HAL
open science

Real-time triggering of a Functional Electrical Stimulation device using Electroencephalography - a hardware simulation study

Gorish Aggarwal

To cite this version:

Gorish Aggarwal. Real-time triggering of a Functional Electrical Stimulation device using Electroencephalography - a hardware simulation study. [Research Report] Lirmm, University of Montpellier; Inria. 2014. hal-01064082

HAL Id: hal-01064082

https://inria.hal.science/hal-01064082v1

Submitted on 15 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Real-time triggering of a Functional Electrical Stimulation device using Electroencephalography - a hardware simulation study

Gorish Aggarwal

Abstract — A Brain Computer Interface (BCI) is a direct communication link between the brain and an external device. The EEG signal captured through this device can be used to stimulate the motor neurons through a Functional Electrical Stimulator, thus allowing the brain to control the muscles externally. This requires the processing of the EEG signal, to extract features corresponding to the task imaged by the brain.

This report gives the description of the internship project undertaken to develop a mechanism such as above, which can collect EEG data, process the acquired data, extract relevant information and use that information to trigger an FES device for stimulating motor muscles. The developed system can be used to develop a low cost, clinically valid home-based tool to monitor post-stroke neuro-rehabilitation device for use both in rural and urban setting.

Index Terms— BCI, EEG, Emotiv EPOC, Feature Extraction, ERD, OpenViBe, Psychophysics Toolbox, FES, Phenix Liberty+ Console

In team :

I. Introduction

The project aimed at the development of a low-cost cyber physical system by integrating bio-signal sensors, e.g. electroencephalogram (EEG), electromyogram (EMG) and electrooculogram (EOG), for remote monitoring. The device developed during the internship period is able to record data from an EEG capturing device such as Emotiv EPOC, process the captured data online and provide a feedback to the patient regarding his/her mental state or the motor imagery task imagined by him. Following were the main objectives achieved during the internship:

1. Development of a bio signal data-acquisition system using a brain computer interface device (Emotiv EPOC) and open-source electronics prototyping platform, such as Matlab.
2. Processing of the acquired data offline to extract the required EEG features and improvement of the data for better detection of ERD.
3. Real time collection and processing of the EEG data to detect ERD and provide biofeedback simultaneously.
4. Triggering the FES device Phenix [1] based on the ERD detected to stimulate motor muscles.
5.

The internship was done under the supervision of Prof. David Guiraud, Directeur de Recherche Inria, HDR, DEMAR and Dr. Anirban Dutta, starting research scientist at Team DEMAR, LIRMM laboratories, INRIA-UM2. The hardware including the laptop computer used in this project was provided by Dr. Mitsuhiro Hayashibe, research scientist at Team DEMAR, LIRMM laboratories, INRIA-UM2. Furthermore, Dr. Mitsuhiro Hayashibe demoed and helped understanding his Matlab interface for triggering the Vivaltis FES system. The requirement of such a post neuro-rehabilitation device and the benefits of the tasks undertaken and accomplished during the internship are discussed later in this paper.

II. About the DEMAR Team

DEMAR [3] or the DEambulation et Mouvement ARtificiel Team is a research team interested in studying various aspects of Functional Electrical Stimulation (FES) such as implanted stimulators, sensors, biomechanics, closed loop controlled of stimulated muscles, and movement synthesis and analysis as well as patient interface. This promotes technology transfer so as to provide patients with an industrial medical device as an efficient solution to their neural problems. The DEMAR Research aims at the following tasks:

- Modeling and identification of the human sensory-motor system.
- Interfacing artificial and natural parts through neuro-prosthetic devices: both stimulation and recording.

The DEMAR project is led by Dr. David GUIRAUD. It is hosted at LIRMM [6]. The following institutions are mainly involved in this project:

A. INRIA

INRIA [4], the French National Institute for Research in Computer Science and Control, operating under the dual authority of the French Ministry of National Education, Advanced Instruction and Research and the Ministry of Economy, Finance and Industry, is a public institution established in 1967, dedicated to fundamental and applied research in Information and Communication Science and Technology (ICST). The Institute plays a major role in technology transfer by fostering training through research, diffusion of scientific and technical information, development, as well as providing expert advice and participating in international programs.

B. University of Montpellier 2

The UM2 [5] is a Research University Intensive whose training and research activities cover all scientific and technological fields: fundamental physics, biology, applied biology and ecology, chemistry, engineering, earth sciences, and mathematics, management, science education.

C. LIRMM

Montpellier Laboratory of Informatics, Robotics and Microelectronics (LIRMM) [6] is a research laboratory jointly funded by INRIA, CNRS [7] and the Universities of Montpellier 2 and 1. LIRMM research activities revolve around information and communication sciences and technologies.

III. Motivation

Stroke is defined as an episode of neurological dysfunction caused by focal cerebral, spinal, or retinal infarction. Stroke occurs when part of the brain is suddenly deprived of blood. There are mainly two types of stroke:

- Ischemic stroke:

The interruption of blood flow is caused due to clot (coagulated blood) that blocks an artery to the brain. In 80% of cases, strokes are ischemic stroke.

- Hemorrhagic stroke:

They are uncommon. It is the stoppage of flow of blood due to the rupture of an artery in the brain.

Stroke is a global health problem and fourth leading cause of disability worldwide. In France, stroke is the leading cause of acquired disability in adults. It affects more than 150000 people in France and is also the leading cause of paralysis [8]. The incidences of stroke are also increasing in developing nations such as India due to demographic transition and rapid shift in the socioeconomic milieu. Indian studies have estimated the increase in number of stroke cases from 27-34/100,000 in the 35-44 age groups to 822-1116/100,000 in the 75+ age group [9]. Moreover, in India, there is a high prevalence of stroke in younger people (18-32% of all stroke cases), when compared with high-income countries, which creates an increased burden on the quality of life and on the society [10].

Stroke constitutes a huge trouble for the health care system and the overall economy primarily due to the long-term nursing care even in a developed nation such as France [8]. Unfortunately, India is also in dire need for adequately trained medical personnel for neuro-rehabilitation. This scarcity may strain India's health care system in the coming years due to the treatment and long-term care for an increasing population of neurologically disabled people.

To help the stroke patients overcome their deficiencies, we have to create a simple to use, clinically valid home-based tool to monitor post-stroke neuro-rehabilitation, that can be used both in rural and urban setting. Recently, it has become possible to customize, monitor, and support neuro-rehabilitation at remote sites [11]. This will allow health care providers to continue to monitor patients' progress, identify areas in need of improvement, and ultimately improve function and minimize disability and costs. Thus, such a device will bring about a paradigm shift in the field of medicine and healthcare.

IV. Brain Computer Interface System

Stroke leads to motor disability, which leaves stroke survivors with paralysis. Therefore a BCI may provide brain with a channel of communication that can be used for controlling prosthetic devices. The first ever demonstration of the EEG (Electroencephalography) technique (a BCI modality) was given by Hans Berger in 1929 [12]. The doctors and scientists have since then used it as a basic tool for studying the brain. Researches are being carried out to use the EEG technique to decode people's intentions. This will allow people to directly control devices using their brain activity [13].

A Brain-Computer Interface (BCI) is a direct communication channel between the brain and an external device. [14] In a BCI, the messages or commands sent by the individual to the external world do not pass through the brain's normal pathways of peripheral nerves and muscles. Rather, the EEG signals are recorded from the scalp or the surface of the brain. The main application of a BCI is to help an individual with severe motor disabilities to effectively control devices such as computers, speech synthesizers, assistive appliances and neural prostheses. This will improve quality of life for people with severe disabilities such as stroke [15].

Our brains are filled with neurons, individual nerve cells interconnected to each other by dendrites and axons to form an enormous neural network. They communicate together through these dendrites and axons using small electric impulses of the order of microvolts [16].

These variations can be represented in the form of electrical maps. Thus, a mental activity can be seen as a sequential continuation of brain electrical states. Based on recordings of the brain's electrical activity, we would like to reflect upon the functional states of the brain and identify those states, which can possibly represent a mental task and be able to detect the same state later, when a similar mental activity is performed.

Before moving to operation of BCI and its application let us first discuss the three types of BCI.

A. Types of BCI

There are three major types of BCI, which are differentiated on the basis on their placement in the body and their functions.

1) Invasive BCI

It is a brain signal reading process, in which the device is directly implanted to the inside of grey matter of brain during neurosurgery. Invasive BCIs are used for repairing damaged sight and providing improved functionality to the paralyzed people. But, Invasive BCIs are prone to building up of scar tissue, which leads to a weakening signal as time goes on and as the body reacts to a foreign object in the brain. Invasive BCI are mainly used in animal models [17].

2) *Partially Invasive BCI*

It is another brain signal reading process in which the device is implanted to the inside of the skull but outside the grey matter. They produce better resolution signal than non-invasive BCIs at points where the bone tissue of the cranium deflects and deforms signals reaching the scalp. They are at a lower risk of forming scar tissue in the brain than fully invasive BCIs. Electrocorticography (ECoG) is the example of partially invasive BCI [18].

