

NON-INVASIVE BRAIN EEG ROBOT NAVIGATION  
FRAMEWORK BASED ON EMOTIV EPOC

By

Hossam El-Sayed Mohamed Soffar

12286

Dissertation submitted in partial fulfilment of the requirements for the  
Electrical & Electronics Engineering Department  
Universiti Teknologi Petronas

MAY 2013

Supervisor name:

Dr. Aamir Saeed Malik

CERTIFICATION OF APPROVAL

Non-invasive Brain EEG Robot Navigation  
Framework Based on Emotiv Epoc

By

Hossam El-Sayed Mohamed Soffar

A project dissertation submitted to the  
Electrical and Electronics Engineering Programme  
Universiti Teknologi PETRONAS  
In partial fulfilment of the requirement for the  
BACHELOR OF ENGINEERING (Hons)  
(ELECTRICAL AND ELECTRONICS ENGINEERING)

Approved by,

---

(Dr. Aamir Saeed Malik)

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK  
AUGUST 2013

## **CERTIFICATION OF ORIGINALITY**

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgments, and that the original work contained herein have not been undertaken nor done unspecified sources or persons.

Produced by,

**HOSSAM EL-SAYED SOFFAR**

---

## **ACKNOWLEDGEMENT**

I would like to articulate my deepest gratitude to all those who provided me the possibility to complete my Final Year Project and the report. I give a special gratitude to Dr. Aamir Saeed Malik, my direct supervisor, for his suggestions and supervision.

A special thanks goes to my friends, who helped me in the areas I needed guidance in. Lastly, I would like to show gratitude to GOD for giving me the courage and strength to go through this task.

## ABSTRACT

A Brain computer interface (BCI) has introduced new dimensions and created a new era for creative applications for developers and researchers giving alternative communication channels for people suffering of motor disabilities. The motor system is currently the primary focus, where EEG signals are being obtained while the subject is imagining or performing a motor response. With the help of an EEG signals, navigating a robot or controlling a wheel chair has come from science fiction movies or stories to reality. The purpose of this study is to design a navigation framework for a robot or a wheel chair using the capabilities of the brain EEG using an online acquisition device in our case it's Emotiv Epoc. The framework is based on SSVEP focusing on visual cortex EEG signal analysis, the methodology used to complete this application is agile software development which basically is based on iterative and incremental development providing a rapid and flexible product. The tools used to implement the framework are divided into two categories hardware and software as for the hardware EEG acquisition device is used to acquire EEG signals, a robot based on NXT logo is used to demonstrate the control capabilities of the framework, as for the software Emotiv SDK , Open vibe as a signal processing and visual studio to design the a GUI as a man in the middle for interfacing the EEG signal processing platform with the robot.

# Table of Contents

<b>ACKNOWLEDGEMENT</b> .....	<b>IV</b>
<b>ABSTRACT</b> .....	<b>V</b>
<b>Table of Contents</b> .....	<b>VI</b>
<b>LIST OF FIGURES</b> .....	<b>VIII</b>
<b>ABBREVIATIONS AND NOMENCLATURES</b> .....	<b>X</b>
<b>CHAPTER 1 INTRODUCTION</b> .....	<b>1</b>
1.1 Background Study:.....	1
1.2 Problem Statement:.....	2
1.3 Objectives & Scope of Study:.....	2
1.3.1 Objectives:.....	2
1.3.2 Scope of Study:.....	2
1.4 Project Feasibility.....	3
<b>CHAPTER 2 LITERATURE REVIEW</b> .....	<b>4</b>
2.1 Brain structure:.....	4
2.2 The Anatomy of Movement.....	5
2.3 Electroencephalography (EEG):.....	5
2.4 Brain computer interface:.....	6
2.4.1 SSEVEP:.....	11
2.4.1.1 Open vibe:.....	12
<b>CHAPTER 3 METHADODOLOGY</b> .....	<b>14</b>
3.1 Project Workflow.....	14
3.2 Research methodology.....	17
3.3 Tools.....	17
3.3.1 Software:.....	17
3.3.2 Hardware:.....	18
3.4 Gantt chart.....	19
3.4.1 FYP 1.....	19
3.4.2 FYP 2.....	19
<b>CHAPTER 4</b> .....	<b>20</b>
<b>RESULTS AND DISCUSSION</b> .....	<b>20</b>
4.1 RESULTS AND DISCUSSION:.....	20
4.1.1 First Approach Using Emotiv API.....	20

4.1.2 Second Approach Using Designing SSVEP BCI (Using Open Vibe) .....	25
<b>CHAPTER 5</b> .....	<b>38</b>
<b>CONCLUSION AND RECOMMENDATIONS</b> .....	<b>38</b>
<b>5.1 CONCLUSION</b> .....	<b>38</b>
<b>5.2 Recommendations and Future work:</b> .....	<b>39</b>
<b>REFERENCES</b> .....	<b>40</b>
<b>APPENDICES</b> .....	<b>43</b>
APPENDIX A.....	43
APPENDIX B:.....	49

## LIST OF FIGURES

FIGURE 1: THE CEREBRUM IS BEING SUBDIVIDED INTO FOUR LOBES <sup>[1]</sup> .....	5
FIGURE 2: PRINCIPAL CORTICAL DOMAINS OF THE MOTOR SYSTEM.....	6
FIGURE 3:THE EMOTIV EPOC HEADSET IS USING.....	8
FIGURE 4 EMOTIV EPOC HEADSET .....	9
FIGURE 5: A TYPICAL BCI SYSTEM ARCHITECTURE.....	9
FIGURE 6: SCHEMATIC DIAGRAM OF SIGNAL PROCESSING APPLIED TO AN EEG RECORDING FROM SIGNAL ACQUISITION THROUGH TO CONTROL COMMANDS GENERATED BY THE FINAL APPLICATION. <sup>[22]</sup> .....	10
FIGURE 7: EXAMPLE OF VARIOUS FREQUENCIES .....	11
FIGURE 8 : EEG TOPOGRAPHY SSVEP EFFECT ON VISUAL CORTEX .....	11
FIGURE 9:OPEN VIBE GRAPHICAL INTERFACE .....	12
FIGURE 10 SPATIAL FILTER TRAINER.....	13
FIGURE 11: FLOW CHART FOR IMPLEMENTING FIRST STAGE FOR THE PROJECT.....	15
FIGURE 12 : DECISION MAKING ALGORITHM .....	16
FIGURE 13: BCI ARCHITECTURE .....	20
FIGURE 14: SYSTEM DESIGN ARCHITECTURE.....	21
FIGURE 15: FLOW DIAGRAM FOR USING EMOTIV WITH PUZZLE BOX TO CONTROL THE NXT ROBOT	22
FIGURE 16: CONTROL PANEL .....	23
FIGURE 17:EMOKEY.....	23
FIGURE 18: PUZZLE BOX.....	23
FIGURE 19: ACQUISITION SERVER.....	25
FIGURE 20: SAMPLE OF EEG RECORDING.....	25
FIGURE 21 OPEN VIBE ACQUISITION SERVER .....	26
FIGURE 22: SAMPLE OF OPEN VIBE BOXES.....	27
FIGURE 23: FLOW DIAGRAM OF THE SYSTEM.....	28
FIGURE 24: SSVEP STIMULUS SCREEN .....	29
FIGURE 25: CONFIGURATION BOX FOR TRAINING DATA.....	29
FIGURE 26EEG SIGNAL ON LEFT, BAND BASS FILTERED SIGNAL ON RIGHT .....	31
FIGURE 27: CLASSIFIER BOXES .....	32
FIGURE 28: SPATIAL FILTER (OUTPUT SIGNALS) FOLLOWED BY A BAND BASS FILTER .....	33
USING THE GENERATED CLASSIFIERS WHEN THE USER FOCUS ON ONE OF THE STIMULUS AN EVENT IN THE VRPN SERVER WILL BE EVOKED AND ANOTHER STANDALONE VRPN CLIENT WILL READ THOSE EVENTS AND ACT UP ON IT , OUR CLIENT IN THIS CONDITION IS THE APPLICATION RESPONSIBLE FOR THE ROBOT NAVIGATION AND THE DECISION MAKING.( FIGURE 29)(REFER TO APPENDIX B).....	34
FIGURE 30: ONLINE TESTING WITH A ROBOT .....	34
FIGURE 31: STIMULUS FOR ONLINE TEST.....	34



FIGURE 32 : SCRIPT FOR ONLINE TESTING .....	34
FIGURE 33: GUI FOR THE PROGRAM INTERFACE.....	35
FIGURE 34: SAMPLE CODE FOR ESTABLISHING CONNECTION .....	36
FIGURE 35: INTERFACE PROGRAM IN ACTION .....	36
FIGURE 36: SYSTEM FLOW .....	37
FIGURE 37: DECISION MAKING ALGORITHM.....	37

## ABBREVIATIONS AND NOMENCLATURES

- BCI Brain Computer Interface
- EEG Electroencephalogram
- API Application Programming Interface
- SCP slow cortical potentials
- TBI Traumatic brain injury
- SCI spinal cord injury
- SSVEP steady state visual evoked potentials
- VRPN Virtual-Reality Peripheral Network
- LDA Linear Discriminant Analysis
- SVM Support Vector Machine

# CHAPTER 1

## INTRODUCTION

### 1.1 Background Study:

BCI is an underdevelopment field which allows interaction without any muscular activity. Basically every communication or control event naturally requires peripheral nerves and muscles<sup>[1]</sup>. The intended process or event always starts with the user's decision triggering a complex chain of processes where typically some of the brain areas are activated, and hence signals are sent through the nervous system through the motor pathways in the event of a motor movement<sup>[1]</sup>. The field is as vast as it needs an understanding of the users and their requirements this kind of technology is mostly intended for people who have limitations on their physique that prevents them from physically interaction with the surrounding environment.

Few years back the BCIs were just some laboratory gadgets and a sort of science fiction movies ideas although the BCIs has gone through development for many years but yet still it's being faced with a lot of difficulties it's still expensive , time consuming and excruciatingly infrequent which at the same time limit its speed and usability.

Numerous BCI concepts and application have been developed using a cutting edge technologies provided by numerous companies as Emotiv and Neurosky which are releasing their EEG devices (Headsets) for the consumer market as well as the research market<sup>[2, 3, 4]</sup>.

BCI controllers can be implemented in two different ways, invasive which needs a medical procedure penetrating the skin or another way of implementation called non-invasive methods mostly using a headset or a plastic cap with electrodes attached<sup>[2, 4]</sup>. A BCI device or an application can be designed in various ways could be used in turning lights or home appliances on/off, controlling a wheel chair control a computer based programs a web browser or could even be used in playing music.

## **1.2 Problem Statement:**

Spinal cord injury, Traumatic brain injury, and Stroke are the most well-known leading causes of disability which affects over millions of individuals annually. Loss of hand or a movement function causes a severe decrease in quality of life for these affected individuals.

Imagine walking up one day not able to move your arms or legs trying to move your finger but you are still unable to move them. Trying to ask for help or scream but you can't get your voice out even your facial muscles are not working since then you'll become a prisoner in your own body.

A person suffering from paralysis or immobile cannot interact with the surrounded environments just like the others can do, understanding the needs and problems of the brain computer interface will allow researchers to work more on the reliability of the system and develop new systems and devices which may make it easier for the users of BCI giving them another chance to live their life as normal people do.

## **1.3 Objectives & Scope of Study:**

### **1.3.1 Objectives:**

- To design a general BCI Framework based on Emotiv for controlling a robot.
- Analyze the used method and identify its drawbacks then come up with a new/modified approach for the BCI.
- Explore the use of EEG technology particularly BCI devices.

### **1.3.2 Scope of Study:**

A BCI using Electroencephalography (EEG) input will be used as an input for EEG data to the designed framework making use of the Emotiv API and its predefined and programmed detection suites with the help of C# on the base of .NET framework as a programming and interfacing language, another approach will be used to design an SSEVEP based BCI and compare it with the initial approach.

#### **1.4 Project Feasibility**

- The proposed project is feasible since the facilities lab equipment and software licenses which need to complete the study is provided by “University Technology PETRONAS, UTP”
- Regarding the proposed timeframe we believe that the project can be completed within that time frame for FYP1, and 2.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Brain structure:

Being the most complex organ in the human body and the biggest, it's made up of more than hundred billion nerve cells that do communicate between each other in a huge number of connections paths named as synapses <sup>[5]</sup>.

Our human brain is consisted of two cerebral hemispheres; the left one and the right one are connected by a large amount of nerves cells. Our brain is divided into four main lobes named as (frontal lobe, occipital lobe, parietal lobe, and temporal lobe) <sup>[6, 5]</sup>.

- Frontal lobe: located at the front area of our skull in front of the partial lobe. It's mostly involved with purposeful acts like judgment, creativity, retaining longer term memories ,problem solving, planning and recognizing future consequences. <sup>[6]</sup>
- Parietal lobe: on the top back area of the skull it's positioned above the occipital lobe. Its involved with processing higher sensory information and language functions also knowledge of numbers and their relations. <sup>[6]</sup>
- Occipital lobe: positioned at middle back of the skull. It's primarily responsible for controlling most of the vision and visual processing. <sup>[6]</sup>
- Temporal lobes: positioned at the left and right side of the brain they are above and around the ears from the sides. This area is primarily involved with hearing, memory, meaning, and language. <sup>[6]</sup>

The sensory cortex and the motor cortex which are responsible for the movement (motor) functions are narrow bands across the top middle area of the brain. This is primarily involved with balance, posture, motor movement, and some areas of cognition <sup>[6, 7]</sup>.

The cortical areas which are important for BCIs devices and useful in the analysis of the behavior of the EEG are as follow: (motor, somatosensory, posterior parietal, and visual) cortex <sup>[8]</sup>. Refer to Figure 1.

