KAUNAS UNIVERSITY OF TECHNOLOGY

ELECTRICAL AND ELECTRONICS ENGINEERING FACULTY

Aravind Prasad

Feature Extraction and Classification for Motor Imagery in EEG Signals

Master's Degree Final Project

Supervisor

prof. dr. Vaidotas Marozas

KAUNAS, 2016

KAUNAS UNIVERSITY OF TECHNOLOGY

ELECTRICAL AND ELECTRONICS ENGINEERING FACULTY

Feature Extraction and Classification for Motor Imagery in EEG Signals

Master's Degree Final Project

Electronics Engineering (621H61002)

Supervisor

(signature) prof. dr. Vaidotas Marozas

(date)

Reviewer

(signature) prof. dr. Vytautas Deksnys

(date)

Project made by

(signature) Aravind Prasad

(date)

KAUNAS, 2016

KAUNAS UNIVERSITY OF TECHNOLOGY

Electrical and Electronics Engineering faculty

Aravind Prasad

Electronics Engineering (621H61002)

"Feature Extraction and Classification for Motor Imagery in EEG Signals"

DECLARATION OF ACADEMIC INTEGRITY

10 June 2016

Kaunas

I confirm that the final project of mine, **Aravind Prasad**, on the subject "Feature Extraction and Classification for Motor Imagery in EEG Signals" is written completely by myself; all the provided data and research results are correct and have been obtained honestly. None of the parts of this thesis have been plagiarized from any printed, Internet-based or otherwise recorded sources. All direct and indirect quotations from external resources are indicated in the list of references. No monetary funds (unless required by law) have been paid to anyone for any contribution to this thesis.

I fully and completely understand that any discovery of any manifestations/case/facts of dishonesty inevitably results in me incurring a penalty according to the procedure(s) effective at Kaunas University of Technology.

(name and surname filled in by hand) *(signature)*

Aravind Prasad / Feature Extraction and Classification for Motor Imagery in EEG Signals. *Master's* Final Project / supervisor prof. dr. Vaidotas Marozas; Kaunas University of Technology, Faculty of **Electrical and Electronics Engineering**.

Research area and field: Biomedical Engineering / Electroencephalography

Keywords: *EEG, BCI, Hjorth, LDA, kurtosis, motor imagery*

Kaunas, 2016. 57 p.

Summary

Electroencephalography is a non-invasive technique which is used for recording the neurophysiological reactions in the brain. It measures the activity of neurons. This report consists of different steps taken for finding that it is possible to control bionic arm with imaginary data of motor movement. The electroencephalographic signals were obtained from Physionet biosignal database. Feature extraction and its analysis is done for ten subjects. The different features were calculated for different segments of the obtained signal. The features extracted were inspired by Hjorth parameters and a higher order statistics - kurtosis. The signal processing algorithm for the process is explained in the report. The supervised feature classification is implemented using the Linear Discriminant Analysis. The obtained accuracy for the classifier was found to be around 60-70% depending on the electrodes and type of data (real or imaginary).

Aravind Prasad / Numanomų motorikos požymių išskyrimas ir klasifikavimas elektroencefalografiniuose signaluose / vadovas prof. dr. Vaidotas Marozas; Kauno Technologijos Universitetas, Elektros ir Elektronikos inžinerijos fakultetas, Elektronikos inžinerijos katedra.

Tyrimų sritis ir kryptis: Biomedicininė Inžinerija / Elektroencefalografija

Raktažodžiai: *EEG, smegenų kompiuterio sąsaja, Hjorth, tiesinė diskriminatinė analizė, ekscesas, motorikos požymiai*

Kaunas, 2016. 57 p.

Santrauka

Šio darbo tikslas - sukurti signalų apdorojimo algoritmą leidžiantį elektroencefalografiniuose signaluose atpažinti numanomus motorinius judesius ir paversti juos komandomis siunčiamomis į bioninę ranką. Analizuoti elektroencefalografiniai signalai gauti iš Physionet biosignalų duomenų bazės. Signalų apdorojimo algoritmą sudaro pirminio apdorojimo, požymių skaičiavimo ir klasifikatoriaus dalys. Siekiant algoritmo paprastumo ir greitaveikos pasirinkta požymius įvertinti laiko srityje analizuojant signalų variabilumo savybes. Signalų požymių rinkinį sudaro Hjorth‘o pasiūlyti parametrai (aktyvumas, mobilumas ir kompleksiškumas) bei ketvirtos eilės statistinis parametras – ekscesas. Klasifikavimui naudota tiesinė diskriminantinė analizė. Algoritmas ištirtas su signalais užregistruotais atliekant realius ir numanomus motorinius judesius. Gautas klasifikavimo tikslumas siekė 60-70 %, priklausomai nuo elektrodų ir duomenų tipo (realus ar numanomi).

Acknowledgements

This thesis work would not be possible without the help of many people.

First and foremost I would like to thank my supervisor and my guide doc. dr. V. Marozas. This would be an impossible feat without his timely guidance and impeccable knowledge. He has helped me a lot in many tight spots where I was struggling with the simulations.

Next I would like to thank all the phD students and research associates in the Biomedical Engineering Department for giving me space in the lab and helping me with my comfort within the lab which gave me a good environment to work and also for helping me out of difficult scenarios by discussing the issues and finding solutions. On this note, I would like to thank Andrius Sakalauskas for his help in translating the summary to Lithuanian language.

I would also like to thank Project Coordinators prof. D Andriukaitis and prof. D Gailius, Department Head prof. D. Navikas and all the lecturers from the Electrical and Electronics department for their suggestions and ideas.

Next I would like to thank my wife Sripriya Suseela for being there for me.

Last but not the least I would like to thank my family, friends and colleagues for their inspiration and support.

Contents

List of figures

List of Tables

List of Abbreviations

BCI - Brain Computer Interface

BMI - Brain Machine Interface

EEG - Electroencephalography

EMG - Electromyography

EOG - Electrooculography

MEG - Magnetoencephalography

MI - Motor Imagery

ERD - Event-Related Desynchronization

ERS - Event-Related Synchronization

EDF - European Data Format

FIR - Finite Impulse Response

DFT - Discrete Fourier Transform

LDA - Linear Discriminant Analysis

SVM - Support Vector Machine

S D - Standard Deviation

Introduction

A Brain Computer Interface (BCI), also called as a Brain Machine Interface (BMI), is a system which is used as a medium of communication between the human brain and the external world. It is comprises of both hardware and software and enables us humans to interact with the surroundings. This interaction happens without the involvement of the peripheral nerves and muscles. It uses control signals generated from electroencephalographic (EEG) activity. EEG mostly measures the currents that are generated during synaptic excitations of the dendrites of neurons in the cerebral cortex. A BCI is a non-muscular system which helps in transmitting a person's commands or thoughts to external devices such as computers, speech synthesizers, assistive appliances, and neural prosthetics. It is very useful and life changing for individuals with severe motor disabilities. An interface like this would improve their life and would, at the same time, reduce the cost of intensive care. It is an artificial intelligence system which can understand and recognize a particular set of patterns in the EEG signals. The following five consecutive stages are the different processes used for pattern recognition: signal acquisition, pre-processing or signal enhancement, feature extraction, classification, and the control interface. The signal acquisition stage is where the signals from the brain are captured. This stage also performs noise reduction and artefact processing. The pre-processing stage prepares the signals in a suitable form for further processing. The feature extraction stage recognises discriminative data in the brain signals that have been recorded. Once the EEG data is taken, the signal is mapped to a vector containing effective and discriminant features from the observed signals. This is known as the feature vector. The extraction of information from this data is a very challenging task. Brain signals are mixed with other signals. These signals come from a set of brain activities and they overlap in both time and space. Moreover, the signal is not usually stationary and may also be distorted by artefacts such as electromyography (EMG) or electrooculography (EOG). EMG artefacts correspond to the muscle movement obtained during the swallowing action or the movement of the tongue and EOG artefacts are the ones obtained during the blinking action of the eye. The feature vector must also be of a low dimension, in order to reduce the complexity of the feature extraction stage, but without relevant information loss. The classification stage classifies the signals taking the feature vectors into account. The choice of good discriminative features is therefore essential to achieve effective pattern recognition, in order to decipher the user's intentions. Finally the control interface stage translates the classified signals into meaningful commands for any connected device, such as a wheelchair or a computer [1]. The main problem in BCI design is the accuracy of classification the EEG signals. At present, no one single method of both feature extraction and classification could achieve a one hundred percentage accuracy [2]. Keeping these things in mind it was decided to design a signal processing system suitable for investigation of motor, real and imaginary, tasks in artificial limb control. This was set as the aim of this work and in order to achieve this aim, certain goals were set. These goals are discussed in the section 1.4. This report discusses all the methods and procedures followed in order to achieve this aim.

1 Literature review and analysis

This chapter explains about what a BCI is and how it works. There is also a discussion about different parts of brain and where the sensorimotor region is. These details are understood from the physiology of the brain. The aim and goals are set with the help of this literature analysis.

1.1 Brain Computer Interface

BCI is a challenging application of signal processing, machine learning, and neuroscience [3]. A BCI captures different activities from the brain which are associated with real motion, mental tasks and external stimuli and in turn enables non-muscular communication and a control channel for conveying messages and commands to the external world [4]. A non-invasive BMI uses recordings of brain activities such as EEG and magneto encephalogram (MEG). Using EEG is the most practical for engineering applications because of the simplicity of the device and high temporal resolution [5]. A typical EEG signal is shown in Figure 1. An EEG signal is extremely complex and has a lot of cross over frequencies from other parts of the human body.

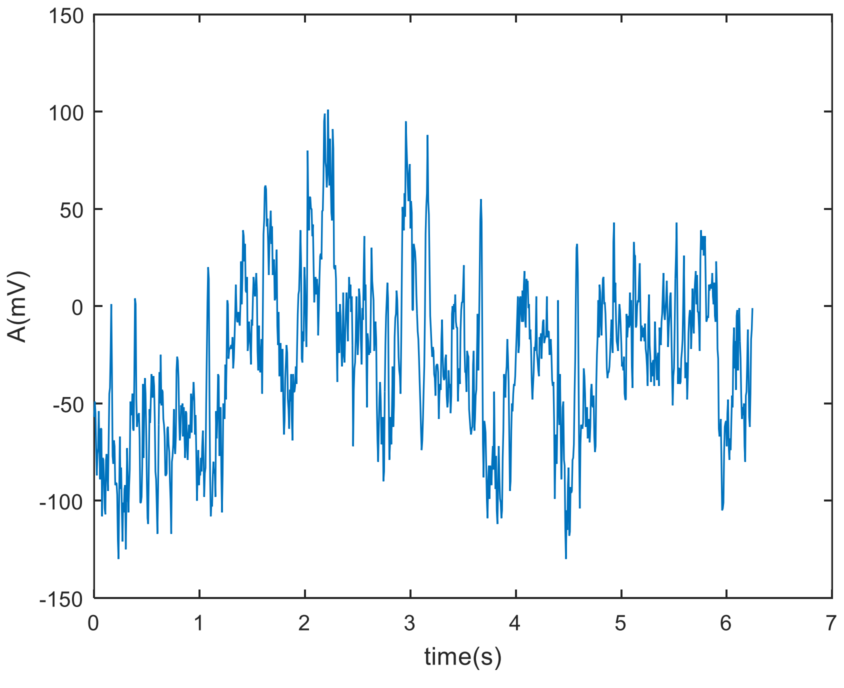

Figure 1 A typical EEG signal for a single channel

The BCI systems allows the control of an artificial device based on the features extracted from voluntary electric, magnetic, or other physical manifestations of brain activity collected either epidurally or subdurally, from the sensorimotor cortex or from the scalp or in invasive electrophysiological manner [6]. Figure 2 shows a basic BCI Layout.

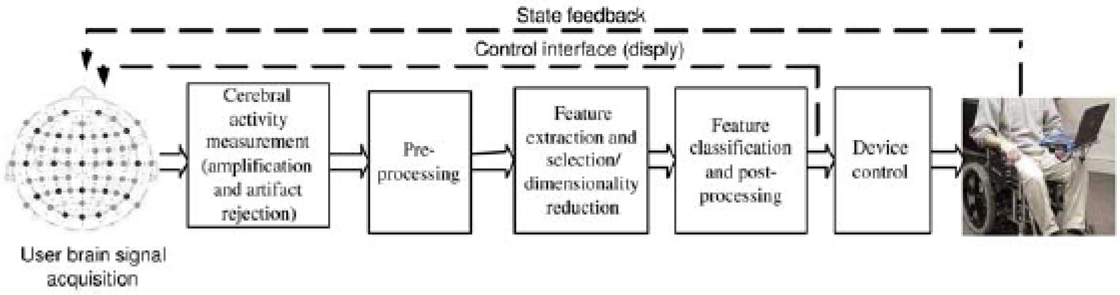

Figure 2 Basic BCI Layout [6]

A BCI need not pertain to people with disabilities, it can also be used for accessing areas in an industrial environment where it is unsafe for human hands to reach and carry out their work. It can be used as an extension of a human limb where it is required. In this case a human can control the BCI as he controls his own limb.

When a movement or a motion is carried out, the potentials are developed in certain parts of the brain. These potentials are then recognized by the respective activated electrodes, and the respective features are calculated. These potentials developed due to movement is seen in the sensorimotor area of the brain as discussed and this is the motor imagery of the brain.

An important technique for enabling BCIs related with motor imagery (MI-BCI) is efficient decoding of signals from the motor cortex area. This decoding will lead to practical biomedical applications in rehabilitation and neuro prosthesis. For instance, real and imaginary movements of hands and feet invoke a change in the sensorimotor rhythm in different regions of the brain. Hence, if we can capture these changes in the rhythm from EEG, in the presence of measurement noise and rhythms related to other brain activities, we can classify the EEG signal associated with imagination of different motor actions such as hand, arm, or foot movement [5].

Non-invasive EEG is a weak signal and it is very easily affected by noises from the electrodes and even the supply. This increases the difficulty in feature extraction. The feature extraction methods are generally divided into time and frequency methods, parametric modelling and modelling the neural firing rate. Time and frequency methods generally contain power spectrum analysis, wavelet analysis and common spatial patterns. Power spectrum has been widely used due to its easy application and high computational speed. But power spectrum analysis does not provide any time domain information [7].

