieved ranges from 85% up to 97.5% among the subjects while the overall accuracy was found to be equal to 92.1% for all commands. Further analysis of the experimental data related with the classification accuracy between different genders and age groups showed that female subjects performed slightly better than male ones (92.9% to 91.3%, respectively), while there was just a trivial difference detected between subjects aged from 20 to 28 years and subjects aged from 32 to 40 years (92.2% to 91.9%, respectively).

Considering both the classification accuracy achieved, by applying real-time evaluation, and the robustness evinced by the fact that subjects involved during training were different than those during the experimental evaluation, it is concluded that this work has the potential to be incorporated in applications such as the motion assistance to handicapped persons.

In the future, this work can be improved by experimenting with hybrid BCIs where alpha brainwaves will be used along with brain signals of other type(s) such as P300 and SSVEP [38].

Moreover, task metrics, such as task completion time and path length traveled, and ergonomic metric, such as mental workload of participants, can be additionally used for the accomplishment of multivariable evaluation of the performance of the system built [14].

Additionally, robot guidance can be assisted via additional sensors embedded into the robotic vehicle [39].

The detrimental effect of artifacts on EEG data can be removed by using modern algorithms that combine source decomposition with blind source separation and adaptive filtering [40].

Finally, enhanced performance can be achieved by applying advanced methods which have been proposed in order to add new knowledge to already learned models of robot semantic localization [41].

Bibliography – Citations – Internet Sources

1. *World Report on Disability*. (n.d.). Retrieved March 14, 2021, from <https://www.who.int/teams/noncommunicable-diseases/sensory-functions-disability-and-rehabilitation/world-report-on-disability>
2. Dixon, A. M. R., Allstot, E. G., Gangopadhyay, D., & Allstot, D. J. (2012). Compressed Sensing System Considerations for ECG and EMG Wireless Biosensors. *IEEE Transactions on Biomedical Circuits and Systems*, 6(2). <https://doi.org/10.1109/TBCAS.2012.2193668>
3. Perdiz, J., Pires, G., & Nunes, U. J. (2017). Emotional state detection based on EMG and EOG biosignals: A short survey. *2017 IEEE 5th Portuguese Meeting on Bioengineering (ENBENG)*. <https://doi.org/10.1109/ENBENG.2017.7889451>
4. Valais, I.; Koulouras, G.; Fountos, G.; Michail, C.; Kandris, D.; Athinaios, S. Design and Construction of a Prototype ECG Simulator. *EJST2014*, 9, 11–18.
5. Subha, D. P., Joseph, P. K., Acharya U, R., & Lim, C. M. (2010). EEG Signal Analysis: A Survey. *Journal of Medical Systems*, 34(2). <https://doi.org/10.1007/s10916-008-9231-z>
6. Schomer, D. L., & Lopes da Silva, F. H. (Eds.). (2017). *Niedermeyer's Electroencephalography* (Vol. 1). Oxford University Press. <https://doi.org/10.1093/med/9780190228484.001.0001>
7. Haykin, Simon (Mc Master University, C. (n.d.). *Neural Networks and Learning Machines* (3d edition). Pearson.
8. Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). Brain–computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6). [https://doi.org/10.1016/S1388-2457\(02\)00057-3](https://doi.org/10.1016/S1388-2457(02)00057-3)
9. Abdulkader, S. N., Atia, A., & Mostafa, M.-S. M. (2015). Brain computer interfacing: Applications and challenges. *Egyptian Informatics Journal*, 16(2). <https://doi.org/10.1016/j.eij.2015.06.002>
10. Katona, J., & Kovari, A. (2016). A Brain–Computer Interface Project Applied in Computer Engineering. *IEEE Transactions on Education*, 59(4). <https://doi.org/10.1109/TE.2016.2558163>
11. Katona J, & Kovari A. (2018). The Evaluation of BCI and PEBL-based Attention Tests. *Acta Polytechnica Hungarica*, 15(3). <https://doi.org/10.12700/APH.15.3.2018.3.13>
12. Katona, j, & Kovari, A. (2018). Examining the Learning Efficiency by a Brain-Computer Interface System. *Acta Polytechnica Hungarica*, 15(3). <https://doi.org/10.12700/APH.15.3.2018.3.14>
13. Nicolas-Alonso, L. F., & Gomez-Gil, J. (2012). Brain Computer Interfaces, a Review. *Sensors*, 12(2). <https://doi.org/10.3390/s120201211>
14. Bi, L., Fan, X.-A., & Liu, Y. (2013). EEG-Based Brain-Controlled Mobile Robots: A Survey. *IEEE Transactions on Human-Machine Systems*, 43(2). <https://doi.org/10.1109/TSMCC.2012.2219046>
15. Minguillon, J., Lopez-Gordo, M. A., & Pelayo, F. (2017). Trends in EEG-BCI for daily-life: Requirements for artifact removal. *Biomedical Signal Processing and Control*, 31. <https://doi.org/10.1016/j.bspc.2016.09.005>
16. Padmavathi, R.; Ranganathan, V. A review on EEG based brain computer interface systems. *Int. J. Emerg. Technol. Adv. Eng.* 2014, 4, 683–696.
17. Gandhi, V. *Brain-Computer Interfacing for Assistive Robotics: Electroencephalograms, Recurrent Quantum Neural Networks and User-Centric Graphical Interfaces*, 1st ed.; Academic Press: London, UK, 2014; pp. 29–30
18. Alexandridis, A., Chondrodima, E., Giannopoulos, N., & Sarimveis, H. (2017). A Fast and Efficient Method for Training Categorical Radial Basis Function Networks. *IEEE*