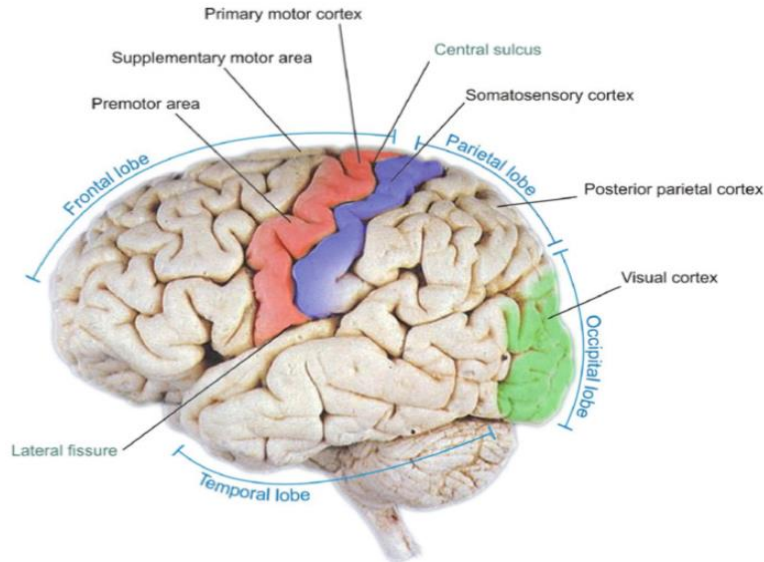


Figure 1: The cerebrum is being subdivided into four lobes.<sup>[1]</sup>

## 2.2 The Anatomy of Movement

Motor functions are almost involved into all of our life aspects, talking, gesturing, and walking any of our daily activity which require a decision and a movement response. But even a simple movement like picking up a cup of coffee can be considered as a complex motor function to study.<sup>[9]</sup> One of the main and most vital brain areas which is involved in any of the motor function is the primary motor cortex area named as (M1). The main role of the motor cortex area is to generate the neural pulses signals which will have control over the execution of the intended different movement functions.<sup>[9]</sup>

## 2.3 Electroencephalography (EEG):

EEG is defined as fluctuations or alternating electrical activity produced and recorded from the scalp surface by using electrodes or conductive media using amplifiers to accurately read the scalp voltage fluctuations<sup>[11, 10]</sup>.

It is also a field of interest to researchers as it's used to experiment neurological activity behavior<sup>[12]</sup>. The electrodes positioning have been named by letter and numbers, Letters representing different parts of the human brain lobes, even number refer to the right side of the brain yet odd refer to the left side of the brain.<sup>[11]</sup>

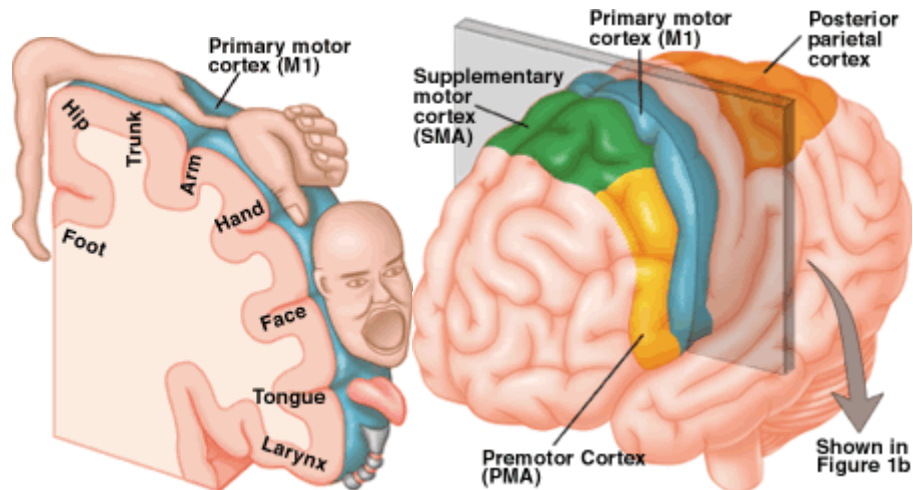


Figure 2: Principal cortical domains of the motor system.

- **Brain waves:**

Our brain emits electro-chemical impulses of different frequencies- resulted from the electrical movement in the neurons EEG has frequency bands can be categorized as follow delta (0.1 to 3.5) Hz, theta (4 to 7.5) Hz, alpha (8 to 13) Hz, and beta (14 to 30) Hz. [13]

## 2.4 Brain computer interface:

Several types of computer interfaces are designed and already implemented for disabled people [14]. Non-invasive BCI has witnessed a great leap in the last two decades in performance and reliability. [15]

BCI systems can be categorized in two types (endogenous or exogenous). [22]

1. **Endogenous type** depend on the nature of the electrophysiological, as the amplitude in one of the frequency bands. It's based on motor images (sensorimotor rhythms) or slow cortical potential (SCP) as discussed below. [22]

**Note:** Both types require high level training.

- **SCPs (slow cortical potentials)** can be illustrated as slow voltage changes found on the cerebral cortex, with in a duration of time from 0.5 seconds to 10 seconds. The Negative SCPs are mainly associated with movement. [22]



- **Sensorimotor rhythms based BCI.** Based on motor images or other mental tasks (rotating moving an imaginary object) those kind of mental tasks generates some changes in amplitude in  $\mu$  sensorimotor rhythms 8 Hz to 12 Hz and  $\beta$  sensorimotor rhythms 16 Hz to 24 Hz. <sup>[22]</sup>

**Note:** rhythms change while making, imagining or preparing an actual movement.

2. **Exogenous BCI systems** related to electro physical activity evoked by external stimuli. (P300 evoked potentials or (SSVEP)).<sup>[22]</sup>

**Note:** Does not require intensive training

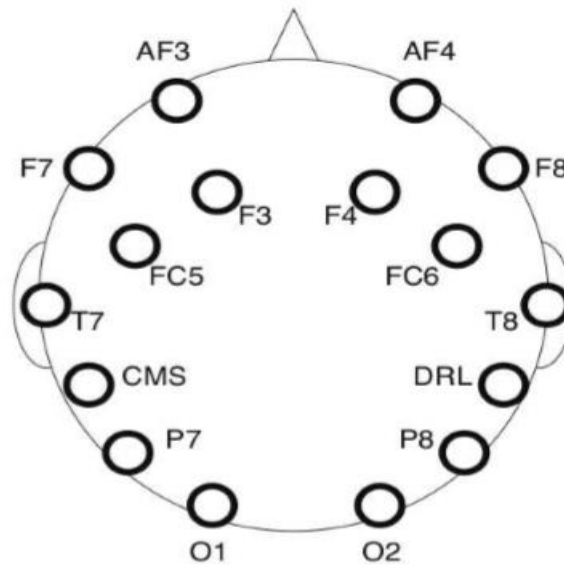
- **P300** evoked potentials. It's is an amplitude peak which can appear on the EEG signals around 300 ms after a stimulus being shown to the subject. The user is being shown with stimulus. The stimulus are infrequent they cause a P300 potential to appear on the EEG signal it's noticed mainly in the central and parietal regions.<sup>[22]</sup>
- **SSVEP.** Evoked visual potentials are detected the visual region after a number of choices are presented on screen in the form of stimuli flashing at unique rates. When a user gaze directly at one of these flashing stimulus brain produces signals of same frequency of the stimulus those can be noticed while analyzing the spectrum of the EEG signal.<sup>[22,25]</sup>

- **Emotiv EPOC:**

Emotiv EPOC is a consumer BCI based on EEG technology, making it much more affordable compared to ordinary EEG systems. It offers 14 electrodes mounted on headset that is effortlessly set up and connected wirelessly to a computer. <sup>[16, 17]</sup>

- **Emotiv SDK**

For The Emotiv® EPOC headset comes with a Software Develop Kit (SDK).<sup>[18]</sup> This SDK allows developers to get data from the device in real time, by introducing its functionality into various developing environments<sup>[19]</sup>.



*Figure 3: The Emotiv EPOC headset is using..*

*(Adopted from Duvinage, M., Castermans, T., Dutoit, T., Petieau, M., Hoellinger, T., De Saedeleer, C, A P300-*

*Figure 1)*

The Emotiv APIs which is provided by the standard SDK consist of three main suits or algorithms internally programmed and they are as follow; The Expressiv<sup>TM</sup> which detect the facial expression, the Affectiv<sup>TM</sup> suite detects the mood, concentration, meditation and excitement, and the Cognitiv<sup>TM</sup> suite which is mostly related to the movement and motor functions so it can be used to detect /translate the user intent.<sup>[20, 21]</sup>



Figure 4 Emotiv Epoc Headset

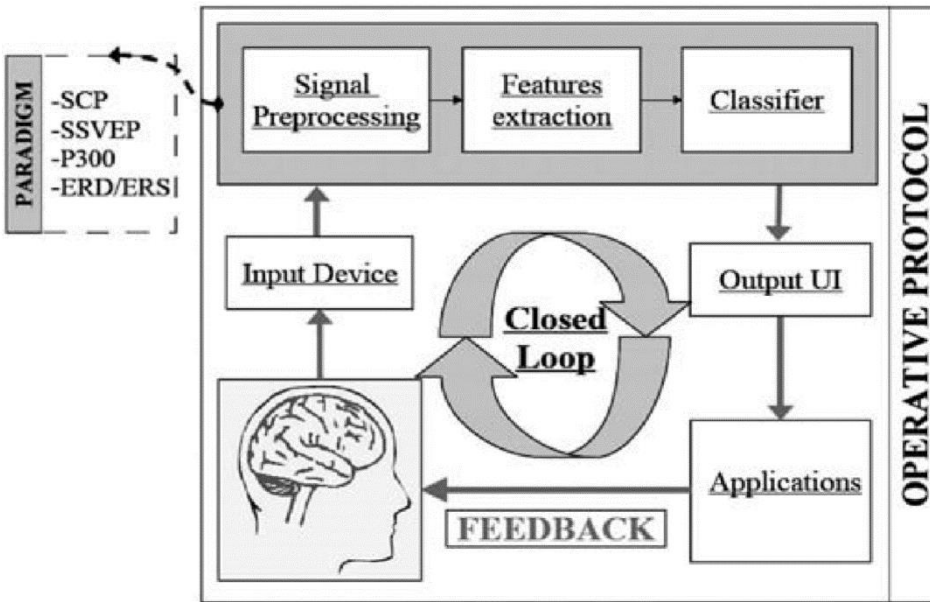


Figure 5: A typical BCI System architecture

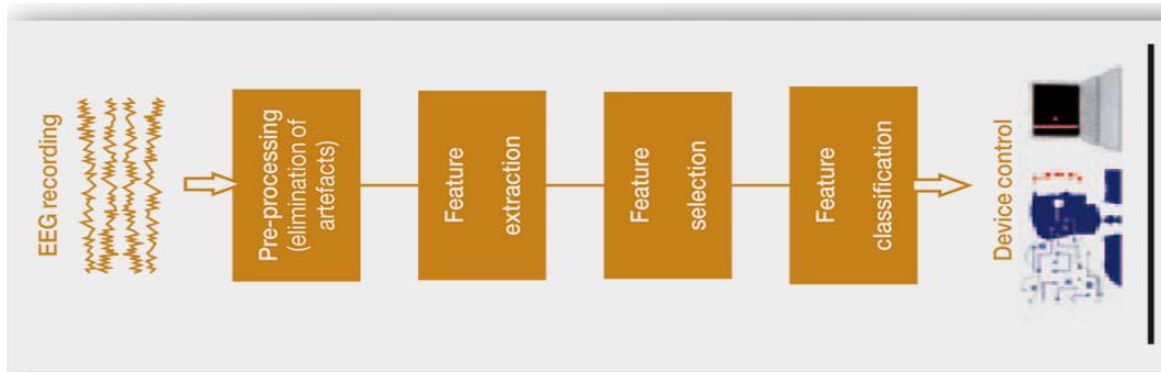


Figure 6: Schematic diagram of signal processing applied to an EEG recording from signal acquisition through to control commands generated by the final application. <sup>[22]</sup>

### 2.4.1 SSEVEP:

SSVEPs BCI are biological feedbacks originated from visual cortex corresponding to a flashing or flickering stimulations in the visual center. They are detected within the visual region after a number of stimulus are presented on screen in the form of stimuli flashing at unique rates. When a user gazes directly at one of those flashing stimulus our brain will produce some response signals which carry the same frequency of the flickering stimulus.

In order for the SSVEP to operate as expected, the two colors change must happen at the same intervals (for a screen which has a refresh rate of 60Hz it's advised that the possible frequencies which can be used should be 30, 20, 15, 12, and 10)

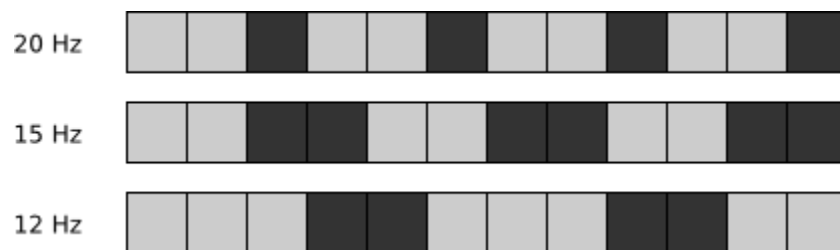


Figure 7: Example of Various frequencies

SSVEP based BCIs, visual stimulus modulated at different frequencies are simultaneously presented to the user. Each pattern is associated with an action in an output. When the user focuses his eyes attention on a certain pattern which is shown as stimulus on a screen or any other display device/ method, the corresponding stimulating frequency are dominantly appearing in the spectral of the EEG signals recording. The action associated to that dominant frequency is then performed followed after the EEG signal analysis and processing.

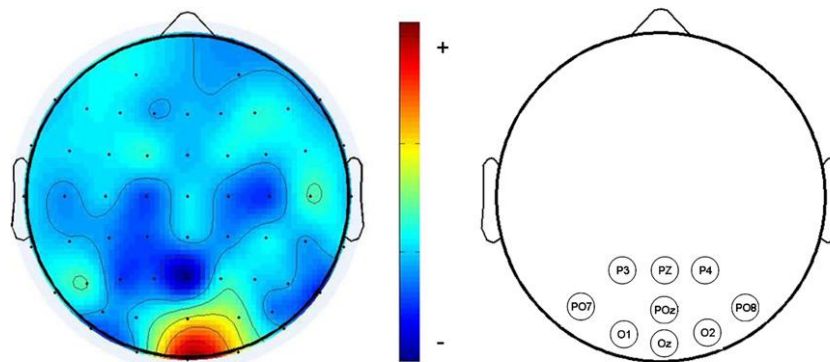


Figure 8 : EEG Topography SSVEP Effect on visual cortex

### 2.4.1.1 Open vibe:

Open Vibe is a signal processing platform aims for experimenting and designing which is related to BCI and EEG signals in general. Can also be referred to as real-time neuroscience it's basically used to acquire, filter, and classify EEG signal on real time. The platform comes equipped with many signal processing tools and algorithms used to filter and classify signals. Epoching, signal averaging, Linear combinations, Spatial and Temporal like Common Spatial filter, Fourier transformations .several machine learning algorithms are included with the platform to translate signals into command like. LDA, SVM,Classifier combination (one versus the rest) [25]

Signal processing algorithms is being represented each by a block or a box. To pass the input from a box to the other simply linking them together. It's the same as Simulink Boxes (Matlab) each box has a function and inputs and outputs

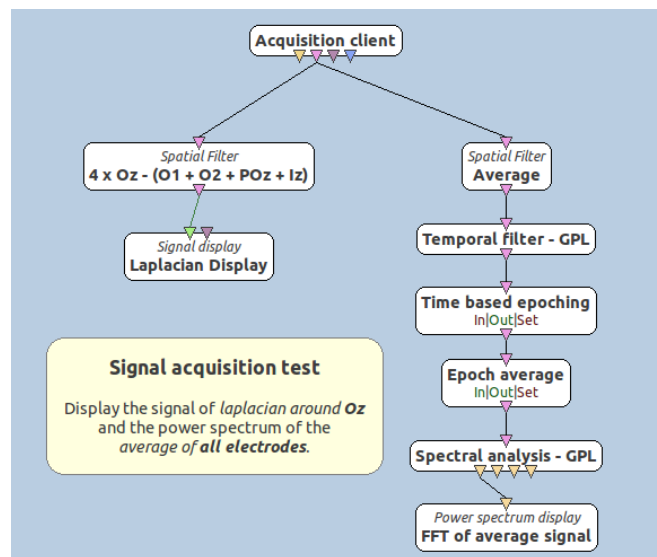


Figure 9:open vibe graphical interface

## Classification and Feature extraction:

### CSP (Common Spatial Pattern)

The CSP algorithm's function is to increase the signal variance for one event and minimize the variance for the second event. This box computes spatial filters regarding to the CSP algorithm. This algorithm aims to improve the discrimination of two types of events.