3) Non - Invasive BCI

It is the most useful neuron signal imaging method. The device is implanted to the outside of the skull, on the scalp. Although they are easy to wear, non-invasive implants produce poor signal resolution because the skull dampens signals, dispersing and blurring the electromagnetic waves created by the neurons. Still due to their much easier implementation, there are most widely used for research and medical purposes. Of the various Non-Invasive BCIs, Electroencephalography (EEG) has been most widely studied and most of the researches these days are based on EEG. Besides the EEG, the other non-invasive techniques include Magneto encephalography (MEG), Magnetic resonance imaging (MRI) and functional magnetic resonance imaging

(fMRI) [19].

B. Operation of BCI

The project is based on the motor imagery task performed by a subject. Its operation is based on a device for measuring the cerebral activity when the individual performs a motor imagery following a stimulus.

Once the signals are recorded and amplified by the BCI, the processing of the signals is done. First, a preprocessing is done to improve the quality of the signal, and then the various features are extracted to keep the most relevant information in the signal, which corresponds to the imagery state. Finally, each of the states is classified to identify a particular mental task. Finally, when the task of motor imagery is identified, the system may then give a command to control a machine or send a visual or audible feedback for the subject.

1) Measuring Brain Activity (Technical Non-invasive)

There are several criteria to be taken care of when measuring cerebral activity:

a. The measurement technique: way to measure the cerebral activity: invasive / non- invasive.
b. Spatial resolution: the ability of the system to discriminate between two or more spatially close mental tasks.
c. The temporal resolution: the frequency of the temporal coverage of system acquisition. This criterion is very important as it decides the kind of waves our BCI can capture.

The most used technique to capture cerebral activity is EEG.

- *Electroencephalography (EEG)*

Electroencephalography (EEG) is the recording of electrical activity along the scalp produced by the firing of neurons within the brain [20]. In clinical contexts, EEG refers to the recording of the brain's spontaneous electrical activity over a short period of time, usually 20–40 minutes. EEG device is attached to the scalp through electrodes. These electrodes are connected to a machine, which records the electrical impulses. The signals are then processed to finally get the mental state of the brain during the recording period. Each area of the brain produces a different brain wave strip according to the mental state, the kind of tasks he was imagining and the general condition of the patient.

This method presents both advantages and disadvantages, we can mention:

a. Its high temporal resolution: Rapid electrical phenomena or relatively high frequency.
b. Easy set up and noninvasive: Possibility to study the activity of valid and invalid subjects without any surgical intervention several environments.
c. Reduced cost: Cost of EEG is relatively less compared to other techniques existing measure of cerebral activity.

However, the EEG is limited in its spatial resolution and its range of frequency. The EEG signal consists of different frequency bands defined by their wave to characterize and identify psychological states Neuroscience: [21]

a. Delta Waves: Delta is the frequency range up to 4 Hz. It tends to be the highest in amplitude and the slowest waves. It is seen normally in adults in slow wave sleep.
b. Theta Waves: Theta is the frequency range from 4 Hz to 7 Hz. Theta is seen normally in young children. It may be seen in drowsiness or arousal in older children and adults; it can also be seen in meditation.
c. Alpha waves: Alpha is the frequency range from 8 Hz to 12 Hz. Named initially by Hans Berger, the activity of these waves are seen in the posterior regions of the head on both sides, being higher in amplitude on the dominant side. It is brought out by closing the eyes and by relaxation.
d. Mu rhythm: Mu rhythm is alpha-range activity that is seen over the sensorimotor cortex. It characteristically attenuates with movement of the contralateral arm (or mental imagery of movement of the contralateral arm).
e. Beta Waves: Beta is the frequency range from 12 Hz to about 30 Hz. It is seen usually on both sides in symmetrical distribution and is most evident frontally. Beta activity is closely linked to motor behavior and is generally attenuated during active movements [15]. We are going to mainly capture the beta waves and test our EEG systems on them.
f. Gamma Waves: Gamma is the frequency range approximately 30–100 Hz. Gamma rhythms are thought to represent binding of different populations of neurons together into a network for the purpose of carrying out a certain cognitive or motor function.

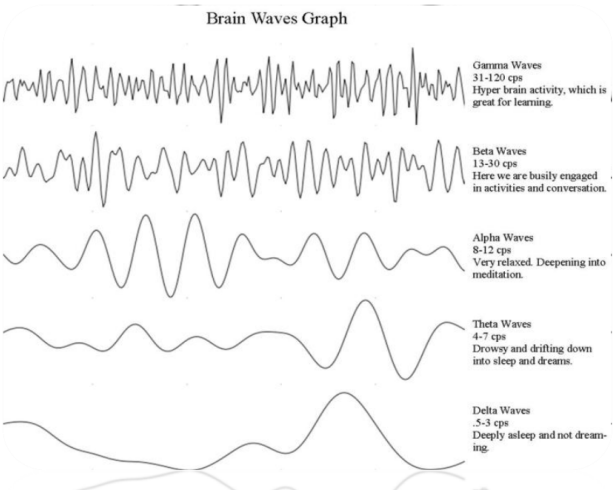

Figure 1: Brain Waves of an EEG signal.

The normal Electroencephalography (EEG) varies by age. The EEG in childhood generally has slower frequency oscillations than the adult EEG.

2) Feature Extraction

Extraction of features is a fundamental step in the process of a BCI, prior to the classification. The analysis of this type of signals is very complicated due to the nature of the signal and very important in the reasoning scientific design, which leads to the BCI system.

The first step involved in BCI systems is the data collection and filtering. The filters are designed such that there is no change or distortion to the signals. Interfering low frequency components such as breathing is removed by passing the signal through a high pass filter with a cut-off frequency of less than 0.5 Hz. Whereas, high-frequency noise is reduced by passing the signal through a low pass filter with a cut - off frequency of approximately 40–70 Hz [22].

The original EEG signal is a time domain signal. The signal energy distribution is scattered and its features are buried away in the noise. In order to extract the signal features, the EEG is analyzed to give a measure of the signal energy as a function of time or/and frequency. Based on previous studies, feature extraction in frequency domain is one of the best ways to recognize the mental tasks based on EEG signals [22].

In this project, we have used the Fast Fourier Transform (FFT) for finding the signal energy. The discrete FFT is applied to the signal and the energy in the required band is found.

EEG signal is non-stationary that means its spectrum changes with time. Such a signal can be approximated as piecewise stationary i.e. a sequence of independent stationary signal segments [23]. The Fourier functions do not adequately represent non-stationary signals. Therefore, appropriate windows have been applied to the Fourier functions, which provide short time Fourier transform (STFT). The discrete STFT equation is:

$$X_{STFT}[m,n] = \sum_{k=0}^{L-1} x[k]w[k-m]e^{-\frac{j2\pi nk}{L}}$$

Where $x[k]$ denotes a signal and $w[k]$ denotes an L-point window function. The STFT is the Fourier transform of the product $x[k]w[k-m]$ [22]. To compute the time - frequency representation of the signal, the whole signal is divided into small sequential or overlapping data frames/windows and fast Fourier transform (FFT) applied to each one. The output of the successive STFTs provides a time frequency representation of the signal. This is possible by reducing the complete signal into small packets and applying FFT to window. The STFT is applied to 250 millisecond EEG signal each divided into a 32 point segments. Each segment is multiplied by 32 point Hamming window, and then the FFT algorithm is applied to each segment. In these windows, the beta bands with frequency 12-18 Hz are used and their values are normalized. This allows us to get a power value in each of the 250 milliseconds window, thus providing us with a quantitative analysis of the type of signal during that time scale through ERS/ERD. Fig. 3 shows the general idea of the above process.

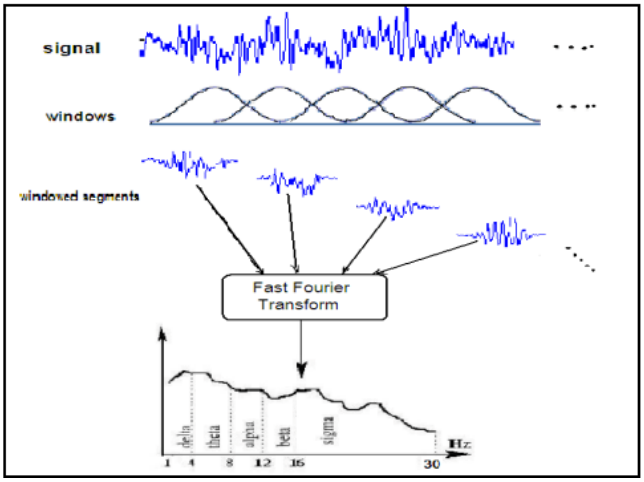

Figure 2: Spectral Transformations

3) The electrophysiological signals

The BCI systems on electro physiological signals are classes in two main groups: endogenous and exogenous system [26].

a. Exogenous Systems:

Exogenous BCI systems depend on the electro physical activity evoked by external stimuli and do not require intensive training. The main characteristics of exogenous systems based on P300 evoked potentials or steady state visual evoked potentials (SSVEP) are described below.

