|  |
| --- |
| MSC COMPUTER SCIENCE DISSERTATION |
| https://www.hope.ac.uk/media/liverpoolhope/styleassets/cssimages/media,36583,en.png |
| EEG Based BCI Prosthetic Using Motor Imagery |
| Daniel Elstob |
| 14011862@hope.ac.uk |
| MSc Computer Science  CSCM007 |
| Under the supervision of Emanuele Secco |
| 15th January 2016 |
|  |

|  |
| --- |
|  |

# Acknowledgement

Firstly, I would like to express my sincere gratitude to my advisor Emanuele Secco for the continuous support of the research and writing of the dissertation.

Besides my advisor, I would like to thank Dr. David Reid for providing the EEG headset used in this study and for the support provided during the writing of this paper.

Contents

[Acknowledgement 1](#_Toc441731364)

[Figures 4](#_Toc441731365)

[Abstract 7](#_Toc441731366)

[Chapter 1: Introduction 8](#_Toc441731367)

[1.1 Background 8](#_Toc441731368)

[1.2 Related Research 9](#_Toc441731369)

[1.3 Aims and Objectives 10](#_Toc441731370)

[1.4 Motivation 11](#_Toc441731371)

[1.5 Scope of Thesis 11](#_Toc441731372)

[Chapter 2 Preliminaries 13](#_Toc441731373)

[2.1 Structure of the Brain 13](#_Toc441731374)

[2.2 ElectroEncephaloGram Signal 13](#_Toc441731375)

[2.2.1 Introduction 13](#_Toc441731376)

[2.2.2 EEG Artifacts 14](#_Toc441731377)

[2.3 Brain Plasticity 15](#_Toc441731378)

[2.4 BCI Illiteracy 15](#_Toc441731379)

[Chapter 3 Methodology 16](#_Toc441731380)

[3.1 Overview 16](#_Toc441731381)

[3.2 EEG Headset 16](#_Toc441731382)

[3.3 Cognitive Suite 18](#_Toc441731383)

[3.4 OpenViBE 18](#_Toc441731384)

[3.4.1 Overview 18](#_Toc441731385)

[3.4.2 System Features 19](#_Toc441731386)

[3.4.3 Installation and Compatibility 21](#_Toc441731387)

[3.4.4 Connecting the Server and the Designer 21](#_Toc441731388)

[3.4.5 Designer scenarios 24](#_Toc441731389)

[3.5 Arduino 28](#_Toc441731390)

[3.5.1 Arduino Software 29](#_Toc441731391)

[3.6 Python Software 29](#_Toc441731392)

[3.7 3D Printed Arm 30](#_Toc441731393)

[Chapter 4 Implementation 31](#_Toc441731394)

[4.1 3D Printed Arm and Arduino 31](#_Toc441731395)

[4.2 System One – Cognitive Suite 32](#_Toc441731396)

[4.3 System Two - OpenViBE 34](#_Toc441731397)

[4.3.1 Acquisition 34](#_Toc441731398)

[4.3.2 Training CSP 36](#_Toc441731399)

[4.3.3 Linear Discrimination Analysis Classifier 37](#_Toc441731400)

[4.3.4 Experiment Analysis 40](#_Toc441731401)

[4.3.5 Real-time One 43](#_Toc441731402)

[4.3.6 Real-time Two 44](#_Toc441731403)

[4.3.7 Python Scripting Box and Arduino Code 46](#_Toc441731404)

[Chapter 5 Evaluation of Results 49](#_Toc441731405)

[5.1 Testing Arm and Arduino 49](#_Toc441731406)

[5.2 System One – Cognitive Suite 49](#_Toc441731407)

[5.3 System Two - OpenViBE 49](#_Toc441731408)

[5.3.1 Acquisition 49](#_Toc441731409)

[5.3.2 Training CSP 51](#_Toc441731410)

[5.3.3 Classify 51](#_Toc441731411)

[5.3.4 Analysis 51](#_Toc441731412)

[5.3.5 Real-time One 53](#_Toc441731413)

[5.3.5 Real time Two 53](#_Toc441731414)

[5.3.6 Python Box 53](#_Toc441731415)

[Chapter 6 Conclusions and Future Research 54](#_Toc441731416)

[6.1 Conclusions 54](#_Toc441731417)

[6.1.1 3D Arm and Arduino UNO 54](#_Toc441731418)

[6.1.2 Headset 54](#_Toc441731419)

[6.1.3 System One – Cognitive Suite 55](#_Toc441731420)

[6.1.4 System Two - OpenViBE 55](#_Toc441731421)

[6.1.6 Motor Imagery 55](#_Toc441731422)

[6.1.7 Current BCI 56](#_Toc441731423)

[6.2 Limitations 56](#_Toc441731424)

[6.3 BCI Illiteracy 56](#_Toc441731425)

[6.4 Robustness and Applicability 57](#_Toc441731426)

[6.5 Final Conclusion 57](#_Toc441731427)

[6.6 Future Research 58](#_Toc441731428)

[References 59](#_Toc441731429)

[Appendix A – CD 62](#_Toc441731430)

# Figures

[Figure 1 Motor Movement and Motor Imagery (Miller et al., 2010) 13](#_Toc441731431)

[Figure 2 Prosthetic Arm: Discussed in Section 4.1 16](#_Toc441731432)

[Figure 3 Emotiv EPOC+ Headset 17](#_Toc441731433)

[Figure 4 14 Labelled Electrodes of the Emotiv EPOC+ Headset 17](#_Toc441731434)

[Figure 5 Emotiv Cognitive Suite 18](#_Toc441731435)

[Figure 6 Acquisition Server GUI 21](#_Toc441731436)

[Figure 7 Temporal Filter Settings for System Two- OpenViBE 24](#_Toc441731437)

[Figure 8 Acquisition Scenario for System Two- OpenViBE 25](#_Toc441731438)

[Figure 9 CSP Trainer for System Two- OpenViBE 26](#_Toc441731439)

[Figure 10 Classifier Trainer for System Two- OpenViBE 27](#_Toc441731440)

[Figure 11 Online Scenario for System Two- OpenViBE 28](#_Toc441731441)

[Figure 12 Arduino UNO 29](#_Toc441731442)

[Figure 13 Basic Structure of the Arduino Code 29](#_Toc441731443)

[Figure 14 3D Printed Arm 31](#_Toc441731444)

[Figure 15 Sample Code for Controlling Prosthetic 32](#_Toc441731445)

[Figure 16 Emotiv Full Strength Signal Sensors 33](#_Toc441731446)

[Figure 17 Training the Neutral Command 34](#_Toc441731447)

[Figure 18 Cognitive Suite showing the cube being pushed and the assigned key broadcasting 34](#_Toc441731448)

[Figure 19 OpenViBE Connected with EPOC+ 35](#_Toc441731449)

[Figure 20 Acquisition.xml GUI 36](#_Toc441731450)

[Figure 21 CSP Spatial Filter Training Success Message 37](#_Toc441731451)

[Figure 22 Controls for the scenarios 37](#_Toc441731452)

[Figure 23 LDA Classifier Scenario Part 1 38](#_Toc441731453)

[Figure 24 LDA Classifier Scenario Part 2 38](#_Toc441731454)

[Figure 25 LDA Classifier Scenario Part 3 39](#_Toc441731455)

[Figure 26 LDA Classifier Scenario Part 4 39](#_Toc441731456)

[Figure 27 LDA Classifier Scenario Part 5 39](#_Toc441731457)

[Figure 28 Acquisition1.ov LDA Results 40](#_Toc441731458)

[Figure 29 Aquisition2.ov LDA Results 40](#_Toc441731459)

[Figure 30 Analysis.xml 41](#_Toc441731460)

[Figure 31 Analysis GUI 42](#_Toc441731461)

[Figure 32 Direction of Motor Imagery Strength 42](#_Toc441731462)

[Figure 33 Overall Performance Metrics 43](#_Toc441731463)

[Figure 34 Real-time One 43](#_Toc441731464)

[Figure 35 Real-time Two Scenario 44](#_Toc441731465)

[Figure 36 Real-time Two GUI 45](#_Toc441731466)

[Figure 37 Real-time Two Overall Performance 45](#_Toc441731467)

[Figure 38 Second Test Real-time Two 46](#_Toc441731468)

[Figure 39 Second Test Real-time Two 46](#_Toc441731469)

[Figure 40 Python Scripting Box 47](#_Toc441731470)

[Figure 41 Opening COM1 using Python Script 47](#_Toc441731471)

[Figure 42 Python Scripting Box Extracting Stimulation Codes and Printing 47](#_Toc441731472)

[Figure 43 Arduino Serial Recieving the Characters 47](#_Toc441731473)

[Figure 44 Arduino code for System Two 48](#_Toc441731474)

[Figure 45 Check Signal 1 50](#_Toc441731475)

[Figure 46 Check Signal 2 51](#_Toc441731476)

[Figure 47 100% Accuracy for just left motor imagery 52](#_Toc441731477)

[Figure 48 Introduction of right motor imagery 53](#_Toc441731478)

# Abstract

Brain computer interfaces provide the opportunity to control external devices using the brain. In this thesis we research into the field of brain computer interfaces and implement two systems that will control a simple prosthetic. Excellent results are shown in the first system that used the emotiv cognitive suite and Arduino software to control the prosthetic through character input associated with the taught actions of the cognitive suit. This system provides evidence of the feasibility of brain signals being a viable approach to controlling the chosen prosthetic.

Excellent results are also shown in the second system. This stem allowed for the training and clasification of electroencephalogram signals for motor imagery tasks. When analysing the system clear visual representations of the performance and accuracy are presented in the results using a confusion matrix, accuracy measurement and a feedback bar signfying signal strength. Experiement with various acquisiton datasets were carried out and with a critical evaluation of the results given. Finally depending on the classification of the brain signal a Python script will output a message to the Arduino to control the prosthetic.

We conclude with an evaluation of overall good results for the designed and implemented brain computer interface systems and suggest future work improvements for increased control and accuracy in relation to hardware and software used.

# Chapter 1: Introduction

## 1.1 Background

With the current knowledge of how the brain works researchers are able to develop a wide range of applications that can improve life quality to those with muscular or motor-neuron disabilities using devices called Brain Computer Interfaces (BCI).

BCI is defined as “a communication and/or control system that allows real-time interaction between the human brain and external devices” (Mak and Wolpaw, 2009). Researchers in this field have focused on creating a BCI that could assist or repair human cognitive functions such as restoring hand grasp (Pfurtscheller et al., 2003) or even augmentations like controllable prosthesis which “aim to provide a communication channel equivalent to “typing” on a computer” (Sajda et al., 2008). As a result of this the research and development of BCIs are largely towards neuroprosthetics, prosthetics that aim towards the rehabilitation of patients for instance Müller-Putz et al. showed that BCI can be used “for the control of neuroprosthesis in patients with high spinal cord lesions” (Müller-Putz et al., 2005).

Ultimately the purpose of a BCI is to acquire the desired signals or intent of the user however these BCIs can be either invasive or non-invasive depending on the requirements of the user. An invasive BCI means that the system or device is integrated into the individual for instance electrodes directly into the brain matter or partially integrated into the inner skull. This integration provides the clearest and strongest signals however it comes with the risk of rejection from the body as the electrodes are a foreign body that the user’s body may fight or scar tissue may build up over time reducing the signal recorded.

These types are more extreme than a non-invasive which can operate outside the individual’s body and not cause any unnecessary harm to the user’s body. Non-invasive BCI are used by researchers and users that have disabilities such as locked in syndrome which is described by Smith and Delargy that “characteristics of the syndrome are quadriplegia and anarthria with preservation of consciousness.” (Smith and Delargy, 2005). For individuals such as these invasive BCI can offer a restoration for sensory functions, transmission of any sensory information to the brain or even stimulate an individual’s brain through artificially generated electrical signals however if the user simply requires communication or control of an external device then non-invasive BCI could be the solution. Non-invasive BCI are accessible to all, reading the electroencephalogram (EEG) signals from outside of the skull, usually through wet sensors, and provide a communication path between the brain and computer.

Although this is well established field of research the real world applications for a non-invasive approach has focused more towards meditation like application such as those offered by the company Emotiv, or the domains of virtual reality (Anatole et al., 2008) and gaming (Nijholt, 2009 and Rossini et al., 2009).

To briefly cover the history of BCIs, research in this area began in 1973 by Vidal who was researching into “direct brain-computer communication” (Vidal, 1973) which progress into his work on “Real-time detection of brain events in EEG” (Vidal, 1977) and provoking interest in DARPA. With this new found interest and with his BCI showing promising results for areas such as medical applications and potential military operations, research was undertaken on all the different forms of augmentation it could and would be integrated into the main focal point of prosthetics.

Up until the 1990’s research did not progress much at which point there was a breakthrough in the research with the pioneering research done by Wolpaw et al. who produced an alternate approach to a BCI system using electroencephalograms (Wolpaw et al., 1990).

Following this advancement another communication approach by Wolpaw et al. was designed for a system with a cursor on the screen which the subject would control with thought (Wolpaw et al., 1991).

In 1998 Farwell and Donchin designed another approach to use “the P300 component of the event-related brain potential (ERP)” (Farwell and Donchin, 1998) in order for individuals with motor related disabilities to have a form of communication. This system required the user to concentrate on screen filled with letters, each flashing at a different frequency, at which point the computer would, depending on the P300 signal received, detect the chosen letter.

