In the Name of Allah, the Most Compassionate, the Most Merciful

EMBEDDED COMPUTATIONAL INTELLIGENCE BASED DEVELOPMENT OF BRAIN CONTROL INTERFACE FOR 2-DOF ROBOTIC MANIPULATOR FOR UPPER LIMB

By

Asif Ishfaque

2011-NUST-MS PHD- MTS-09

MS-70 (MTS)

Submitted to the Department of Mechatronics Engineering in fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In

MECHTRONICS ENGINEERING

Thesis Supervisor

Brig. Dr. Javaid Iqbal

College of Electrical & Mechanical Engineering

National University of Sciences & Technology

2013

DECLARATION

I hereby declare that I have developed this thesis entirely on the basis of my personal efforts under the sincere guidance of my supervisor Brig. Dr. Javaid Iqbal. All the sources used in this thesis have been cited and the contents of this thesis have not been plagiarized. No portion of the work presented in this thesis has been submitted in support of any application for any other degree of qualification to this or any other university or institute of learning.

Asif Ishfaque

ACKNOWLEDGEMENTS

Innumerable words of praise and thanks to Allah, the Almighty, and the Creator of the universe for carving the path for me and always helping me out in the best possible way. Without His Will and Mercy, I would not have been able to accomplish this milestone. I am grateful to my parents for their immense love, moral support, encouragement and prayers throughout my academic career.

I am deeply beholden to my supervisor Brig. Dr. Javaid Iqbal for his continuous guidance, inspiration, and patience. His ability of management and foresightedness taught me a lot of things which will be more helpful for me in my practical life.

I gratefully acknowledge the help and guidance provided by Guidance and Examination Committee members Mr. Ahsan Javed Awan, Dr. Umer Shahbaz and Dr. Khurram Kamal. Their valuable suggestions and comments were a great source to improve the research work presented in this thesis.

DEDICATION

To my parents, wife and teachers

ABSTRACT

The sever accidents, paralysis attacks and different diseases are major cause of interrupting the normal communication channel of brain with other body parts. The individuals, victim of above sever cases can't live a normal life and are burden on the society. To help these affected people who may have partially or completely lost independent motions of their limbs, there is requirement of a system which can bypass the normal communication channel of the brain and sends messages to the exterior world. To implement such kind of system, acquiring, filtering, feature extraction and classifying the brain signals is a major task. The focus of this research is to classify the EEG signals dataset by Artificial Neural Network, Support Vector Machine and well-known statistical techniques e.g. Linear Discriminant Analysis, Quadratic Discriminant Analysis, Naive Bayes and Decision Trees and also compare them to identify a suitable technique for hardware implementation. The performances of these classifiers are compared on the basis of confusion matrix and mean square error. The most efficient method will be used to give signals to microcontroller to control the motion of 2-DoF Robotic Manipulator for Upper Limb Prosthesis. The 2-DoF Robotic Manipulator designed in NUST will be used for testing. This research enhances the future potential capabilities of BCI systems.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

The communicative ability of human beings is one of the most important factors of making life enjoyable. It helps the humans to express their desires, feelings, and ideas and to cope with daily life. On the other hand, there are so many individuals who because of sever accidents, brainstem stroke, spinal cord or brain injury, cerebral palsy, Amyotrophic lateral sclerosis (ALS) and several other diseases have lost above mentioned communicative potentials and most of them are in locked-in syndrome. The locked-in syndrome is that kind of disorder in which an individual is fully aware and conscious of what is happening in his surroundings but is not capable to express his desires and feelings. To help these effected people who may have partially or completely lost independent motions of their limbs and to give them a small degree of autonomy of limbs motion there are 3 options for restoring function. The first way to restore the above deficiencies is to increase the working competencies of remaining pathways. Muscles that are not affected can compensate for the deficient pathways. Eye blinking is an efficient way to answer the question that one can understand very well and similarly people being affected by Dysarthria can use their hand movements to produce a synthetic speech. The second option which is costly is to restore the functions that are being dead locked by detouring around breaks in the neural pathways that control muscles. Last and very much feasible approach is to bypass the normal communication channel of the brain and give another non-muscular pathway that can cover the deficiencies. This approach is commonly known as Brain Computer Interface (BCI).

Brain Computer Interface is the highest level of Human-Machine Interaction, where the brain signals are translated into control signals to drive systems in real world. These signals are called electroencephalographic signals or EEG. EEG is a representative signal that contains information about the electrical condition of the brain. The roots of EEG lie in the work done by English physiologist Richard Caton in 1875. Using a sensitive galvanometer, Caton made electrical recordings from the brains of animals. The human EEG was first described by an Austrian psychiatrist Hans Berger in 1929. He observed that waking and sleeping EEG patterns were distinctly different from each other.

Brain computer interface is a communication as well as a control system, like other systems it has also input, translating algorithms and output [1]. The input in BCI is Electroencephalographic signals recorded from the brain. There are numerous translating algorithms used for manipulating the EEG signals to desired signals for the control of any actuator. The output of a system is any device that can be controlled by the computer generated commands.

1.1 Motivation

Since independence in 1947 from British, Pakistan has venerated three wars with India. In these wars, millions of soldiers have been martyred and many of them lost their limbs, along with these thousands of Pakistanis had also suffered because of daily road accidents and lost the independent motions of their limbs. These affected people are living miserable lives and are burden on the society.

This project will allow most individuals, even those who are totally locked-in or have lost their arms to live long lives, so that the personal, social, and economic burdens of their disabilities will be minimized.

The BCI field is in its rather early stages, and is still, for the most part, dominated by research, as opposed to manufacture. There have been quite a few breakthroughs up till now in this field, but the science of using brain signals for controlling real systems still has a long way to go. The aim of the research is to classify the brain signals by applying different classification techniques on different time domain and frequency domain features and the most efficient technique will be used to control the motion of a two Degree of Freedom (DoF) robotic manipulator. The brain signals are related to the movement of the upper limbs in forward and reverse direction. This is a four class problem.

1.2 Goals and Objectives

The basic goal of this research is to compare the performances of different well known classification techniques on the basis of Confusion Matrices for the development of offline Brian Computer Interface. There are six major objectives of this research.

- **a.** Acquisition of EEG data.
- **b.** Implementation of Best Pre-processing schemes; this includes filtration, windowing & dimensionality reduction.
- **c.** Testing different time and frequency domain features of the EEG signal and choosing the best of them for classifying the mental tasks.

d. Classification of the mental tasks.

e. Comparing the classification results using confusion matrices and estimating the most efficient classifier.

f. Using most efficient classification technique for controlling the motion of 2-DoF Robotic Manipulator for Upper Limb.

CHAPTER 2: LITERATURE REVIEW

The response of BCI system depends upon the input which is EEG signals. Electroencephalogram is the tool which is use to image the brain while performing a cognitive action. EEG allows us to view and record the changes in your brain activity during the time you are performing the task. A brief historical review on EEG is that, in 1875 Richard Caton (an English scientist) found the electrical activity of the brain. He used a sensitive galvanometer to observe the electrical signal of an animal’s brain. A Russian scientist, Vasili Yakovlevich Danilevsky had made similar experiments in 1876 and published them in his doctoral thesis in 1877.In 1883 Ernst von Fleischl-Marxow (an Austrian physiologist and physician)discovered that measurable currents were recognize on the surface of scalp during irritation of various sense organs that is a major prerequisite for the electroencephalogram.In 1912, a Soviet physiologist Vladimir V. Pravdich-Neminsky published photographic recordings of Brain Waves in dogs.Hans Berger (around 1929, by an Austrian Psychiatrist) was the first to record electroencephalographs from humans and discovered the alpha waves.In 1957 The toposcope (imaging of electrical brain activity) is found.

The fundamentals of Brain Computer Interface are to filter the data, reduce dimensionality of data, feature extraction and to classify these signals. For filtration the technique which is mostly practiced by numerous EEG researchers is Butterworth Filter. Different techniques have been implemented by numerous researchers to reduce the dimensionality of data, among them the most common are PCA and Independent Component Analysis (ICA). PCA [2, 3, 4], is a linear transformation technique and it shows optimal representation of data with respect to minimum mean square error. One key drawback of this technique is that the artifacts should not correlate with the data. ICA [5, 6, 7, 8], is a statistical technique which divides the mixtures of varied signals into its sources with having no past information about the nature of the signal. It has a drawback that it may affect the power spectrum of the signal along with one assumption that the sources must be mutually statistically independent and at the same time it is a most appropriate tool to eliminate the artifacts from the signal.

The features reflect similarities to a certain class as well as differences from the rest of the classes. Auto Regressive Components (AR) is popular method of exacting the features [9 10]. Here we extract the coefficients of the filter because it is expected that different kind of thinking produces different kind of filter coefficient. The coefficient will be serving as features. Common Spatial

Pattern (CSP) is a feature extraction technique [11, 12] in which brain signals are projected to a subspace, where the between classes differences are prominent and resemblances are minimized. The most practiced features in BCI are Standard Deviation, Activity, Mobility, Complexity and variance in time domain. In frequency Power Spectral Density is the most popular choice.

The purpose of classification step in a BCI system is to identify the operator's intention**.** Different attempts have been made to classify the brain signals, namely Bayesian analysis [13, 14, 15], it is a statistical classifier with an aim to allocate the observed feature vector to the categorized class to which it has the maximum probability of belonging and also creates nonlinear decision boundaries. Linear Discriminant Analysis [16, 17, 18, 19], is simple classifier and easy to implement, but it fails to produce strong results in the presence of noise and outliers. Support Vector Machine [20, 21, 22, 23], is one of the most popular classifiers for development of BCI. It is linear classifier and produces results with great speed. Its aim is to maximize the distance between the hyper-planes and the closest training samples**.** It fails to produce strong results in the presence of very strong noise. K-Nearest Neighbor Classifier (k-NNC) [24, 25], is a non-linear classifiers. It involves the metric distances, which are estimation of resemblances of features of the test vector and the features of each class. Its classifying accuracy decreases in the presence of high dimensional feature data. Artificial Neural Networks [26, 27, 28, 29], is a non-linear, very flexible and most implemented classifier in BCI. It has so many architectures e.g. ANFIS, SOFNN, PNN, and FIRNN. Self-Organizing Fuzzy Neural Networks (SOFNN) shows better results for classification because it joins the principles of both neural networks and the fuzzy logic in a single framework. Adaptive Autoregressive model [30] has also been successfully implemented for separation of brain signals recorded during left and right motor imagery.