- *Evoked Potential P300*

The P300 is an evoked potential produced by the nervous system, particularly in the parietal cortex of the brain in response to visual stimulation (or hearing). This potential corresponds to an increase potential 300 millisecond after the occurrence of a stimulus [24].

Using this response, researchers have created a BCI with which the subject can communicate by choosing letters on a screen. The average speed is very slow about 5 letters per minute.

- *Steady state visually evoked potentials (SSVEP)*

SSVEP is an electrical potential difference, which can be derived from the scalp after a visual stimulus. They are called "steady state" VEPs because the individual responses overlap and result a quasi-sinusoid oscillation with the same frequency as the stimulus [25]. The goal is to detect this frequency reliably with high accuracy and furthermore to detect when the frequency is not present, thus when the person does not look at the stimulus.

The most common application of Exogenous Systems is the control of robots. Limitations of Exogenous system include need of feedback and very low feasibility for humans use.

b. Endogenous Systems:

Endogenous BCI systems depend on the user's ability to control their electrophysiological activity, such as the EEG amplitude in a specific frequency band on a specific area of the cerebral cortex. Endogenous BCI systems include those based on motor images (sensorimotor rhythms) or slow cortical potentials (SCPs). Both require a period of intensive training.

i. Slow cortical potential (SCP):

Slow cortical potentials are defined as the positive or negative polarizations of the electroencephalogram (EEG) that last from 300ms to several seconds with magnitudes up to 50 µV [27].

SCP amplitudes are regulated within tight limits by a negative feedback-loop consisting of a cortical-basal ganglia threshold regulation system that maintains cortical activation within acceptable medium limits.

Surface negative slow cortical potentials (SCPs) are typically associated with movement and other functions that imply cortex activity. Surface positive SCPs appear during behavioral inhibition and motivational inertia [28].

ii. Even related desynchronization (ERD)/ synchronization (ERS)

The ERD/ERS are different responses of neuronal structures in the brain. It is not phase-locked to the event. The ERD/ERS comes after a slow cortical potential and characterized by a high power spike (or drop) in a specific band of frequency [29]. Either the same or different locations on the scalp can display ERD and ERS simultaneously. Thus, we will be looking at the sudden increase or decrease of power in a certain window for the beta band (12-18Hz).

C. Applications of BCI

The applications of BCI are very broad and include neural prosthetics, robotic control, virtual world, speechless communication and even environmental control [30]. Initially, intended to be used for neuro-prosthetics applications that aimed at restoring damaged hearing, sight and movement. But, now this field is emerging as a major research area in robotics and

virtual technology. The Asimo robot uses BCI to be controlled by the thoughts of a nearby human [31]. Communication is possible by just thinking over the words in our minds and is being used to support people who are completely paralyzed or unable to speak. Virtual reality can provide a safe environment for training and tuning neural controlled interfaces to real-world devices, such as a power wheelchair [32].

V. Experimental Setup

A. BCI Device – Emotiv EPOC

Emotiv EPOC developed by Emotiv Systems is a high resolution, neuro-signal acquisition and processing, wireless headset to detect player thoughts, feelings and expressions.

The EPOC brain helmet has eighteen sockets and can hold sixteen nodes or saline sensor pads for accurate spatial resolution. The remaining two sockets usually hold rubber pads, known as the secondary reference sensors, which are located immediately below and behind the ears. The two primary reference sensors, which generally hold a normal sensor pad, are located immediately above and behind the ears and provide reference values for the other sensors. The sensors pads detect electrical activity on the surface of the brain to tune into electric signals produced by the brain in real time. The Emotiv EPOC headset also contains a 2-axis gyroscope to detect head movements for information for cursor and camera controls [33]. The Emotiv EPOC samples sequentially at a sampling rate of 128 samples per second. It also provides a high resolution of 14 bits with 1 least significant bit. This nearly equals 0.51 microvolts. Also, the bandwidth of the device is 0.2-45 Hz with notch filters at 50 and 60 Hz [34].

The Emotiv EPOC BCI device was preferred compared to other devices in the market such as OpenBCI, Muse or Emotiv EEG especially due to its high-resolution performance and a large number of electrodes at a low market price of 299$ which gives it an edge over the other devices. The device also provides a gyro sensor which can be used later on while using eye-tracking in the project to determine the exact direction in which the patients wants to move. The dynamic range of the device is also quite high. Also, the Emotiv EPOC comes with Emokey software, which can be used to create one's personal, set of brain controlled commands. Open-source Matlab toolboxes such as EEGLAB, Fieldtrip, and the Neurophysiological Biomarker Toolbox (NBT) can be used to process data from the Emotiv EPOC [34].

B. Acquisition System – OpenViBe

OpenViBe is a software platform dedicated to designing, testing and using brain-computer interfaces. It is used in the field of real-time neurosciences (that is, for real-time processing of brain signals). It can be used to acquire, filter, process, classify and visualize brain signals in real time.

The purpose of OpenViBe is to get data from the EEG device through the Acquisition Server and then send it to one or more clients. This client is usually, for now, the OpenViBe designer. The Acquisition Server and the clients (Designers) can be either on the same machine or different machines on the same network, or any combination of these [37]. OpenViBe Acquisition Server is a tool designed to communicate with hardware signal acquisition devices and forward acquired signals and other experiment information to other applications in a way compliant with the OpenViBe format specification. The acquisition server doesn't communicate directly with acquisition devices. Instead, it provides the user with a set of drivers to choose from, each one being dedicated to a given device model through which it connects to each acquisition device [38].

With the Emotiv EPOC driver, OpenViBe provides a host of properties. The main ones includes: Sample count per sent block which can vary the amount of time required to obtain a single window and an option to add 2 channels for gyroscope. It is necessary to set the samples count per sent block to the correct value to avoid missing any sample points. In our case, this value was set to be 32 which allow us to have a window size of 250millseconds due to the fixed sampling frequency of EPOC at 128 Hz. OpenViBe acquisition server also provides the options to control the amount of Drift Tolerance and Jitter Estimation Count for drift. The Acquisition Server monitors the behavior of the driver in order to ensure that it actually sends the number of samples it should base on the theoretical sampling frequency of the device. When the drift reaches a threshold, an automatic correction is done i.e. the Acquisition Server removes or duplicates some samples, and sends stimulation(s) to any acquisition client connected. This threshold value can set by the user and is known as Drift Tolerance and is measured in milliseconds. Jitter Estimation Count is the window in number of "blocks" used to compute the drift [38].

OpenViBe also provides us with an OpenViBe Designer tool, which is an authoring tool, dedicated to creating and executing OpenViBe scenarios. It is targeted at a broad range of users both programmers and non-programmers. The OpenViBe Designer relies on a graphical user interface to provide signal-processing tools in an intuitive way.

The OpenViBe consortium has developed various open source software for the design, testing and use of BCI devices. They can be used to acquire, filter, process, classify and visualize brain signals in real time. The high modularity of OpenViBe provides it an edge over other available EEG processing tools. The software modules are written in C++ that can be integrated easily and efficiently to design BCI applications. OpenViBe is a complete program for Brain-Computer Interface and neuroscience research. It was developed initially by the INRIA research Institute and is available immediately and freely [36].
OpenViBe is mainly used in fields such as medical (assistance to disabled people, real-time biofeedback, neuro-feedback and real-time diagnosis), multimedia (virtual reality, video games), robotics and all other application fields related to brain-computer

interfaces and real-time neurosciences. OpenViBe users do not necessarily require the knowledge of programming as the OpenViBe Designer application provides a GUI control to create and execute various scenarios [35].

The reason to prefer OpenViBe to other neuro-feedback software is due to the fact that it supports BCIs such as Emotiv EPOC, which is required by us in our project. It is also an open-source platform and hence, we can develop modules according to our requirements. It also features a high level of Modularity and reusability. Thus, OpenViBe is a reliable product by INRIA and hence, OpenViBe was chose for this project.

Matlab was used to acquire data from OpenViBe Acquisition Server. The OpenViBe designer tool was not used instead as the OpenViBe does not support the Vivaltis Stimulator [46] till date and finally, it is required to trigger that stimulator to provide human stimulus to the patient.

C. Test Benchmarking of the signal

It was done to ensure that the acquisition system developed could regenerate the initial signal correctly and without replicating or losing any data points. The initial signal was generated through the Ni-DAQ Device through another computer [39]. Few sinusoidal waves were generated with a sampling frequency of 12-18 Hz (Beta band). The NI-DAQ was connected to the electrodes of Emotiv EPOC. The wireless EPOC sampled the signal at 128 Hz and transmitted them to the USB Dongle. Thus, the data was collected in the computer and viewed on Matlab.