Several feature types have been used with EEG-based BCIs. Among them, the most popular are logarithmic band power estimates with more or less subject-specific optimization of the frequency bands. In this thesis, the features are calculated using a generalization of Hjorth parameters. It is also called as Time Domain Parameters [8]. Another temporal feature called kurtosis, which is a higher order statistic, is also calculated.

The signal processing is done in Matlab [9]. It is a high level technical computing language with graphical interface used for intensive mathematical computations. It is designed to be more efficient and more accurate than typical programming languages like C++ and Java. It provides users with various tools for data analysis and visualization. It is the primary tool used in this thesis report for implementation of the feature extraction process using Hjorth parameters and kurtosis. The software also provides various toolboxes which can be used for EEG signal processing [10].

1.2 Physiology of human brain

There are three main parts of brain namely, cerebrum, cerebellum, and brainstem:

- The **cerebrum** is the largest part of the brain. It is composed of 2 hemispheres, the right and left. The different functions like interpretation of touch, vision and hearing, as well as speech, reasoning, emotions, learning, and fine control of movement are done by this part of the brain.
- The **cerebellum** is situated under the cerebrum. Its main use is to coordinate muscle movements, maintain posture, and balance.
- The **brainstem** is a set of different parts which includes the midbrain, pons, and medulla. Apart from performing many automatic functions such as breathing, heart rate, body temperature, wake and sleep cycles, digestion, sneezing, coughing, vomiting, and swallowing, its main work is to act as a relay centre connecting the cerebrum and cerebellum to the spinal cord.

The above mentioned cerebral hemispheres have distinct fissures/folds. These divide the brain into lobes. Each of the hemispheres has 4 lobes: frontal, temporal, parietal, and occipital as shown in Figure 3. Each lobe in turn may be divided, once again, into different areas based on very specific functions. It’s very important to realize and understand that each lobe of the brain does not function alone. There are very complex relationships between the lobes of the brain and between the right and left hemispheres. The different lobes with their specific functions are mentioned below [11].

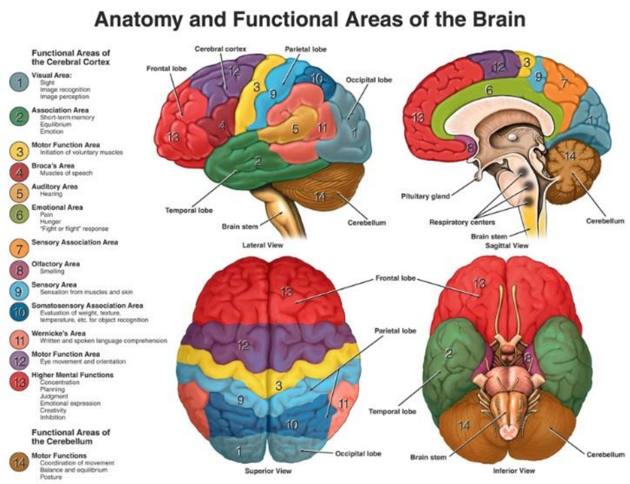

Figure 3 Functional Areas of Brain [12]

The frontal lobe deals with the personality, behaviour, emotions, judgment, planning, problem solving, and speech - speaking and writing (Broca’s area). It also deals with the body movement (motor strip). This is the main area we are focussing on, in this thesis, as the motor movements can be deciphered clearly from the electrodes in this area. The frontal lobe also deals with intelligence, concentration, self-awareness.

The parietal lobe interprets language, words, sense of touch, pain, temperature (sensory strip). It also interprets signals from vision, hearing, motor, sensory and memory. Spatial and visual perception are also observed in this area.

The occipital lobe interprets vision (colour, light, movement). The temporal lobe is where a person can understand language (Wernicke’s area). This lobe also deals with memory, hearing, sequencing and organization

From Figure 3, we can see that the Motor functions of human body is mainly carried out in the areas 3 and 12. Also the muscle responses are recorded in the sensory area 9. This is part of the frontal lobe and the parietal lobe. The motor imagery signals will be more profoundly found in this area of the human brain. Hence it would be wise to select the electrodes within this area [13]. The electrode selection criteria will be discussed in this thesis.

The laterality of the brain is a huge factor in analysing the signals. There are mainly 2 types of laterality when speaking about brain cognitive functions, contralateral and ipsilateral. Contralateral pertains to controlling opposite side of the body and ipsilateral refers to control in the same side of the body. Usually for a human being the cognitive functions are contralateral, but in cases of people who have suffered from stroke, they cannot perform motor action depending on which side of the brain is affected. Hence they need BCI's which can be controlled using ipsilateral motor signals [14].

From the above description of the brain anatomy, we can understand that multiple actions involving all the parts of human body occurs at different parts of the brain and it overlaps to the different regions of the brain. From this we can deduce the complexity of the measured signals. Hence to understand these signals and to classify them based on the motion of the limb is a very challenging and tedious process.

1.3 Current research with respect to motor imagery

Current research aims at identifying voluntary brain activation in patients who are behaviourally diagnosed as being unconscious, but are able to perform commands by modulating their brain activity patterns. This involves machine learning techniques and feature extraction methods such as applied in brain computer interfaces [15].

Several EEG studies also confirm the notion that motor imagery can activate primary sensorimotor areas [13]. It has been observed that when a subject does a voluntary or an involuntary action, the action potentials in the sensorimotor cortex spikes and these spikes are usually observed in the frequency band from 7.5 Hz to 31 Hz.

In another study, a system that allows disabled persons to improve or recover their mobility and communication within the surrounding environment was implemented and validated. The system is based on a software controller that offers the user a communication interface that is matched with the individual's residual motor abilities [16].

The studies for sensorimotor activities are taking place mainly in the *Mu* rhythm and *beta* rhythm of EEG signals. *Mu* rhythms are the *alpha* rhythms in the sensorimotor area. Several studies have examined *Mu* and *beta* rhythm activity during motor imagery. In the recordings from subdural electrodes over sensorimotor cortex, it was found that *Mu* rhythm desynchronization occurs during actual movement but not during thinking about the movement. It has also been found that both handling an object and imagining handling it were associated with desynchronization in the 8-12 Hz band, but the topographies of desynchronization differed. In contrast to that, another study reported that slow potentials associated with actual and imagined hand movements had similar topographies. Yet another study reported that both imagery and movement produce desynchronization in *Mu* and *beta* bands over contralateral sensorimotor areas. Thus, the degree of similarity between the patterns of cortical activation associated with actual movement and those associated with motor imagery remains uncertain [17].

Mu rhythms in the range of 8–12 Hz and *beta* rhythms in the range of 13–30 Hz both originate in the sensorimotor cortex and are displayed when a person is not engaged in processing sensorimotor inputs or in producing motor outputs. They are mostly prominent in frontal and parietal locations. A voluntary movement results in a circumscribed desynchronization in the *Mu* and lower *beta* bands. This desynchronization is called event-related desynchronization (ERD). After a voluntary movement, the power in the brain rhythms increases. This phenomenon, called event-related synchronization (ERS). *Gamma* rhythm is a high-frequency rhythm in the EEG. Upon the occurrence of a movement, the amplitude of *gamma* rhythm increases. *Gamma* rhythms are usually more prominent in the primary sensory area [18].

At present there are a variety of toolboxes which are used to implement the whole processing algorithm for EEG signals. Some of these toolboxes are:

1. EEGLAB
2. Fieldtrip
3. BrainVision Analyzer
4. EEProbe
5. BioSig

These toolboxes are available under the GNU General Public Licence. Most of them are used for processing continuous and event related EEGs and also provide an interactive graphic user interface. They utilize various implementations of pre-processing, feature extraction and classification. The user can conveniently select the required process and extract the required signal depending on their research.

In this thesis none of the above tools are used and all the algorithms are self-implemented and inbuilt functions in Matlab have also been used.

1.4 Aim and goals

The aim of this work is to design a signal processing system suitable for investigation of motor real and imaginary tasks in artificial limb control.

In order to achieve this aim following goals were set:

1. Critical analysis of scientific literature in relevant digital signal processing and brain – computer interface areas.
2. Development and investigation of EEG pre-processing, feature extraction.
3. Selecting the minimal set of most effective EEG channels.
4. Investigation of classification algorithm for motor real and imaginary tasks.

2 Signal database

The signal database is from PhysioNet [19]. It is an open source for different types of biomedical signals. The dataset was created and contributed to PhysioNet by the developers of the BCI2000 instrumentation system, which they used in making these recordings [20].

2.1 Signal Acquisition

The EEGs were recorded from 64 electrodes as per the international 10-10 system. The data are provided in EDF+ format (containing 64 EEG channels, each sampled at 160 samples per second, and an annotation channel). This set of data were used for this research work.

The data set consists of over 1500 one- and two-minute EEG recordings, obtained from 109 volunteers. This database has details pertaining to motor imagery and would be very useful for my area of research.

The 21 electrodes as highlighted in Figure 4 was selected as this region corresponding to the sensory motor cortex. This was discussed in section 1.2

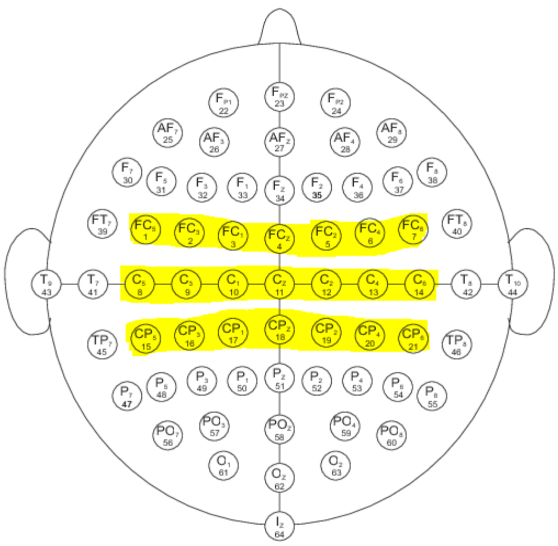

Figure 4 10-10 Electrode Placement

2.2 Experimental protocol

Subjects performed different motor imagery tasks while 64-channel EEG were recorded using the BCI2000 system (Measurement Computing, Inc.; Data Translation, Inc.; National Instruments, Inc.). Each subject performed 14 experimental runs: two one-minute baseline runs (one with eyes open, one with eyes closed), and three two-minute runs of each of the four following tasks:

1. A target appears on either the left or the right side of the screen. The subject opens and closes the corresponding fist until the target disappears. Then the subject relaxes.
2. A target appears on either the left or the right side of the screen. The subject imagines opening and closing the corresponding fist until the target disappears. Then the subject relaxes.

The representation of the trials done is given in the Figure 5. This figure shows the action done by the subject on receiving the cue, i.e. when a target comes on the right side of the screen.

The experimental runs were:

1. Task 1 (open and close left or right fist)
2. Task 2 (imagine opening and closing left or right fist)

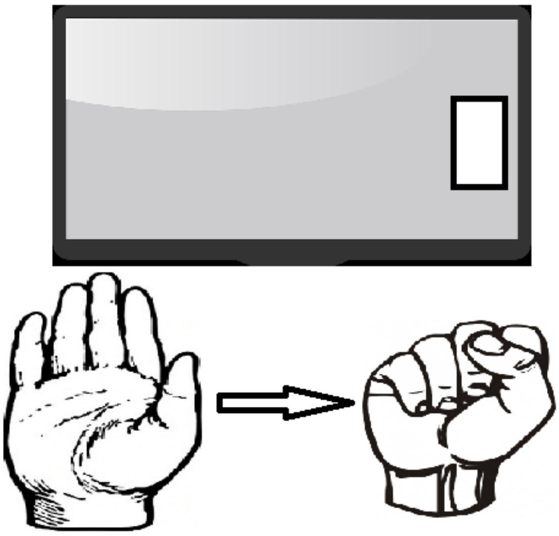

Figure 5 Real hand movement by the subject on seeing the target on the right side of the screen

Each annotation includes one of three codes (T0, T1, or T2):

1. T0 corresponds to rest
2. T1 corresponds to onset of motion (real or imagined) of the left fist.
3. T2 corresponds to onset of motion (real or imagined) of the right fist.

The annotation data plays a major role in the signal processing algorithm. This data is used explicitly for the feature extraction process and the classification algorithm. If this data is not known it would be nearly impossible to do the further steps.

The trails conducted in this database gives a specific insight into how motor signals are observed while imagining it. So in a way it shows how the motor signals will be generated on sensorimotor area of people with disabilities. Hence this database was selected and used for conducting this research.

The data is of 2 minutes and the sampling frequency of the signal is 160 Hz. From this we can see that the number of samples are 19200. The tasks were performed for 4.1 seconds and the relaxing time for the subject was 4.2 seconds. From this we can see that there were around 30 sets of task and relax cycle performed.

3 Research Methodology - The signal processing algorithm

The research methodology used for this work is designing and implementation of a signal processing algorithm. This algorithm pre-processes the EEG signal, extracts the features and classifies it according to the training data given to the classifier.

3.1 Overview of algorithm

In this chapter, we discuss about the different stages of the signal processing and why we use this. We discuss about the signal pre-processing, the different extracted features and the different classification processes. These processes are implemented in Matlab [9]. It is a software used for technical computing which includes the signal processing toolbox. This toolbox is used extensively in this section. The signal processing algorithm was designed on the basis of the literature analysis. It is a standard procedure followed for all the BCI systems [21].

The signal processing algorithm was designed as shown in Figure 6.

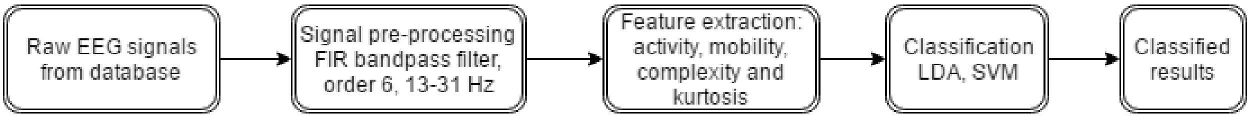

Figure 6: EEG signal processing algorithm

3.2 Pre-processing

As discussed in multiple research papers, the *Mu* rhythm and the *beta* rhythm for the EEG signals fall in the range of frequencies 8Hz to 30 Hz [18]. Hence it was decided to filter the signal from 13 Hz to 31 Hz for the *beta* rhythm as this is the frequency range where the motor movements are recorded.