Different attempts have been made by the researchers to actuate different devices with the help of brain signals. In 2004, Wolpaw and McFarland allowed an individual to travel a cursor about a 2 dimensional screen. In the same year Millán, et al. allowed an individual to transfer a robot about the room. In 2008, a brain controlled wheelchair to navigate in familiar environments was designed in National University of Singapore.

Different universities are exceling in BCI field especially EPFL University Switzerland and ESSEX University England. In EPFL, they have developed a wheel chair whose motion can be controlled by using brain signals. Recently they are working on Brain-Machine Interface for drivers and Brain-Coupled Interactive Devices. In ESSEX, they have also developed a brain control wheelchair and on Mining for Novel Signatures in Multi-Channel EEG for Brain-Computer Interfaces. Now they are working on Analogue Evolutionary Brain Computer Interfaces under the supervision of Dr. Francisco Sepulveda.

CHAPTER 3: BRAIN COMPUTER INTERFACE

Brain Computer Interface is a communication system, which bypasses the brain's normal output pathways of muscles and peripheral nerves and allows a patient to control its external world only by means of brain signals. Brain computer interface is a communication as well as a control system, like other systems it has also input (Electroencephalography signals), the algorithms for preprocessing of signals and to transfer the input into output and finally output (actuators.) this can be visualized in Fig.1.

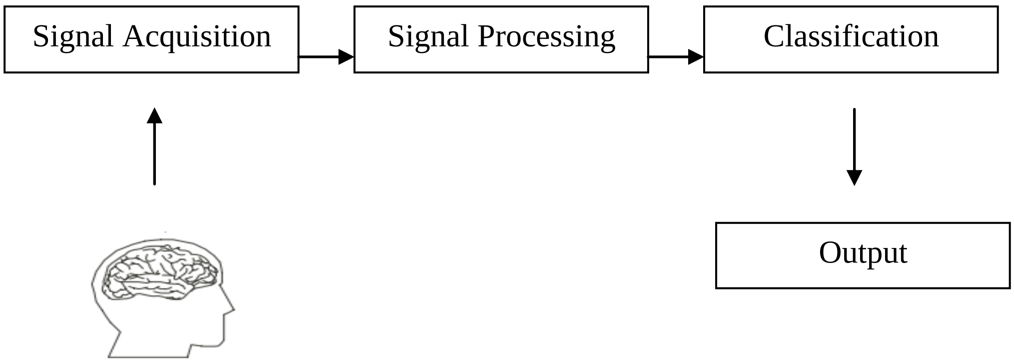


Fig.1 Basic Parts of BCI System

According to literature, the BCI systems may be classified as invasive/non-invasive, dependent / independent and synchronous/asynchronous. BCI systems where EEG signals are picked directly from the scalp are called non-invasive systems. Their counterparts, in which EEG signal is picked by cutting into the brain, are conversely called invasive systems. The pictorial view of above types can be seen in Fig.2. The BCI systems may also be dependent or independent; depending on the brain's normal output pathways. Even though invasive systems render much stronger signals, for most practical purposes, it is preferred to use non-invasive, independent BCIs.

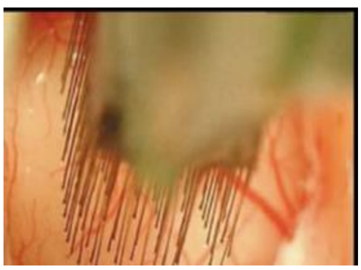


Fig.2 (a) Invasive method of recording of EEG signal.

Fig.2 (b) Non-invasive method of recording of EEG

In the recent years, the BCI systems have further been divided into two categories: Synchronous BCI systems and Asynchronous BCI systems. This distinction has been made on the basis of the data recording protocol. A BCI system is called synchronous when the user gives commands at preset intervals of time and the whole system moves in finite time windows. On the other hand asynchronous systems allow the BCI to respond to spontaneous activity, so that the user does not have to think of what he wants upon cue onset. In general, the Asynchronous BCI is far more complex as compared to synchronous systems.

3.1 Signal Acquisition

In BCI, the input of the system is electroencephalography signals which are recorded from the scalp. The dynamics of the EEG signals are mentioned below.

3.1.1 Electroencephalography signals

Electroencephalography signals are being recorded from the brain to detect the actions being performed. Human brain consists of special type of cells known as Neurons in a very complex architecture. These cells transmit the information from brain through electrical and chemical changes. In BCI, we try to capture the electrical differences produced in these cells. A neuron cell consist of a cell body, dendrites and most important an axon. In normal state, the neurons are in resting potential. The neurons potential fluctuates when there is an arrival of an impulse. The minimum summated voltages that can actuate a neuron or change its resting position is -43mv, after receiving this much potential the axon fires and generate a potential of a +30mv. This can be visualized in Fig. 3. The difference between the electrical signal intensity and the position of scalp from where this activity is measured, we can monitor the brain activity that which part of the body is now in action.

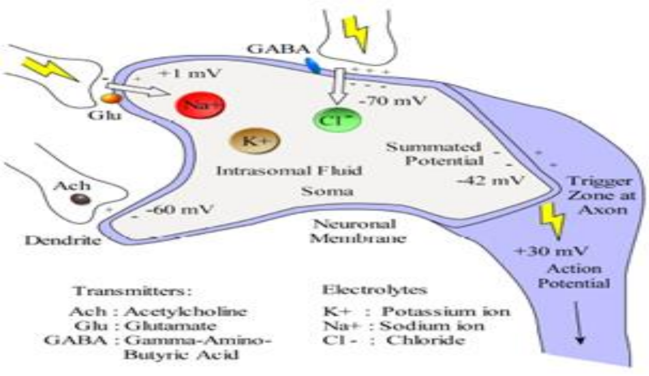


Fig .3 Summated and action voltages of a Neuron.

3.2 Signal processing:

Signal processing includes filtration, dimensionality reduction, windowing and feature extraction. In filtration the basic aim is to remove the artifacts (eye blinking, muscles movements, line noise, and baseline noise) that affect the performance of the signals. Windowing is done to include the one particular action which requires approximately 20-200msec. Features are characteristics of the signals and they help in classification of the signals. The features are of both time as well as frequency domain.

3.2.1 Filtration

Filtration is one of the fundamental steps of BCI. The basic aim of the filtration is to remove the artifacts from the signal that can change classification accuracy. The most commonly known artifacts are eye blinking, muscles movements, line noise, and baseline noise. The filter which gets popularity in BCI is Butterworth filter.

3.2.1.1 *Butterworth Filter*

Butterworth filter has a very flat frequency response. It rolls off towards zero in band stop filtration. Its performance increases as we increase the order of the filter. 6th order filter is very much common because of its highly smooth response. The comparison of Butterworth filter with other famous linear filters like Chebyshev and Elliptic is graphically shown in Fig. 4.

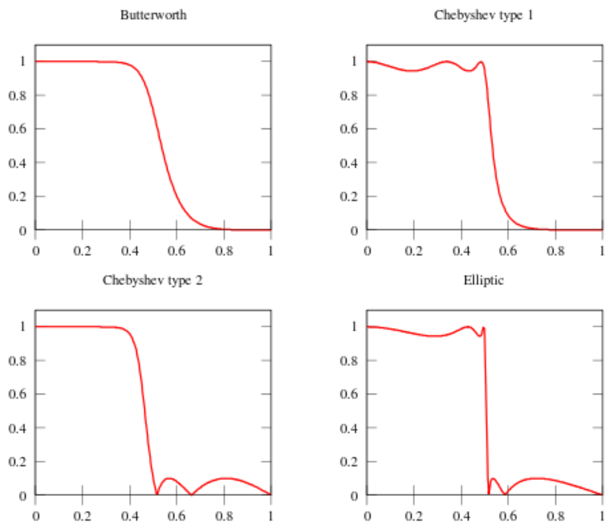


Fig.4. Comparison of different filters response

The above figure clears that Butterworth has a flat response and have zero ripples.

3.2.2 Dimensionality Reduction

EEG data has high dimensionality and it is difficult to process the data in real world so dimensionality reduction is one of the fundamentals of BCI. Different techniques have been implemented by numerous researchers to reduce the dimensionality of data, among them the most common is Principal Component Analysis (PCA).

3.2.2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a technique to reduce the dimensionality of a dataset consisting of a large number of interrelated variables by projection methods, in such a way that minimizes the loss of information. Principal component analysis is to transform m input vectors (variables) having the same length L molded in the m-dimensional vector x = $[x_1, x_2, ...x_m]^T$ into a vector y according to

$$y = A(x - m_x) \tag{1}$$

The vector m_x used in in Eq. (1) is the vector of mean values of all input vectors and can be found by following relation,

$$m_x = E\{x\} = \frac{1}{L}\sum_{l=1}^{L} x_k \tag{2}$$

Matrix A used in Eq. (1) can be calculated by determining the covariance matrix D_x. Rows of the matrix A are produced by the eigenvectors d of D_x in accordance with the cross ponding descending order eigenvalues. The D_x can be evaluated by the following relationship,

$$D_x = E\{(x - m_x)(x - m_x)^T\} = \frac{1}{L}\sum_{l=1}^{L} x_l\, x_l^T - m_x m_x^T \tag{3}$$

As we mentioned earlier that input vector x is of m- dimensional it is clear that the size of Dx will be m x m. The main diagonal elements Dx (i, i) are the variances of x

$$D_x(i,i) = E\{(x_i - m_i)^2\} \tag{4}$$

and the other values Dx (i, j) are the covariance between input variables xi, xj

$$D_x(i,j) = E\{(x_i - m_i)(x_j - m_j)\} \tag{5}$$

The inverse of PCA can be calculated because the rows A are orthonormal,

$$x = A^T y + m_x \tag{6}$$

So by following the above steps one can reduce the dimensionality of data. The next main goal is the selection of principal components.