1. *Algorithm to acquire a single dataset:*

To record EEG data from the Emotiv EPOC headset, we need to connect the OpenViBe acquisition server containing the lab-streaming layer (to get data in Matlab) with the Emotiv USB Dongle. In the Matlab, we define a timer function that will work with a fixed refresh rate of 16 Hz. This refresh rate was chosen after paying considerable attention to the data received at different refresh rates. It was found that the Emotiv EPOC does not provide continuous data windows. Instead, it trims the collective bundles of data in some cases and at other times, the data windows were empty. For example, with a sample count per sent block of 32 and a refresh rate of 32 Hz, we got a few repeated datasets. This is due to the fact that only the last 32 windows were pulled from windows with an actual size of 64. Hence some data points got lost. Since the total data points in a large time interval remained the same, hence some data points were also repeated. This was observed experimentally. Hence, an optimum refresh rate of 16Hz was found which provided a data window of 32 points after every 3 empty data window. Thus, all the 32 non-empty windows were combined into a 3-D matrix to form a single dataset. The matrix stores data for 14 channels, 32 data points in 1 window and the number of windows for which the data was found.

2. *Code Snippet:*

```
assignin('base', buffername, create_streambuffer(inlet.info()));
th = timer('Period', 1.0/refreshrate,'ExecutionMode','fixedRate','TimerFcn',@(timerhandle,
eventdata)on_timer,'StartDelay',0.2,'Tag',['lsl_' genvarname(streamname) '_timer']);
%The timer command defines a timer instance with the various object properties to define the timer. The
timer is discussed in detail in the report.
start(th);                                                          %The start command initiates the
timer instance th.
function on_timer
try                                                                 %Try block to catch any error that
might occur during the collection and storage of data.
[chunkdata,timestamps] = inlet.pull_chunk();                       %pull_chunk() gets the data from EmotiV
EPOC in chunkdata and also a timestamp on the data.
if i < range+1                                                     %Check to ensure whether the data is in
range.
if ~(isempty(chunkdata(1,:)))                                   %Since the data in this case is received once
in four timer values it is a check to ignore empty matrix.
B(:,:,i) = chunkdata;                                         %Chunkdata value is stored in matrix B at the
appropriate position.
i = i+1;                                                       %The variable i is incremented.
end
else
display('Data Collected');                                      %'Data Collected' is displayed on the screen to
indicate that data has been collected.
assignin('base',data,B);                                        %'Data has been assigned to a new variable
'data' in the workspace.
display('Data Stored');                                         %'Data Stored' is displayed on the screen to
indicate that data has been stored in workspace.
stop(th);                                                       %Command to stop the timer instance th.
delete(th);                                                     %Command to delete the timer instance th.
return;                                                         %Command to return the control to the
workspace.
end
catch e                                                             %Catch Block to display the error
message in case of error.
.
.
.
```

3. Results:

The initial signal to the NI-DAQ device was sent at 15 Hz and with amplitude of 20mV. This amplitude was high as compared to the normal brain signal but we were limited by the resolution of the NI-DAQ Device.

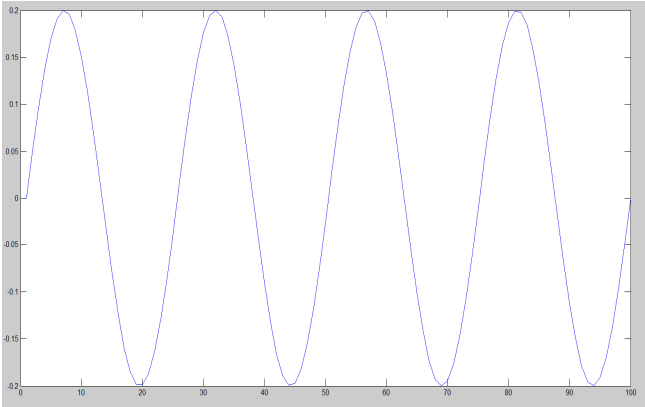

Figure 3: Original Signal provided to NI-DAQ Device

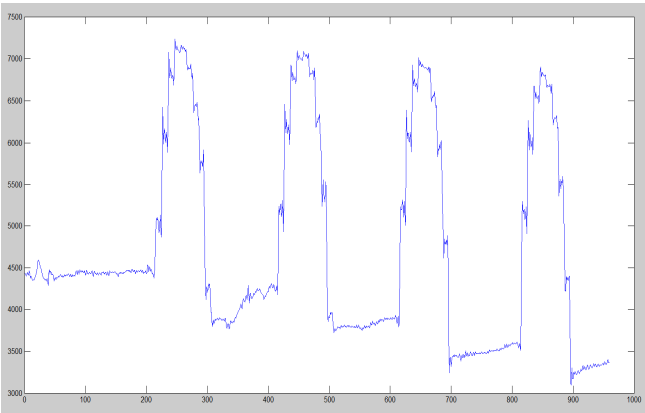

Figure 4: Original positive part signal received from EPOC after sampling at the 9th channel

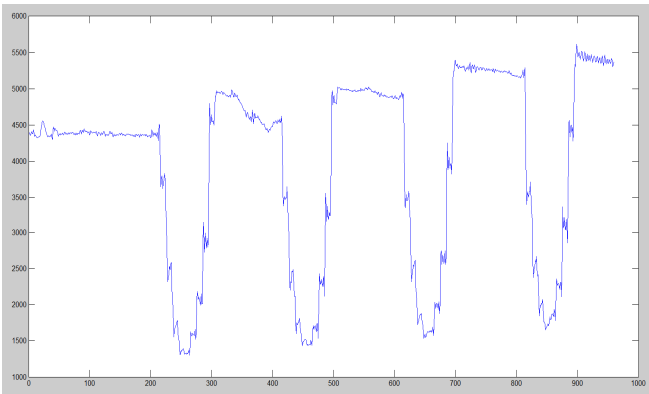

Figure 5: Original negative part signal received from EPOC after sampling at the 6th channel

As we can see, the developed system duplicated the signal quite well in the positive part of the sinusoid without repeating or missing up on any data points. This was the first and foremost requirement of a EEG capturing system since to detect the ERD, all the data points must be distinctly available to us. If any of the data set is missed or duplicated, we may end up missing the required ERD or detecting the same ERD twice. The noise in the sinusoid is the transmission noise and its amplitude is very low in comparison to the amplitude of the sinusoid. This noise is the due to:

- Noise produced by the NI-DAQ while generating the sinusoid
- Noise introduced at the point of contact of saline electrodes of Emotiv EPOC and the ends of NI-DAQ.
- Noise captured by the EPOC due to external disturbances and crossovers.

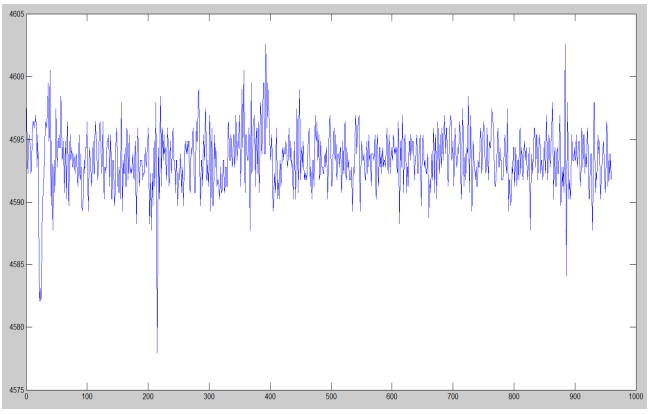

Figure 6: Noise received from EPOC after sampling at the 14th channel

The phenomenon of crosstalk was also observed between various channels especially those, which had electrodes near to the 2 to which the NI-DAQ was actually connected. This is due to the fact that the Analog to Digital converter used in the device cannot sample at the very fast rate. Therefore, when signal of high amplitude are provided to EPOC, the capacitors remain a bit charged after each cycle. This leads to crosstalk between the channels. The crosstalk between different channels is low and goes on decreasing as the distance from the channel receiving the signal increases as can be observed from the Figure 8. The maximum crosstalk is observed to be about 3% of the total range of signal.