It is a common practice to filter the signals when dealing with EEG data. It is done in order to temporally smooth them and remove the noise. These noises are considered to be generally found at higher frequencies. Many classic studies list in their methods section a cut-off frequency of 40 Hz or even 30 Hz [22]. Upon doing so, a lot of required data gets filtered out and many features get filtered out. Hence in order to get back to the basics and understand the EEG behaviour, a normal bandpass filtering of the signal was done. By doing this we have very less expensive pre-processing, mathematically.

The raw signal was passed through a bandpass filter of passband 13-31 Hz [18]. In order to analyse the real time data, a causal filter should be used [23]. Hence it was decided to use a Finite Impulse Response (FIR) bandpass filter for this process. A low order of 6 and a little higher order of 10 was set in order to understand the filtered data. The classified results of both these orders were found to be varying and different.

The *Mu* rhythm which lies in the frequency range 7.5-12 Hz corresponds to the rest state of the sensorimotor neurons as discussed in section 1.3. The activity in the beta band correlates to the motor activity. The experiments were carried out in the beta rhythm and the observations were made in this segment.

3.3 Feature extraction

The features that are selected are the statistical descriptors based on the time domain of the EEG. Activity, Complexity, and Mobility were derived in 1970 by Bo Hjorth, and are collectively known as the Hjorth Parameters [8]. Hjorth parameters were selected as features for many studies pertaining to motor imagery. Also a higher order statistical feature, kurtosis was also selected and the feature vectors were calculated by using sliding window method for a specific set of windows.

3.3.1 Hjorth Parameters

This method is interesting because it represents each time step (or window) using only these three attributes and this is done without conventional frequency domain description (such as that of Discrete Fourier Transform (DFT)). The signal is measured for successive epochs (or windows) of one to several seconds. Two of the attributes are obtained from the first and second time derivatives of the amplitude fluctuations in the signal. The first derivative is the rate of change of the signal's amplitude. At peaks and troughs the first derivative is zero. At other points it will be positive or negative depending on whether the amplitude is increasing or decreasing with time. The steeper the slope of the wave is, the greater the amplitude of the first derivative. The second derivative is determined by taking the derivative of the first derivative of the signal. Peaks and troughs in the first derivative, which correspond to points of greatest slope in the original signal, results in zero amplitude in the second derivative, and so forth [24].

1. Activity: The activity parameter represents the signal power, the variance of a time function *y(t)*. This can indicate the surface of power spectrum in the frequency domain.

$$A = var\big(y(t)\big) = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{length(y(t))} \tag{1}$$

where $\bar{y}$ is the mean.

2. Mobility: The mobility parameter represents the mean frequency, or the proportion of standard deviation of the power spectrum. This is defined as the square root of variance of the first derivative of the signal:

$$M = \sqrt{\frac{A(\frac{dy}{dt})}{A}} \tag{2}$$

where, *A* is the activity.

3. Complexity: The Complexity parameter represents the change in frequency. The parameter compares the signal's similarity to a pure sine wave, where the value converges to 1 if the signal is more similar:

$$C = \frac{M(y(t)\frac{dy}{dt})}{M(y(t))} \tag{3}$$

where, *M* is the mobility.

There is no clear agreement as to what the measurements of Hjorth features mean in terms of mental states. It is common sense to assume that the longer a subject remains focused on a specific mental task, the more stable the signal is, and therefore lower the variance of the amplitude fluctuation. However, this assumption does not address the possible effects of fatigue, habituation and boredom [24].

3.3.2 Kurtosis

There are major methods to extract EEG feature, such as temporal features, spectral features and nonlinear features. Few of the temporal features are, mean absolute amplitude, standard variance and kurtosis. Here we have selected kurtosis as a feature vector [25].

Kurtosis is a measure of how outlier-prone a distribution is. The kurtosis of the normal distribution is 3. Distributions that are more outlier-prone than the normal distribution have kurtosis greater than 3; distributions that are less outlier-prone have kurtosis less than 3. Kurtosis is a 4^{th} order statistics.

The kurtosis of a distribution is defined as:

$$K = \frac{E(y - \bar{y})^4}{\sigma^4} \tag{4}$$

where, $\bar{y}$ is the mean of y, σ is the standard deviation of y, and E (t) represents the expected value of the quantity t.

These parameters are mainly used as features for the classification of motor imagery [6]. The above mentioned features were selected as a BCI needs real time computation and the features should not take long time to calculate and hence it is preferred to extract the time domain parameters.

Separate functions were created for implementing these features. As this is a time series function a sliding window of 160 samples were created and was incremented by 1 sample till the end of signal. But on implementing this sliding window the number of features extracted were in large quantities, but as we have discussed if the feature extraction process is not done properly, then the classification accuracy reduces drastically. This was one of my observations.

Hence another sliding window of 672 samples were taken and the number of windows were calculated. The calculated value was around 30, which corresponds to the total number of annotations of the signal.

3.4 Classification

Patterns of brain activity are considered to be dynamic random processes due to physiological and technical factors. In terms of the physiology, the patterns undergo a change when the individual does not concentrate, due to fatigue, due to disease progression, and during the long process of training, as training takes time and becomes boring after following the same routine again and again. In technical terms, the patterns undergo change as an effect of noises from the amplifier, background noises, and also the change in the electrode impedances. Therefore, time course of the generated time series signals (e.g. EEG) should be taken into consideration during the feature extraction process. Hence, to use this temporal information, three different approaches have been proposed and used:

1. Concatenation of features from different time segments: extracting features from several time segments and concatenating them into a single feature vector.
2. Combination of classifications at different time segments: it consists in performing the feature extraction and classification steps on several time segments and then combining the results of the different classifiers.
3. Dynamic classification: it consists of extracting features from several time segments to build a temporal sequence of feature vectors. This sequence can be classified using a dynamic classifier [6].

For this thesis the first approach was used for calculating the feature vector. The different features were extracted according to 2 experimental setups:

1. Sliding window of 160 samples incremented by 1 sample,
2. Sliding window of 672 samples incremented by 656 samples.

The different types of classifiers used are, linear discriminant classifier, support vector machines, neural networks, bayesian classifiers and nearest neighbour classifiers. These are just some of the major classifiers used.

The 3 terms described are the 3 possible sources of classification errors which are inherent:

1. Noise: represents the irreducible noise within the system;
2. Bias: represents a systematic error which is the divergence between the estimated mapping (i.e. the estimated class label) and the best mapping (i.e. the true class label). This term depends strongly on the classification method that has been chosen to obtain *f* (linear, quadratic).
3. Variance: reflects the sensitivity to the used training set T.

Linear Classifiers have a high bias but low variance. In this study we are trying to implement two linear classifiers:

1. Linear Discriminant classifier,
2. Support Vector Machine.

3.4.1 Linear discriminant classifier

Linear discriminant analysis (LDA) classifier is used because it is one of the most effective linear classification methods for BCI and because it is also used by BCI2000 software on the testing paradigm.

The basic idea for implementation of LDA is simple. A linear function of attributes is computed for each identified class. The class function with the highest score represents the predicted class [26]. LDA has low complexity. They are classified under stable classifiers as small variations in the training set do not considerably affect their performance. LDA has been successfully used in many motor imagery based BCIs. Its computational requirement is very low and it is simple to use. This classifier generally provides good results.

The idea of LDA is to find a weight vector *w* so that two projected clusters *c1* and *c2* of *N1* and *N2* training feature vectors *x1* and *x2* on *w* can be well separated from each other by hyperplanes while keeping small variance of each cluster.

In the case of multiclass separation problem, several hyperplanes are used. The strategy generally used in this case is the one versus the rest which separate each class from all the others. This technique is suitable for the on-line BCIs because it has a very low computational requirement. It is simple to use and generally provides good results.

The main drawback of LDA is its linearity that can provide poor results on complex nonlinear EEG data [6].

LDA was implemented in Matlab using the *fitcdiscr( )* function. The syntax of the function is:

```
obj = fitcdiscr(x,y)

x - Predictor Values

y - Classification values
```

3.4.2 Support Vector Machine

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Support vector machines were originally designed for binary classification. The main idea of this approach is to construct a hyperplane in order to separate two classes ($ci \in \{-1, 1\}$ or $ci \in \{0, 1\}$) so that the margin (the distance between the hyperplane and the nearest point(s) of each classes) is maximal and with minimal error.

For implementing SVM, there is an inbuilt tool in Matlab called the "*classification learner*". This tool can be put to good use for classification. This tool learns an algorithm by itself depending on the given input, i.e. the predictor values and the classification values

3.4.3 Classification performance measures

For understanding the classification results clearly, some classification performance measures have to be calculated. The different classification performance measures usually used are accuracy, sensitivity, specificity, precision, recall, f1 score and gmean. All of these are statistical descriptors which are calculated using the positive and negative classification calculated. With that data we can calculate the true positives and the true negatives.

The accuracy is calculated using the formula

$$Accuracy = \frac{(Tp + Tn)}{N} \tag{5}$$

Where *Tp* is the true positive, *Tn* is the true negative and *N* total number of segments.

The Specificity also known as the true negative rate measures the correctly identified proportion of negatives. It is calculated using the formula

$$Specificity = \frac{Tn}{n} \tag{5}$$

Where *Tn* is the true negative and *n* is the length if negatives

4 Results

Different results were observed according to the set goals. The dataset was taken and bandpass filtered as mentioned in section 3.1. This constitutes the signal pre-processing. Then the features were extracted and then classified using the LDA classifier. These results are not efficient but promising as a very mild pre-processing is done. The accuracy of the classification is found to be around 60-65% which is dependent on the type of features extracted and the channels taken into consideration.

4.1 Pre-processing results

A normal bandpass filtering is done as discussed previously. The filter used was FIR bandpass filter. The order selected was 6 and 10. This was done in order to observe how the classifier reacts to the different levels of filtering. The filtering was done in the frequency band 13-31 Hz. The filtered data was plotted (see Figure 7) for the 21 selected channels as we discussed in section 2.1

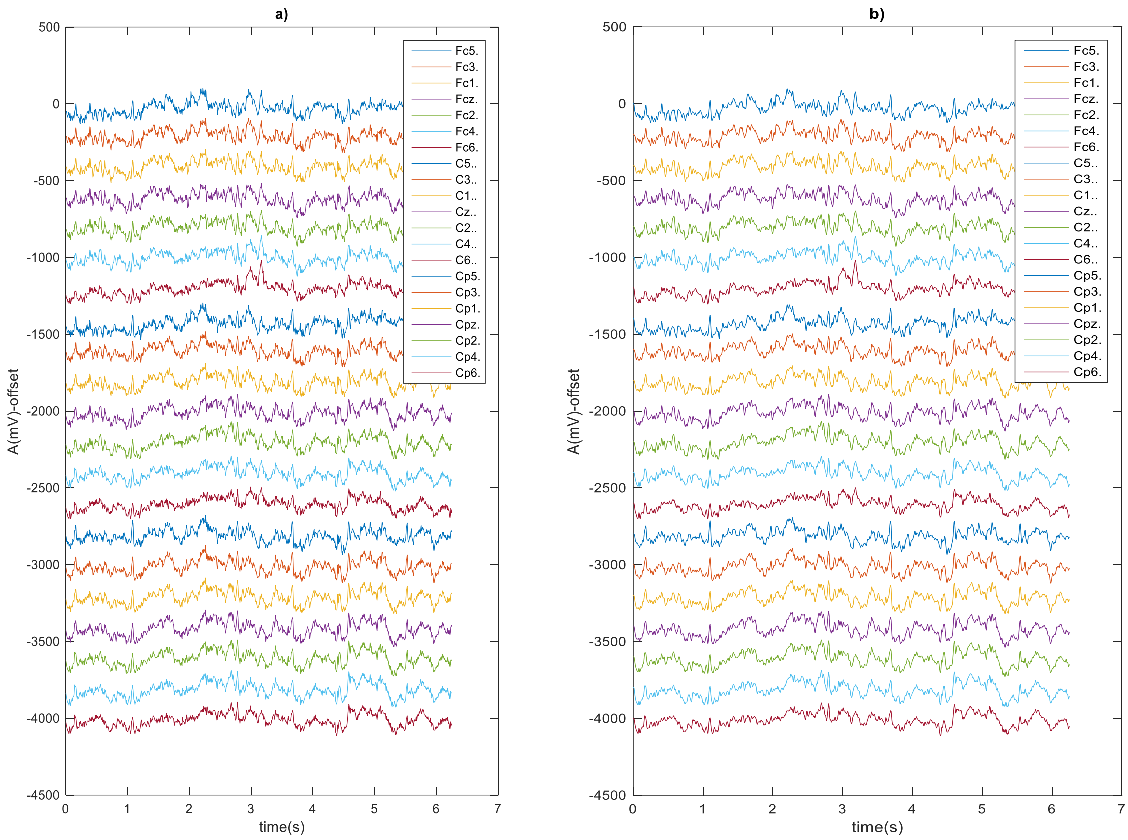

Figure 7 Illustration of original EEG data (a) and filtered EEG data, order 6 (b) for 21 channels