The main objective for using PCA is to estimate the minimum number of principal components required have variance ≥95% of the complete data. For most of the experimental data, it is observed that the total variance lies within the first few principal components but to find out the exact number of components several methods have been proposed in the literature. The most commonly practiced are mentioned below,

i) Kaiser Gutman (KG) rule - The KG rule is based on if else condition about a single value, without considering the amount of variance. According to this rule only those PCs will be considered whose eigenvalues are ≥1. For large variable spaces, the KG- rule usually holds too many PCs.

ii) Cumulative Variance - In this method, we have to compute the variance of individual PCs and set a threshold T which in most of the cases ≥95%. The Cumulative sum of variance of number of PCs that exceed the threshold value is considered in our data and remaining all other is discarded. From the basis of PCA the total variance of the data can be calculated as

$$T_P = \sum_{i=1}^{P} \lambda_i \tag{7}$$

Where $\boldsymbol{\lambda}_i$ is eigenvalues of i-th PC (eigenvector). The clear definition of the cumulative variance computed for the first k PCs is therefore

$$T_k = \frac{1}{T_P} \sum_{i=1}^{k} \lambda_i \tag{8}$$

So to select the minimum number of PCs $T_k > T$.

3.2.3 Feature Extraction

The characteristics on the basis of which signals can be defined and distinguished are called signal features. EEG signals are stochastic in nature, i.e. single values of the signal are unreliable and meaningless. They are time-variant and non-stationary, with frequency components changing around every 200 ms.

Once digitized, the brain signals are divided into features, suitable for decoding the user's intent. These features have information embedded in the signal that is affected when the required task is carried out. BCIs may use signal features that are in the time domain (e.g., evoked potential amplitudes or neuronal firing rates), the frequency domain (e.g., mu or beta rhythm amplitudes) or both. It is also possible for a BCI to use signal features like sets of autoregressive parameters, which correlate with the user's intent but do not necessarily reflect specific brain events. Some of the features used in present day BCI systems are given below.

3.2.3.1 Mu and Beta Rhythms

Mu-Rhythm is a term given to α-rhythms (information in the alpha-band) in the central sensorimotor (C) region of the scalp overlying the sensorimotor cortex. This area is considered to contain the maximum information about movement of limbs, its imagination and preparation.

Mu-rhythms are often associated with beta rhythms. Together they arise in areas most directly connected to the brain's normal motor output channels, and so give the maximum information about movement (imagined or real). These rhythms increase when the user is at rest, and decrease in amplitude as movement occurs. They have high spatial information which make them good features for classifying different limb movements. Furthermore, with some training, people can learn to control the amplitude of their mu and beta rhythms thus improving the performance of the system.

3.2.3.2 ERD/ERS Maps

When mental tasks are performed, changes in the event related potential (ERP) and the oscillatory brain signal occur, which are characterized by two parameters known as the event related desynchronization and synchronization (ERD/ERS). A decrease in mu or beta rhythm occurring due to movement is called event-related desynchronization or ERD. Its opposite, rhythm increase, or event-related synchronization (ERS) occurs after completion of movement and with relaxation. When different parts of the body are moved (either actually or imagined) they generate ERD/ERS that are spatially and temporally apart, so that the region of origination of the movement can be known. Even if the movement is from the same limb, thus reducing spatial distance, maps of significant ERD/ERS changes can still be calculated in time-frequency domain independently for each movement.

3.2.3.3 Auto-Regressive (AR) / Adaptive Auto-Regressive (AAR) Features

In the statistical literature, regression stands for a functional relationship between two or more correlated variables used to predict values of an unknown variable. In the case when variables are time-related measurements from an ongoing event, e.g., an EEG, one may employ autoregressive analysis to analyze the event. AR modeling involves the computation of p coefficients (p = the model order), which can be used to predict the value of a signal at time t using a linearly weighted combination of these coefficients (a (n), n = 1, 2, …., p) and the p previous sample values.

3.2.3.4 Energy

The changes in the energy of the EEG signal can be used to determine a change in user intent. Vuckovic et al calculated the energy of the EEG signal using Gabor transform. They showed that there are statistically significant differences between Energy density maps for different movements in time-frequency domain and calculated those using different statistical techniques.

3.2.3.5 Fractal Dimensions

EEG signals are non-linear in nature as they originate from a highly complex, non-linear system, the brain. Moreover, they possess a self-affinity property, i.e. they comprise of rescaled copies of themselves at progressively small scales. This characteristic of the EEG signal can be demonstrated by its fractal dimension values. These values show a characteristic response in the ERD/ERS maps for different movements (or their imagination) and can thus be used to reveal the embedded potential responses in the human brain for differentiating between the relaxing and imaging periods. A high FD value indicates a more complex structure of EEG signals during the imagination of movements over the sensorimotor areas than during the relaxing period.

3.2.3.6 Slow Cortical Potentials

Slow Cortical Potentials are slow voltage changes generated in the cortex corresponding to movement or other functions involving cortical activation. They are reliable features, however due to their slow response (0.5 - 10 s) they are unsuitable for use in online systems

3.2.3.7 P300

Infrequent or particularly significant auditory, visual, or somatosensory stimuli, when interspersed with frequent or routine stimuli, induce in the EEG signal, a positive peak at 300ms, named P300. The amplitude of this P300 is much higher than the rest of the signal suggesting the triggering of an event.

3.3 Classification

The objective of classification in a BCI is to recognize the individual intentions on the basis of features. For classification both supervised learning algorithms and unsupervised algorithms are applied to get maximum percentage of accuracy. Classification algorithms can be applied to online and offline sessions of learning or both kinds of sessions. Offline classification is much easier than online classification and in offline session the analysis of data can be reviewed as many times as u can, because here we are independent form the restrictions of time. Different attempts have been made to classify the brain signals. The detailed description of most practiced classifiers is given below.

3.3.1 Artificial Neural Networks

Artificial Neural Networks are actually designed on the basis of the working of the neurons in the brain and their way of learning the new objects. The determination is to mimic the brain processing that immediately solve certain problems. It is very common in BCI and it is a non-linear technique. One basic application of ANNs is to recognize a pattern, because it has ability to learn from the training data. Basic elements of ANN are Nodes and connections, which are improved by the provided training algorithm during the training phase and try to achieve the maximum percentage accuracy during testing phase. The generalized structure of ANN can be visualized in Fig.5.

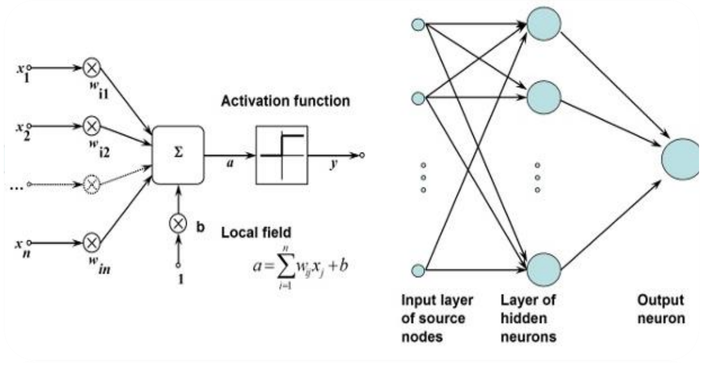


Fig. 5 Generalized structure of ANN

For successful implementation of ANN, one should set the following parameters very carefully,

a. Selection of ANN architecture
b. Number of layers
c. Number of neuron's in input layer, hidden layer and output layer
d. Activation function
e. Training algorithms to update the values of bias and weights
f. Number of maximum epochs

For ANN, the most difficult task is to estimate the correct number of neurons in the hidden layer. Different researchers have proposed different approaches for this problem and some of them are following,

a. The number of hidden neurons should be 2/3 the size of the input layer plus the size of the output layer.
b. The number of hidden neurons should be between the size of the input layer and the size of the output layer.
c. The number of hidden neurons should be less than twice the size of the input layer.

3.3.2 Linear Discriminant Analysis

LDA is a very simple technique that delivers adequate level of correctness of classification without performing high calculations. This technique maximizes the ratio of between class variance and the variance within the class and thereby promises maximal separability. One main feature of LDA is that it does not change the position of the data. It draws a linear decision region between the given classes.

Linear discriminant analysis expresses a linear discrimination function and in order to differentiate the classes, it represents a hyper plane in the feature space. The side of the hyper plane where the vector is found is actual measure of the decision of class to which the feature vector belongs.. In the case where the number of classes are greater than two (N>2), more than one hyper planes will be drawn and all are linear. The plane can be expressed mathematically as:

$$g(x) = w^T + w_0 \tag{9}$$

Where, x is the feature vector (which is to classify), w is a weight vector and w_o is threshold. When the data overlapping between classes is maximum then its performance will be decreased because quadratic decision plane does not come in the definition of LDA.

There is another technique which is also very much closely related to the LDA; that is Quadratic Discriminant Analysis (QDA). It classifies the data by drawing quadratic decision boundary. This technique is practiced on many EEG data classification analysis. The difference between LDA and QDA decision boundaries can be visualized in Fig.6.