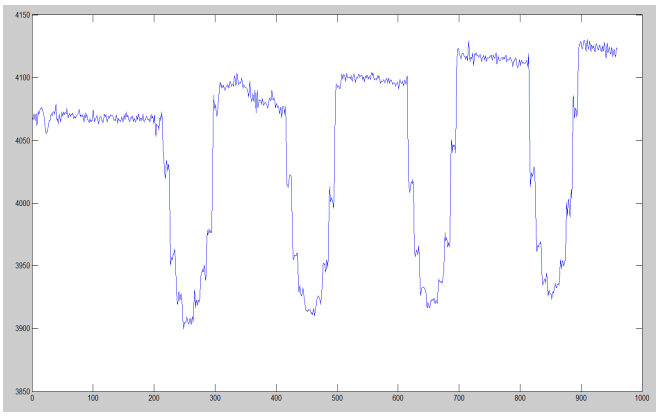

Figure 7: Crosstalk Signal received from EPOC after sampling at the 7th channel

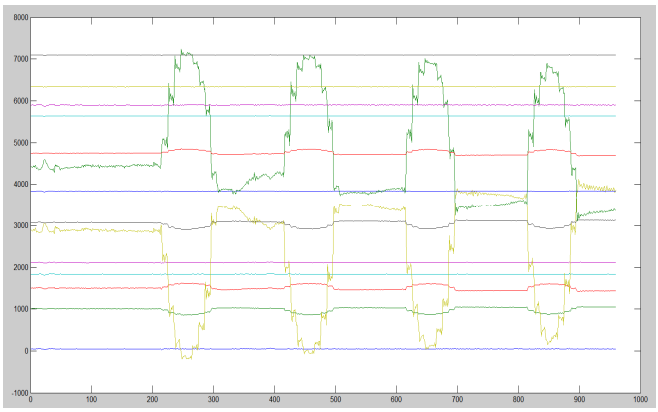

Figure 8: Complete Signal received from EPOC from all channels and after shifting to ease understanding

D. Creating a moving window

To enhance the detection of ERD and also to streamline the available data, another algorithm has been developed a way to overlap various the adjoining data sets and create a different set of matrices with same or different window size. This allows for more perfect and complete detection of ERD as even the data sets with peaks at end of the window can also be detected in the new datasets. The method allows for various amounts of overlap of 2 consecutive datasets in multiples of 25 and also allows us to vary the window size. The algorithm picks up the required number of data points from the last dataset and appends to it the

data points from the original dataset. Thus, this kind of a system allows for ERD to be detected in even those cases in which the spikes in power in the beta band comes at the extreme ends of the windows, thus improving accuracy and efficiency.

If the approach is used in online acquisition and processing of data, it can allow for a more discrete control in detection of ERD and trigger of FES as we will study later. This is due to the fact that there will a smoother transition of data points from one to another data set.

1. *Code Snippet:*

```
ndatapoints = datapoints/32;                                        %Normalized Data points to a ratio
of 32.
noverlap = overlap/25;                                              %Normalized Overlap Percentage to a
ratio of 25.
%The particular values of 32 and 25 for normalizing the no. of data points and overlap have been chosen to
keep the code simple and keep integral size.
overlapcount = ceil(datapoints*overlap/100);                        %No. of data points to be overlap
with the consecutive window.
windowcount = size(a,3);                                            %No. of windows in the original
dataset.
indatapts = size(a,2);                                              %No. of data points in a single
window in the original dataset.
channels = size(a,1);                                               %No. of channels in the data.
newwindowcount = ((windowcount*4/ndatapoints-noverlap)/(4-noverlap));   %No. of windows in the new moving
dataset.
B = zeros(channels,datapoints,newwindowcount);                      %Initialization of the matrix B
containing the new dataset with zeros to the appropriate size.
B(:,:,1) = reshape(a(1:channels,1:indatapts,1:ndatapoints),channels,indatapts*ndatapoints); %Obtaining the
first window of the new matrix B by reshaping the required no. of starting windows of original matrix a.
for i = 2 : newwindowcount                                          %Loop to obtain the rest of the
windows of B.
for j = 1 : overlapcount                                        %Loop to acquire the initial
overlapping data points from the previous window of the new datasets.
B(:, j, i) = B(:, j + datapoints - overlapcount, i-1);          %Copying the data points to the new window
from the previous window.
end
p = datapoints*(i-1)-overlapcount*(i-2)+1;                      %The index of current data point with
respect to the first data point of the entire dataset.
r = mod(p, windowcount);                                        %The index of current data pointwith
respect to the first data pointof the current window.
q = ceil(p/windowcount);                                        %The window no. of the original dataset
which contains the current datapoint.
if(r < 1)                                                       %Adjustments in the above two values to
r and q to account for Matlab's initial matrix index to be 1 and not 0.
q = q-1;
r = r+windowcount;
end
for j = overlapcount + 1 : datapoints                           %Loop to obtain the rest of the data
points from the qth window of the original dataset.
B(:,j,i) = a(:,r,q);                                            %Copying the datapoints to the new window
from the qth window at the appropriate index.
r = r+1;                                                        %Icrementing the index of the current
datapoint.
if(r > windowcount)                                             %Adjustments in the above two values to r
and q to account for the case of overshoot of r above the windowcount.
r = r-windowcount;
q = q+1;
end
end
end
```

VI. Offline Detection of ERD

ERD or Event Related De-synchronization as explained before is the short-lasting attenuation or blocking of rhythms ERD or Event Related De-synchronization as explained above is the short-lasting attenuation or blocking of rhythms within the beta band [29]. As the power level of EEG data goes below normal, we can deduce that an ERD has occurred.

EEG Data was initially stimulated using another computer and passed to the Emotiv EPOC through the NI-DAQ Device. The EEG data was collected for all the electrodes and processing was done on it to detect the fall in power level in a particular window. This allows us to detect the point of time at which the ERD occurred.

A. Methodology

The power in the beta band for the complete data was calculated using the band power function [40]. The band power function was run on each window of data to calculate the power. It takes as an input the data, sampling frequency and the range of

frequency for which band power is to be calculated. The band power function uses the periodogram to filter the data with a hamming window of the same length as that of x and compute the power spectral estimates in the beta band [40]. The hamming window has a magnitude and phase response as follows:

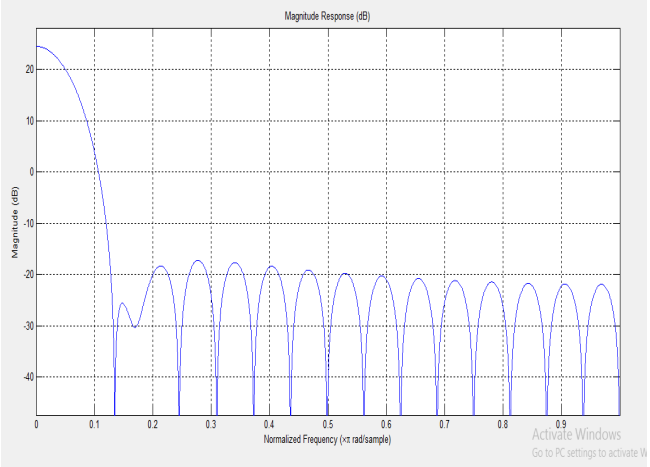

Figure 9: Magnitude Response for the hamming window.

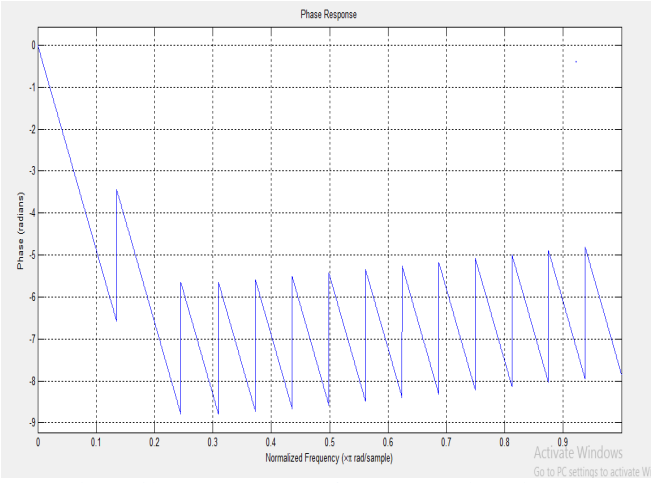

Figure 10: Phase response for the hamming window.

B. RESULTS

The ERD signal was provided to two channels of the Emotiv EPOC and the data was collected. 5 datasets were taken and then processed to get the results for the presence and location of ERD in the signal. The results were found as follows in channel 1&14:

For Channel 1
Trial 1: No ERD
Trial 2: At about 41st window out of total 50 windows captured
Trial 3: No ERD
Trial 4: In Between. At about 27th window
Trial 5: No ERD

For Channel 14
Trial 1: At about 16 window out of total 50 windows captured
Trial 2: Near the start. At about 5-7th window
Trial 3: Very few chances of ERD. A small power spike at about 7th window
Trial 4: In Between. At about 27th window
Trial 5: No ERD

For the rest of the channels, there was no ERD detected. However there was some crosstalk as explained earlier and hence there was a small power spike in a few of them.