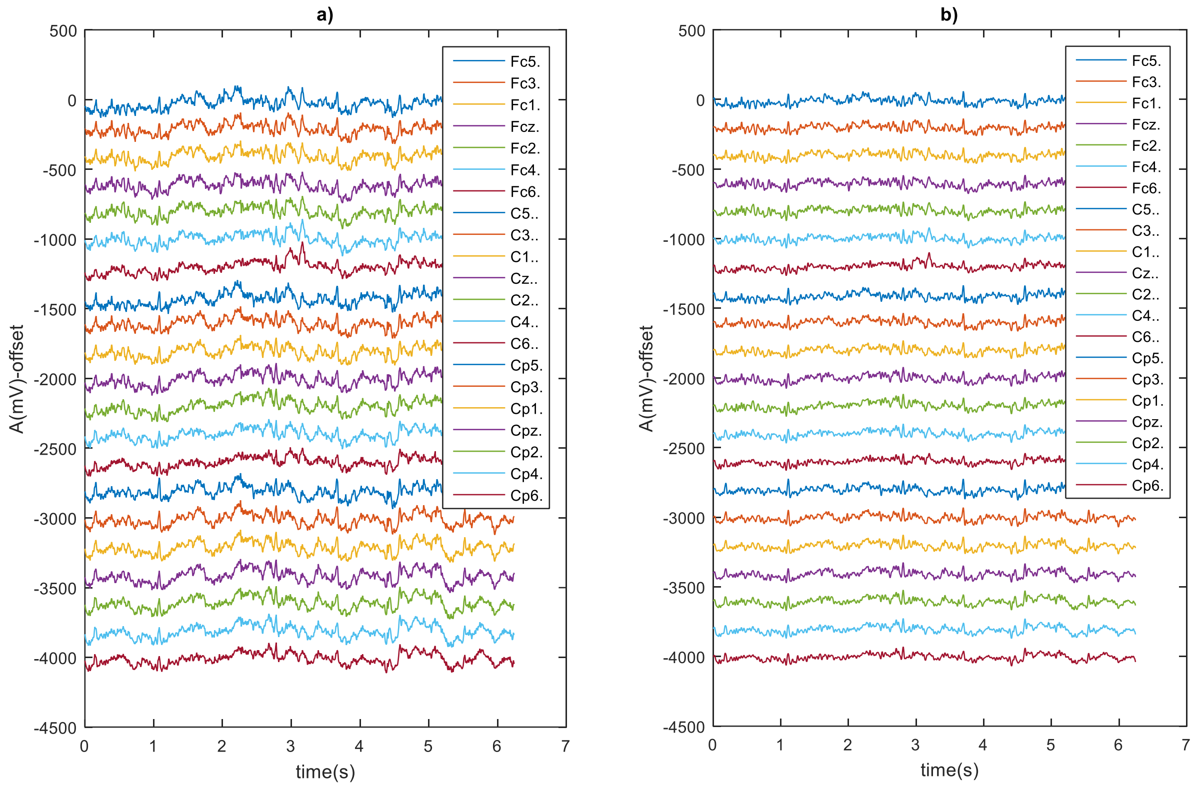

Figure 8 Illustration of original EEG data (a) and filtered EEG data, order 10 (b) for 21 channels

As seen in Figure 8, it is clear that on using a higher order filter, most of the noise gets filtered and there is a significant change from the original signal. But on doing so, there is a huge probability of losing valuable data. On using this filter order, the classification accuracy is observed to be lesser which is discussed in Table 1.

In-spite of the filtering, the EEG data looks complex due to the high complexity and dimensionality of the EEG signals. There seems to be no much difference in the signals as a simple bandpass filter was used for the pre-processing of the data. Hence we have a statistically less expensive computed data and it can be effectively used for real time signal processing and feature extraction and the further processing will be simpler and faster, not efficient but faster.

4.2 Feature extraction results

The different features were extracted according to 2 experimental setups:

1. Sliding window of 160 samples incremented by 1 sample(see Figure 9)
2. Sliding window of 672 samples incremented by 656 samples(see Figure 10)

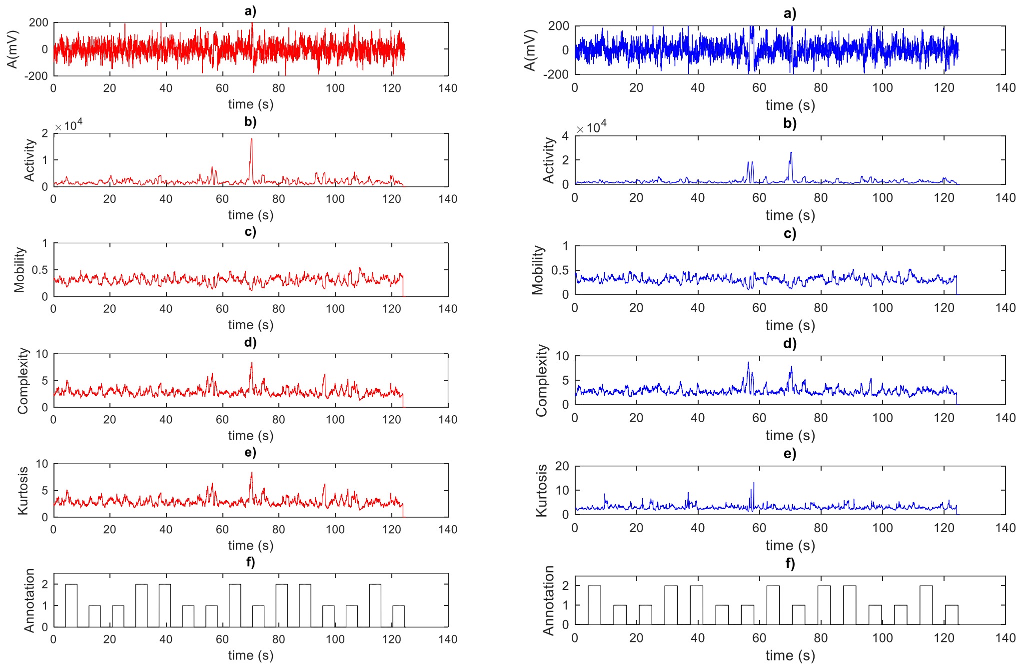

Figure 9 Illustration of extracted features for real motor movement of 1 subject – C3 and C4 channel: filtered signal (a), activity feature (b), mobility feature (c), complexity feature (d), annotation data 0-rest, 1-left, 2-right.

From Figure 9 it can be seen that the calculated features are of the same length of the total signal, i.e. it has same number of features as that of the signal. To be specific, 20000 features were extracted for activity, mobility, complexity and kurtosis. Hence the feature vector becomes very huge. As it was discussed in the section 3.3, one has to be careful while extracting the features as it may lead to over fitting the classifier and hence unexpected errors may be encountered. Even in this case it is the same. And this observation is seen in the next section. The classification accuracy for this particular setup reduces drastically.

It is very difficult to represent these extracted features in scatter plot as there are lot of features. Hence it is difficult to analyse these features in terms of simple plots. Due to the size of these extracted features, it takes a lot of time to compute the feature vector. So it is not advisable to use this for a real time BCI.

The above experiment helped in analysing the feature extraction procedure in-depth and helped me figure out the new algorithm in which only 30 features are extracted.

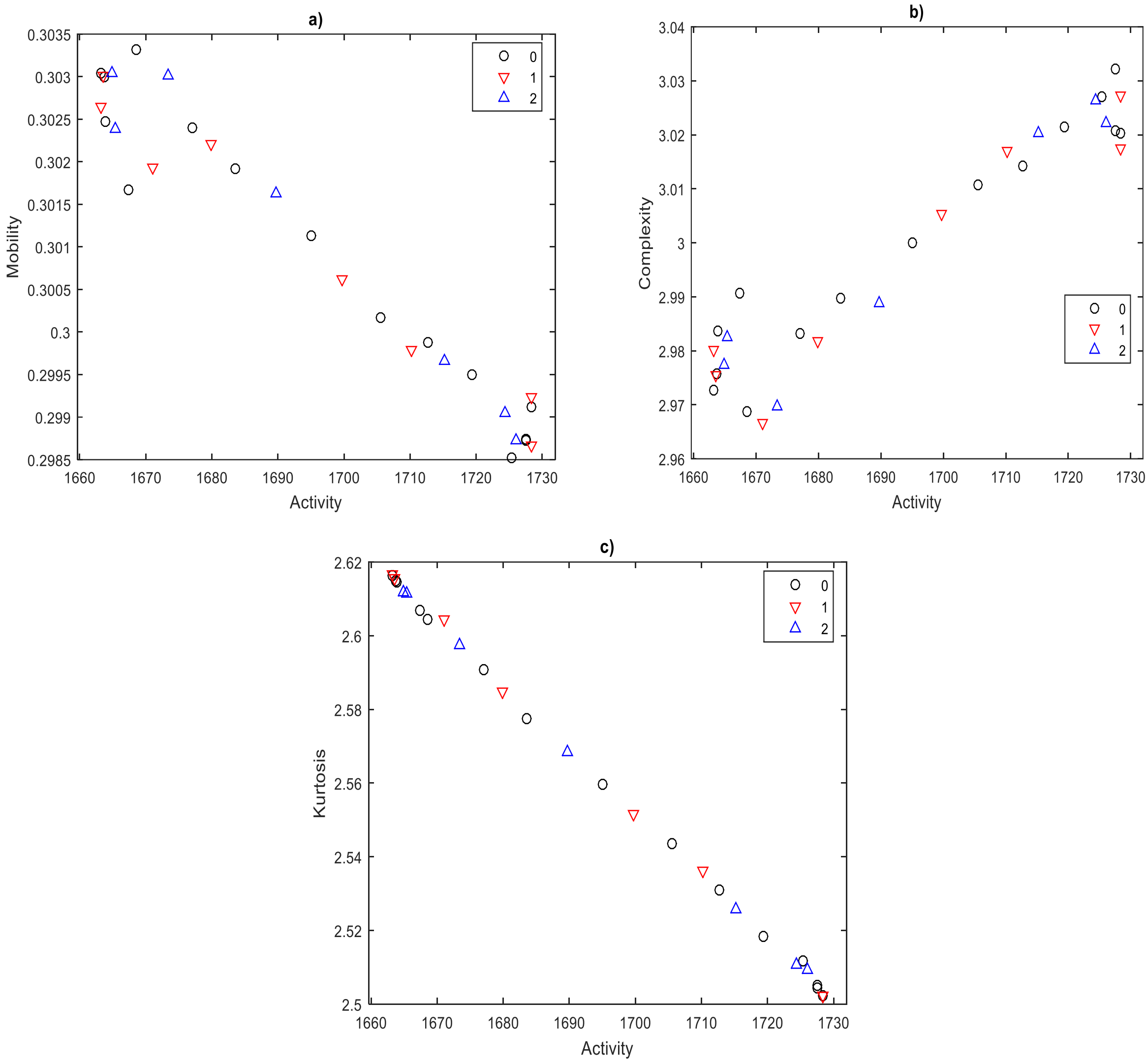

Figure 10 Scatter plots of extracted features for real motor movement of 1 subject – C3 channel: activity feature vs mobility feature (a), activity feature vs complexity feature (b), activity feature vs kurtosis feature (c) annotation data 0-rest, 1-left, 2-right

In Figure 10, individual features extracted using the second setup is plotted. Here the data looks much simpler and instead of the huge 20000 features in the previous setup, here we have just 30 features for each of the 4 feature vectors. This corresponds to the 30 windows used for extracting the features. This brings about a huge change in the size of the data and as it is seen in the next section, this setup gives us a better classification accuracy than that of the previous setup.

4.3 Classification results

In this work, the main aim was to prove that it is possible to control bionic arm with imaginary data of motor movement. In order to achieve and understand this aim 2 types of experiments were conducted and their accuracy of classification was observed for both the sliding windows:

1. Training and testing with real data,
2. Training and testing with imaginary data.

Both these steps were done for the data received after the feature extraction process.

4.3.1 LDA classifier

The LDA classifier was used in order to classify the extracted features and output was observed. The test were done on 21 electrodes and in the plots electrodes are in the order:

'Fc5' 'Fc3' 'Fc1' 'Fcz' 'Fc2' 'Fc4' 'Fc6' 'C5' 'C3' 'C1' 'Cz' 'C2' 'C4' 'C6' 'Cp5' 'Cp3' 'Cp1' 'Cpz' 'Cp2' 'Cp4' 'Cp6'

Classification accuracy is calculated using the formula mentioned in section 3.4. It is one of the statistical descriptors which is used for a better understanding of the classified features. It shows the relation between the actual and the predicted classes, hence we can find out how well the applied algorithm is effective.

The mean accuracies for all the 21 channels of all the 10 subjects is shown in the Table 1. From this table we can see that order 6 of the FIR bandpass filter gives better accuracy than that of order 10. Hence it was decided to use order 6 filter for all the calculations.

Accuracy was calculated for a single subject (subject 1) for both the experimental setups mentioned in the feature extraction process. Upon comparison it was found that the second experimental setup had better accuracy than that of the first. Hence from this particular comparison it was decided to go forward and do the detailed calculations with the second experimental setup where a sliding window of 4.2 seconds was moved by 4.1 seconds. The mean accuracy for the first setup was found to be close to 50% and for the second setup was found to be close to 60%. The Figure 11 shown below, shows the comparison of both setups for a single subject.