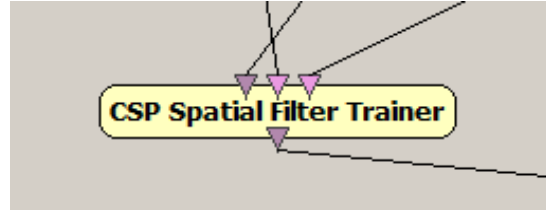


Figure 10 Spatial Filter Trainer

This algorithm is being used for discriminating two events like motor-imagery tasks (e.g. left and right hand movement). Also can be used for any other experiment where there is two events which we need to discriminate between them.

Let's consider the following example :

- Input channels list: C3 C4 FC3 FC4 C5 C1 C2 C6 CP3 CP4 (10 channels)

Spatial filter coefficients: 4 0 -1 0 -1 -1 0 0 -1 0 0 4 0 -1 0 0 -1 -1 0 -1 (20 values)

- Number of output channels: 2
- Number of input channels: 10

The output channels becomes :

$$\begin{aligned} &4 * C3 + 0 * C4 + (-1) * FC3 + 0 * FC4 + (-1) * C5 + (-1) * C1 + 0 * C2 + 0 \\ &* C6 + (-1) * CP3 + 0 * CP4 \\ &C3 - FC3 - C5 - C1 - CP3 \\ &0 * C3 + 4 * C4 + 0 * FC3 + (-1) * FC4 + 0 * C5 + 0 * C1 + (-1) * C2 + (-1) \\ &* C6 + 0 * CP3 + (-1) * CP4 \\ &C4 - FC4 - C2 - C6 - CP4 \end{aligned}$$

This is basically a Surface Laplacian around C4 and C5.

$$Ock = \text{Sum on } j ( Sjk * ICj )$$

### Linear discriminant analysis (LDA)

Used as a pattern recognition for machine learning for finding the linear combination of the EEG features which can be associated, characterize or separates two or more classes of objects or events.

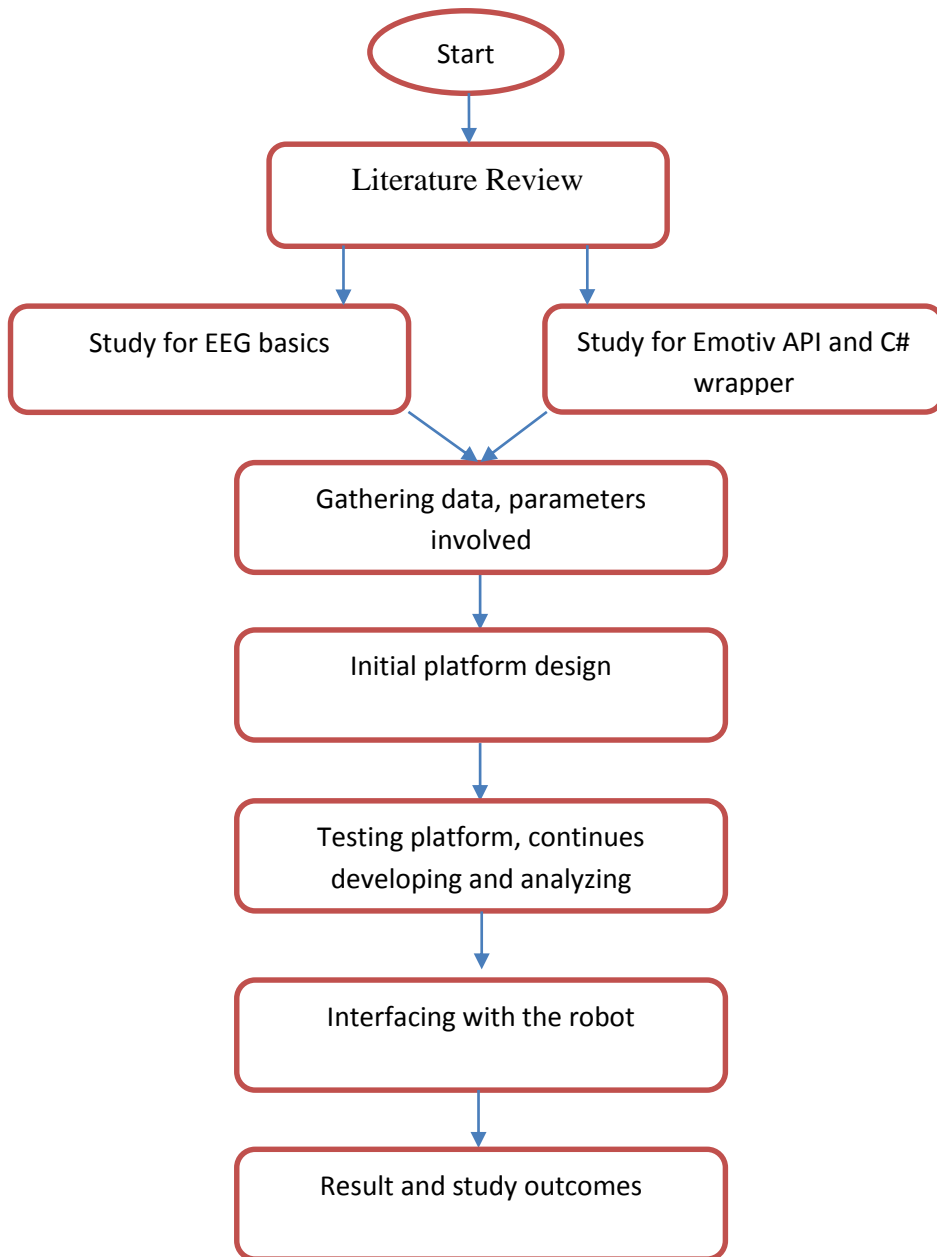
# CHAPTER 3

## METHADODOLOGY

### 3.1 Project Workflow

#### 1- Experiment 1 (Using Emotive detection suits)

Figure 3 shows a work flow consisting of the planned process workflow for this project.





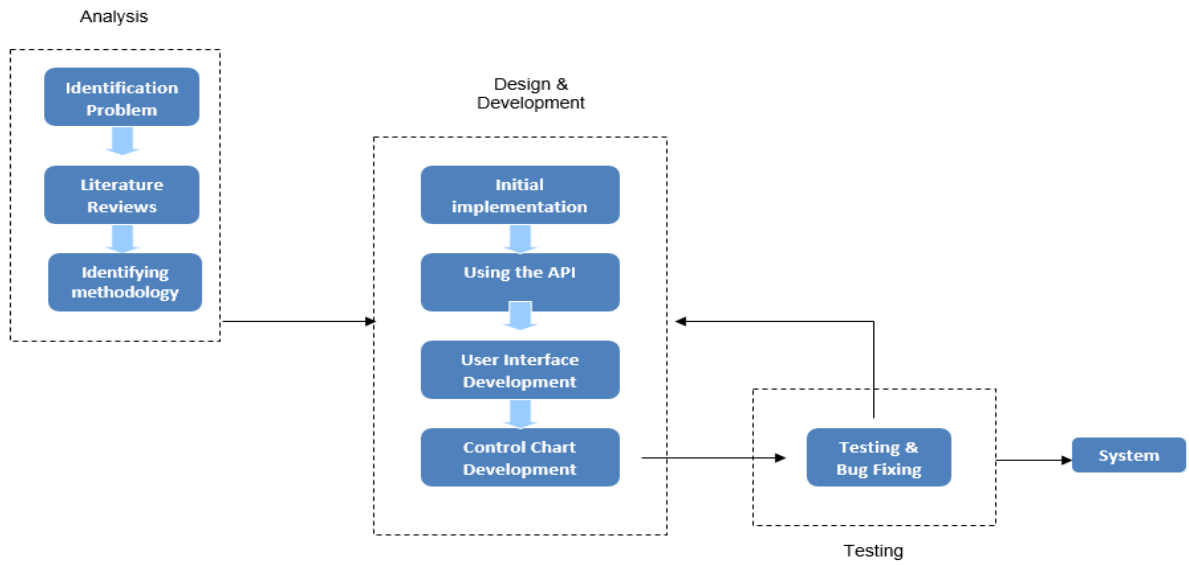
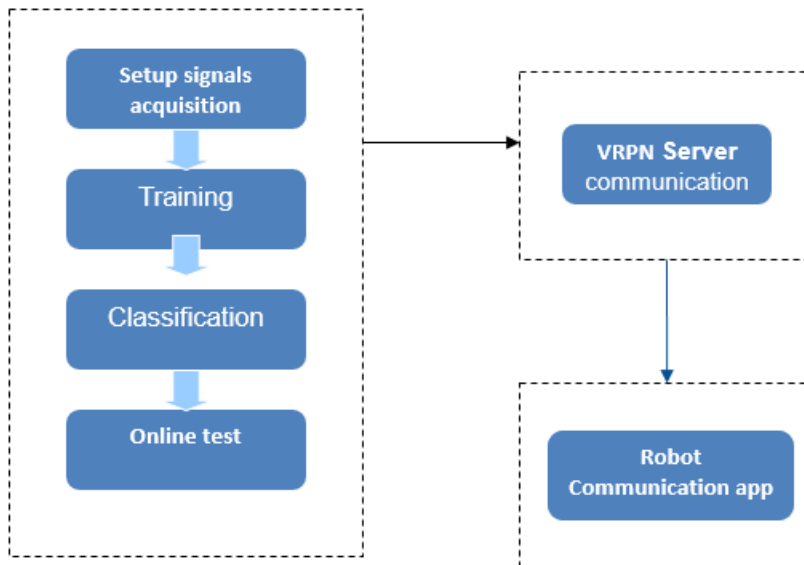


Figure 11: Flow chart for implementing first stage for the project.

## 2-Experiment 2 (SSVEP)



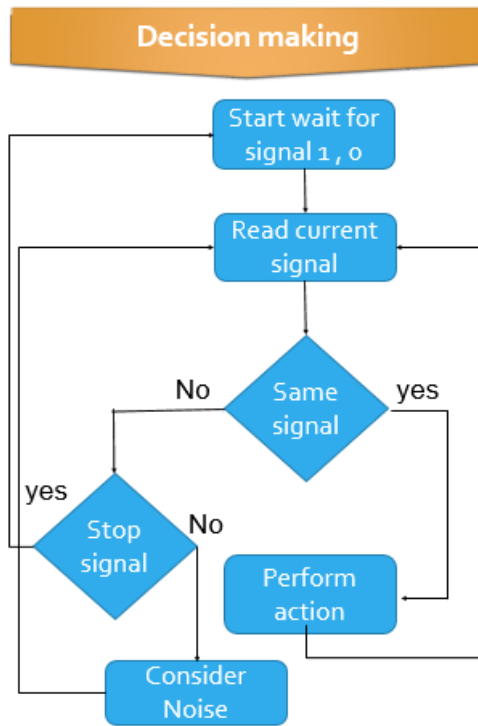


Figure 12 : Decision making algorithm

## 3.2 Research methodology

The methodology that will be used to complete this platform is agile development approach. It refers to the implementation of iterative and incremental development, It promotes adaptive planning, creative development and fast and efficient delivery.

## 3.3 Tools

### 3.3.1 Software:

There is some software and development tools will be used to complete the project

- **Emotiv SDK (Software Development Kit)**

SDK contains high resolution, neuro-signal acquisition and processing wireless neuro-headset the SDK API is exposed and can be used with various programming language like Visual studio (visual basic, C, and C #), java and python to create creative application using the SDK libraries and algorithms for detection different kind of movement or actions

- **Visual Studio (2010)**

It's development tool based on .NET framework giving the developer a vast choices of programming language for designing application on windows platform, the programming language which will be used mainly it will be C#.

- **Open Vibe:**

Open Vibe is a software platform aims to experiment and design and test BCI (Brain computer interface systems). It's also can be described as a real-time neuroscience it's used to acquire, filter, process, classify brain signals in real time.<sup>[25]</sup>

- **VRPN (Virtual-Reality Peripheral Network)**

Is an independent network-transparent system for accessing virtual devices implemented by Russell M. Taylor II (Department of Computer Science of the University of North Carolina at Chapel Hill). Mainly used to provide a practical interface communication for various input devices. Also it provides

1. Time-stamping of data
2. Automatic reconnection

### 3.3.2 Hardware:

- **Emotiv EPOC**

The Emotiv EPOC is a 14 channels brain computer interface. It uses 14 different sensors allocated in different parts around the scalp to read electrical signals generated by the human brain to monitor and detect user thoughts (Cognitive), feelings and expressions (expressive).<sup>[16]</sup> Using its own algorithms implemented in the SDK.

- **NXT:**

LEGO NXT is one of the basic programmable robotic kit, there is a small computer called Cara which contain the programming algorithms. It can process input from various sensors up to four and control up to three motors.

- SDK: information on host USB drivers.
- BDK: protocols for Bluetooth communications.