Levine et al. explored “a direct brain interface based on event-relate potential” (Levine et al, 2000) with this research presenting a system capable of equal accuracy as the other current communication systems.

## 1.2 Related Research

The idea that EEG BCI can improve the daily lives of some if not all patients is a common idea. Höhne et al. presented “a BCI system designed to establish external control for severely motor-impaired patients within a very short time” (Höhne et al., 2014). This study included patients of various degrees of disabilities with two being in a locked-in state. This study showed how valuable and feasible a BCI system for these types of disabilities are as “within only six experimental sessions, three out of four patients were able to gain significant control over the BCI” (Höhne et al., 2014) and they further explain that their system could outperform some of the best assistive technologies these patients were using.

Researchers such as Bougrain et al. performed studies on “decoding intracranial data recorded in the cortex of a monkey and replicates the associated movements on a JACO robotic arm by Kinova” (Bougrain et al., 2012). This research used the OpenViBE platform as a foundation to create their own interface to record and process signals, to classify the extract the desired features and turning them into commands, finally creating a feedback system for those commands.

Qin et al. performed “classification of motor imagery for brain-computer interface applications, by means of source analysis of scalp-recorded EEGs” (Qin et al., 2004) and achieved a classification rate of around 80% from their subjects.

Johnson researches into an “EEG based BCI that uses steady-state visual evoked potentials on a healthy individual to asynchronously control a 6-degree of freedom robotic arm through custom software in real-time with high accuracy (Johnson, [no date]). This study is aimed and implementing a BCI for healthy human rather that for medical related reasons but uses a system that could be slow for a control system of external device such as a prosthetics due to the nature of Steady-State Visually Evoked Potentials (SSVEP) which requires the user to look at a symbol or an area of a screen that is flashing repeatedly causing a specific signal in the brain that will be recognised by the system and perform a task.

## 1.3 Aims and Objectives

The main aim of this thesis is to use a consumer grade headset to acquire raw EEG signals and process this data in order to identify and classify the intent of the user. This intent will then be used to control a prosthetic. The task has been broken down into the following:

* Acquire and record EEG signals
* Filter of EEG signals
* Train a classifier
* Analyse performance and accuracy of classification
* Perform a real-time scenario with feedback
* Perform a real-time analysis
* User intent to control a prosthetic

To complete these aim and objectives this thesis will create and test a prosthetic, create a prototype system to demonstrate the feasibility of a BCI system controlling the prosthetic and finally a fully function BCI system will be created that will meet all aims and objects with no omissions.

## 1.4 Motivation

This thesis will aim to design a working prosthetic hand that uses non-invasive EEG based motor imagery and will provide the user the ability to control the prosthetic hand using motor imagery signals. To do this the acquired EEG signal will be processed and classified to produce the highest accuracy possible that will be used as an output for control.

The overall motivation is to create a for user that suffer from motor neuron disabilities such as locked in syndrome, a medical condition where the majority of the face and body are paralysed and would leave the user with little to zero communication or control of the real world. Through an EEG based prosthetic they could regain some of that control and interaction.

The project will aim to be as affordable as possible utilising 3D printing for he prosthetic and commercial grade EEG headsets to show that this affordable system can still achieve excellent results even without more expensive medical grade headsets.

## 1.5 Scope of Thesis

This study will attempt to acquire, process and implement of a BCI for the control of a prosthetic arm. This begins with Chapter 2 introducing the biological source of the signal with Section 2.1 exploring how to acquire the desired response signals in an individual.

There are various approaches to acquiring EEG signals and in Section 2.2 EEG is introduced. This signal is investigated and critically reviewed for its ease of acquisition and how it can be acquired, what real world application and research it is currently being used for and the robustness of the signal when implemented in these scenarios.

Section 2.3 and Section 2.4 discusses the feasibility of motor imagery being used to train EEG systems and considers about the individuals that are BCI illiterate and unable to adapt.

Chapter 3 introduces the methodology of the research and the flow of processing the signal in order to create the BCI system. Section 3.2 presents the headset and explains how it acquires signals by Section 3.3. Section 3.4 highlights the key parts of the processing mechanism and produces an exhaustive report of these processing methods used shown in Section 3.4.5 to refine, extract features and classify any incoming signals. Section 3.5 introduces the microcontroller that will be used to control the prosthetic of Section 3.7.

Chapter 4 will present the steps of implementing this project with the testing of the prosthetic with serial monitor inputs to Arduino in 4.1. 4.2 present a fully functional but simple BCI system that can control the prosthetic. 4.3 introduces the second BCI system that has demonstrates the step by step acquisition and signal processing tasks needed to create a BCI system both offline and real-time ability to control the prosthetic.

Chapter 5 produces an evaluation of the results produced during the implementation of Chapter 4 and is split into the same subsections as Chapter 4.

Finally Chapter 6 provides and overall conclusion of the project including the methodology used, the limitations and BCI illiteracy of users when using the two systems, the robustness and the real world or clinical applicability of the created BCI systems finishing with a final conclusion of the project. Section 6.6 discusses the future work of this project in order to improve and create a highly accurate system.

# Chapter 2 Preliminaries

## 2.1 Structure of the Brain

Motor imagery is a mental process in which the individual simulates a given action in their mind. Hema et al. describe it as “the ability of an individual to control his EEG through imaginary mental talks enables him to control devices through a brain machine interface” (Hema et al., 2009). This study continues explaining that the process involves “preparation for movement, passive observations of action and mental operations of motor representations implicitly or explicitly” (Hema et al., 2009).

Miller et al. discuss the importance of motor imagery and “demonstrate the role of primary motor areas in movement imagery” (Miller et al., 2010). As shown in Figure 1 Miller et al. discuss how motor imagery relies on the same brain systems that are using in actual performance of movement actions (Miller et al., 2010).

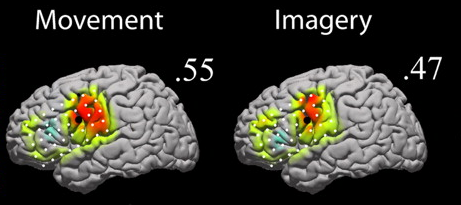


Figure Motor Movement and Motor Imagery (Miller et al., 2010)

## 2.2 ElectroEncephaloGram Signal

### 2.2.1 Introduction

ElectroEncephaloGram (EEG) is a recording of brain activity acquired using sensors on the scalp that measure the voltages fluctuations resulting from ionic current flows within the neurons of the brain. These signals are recorded and processed in order to be analysed and to extract the important features of them. Usually the headset used to acquire the EEG signals will have electrodes covered in saline or conductive gel as this makes the electrodes more sensitive and therefore will be able to gather better EEG signal data.

The types of headsets can be split into three groups; the top tier is medical grade which are the best as they gather the best quality signals however they are very expensive, the middle tier is the commercial grade headset which have a lower number of electrodes than the medical grade and are more affordable but could be considered expensive compared to the bottom tier openEEG which can vary on design and how many electrodes. As they are the bottom tier they could be considered the worst for acquisition of signals and to use for a BCI system but they are also the cheapest and tend to have open source hardware.

Some researchers focus on the calibration of each acquisition for each subject such as the study by Ang et al. which discusses the lack of a “direct objective measure to determine if a subject is performing motor imagery correctly” (Ang et al., 2011). This can be measure to a certain degree using confusion matrixes or accuracy measurements however one classification won’t work for everyone and therefore calibrated to an individual. Furthermore when considering the effectiveness of using EEG as an approach for “Although the proof-of concept was given decades ago, the reliable translation of user intent into device control commands is still a major challenge” (Ang et al., 2011).

These EEG signals can be formalised as equation (1) where E is the EEG signal, N is the number of trails, C is the number of channels used and T is the range of time domain. However in order to be able to apply classification to the EEG signal it must be turned into a feature vector using the equation (2) and resulting in the problem of how to extract the desired features in order to receive the best classification for that individual. This will be further discussed in Section 3.4.2.2 and Section 3.4.2.3.

(1)

(2)

### 2.2.2 EEG Artifacts

Within signal data there can be unwanted noise and artifacts that are a result of interference such as facial muscle contraction, blinking or poor contact between the scalp and the sensor. As a result filtering EEG signals to remove these distortions is important prior to other processing tasks such as classification. For this reason EEG signal can be processed through multiple filtration, analysis and/or processing stages in order to filter the signal and remove the unwanted noise.

The reason for doing all of this is that artifacts and other signals pose possible issues for the accuracy analysis of the EEG signal. Only when the EEG signal has been correctly processed will the data be ready to be used by a Brain Computer Interface System.

## 2.3 Brain Plasticity

Brain plasticity is the ability for the brain to adapt to a situation for example a head trauma would require remapping of the brain or if a limb is lost it could be replaced with a prosthetic. It is this plasticity that allows the opportunity to research into neuroprosthetics for those that require them and is an ability that can occur throughout a person life, at any age or any type of trauma small or large, psychological or physical.

Ungerleider et al. investigate the brain plasticity during motor skill learning and looking at the “dynamic neural changes that occur in the motor system during the different phases of learning” (Ungerleider et al., 2002). This study presents their investigations into how the motor cortex can change during different types of learning.

## 2.4 BCI Illiteracy

One of the biggest challenges in BCI research is to understand and solve the problem of ‘BCI Illiteracy’ which is the issue that there are individuals that struggle to, or cannot, control BCI devices. This is discussed by Vidaurre and Blankertz where they state that an “estimated 15 to 30%” (Vidaurre and Blankertz, 2009) are BCI illiterate. They illustrate the key issues with users that have BCI illiteracy and present their approach and techniques to aid or solve this problem. They state that by “using this approach, which does not involve any offline calibration measurement, good performance was obtained by good BCI participants (also one novice) after 3–6 min of adaptation” (Vidaurre and Blankertz, 2009). This research offers new insight into BCI systems that could work for the up to 30% individuals with BCI illiteracy.

# Chapter 3 Methodology

## 3.1 Overview

In this chapter the chosen hardware and software will be presented and discussed, how they have been used and the underlying information of how they work will be shown. Figure 2 shows the overall design of the project, consisting of the headset, software platform, code and the assembled 3D printed prosthetic.

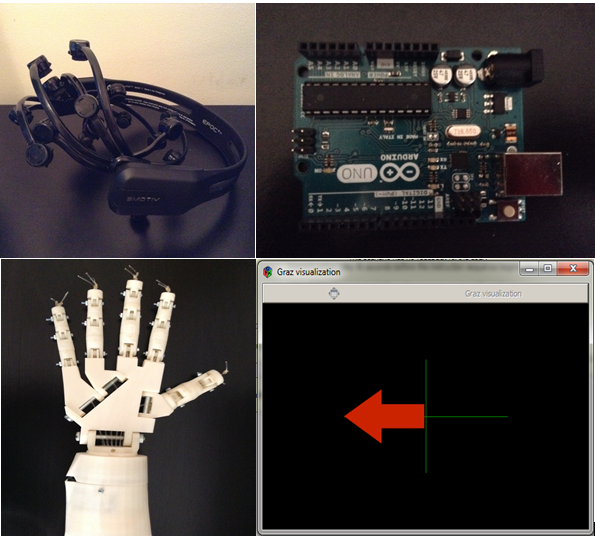


Figure Prosthetic Arm: Discussed in Section 4.1

The chosen headset Figure 3 and the software Cognitive Suite in 3.3 and OpenViBE in 3.4 and the Arduino in 3.5 and the prosthetic in Section 4.1 are the main components to the project we will introduce.

## 3.2 EEG Headset

The chosen headset for this thesis will be the Emotiv EPOC+ premium, a commercial scientific contextual EEG headset that will be used to record EEG signals from the brain in real-time while performing mental tasks and is shown in Figure 2. Compared to other medical grade headsets the number of electrodes may be limited however in the commercial market it is one of the best and often chosen for motor imagery related research.



Figure 3 Emotiv EPOC+ Headset

The EPOC+ headset is limited to the following 14 electrodes as shown in Figure 4; AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, and AF4. The number of electrodes on the front of the scalp is why it is chosen for motor imagery research. If the research was more visual focus such as SSVEP then the headset could simply be put on reverse so that the majority of electrodes are on the back of the head which processes the visual signals. Along with this is the Research Edition SDK that will be used by the used software to acquire the EEG signal data from the headset and is the needed SDK for the chosen software platforms that will be discussed in Sections 3.3 and 3.4.

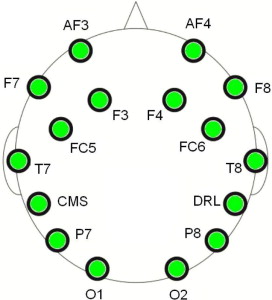


Figure 14 Labelled Electrodes of the Emotiv EPOC+ Headset

## 3.3 Cognitive Suite

For the first BCI system the Emotiv software Control Panel will be used to acquire the signals then the Cognitive Suite will be used to train and control the prosthesis. The Cognitive Suite is part of the control panel and has been shown in Figure 5 and is essentially a floating cube on the left side of the figure that can be manipulated and maneuverer into different directions using the actions that can be trained on the right side of the figure. A more detailed explanation of implementation of this is in Section 4.2.