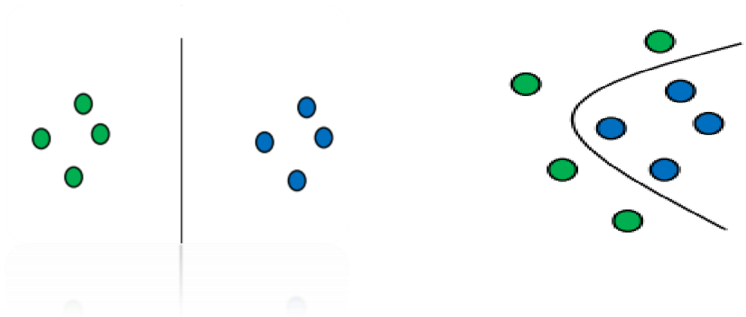


Fig.6. Difference between LDA and QDA decision boundaries

3.3.3 Decision Trees

The decision tree is used to predict the response by recursive partition of the instance space. The appealing point of DT is due to the fact that it signifies rules. DT has a tree like structure and has nodes, where each node is either a leaf or a decision node. It is a powerful tool to classify an example by opening at the root of the tree and moving through it until a leaf node, which delivers the classification of the instance. We will use Classification Tree algorithm in our current work.in this tree we start with all input data, and examine all possible binary splits on every predictor. Select a split with best optimization criterion. In our case we use Gini's diversity index (gdi) as an optimization criterion. The Gini index of a node is,

$$gdi = 1 - \sum_i P^2(i) \tag{10}$$

Where the sum is over the classes i at the node, and P (i) is the observed fraction of classes with class i that reach the node. For classification tree stops partition of the instance space when it will reach to the pure node; a node is pure if it contains only observations of one class.

3.3.4 Naive Bayes Classifier

Naïve Bayes classifier is also very common in BCI and it is practiced in many classification problems. This technique is based on so called Bayesian Theorem and it is suitable in a case where we have a high dimensionality. In this technique the first important parameter is the information about the Prior Probabilities, this information will be based on the previous experience. The next calculation is about the Likelihood of the class. The final classification is produced by combining both sources of information, i.e., the prior and the likelihood, to form a posterior probability using the so-called Bayes' rule.

3.3.5 Hidden Markov Models (HMV)

The Hidden Markov models are used to overcome the problem of non-stationaries of EEG signals by modeling the dynamic changes in it [27]. An HMM is a first order time domain process which generates probabilistic sequences and emits an output according to the specified probability distribution. Its unique characteristic is that the conditional probability governing the next state depends only on the current state. The state transitions themselves occur according to the distribution of transition probabilities [28]. Classification using HMMS is based on the selection of maximum best path probability for the feature sequence.

3.4 Output

The devices that can be connected to computer can be controlled with the help of BCI, but till now it has limited application because of its complexity and because of its very limited capacity of transfer rate between BCI system and actuator which is about 20 to 40bits/min [1]. Some of the applications possible with current BCIs

a. Wheel Chair Control
b. Artificial Prosthetic Limbs.
c. Spelling Devices
d. Games

CHAPTER 4: TIME DOMAIN ANALYSIS OF EEG SIGNALS

Time domain analysis of EEG signals is first approach to accomplish the tasks mentioned in objectives. In this approach all the important time domain features have been used for classification. The block diagram of the first approach can be visualized in Fig.7.

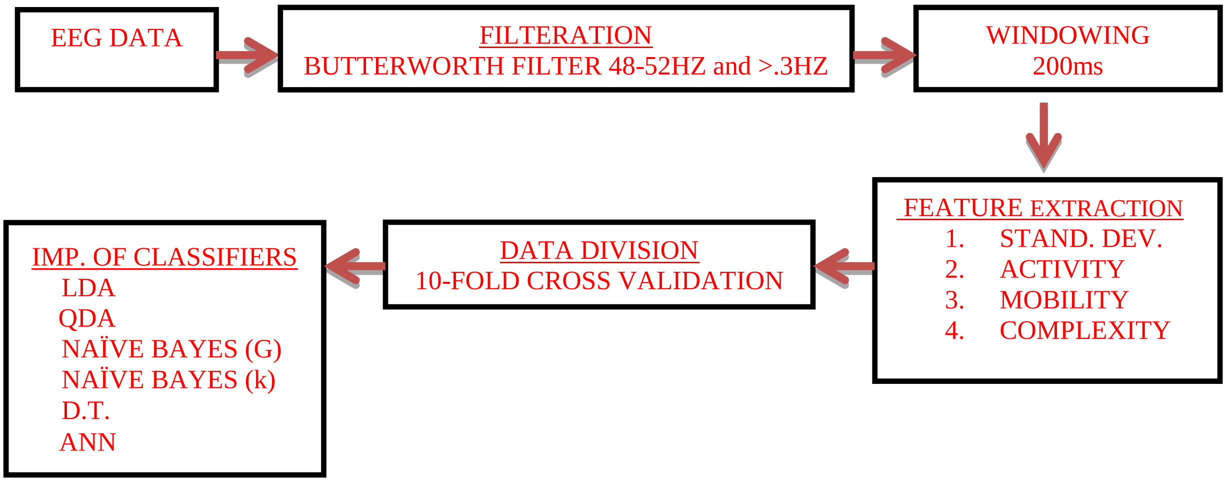


Fig.7. Block diagram of time domain analysis of EEG signals

4.1 EEG Data.

We used a publicly available data set.The data is of a right handed male having 21 years of age with no notorious medical disorders. The EEG data is recorded with closed eyes and comprises of random actions of right and left hand. The each independent motion has 19 columns and 3008 rows. Every single row denotes one electrode. The implanted electrodes have an order of FP1 FP2 F3 F4 C3 C4 P3 P4 O1 O2 F7 F8 T3 T4 T5 T6 FZ CZ PZ and can be visualized in Fig.8.

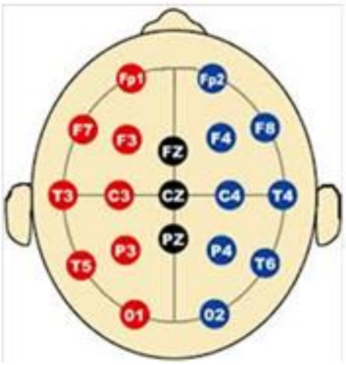


Fig.8. Human brain with 19 electrodes positions

The Neurofax EEG System has been used for recording of data at 500Hz. The common reference used for transformation of data is Eemagine. The recorded data used in our research involves both backward and forward movements of Right hand and the backward and forward movements of left hand. This establishes a classification problem of discriminating amongst 4 classes. The three trails of each class can be visualized in Fig.9, Fig.10, Fig.11, and Fig.12.

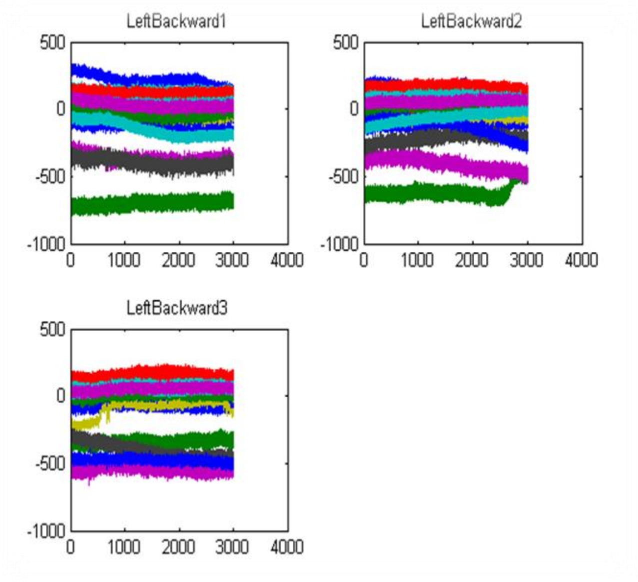


Fig.9 Left Hand Backward Motion.

This figure shows the 3008 rows of 19 electrodes of Left Hand Backward Motion of upper Limb.

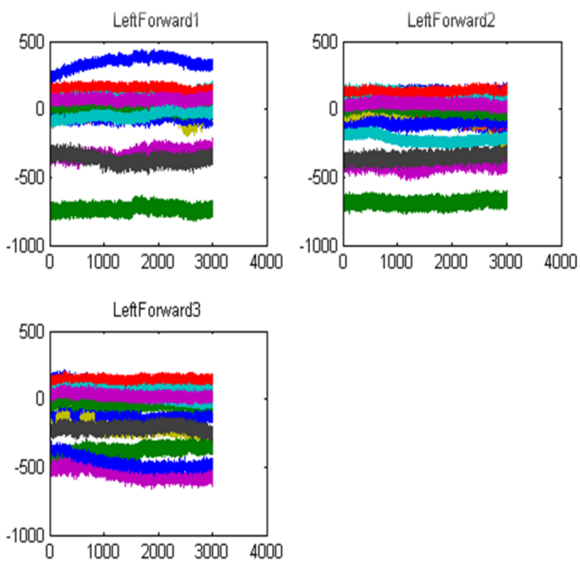


Fig.10 Left Hand Forward Motion.

This figure shows the 3008 rows of 19 electrodes of Left Hand Forward Motion of upper Limb.

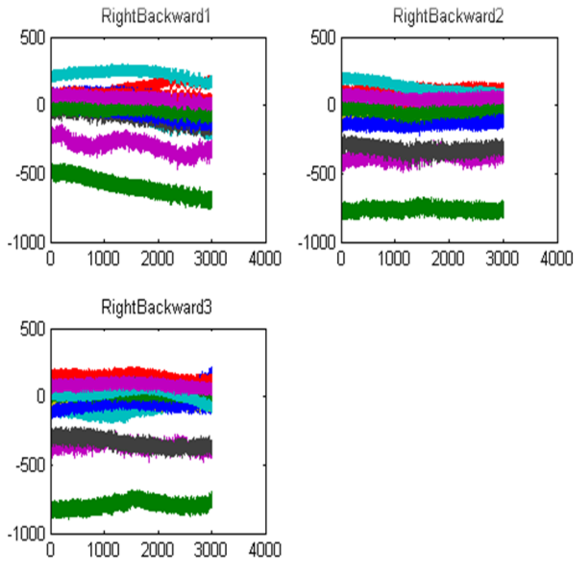


Fig.11 Right Hand Backward Motion.