### 3.4 Gantt chart

#### 3.4.1 FYP 1

No.	Detail/ Week	1	2	3	4	5	6	7		8	9	10	11	12	13	14	
1	Selection of Project Topic	█	█						Mid-semester break								
2	Submit Project Proposal			█													
3	Preliminary Research Work																
4	Submission of Extended						20-2										
5	Project work continues	Gathering Requirements															
6	Proposal Defense										█						
7	Project work continues																
8	Project work continues																
9	Project work continues																
10	Submission of Interim																
11	Project work continues																
12	Submission of Interim													█			

#### 3.4.2 FYP 2

No.	Detail/ Week	1	2	3	4	5	6	7		8	9	10	11	12	13	14	
1	Project Work Continues								Mid-Semester Break								
2	Project work continues	Coding															
3	Submission of Progress Report				█												
4	Project work continues																
5	Project Work Continues																
6	Project work continues																
7	Pre-SEDEX										█						
8	Project work continues																
9	Submission of Draft Report																
10	Project work continues																
11	Submission of Dissertation (soft bound)																
12	Project work continues																
13	Submission of Technical Paper																
14	Project work continues																
15	Oral Presentation																
16	Project work continues																
17	Submission of Project Dissertation (Hard Bound)																

## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4.1 RESULTS AND DISCUSSION:

##### 4.1.1 First Approach Using Emotiv API

Among the achievements which are expected in this project, the first one would be obtained after the feasibility studies of this project. The collection of all information needed on the study which includes the theory of the EEG signal and the structure of BCI systems and an implementation of a simple BCI controlled robot based on Emotiv to be achieved

Part I: the objective of the first stage of the project is to use the Emotiv API to communicate with a simple robot for a movement into four directions shown below in Figure 7: the platform shown in figure 8: the architecture for the system.

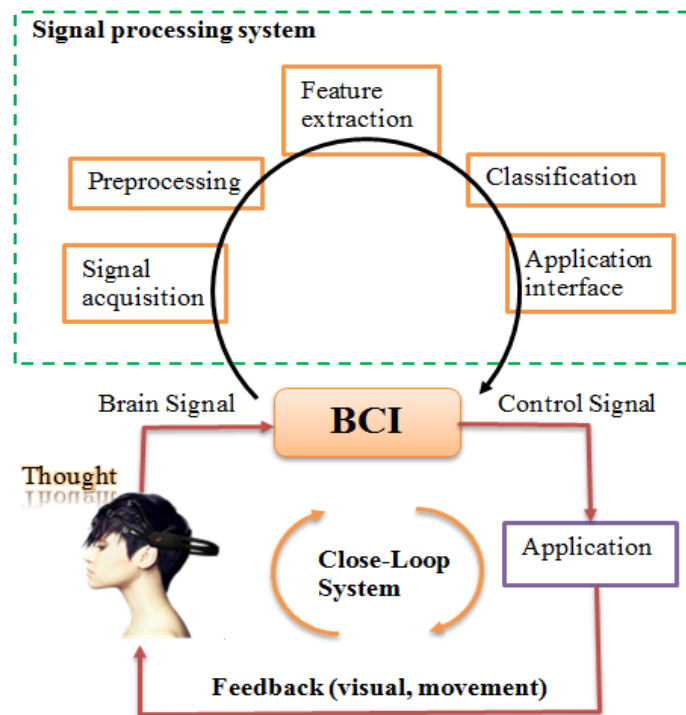


Figure 13: BCI architecture

We are able to move the robot into 4 directions Forward ,backward,left and right.

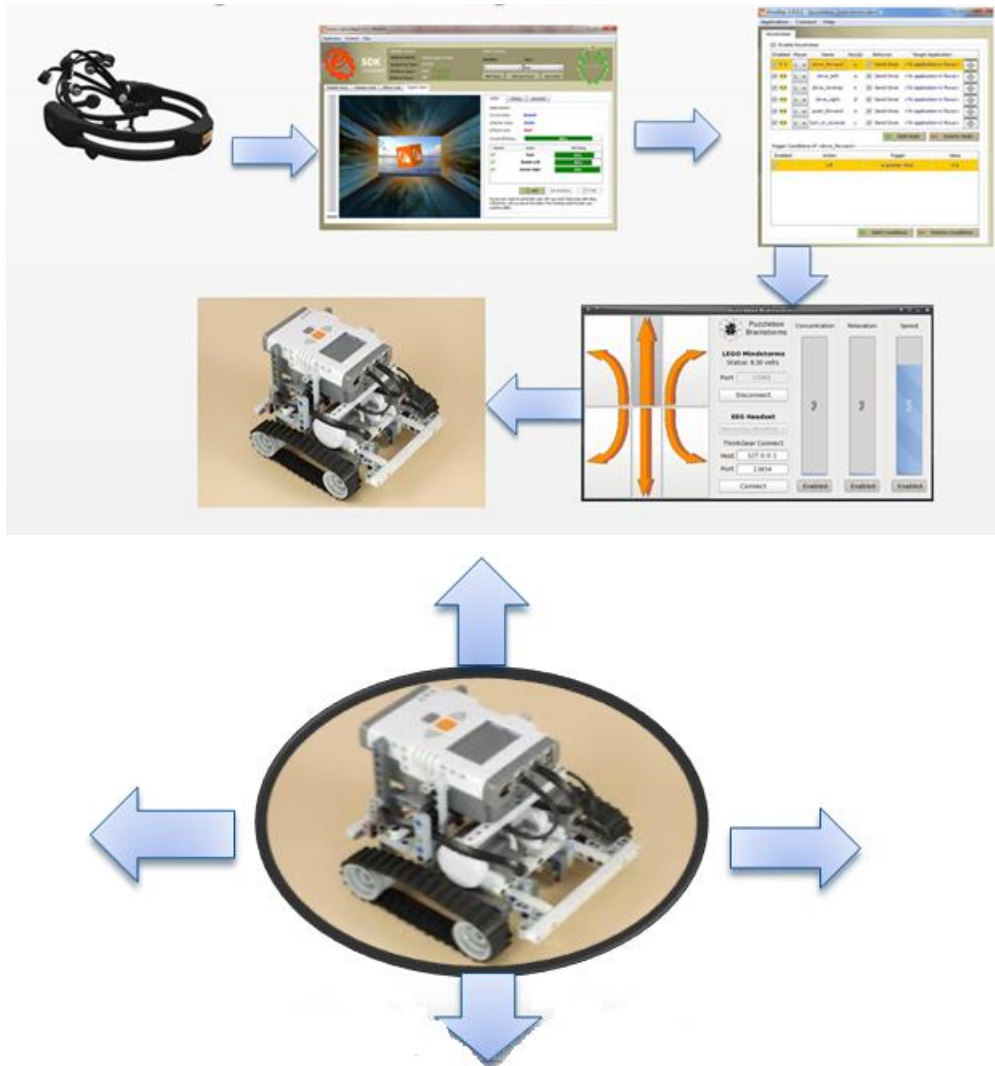


Figure 14: System design architecture

Emotiv SDK, Emokey, puzzle box and NXT robot is used to achieve a BCI system to control a robot according to user thought. The subject should go for extensive training using the cognitive suite provided by Emotiv.

The trained movement is then attached to the Emokey program to set the rules so when a user think about pushing the Emokey send the key W to puzzle box which is then stimulate pressing the letter (W) and send a move forward command to the robot same procedure is then repeated to move the robot back ward and left and right.

Shown below the flow diagram for designing the system:

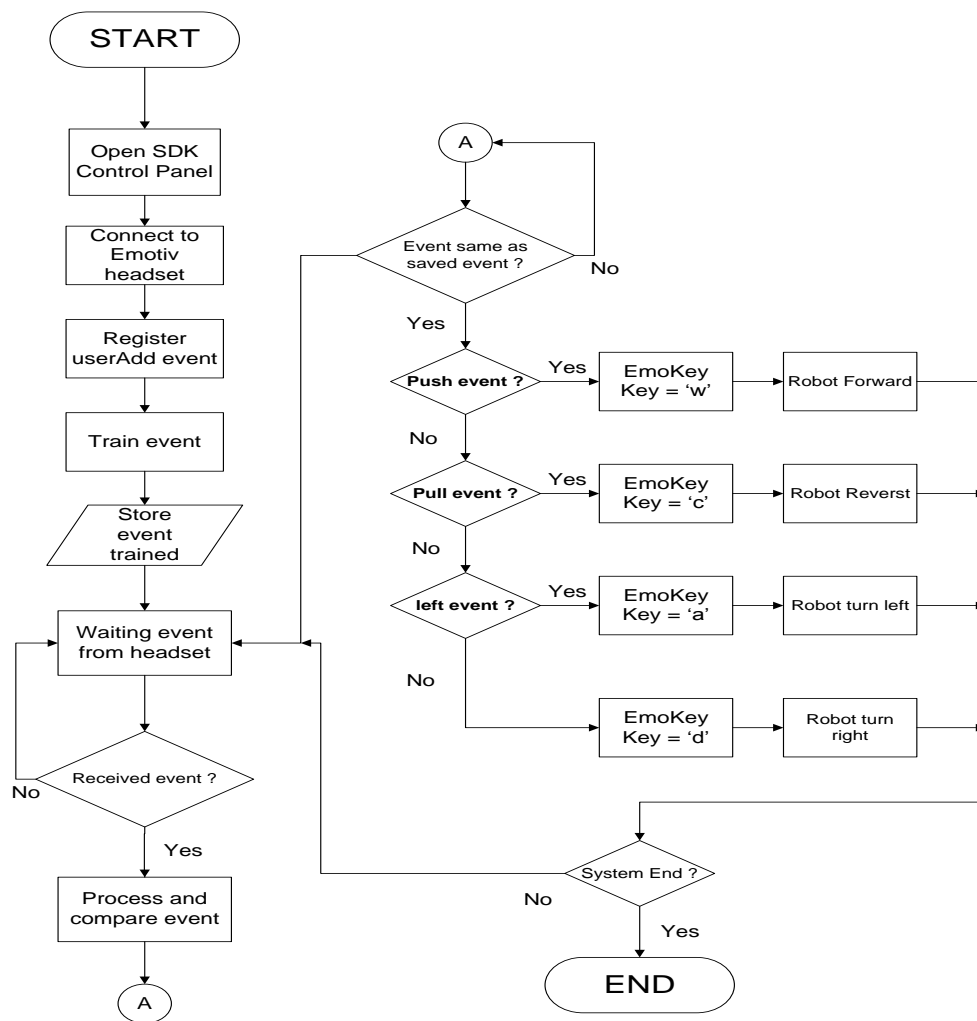


Figure 15: flow diagram for using Emotiv with puzzle box to control the NXT robot



Three main programs are used to achieve the objective of controlling the robot with Emotiv headset:

- **Control panel** for Emotiv: is used for training purposes for different movement.
- **Emo key:** is a small program used to associate the Emotiv control panel with different applications, simply by setting rules and actions for different movement.
- **Puzzelbox:** a simple program used to communicate with the lego NXT robot to control its directions.



Figure 16: Control panel



Figure 17: Emokey

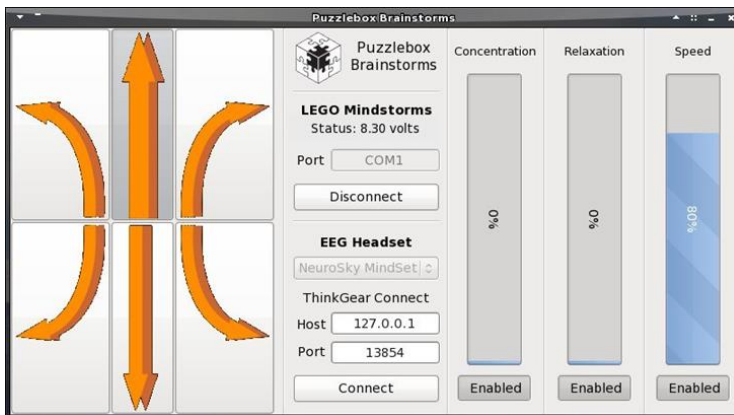


Figure 18: puzzle box

While designing the system some drawbacks have been noticed listed below the main drawbacks concluded.

1. Emotiv Cognitiv training is so difficult

There is a difficulty varies from user to another while training the cognitive suite for the Emotiv SDK.

2. Required high Level relaxation and concentration.

The process is so tiring and it needs a clear mind and a high level for concentrations subjects usually feels tired after approximately an hour of training and their skill radically decrease for repeating the same thought again.

3. The interface between Emotiv headset and NXT Robot requires several software – Emotiv SDK, EmoKey and puzzle box.

### 4.1.2 Second Approach Using Designing SSVEP BCI (Using Open Vibe)

Using open vibe as a designing environment for designing our system.

The first step is to connect the open vibe with the head set and that's is done by using the Acquisition server Module.

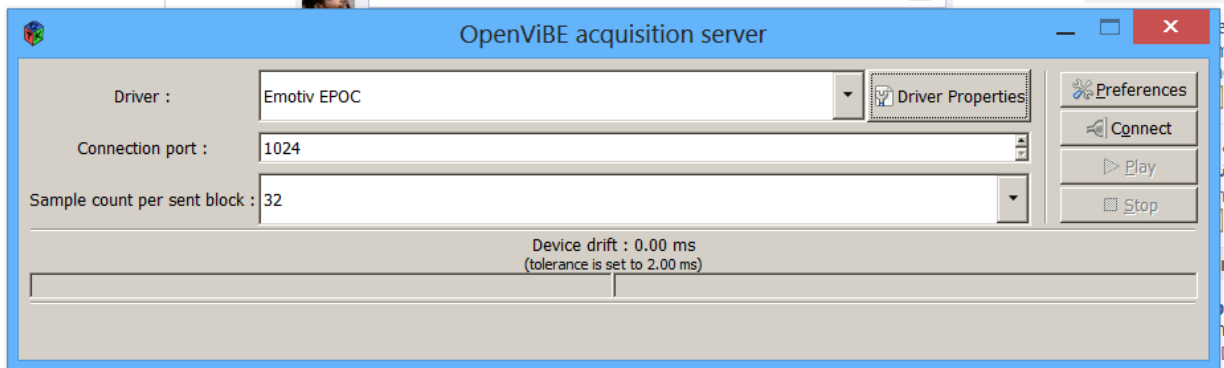


Figure 19: acquisition server

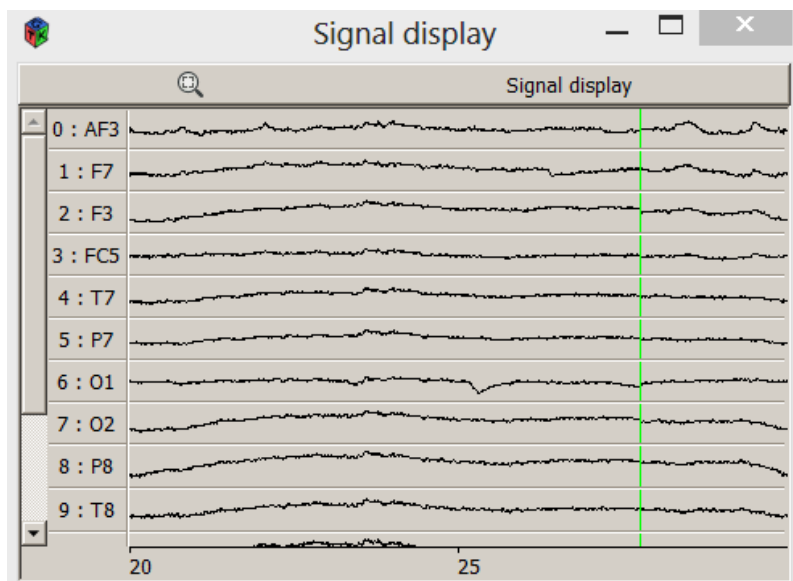


Figure 20: Sample of EEG Recording

We should carefully select the type of the acquisition device we are using we are using and then we will port it to the Design framework. This can be simply done by selecting our own acquisition device from the acquisition server program which will establish a server and wait for connection to transmit EEG singles.