Table 1 Mean for 21 channels of 10 subjects-both real and imaginary – FIR order 6 and 10

	Order 6		**Order 10**	
	Real	**Imaginary**	**Real**	**Imaginary**
Channel	**Mean**	**Mean**	**Mean**	**Mean**
Fc5	57.33	64.67	59.33	61.33
Fc3	54.67	60.00	54.00	57.33
Fc1	56.67	58.67	56.00	56.67
Fcz	59.33	57.33	60.00	59.33
Fc2	62	58.67	60.00	55.33
Fc4	60.67	60.00	56.67	56.67
Fc6	61.33	61.33	54.00	61.33
C5	58	59.33	56.67	58.67
C3	61.33	60.00	54.67	57.33
C1	60.67	60.00	52.67	59.33
Cz	58	56.67	55.33	60.67
C2	56.67	60.00	58.67	58.00
C4	56.67	58.00	54.67	56.00
C6	55.33	56.67	52.67	58.00
Cp5	57.33	60.00	56.00	57.33
Cp3	60	59.33	61.33	57.33
Cp1	60	58.00	56.67	58.00
Cpz	60	60.67	60.00	59.33
Cp2	60	58.67	55.33	57.33
Cp4	58.67	57.33	59.33	56.67
Cp6	57.33	58.67	56.00	60.00

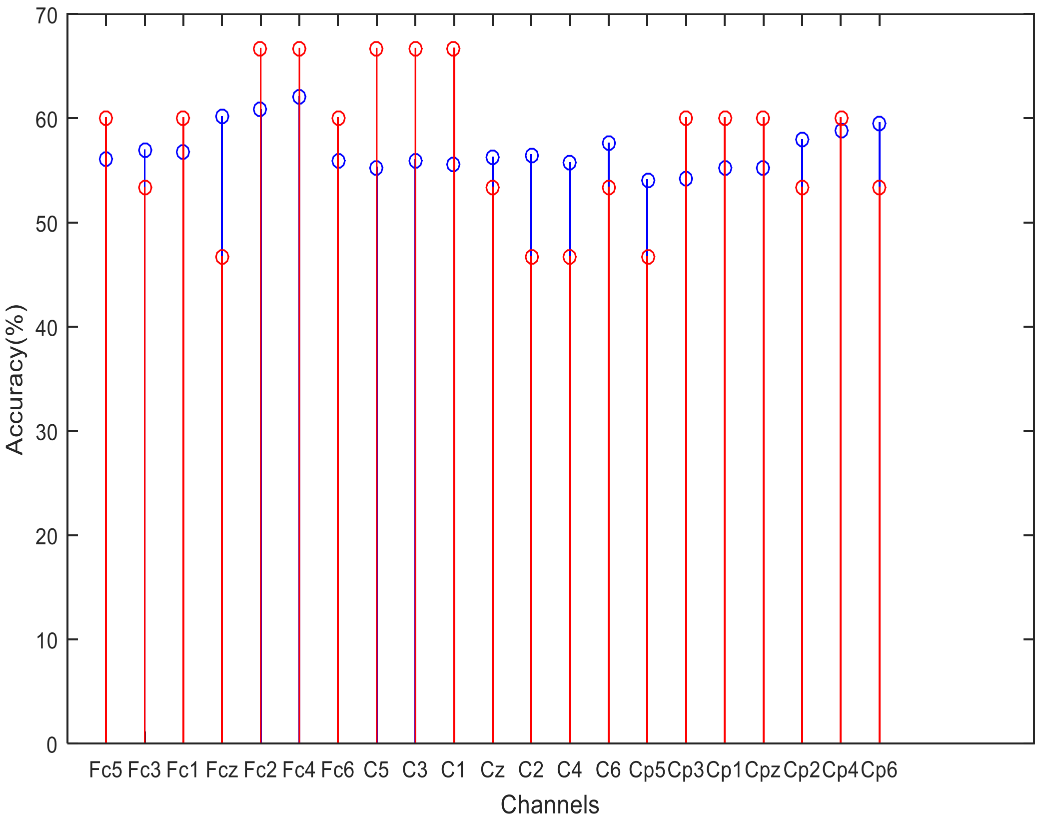

Figure 11 Comparison of accuracy for 21 electrodes according to the feature extraction setups: blue (sliding window size 160 samples), red (sliding window size 672 samples),

Next the classification accuracy for 2 subjects were plotted and observed. This was done in order to understand the trend and how effective the classifier is. From the Figure 12 we can see that the imaginary data shows close to 90% efficiency in one of the electrodes and some other electrodes show close to 80% which is a very good classification value and for the real data the accuracy for most of the electrodes is from 60-70%.

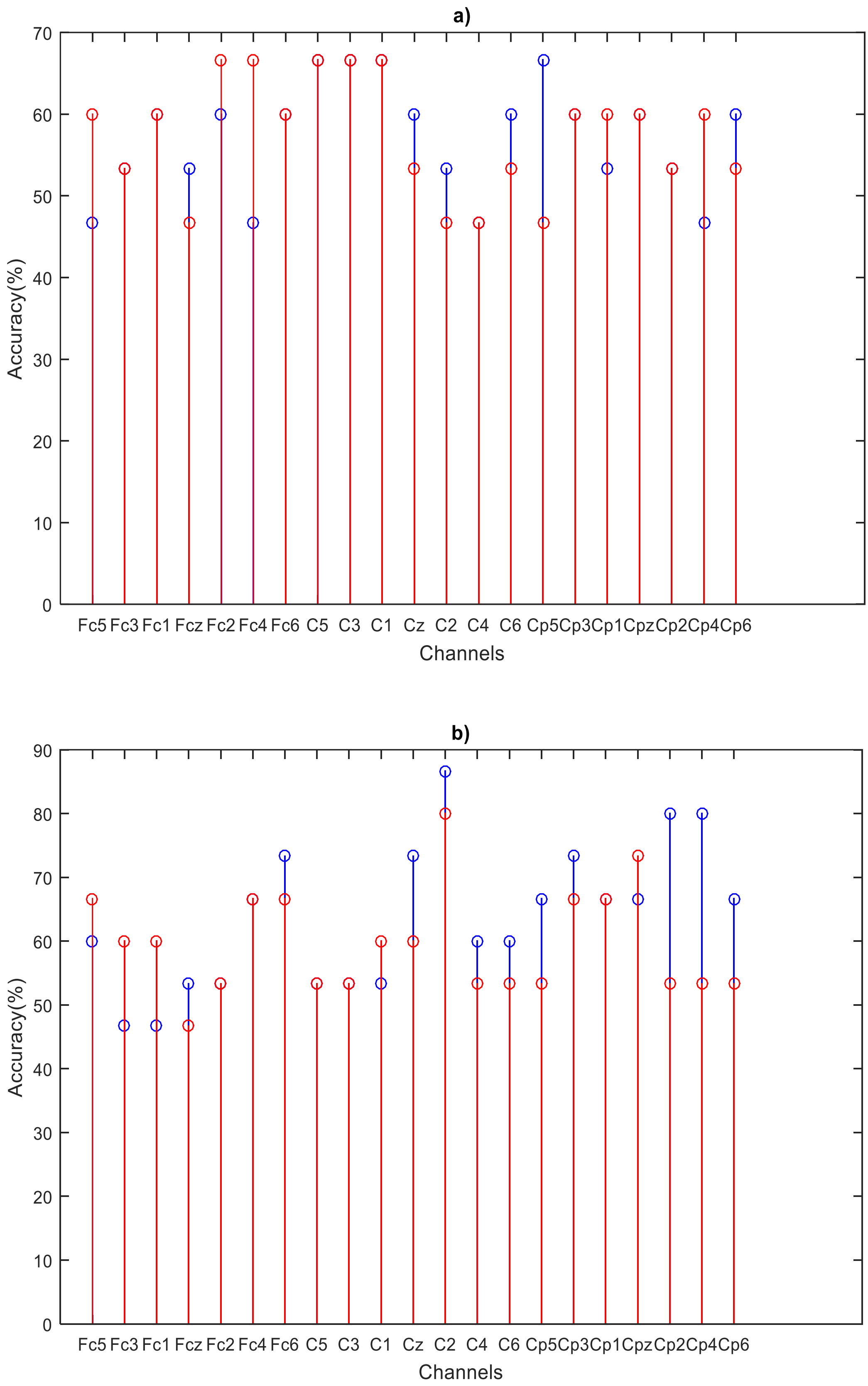

Figure 12 Classification accuracy of 2 subjects for 21 electrodes, real motion (a), imagined motion (b)

Table 2 Mean and standard deviation for 21 channels of 10 subjects-both real and imaginary

	Real		**Imaginary**	
Channel	**Mean**	**S.D**	**Mean**	**S.D**
Fc5	57,33	8,43	64,67	8,34
Fc3	54,67	5,26	60,00	9,94
Fc1	56,67	5,67	58,67	9,32
Fcz	59,33	11,1	57,33	7,83
Fc2	62	9,45	58,67	6,13
Fc4	60,67	9,14	60,00	8,89
Fc6	61,33	6,13	61,33	7,57
C5	58	7,73	59,33	7,34
C3	61,33	4,22	60,00	6,29
C1	60,67	6,63	60,00	6,29
Cz	58	5,49	56,67	9,03
C2	56,67	8,46	60,00	13,33
C4	56,67	11	58,00	6,32
C6	55,33	7,06	56,67	6,48
Cp5	57,33	7,83	60,00	8,31
Cp3	60	9,43	59,33	7,98
Cp1	60	7,7	58,00	7,06
Cpz	60	4,44	60,67	9,14
Cp2	60	9,94	58,67	9,84
Cp4	58,67	9,84	57,33	10,04
Cp6	57,33	9	58,67	5,26

Table 3 Accuracies of the 10 subjects with the mean of all the 21 electrodes.

	Accuracy	
Subjects	**Real**	**Imaginary**
1	57.46	59.68
2	57.14	63.81
3	58.09	64.13
4	59.36	57.14
5	61.26	59.05
6	60.95	54.60
7	62.54	61.90
8	53.01	55.87
9	63.81	59.68
10	53.01	56.51

Based on the accuracies of the individual subjects for real data, the highest accuracy of classification is observed in the 9th subject whereas the 8th and the 10th subject show the least classification accuracy. In accordance with this data, the mean accuracy of all the 10 subjects were plotted with the accuracy of the 9th subject in Figure 13

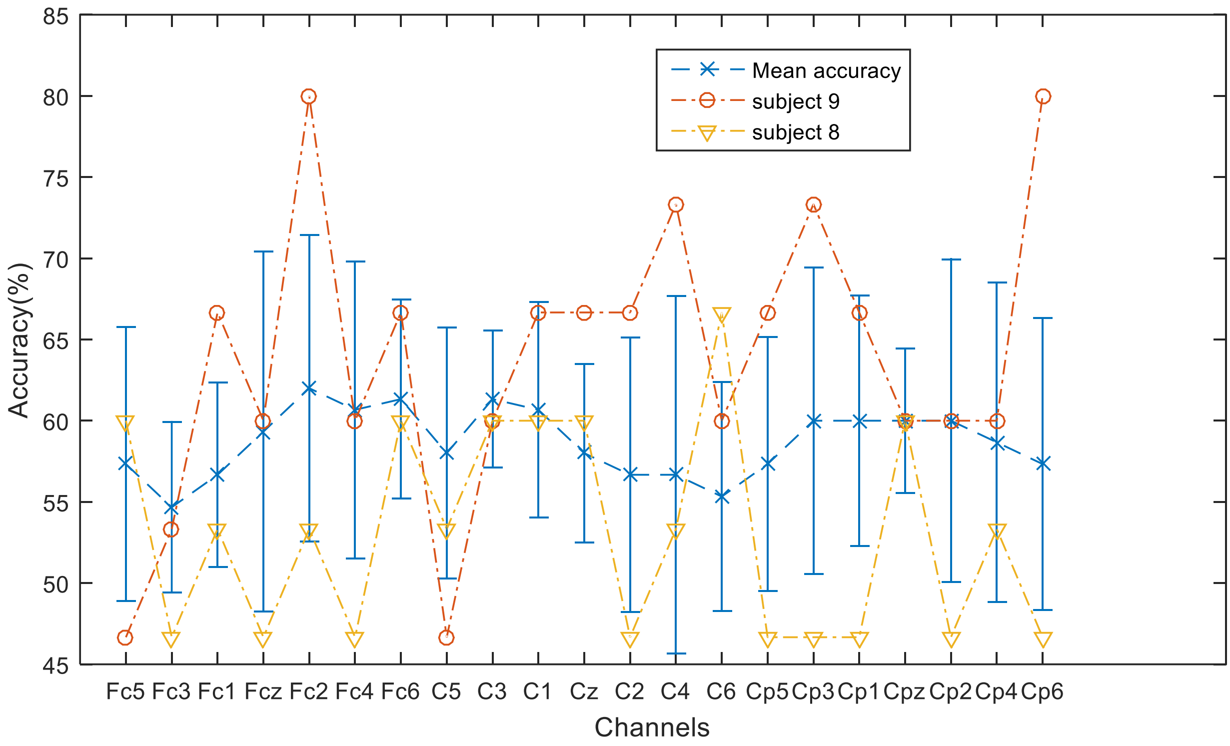

Figure 13 Mean classification accuracy and its standard deviation for real data of subject 9 and subject 8

It can be observed from the above figure that the subject 9 gives us accuracy over 80 % in some channels which is a very promising result for such a simple algorithm. We can also see that all the subjects have their lowest accuracy as 46.67% in some channels. This also shows that most of the electrodes of this subject have the highest accuracy close to the standard deviation. Hence we can assume that the trial of the subject 9 was more successful than that of subject 8 which has very less classification accuracy.

Next on the basis of the accuracies of the individual subjects for imaginary data, the highest accuracy of classification is observed in the 3rd subject whereas the 9th subject shows the least classification accuracy. In accordance with this data, the standard deviation of the mentioned 2 subjects were plotted and observed in Figure 14.

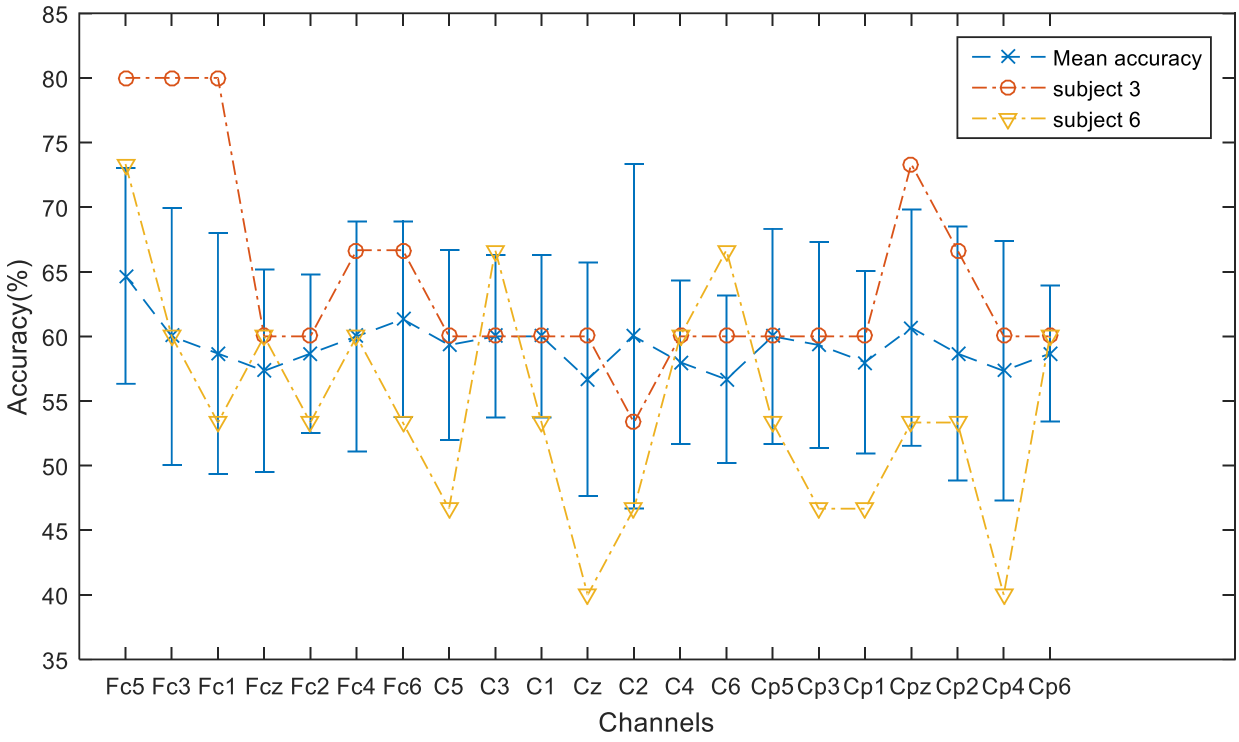

Figure 14 Mean classification accuracy and its standard deviation for imaginary data of 2 subjects

The accuracy for some of the channels of the imaginary data is around 80% for subject 3 and the lowest you can see is around 40% for subject 6. But then it is not bad for such a simple algorithm. In general, the imaginary data classifies better than the real data.