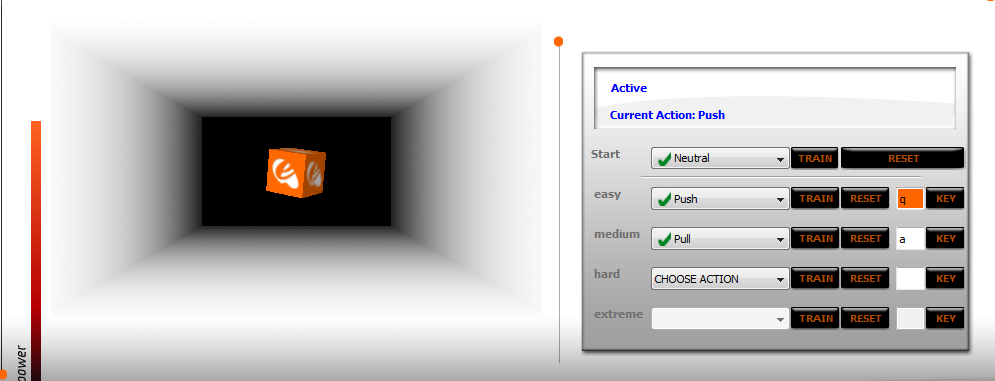


Figure Emotiv Cognitive Suite

## 3.4 OpenViBE

The main chosen software to use is OpenViBE due to it being an open source, stand-alone software that can be used for quick and robust prototypes or BCI systems with a main focus on virtual reality and gaming. In the following subsections each component that will be used to create BCI system is explained.

### 3.4.1 Overview

OpenViBE is an open-source “software platform which enables to design, test and use Brain-Computer Interfaces” (Renard et al., 2010) and is developed at the French National Institute for Research in Computer Science and Control (INRIA). The platform consists of a set of software modules called boxes that allow a user to design and develop functional BCIs with a focus on virtual reality applications.

Consequently the developer of this platform have made it possible for it to be integrated into various application fields ranging from medical such as assisting individual in their daily life’s particularly disabled people, real-time biofeedback for medical analysis , and real-time diagnosis for analysis. One of the key focuses of the developers and researcher is to have a BCI for virtual reality or gaming and is starting to show a lot of interest in the controlling of robots.

### 3.4.2 System Features

#### 3.4.2.1 Modularity and Reusability

The OpenViBE platform essentially modules called boxes that each serve a single purpose or task to do related to signal processing for instance they could involve acquisition, filtering, classifying and visualization of cerebral data or scenario tasks such as motor imagery. These boxes can also offer interaction with external applications for example TCP/UDP. A large benefit to having the box concept is the robustness of the system. Boxes can be added or removed or edited and users are able to make any changes they desire easily.

#### **3.4.2.2 Types of Users**

OpenViBE is design in such a way that it can be used by various users regardless of experience programming or not. This means it can be used by Medical professional, virtual reality designers, researchers in robotics or signal processing or even neurologists for analysis. The platform can be split into four different categories of users. The developer and the application developer are both programmers; on the other hand the author and the operator do not need any programming skills.

The Developer will use the platform to program, design and test new functionalities that usually are plug-in type implementations whereas the application developer creates standalone application such as virtual reality that a BCI user would interact with.

The Author is not a programmer but instead uses the existing scenarios to create new scenarios usually through reconfigurations. The operator who is also a non-programmer and would more likely be a clinician or practitioner who has some knowledge of neurophysiological signals and would simple run the existing scenarios perhaps for one of the other users.

#### **3.4.2.3 Portability**

As OpenViBE operates independently of software targets and hardware devices it is able to with the majority of acquisition hardware such as EEG headsets and on various operating systems such as Linux, Windows or Mac.

#### 3.4.2.4 Supported Devices

There is an extensive range of supported hardware that can be used to connect with OpenViBE however the limiting factor is whether the integrated drivers are up to date which depend on the version of OpenViBE being used.

#### 3.4.2.5 Plug-ins

Three types of plug-ins are used the driver plug-in which provides additional acquisition devices to the acquisition server though the devices SDKs or a physical connection with the device. The second plug-in is for the algorithms and it allows developers to freely create and add in any new algorithms. Lastly the box plug-in will usually rely on the algorithm plug-ins to create different signal processing functionalities for the boxes.

#### 3.4.2.6 Tools

Boxes that represent processing module can be connected and be functionally correct without the user needed to write a single line of code. The author user will have access to existing modules in a panel to the right of the platform that can be dragged onto the scenario window. Each box has inputs at the top and outputs at the bottom. The boxes are connectable through their inputs and outputs. Double clicking on a box opens up the configuration panel to allow for any changes to be made or configuration to be altered.

An embedded player engine allows the author to test and debug their scenario in real-time. The Acquisition Server provides a generic interface to various kinds of acquisition device systems. Server connection to the hardware is dependent on manufacturers’ specifications: some devices will be shipped with a specific SDK such as the Emotiv’s research SDK. The Designer is used mainly by the author and helps to build complete and functionally scenarios based on existing software modules for use of acquisition, classifying or visualisation.

#### 3.4.2.7 Existing Scenarios and Tutorials

The platform comes with a large amount of existing scenarios to act as tutorials for the user. They provide the user with an overview of the scenario or boxes functionality, ranging from signal acquisition, filtration, classification, visualization and rendering. There is also clear documentation online for users to read in their own time for more information about the boxes used or the relevant algorithms behind the box.

### 3.4.3 Installation and Compatibility

OpenViBE can be installed on computers running Windows or Linux for both 32 and 64-Bit. However smooth running on 64-Bit is not guaranteed for Windows as some drivers have yet to be included. It is advisable to use a 32-Bit system for example, if using the acquisition server for an online application.

### 3.4.4 Connecting the Server and the Designer

The purpose of OpenViBE is to acquire data from one of the supported EEG devices through the Acquisition Server and send it to one or more clients for recording or processing. In our experiments the client will be the OpenViBE designer, responsible for hosting the main part of the BCI application. The following subsection will explain the configuration that can be done on the Acquisition Server and how that will affect this thesis.

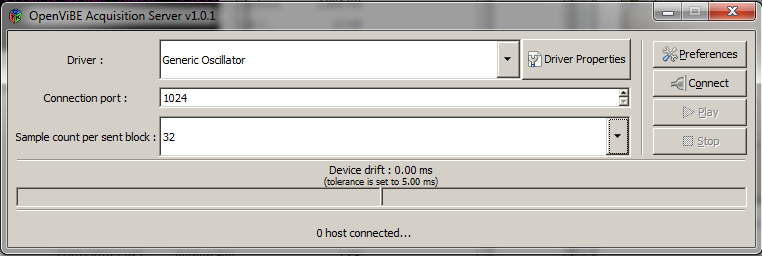


Figure 6 Acquisition Server GUI

#### 3.4.4.2 Channel Selection and Naming

Which channels will be used is an important thing to specify in order to remove as much noise as possible from unneeded electrodes and the number of channels used in the acquisition is correct. When connecting to a driver the default number of channels will be selected for instance the EPOC+ will automatically set to 14 channels with the additional option to add two more for the gyroscope that is inbuilt to the headset. If any of the electrodes are unneeded or will be unused then they must be removed from amplifier. Naming the channels using the driver properties setting (Figure 6) will allow for an easier signal analysis and identification however is optional and can be left as default.

#### 3.4.4.3 Sampling Frequency

When the EPOC+ acquires the EEG signal it will be received as analog and will need to be digitalised. This conversion is performed automatically using a multichannel analogue-to-digital converter (ADC) and the sampling rate must be enough to represent the change in the analogue signal which is essentially multiplying the input signal with a series of Dirac-impulses and can be show with equation (3) where is the input signal, the sampled signal, a Dirac-impulse and the period of time. We can times signals in the time domain which results in a convolution in the frequency domain and is shown in equation (4). Finally as there will be scenarios where both time and frequency will be in the sampled signal hence equation (5) should be used to avoid overlaps in the datasets.

(3)

(4)

(5)

Failure to adhere to this condition results in a permanently distorted signal. However in a multichannel recording, setting the sampling frequency too high will lead to significant increase in memory use over time.

With all these considerations and with the chosen headset using all 14 channels the sampling was increased from the default of 32 into 128 which is the maximum amount of data that can be sent during acquisition with the EPOC+.

#### 3.4.4.4 Filtering

Filtering is another important step in signal processing and is used to eliminate unwanted noise from incoming signals that were not already from sampling and to minimize artifacts in the remaining signal and depending on the filter chosen results may vary. The aim is to improve signal quality through minimizing any background noise or external interference from the user or other devices.

Filtering can come with an inconvenience in that they can be remove potentially valuable data when filtering and if used incorrectly can remove the EEG signal completely. The two popular filtering techniques that are used are spatial filtering and temporal filtering.

Spatial filters take the data from two or more recordings and locate features of specific characteristics. One example of a spatial filter is the Common Spatial Pattern (CSP) filter shown in BCI systems (Ramoser et al., 2000) or in more individual specific filtering research by Gruger et al. (Guger et al., 2000). CSP Spatial Filtering will need to be used in this project as a normal spatial filter will not work with the EPOC+.

Temporal filters use methods like band pass filtering and Fourier analysis for example are the most common EEG filters and can be split into the following four categories; low pass, high pass, band pass and notch pass. Low pass filters are used for high frequencies that are attenuated whereas high pass filters are used for low frequencies that are attenuated. Band pass filters are the passes in a user specified frequency range only. Finally a notch filter rejects just one specific frequency and lets the remaining frequencies pass.

Each of these filters can be split into different filter methods such as Butterworth or Chebychev as well as being able to select the upper and lower pass band edge in Hertz (Hz). Furthermore Chebychev allows for the pass band ripple to be selected which is fluctuations of a pass band measure in decibels. Taking into account these options this project will be using the filter settings as shown in Figure 7. Limiting the Hz to between 8 and 30 removes the higher frequenting EEG signal that won’t be needed in the motor imagery project.

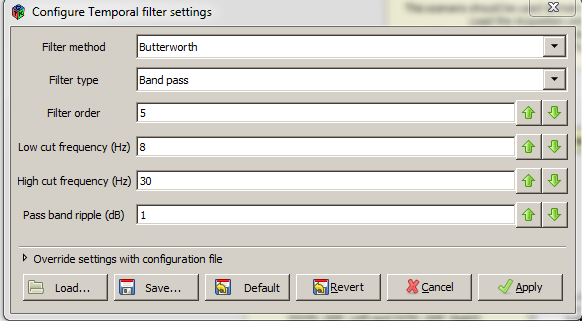


Figure Temporal Filter Settings for System Two- OpenViBE

#### 3.4.4.5 Averaging

Averaging is a process that involves enhancing components from signals whilst there is still considerable amount of noise. For a measurement *m* consisting of signal *s* and noise *n* over N trials, represents the *kth* sample point in the *jth* trial. The mean of the N sampled trials is given by equation (6).

(6)

To reduce noise, we have to choose a N large enough so that . From equation (6) we can derive the variance with equation (7), which indicates that the estimate of s in the average improves with a factor of

(7)

### 3.4.5 Designer scenarios

#### 3.4.5.1 Acquisition Scenario

The acquisition scenario involves receiving the raw signal data from the acquisition server discussed in Section 3.4.4. Figure 8 demonstrates a typical design that will process the raw signal and in combination with the ‘Lua Stimulator’ provides cues for the user to perform a motor imagery related task. The ‘Lua Stimulator’ uses the LUA Programming Language due to it being “a powerful, fast, lightweight, embeddable scripting language” (LUA, 2015). Once the specified numbers of cues have been completed the new signal data and the stimulation codes will be save together in an .ov file.

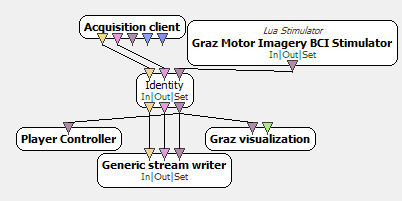


Figure 8 Acquisition Scenario for System Two- OpenViBE

#### 3.4.5.2 Common Spatial Pattern Trainer Scenario

As mentioned briefly before the scenarios will need to use a Common Spatial Pattern (CSP) which was firstly used to improve the separation of two types of input signals. The main idea behind the CSP special filter algorithm is to use a linear transformation to demonstrate the EEG data in a low-dimensional spatial space with a projection matrix. The rows of this matrix are the weights for channels and this transformation can maximize the variance of two class signal matrices. CSP is based on the diagonalization of the covariance matrices of both classes which will be examined and analysed in the Section 5.3.2 to compare performances of different CSP filters.

This algorithm is going to be described using a scenario that will classify left and right movements similar to this thesis. and are the pre-processed EEG matrices under two classes (left and right) with , where *N* is the number of channels and S is the number of samples per channel. The normalized spatial covariance is computed with the equations (13) where is the transpose of *c* and trace(*E*) computes the sum of the diagonal elements of *E*.

(13)

After computing the spatial covariance, the averaged normalized covariance and are calculated by averaging all the trials of each group. Now the composite spatial covariance can be computed with the equation (14) where is a matrix of eigenvectors and ∑ is the diagonal matrix of eigenvalues.

(14)

Therefore the scenario in Figure 9 will be used to train the CSP spatial filter offline in order to remove the higher frequencies that won’t be needed with the motor imagery task and divide the remaining signals into the two groups left and right based on their stimulation codes and into four second chunks. Once this processing has finished a configuration file should be created that can be used in other processing scenarios to perform the filtering task taught here.