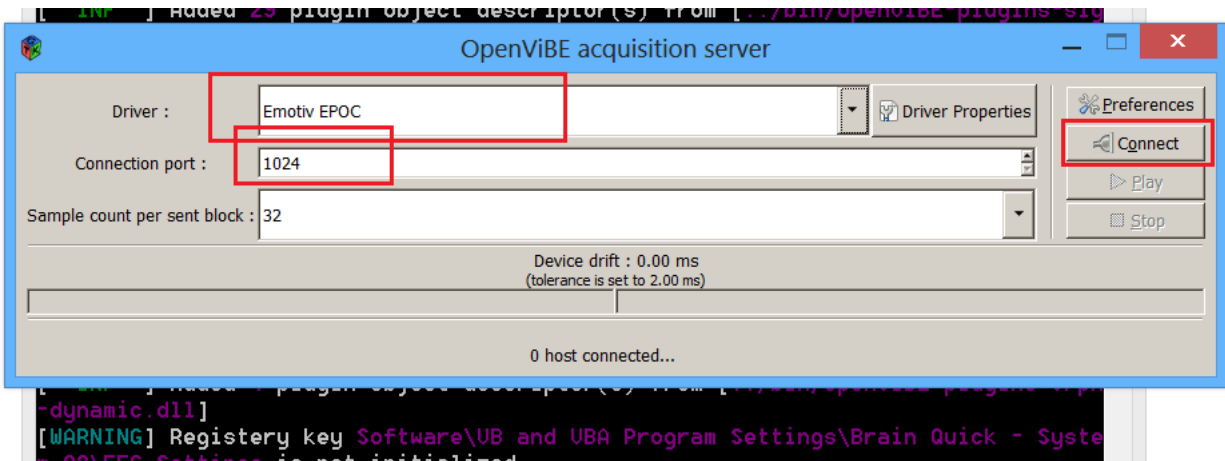


Figure 21 Open vibe acquisition server

Name	Description
Acquisition	
Acquisition client	A generic network based acquisition client
Classification	
Classifier processor	Generic classification, relying on several box algorithms
Classifier trainer	Generic classifier trainer, relying on several box algorithms
Confusion Matrix	Make a confusion matrix out of classification results coming from one cl
Voting Classifier	
Feature extraction	
Feature aggregator	Aggregates features in a vector
File reading and writing	
Brainamp	
CSV	
EDF	
GDF	
OpenVIBE	
Signal Concatenation	Concatenates multiple signal streams
Samples	
Clock stimulator	Periodic stimulation generator
Codec Toolkit textbox	Sample box to test the codec toolkit. Identity (input = output).
Identity	Duplicates input to output
Simple 3D viewer	3D object handler/viewer
Sinus oscillator	Simple sinus signal generator
Time signal	Simple time signal generator (for use with DSP)
Scripting	
Lua Stimulator	Generates some stimulations according to an Lua script
Matlab Scripting	Executes matlab scripts. To be used correctly, you must have Matlab ins
Signal processing	
Averaging	
Epoch average	Averages matrices among time, this can be used to enhance ERPs
Signal average	Computes the average of each input buffer.
Basic	
Channel Rename	Renames channels of different types of streamed matrices
Channel Selector	...

Figure 22: Sample of Open vibe Boxes

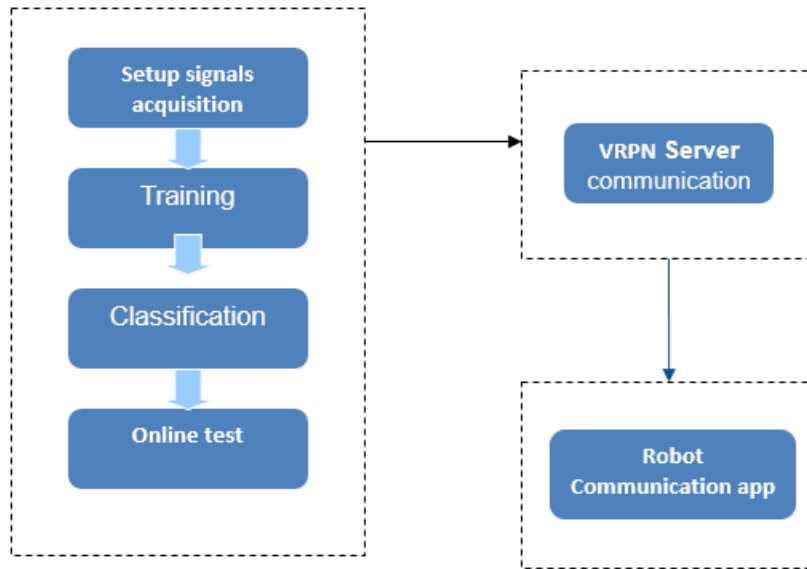


Figure 23: Flow diagram of the system

### **Training**

The system starts with a training session where the user will be asked to look and focus at the visual stimulus randomly (pre-configured ordered) used for acquisition of the training data necessary to train the SSVEP classifiers for the sake of collecting data which will be used to generate classifiers later on .



Figure 24: SSVEP Stimulus Screen

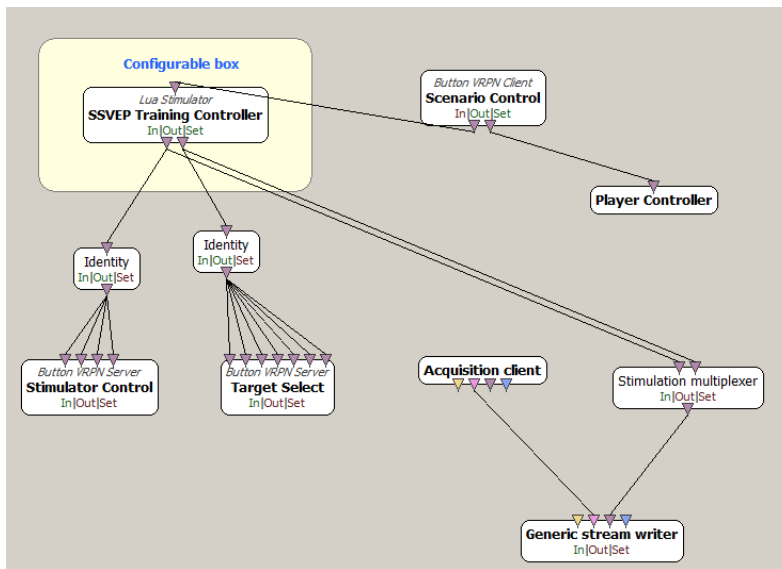
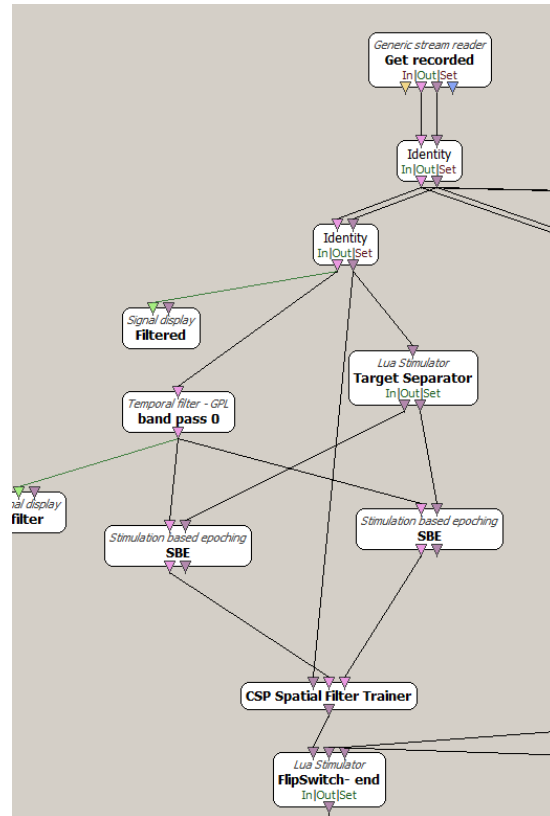


Figure 25: Configuration box for training Data

## Feature extraction

In this stage, certain features are extracted from the digitized EEG signal. Determined frequency range is elected and the amplitude relative to some reference level is measured. The features can be represented as some certain frequency bands on the power spectrum. If the feature combinations are representing mental tasks which they may overlap each other, it will be extremely difficult to classify those mental task even with a well implemented classifier.

On the other hand, if the feature sets are not overlapping and different, most of the classifiers can be used to classify them.<sup>[25]</sup>





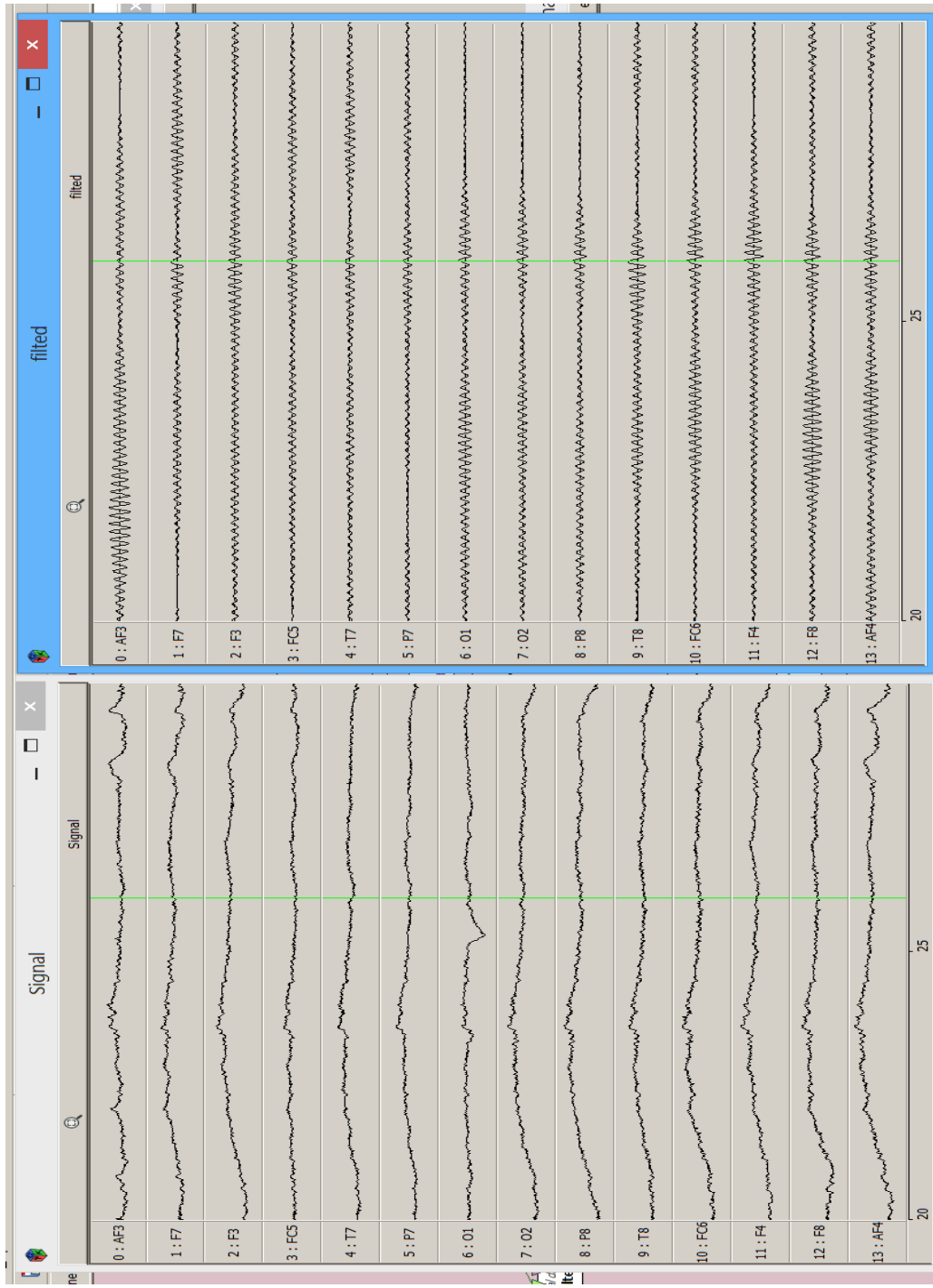


Figure 26 EEG signal on left, Band pass filtered signal on right

## Classifier Training

This scenario will create the final classifiers for online testing. For each frequency used there will be on classifier file.

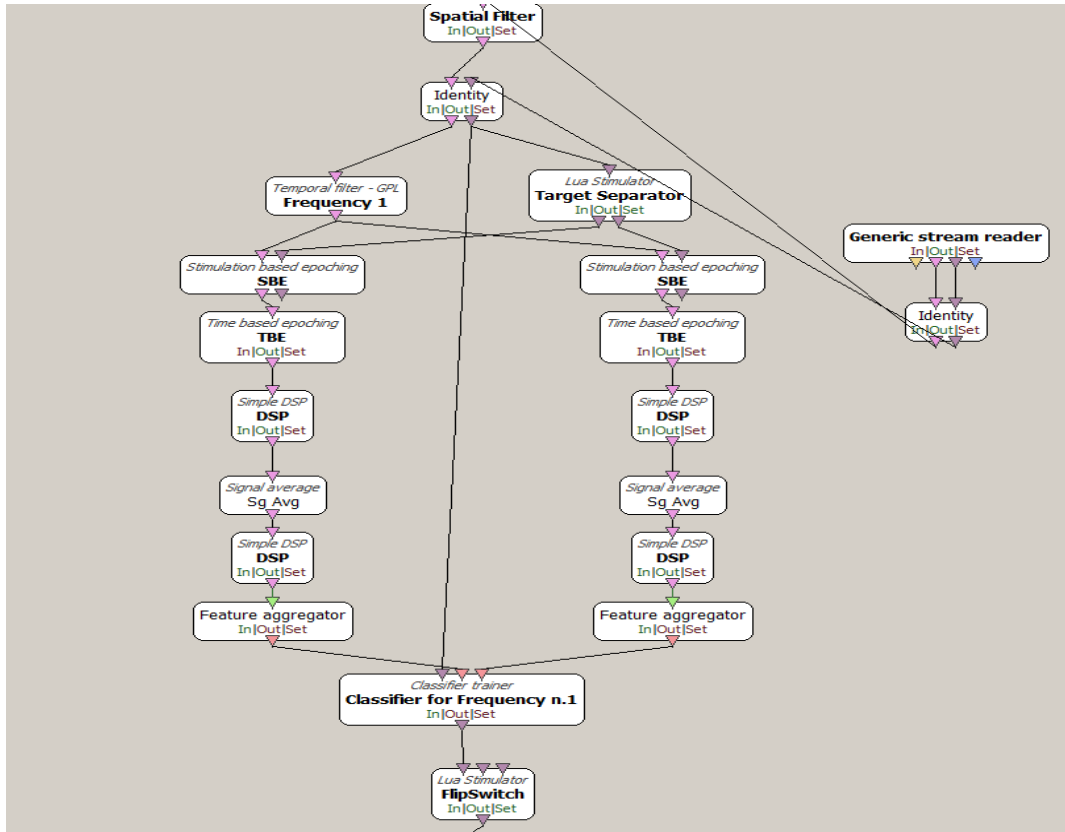


Figure 27: Classifier boxes

## Classification

The features extracted step is the input for the classifier algorithm. Mostly all BCIs can be set to classify various number of classes, around 2 - 5. The classifier could be a simple linear model to a complex nonlinear like a neural network that can be trained to select the classes intended.