The lower accuracy we see for is single channel is not bad classification but just the lack of data in that particular electrode. In simple terms, when that particular movement happens or the imagination of the movement happens, the neurons in that area are not fired as extensively as other areas.

The classifier was also run with the 160 samples window of feature extraction. This was done only for one subject as the feature vectors generated are very huge and they are time consuming to execute. Hence it is not viable for a real time BCI system.

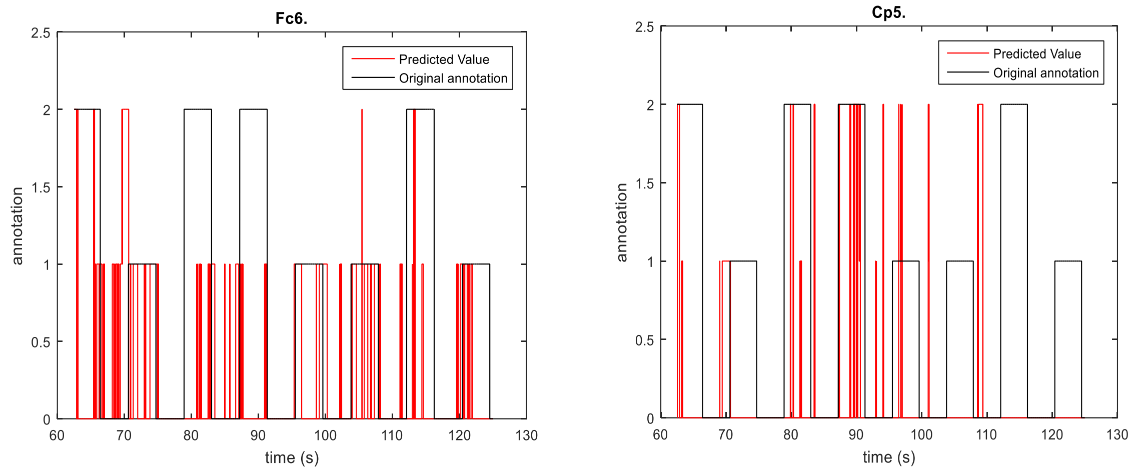

Figure 15 Illustration of predicted data from the LDA classifier for 2 channels Fc6 and Cp5. annotation data 0-rest, 1-left, 2-right

For the classification purpose the features were divided into 2. The 2 minute signal was divided into 1 minute each and the first half of the features were used to train the classifier and the second half were used for testing the classifier. Upon testing the results were found to be as shown in Figure 15. These 2 channels were selected to display as they have classified the respective annotation data, i.e. Fc6 corresponds to left hand (1) and Cp5 corresponds to right hand (2). Even though the classification is happening, the accuracy is very less. This is due to the factor that the feature vector has high resolution. The features were calculated for 1 second window. Hence the classifier classifies the rest state in between the left hand and right hand motor movement.

4.3.2 SVM classifier

The SVM classifier was tried in order to do a comparison study to LDA classifier. It was implemented using the classification learner tool in Matlab. For classifying using in the SVM classifier, the feature vector is loaded into the tool. The predictors and the responses are selected. Once that is done then the linear SVM classifier is run and the data is classified. But on implementing this classifier the rest states were being correctly classified and the left and right data were not classified as expected. This observation is shown in the Figure 16.

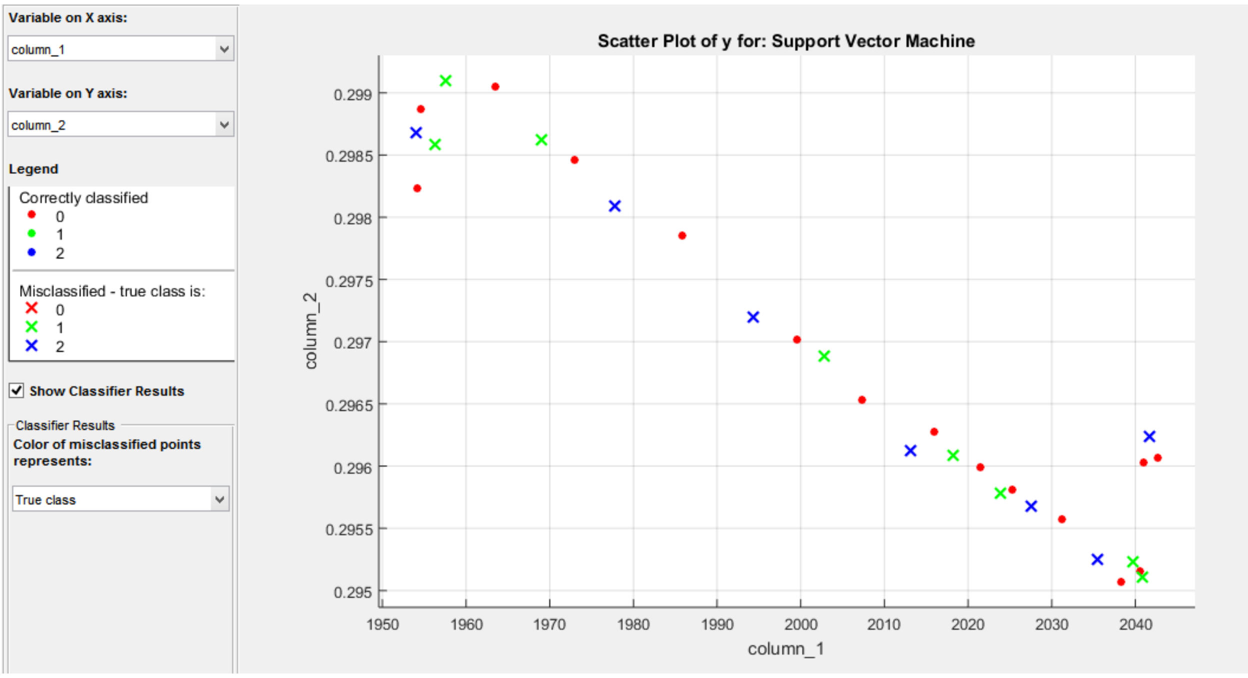

Figure 16 Scatted plot after classifying using SVM classifier

The features as shown in the scatter plot is the plot between activity and mobility. These are the 2 Hjorth parameters as discussed in section 3.3.1.

4.4 Selection of optimal EEG channels

21 channels in the sensorimotor cortex were selected as discusses in section 1.1. The data in these channels were observed to understand what would be the optimum channels. This was in order to generalize the channel selection to a fewer channels. But this is not a valid generalization as different subjects have different areas of the sensorimotor cortex activated. Many researches have been done on this front only to a fewer success. But even then there was an attempt made to analyse the optimum channels.

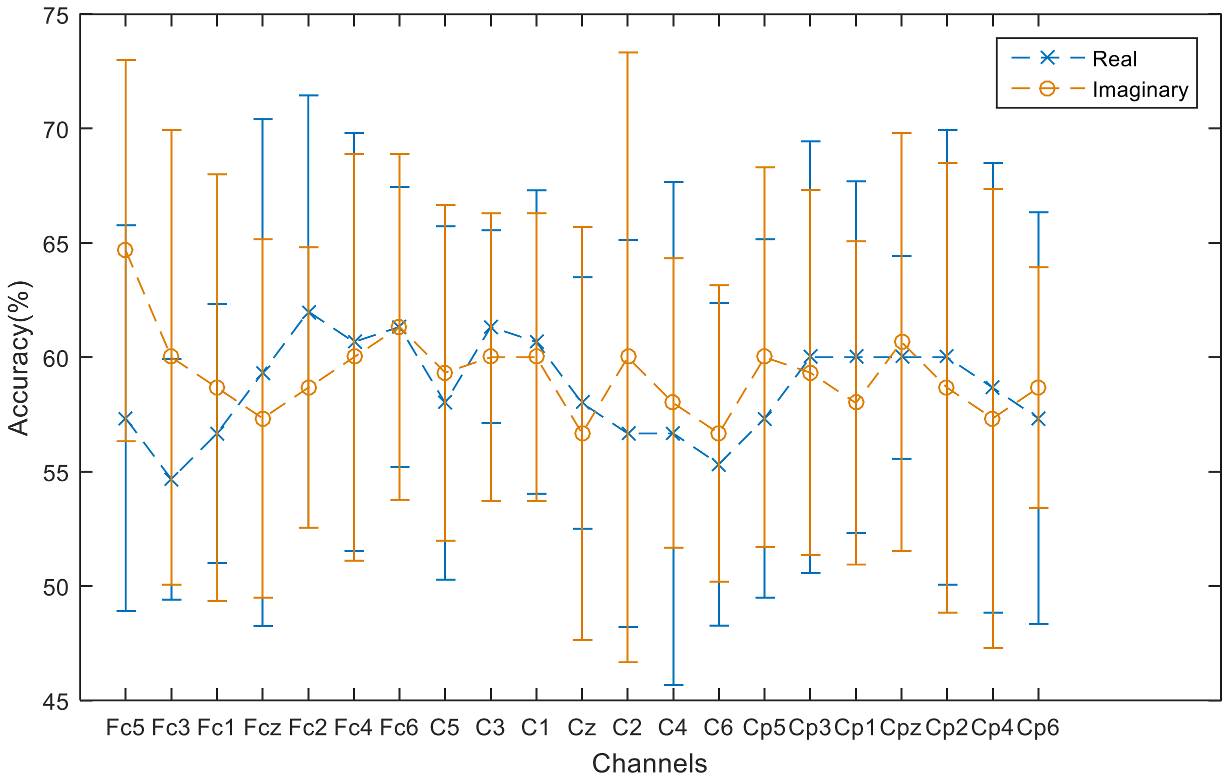

Figure 17 Mean accuracy and standard deviation for 10 subjects

The Table 2 shows the mean and standard deviation (SD) of accuracies of each channel for the 10 subjects for both real and imaginary data. The same data is plotted in Figure 17. From this we can see that the smallest standard deviation for the real data is in the C3 electrode and hence we can assume that this electrode classifies more consistently than the other electrodes. And for the imaginary data the electrode Cp6 seems to be more consistent over the set of 10 subjects. For the assumptions made, the laterality of the brain is not taken into consideration and if the sample size is increased the results may vary.

5 Conclusions and future work

Critical literature analysis showed that the signal processing and the machine learning for a BCI is very challenging. It helped to conclude that none of the researches have been able to create efficient BCI's. It also helped to understand that the EEG signals are very complex and multi-dimensional with crossing over of frequencies from different processes of human body. Hence any work done on this front need not produce any valid results.

The signal pre-processing done in this thesis is very basic and hence the accuracy of the classifications is affected adversely. But this had to be done as this is a design for a real time BCI. Hence mathematical calculations have to be reduced. This is the same reason for the selection of Hjorth parameters and kurtosis as the features. They are the time domain parameters and manipulation of the variance of the signal.

The signals were acquired from physionet and the pre-processing was done using a FIR bandpass filter of 6^{th} order and 10^{th} order. The filtered signal of the 6^{th} order bandpass filter was found to give more accuracy after classification. The feature extraction procedure was implemented successfully and the output was observed. Activity, Mobility, Complexity and Kurtosis were the extracted features. These features were extracted as they represent the time domain parameters and are very fast to compute. Hence it is very beneficial for real-time BCIs.

The signal was classified using LDA which is a linear classifier. By using LDA classifier, it was observed that the output accuracy of classification was observed in the range of 60 to 70% depending on the electrode selected, the mean values and standard deviations were plotted successfully.

A lot of research in this area have come to the conclusion that adding irrelevant features have the possibility of degrading the performance of classifiers. The same goes with the case if the number of training samples is small/large relative to the number of features. These problems can be avoided by selecting relevant features (i.e., feature selection) or extracting new features containing maximal information about the class label from the original ones. This difference was observed on adding the different sliding windows for feature extraction. It helped in reducing the losses during classification. Hence the feature extracting process has to be done with care.

The electrode selection criteria depends on the brain physiology. This is the reason why the 21 electrodes in the sensorimotor area were selected. Sensorimotor area of the brain is where all the movement related potentials are generated. Narrowing down the electrodes is impossible as different region of the sensorimotor cortex is activated for different individuals.

The laterality of the brain plays a huge role in the analysis of the data. The left hand motion is measured clearly in the right side of the brain and vice versa. But as we analyse the data in depth, we should be able to detect a part of the left hand movement on the left side of the brain. This particular area in EEG signal analysis is just budding and can help a lot of people affected by stroke.