This figure shows the 3008 rows of 19 electrodes of Right Hand Backward Motion of upper Limb.

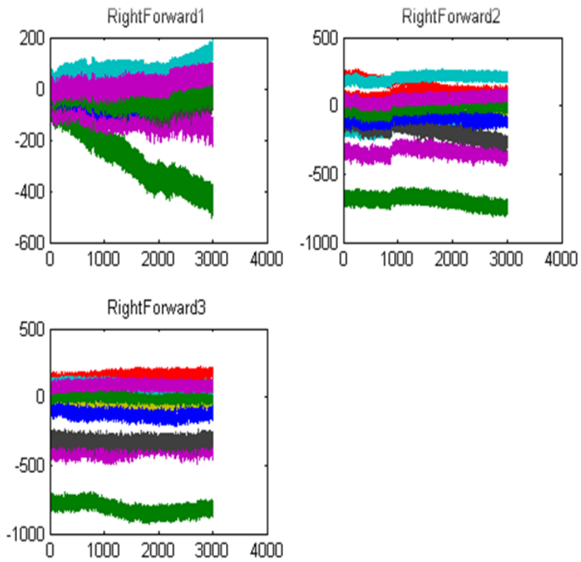


Fig.12 Right Hand Forward Motion.

This figure shows the 3008 rows of 19 electrodes of Right Hand Forward Motion of upper Limb.

4.2 Filtration

Filtration is a process which is used to remove the unwanted signals from the original data; it is one of the fundamental steps of successful implantation of the BCI. The filtration process can be visualized in Fig.13.

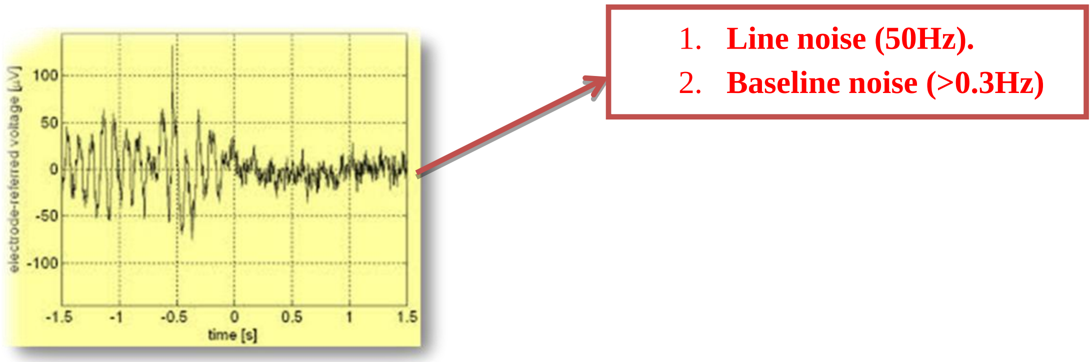


Fig.13 Artifacts removed from EEG signals.

A six order Butterworth filter has been used to remove the following two artifacts,

a. **Line noise:** the frequency of the AC signals flowing in the line can also affect the EEG signals. In Pakistan the frequency is of 50 Hz. This is removed by using a band stop filter 48Hz to 52Hz.
b. **Baseline Noise:** This noise does not allow the signals to be at zero base line and because of this noise the signal fluctuates on the base line. This is removed by using high pass filter with frequency less than 0.3Hz.

4.3 Windowing

The eye movement, muscle activity and the movement of the subject are important sources of causing large amplitude outliers in the EEG signals. To reduce the effect of such artifacts it is a better practice to make the windows of the data. The features are exacted from each window. In our case we make a window of 0.2 sec and each window has 100 rows and 1 column. This can be visualized in Fig.14.

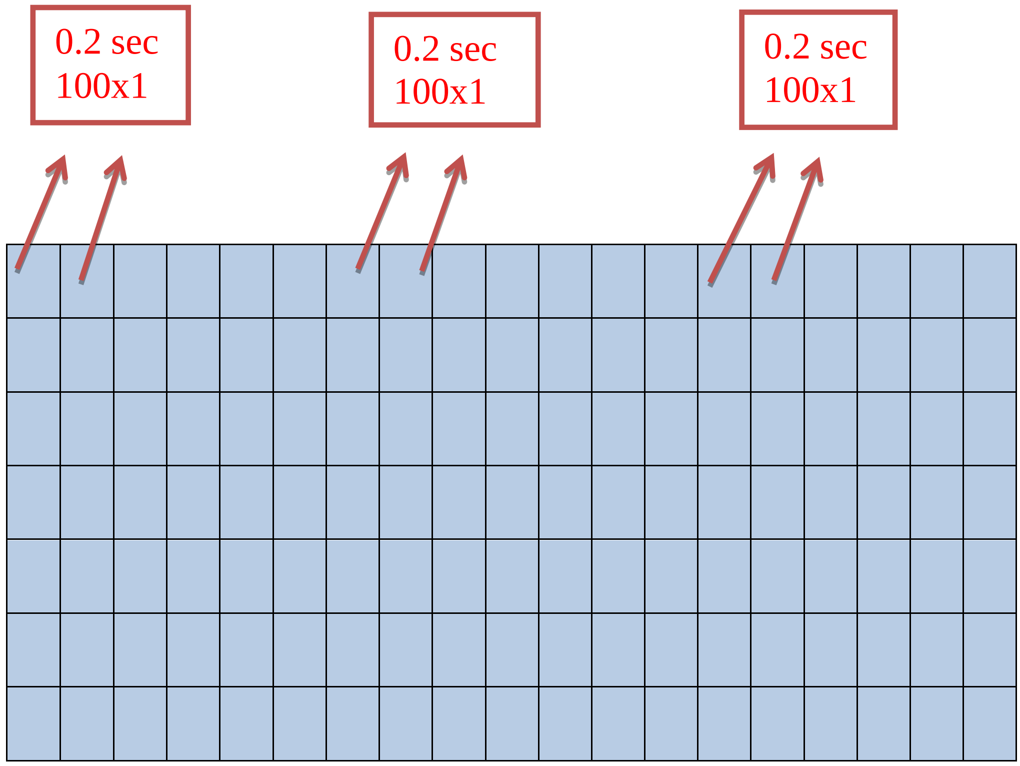


Fig.14 Windowing of EEG data.

Each window has 100 rows and 1column.Each trial has 3008 rows and 19 columns, so each contains 570 windows. The data has 6840 windows.

4.4 Feature Extraction:

Different thinking activities result in different patterns of brain signals. BCI is seen as a pattern recognition system that classifies each pattern into a class according to its features. BCI extracts some features from brain signals that reflect similarities to a certain class as well as differences from the rest of the classes. The features are measured or derived from the properties of the signals which contain the discriminative information needed to distinguish their types.

Proper features selection is the biggest task in BCI because it helps the classifier to recognize the user intension. Time domain features are related to changes in the amplitude of neurophysiologic signals either occurring time-locked to the presentation of stimuli or time-locked to actions of the user of a BCI system The time domain features used in this research are following,

4.4.1 Standard Deviation.

The standard deviation of a data set is a measure of how spread out the data is. It can be mathematically expressed as,

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

4.4.2 Activity:

Activity is equal to the variance of the signal. It can be mathematically expressed as,

$$Activity(x) = \frac{\sum_{n=1}^{N}\left(x(n) - \bar{x}\right)^2}{N}$$

4.4.3Mobility

Mobility is a measure of the signal mean frequency.

$$Mobility(x) = \sqrt{\frac{\mathrm{var}(x')}{\mathrm{var}(x)}}$$

Where x' stands for the derivate of signal x.

4.4.4 Complexity:

Complexity measures the deviation of the signal from the sine shape.

$$Complexity(x) = \frac{Mobility(x')}{Mobility(x)}$$

4.5 Data Division:

The division of data into training and testing is important step before feeding to the classifier. The method adopted to perform such division is 10-fold cross validation method. In this method the total data has been divided into 10 equal parts which are named as folds. The data division can be seen in Fig.15.

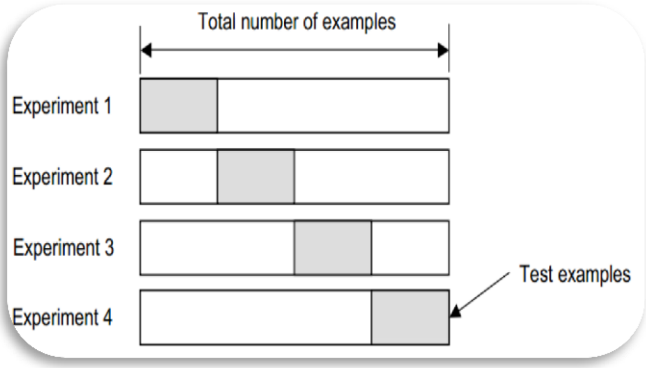


Fig.15 10-fold Cross Validation Method

In this method 9 folds are used for training and 1 fold is used for testing, this process continues in 10 loops and each time the testing fold changes from first fold to second fold and remaining data is used for training. This method also serves the purpose of the validation of the data. This method is far better than other data division methods like Holdout method and Leave-one-out cross-validation method.