Basically the classifier can calculate the probabilities for class's inputs and it will choose the one with the highest probability at this stage. <sup>[25]</sup> (Refer to appendix B)

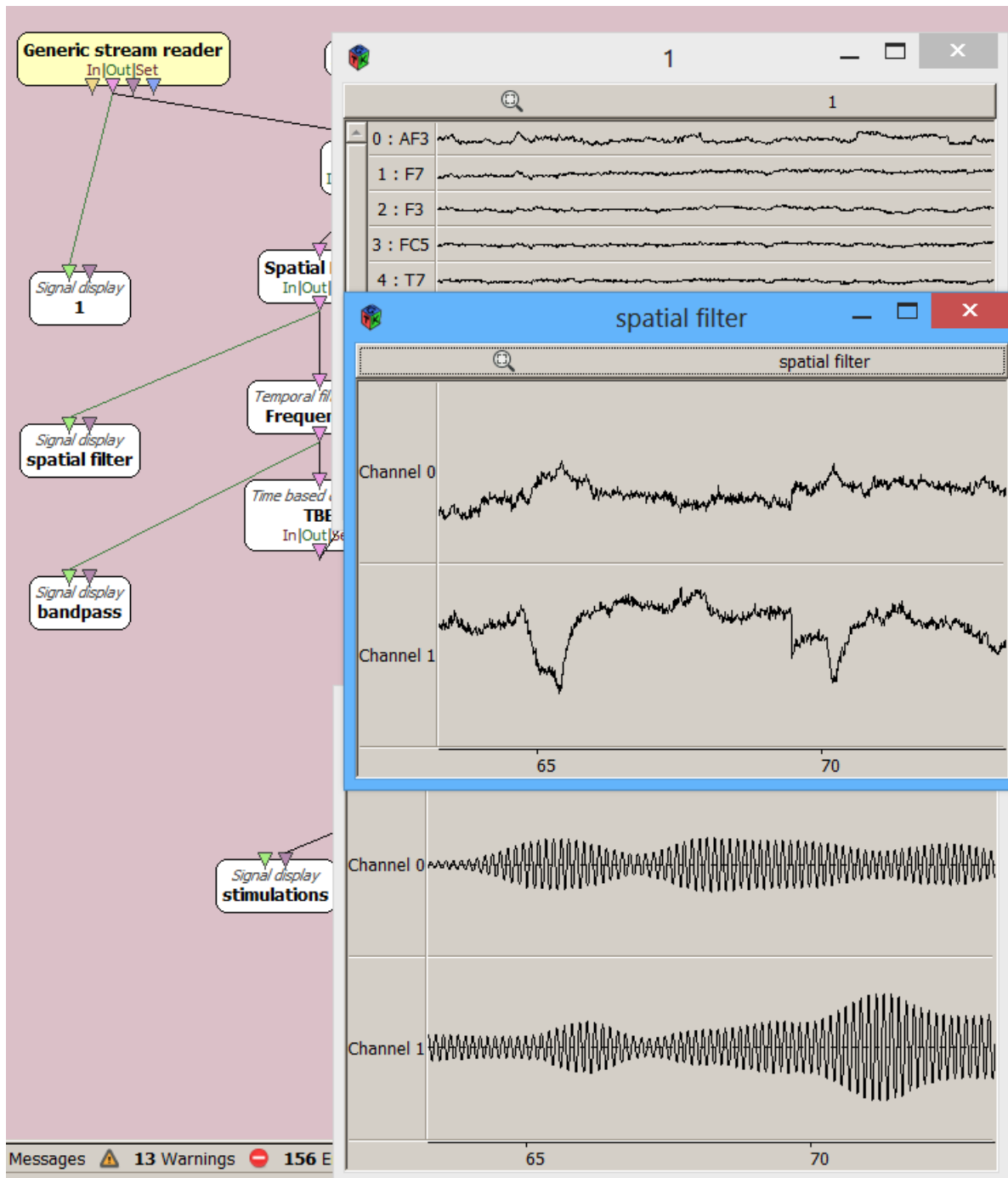


Figure 28: Spatial Filter (output signals) followed by a band bass filter

**Online test :**

Using the generated classifiers when the user focus on one of the stimulus an event in the VRPN server will be evoked and another Standalone VRPN Client will read those events and act up on it , our client in this condition is the application responsible for the robot navigation and the decision making.( Figure 29)(Refer to appendix B)

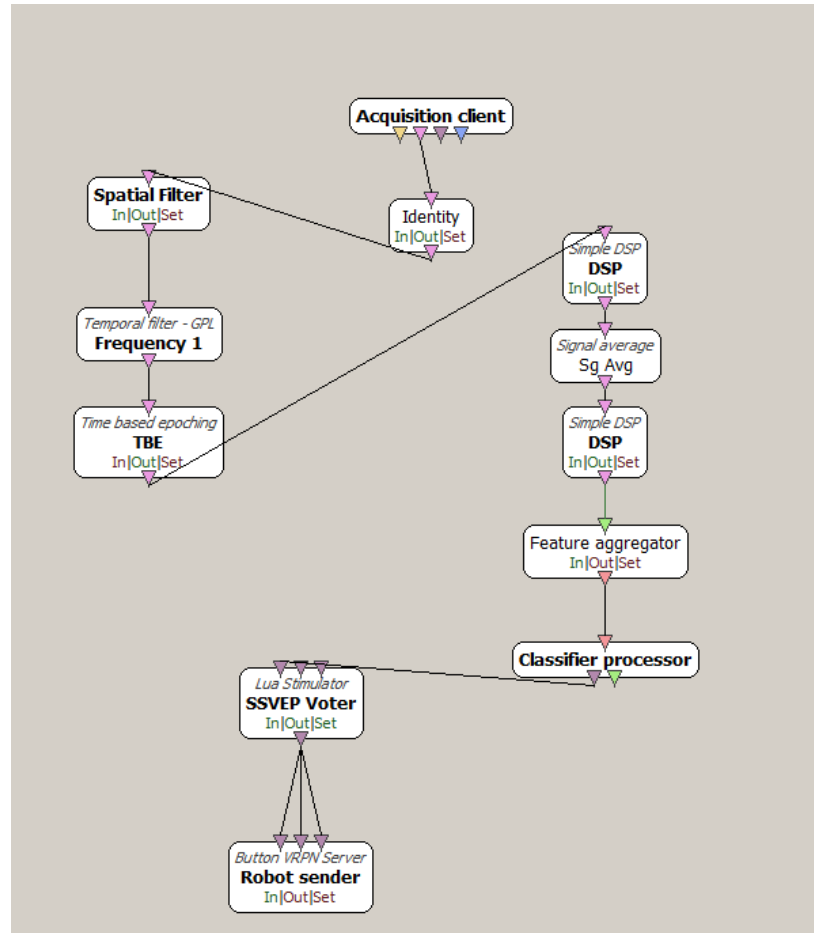


Figure 30: Online testing with a robot

**Stimulus script:**

The scrip will display the flickering stimulus to the user where the user only needed to focus on of them to achieve the result intended , the stimulus displayed here are the same as the one shown on the training scheme (EEG acquisition step).

(Refer to appendix B)



Figure 31: Stimulus for online test

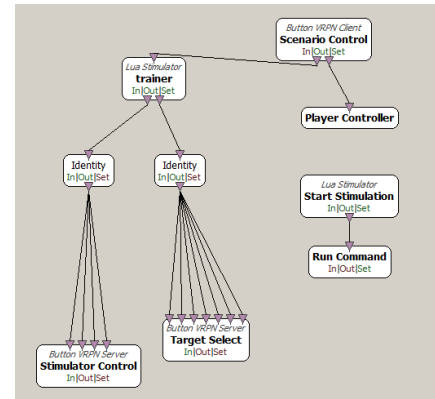


Figure 32 : Script for online testing

## The main in the middle (Interface program)

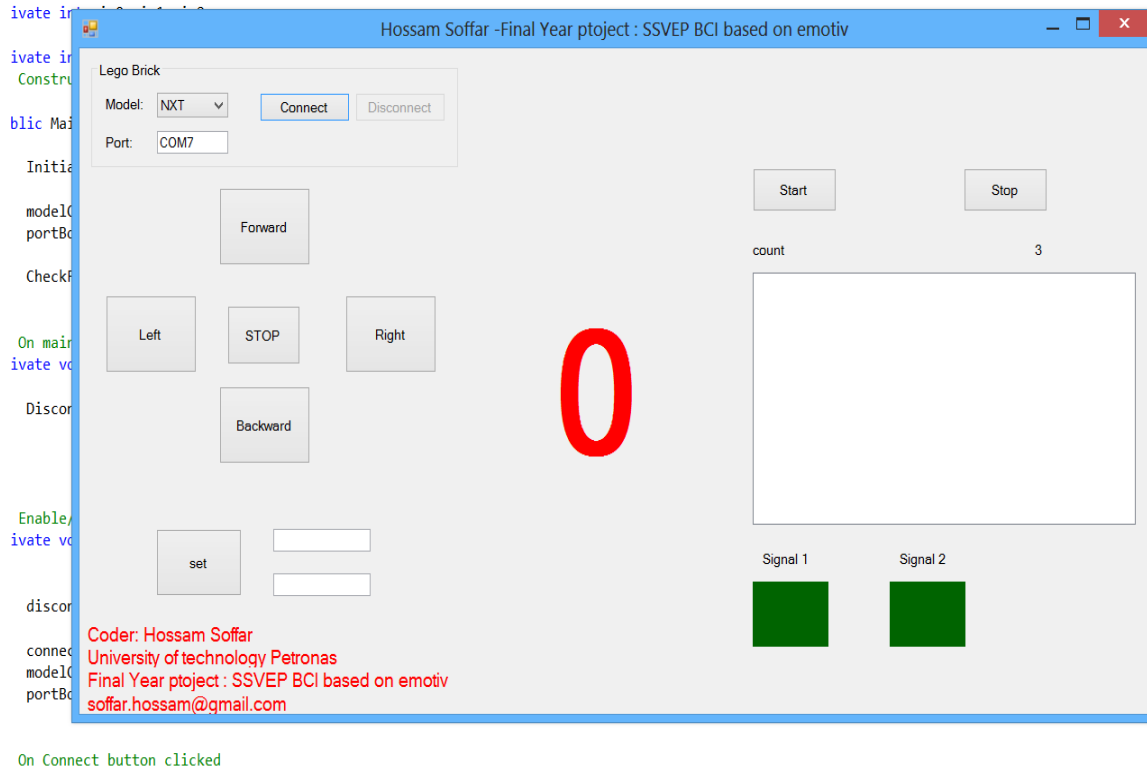


Figure 33: GUI for the program interface

A C# program is written to act as a man in the middle which will take the processing platform events , reprocess it and control the robot, the program initiate a Bluetooth connection with the NXT report to send the commands , at the same time a computer based network called VRPN (Virtual reality peripheral network ) will handle the communication between the interface program and the processing platform as a TCP/IP communication the program will receive which even the processing program detects and perform the action corresponding to it .

```

// Set power of A and C motor of Lego RCX brick
public void SetMotorsPower( float leftMotor, float rightMotor )
{
    int leftMotorPower = (int) ( leftMotor * maxMotorPower );
    int rightMotorPower = (int) ( rightMotor * maxMotorPower );

    // System.Diagnostics.Debug.WriteLine( "l: " + leftMotorPower );

    // return if there are no changes
    if ( ( leftMotorPower == lastLeftMotorPower ) && ( rightMotorPower == lastRightMotorPower ) )
        return;

    // check if we need to stop
    if ( ( leftMotorPower == 0 ) && ( rightMotorPower == 0 ) )
    {
        rcx.SetMotorOn( RCXBrick.Motor.AC, false );
    }
    else
    {
        int motorsToOn = 0;
        int motorsToOff = 0;
        int motorsToMoveForward = 0;
        int motorsToMoveBackward = 0;
    }
}

```

Figure 34: Sample code for establishing connection

The whole process starts when the subject starts monitoring the transmitted signals from Openvibe we click connect means we initiate an underground link between two programs and our program ready to receive the detected events, looking at the log list box it list the signals received when the event starts and when it ends the more the user focus on the stimulus the more frequent only one signal will appear.

Here the program will take the decision to perform the action linked with each signal, if received signal is 2 means stop moving the robot if it's 1 then rotate right if it's 0 then move forward.