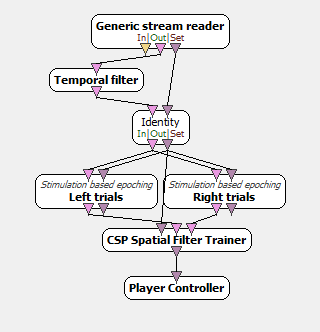


Figure 9 CSP Trainer for System Two- OpenViBE

#### 3.4.5.3 Linear Discriminant Analysis Classifier Scenario

After considering the study by Lotte et al., which investigates the classification algorithms of BCI systems, they discuss how linear classifiers “are probably the most popular algorithms for BCI applications” (Lotte et al., 2007). OpenViBE offers the two popular types of linear classifiers Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM).

In this project the classifier for the ‘Classifier trainer’ will be LDA that will “use hyperplanes to separate the data representing the different classes” (Lotte et al., 2007) and therefore expects there to be two different inputs a negative class and a positive class for instance and .

The scenario shown in Figure 10 can be used to train the classifier to detect left and right imagery movements. A detailed explanation of each component will be illustrated in Section 5.3.3.

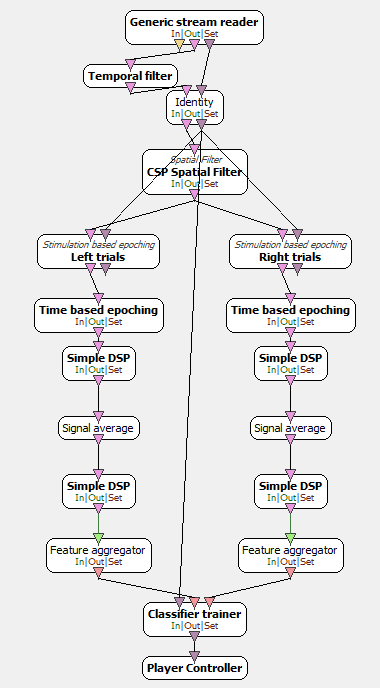


Figure 10 Classifier Trainer for System Two- OpenViBE

#### 3.4.5.4 Online Scenario

Figure 11 is a typical online scenario containing the previously trained CSP spatial filter and LDA classifier. The online scenario will be a foundation to creating a better real-time scenario. It will provide the user will a similar scenario to the acquisition scenario but this time will be able to classify the signal into left or right in real time and provide the feedback to the user of which direction the user is thinking.

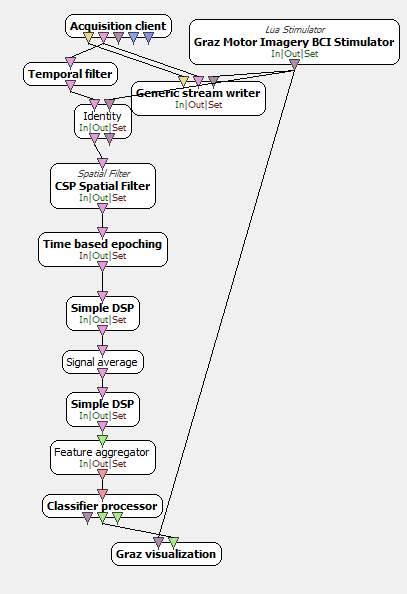


Figure 11 Online Scenario for System Two- OpenViBE

## 3.5 Arduino

The Arduino UNO shown in Figure 12 is the chosen microcontroller of the prosthetic arm and will be able receive inputs from the ‘Serial Monitor’ window which will, based on the code, send a command to the appropriate servos to perform an action such as open hand or close hand. It has been chosen due to being open-source and is basically a micro-computer that can be used in control type projects.

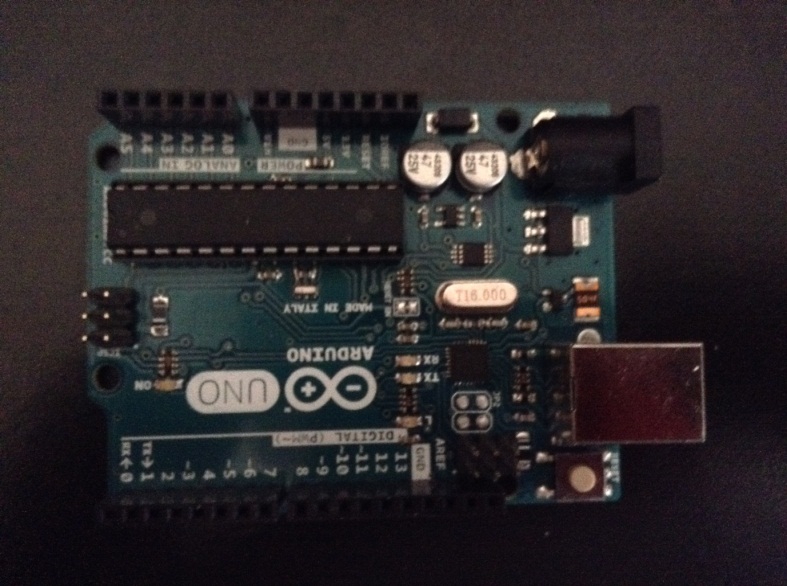


Figure Arduino UNO

### 3.5.1 Arduino Software

The software for Arduino is completely open-source and is an Integrated Development Environment (IDE). The Arduino IDE can provide an easy way for writing code, called a sketch, and uploading the code to any Arduino board.

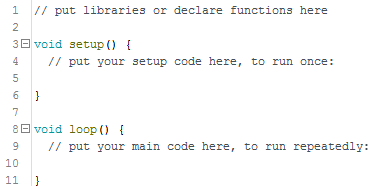


Figure Basic Structure of the Arduino Code

Figure 13 shows the key parts of a sketch. A typical sketch can be split into three parts. Part one will be were any libraries, variable or hardware will be declared or initialised. Section two is run at the being on the program to setup the code and section three will repeatedly loop so long as there is power available. Additional methods can be input after part three if the user wants to have methods that are not influenced by the loop() statement, for instance

## 3.6 Python Software

“Python is an interpreted, object-oriented, high-level programming language with dynamic semantics” (Python 2015). It allows for the

OpenViBE uses a Python plugin to allow users to create their own scripts that will process or produce data to and from the OpenViBE scenario for the stream types signal stimulation and streamed matrix. The scripts for OpenViBE follow a similar structure of Arduino in that there are three parts; part one will initialise variables and is called once at the start of the scenario, process if a loop and uninitialized is called once when the scenario ends.

## 3.7 3D Printed Arm

Current prosthesis can be very expensive and by 3D printing a much cheaper but still functional version, allows for customisation of prototypes to a user’s specific design quickly and affordably. This 3D printed arm will be what this project uses for a prosthetics when testing the BCI systems.

# Chapter 4 Implementation

## 4.1 3D Printed Arm and Arduino

The chosen design (Figure 14) is a slightly older version of the InMoov by Gael Langevin (Thingiverse, 2012) that was 3D printed with the intention of being used as the prosthetic.

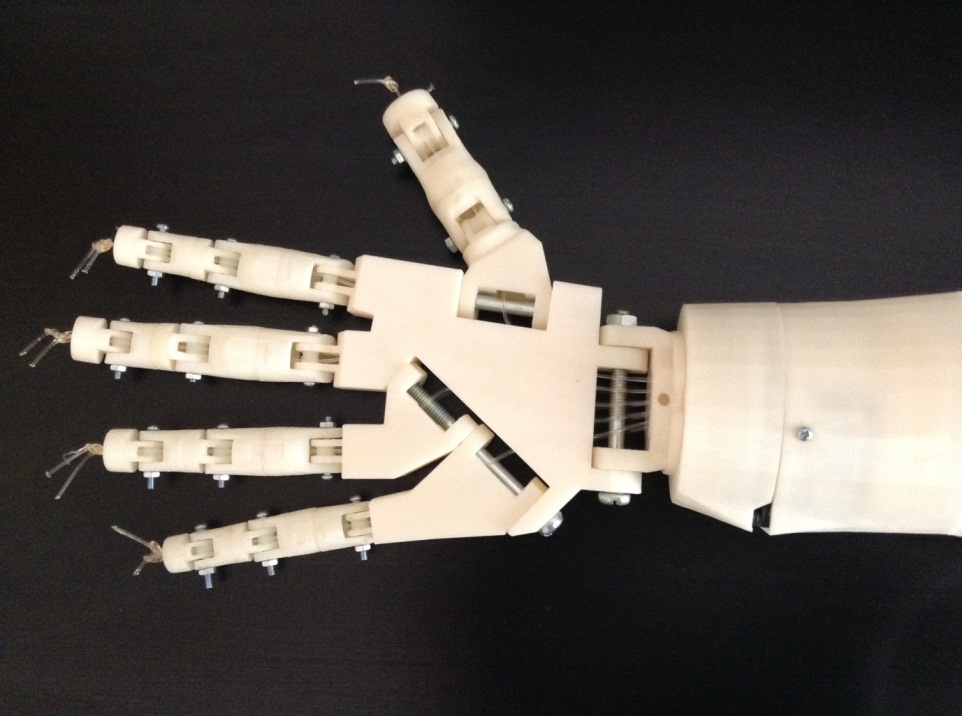


Figure 14 3D Printed Arm

To control the 3D printed arm an Arduino UNO has been used to control five servos, with each servo controlling a finger or thumb. The Arduino code will continuously wait to receive input to the serial monitor which once received will process the input character and move the servos accordingly. Figure 15 shows the Arduino code, a simple switch case that depending on the input will move the fingers by a set amount of degrees, additionally there are some methods for moving all fingers at once such as to open or close the hand. With this setup the Arduino has been used to successful in moving the fingers individually or into an open hand and closed hand positions. Full code is in Appendix A.

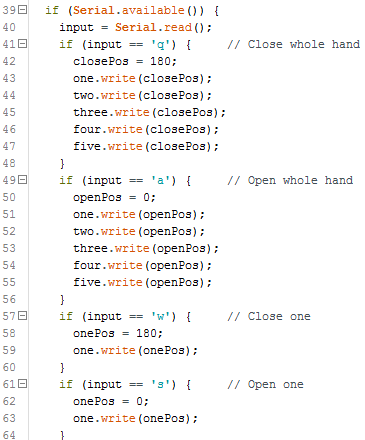


Figure Sample Code for Controlling Prosthetic

As the code needs to only be simple and functional, the serial monitor will be read straight into loop() and ‘if’ statements will parse through the data and make and motion depending on the character.

## 4.2 System One – Cognitive Suite

Following successful testing of the arm and in order to investigate the feasibility of an EEG based prosthetic, a smaller and simpler project was first implemented. In this project the Control Panel described earlier is used to acquire the EEG signals once the electrodes are sufficiently covered in saline and placed on the head correctly the headset can receive EEG signals. As shown in Figure 16 which is from the Emotiv control panel to show the strength of the signal for each electrode, the connectivity is all green meaning the best it can be.

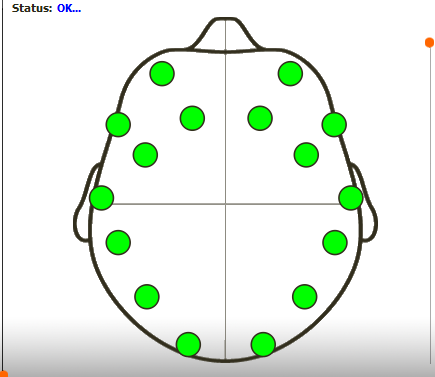


Figure Emotiv Full Strength Signal Sensors

These signals are then sent via a USB to the computer at which point the Cognitive Suite uses them to train the actions for the cube (Figures 17, 18) and to output an associated character to the Arduino UNO which would, depending on the character, manoeuvre the 3D prosthetic similar to the testing with the full code is in Appendix A.

The Cognitive Suit allows the user to create and train many different actions that can each perform a different action and a feature of the cognitive suite is that it will output the assigned key to whichever window is in focus for instance in Figure 18 the cube is being pushed and the assigned key ‘q’ is being broadcast. This means that if the focus window is the Arduino’s serial monitor it can easily and directly input letters which when received by Arduino will run it through the code and perform the finger or hand motions.

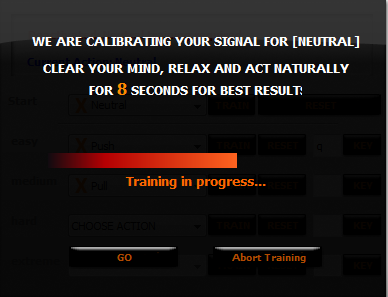


Figure Training the Neutral Command

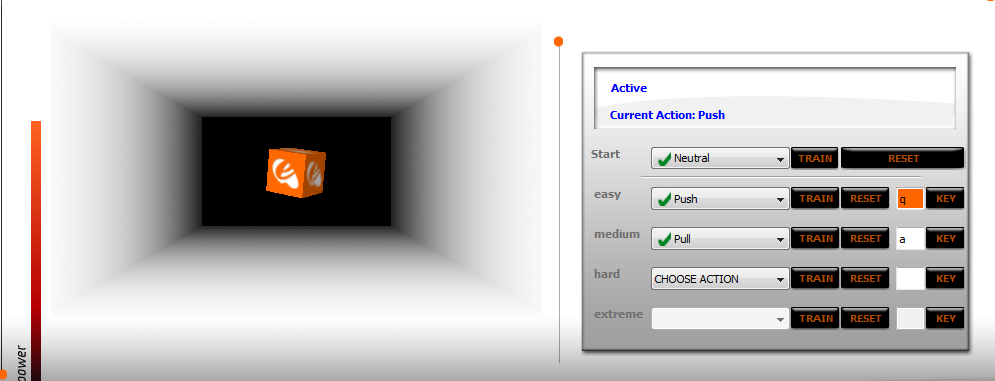


Figure Cognitive Suite showing the cube being pushed and the assigned key broadcasting

## 4.3 System Two - OpenViBE

### 4.3.1 Acquisition

Using the same Emotiv Control Panel to check the electrodes signal strength the OpenViBE acquisition server will take those signals and pass then onto any scenario that has an acquisition client. At this point the OpenViBE Acquisition Sever can be used to connect the incoming EEG signals with the OpenViBE platform Figure 19, which shows that successful connection with the headset and the amount of drift being received, which most will be automatically corrected to a certain degree. The acquisition server for OpenViBE support the uses of the Emotiv EPOC however in order to receive the EEG signals it must be configure so that it knows the path for the Emotiv Research SDK that comes with the headset on purchase. As mentioned the Emotiv EPOC+ sends 128 samples of data to the computer the Acquisition server has been changes from the default 32 to match the EPOC+.