4.6 Classification:

The objective of the classification in a BCI system is to recognize the individual's intentions, either using classification algorithms or by regression, but using classification algorithms is now days the most common approach. Classification algorithms can be applied to online oroffline sessions of learning or both kinds of sessions. In our research, we perform classification of offline sessions in supervisory mode of learning and also provide the labeled data set .Here we implement well-known supervised learning classification techniques i.e. Linear Discriminant Analysis, Quadratic Discriminant Analysis, Artificial Neural Networks, Naive Bayes and Decision Trees.

4.6.1 Linear Discriminant Analysis (LDA):

This technique maximizes the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability.

4.6.2 Quadratic Discriminant Analysis (QDA):

This technique is same as that of LDA but the only difference is the shape of the decision boundary. The LDA creates linear decision boundary and is less flexible, on the other hand QDA creates quadratic decision boundary and is more flexible

4.6.3 Artificial Neural Networks (ANN):

Artificial neural networks are models inspired by animal centered nervous systems (in particular the brain) that are capable of machine learning and pattern recognition. They are usually presented as systems of interconnected "neurons" that can compute values from inputs by feeding information through the network.

We used Feed Forward Architecture for classification. There are 3 layers and numbers of neurons in each layer are 19, 20 and 1 respectively. The TANSIG and PURELIN are used as an activation function. The network is trained with Scaled Conjugate Gradient Back propagation. Maximum number of epochs is 2000.

4.6.4 Naive Bayes (N.B):

The Naive Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. In this research we apply two architectures of the classifiers.

- Naïve Bayes with Normal (Gaussian) distribution.
- Naïve Bayes with Kernel smoothing density estimate.

4.6.5 Decision Trees (D.T):

The decision tree is used to predict the response by recursive partition of the instance space. It has two types of trees; Classification Trees and Regression Trees. In this research we used classification tree and optimization criterion 'GDI‘. (Gini's Diversity Index).

4.7 Results:

The final classification achieved by implementing all the above mentioned classifiers on individual features can be visualized in the charts shown below.

4.7.1 Results (Standard Deviation):

In this result as shown in Fig.16, all the above mentioned classifiers are implemented on the EEG data when standard deviation is used as a feature.

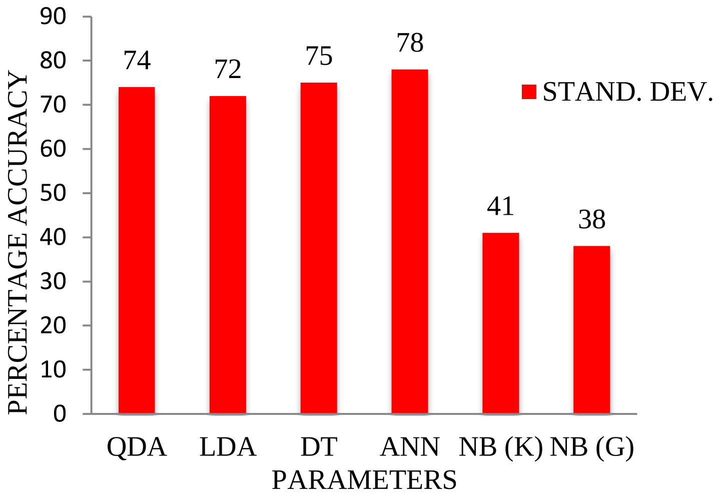


Fig.16. Results of classifiers when SD is used as a feature

This chart shows that QDA shows better results than LDA because here the data required quadratic boundaries for classification. The best results are shown by ANN but it takes too much time to give the final output. Naïve Bayes classifier is not well suited for this type of application.

4.7.2 Results (Activity):

In this result as shown in Fig.17, all the above mentioned classifiers are implemented on the EEG data when activity is used as a feature.

This chart proves that QDA is the best technique to classify the EEG data when activity (variance) is used as a feature. QDA is more flexible in creating discrimination among data. Again Naïve Bayes classifier is not well suited for this type of application.

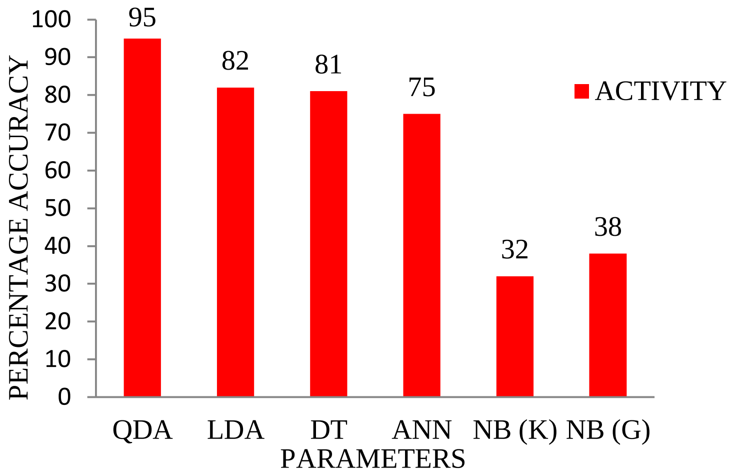


Fig.17. Results of classifiers when Activity is used as a feature

4.7.3 Results (Mobility):

In this result as shown in Fig.18, all the above mentioned classifiers are implemented on the EEG data when mobility is used as a feature.

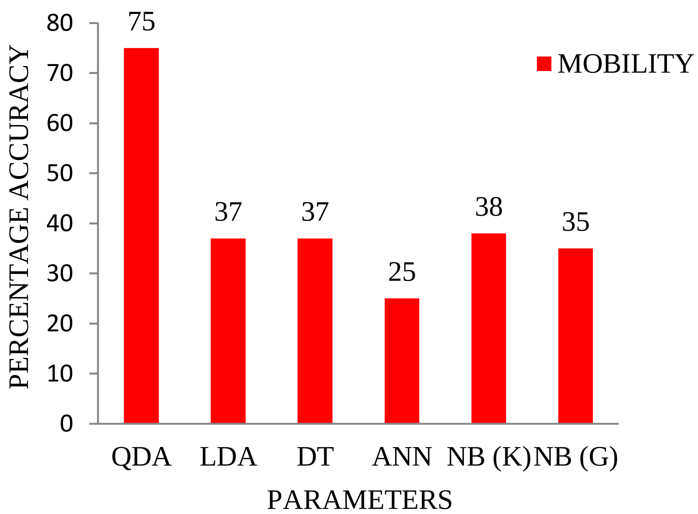


Fig.18. Results of classifiers when Mobility is used as a feature

QDA classifies this data very well because the overlapping of the data and QDA is best choice for the case where the overlapping is maximum.

4.7.4 Results (Complexity):

In this result as shown in Fig.19, all the above mentioned classifiers are implemented on the EEG data when complexity is used as a feature.

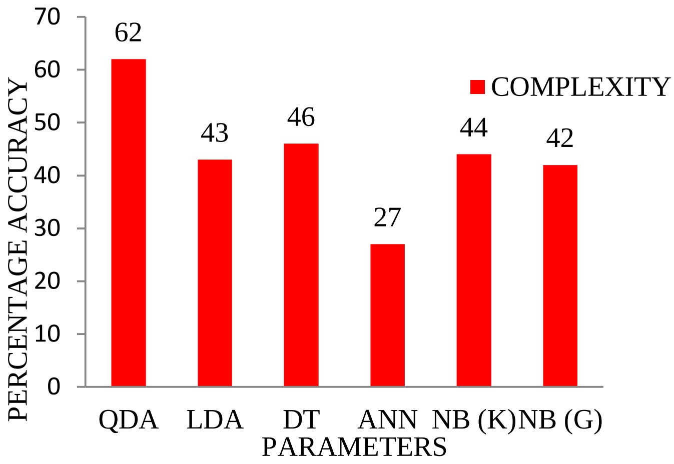


Fig.19. Results of classifiers when Complexity is used as a feature

The classification accuracies are much less than the other feature's classification result but still QDA performance is far better than other classifiers. Naïve Bayes classification accuracy also increased and shows better result than LDA and ANN.

The combine results as shown in Fig.20, of all the features along with the implementation of all the mentioned classifiers,

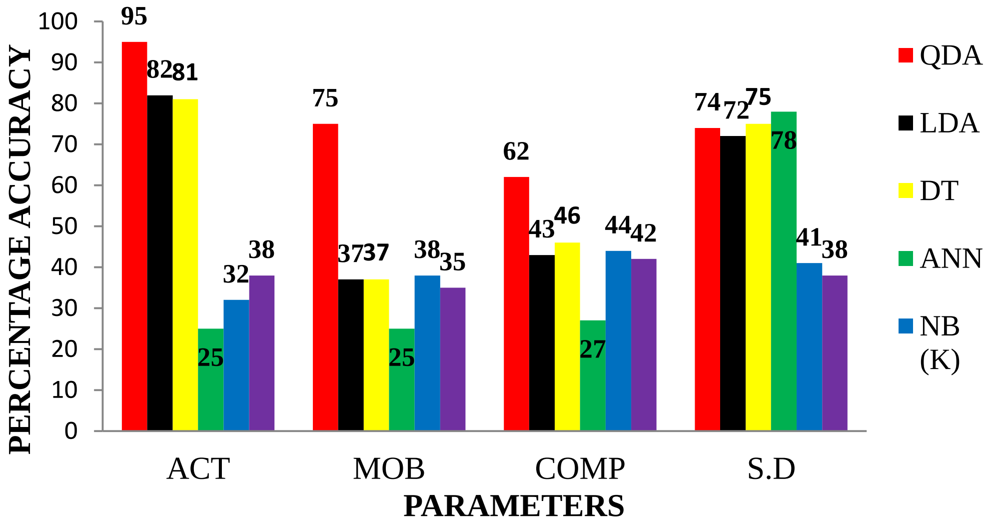


Fig.20. Combine results of all classifiers

The combine result proves the superiority of QDA in classification of the EEG data in all above mentioned time domain features.