We will discuss Trial 1 in detail:

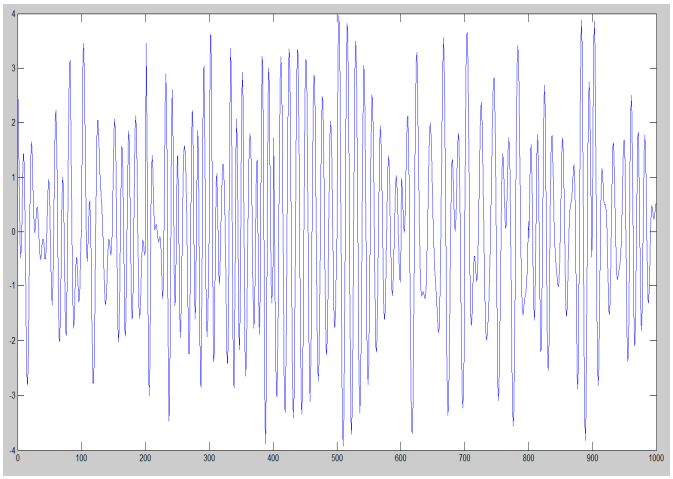

Figure 11: Original Data for channel 14 of trial 1.

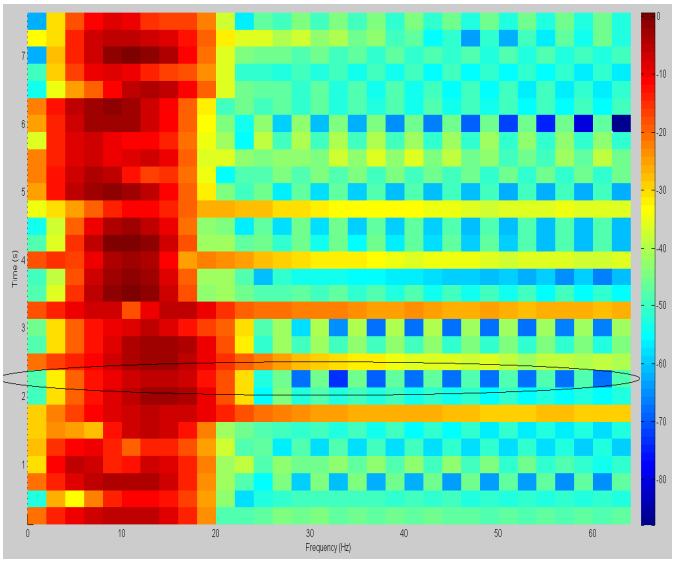

Figure 12: Spectrogram for the above data.

The power spectral density shows us that there is an increase of power between t = 2.125 and t = 2.375. We have plotted a single strip of power at t = 2.125 sec to show the increase in power between frequency band of 20 – 30 Hz.

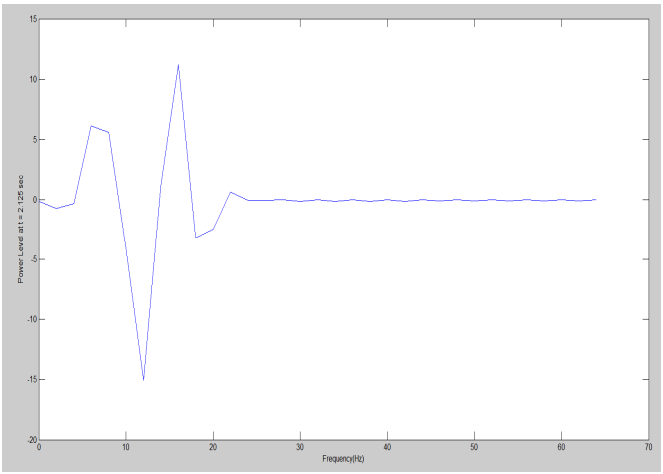

Figure 13: Power Level of the data at t = 2.125 sec

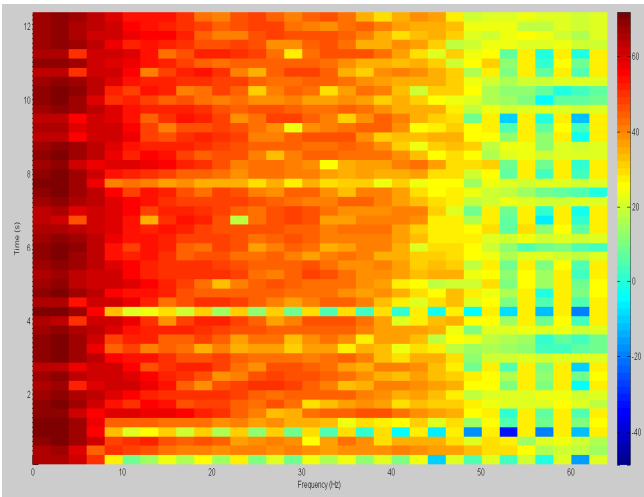

Figure 14: Spectrogram for the collected and processed data

As we can see the spectrogram of the processed data shows two major frequency peaks in the beta bands (10 to 20). One is at t = 0.8 sec and the other at t = 4.2 sec with the second one being more prominent. This is similar to the result we obtained in the original data as is visible from the spectrograph. The difference is in the fact that in the original spectrogram, the 2 power spikes are distributed in the entire graph at t = 2 and 6 sec. However, in the processed spectrogram, the power spikes are at t = 1 and 4 sec. This is due to the fact that the input and output from the two computers were not synchronized and also the data provided for ERD contained 1000 data points while the data collected was 1600 data points.

The following graph is ratio of beta power of the signal to the total power in the signal vs the window number. This graph has a very clear spike near the 16th window where we can see that power percentage rises above 60 %. Hence this is the point when ERD was simulated. In real case scenario, we will see that power due an ERD falls rather than rises. This graph also contains another smaller peak at the 6th window out of the total 50. This window corresponds to the peak in the above spectrogram at t = 1 sec. The rise in power may be due to the noise in the system. The beta power was calculated using the band power function of Matlab as explained earlier.

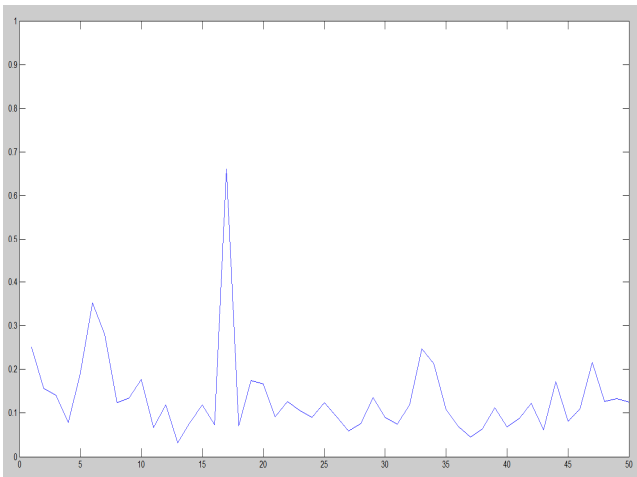

Figure 15: Percentage Band power in beta band in channel no. 14.

For the same data set, the result of the above in the other channel to which data was provided - channel 1 is:

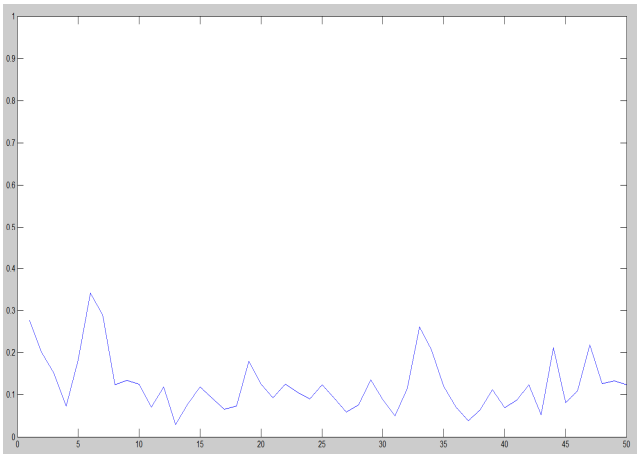

Figure 16: Percentage Band power in beta band in channel no. 1

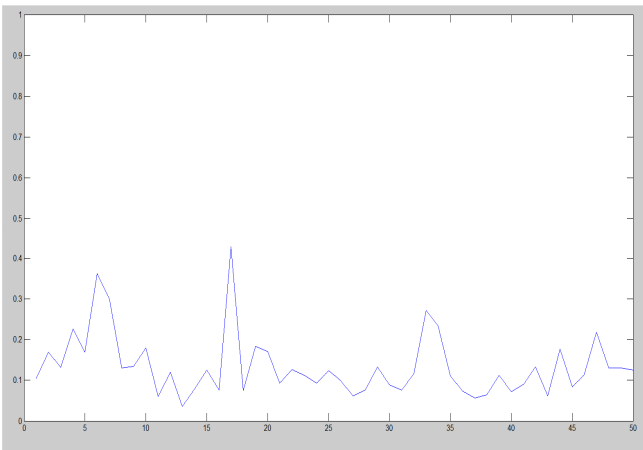

Figure 17: Percentage Band power in beta band in channel no. 13.

We also found a crosstalk due to which there was a peak of power in some other channels as well.