Figure 35: interface program in action

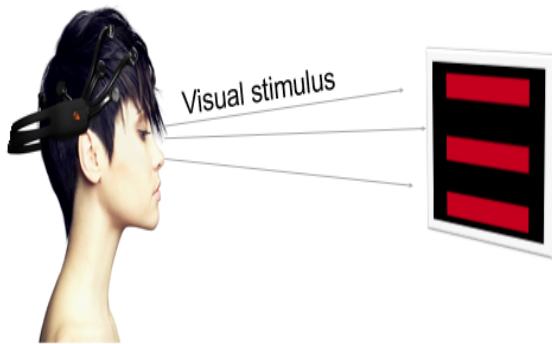


Figure 36: System Flow

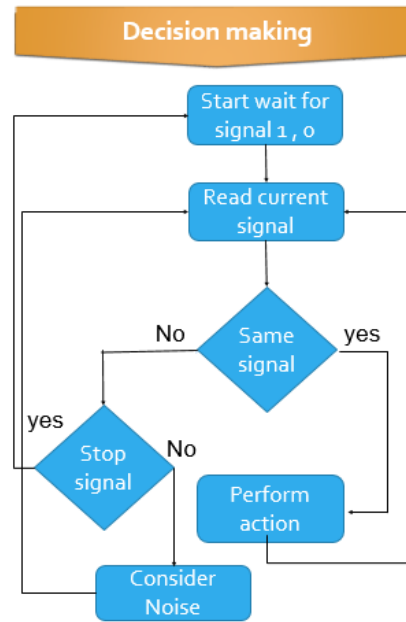


Figure 37: Decision making algorithm

The decision making scheme is introduced as it was noticed that in some events the processing program is not giving accurate reading of the events which will lead to low performance of the framework since the robot will move into position which is not intended, the algorithm works as follow the program will only consider an event as a valid event only if its detected more than 3 times in one second this particular even will be going on, till the user starts to focus only on the stop stimuli , then the use can select another movement , which means that if we want the robot to move forward we have to give a stop command after the forward event then we can rotate , if we are moving forward then we directly focused into the rotating stimulus the program will consider it as noise and it will keep moving forward so after each even a stop even should be triggered.

## CHAPTER 5

### CONCLUSION AND RECOMMENDATIONS

#### 5.1 CONCLUSION

A robot navigation framework based on EEG was successfully built, a user can navigate a robot by only using his brain EEG signals the framework was built using two different approaches their performance and reliability was evaluated, each one has its own drawbacks and advantages over the other.

As for the first approach using Emotiv SDK and detection suits while users were successfully able to navigate the robot in four directions yet by analyzing the system efficiency and performance we have identified some drawbacks and difficulty for the system the major drawback was the training scheme used in the system as it's so difficult and not reliable. It needs the user to be in his full concentration also at every trial they user has to undergo the training, training in average can take up to 20++ minutes.

We noticed that different users were more skilled than other users even if they went through the same training process as it's mostly depends on the user ability to repeat or re-generate the same though signature hence the system will be able to classify it and perform the intended action in that case the system is not universal and a lot of users will face issue using the system.

As for the second approach the SSVEP it's challenging to design the system for one frequency and it's more challenging to combine different frequencies yet it's still more reliable than the first approach using the cognitive detection suits.

Using the second approach by designing the SSVEP system we could overcome of the draw backs above

1-we increased the reliability and the performance of the framework

2-The user no longer has to concentrate too much so we decrease the mental stress, the user only needs to focus on the flickering screen.



3-the framework design can be adopted easily in different kind of applications like wheel chair, or even a phone which operates with the same technique.

4-while the first approach limits the use of only 4 events, SSVEP can go far beyond that.

Using open vibe as processing platform is beneficial as it has vast number of detection and classifying algorithm also gives the advantages for the user to write his own algorithm it's easy and reliable.

## **5.2 Recommendations and Future work:**

We believe a higher performance can be achieved by studying more some of the main factors related to the experiments briefed below some of those factors.

1- Introducing an online artifacts removal and test the system performance.

2-Using different classifiers and filters can have an impact in the performance for eg. Fuzzy neural network classifier.

3-Testing the performance with different flickering frequencies.

4-Setup the framework with different EEG acquisition device and try to adapt and select the best electrodes positions around the visual cortex.

5- Testing the performance of the framework in different conditions and monitor the difference.

## REFERENCES

- [1] Graimann, B., Allison, B., & Pfurtscheller, G. (2010). Brain–computer interfaces: A gentle introduction. *Brain-Computer Interfaces*, 1-27.
- [2] Birbaumer, N., Brain-computer-interface research: coming of age. *Clinical neurophysiology: official journal of the International Federation of Clinical Neurophysiology*, 2006. 117(3): p. 479.
- [3] Izzetoglu, M., et al., Functional brain imaging using near-infrared technology. *IEEE Engineering in Medicine and Biology Magazine*, 2007. 26(4): p. 38.
- [4] Oum, K. V. (2010). *Brain computer interface gaming: development of concentration based game design for research environments* (Doctoral dissertation, Drexel University).
- [5] Lin, W., Burgess, R. W., Dominguez, B., Pfaff, S. L., Sanes, J. R., & Lee, K. F. (2001). Distinct roles of nerve and muscle in postsynaptic differentiation of the neuromuscular synapse. *Nature*, 410(6832), 1057-1064.
- [6] inaniLoquence (2010) *The Four Lobes of The Brain*. [online] Available at: <http://inaniLoquence.hubpages.com/hub/The4LobesofTheBrain>
- [7] JENSEN, E. (1998). *Teaching with the brain in mind*. Alexandria, Va, Association for Supervision and Curriculum Development. P.9-12
- [8] GRAIMANN, B., ALLISON, B., & PFURTSCHELLER, G. (2010). *Brain-computer interfaces revolutionizing human-computer interaction*. Heidelberg, Springer-Verlag. P.12
- [9] Brainconnection.positscience.com (2013) BrainConnection.com - *The Anatomy of Movement*. [online] Available at: <http://brainconnection.positscience.com/topics/?main=anat/motor-anat>.
- [10] E. Niedermeyer, F. H. Lopes da Silva. 1993. *Electroencephalography: Basic principles, clinical applications and related fields*, 3rd edition, Lippincott, Williams & Wilkins, Philadelphia

- [11] Teplan, M. (2002). Fundamentals of EEG measurement. *Measurement science review*, 2(2), 1-11.
- [12] Adelson, M., & Schapire, R. Emotiv Experimenter.
- [13] TONG, S., & THAKOR, N. V. (2009). *Quantitative EEG analysis methods and clinical applications*. Boston, Artech House. P.51
- [14] (Wickelgren, I., 2003, Neuroscience: Tapping the mind, *Science* 299(5606):496–499.
- [15] Wolpaw, J. R., Birbaumer, N., Heetderks, W. J., McFarland, D. J., Peckham, P H., Schalk, G., Donchin, E., Quatrano, L. A., Robinson, C. J., and Vaughan, T. M., 2000b, Brain-computer interface technology: A
- [16] Emotiv.com (2013) EEG Research | EEG Test | Home EEG Machine. [online] Available at: <http://emotiv.com/store/sdk/bci/research-edition-sdk/> [Accessed: 17 Feb 2013].
- [17] Stytsenko, K., Jablonskis, E., & Prahm, C. (2011, August). Evaluation of consumer EEG device emotiv epoc. In MEi: CogSci Conference 2011, Ljubljana.)
- [18] (Gonzalez-Sanchez, J., Chavez-Echeagaray, M.E., Atkinson, R. and Burleson, W. 2011, ABE: An Agent- Based Software Architecture for a Multimodal Emotion Recognition Framework. In Proc. of Ninth Working IEEE/IFIP Conference on Software Architecture, 2011, 187-193.)
- [19] Bernays, R., Mone, J., Yau, P., Murcia, M., Gonzalez-Sanchez, J., Chavez-Echeagaray, M. E., ... & Atkinson, R. (2012, October). Lost in the dark: emotion adaption. In Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology (pp. 79-80). ACM.
- [20] Emotiv. (2010). Comparison with other EEG units. Retrieved 2011, from Emotiv Community: <http://www.emotiv.com/forum/forum4/topic127/>
- [21] Duvinage, M., Castermans, T., Dutoit, T., Petieau, M., Hoellinger, T., De Saedeleer, C., ... & Cheron, G. (2012, February). A P300-based Quantitative Comparison between the Emotiv

Epoc Headset and a Medical EEG Device. In *Biomedical Engineering/765: Telehealth/766: Assistive Technologies*. ACTA Press.

[22] Brain-Computer-Interface. [online] Available at:

[http://www.fgcsic.es/lychnos/en\\_en/articles/Brain-Computer-Interface](http://www.fgcsic.es/lychnos/en_en/articles/Brain-Computer-Interface) [Accessed: 4 April 2013].

[23] Aloise F., Schettini F., Aricò P., Bianchi L., Riccio A., Mecella M., Babiloni F., Mattia D., Cincotti F. (2010). *Advanced Brain Computer Interface For Communication And Control*. Rome: ACM.

[24] VRPN: A Device-Independent, Network-Transparent VR Peripheral

System [http://www.cs.unc.edu/Research/vrpn/VRST\\_2001\\_conference/vrst\\_vrpn\\_paper\\_reprint.pdf](http://www.cs.unc.edu/Research/vrpn/VRST_2001_conference/vrst_vrpn_paper_reprint.pdf)

[25] [Openvibe.inria.fr](http://openvibe.inria.fr). OpenViBE | Software for Brain Computer Interfaces and Real Time Neurosciences

**In-text:** ([Openvibe.inria.fr](http://openvibe.inria.fr) 2013) **Bibliography:** [Openvibe.inria.fr](http://openvibe.inria.fr). 2013. *OpenViBE | Software for Brain Computer Interfaces and Real Time Neurosciences*. [online] Available at: <http://openvibe.inria.fr/> [Accessed: 11 Aug 2013].

## APPENDICES

### APPENDIX A: Coding for the interface program:

```
namespace LegoDriver
{
    using System;
    using AForge.Robotics.Lego;

    // Lego RCX car driver class
    public class NXTDriver : ILegoDriver
    {
        private NXTBrick nxt = new NXTBrick( );
        private string serialPort;
        // maximum power of NXT motors
        private int maxMotorPower = 100;

        // last power for each motor (required to avoid redundant calls to NXT API)
        private int lastLeftMotorPower = 0;
        private int lastRightMotorPower = 0;

        // Constructor
        public NXTDriver( string serialPort )
        {
            this.serialPort = serialPort;
        }

        // Connect to Lego NXT brick
        public bool Connect( )
        {
            if ( !nxt.Connect( serialPort ) )
            {
                // failed to connect
                return false;
            }

            // play something on connection
            nxt.PlayTone( 300, 300 );

            return true;
        }

        // Disconnect from Lego NXT brick
        public void Disconnect( )
        {
            nxt.PlayTone( 150, 300 );
            nxt.Disconnect( );
        }

        // Set power of A and C motor of Lego NXT brick
        public void SetMotorsPower( float leftMotor, float rightMotor )
        {
            int leftMotorPower = (int) ( leftMotor * maxMotorPower );
            int rightMotorPower = (int) ( rightMotor * maxMotorPower );
        }
    }
}
```

```

if ( leftMotorPower != 0 )
{
    if ( leftMotorPower > 0 )
        leftMotorPower = 50 + (int) ( 0.5 * leftMotorPower );
    else
        leftMotorPower = -50 + (int) ( 0.5 * leftMotorPower );
}
if ( rightMotorPower != 0 )
{
    if ( rightMotorPower > 0 )
        rightMotorPower = 50 + (int) ( 0.5 * rightMotorPower );
    else
        rightMotorPower = -50 + (int) ( 0.5 * rightMotorPower );
}

// System.Diagnostics.Debug.WriteLine( "l: " + leftMotorPower + ", r: " + rightMotorPower );

// return if there are no changes
if ( ( leftMotorPower == lastLeftMotorPower ) && ( rightMotorPower == lastRightMotorPower ) )
    return;

// check if we need to stop
if ( ( leftMotorPower == 0 ) && ( rightMotorPower == 0 ) )
{
    NXTBrick.MotorState motorsState = new NXTBrick.MotorState( );

    motorsState.Mode = NXTBrick.MotorMode.None;
    motorsState.RunState = NXTBrick.MotorRunState.Idle;

    nxt.SetMotorState( NXTBrick.Motor.All, motorsState );
}
else
{
    NXTBrick.MotorState leftMotorState = new NXTBrick.MotorState( );
    NXTBrick.MotorState rightMotorState = new NXTBrick.MotorState( );

    // left motor settings
    leftMotorState.Mode = NXTBrick.MotorMode.On;
    leftMotorState.RunState = NXTBrick.MotorRunState.Running;
    leftMotorState.TachoLimit = 0;
    leftMotorState.Power = leftMotorPower;
    leftMotorState.TurnRatio = 50;
    rightMotorState.Mode = NXTBrick.MotorMode.On;
    rightMotorState.RunState = NXTBrick.MotorRunState.Running;
    rightMotorState.TachoLimit = 0;
    rightMotorState.Power = rightMotorPower;
    rightMotorState.TurnRatio = 50;

    nxt.SetMotorState( NXTBrick.Motor.A, leftMotorState );
    nxt.SetMotorState( NXTBrick.Motor.C, rightMotorState );
}

```

```

namespace LegoDriver
{
    using System;
    using AForge.Robotics.Lego;

    // Lego RCX car driver class
    public class RCXDriver : ILegoDriver
    {
        private RCXBrick rcx = new RCXBrick( );
        // maximum power of RCX motors
        private int maxMotorPower = 7;

        // last power for each motor (required to avoid redundant calls to RCX API)
        private int lastLeftMotorPower = 0;
        private int lastRightMotorPower = 0;

        // Constructor
        public RCXDriver( )
        {
        }

        // Connect to Lego RCX brick
        public bool Connect( )
        {
            if ( !rcx.Connect( ) )
            {
                // failed to connect
                return false;
            }

            // play something on connection
            rcx.PlaySound( RCXBrick.SoundType.BeepBeep );

            return true;
        }

        // Disconnect from Lego RCX brick
        public void Disconnect( )
        {
            rcx.PlaySound( RCXBrick.SoundType.Blip );
            rcx.Disconnect( );
        }

        // Set power of A and C motor of Lego RCX brick
        public void SetMotorsPower( float leftMotor, float rightMotor )
        {
            int leftMotorPower = (int) ( leftMotor * maxMotorPower );
            int rightMotorPower = (int) ( rightMotor * maxMotorPower );

            // System.Diagnostics.Debug.WriteLine( "l: " + leftMotorPower + ", r: " + rightMotorPower );
        }
    }
}