CHAPTER 5: FREQUENCY DOMAIN ANALYSIS OF EEG SIGNALS

Frequency domain analysis of EEG signals is second approach to accomplish the tasks mentioned in objectives. In this approach the important time domain features named as Power Spectral density (PSD) has been used for classification. The block diagram of second approach is shown in Fig.21.

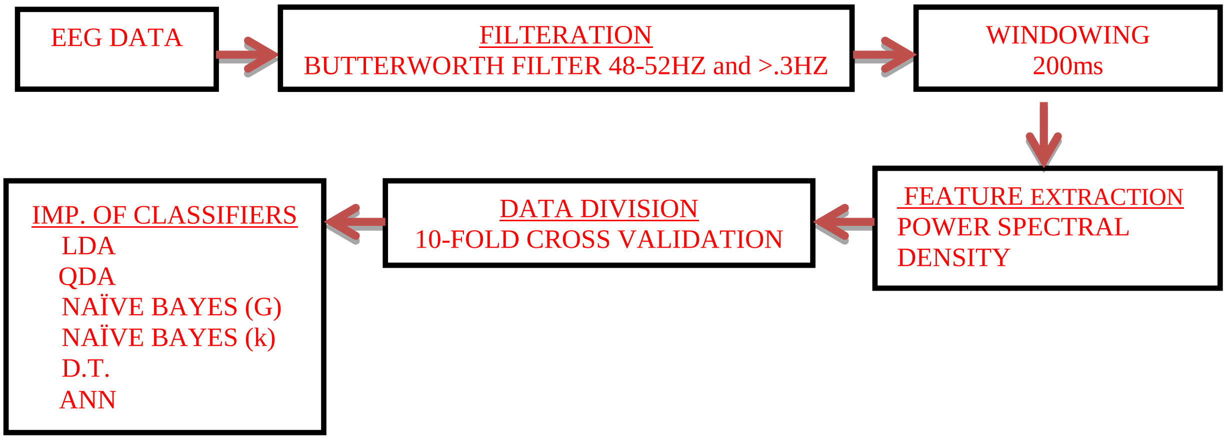


Fig.21. Block diagram of frequency domain analysis of EEG signals

In this approch the EEG data, filteration ,windowing, data division and classifiers are the same , the only difference liesss in the feature that has been extracted in frequency domain.

5.1 Feature Extraction:

A lot of important information can be extracted from the frequency of an EEG signal [13]. Based on the information it carries, the frequency of an EEG signal (0-80Hz) is broadly divided into 5 smaller bands, for ease of classification.

a. Delta Band ~ 0-4 Hz
b. Theta Band ~ 4-7 Hz
c. Alpha Band~ 8-13 Hz
d. Beta Band ~ 14-30 Hz
e. Gamma Band~ 30-80Hz

The feature selected for this problem is power spectral density.

5.1.1 Power Spectral Density

Power Spectral Density of the EEG signal also has many properties that maybe used as features in BCI systems. Band Value Power obtained from power spectral density is one of such features that have been used extensively. Power Spectral Density assumes linearity, gaussianality and minimum-phase within the EEG signals, i.e., the amplitudes of EEG signals are normally distributed, their statistical properties do not vary over time, and their frequency components are uncorrelated. For this reason, they are more commonly used for comparison purposes. The PSD has been calculated from each of the window as shown in Fig.22.

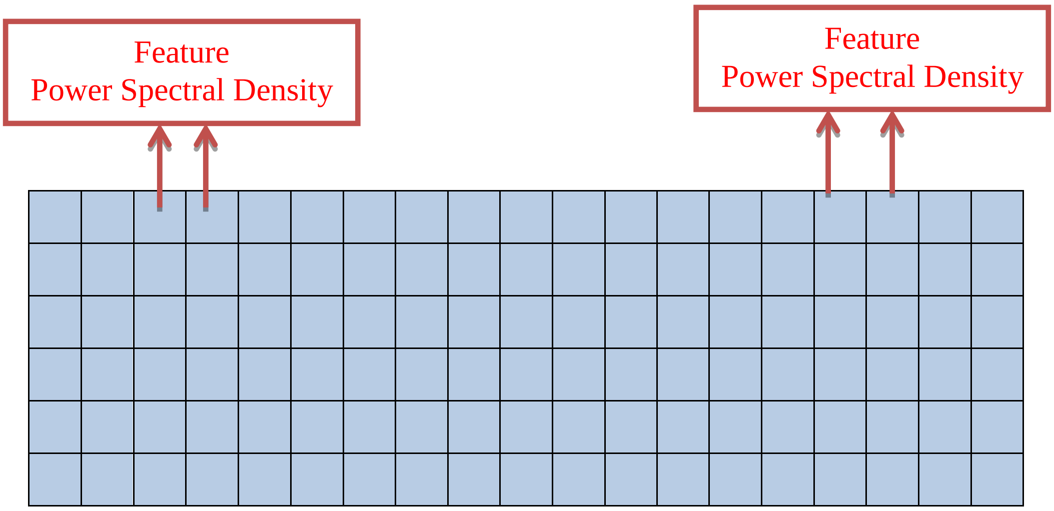


Fig.22. Feature extraction from each window.

5.2 Results:

In this result as shown in Fig.23, all the above mentioned classifiers are implemented on the EEG data when Power Spectral Density is used as a feature.

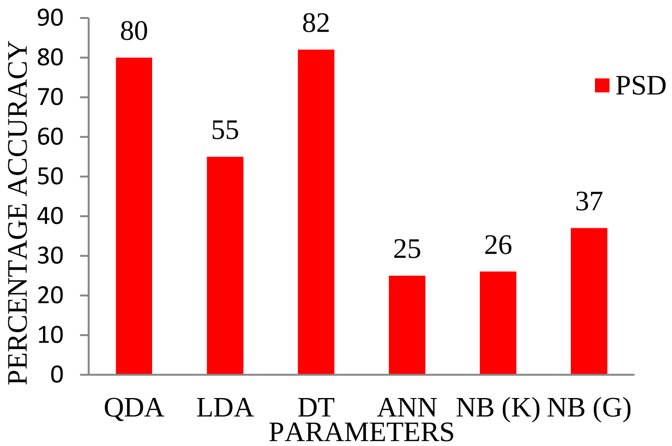


Fig.23. Results of classifiers when PSD is used as a feature

This result shows the superiority of decision trees. Here the decision trees show 82% accuracy. The frequency domain parameters can easily be classified by using D.T

CHAPTER 6: TIME &FREQUENCY DOMAIN ANALYSIS OF EEG SIGNALS WITH DIMENSIONALITY REDUCTION

Time &frequency domain analysis of EEG signals with dimensionality reduction is the third approach to accomplish the tasks mentioned in objectives. In this approach important time domain features and frequency domain features have been used for classification as mentioned in the previous analysis. The one new technique added to this approach is that before feature exaction the dimensionality of data has been reduced by using Principal Component Analysis (PCA). The PCA analyzed data has 5 columns and the rows are unchanged. The block diagram of third approach can be visualized in Fig.24.

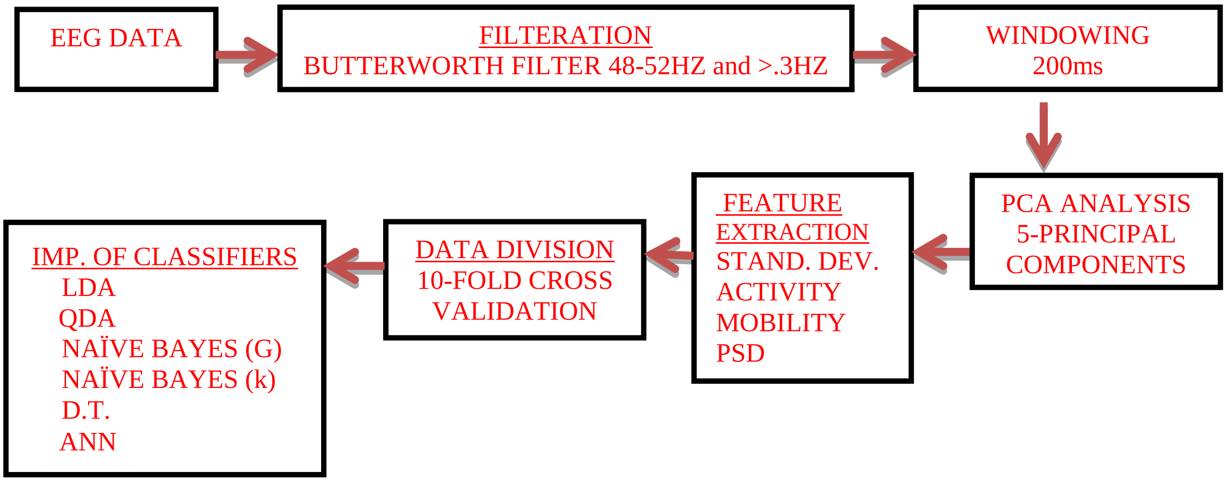


Fig.24. Block diagram of time &frequency domain analysis of eeg signals with dimensionality reduction

In this approch the EEG data, filteration ,windowing, data division, feature exaction and classifiers are the same , the only difference is lies in the dimionality reduction of the data.

6.1 Principal Component Analysis:

Principal Component Analysis (PCA) is a technique to reduce the dimensionality of a dataset consisting of a large number of interrelated variables by projection methods, in such a way that minimizes the loss of information.

In our research, the single trail has dimensionality of 19 columns and 3008 rows. This is a huge data for processing as well as for proper communication. This data takes a lot of time to be processed. The only solution for this type of problem is to reduce the dimensionality of the data. Different methods have been proposed by the researcher for this task, but the most suitable for EEG data is PCA.

The selection of appropriate number of principal components is the next biggest change. For this task, the cumulative variance approach has been used. In this approach we set the threshold value equal to or greater than 95%. We calculate the variance and find out that first five components gave the variance greater than 95%. The new data has 5 columns and the same rows. This can be visualized in Fig.25.