The future work on this thematic can be related to the modification of the training samples and the testing samples for the classifier. It can be done by increasing and decreasing the feature samples. This should be done as there is a lot of rest pattern apart from the right and the left hand motion in the database selected. Hence more data is classified as rest. So if this can be figured out then the classification accuracy can be increased by a huge margin. At present the classification accuracy for the sliding window of 160 samples comes around 50%-55% which is neither good nor efficient. But this accuracy is only because of the high resolution of extracted features. The second sliding window of 672 samples gives a better accuracy of around 60-70% depending on the electrodes.

Finally it can be concluded from the experiment that the imaginary data does not require any prior training with the real data. This was observed when the imaginary data was classified and gave a better classification accuracy than the real data. It means that if a person with a missing limb is made to think that he has a limb and is able to move it, then it would generate the motor imagery pertaining to the imagined motion in the sensorimotor area of the brain.

6 References

[1] N.-A. LF and G.-G. J, "Brain Computer Interfaces, a Review," *Sensors (Basel),* vol. 12, no. 2, p. 1211–1279., 2012.

[2] N.-J. Huan and R. Palaniappan, "BCI design using mental task classification," pp. 1-9.

[3] G. Dornhege, J. D. R. Millan, T. Hinterberger, D. McFarland and and K.-R.M¨uller, "Toward Brain-Computer Interfacing," 2007..

[4] R. Scherer, F. Lee, A. Schlogl, R. Leeb, H. Bischof and G. Pfurtscheller, "Towards self-paced (asynchronous) Brain-Computer Communication: Navigation through virtual worlds," *IEEE,* pp. 1-9.

[5] N. Tomida, T. Tanaka, S. Ono, M. Yamagishi and H. Higashi, "Active Data Selection for Motor Imagery," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING,* vol. 62, no. 2, p. 9, 2015.

[6] T. Al-ani and D. Trad, "Signal Processing and Classification Approaches for Brain-computer Interface," in *Intelligent and Biosensors*, 2010, pp. 25-66.

[7] X. Wang, A. Wang, S. Zheng, Yingzi Lin 2 and M. Yu, "A Multiple Autocorrelation Analysis Method for Motor Imagery EEG Feature Extraction," *IEEE,* pp. 3000-3004, 2014.

[8] C. Vidaurre, Nicole Krämerc, B. Blankertz and A. Schlögl, "Time Domain Parameters as a feature for EEG-based Brain Computer Interfaces," *Elsevier,* vol. 22, no. Special Issue, p. 6, 2009.

[9] J. Little and C. Moler, " The MathWorks, Inc.," Matlab, 1994-2016 . [Online]. Available: http://se.mathworks.com/index.html?s_tid=gn_logo. [Accessed 2014; 2015; 2016].

[10] Colm Seale and P. E. Jones, "Real-Time Processing of EEG Signals," 2012.

[11] T. Hines, "Mayfield Brain and Spine," Mayfield Clinic, April 2014-2016. [Online]. Available: http://www.mayfieldclinic.com/PE-AnatBrain.htm. [Accessed December 2015].

[12] Tauseo, "Texila American University," [Online]. Available: http://blog.tauedu.org/anatomy-and-functional-areas-of-the-brain/. [Accessed 22 May 2016].

[13] T. Wang, J. Deng and B. He, "Classifying EEG-based motor imagery tasks by means of time–frequency synthesized spatial patterns," *Elsevier,* vol. 115, p. 2744–2753, 2004.

[14] D. T. Bundy, M. Wronkiewicz, M. Sharma, D. W. Moran, M. Corbetta and E. C. Leuthardt, "Using ipsilateral motor signals in the unaffected cerebral hemisphere as a signal platform for brain–computer interfaces in hemiplegic stroke survivors," *Journal of Neural Engineering,* vol. 9, no. 3, pp. 1-12, 2012.

[15] Y. Holler, J. Bergmann, A. Thomschewski, M. Kronbichler, P. Holler, J. S. Crone, E. V. Schmid, K. Butz, R. Nardone and E. Trinka, "Comparison of EEG-Features and Classification Methods for Motor Imagery in Patients with Disorders of Consciousness," *PLOS ONE,* no. 11, p. 15, 2013.

[16] F. Cincotti, F. Aloise, S. Bufalari, G. Schalk, G. Oriolo, A. Cherubini, F. Davide, F. Babiloni, M. G. Marciani and D. Mattia, "Non-Invasive Brain-Computer Interface System to Operate Assistive Devices," *IEEE,* pp. 2532-2535, 2007.

[17] D. J. McFarland, L. A. Miner, T. M. Vaughan and J. R. Wolpaw, "Mu and Beta Rhythm Topographies During Motor Imagery and Actual Movements," *Brain Topography,* vol. 12, pp. 177-186, 2000.

[18] A. Bashashati, M. Fatourechi, R. K. Ward and G. E. Birch, "A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals," *IOP Publishing,* no. 4, pp. 32-57, 2007.

[19] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K and S. HE, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals.," *Circulation Electronic Pages,* vol. 23, no. Circulation 101, p. 5, 2000 (June 13)..

[20] G. Schalk, D. McFarland, T. Hinterberger, N. Birbaumer and J. Wolpaw, "BCI2000: A General-Purpose Brain-Computer Interface (BCI) System," *IEEE Transactions on Biomedical Engineering,* vol. 51, no. 6, pp. 1034-1043, 2004 [In 2008, this paper received the Best Paper Award from IEEE TBME.].

[21] F. Lotte, "Study of Electroencephalographic Signal Processing and Classification Techniques towards the use of Brain-Computer Interfaces in Virtual Reality Applications.," https://tel.archives-ouvertes.fr/tel-00356346v2, INSA de Rennes,, 2008.

[22] R. VanRullen, "Four common conceptual fallacies in mapping the time course of recognition," *Front. Psychol,* vol. 2, no. 365, pp. 1-6, 2011.

[23] K. Matthias, M. Peter, G. Ulf, L. Thomas and R. Helge, "BCI competition 2003-data set IIb: support vector machines for the P300 speller paradigm IEEE Trans Biomed Eng," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING,* vol. 51, no. 6, pp. 1073-1076, 2004.

[24] E. R. Miranda and A. Brouse, "Interfacing the Brain Directly with Musical Systems," *LEONARDO,* vol. 38, no. 4, pp. 331-336, 2005.

[25] L. Jing, Y. Li and H. Zhang, "A New Improved Feature Extraction Method in Memory EEG Data," *Joint International Mechanical, Electronic and Information Technology Conference,* pp. 589-598, 2015.

[26] R. Aldea and M. Fira, "Classifications of Motor Imagery Tasks in BCI using LDA," *IJARAI,* vol. 3, no. 7, pp. 5-9, 2014.

[27] José del R. Millán, Josep Mouriño, I. Student Member, Marco Franzé, Febo Cincotti, Markus Varsta, Jukka Heikkonen and a. F. Babiloni, "A Local Neural Classifier for the Recognition of EEG Patterns Associated to Mental Tasks," *IEEE,* pp. 678-686, 2002.

[28] F. O. Erfanian and a. Abbas, "A Tree-Structure Mutual Information-Based Feature Extraction and Its Application to EEG-Based Brain-Computer Interfacing," *IEEE EMBS,* pp. 5075-5078, 2007..

[29] J. del R. Millha, J. Mouriiioa, M.G. Marcianib, F. Babilonib, F. Topan, I. Canale', J. Heikkonend and K. Kaskid, "Adaptive Brain Interfaces for Physically-Disabled People," *IEEE,* 1998.

[30] M. I. Dean J. Krusienski, Dennis J. McFarland and a. J. R. Wolpaw, "An Evaluation of Autoregressive Spectral Estimation Model Order for Brain-Computer Interface Applications," *IEEE,* pp. 1323-1326, 2006.

[31] J. R.Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller and and T. M.Vaughan, "Brain–computer interfaces for communication and control," *Clin. Neurophysiol,* Vols. 113,, no. 6, p. 767–791, 2002..

Annexe - Matlab Code

```
clc;
close all;
clear all;

%%
% READ Multiple User data

%nk='C:\Users\User\Documents\Aravind
Prasad_Thesis\Dropbox\Magistrantura_AravindPrasad\Thesis\Physionet\Real_Imagi
narySignals\';
nk='C:\Users\Unni\Dropbox\Magistrantura_AravindPrasad\Thesis\Physionet\Real_I
maginarySignals\';

%S001
[dr1,hr1] = ReadEDF([nk,'S001R03.edf']);%Task 1 (open and close left or right
fist)
[di1,hi1] = ReadEDF([nk,'S001R04.edf']);%Task 2 (imagine opening and closing
left or right fist)
%S002
[dr2,hr2] = ReadEDF([nk,'S002R03.edf']);%Task 1 (open and close left or right
fist)
[di2,hi2] = ReadEDF([nk,'S002R04.edf']);%Task 2 (imagine opening and closing
left or right fist)
%S003
[dr3,hr3] = ReadEDF([nk,'S003R03.edf']);%Task 1 (open and close left or right
fist)
[di3,hi3] = ReadEDF([nk,'S003R04.edf']);%Task 2 (imagine opening and closing
left or right fist)
%S004
[dr4,hr4] = ReadEDF([nk,'S004R03.edf']);%Task 1 (open and close left or right
fist)
[di4,hi4] = ReadEDF([nk,'S004R04.edf']);%Task 2 (imagine opening and closing
left or right fist)
%S005
[dr5,hr5] = ReadEDF([nk,'S005R03.edf']);%Task 1 (open and close left or right
fist)
[di5,hi5] = ReadEDF([nk,'S005R04.edf']);%Task 2 (imagine opening and closing
left or right fist)
%S006
[dr6,hr6] = ReadEDF([nk,'S006R03.edf']);%Task 1 (open and close left or right
fist)
[di6,hi6] = ReadEDF([nk,'S006R04.edf']);%Task 2 (imagine opening and closing
left or right fist)
%S007
[dr7,hr7] = ReadEDF([nk,'S007R03.edf']);%Task 1 (open and close left or right
fist)
[di7,hi7] = ReadEDF([nk,'S007R04.edf']);%Task 2 (imagine opening and closing
left or right fist)
%S008
[dr8,hr8] = ReadEDF([nk,'S008R03.edf']);%Task 1 (open and close left or right
fist)
[di8,hi8] = ReadEDF([nk,'S008R04.edf']);%Task 2 (imagine opening and closing
left or right fist)
%S009
[dr9,hr9] = ReadEDF([nk,'S009R03.edf']);%Task 1 (open and close left or right
fist)
[di9,hi9] = ReadEDF([nk,'S009R04.edf']);%Task 2 (imagine opening and closing
left or right fist)
%S0010
[dr10,hr10] = ReadEDF([nk,'S010R03.edf']);%Task 1 (open and close left or
right fist)
```

```
[di10,hi10] = ReadEDF([nk,'S010R04.edf']);%Task 2 (imagine opening and
closing left or right fist)



%%
% Filtering the Signal

[filtsig_dr1]=channeldata( dr1 );
[filtsig_di1]=channeldata( di1 );

[filtsig_dr2]=channeldata( dr2 );
[filtsig_di2]=channeldata( di2 );

[filtsig_dr3]=channeldata( dr3 );
[filtsig_di3]=channeldata( di3 );

[filtsig_dr4]=channeldata( dr4 );
[filtsig_di4]=channeldata( di4 );

[filtsig_dr5]=channeldata( dr5 );
[filtsig_di5]=channeldata( di5 );

[filtsig_dr6]=channeldata( dr6 );
[filtsig_di6]=channeldata( di6 );

[filtsig_dr7]=channeldata( dr7 );
[filtsig_di7]=channeldata( di7 );

[filtsig_dr8]=channeldata( dr8 );
[filtsig_di8]=channeldata( di8 );

[filtsig_dr9]=channeldata( dr9 );
[filtsig_di9]=channeldata( di9 );

[filtsig_dr10]=channeldata( dr10 );
[filtsig_di10]=channeldata( di10 );

%%
win_width=672;
slideincr = 656;
% win_number = ((length(data{1,1})-win_width)/slideincr);
win_number = 30;

[ A_dr1,M_dr1,C_dr1,K_dr1,amck_dr1 ]=feat(
win_number,win_width,filtsig_dr1,hr1.labels(1:21) );
[ A_di1,M_di1,C_di1,K_di1,amck_di1 ]=feat(
win_number,win_width,filtsig_di1,hi1.labels(1:21) );

[ A_dr2,M_dr2,C_dr2,K_dr2,amck_dr2 ]=feat(
win_number,win_width,filtsig_dr2,hr2.labels(1:21) );
[ A_di2,M_di2,C_di2,K_di2,amck_di2 ]=feat(
win_number,win_width,filtsig_di2,hi2.labels(1:21) );

[ A_dr3,M_dr3,C_dr3,K_dr3,amck_dr3 ]=feat(
win_number,win_width,filtsig_dr3,hr3.labels(1:21) );
[ A_di3,M_di3,C_di3,K_di3,amck_di3 ]=feat(
win_number,win_width,filtsig_di3,hi3.labels(1:21) );
```

```
[ A_dr4,M_dr4,C_dr4,K_dr4,amck_dr4 ]=feat(
win_number,win_width,filtsig_dr4,hr4.labels(1:21) );
[ A_di4,M_di4,C_di4,K_di4,amck_di4 ]=feat(
win_number,win_width,filtsig_di4,hi4.labels(1:21) );

[ A_dr5,M_dr5,C_dr5,K_dr5,amck_dr5 ]=feat(
win_number,win_width,filtsig_dr5,hr5.labels(1:21) );
[ A_di5,M_di5,C_di5,K_di5,amck_di5 ]=feat(
win_number,win_width,filtsig_di5,hi5.labels(1:21) );

[ A_dr6,M_dr6,C_dr6,K_dr6,amck_dr6 ]=feat(
win_number,win_width,filtsig_dr6,hr6.labels(1:21) );
[ A_di6,M_di6,C_di6,K_di6,amck_di6 ]=feat(
win_number,win_width,filtsig_di6,hi6.labels(1:21) );

[ A_dr7,M_dr7,C_dr7,K_dr7,amck_dr7 ]=feat(
win_number,win_width,filtsig_dr7,hr7.labels(1:21) );
[ A_di7,M_di7,C_di7,K_di7,amck_di7 ]=feat(
win_number,win_width,filtsig_di7,hi7.labels(1:21) );

[ A_dr8,M_dr8,C_dr8,K_dr8,amck_dr8 ]=feat(
win_number,win_width,filtsig_dr8,hr8.labels(1:21) );
[ A_di8,M_di8,C_di8,K_di8,amck_di8 ]=feat(
win_number,win_width,filtsig_di8,hi8.labels(1:21) );

[ A_dr9,M_dr9,C_dr9,K_dr9,amck_dr9 ]=feat(
win_number,win_width,filtsig_dr9,hr9.labels(1:21) );
[ A_di9,M_di9,C_di9,K_di9,amck_di9 ]=feat(
win_number,win_width,filtsig_di9,hi9.labels(1:21) );

[ A_dr10,M_dr10,C_dr10,K_dr10,amck_dr10 ]=feat(
win_number,win_width,filtsig_dr10,hr10.labels(1:21) );
[ A_di10,M_di10,C_di10,K_di10,amck_di10 ]=feat(
win_number,win_width,filtsig_di10,hi10.labels(1:21) );

%%

% Annotation data calculation

[anno_hr1]=annotation(hr1);
[anno_hi1]=annotation(hi1);

[anno_hr2]=annotation(hr2);
[anno_hi2]=annotation(hi2);

[anno_hr3]=annotation(hr3);
[anno_hi3]=annotation(hi3);

[anno_hr4]=annotation(hr4);
[anno_hi4]=annotation(hi4);

[anno_hr5]=annotation(hr5);
[anno_hi5]=annotation(hi5);

[anno_hr6]=annotation(hr6);
[anno_hi6]=annotation(hi6);

[anno_hr7]=annotation(hr7);
[anno_hi7]=annotation(hi7);
```

```
[anno_hr8]=annotation(hr8);
[anno_hi8]=annotation(hi8);

[anno_hr9]=annotation(hr9);
[anno_hi9]=annotation(hi9);

[anno_hr10]=annotation(hr10);
[anno_hi10]=annotation(hi10);

%%
% Classification using LDA

[lda_dr1,predicted_dr1,Resub_dr1,Pred_dr1,stat_dr1]=newldaclassification(amck_dr1,anno_hr1,hr1);
[lda_di1,predicted_di1,Resub_di1,Pred_di1,stat_di1]=newldaclassification(amck_di1,anno_hi1,hi1);

[lda_dr2,predicted_dr2,Resub_dr2,Pred_dr2,stat_dr2]=newldaclassification(amck_dr2,anno_hr2,hr2);
[lda_di2,predicted_di2,Resub_di2,Pred_di2,stat_di2]=newldaclassification(amck_di2,anno_hi2,hi2);

[lda_dr3,predicted_dr3,Resub_dr3,Pred_dr3,stat_dr3]=newldaclassification(amck_dr3,anno_hr3,hr3);
[lda_di3,predicted_di3,Resub_di3,Pred_di3,stat_di3]=newldaclassification(amck_di3,anno_hi3,hi3);

[lda_dr4,predicted_dr4,Resub_dr4,Pred_dr4,stat_dr4]=newldaclassification(amck_dr4,anno_hr4,hr4);
[lda_di4,predicted_di4,Resub_di4,Pred_di4,stat_di4]=newldaclassification(amck_di4,anno_hi4,hi4);

[lda_dr5,predicted_dr5,Resub_dr5,Pred_dr5,stat_dr5]=newldaclassification(amck_dr5,anno_hr5,hr5);
[lda_di5,predicted_di5,Resub_di5,Pred_di5,stat_di5]=newldaclassification(amck_di5,anno_hi5,hi5);

[lda_dr6,predicted_dr6,Resub_dr6,Pred_dr6,stat_dr6]=newldaclassification(amck_dr6,anno_hr6,hr6);
[lda_di6,predicted_di6,Resub_di6,Pred_di6,stat_di6]=newldaclassification(amck_di6,anno_hi6,hi6);

[lda_dr7,predicted_dr7,Resub_dr7,Pred_dr7,stat_dr7]=newldaclassification(amck_dr7,anno_hr7,hr7);
[lda_di7,predicted_di7,Resub_di7,Pred_di7,stat_di7]=newldaclassification(amck_di7,anno_hi7,hi7);

[lda_dr8,predicted_dr8,Resub_dr8,Pred_dr8,stat_dr8]=newldaclassification(amck_dr8,anno_hr8,hr8);
[lda_di8,predicted_di8,Resub_di8,Pred_di8,stat_di8]=newldaclassification(amck_di8,anno_hi8,hi8);

[lda_dr9,predicted_dr9,Resub_dr9,Pred_dr9,stat_dr9]=newldaclassification(amck_dr9,anno_hr9,hr9);
[lda_di9,predicted_di9,Resub_di9,Pred_di9,stat_di9]=newldaclassification(amck_di9,anno_hi9,hi9);

[lda_dr10,predicted_dr10,Resub_dr10,Pred_dr10,stat_dr10]=newldaclassification(amck_dr10,anno_hr10,hr10);
[lda_di10,predicted_di10,Resub_di10,Pred_di10,stat_di10]=newldaclassification(amck_di10,anno_hi10,hi10);
```