```

```

|
// return if there are no changes
if ( ( leftMotorPower == lastLeftMotorPower ) && ( rightMotorPower == lastRightMotorPower ) )
    return;

// check if we need to stop
if ( ( leftMotorPower == 0 ) && ( rightMotorPower == 0 ) )
{
    rcx.SetMotorOn( RCXBrick.Motor.AC, false );
}
else
{
    int motorsToOn = 0;
    int motorsToOff = 0;
    int motorsToMoveForward = 0;
    int motorsToMoveBackward = 0;

    // update left motor
    if ( leftMotorPower != lastLeftMotorPower )
    {
        if ( leftMotorPower == 0 )
        {
            motorsToOff |= (int) RCXBrick.Motor.A;
        }
        else
        {
            motorsToOn |= (int) RCXBrick.Motor.A;

            // check if direction should be changed
            if ( leftMotorPower * lastLeftMotorPower <= 0 )
            {
                if ( leftMotorPower > 0 )
                {
                    motorsToMoveForward |= (int) RCXBrick.Motor.A;
                }
                else
                {
                    motorsToMoveBackward |= (int) RCXBrick.Motor.A;
                }
            }

            rcx.SetMotorPower( RCXBrick.Motor.A, (byte) Math.Abs( leftMotorPower ) );
        }
    }

    // update right motor
    if ( rightMotorPower != lastRightMotorPower )
    {
        if ( rightMotorPower == 0 )
        {
            motorsToOff |= (int) RCXBrick.Motor.C;
        }
        else
        {
            motorsToOn |= (int) RCXBrick.Motor.C;

            // check if direction should be changed
            if ( rightMotorPower * lastRightMotorPower <= 0 )
            {
                if ( rightMotorPower > 0 )
                {
                    motorsToMoveForward |= (int) RCXBrick.Motor.C;
                }
                else
                {
                    motorsToMoveBackward |= (int) RCXBrick.Motor.C;
                }
            }

            rcx.SetMotorPower( RCXBrick.Motor.C, (byte) Math.Abs( rightMotorPower ) );
        }
    }
}
}

```



```

// update right motor
if ( rightMotorPower != lastRightMotorPower )
{
    if ( rightMotorPower == 0 )
    {
        motorsToOff |= (int) RCXBrick.Motor.C;
    }
    else
    {
        motorsToOn |= (int) RCXBrick.Motor.C;

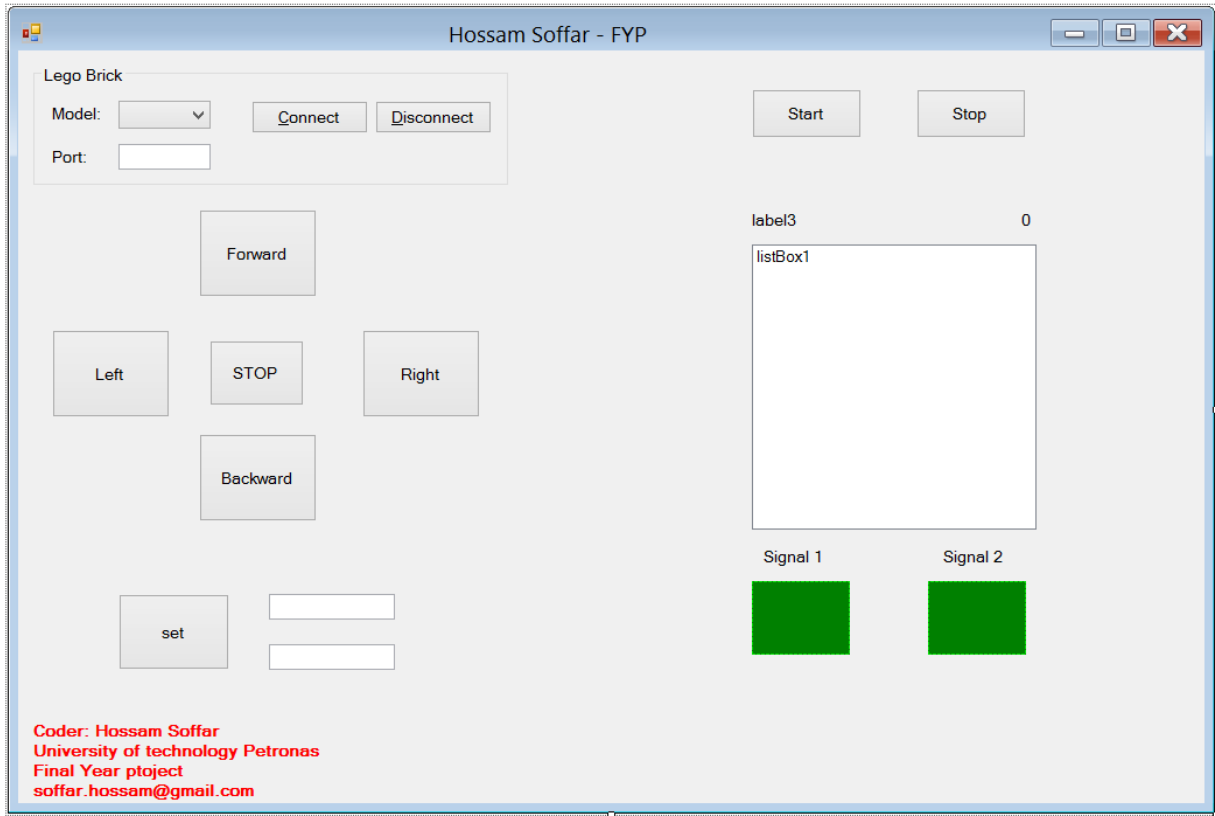
        // check if direction should be changed
        if ( rightMotorPower * lastRightMotorPower <= 0 )
        {
            if ( rightMotorPower > 0 )
            {
                motorsToMoveForward |= (int) RCXBrick.Motor.C;
            }
            else
            {
                motorsToMoveBackward |= (int) RCXBrick.Motor.C;
            }
        }

        rcx.SetMotorPower( RCXBrick.Motor.C, (byte) Math.Abs( rightMotorPower ) );
    }
}

if ( motorsToMoveForward != 0 )
{
    rcx.SetMotorDirection( (RCXBrick.Motor) motorsToMoveForward, true );
}
if ( motorsToMoveBackward != 0 )
{
    rcx.SetMotorDirection( (RCXBrick.Motor) motorsToMoveBackward, false );
}
if ( motorsToOff != 0 )
{
    rcx.SetMotorOn( (RCXBrick.Motor) motorsToOff, false );
}
if ( motorsToOn != 0 )
{
    rcx.SetMotorOn( (RCXBrick.Motor) motorsToOn, true );
}
}

// remember current motors' power to avoid setting the same next time
lastLeftMotorPower = leftMotorPower;
lastRightMotorPower = rightMotorPower;
}
}
}

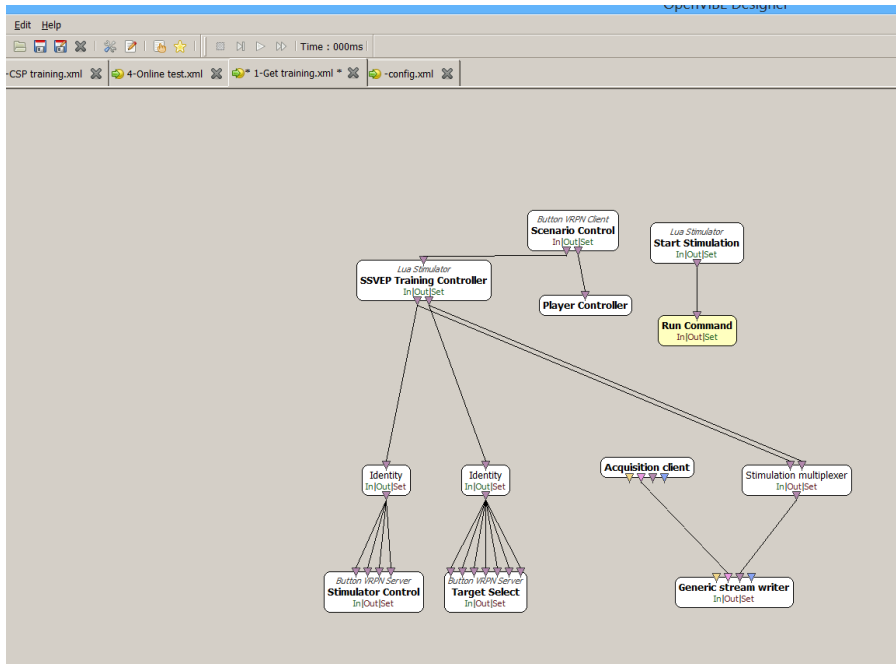
```



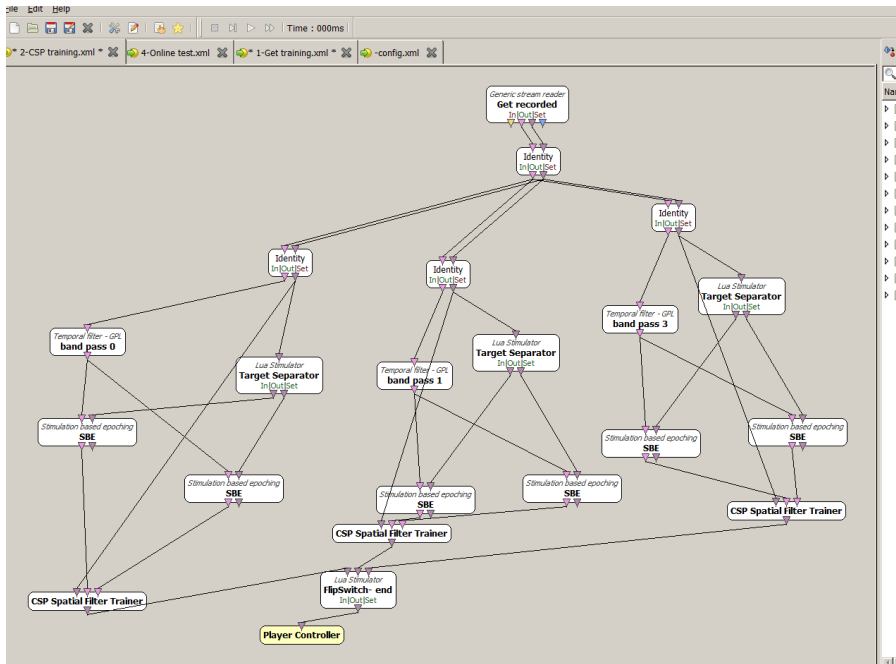
## INTERFACE PROGRAM

# APPENDIX B: Open vibe scenarios for processing

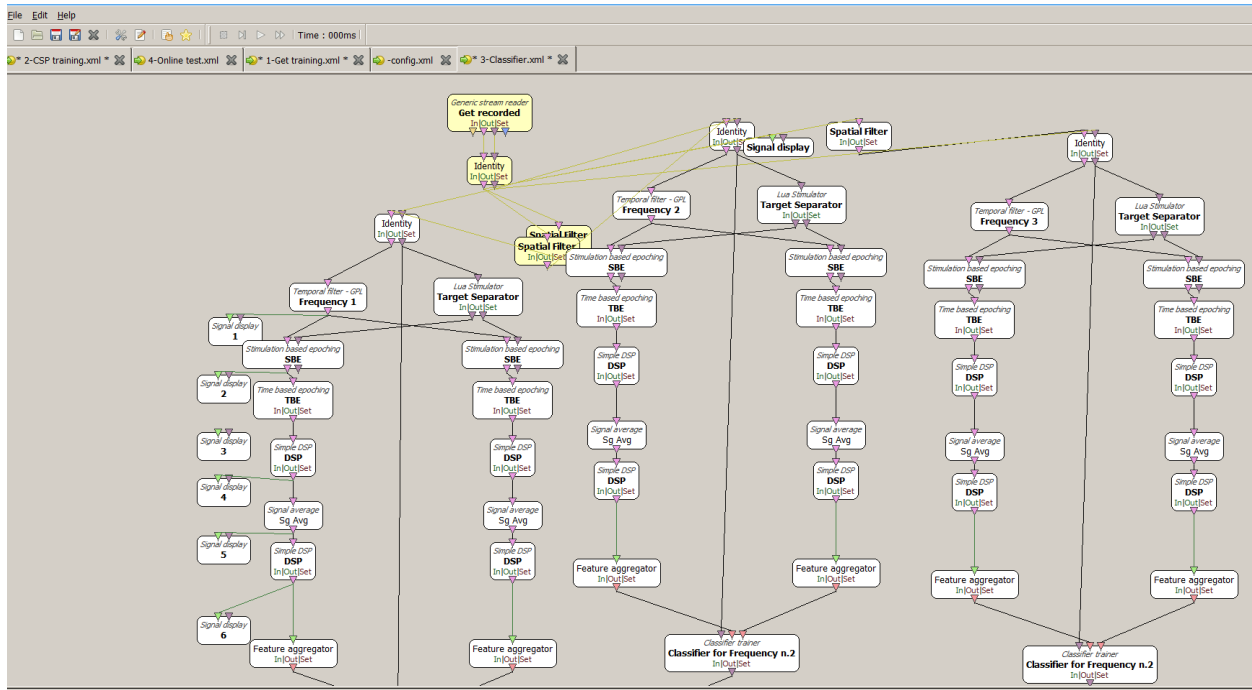
## 1-EEG recording



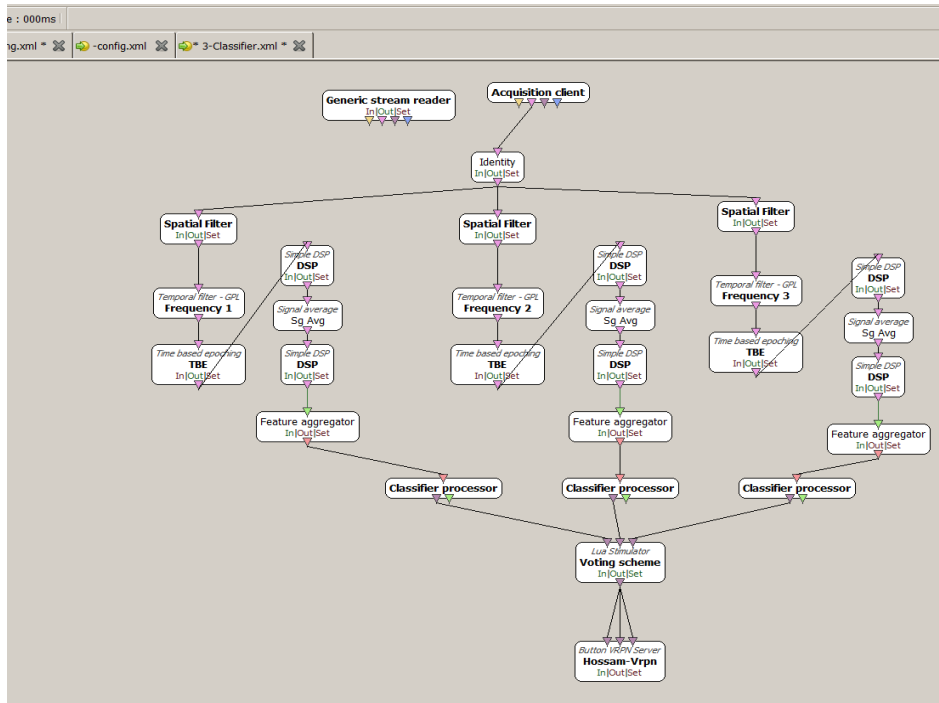
## 2-CSP



### 3-Generating Classifiers



## 4-Online processing and testing



## 5-stimulus display