We can see from Fig. 17 the rise of power in the 16th window. But the power level is well below 60% and hence it cannot be considered an ERD.

VII. Online Collection of EEG Data

After the collection of single datasets and offline processing of the data, we can move on to the online collection and feedback of data to the user. In this part, we give user a set of task that he has to imagine. This is called motor imagery [41], [42]. The computer will collect the data for those tasks separately and store them in the workspace. The user will see a separate screen, which will display the tasks for the user, the power base line for presence of ERD and the current beta power line for the EEG data. The power base line is set at 0.45 of the total. This value is calculated using the previous offline ERD detection. In most of the cases, we observed that the percentage power of ERD was above 0.50 and noise power was below 0.40. Hence, we have chosen an intermediate value of 0.45 to differentiate between the presence and absence of ERD. The current beta band power is displayed in discrete steps of 10 % of maximum to avoid overloading the system and also provides a nice clear interface to the user.

The system developed worked perfectly and was able to provide the feedback to the user. Though, we weren't able to collect actual human data and test the system in natural conditions due to time constraints, we will be explaining the methodology for getting human data in detail. The following code snippet is used for online data acquisition and providing a feedback back to the user.

A. Code Snippet

```
global B;                                                    %declaring the matrix which will
store the EEG data.
B = zeros(14,32,range);                                      %initializing the matrix B of size
14*32*range with zeros. 14 is the number of channels of Emotiv EPOC.
global n;                                                    %contains the name of variable with
which dataset are stored in workspace.
global j;                                                    %j is the variable which will keep
track of the no. of datasets already stored in the workspace.
j = 1;                                                       %initializing j to 1.
```

```
i = 1;                                                          %i is a variable which will keep
track of the no. of window already stored in the matrix B.
count = size(A,2);                                              %count is total no. of datasets
required by the user and calculated here by knowing the size of task matrix A.
taken = evalin('base','whos(''lsl*'')');
buffername = genvarname(['lsl_' streamname '_stream'],{taken.name});
inlet = create_inlet();                                         %create_inlet() calls which a
function which initiates the data stream from the Emotiv EPOC.
assignin('base', buffername, create_streambuffer(inlet.info()));
screenid = max(Screen('Screens'));
Screen('Preference', 'SkipSyncTests', 1);
global win;                                                     %declaring a global window handle -
win.
win = Screen('OpenWindow', screenid, 0);                        %providing the initial screenid to
the win handle.
[w, h] = Screen('WindowSize', win);                             %measuring the windows size and
saving the values to w (width) and (height).
x = 500;
lw = 2;
pow = 0;
Screen('textfont',win, 'times');                                %setting the font of the text to be
displayed on the screen as times.
Screen('textsize',win, 50);                                     %defining the text sixe to be 50
pts.
Screen('DrawLines', win, [0, w; h-x, h-x], 2*lw, [255, 255; 255, 255; 255, 255]);
th = timer('Period', 1.0/refreshrate,'ExecutionMode','fixedRate','TimerFcn',@(timerhandle,
eventdata)on_timer,'StartDelay',0.2,'Tag',['lsl_' genvarname(streamname) '_timer']);
%The timer command defines a timer instance with the various object properties to define the timer. The
timer is discussed in detail in the report.
start(th);                                                      %The start command initiates the
timer instance th.
function on_timer
try                                                             %Try block to catch any error that
might occur during the collection and storage of data.
Screen('drawtext', win, A{j}, w/2.2, h/3, [200, 200, 200, 200]);        %storing the value of A(j) on the
RAM to be displayed on the screen at the pointer's end position as w/2.2 and h/3.
Screen('DrawLines', win, [0, w; h-x, h-x], 2*lw, [255, 255; 255, 255; 255, 255]);
Screen('DrawLines', win, [0, w; h-pow*1000, h-pow*1000], 2*lw, [150, 150; 150, 150; 150, 150]);
Screen('flip', win);                                            %dumping the value of the screen
command on the screen using the flip command.
[chunkdata,timestamps] = inlet.pull_chunk();                %pull_chunk() gets the data from EmotiV EPOC in
chunkdata and also a timestamp on the data.
if j < count + 1                                                %Loop to keep a check on the no. of
datasets that have been stored in the workspace.
if i < range+1                                              %Check to ensure whether the data is in range.
if ~(isempty(chunkdata(1,:)))                           %Since the data in this case is received once in
four timer values it is a check to ignore empty matrix.
pow = bandpower(chunkdata(channelno,:),128,[12 18])/bandpower(chunkdata(channelno,:),128,[0 64]);
B(:,:,i) = chunkdata;                                %Chunkdata value is stored in matrix B at the
appropriate position.
i = i+1;                                              %The variable i is incremented.
end
else
n = strcat(A(j),num2str(j));                            %n is the variable name which will store the
datasets. formed from concatenation of task for which data is being recorded and j.
n =  n{1};                                              %strcat function leads to a 1*1 cell containing a
string and this command is to extract the string from that cell.
display('Data Collected');                              %'Data Collected' is displayed on the screen to
indicate that data has been collected.
assignin('base',n,B);                                 %'Data has been assigned to a new variable 'data' in
the workspace.
display('Data Stored');                                 %'Data Stored' is displayed on the screen to
indicate that data has been stored in workspace.
B = zeros(14,32,range);                                 %Reinitializing the matrix B of size 14*32*range
with zeros to ease error dectection and its rectification.
i = 1;                                                  %Reinitializing the range for a new dataset.
j = j+1;                                                 %Incrementing j as the data has been successfully
stored.
end
else
Screen('drawtext', win, 'Finished', w/2.2, h/3, [200, 200, 200, 200]);
Screen('flip', win);                                        %dumping the value of the screen command on the
screen using the flip command.
pause(2);
Screen('CloseAll');
stop(t);                                                    %Command to stop the timer instance th.
```

```
delete(th);                                          %Command to delete the timer instance th.
return;                                              %Command to return the control to the
workspace.
end
catch e                                                      %Catch Block to display the error
message in case of error.
```

The above code accepts a matrix A as an input which contains the motor imagery tasks to be thought by the user and gives us an interface to display the task and the power values for the signal. The final output is stored in the workspace in the form of number of 3-D matrices for the required number of channels and a specific time. These values can be further used for offline processing.

For a core i7 system, the above code skips about 10% of the data points. The reason for the above is high amount of processing required to acquire the data and provide the feedback. Using only the required channels for processing the data can solve this problem.

B. Psychophysics Toolbox

The Psychophysics Toolbox is a software package for Matlab, which act an interface between Matlab and computer hardware [43]. The Matlab Software, which is a high – level interpreted language is abstracted from hardware details and provides a friendlier development environment but it is not possible to get a precise stimulus display through Matlab. The Psychophysics Toolbox allows software specification of the stimulus, thus allowing us to achieve full control of the hardware for precise stimulus display. This tool makes it easier to synthesize and demonstrates accurately controlled visual and auditory stimuli and interact with the observer. This is a very useful tool for programmers who wish to have a precise timing of the events along with a hardware control.

The core routine of Psychophysics toolbox allows:

- Synchronization with the vertical retrace
- Access to the display frame buffer and color lookup
- Support of millisecond timing
- Access to OpenGL commands from Matlab (thus encapsulating the hardware)
- Facilitation of the collection of observer responses.

The Screen.mex file is the heart of the Psychophysics toolbox and provides many commands to control the display screen [44]. The screen command provides a very fast control of the screen to the user. Also, we can create an infinite number of off-screen windows (in memory, not visible), which can then be shown, sequentially, as a movie, by copying to an onscreen window. Thus, the screen command is a very useful tool, which can display texts, videos and dialogs and also provide frame-accurate timings, thus reducing the amount of chaos in execution and debugging.

The Screen command used in the above code displays the motor imagery activity, the beta power reference line and the current value of power in the beta band for the signal. While performing a real time experiment on a user subject, each data set should be collected right after the base power is shown. Screen command stores these values in the memory and flips them onto the screen. Thus, the time lag between the execution of screen command and displaying the values on the screen is almost negligible. Hence, it is necessary to use the screen command in the above code rather than Matlab's inbuilt graphic functions.

C. Functional Electrical Stimulation

FES or Functional Electrical Stimulation is a rehabilitation technique that uses electrical currents to activate nerves or muscle affected by paralysis [2]. The basic aim is to restore functional movement in people with disabilities.

The FES takes the advantage of the fact that the presence of the electric field in nervous tissue leads to the depolarization of neuronal lipid membranes. Hence, neurons fire action potential in the presence of an electric field. This leads to an artificial contraction of skeletal muscles, which causes an actuation of joints by stimulating muscles, which exert torque about the joint [45]. The amount of torque and hence the joint angle can be controlled by varying the intensity of simulation delivered to the muscles.