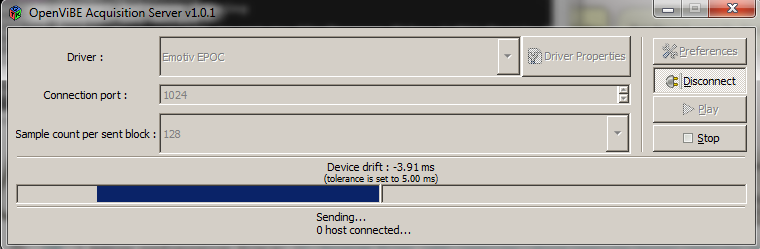


Figure OpenViBE Connected with EPOC+

The Acquistision.xml will present the user will a GUI (Figure 20) that every four seconds will go through the process of a black background, followed by a green cross to indicate a new cue is going to appear and finally the cue in the form of a red arrow that points either left or right. When the cue appears the user must imagine movement of that particular direction until the next green cross appears. This ensures that there will be enough samples per cue for the duration of the acquisition. The more sample the system has for further classification the better the result will be for accuracy.

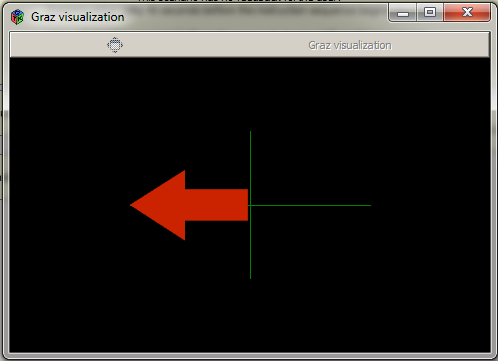


Figure Acquisition.xml GUI

This stage can be considered the training for both the BCI and the user. The more this scenario is run the better the performance of the user from practice and the better the result the BCI will acquire for further processing.

Two experiments were made one with 20 cues and one with 40 cues. It takes roughly 4 minutes for 10 cues therefore the two experiments took 8 and 16 minutes respectively. Due the amount of cues, scenario two can take 16 minutes and is the longest part of the training of the system. This could become very taxing for the user to sit and concentrate for 16 minutes straight however to test if the accuracy of the system will increase with the number of cues which provide more samples were also increased. Once the scenario has finished all of the raw data will be saved into OpenViBE .ov file structure called Acquisition.ov and can be used in any other scenario.

### 4.3.2 Training CSP

Due to the choice in EEG headset being the Emotiv EPOC+ a normal motor imagery scenario cannot be used as it does not have a reference channel or require the channels to be selected for the experiment like a research EEG cap would with a larger amount of electrodes, instead using the CSP spatial filter a scenario can be created that will allow for the filtering of the incoming signals.

After the system has the required raw data the CSP\_Training.xml can be run which trains the CSP Spatial Filter. The process takes the .ov file, running it through a temporal filter with a large frequency band [8Hz – 30Hz] to remove the higher unneeded frequencies, and is split into the desired stimulation classes every four seconds as the original stimulation was presented to the user. At the end of this training the CSP trainer if successful will have trained the CSP filter to filter which signal the user intends to be left motor imagery and right motor imagery into the two classes.

Once the scenario has finished the CSP spatial filter is saved to a configuration file that can be used in scenarios to overwrite the default file so that if the user wants to filter left and right motor imagery of themselves it will be more accuracy as it was created based on their raw signals, however if the new scenario has further inputs such as motor imagery of the legs and arms then the CSP will filter may not distinguish between the different limbs. It would need to be retrained or for a second CSP filter to be trained to look for only legs. The scenario will also leave a message, Figure 21, telling the user that it has succeeded and extra information such as number of samples and the size of the sample set.

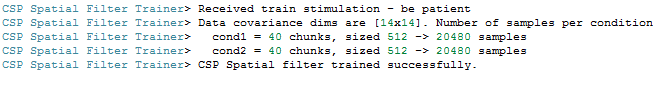


Figure CSP Spatial Filter Training Success Message

Unlike the first scenario this scenario does not require processing to be done in real time meaning that using the controls shown in Figure 22 the training can be increase from 16 minutes to 1 minute.



Figure Controls for the scenarios

### 4.3.3 Linear Discrimination Analysis Classifier

The LDA\_Classify.xml scenario is used to train the LDA classifier to detect and recognise the left and right arm movements. This scenario also uses the acquired EEG signal data and follows a similar design to the CSP trainer however in this scenario the previously trained CSP Spatial Filter is used therefore if the filter had been trained poorly or did not produce the correct stimulation classes then it will affect the performance and accuracy of the LDA classifier. In this study the CSP filter was trained successful and is used prior to the LDA feature extraction to reduce complexity in the signal Figure 23.

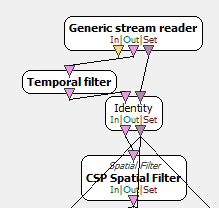


Figure 23 LDA Classifier Scenario Part 1

The remainder of the scenario is split into two columns, the first for left motor imagery and the second column for right motor imagery (Figure 24). Each column is configured to receive four chunks of signal half a second after the cue was shown in the first scenario. This removes the unwanted signal that does not represent a movement or excess data from the previous movement which could be either right or left. Each column is identical except that the ‘Left trials’ stimulation is looking for class 1 and the ‘Right trials’ is looking for class 2.

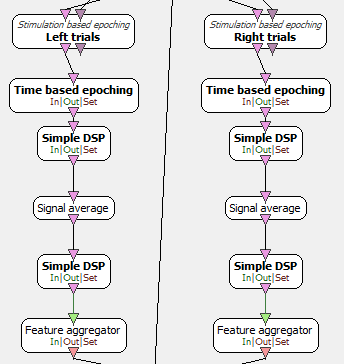


Figure 24 LDA Classifier Scenario Part 2

The rest of the signal processing chain is as follows. The ‘Time based epoching’ splits the signal into chunks of one second. It was decided to use the Simple DSP to apply mathematical formulae to each of the incoming signals and output the resulting altered signal (Figure 25).

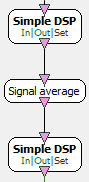


Figure 25 LDA Classifier Scenario Part 3

For the first Simple DSP the equation was used to square all samples, then signal average is used to compute the average of each incoming signal and output the resulting filtered signal containing those averages. Finally for the second Simple DSP the equation used was to compute the log of the square average.

Once this process has been completed the feature aggregator (Figure 26) will take each input, which is usually a stream of matrices, output them into a stream of feature vectors which can be used for classification.



Figure 26 LDA Classifier Scenario Part 4

Figure 27 shows the classifier trainer using LDA will classifier both streams of feature vectors into a configuration file differentiate them based on the stimulations input from the acquisition data file.



Figure 27 LDA Classifier Scenario Part 5

Finally the classifier outputs the classified data into a configuration file to be used in other scenarios such as the Realtime.xml scenario. Similar to the second scenario this scenario is also offline can also be accelerated for the training of the classifier.

Once the training is completed a message in the command line and OpenViBE message box will give information about the cross validation and the accuracy test as shown in the Figure 28. Figures 28 and 29 can be compared and show that the difference in sample results in a 10.2296% different in test accuracy with 84.5918% for Acquisition1.ov and 74.4388% for Acquisition2.ov. This accuracy gives an estimate of how the LDA Classifier will perform but does not guarantee it for practical usage.

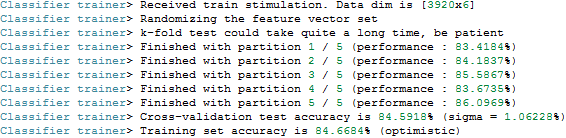


Figure Acquisition1.ov LDA Results

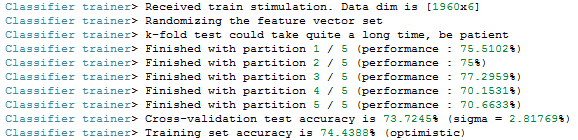


Figure Aquisition2.ov LDA Results

### 4.3.4 Experiment Analysis

For the purpose of examining and analysing the results from the previous scenario’s training and how well the BCI system will perform using performance metric of accuracy, Analysis.xml an offline visual replay of what has been recorded and trained in the previous three scenarios can be used (Figure 30).

In the lower half of Figure 30 the ‘Classifier processor’ sends the results to the ‘Graz visualization’ box for feedback which will be a blue horizontal bar showing the strength of the imagined movement. In order to display the correct feedback the box expect to have two values a negative for one class and a positive for the other class from the classifier processor.

The ‘Confusion Matrix’ in order to compute needs the instruction flow and the classification flow. The instruction flow is received from the acquired data and the classification flow from the ‘Classifier processor’. Once it has both of these inputs it is able to compare them for each class. The ‘Accuracy Measure’ is a real time classifier that compares the inputs from the result of the ‘Classifier processor’ and the targets received from the ‘Stimulation Filter’ as shown in Figure 30.

When run Figure 31 is presented that demonstrates the confusion matrix and accuracy measurements of each classification as the signal data is processed. As shown in Figure 32 the current direction has 0.67 of the result class matching the target class indicated by the top left box colour red. The overall accuracy of the LDA classify is 79.97% as shown in Figure 33, anything over 70% would be considered good with the higher being better. When this BCI was first trained it has an overall accuracy of 74% and through retraining of both the user and the systems this has increased to 84.6684 %.

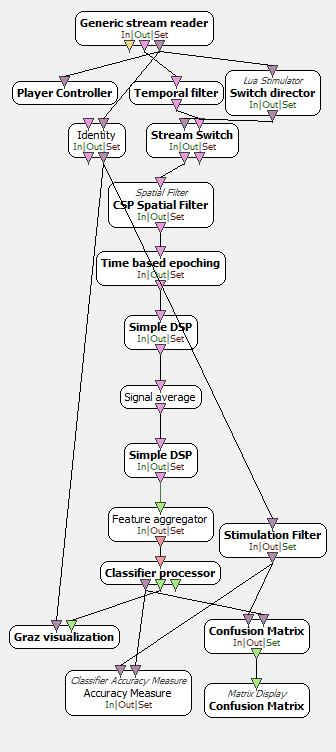


Figure Analysis.xml

By using this scenario the user is able to see a performance analysis of how well the systems can filter and classify the data. The input data could be the Acquisition1.ov or Acquisition2.ov data file or a file from the real-time scenarios explained in Sections 4.3.5 and 4.3.6. Depending on the data file used the corresponding CSP Spatial Filter and LDA Classifier should be used as each data file has different number of samples to analyse. Again this can be sped up or slowed down to see at specific point if the system performs well or if the user wants to simple look at the final measurement of accuracy.