ldaclassification function

```
function [ lda,testx,predicted_data,R_details,P_details ] =
ldaclassification( features,freq_used,annot_v,label,t )
%UNTITLED This function uses the lda classifier and trains it using the
training data and predicts the output of the test data.
%   Detailed explanation goes here

train_annot_v=annot_v(1:10000);
test_annot_v=annot_v(10001:20000);
time1=t(1:10000);
time2=t(10001:20000);
label1=label';

%For 21 channels

for i=1:21


%      x=[features{1,i}{1,1}]; % Acivity
%      x=[features{1,i}{1,2}];% Mobility
%      x=[features{1,i}{1,3}];% Complexity
%      x=[features{1,i}{1,4}];% Kurtosis
%      x=[features{1,i}{1,1};features{1,i}{1,2}];
%      x=[features{1,i}{1,1};features{1,i}{1,2};features{1,i}{1,3}];

x=[features{1,i}{1,1};features{1,i}{1,2};features{1,i}{1,3};features{1,i}{1,4
}];
    annotation_labels=train_annot_v';
    trainx=x(:,1:10000);
    testx=x(:,10001:20000);
    y=trainx';

%     [lda]=fitcdiscr(y,annotation_labels);
    [lda(i)]={fitcdiscr(y,annotation_labels)};

    predicted_data(i)={predict(lda{1,i},testx')};

%     figure();
%     plot(time2,predicted_data,'r',time2,test_annot_v','k');
%     ylim([0,2.5]);
%     legend('Predicted Value','Original annotation');
%     title({'Predicted value for Channel',label1{i},freq_used});
%     xlabel('time (s)');
%     ylabel('annotation');

%%%%%% Resubstitution error and details
    resuberror(i) = {resubLoss(lda{1,i})};
    R(i)={confusionmat(lda{1,i}.Y,resubPredict(lda{1,i}))};
%%%%%% Prediction error and details
    L(i) = {loss(lda{1,i},testx',test_annot_v')};
    T(i)={confusionmat(lda{1,i}.Y,predicted_data{1,i})};
label=header.labels(1:21);
train_annot=annot(1:15);
annotation_labels=train_annot';
test_annot=annot(16:30);
cp = classperf(annotation_labels);
for i=1:21
```

```
    x=[amck{1,i}{1,1};amck{1,i}{1,2};amck{1,i}{1,3};amck{1,i}{1,4}];

    trainx=x(:,1:15);
    testx=x(:,16:30);
    y=trainx';

    [lda(i)]={fitcdiscr(y,annotation_labels)};

    predicted_data(i)={resubPredict(lda{1,i})};
%     predicted_data(i)={predict(lda{1,i},testx')};

    %%%%%% Resubstitution error and details
    resuberror(i) = {resubLoss(lda{1,i})};
    R(i)={confusionmat(lda{1,i}.Y,resubPredict(lda{1,i}))};
%%%%%% Prediction error and details
    L(i) = {loss(lda{1,i},testx',test_annot')};
    T(i)={confusionmat(lda{1,i}.Y,predicted_data{1,i})};

    stat_values(i)={Evaluate(annotation_labels,predicted_data{1,i})};
%     classperf(cp,predicted_data{1,i},annotation_labels) ;



end
lda=[lda;label'];
R_details=[resuberror;R;label'];
P_details=[L;T;label'];
% R=[R;label'];
end
```

Feature function

```
% function [ asig1,msig1,csig1,kurt1,amck ] = feat(
win_number,win_width,filtsig,label,samp )
function [ asig1,msig1,csig1,kurt1,amck ] = feat(
win_number,win_width,filtsig,label )
%feat
%Summary of this function
%This function calculates the different Hjorth Parameter features
% This function in turn calls the other functions Activity, Mobility,
% Complexity and Kurtosis
for j=1:21
    asig=0;
    msig=0;
    csig=0;
    kurt=0;
    for i=1:win_number
        frame = filtsig{1,j}(i:i+win_width);
        [asig(i)]=activity(frame);
        [msig(i)]=mobility(frame);
        [csig(i)]=complexity(frame);
        [kurt(i)] = kurtosis(frame);
    end
%     asig=[asig,zeros(1,samp)];
%     msig=[msig,zeros(1,samp)];
%     csig=[csig,zeros(1,samp)];
%     kurt=[kurt,zeros(1,samp)];
    amck{j}={asig,msig,csig,kurt};
    [asig1(j)]={asig};
    [msig1(j)]={msig};
```

```
    [csig1(j)]={csig};
    [kurt1(j)]={kurt};
end

asig1=[asig1;label'];
msig1=[msig1;label'];
csig1=[csig1;label'];
kurt1=[kurt1;label'];
amck=[amck;label'];
end
```

Channeldata Function

```
function
[Fc5,Fc3,Fc1,Fcz,Fc2,Fc4,Fc6,C5,C3,C1,Cz,C2,C4,C6,Cp5,Cp3,Cp1,Cpz,Cp2,Cp4,Cp6
]=channeldata( data )
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
Fc5=data{1,1};
Fc3=data{1,2};
Fc1=data{1,3};
Fcz=data{1,4};
Fc2=data{1,5};
Fc4=data{1,6};
Fc6=data{1,7};

C5=data{1,8};
C3=data{1,9};
C1=data{1,10};
Cz=data{1,11};
C2=data{1,12};
C4=data{1,13};
C6=data{1,14};

Cp5=data{1,15};
Cp3=data{1,16};
Cp1=data{1,17};
Cpz=data{1,18};
Cp2=data{1,19};
Cp4=data{1,20};
Cp6=data{1,21};

% Filtering the signal

Fs = 160;           % Sampling Frequency
```

```
fn = Fs/2;           % Nyquist Frequency
start = 13;         % Start Frequency
stop = 31;           % Stop Frequency
% waveletFunction = 'db8';
% [Gamma,Beta,Alpha,Theta,Delta]=getBandfromeeg(Fc5,waveletFunction);
W1 = start/fn;
W2 = stop/fn;
Wn=[W1 W2];
b=fir1(6,Wn,'bandpass');

Fc5f=filter(b,1,Fc5);
Fc6f=filter(b,1,Fc6);
Fc3f=filter(b,1,Fc3);
Fc4f=filter(b,1,Fc4);
Fc1f=filter(b,1,Fc1);
Fc2f=filter(b,1,Fc2);
Fczf=filter(b,1,Fcz);

C5f=filter(b,1,C5)';
C6f=filter(b,1,C6)';
C3f=filter(b,1,C3)';
C4f=filter(b,1,C4)';
C1f=filter(b,1,C1)';
C2f=filter(b,1,C2)';
Czf=filter(b,1,Cz);

Cp5f=filter(b,1,Cp5)';
Cp6f=filter(b,1,Cp6)';
Cp3f=filter(b,1,Cp3)';
Cp4f=filter(b,1,Cp4)';
Cp1f=filter(b,1,Cp1)';
Cp2f=filter(b,1,Cp2)';
Cpzf=filter(b,1,Cpz);

filtsig={Fc5f,Fc3f,Fc1f,Fczf,Fc2f,Fc4f,Fc6f,C5f,C3f,C1f,Czf,C2f,C4f,C6f,Cp5f,
Cp3f,Cp1f,Cpzf,Cp2f,Cp4f,Cp6f};

end
end
```

Activity Function

```
function [ act ] = activity( x )
%This function calculates the activity of the windowed EEG function
%   This function basically calculates the varience of the signal
act=var(x);
end
```

Mobility Function

```
function [ mob ] = mobility( x )


d=diff(x);
mob=sqrt((activity(d))/(activity(x)));
```

```
end
```

Complexity Function

```
function [ comp ] = complexity( x )
d=diff(x);
comp=(mobility(d)/mobility(x));
end
```

SVMclassifier Function

```
function [trainedClassifier, validationAccuracy] =
trainsvmClassifier(datasetTable)
% Convert input to table
datasetTable = table(datasetTable);
datasetTable.Properties.VariableNames = {'column'};
% Split matrices in the input table into vectors
datasetTable.column_1 = datasetTable.column(:,1);
datasetTable.column_2 = datasetTable.column(:,2);
datasetTable.column_3 = datasetTable.column(:,3);
datasetTable.column_4 = datasetTable.column(:,4);
datasetTable.column_5 = datasetTable.column(:,5);
datasetTable.column = [];
% Extract predictors and response
predictorNames = {'column_1', 'column_2', 'column_3', 'column_4'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.column_5;
% Train a classifier
template = templateSVM('KernelFunction', 'linear', 'PolynomialOrder', [],
'KernelScale', 'auto', 'BoxConstraint', 1, 'Standardize', 1);
trainedClassifier = fitcecoc(predictors, response, 'Learners', template,
'Coding', 'onevsone', 'PredictorNames', {'column_1' 'column_2' 'column_3'
'column_4'}, 'ResponseName', 'column_5', 'ClassNames', [0 1 2]);

% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 5);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');

%% Uncomment this section to compute validation predictions and scores:
% % Compute validation predictions and scores
% [validationPredictions, validationScores] = kfoldPredict(partitionedModel);
```