The application of FES includes correction of disabilities such as Stroke, Foot Drop, multiple sclerosis, cerebral palsy and Parkinson's disease [48]. The Functional Electrical Stimulators to be used for the rehabilitation of stroke patients is the Phenix Liberty + Console [1] from the company Vivaltis [46]. This stimulation and biofeedback wireless neuromuscular reprogramming contains 2-pathway stimulation and 2 channels for biofeedback. The device must be triggered when the beta power level crosses the base power level, thus detecting an ERD and stimulating the motor muscles.

VIII. Conclusion

In this report, we have discussed in detail the development of a system, which can be used to acquire the EEG signal using Emotiv EPOC and process the signal both offline and online to detect ERD/ERS. The user is also provided with a real-time feedback, which he/she can use, for training purposes to make his motor imagery stronger. The system developed can be used to stimulate muscles by triggering the FES device – Phenix. We have also created efficient algorithms to improve the acquired data, thus reducing the scope for missing of a data point or losing an ERD/ERS. The system has been tested rigorously to detect any

errors or loss of data and it is a huge improvement from the initial data acquisition code provided by OpenViBe. Though, it is customized for use along with the Emotiv EPOC, it can be used with other BCI devices as well.

In future, this system can be used as a base to develop low cost neuro-rehabilitation device for stroke patients on an industrial scale.

IX. References

1. *PHENIX + LIBERTY CONSOLE*. (n.d.). (Vivaltis) Retrieved from Vivaltis: http://www.vivaltis.com/gammes/phenix/phenix-liberty-console-50-448-1.html#content
2. Rushton, D. N. (1997). Functional electrical stimulation. *Physiological Measurement , 18* (4), 24-275.
3. *DEMAR*. (n.d.). Retrieved from INRIA: http://www.lirmm.fr/demar/
4. *INRIA*. (n.d.). Retrieved from http://www.inria.fr/inria/enbref.en.html
5. *University of Montpellier 2* . (n.d.). Retrieved from http://www.univ-montp2.fr/index.php
6. *LIRMM*. (n.d.). Retrieved from http://www.lirmm.fr/lirmm_eng
7. *CNRS*. (n.d.). Retrieved from http://www.cnrs.fr/
8. Neurol, C. a. (2013, 07). Cost of stroke in France. *PubMed* , 1094-100.
9. Dalal, P. (2007). UN millennium development goals: Can we halt the stroke epidemic in India? *Annals of Indian Academy of Neurology , 19* (3), 130-136.
10. A, D. (2012, 11). Neurologic disability: a hidden epidemic for India. *PubMed* , 2146-7.
11. Perez-Marcos D1, S. M.-V. (2012). A fully immersive set-up for remote interaction and neurorehabilitation based on virtual body ownership. *Frontiers in Neurology , 3*, 149-153.
12. M, T. (2005). Hans Berger (1873-1941)--the history of electroencephalography. *Europe PubMed Central* , 307-313.
13. Ranganatha Sitaram, A. C. (2007). *FMRI Brain–Computer Interface: A Tool for Neuroscientific Research and Treatment.* PuBMed Central.
14. Schalk, G. McFarland, D., Hinterberger, T., Birbaumer, N., & Wolpaw, J. (2004). BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. *IEEE , 51*, 1034-43.
15. Ozols, K., Greitans, M., & Shavelis, R. (2012). EEG data acquisition system based on asynchronous sigma-delta modulator . *2012 13th Biennial Baltic Electronics Conference (BEC),* (pp. 183,186,3-5). IEEE.
16. Julien Freudiger, D. T. (2003). *Brain states analysis for direct brain-computer communication.* Ecole polytechnique fédérale de Lausanne. Infoscience.
17. Birbaumer, N. (2006). Breaking the silence: Brain–computer interfaces (BCI) for communication and motor control. *Psychophysiology , 43* (6), 517-532.
18. Eric C Leuthardt, G. S. (2004). A brain–computer interface using electrocorticographic signals in humans. *JOURNAL OF NEURAL ENGINEERING* , 63-71.
19. Wang, C., Phua, K. S., Ang, K. K., Guan, C., Zhang, H., Lin, R., et al. (2009). A feasibility study of non-invasive motor-imagery BCI-based robotic rehabilitation for Stroke patients. *4th International FrC1.4 IEEE EMBS Conference on Neural Engineering* (pp. 271-4). Antalya, Turkey: IEEE.
20. Niedermeyer E, L. d. (2004). *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields.* Lippincott Williams and Wilkins.
21. Cox, S. J. *Electroencephalography.* Rice University , Computational and Applied Mathematics, Houston.
22. Fatehi, T. A., & Suleiman, A.-R. B. Features Extraction Techniques of EEG Signals For BCI Application. *International Arab Conference for Information Technology.* Naif Arab University for Security Sciences.
23. Chunchu Rambabu, B. R. (2014). EEG Signal with Feature Extraction using SVM and ICA Classifiers. *International Journal of Computer Applications , 85* (3).
24. R. Pratap-Chand, M. S. (2009). Cognitive evoked potential (P300): a metric for cerebral concussion. *Acta Neurologica Scandinavica , 78* (3), 185-9.
25. Jacobs, G. (2013). *SSVEP-based BCI control for navigating a robot.* Radboud University Nijmegen. Bachelor Thesis of Artificial Intelligence.
26. Luis Fernando, N.-A. a.-G. (2012, 01). Brain Computer Interfaces, a Review . *Sensors* , 1211-79.
27. Hinterberger T, S. S. (2004, 06). Brain-computer communication and slow cortical potentials. *PubMed* , 1011-8.
28. Birbaumer N, E. T. (1990, 01). Slow potentials of the cerebral cortex and behavior. *PubMed* , 1-41.
29. G Pfurtschellera, A. A. (1979). Evaluation of event-related desynchronization (ERD) preceding and following voluntary self-paced movement. *Electroencephalography and Clinical Neurophysiology , 46* (2), 138-46.
30. Arafat, I. *Brain–Computer Interface: Past, Present & Future.* International Islamic University Chittagong (IIUC) , Dept. of Computer Science &Engineering. Chittagong, Bangladesh: academia.
31. *Honda, ATR and Shimadzu Jointly Develop Brain-Machine Interface Technology Enabling Control of a Robot by Human Thought Alone*. (2009, 03 09). Retrieved from Honda: http://world.honda.com/news/2009/c090331Brain-Machine-Interface-Technology/
32. Vijay Khare, J. S. (2011). Brain Computer Interface Based Real Time Control of Wheelchair Using Electroencephalogram. *International Journal of Soft Computing and Engineering (IJSCE) , 1* (5), 2231-2307.
33. *What Are The Features Of The EPOC?* (n.d.). Retrieved from Emotiv: https://emotiv.zendesk.com/hc/en-us/articles/200785669-What-are-the-features-of-the-EPOC-
34. *Emotiv Support.* (n.d.). Retrieved from Emotiv: http://www.emotiv.com/support.php
35. *Discover OpenViBE.* (n.d.). Retrieved from Openvibe: http://openvibe.inria.fr/discover/
36. Renard, A. L. (n.d.). *OpenViBE: Open-Source Software for Brain-Computer Interfaces*. Retrieved from ercim-news: http://ercim-news.ercim.eu/en78/rd/openvibe-open-source-software-for-brain-computer-interfaces
37. Jozef. (2012, 06 28). *Tutorial : The most basic OpenViBE setup*. Retrieved from OpenVIBE: http://openvibe.inria.fr/tutorial-the-most-basic-openvibe-setup/
38. Ibonnet. (2011, 08 30). *Acquisition Server*. Retrieved from OpenVIBE: http://openvibe.inria.fr/acquisition-server/
39. *Data Acquisition*. (n.d.). Retrieved from NI: http://www.ni.com/data-acquisition/
40. *Bandpower*. (n.d.). Retrieved from Mathworks: http://www.mathworks.in/help/signal/ref/bandpower.html
41. Pfurtscheller, G., & Neuper, C. (2001). Motor imagery and direct brain-computer communication. *Proceedings of the IEEE , 89* (7), 1123-34.
42. Jeannerod, M. (1995). Mental imagery in the motor context. *The Neuropsychology of Mental Imagery , 33* (11), 1419-32.
43. *Psychtoolbox-3.* (n.d.). Retrieved from Psyhtoolbox Wiki: https://www.psychtoolbox.org/HomePage
44. *Screen.mex.* (n.d.). Retrieved from psychtoolbox: https://docs.psychtoolbox.org/Screen
45. Toussaint, M., Andreuy, D., Fraisse, P., & Guiraudy, D. (2010). Wireless Distributed Architecture for Therapeutic Functional Electrical Stimulation: a technology to design network-based muscle control. *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE* (pp. 6218-21). Buenos Aires: IEEE.
46. (n.d.). Retrieved from Vivaltis: http://www.vivaltis.com/

47. Dimiter Prodanov, E. M. (2003, 11). Functional electric stimulation for sensory and motor functions: progress and problems. *Biomedical Reviews* , 23-50.