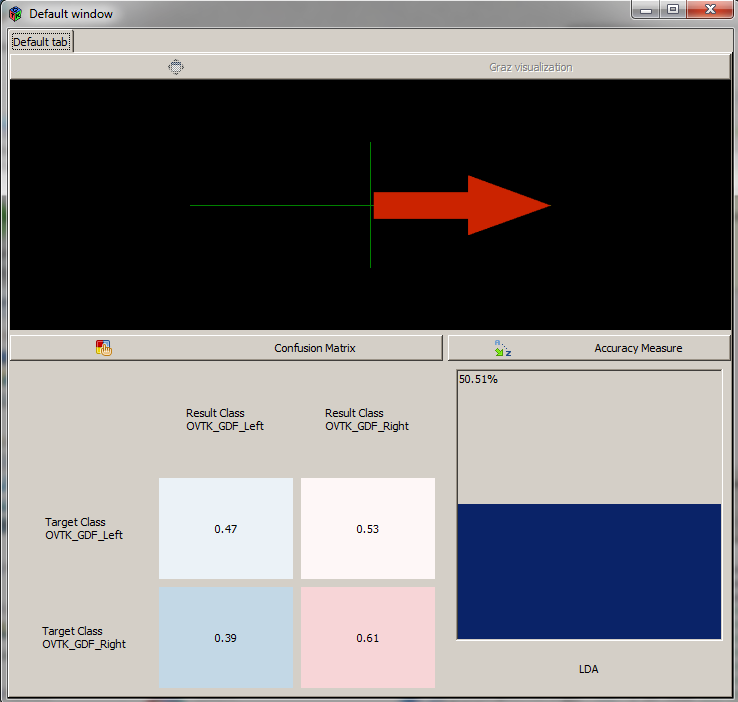


Figure Analysis GUI

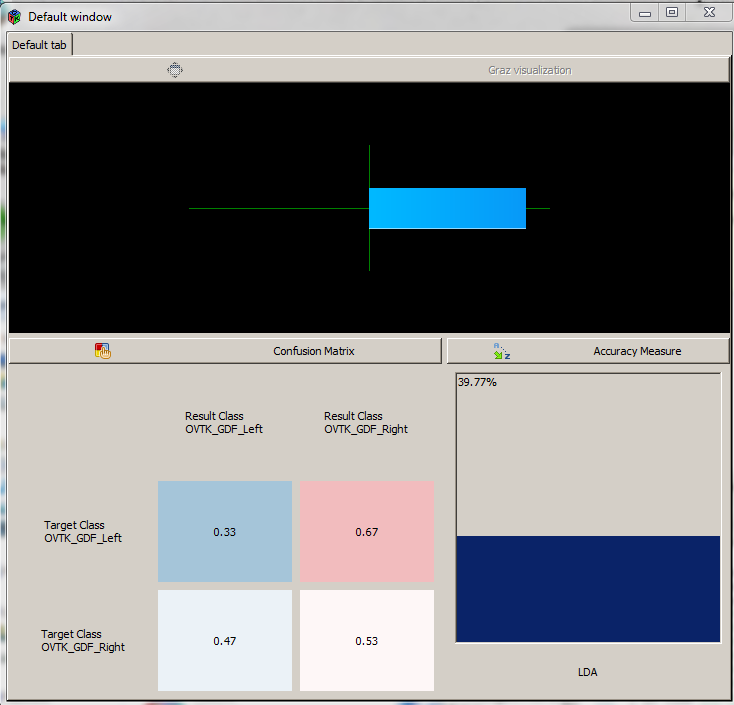


Figure Direction of Motor Imagery Strength

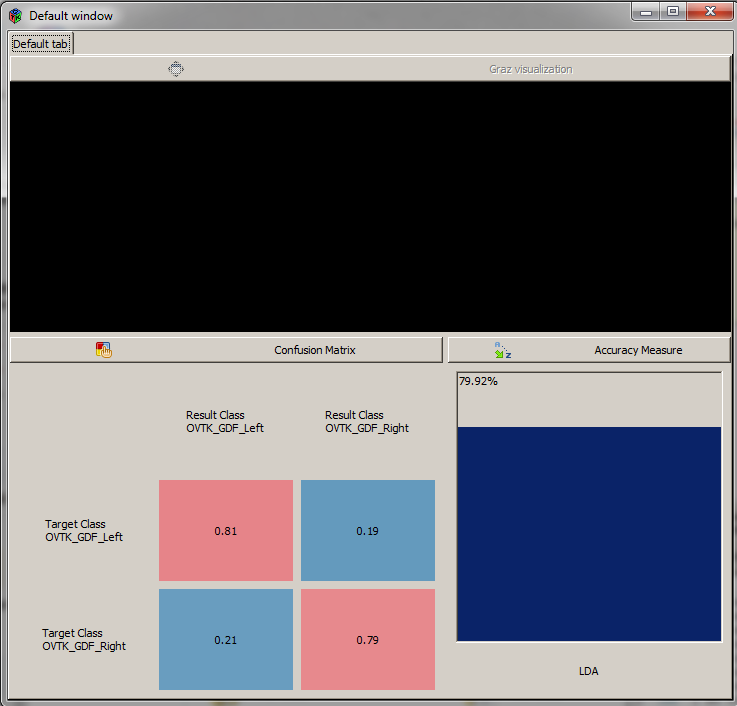


Figure Overall Performance Metrics

### 4.3.5 Real-time One

After Section 4.3.3 the BCI should be complete and functional, further evidenced by 4.3.4 which analysed the performance and accuracy of classifying the data. However in order to see how it performs in real-time another scenario was created similar to Acquisition.xml however in this scenario it would include the CSP Spatial Filter and the LDA Classifier. This would mean it could do the first three scenarios all at once. Considering the paper by Ang et al. (Ang et al., 2011) the user interface that provided cues would also now provide some feedback to the user as shown in Figure 34.

This could be considered distracting however it helps the user as they will know what is happening during the full process and how they are performing in the specific direction.

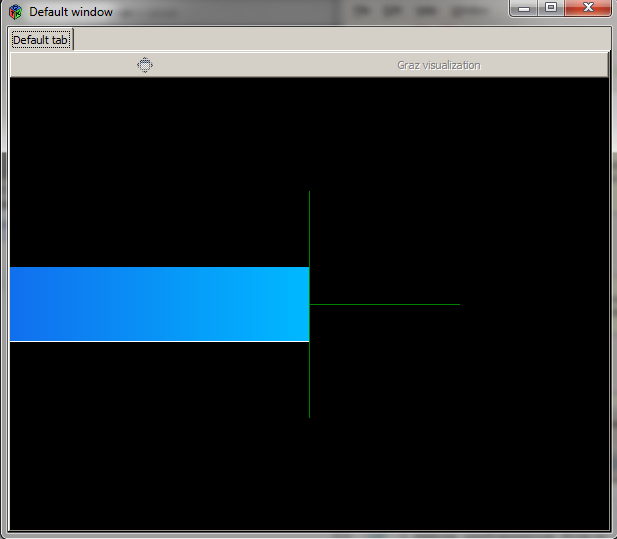


Figure Real-time One

### 4.3.6 Real-time Two

Similar to the Realtime1.xml this scenario (Figure 35) will acquire data in real-time and provide feedback, however a difference in this scenario is that is also provides analytical feedback similar to Analysis.xml with a confusion matrix and an accuracy measurement. This allows the user to see how the system performs in a real-time situations compared to the offline Analysis.xml (Figures 36, 37). As a result of the poor initial tests, shown in Figure 37 a second set of tests were carried out in order to achieve higher results within the 60% to 90% threshold shown in Figure 38 and 39. If the first tests were carried out offline the user may not have noticed until the Analysis.xml was used.

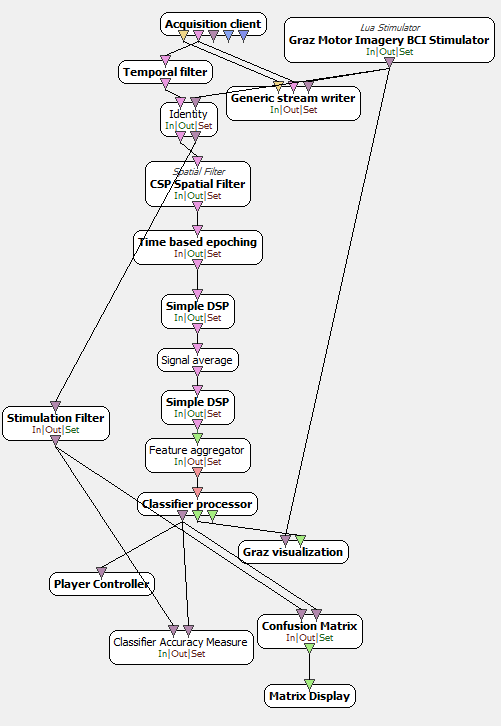


Figure 35 Real-time Two Scenario

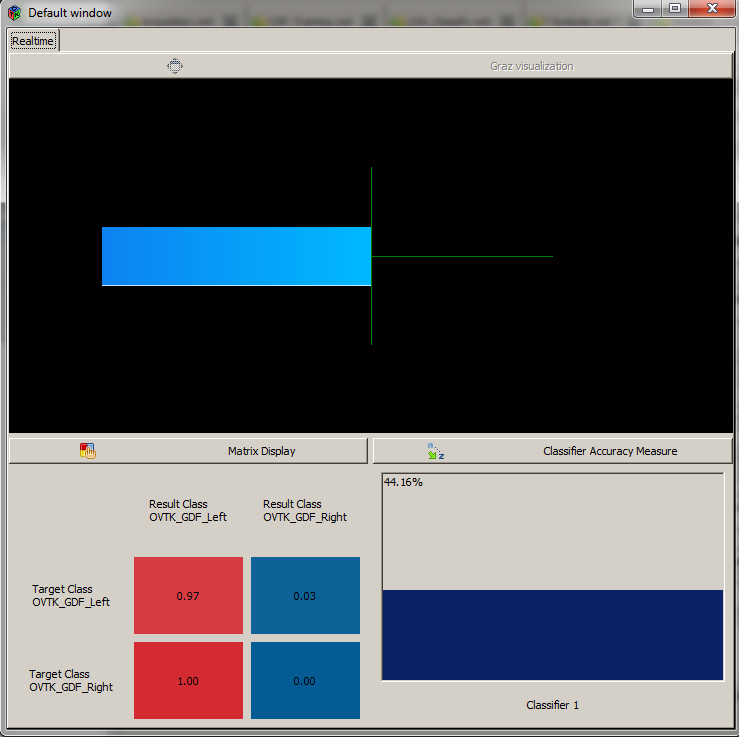


Figure Real-time Two GUI

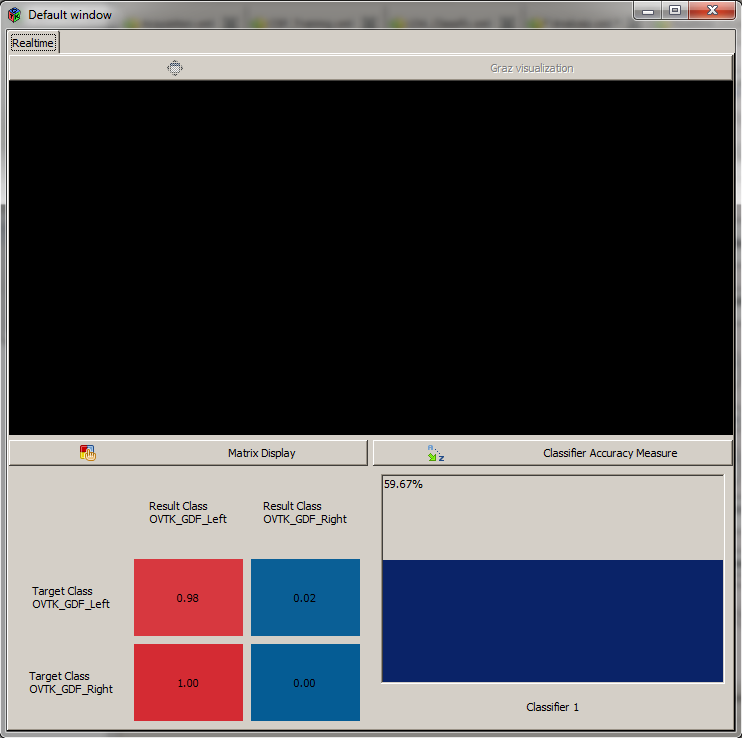


Figure 37 Real-time Two Overall Performance

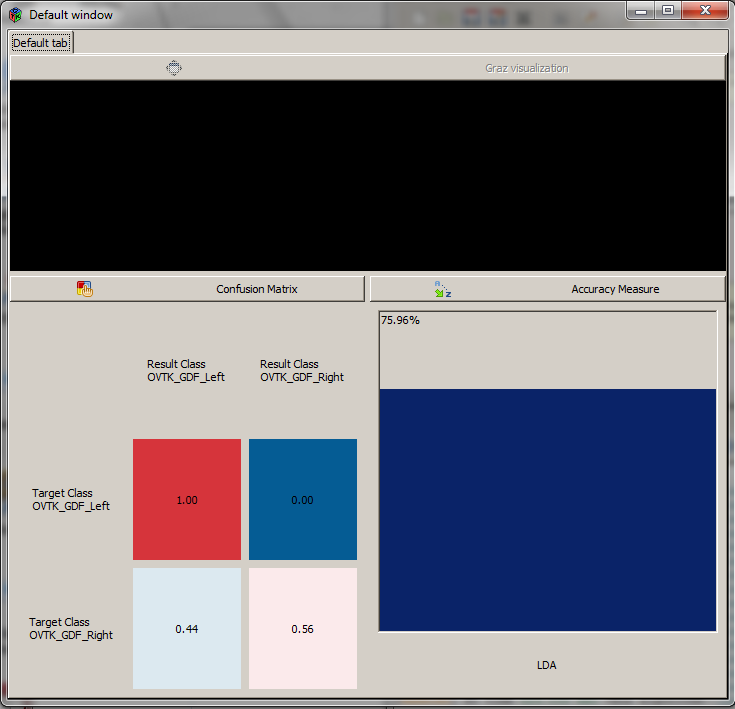


Figure Second Test Real-time Two

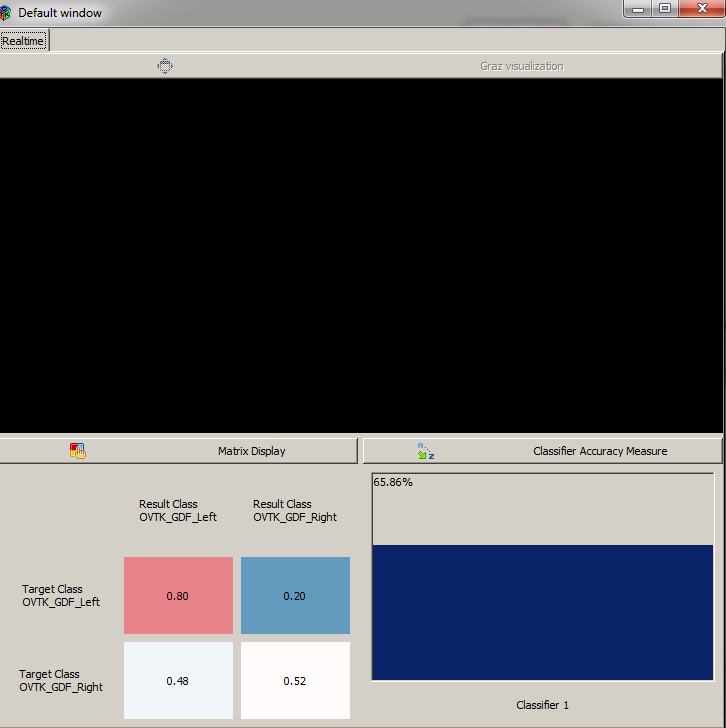


Figure Second Test Real-time Two

### 4.3.7 Python Scripting Box and Arduino Code

The Python scripting box allows users to attach and run a Python script anywhere in the scenario. In this project the box will be attached to the classifier processors in the Analysis.xml, Realtime1.xml and Realtime2.xml. An example is show in Figure 40 and when run the script MyOVBox.py will open COM1 (Figure 41) and send the associated left motor imagery character for example the classifier code is ‘769’ (Figure 42) which is will send the character ‘c’ Figure 43