- Transactions on Neural Networks and Learning Systems*, 28(11).
<https://doi.org/10.1109/TNNLS.2016.2598722>
19. Alexandridis, A., Chondrodima, E., & Sarimveis, H. (2013). Radial Basis Function Network Training Using a Nonsymmetric Partition of the Input Space and Particle Swarm Optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 24(2).
<https://doi.org/10.1109/TNNLS.2012.2227794>
 20. Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., & Yger, F. (2018). A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of Neural Engineering*, 15(3). <https://doi.org/10.1088/1741-2552/aab2f2>
 21. Tanaka, K., Matsunaga, K., & Wang, H. O. (2005). Electroencephalogram-based control of an electric wheelchair. *IEEE Transactions on Robotics*, 21(4).
<https://doi.org/10.1109/TRO.2004.842350>
 22. Choi, K.; Cichocki, A. Control of a wheelchair by motor imagery in real time. *In Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*, Daejeon, Korea, 2–5 November 2008; Springer: Berlin, Germany, 2008; pp. 330–33
 23. Ferreira, A., Silva, R. L., Celeste, W. C., Filho, T. F. B., & Filho, M. S. (2007). Human–machine interface based on muscular and brain signals applied to a robotic wheelchair. *Journal of Physics: Conference Series*, 90. <https://doi.org/10.1088/1742-6596/90/1/012094>
 24. Mandel, C.; Luth, T.; Laue, T.; Röfer, T.; Graser, A.; Krieg-Bruckner, B. Navigating a smart wheelchair with a brain–computer interface interpreting steady-state visual evoked potentials. *In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 10–15 October 2009; pp. 1118–1125
 25. Rebsamen, B.; Burdet, E.; Guan, C.; Zhang, H.; Teo, C.L.; Zeng, Q.; Ang, M.; Laugier, C. A brain-controlled wheelchair based on P300 and path guidance. *In Proceedings of the 1st IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, Pisa, Italy, 20–22 February 2006; pp. 1001–10
 26. Iturrate, I., Antelis, J. M., Kubler, A., & Minguez, J. (2009). A Noninvasive Brain-Actuated Wheelchair Based on a P300 Neurophysiological Protocol and Automated Navigation. *IEEE Transactions on Robotics*, 25(3). <https://doi.org/10.1109/TRO.2009.2020347>
 27. Benevides, A.B.; Bastos, T.F.; Filho, M.S. Proposal of brain–computer interface architecture to command a robotic wheelchair. *In Proceedings of the IEEE International Symposium in Industrial Electronics*, Gdansk, Poland, 27–30 June 2011; pp. 2249–2254.
 28. Samson, V.R.R.; Kitti, B.P.; Kumar, S.P.; Babu, D.S.; Monica, C. Electroencephalogram-Based OpenBCI Devices for Disabled People. *In Proceedings of the 2nd International Conference on Micro-Electronics, Electromagnetics and Telecommunications*, Visakhapatnam, India, 6–7 January 2017; pp. 229–238
 29. Olejarczyk, E., Bogucki, P., & Sobieszek, A. (2017). The EEG Split Alpha Peak: Phenomenological Origins and Methodological Aspects of Detection and Evaluation. *Frontiers in Neuroscience*, 11. <https://doi.org/10.3389/fnins.2017.00506>
 30. Jana, G. C., Swetapadma, A., & Pattnaik, P. K. (2018). Enhancing the performance of motor imagery classification to design a robust brain computer interface using feed forward back-propagation neural network. *Ain Shams Engineering Journal*, 9(4).
<https://doi.org/10.1016/j.asej.2017.12.003>
 31. Subasi, A., & Erçelebi, E. (2005). Classification of EEG signals using neural network and logistic regression. *Computer Methods and Programs in Biomedicine*, 78(2).
<https://doi.org/10.1016/j.cmpb.2004.10.009>
 32. Cross-Entropy Method. *Annals of Operations Research*, 134(1).
<https://doi.org/10.1007/s10479-005-5724-z>
 33. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980

34. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A system for large-scale machine learning. *In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
35. Chollet, F. Keras: The python deep learning library. *Astrophys. Source Code Libr.* 2018. Available online: <https://keras.io/>
36. Wada, Y., Takizawa, Y., Zheng-Yan, J., & Yamaguchi, N. (1994). Gender Differences in Quantitative EEG at Rest and during Photic Stimulation in Normal Young Adults. *Clinical Electroencephalography*, 25(2). <https://doi.org/10.1177/155005949402500209>
37. Corsi-Cabrera, M., Ramos, J., Guevara, M. A., Arce, C., & Gutierrez, S. (1993). Gender Differences in the EEG During Cognitive Activity. *International Journal of Neuroscience*, 72(3–4). <https://doi.org/10.3109/00207459309024114>
38. Amiri, S., Fazel-Rezai, R., & Asadpour, V. (2013). A Review of Hybrid Brain-Computer Interface Systems. *Advances in Human-Computer Interaction*, 2013. <https://doi.org/10.1155/2013/187024>
39. Zantalis, F., Koulouras, G., Karabetsos, S., & Kandris, D. (2019). A Review of Machine Learning and IoT in Smart Transportation. *Future Internet*, 11(4). <https://doi.org/10.3390/fi11040094>
40. Jafarifarmand, A., & Badamchizadeh, M. A. (2019). EEG Artifacts Handling in a Real Practical Brain-Computer Interface Controlled Vehicle. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(6). <https://doi.org/10.1109/TNSRE.2019.2915801>
41. Cruz, E.; Rangel, J.C.; Gomez-Donoso, F.; Bauer, Z.; Cazorla, M.; García-Rodríguez, J. Finding the place: How to train and use convolutional neural networks for a dynamically learning robot. *In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1