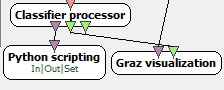


Figure Python Scripting Box



Figure Opening COM1 using Python Script



Figure Python Scripting Box Extracting Stimulation Codes and Printing

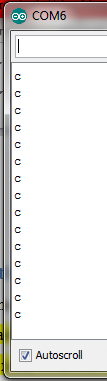


Figure Arduino Serial Recieving the Characters

As the motor imagery scenarios will only allow for two directions the Arduino code for System Two will accept two different character inputs to the serial monitor as shown in Figure 44 which based on the direction will open and close the hand. The code is in Appendix A called sketch\_Dissertation2.ino

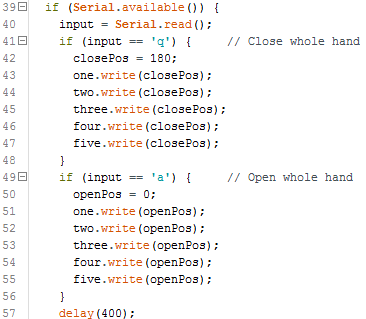


Figure Arduino code for System Two

# Chapter 5 Evaluation of Results

## 5.1 Testing Arm and Arduino

Using the Arduino code, sketch\_Dissertation.ino, and serial monitor testing showed all fingers being moved smoothly when the input is given, demonstrating the arm would perform well as a prosthetic and that the Arduino UNO as a microcontroller.

## 5.2 System One – Cognitive Suite

Results of this system demonstrated successful control of an external device, the 3D arm and the Arduino UNO, using EEG signals. In this it was noticed that when the number of actions trained is increased the accuracy of the system to distinguish between them decreased. This issue is likely caused by the processing techniques in the background and as a result means that training can be hard as interference such as facial movements such as but not limited to; blinking, eyebrow movements, clenching of the teeth or even swallowing as well as a limit to know how many movements a user can make.

The lack of control over the processing in the background for instance the choice in EEG method, the filtration and the classifier makes this system less desirable regardless of it successful functionality. A prosthetic would need to robust and reliably for real world use and a system that cannot allow the user to have a consistent control would not be of use. Therefore the second system is designed using a BCI software platform called OpenViBE.

## 5.3 System Two - OpenViBE

### 5.3.1 Acquisition

To investigate whether having a focus on one side of motor imagery as well as the amount of cues would influence the processing and overall accuracy of the system two experiments were performed. Starting with the Acquisition.xml two recording were made the first Acquisition1.ov focusing on left motor imagery used 40 cues and took 16 minutes to complete whereas Acquisition2.ov focusing on right motor imagery used 20 cues and took only 8 minutes to complete.

This scenario has no feedback for the user other than the instructions provided in the form of a green cross and red arrow and the knowledge that once the required number of cues has been done the GUI will automatically close an a message will be in the OpenViBE command line and platform message section, however a real world user would likely not use these when using the system.

Afterwards to check if the signal has been recorded correctly a simple scenario was created to stream the contents of the acquisition file so that they can be used to quickly check and analysis the signal acquired. It is a simple scenario that takes the just recorded data file and replays the signals back to the user. As seen in Figure 45 and 46, as the signal scrolls along the changes in the signals behaviour when a user imagines the movements can be seen. The green and red dashed line indicated the green cross and red cue that the user followed in the acquisition task. The first line in Figure 45 and the first three line in Figure 46 are the green arrow markers and the last line in Figure 46 is the red cue marker. This helps to see if they are responding fast enough or if the scenario needs the time between cues to be altered. It can also be useful in further scenario, CSP and LDA trainers, as the stimulation epoching could be altered to accommodate the user’s performance. It can also be seen in Figure 40 the moment when the user starts making an imagined movement the signal becomes erratic.

The number of sample produced by these two showed that every 20 cues would produce 10000 more. Even with an increase it sample the accuracy of the system was only 10% larger having a user sit for long periods of time could become taxing and reduce their concentration on the task.

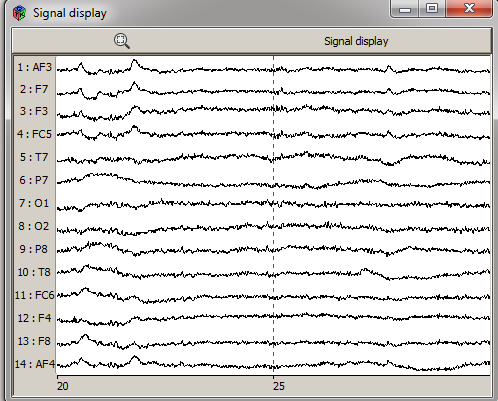


Figure 45 Check Signal 1

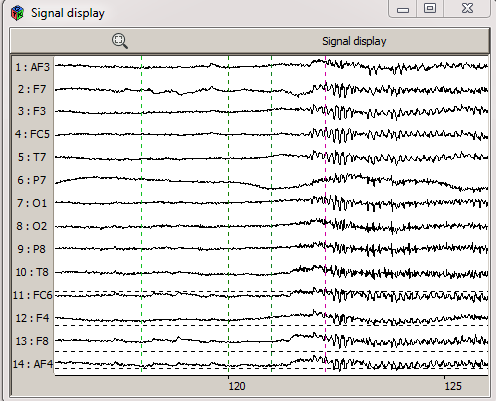


Figure 46 Check Signal 2

### 5.3.2 Training CSP

As the choice of headset was the Emotiv EPOC+ a normal filtering scenario would not be possible instead the CSP filter was needed to successfully remove the noise and/or artifacts from the data. The result from this is that a CSP Spatial Filter was trained to clean the acquisition data and be used by further scenarios.

### 5.3.3 Classify

As shown in the implementation the percentages are 84% and 74%. Normally between 60% and 90% is considered a successful accuracy with the higher being better. Being able to achieve 74% with a commercial headset is excellent even achieving with half the amount of samples is good.

It is possible that a higher accuracy could be achieved by rerunning the training or by using a new acquisition data file for new samples. There is also the consideration that the CSP spatial filter could be improved reducing any remaining noise or artifacts and provide the LDA Classifier with cleaner samples to learn from.

### 5.3.4 Analysis

Using the three performance metrics we can analyse the results of the data. Using the feedback can show how strong the focus in one direction there is but it can also show if the direction fluctuates between the central point at the start of each cue. This can indicate that the user struggles to switch between direction or a neutral state and could help to calibrate the stimulation times between cues to allow for a longer period however this could have the disadvantage of causing the acquisition time to be extended. The other two performance metrics analyse how well the classifier is performing in real-time and can show the accuracy of the user for certain direction. This can help the user to examine their strengths and weaknesses to form a solution for the weaker results and practice concentrating on those more.

From the analysis of Acquisition1.ov it is shown that I as a subject have a much stronger EEG signal for left motor imagery with it almost overpowering the right handed cues. Compared to Acquisition2.ov which had intentions focus on right motor imagery the overall balance was better but again the focus would sometimes overpower the left cues. Finding a balance between the two is difficult to learn as a user and will likely require a longer period of training to achieve.

It is shown in Figure 47 that on Acquisition1.ov the performance achieved 100% accuracy whilst the analysis shown left motor imagery and the performance accuracy dropped as right motor imagery was introduced (Figure 48). This could suggest that if right motor imagery can be train to a similar degree of left motor imagery then a BCI system that can achieve 100% accuracy would be created. The limiting factor in this right now is the user and their ability to focus on motor imagery tasks which is why Acquisition2.ov was created.

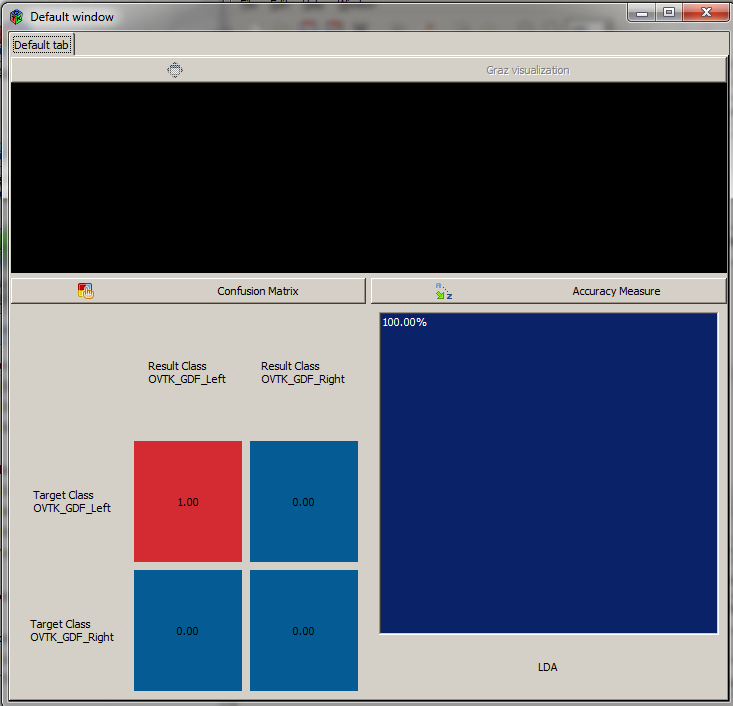


Figure 100% Accuracy for just left motor imagery

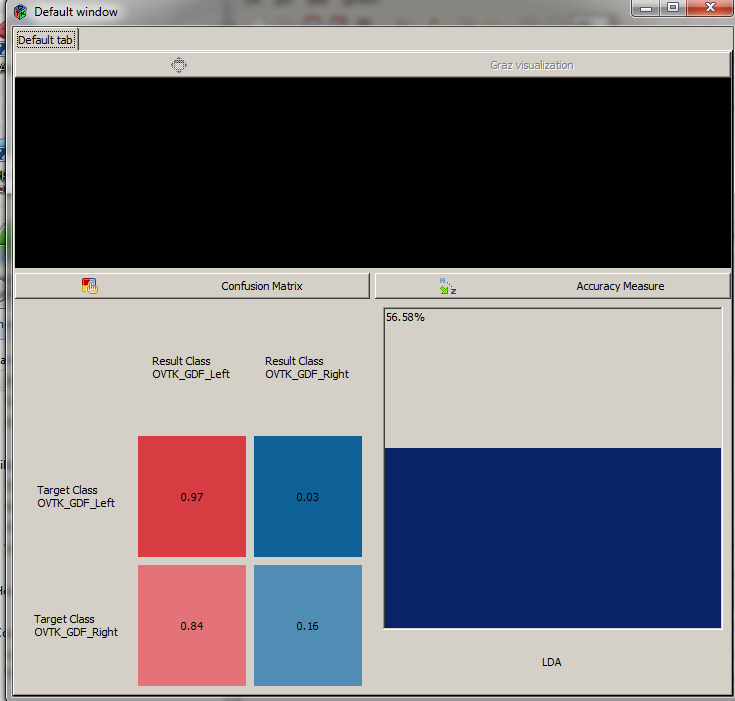


Figure Introduction of right motor imagery

### 5.3.5 Real-time One

The output of this scenario is Realtime1.ov that consists of 20 cues similar to the Acquisition2.ov and hence used the CSP spatial filter and LDA Classifier trained from it. The added feedback provided allowed for better focus towards the required direction.When this data acquired from this scenario was used in the Analysis scenario, it clearly showed that the left motor imagery was much stronger than the right similar to Figure 34.

### 5.3.5 Real time Two

Similar to Realtime1.xml the output of this scenario is Realtime2.ov that consists of 20 cues similar to the Acquisition2.ov and hence used the CSP spatial filter and LDA Classifier trained from it. This result showed much better results during acquisition as the user had more information provided allowing for better adaptation. In this scenario whilst running, it was shown that when completing the motor imagery task I performed better at left sided tasks compared to right sided tasks by a large margin and that it took more concentration to achieve the feedback in the right sided cues (Figure 47).

### 5.3.6 Python Box

By adding in the python box the last aim could be completed as the script provided that the last component of extracting the classifier and successfully outputting a character to Arduino to use.

# Chapter 6 Conclusions and Future Research

## 6.1 Conclusions

### 6.1.1 3D Arm and Arduino UNO

Overall the performance of the 3D printed arm excellent. The servos were able to control the fingers and were responsive to the positions the Arduino code gave.

The choice of the Arduino UNO as a microcontroller is also good as it is powerful enough to perform the intended task and depending on how the data and the amount transferred, it could be a consideration for real world applications.

In system one it is required that the Arduino is attached to a computer in order to receive characters meaning that if it were to become fully portable a solution to this must be found. One solution could be to have a wireless connection similar to how the Emotiv EPOC+ can wirelessly send data to the computer however whether it is system one or system two both currently require a computer to run the software.

### 6.1.2 Headset

The choice of headset was an affordable commercial headset that provided excellent data for the limited number of sensors covering the key sections of the scalp, compared to the very expensive medical headset alternatives that offer numerous electrodes covering the entire scalp.

An issue with the Emotiv EPOC+ is that the design of the headset is not completely ergonomic and after longer periods of wearing the headset can become painfully uncomfortable to wear more than 30 minutes particularly the reference sensors behind the ears begin to apply a lot of pressure to the scalp.

Another issue with using an EEG headset is that every time it is used it must be setup and calibrated each time to ensure a good strength signal. Setup involves maintaining the headset for example if it uses wet electrodes like the EPOC+ then ensuring the sensors are covered in enough saline to easily acquire the EEG signals. Calibration is ensuring that the headset is positioned correctly and with sensors on the scalp to provide the best acquisition of signals which is best seen in the Control Panel (Figure 16).

Finally the Emotiv EPOC+ as a headset can have issues with noise produced which requires the raw signal to be correctly filtered which in this study was done using the Common Spatial Pattern spatial filter.

### 6.1.3 System One – Cognitive Suite

System One produced a successful implementation that met the majority of the aims of this paper but lacked control over the system that was desired and was fairly simplistic. Furthermore could be unreliable depending on the user and the number of taught actions hence System Two was created to attempt a better BCI system.

### 6.1.4 System Two - OpenViBE

OpenViBE was chosen for the second system as it offers greater control over the acquisition and processing tasks for instance filtration and classifying however it can lack the ability to be used for real world applications that aren’t virtual reality or gaming based without the user creating their own scripts or box modules. Another benefit is that it can perform both offline and online analysis which can aid the user or the designer of the scenarios.

Some of the methods attempted to export the classified data is the old and unsupported MATLAB scripting box which required the user to create scripts to process the data going in and out and would only run if the user has MATLAB version 2011a to 2012a. This was successful at extracting all the information however picking out the exact classified information proved difficult and due to time constraint could not be fully achieved.

Instead the ‘Python Scripting’ box was used and a customised script created that was able to extract the stimulation code from the classifier processor and output a message via COM ports to the Arduino serial monitor. This allowed for the system to meet all the aims and objectives.

### 6.1.6 Motor Imagery

Motor imagery was chosen as System Two’s paradigm as the intended users may lack the ability to control certain motor actions or remember all the different action and what they thought behind those actions were whereas with motor imagery users can usually produce good results for less training as imagining movements is fairly easy and simple. This does however come with the disadvantage of fewer controls to use compared with System One which can have many but the accuracy of the system may decrease. Therefore these Systems Two’s motor imagery actions can be more robust and more suitable for the use in a prosthetic.

### 6.1.7 Current BCI

The BCI that are usually implemented in recent times will usually have up to three degrees of freedom whereas with system one more action can be taught depending on the skill of the user and their brain plasticity.

Another reason for limited degrees of freedom is that when using EEG having 60% accuracy for two or three degrees of freedom or as high as 90% is not feasible for most situations as a robust real world application will likely require full 100% control. The OpenViBE scenario could also be altered to have more degrees of freedom by using the time between the cue as a neutral command or if a medical grade headset could be used adding in more motor imagery tasks for example left and right hand, legs and a mixture of both could overall provide a greater system that this study could not produce.

## 6.2 Limitations

In this thesis there are clear limitations with this project as a result of the chosen hardware and software. The only limiting hardware would be the headset with its limited number of electrodes available. The software limitation is OpenViBE which cannot easily output the processed signals especially to other hardware or the Python box which send the character via the COM port.

If this study was to aim towards implementing the designs in to real world or clinical environment then we must consider Sajda et al. paper discussion which explain that one of the challenges that neural prosthesis has is “the transition from experimental to clinical settings” (Sajda et al., 2008) due to the requirement of these systems to be “more robust and autonomous” (Sajda et al., 2008).

## 6.3 BCI Illiteracy

BCI illiteracy is a key issue in designing BCI systems as not everyone use it. In this study we achieved accuracy of 70% to 90% however it still showed that at the start the left motor imagery was far stronger than the right and through repeating training right imagery became stronger but still required a lot of concentration and did not achieve the 100% accuracy like the left did in Figure 47. Fortunately in this study training was quick taking minutes to hours at most to obtain high accuracy and not weeks or months like it could take others to achieve high levels of accuracy which could be deterring for users if the outcome is still a low level of accuracy for all the work put in.

## 6.4 Robustness and Applicability

System One using the Cognitive Suite is a very robust system, although it can have interference when reading data, it can still acquire a large amount of commands and be used to control the prosthetic. These commands can be added, retrained or removed without it having any effect on the system other than the removal of which ever action it represented. This also makes it a functional application that could be used in the real world or clinical situations to help those that desired to have control in the world such as patients with locked in syndrome.

System Two created using OpenViBE is also very robust. Modules and their configuration setting can be easily changed or the modules can be removed with new ones being added. This allows the designer to create prototype scenario without speeding a long time worrying about the flow of data or the types of data needed to be input and output. If certain part of the scenario does not work then the scenario as a whole will continue for instance if the Python script encounter an error whilst initialising OpenViBE would simply uninitialized the script and carried on with the scenario.

A commercial headset such as the Emotiv EPOC+ used in this study is one of the best headsets to use for real world application regardless of whether it is uncomfortable or may have an increase in signal noise from external interference it is still a robust product. Furthermore the fact that it is wireless and therefore portable whilst still producing these high grade signals reinforces the robustness and applicability of this EEG headset.

## 6.5 Final Conclusion

This study had researched, designed and implemented an EEG based BCI using motor imagery in System One and then produced a superior version in System Two that could filter and classify the EEG signal before output a message to Arduino to control the prosthetic.

As a result this study has successfully achieved all of the aims to create a fully functional system and proven the feasibility of the current design systems for highly accurate classification with two robust but affordable systems that could be applied patients requiring applications for external control.

## 6.6 Future Research

This project has successful designed and created a BCI system that will take the raw EEG signal and process it into a viable source of controlling devices. Although the current platform OpenViBE does not have an easy way to export these result to other platforms it was still possible with the created Python script however another better solution could be made that provides a better transference of classifier labels or a box support for serial communication could be created rather than using languages like Python as a bridge between software platforms.

The 3D printed arm design could be improved slightly so that the Arduino board can be placed inside of the arm for portability rather than the current setup with wires to a breadboard and then to the Arduino. This change would require moving all of the servos into different positions throughout the arm instead of all together and would have to be considered so as to not affect the functionality of the arm. At the time of writing this paper a slightly newer version of that incorporated the majority of these design changes by creating a larger sized arm than the one used in this study can be seen in the new Thingiverse page (Thingiverse, 2012).

One concern that hasn’t been approached in this study is that the system would need and awake and sleep mode so that when the user what to move the arm it will move and when they do not want to make any movement it will remain in a sleep state. Implementing a feature that could recognise when the user is paying attention perhaps using frequency bands of brain wave similar to the Emotiv Control Panel or a BCI task that would be learnt so that the user can turn on or off the system or prosthetic.

# References

Anatole, L., Lotte, F., Reilly, R.B., Leeb, R., Hirose, M. and Slater, M., 2008. Brain-computer interfaces, virtual reality, and videogames. *Computer*, (10), pp.66-72.

Ang, K.K., Guan, C., Wang, C., Phua, K.S., Tan, A.H.G. and Chin, Z.Y., 2011, August. Calibrating EEG-based motor imagery brain-computer interface from passive movement. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE* (pp. 4199-4202). IEEE.

Bougrain, L., Rochel, O., Boussaton, O. and Havet, L., 2012. From the decoding of cortical activities to the control of a JACO robotic arm: a whole processing chain. arXiv preprint arXiv:1212.0083.

Farwell, L.A. and Donchin, E. (1988). *Taking off the top of your head: toward a mental prostheses utilizing event-related brain potentials*. doi:10.1016/0013-4694(88)90149-6

Guger, C., Ramoser, H. and Pfurtscheller, G., 2000. Real-time EEG analysis with subject-specific spatial patterns for a brain-computer interface (BCI).Rehabilitation Engineering, IEEE Transactions on, 8(4), pp.447-456.

Hema, C., Paulraj, M., Yaacob, S., Adom, A., and Nagarajan, R. (2009). EEG Motor Imagery Classification of Hand Movements for a Brain Machine Interface

Höhne, J., Holz, E., Staiger-Sälzer, P., Müller, K.R., Kübler, A. and Tangermann, M., 2014. Motor imagery for severely motor-impaired patients: evidence for brain-computer interfacing as superior control solution.

Johnson, G.D., [no date] ASYNCHRONOUS REAL-TIME CONTROL OF A ROBOTIC ARM USING A BRAIN-COMPUTER INTERFACE.

Levine, SP; Huggins, JE; Bement, SL; Kushwaha, RK; Schuh, LA; Rohde, MM; Passaro, EA; Ross, DA; Elisevich, KV et al. (2000). *A direct brain interface based on event-related potentials*. IEEE transactions on rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society. 8 (2). pp. 180–185. doi:10.1109/86.847809. PMID 10896180

Lotte, F., Congedo, M., Lécuyer, A. and Lamarche, F., 2007. A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of neural engineering*, *4*.

LUA, (2015). Lua: about. [online]. Available at: http://www.lua.org/about.html [Accessed 13 Nov. 2015].

Mak, J.N. and Wolpaw, J.R., 2009. Clinical applications of brain-computer interfaces: current state and future prospects. *Biomedical Engineering, IEEE Reviews in*, *2*, pp.187-199.

Miller, K.J., Schalk, G., Fetz, E.E., den Nijs, M., Ojemann, J.G. and Rao, R.P., 2010. Cortical activity during motor execution, motor imagery, and imagery-based online feedback. *Proceedings of the National Academy of Sciences*, *107*(9), pp.4430-4435.

Müller-Putz, G.R., Scherer, R., Pfurtscheller, G. and Rupp, R., 2005. EEG-based neuroprosthesis control: a step towards clinical practice.*Neuroscience letters*, *382*(1), pp.169-174.

Nijholt, A., 2009. BCI for games: A ‘state of the art’survey. In *Entertainment Computing-ICEC 2008* (pp. 225-228). Springer Berlin Heidelberg.

Pfurtscheller G, Muller GR, Pfurtscheller J, Gerner HJ, Rupp R. ‘Thought’--control of functional electrical stimulation to restore hand grasp in a patient with tetraplegia. Neurosci Lett. 2003 Nov 6;351(1):33–6.

Python, (2015). [online]. What is Python? Executive Summary. https://www.python.org/doc/essays/blurb/ [Accessed Jan. 13]

Qin, L., Ding, L. and He, B., 2004. Motor imagery classification by means of source analysis for brain–computer interface applications. *Journal of Neural Engineering*, *1*(3), p.135.

Ramoser, H., Muller-Gerking, J. and Pfurtscheller, G., 2000. Optimal spatial filtering of single trial EEG during imagined hand movement. *Rehabilitation Engineering, IEEE Transactions on*, *8*(4), pp.441-446.

Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Bertrand, O. and Lécuyer, A., 2010. OpenViBE: an open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. Presence: teleoperators and virtual environments, 19(1), pp.35-53.

Rossini, L., Izzo, D. and Summerer, L., 2009, September. Brain-machine interfaces for space applications. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*(pp. 520-523). IEEE.

Sajda, P., Müller, K.R. and Shenoy, K.V., 2008. Brain-computer interfaces.*IEEE Signal Processing Magazine*, *25*(1), pp.16-17.

Smith, E. and Delargy, M., 2005. Locked-in syndrome. *BMJ: British Medical Journal*, *330*(7488), p.406.

Thingiverse, (2012). [online]. Hand robot InMoov by Gael\_Langevin – Thingiverse. http://www.thingiverse.com/thing:17773 [Accessed Aug. 12]

Ungerleider, L.G., Doyon, J. and Karni, A., 2002. Imaging brain plasticity during motor skill learning. Neurobiology of learning and memory, 78(3), pp.553-564.

Vidal, JJ. (1973). *Toward Direct Brain-Computer Communication*. Annual Review of Biophysics and Bioengineering, 2 (1). pp. 157-180.

Vidal, JJ. (1977). *Real-time detection of brain computer events in EEG*. In: Proceeding of the IEEE 65 (5). pp. 633-641.

Vidaurre, C. and Blankertz, B., 2010. Towards a cure for BCI illiteracy. Brain topography, 23(2), pp.194-198.

Wolpaw, J., McFarland, D., Neat, G. (1990). *Development of an Electroencephalogram-based Brain-Computer Interface.* Annals of Neurology, 2 (2). pp. 250-251.

Wolpaw, J., McFarland, D., Neat, G. and Forneris, C. (1991) *An EEG-based brain-computer interface for cursor control*. doi:10.1016/0013-4694(91)90040-B

# Appendix A – CD

All code and software used in the project can be found on the disc provided with this dissertation.