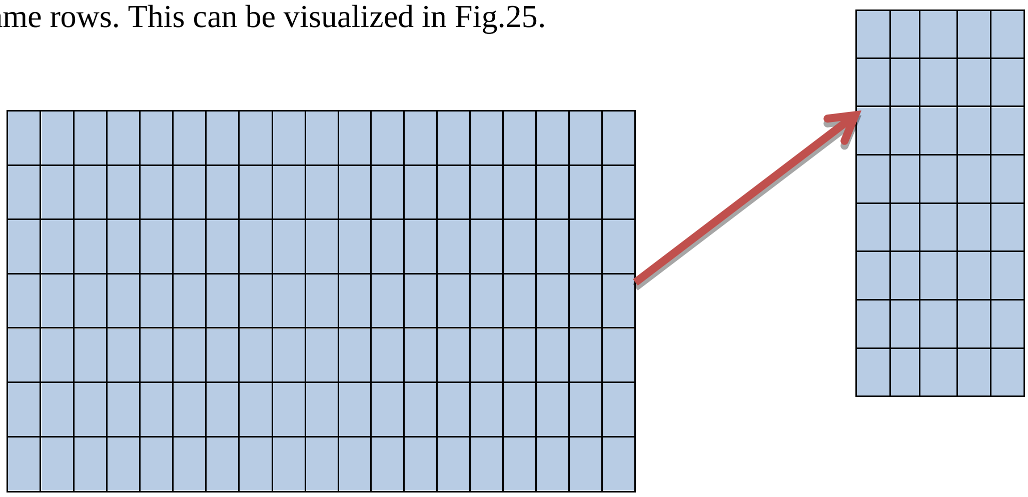


Fig.25. Dimensionality reduction using PCA

The PCA implemented data has 3008 rows and 5 columns and it is difficult to graphically represent the difference of amplitudes in all four movements. So, a down sampling method has been applied in such a way that we select first 10 values from each 100 values of rows and rows have been reduced from 3008 to 338. The same down sampling technique has been applied in all four cases and graphically represented in Fig. 26, Fig. 27, Fig. 28, and Fig. 29. The columns are unchanged during the down sampling.

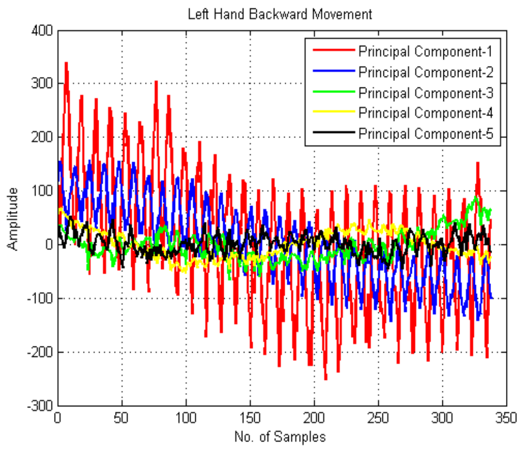


Fig.26. PCA analysis of Left Hand Backward Movement

PCA analysis of left hand backward movement; the five different colored signals represent the final correlation of the complete dataset. The pattern of the signals is very much similar to the sine wave. The amplitude varies at each step and this is actually the main key to classify the movements.

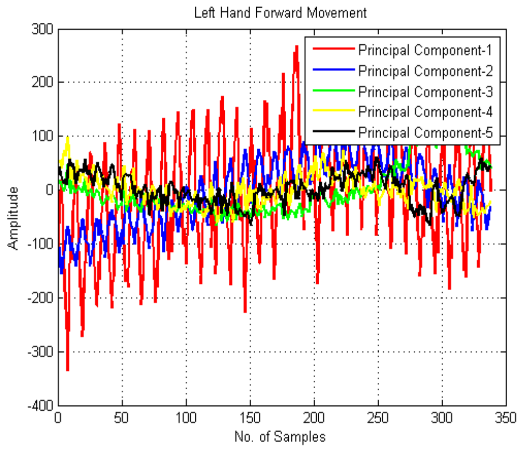


Fig.27. PCA analysis of Left Hand Forward Movement

PCA analysis of left hand forward movement: it shows clear differentiate from the backward movement because the amplitudes of the waves are different at many instances especially the red colored wave but follow the same pattern as that of sine wave.

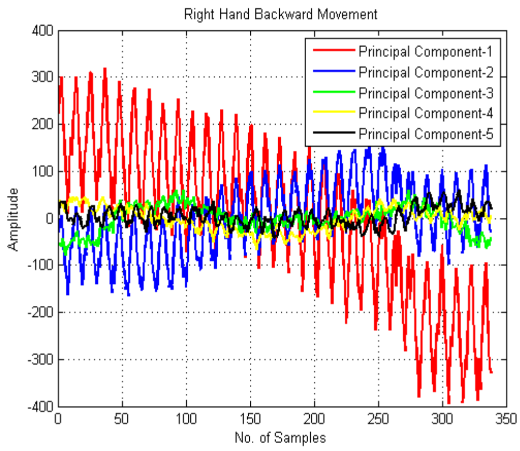


Fig.28. PCA analysis of Right Hand Backward Movement

PCA analysis of right hand backward movement; the initial amplitude of blue colored wave is negative and then increased and reached to the positive side and red colored wave and vice versa. The two main signals red and blue are not following the same pattern as that in left hand backward movement.

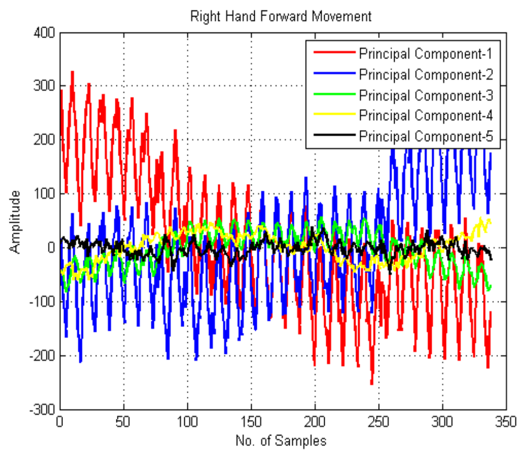


Fig.29. PCA analysis of Right Hand Forward Movement

PCA analysis of right hand forward movement: it follows the same pattern as that of in Fig.20 but having variations of amplitudes at different instances of samples especially the amplitude of the blue colored wave, its amplitude is much higher as compared with other three classes.

Form the above four figures of all classes; the amplitude variation is very obvious and it will assist the classifiers in classification task.

6.2 Results

The final classification achieved by implementing all the above mentioned classifiers on individual features can be visualized in the charts shown below.

6.2.1 Results (Standard Deviation)

In this result as shown in Fig.30, all the above mentioned classifiers are implemented on the EEG data when standard deviation is used as a feature.

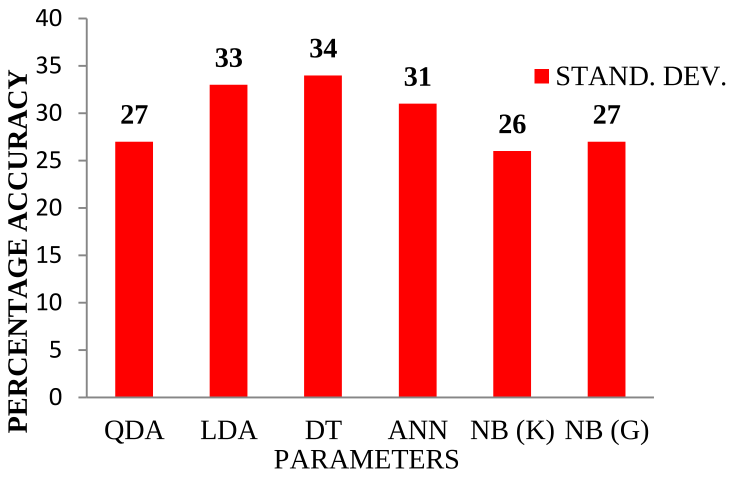


Fig.30. Results of classifiers when SD is used as a feature along with PCA

This results shows that by implementing the PCA the overall result has been distorted. Not even the single classifier is able to classify the data till half of the total percentage.

6.2.2 Results (Activity):

In this result as shown in Fig.31, all the above mentioned classifiers are implemented on the EEG data when activity is used as a feature.

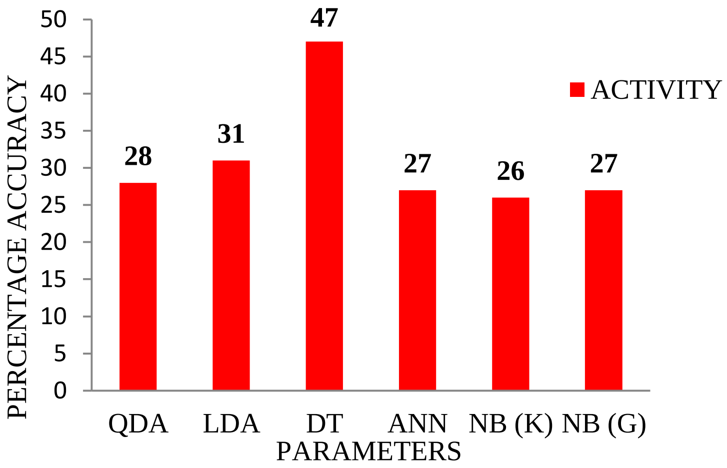


Fig.31. Results of classifiers when Activity is used as a feature along with PCA

In this result, only DT is able to classify the data but still at very low percentage.

6.2.3 Results (Mobility):

In this result as shown in Fig.32, all the above mentioned classifiers are implemented on the EEG data when mobility is used as a feature.

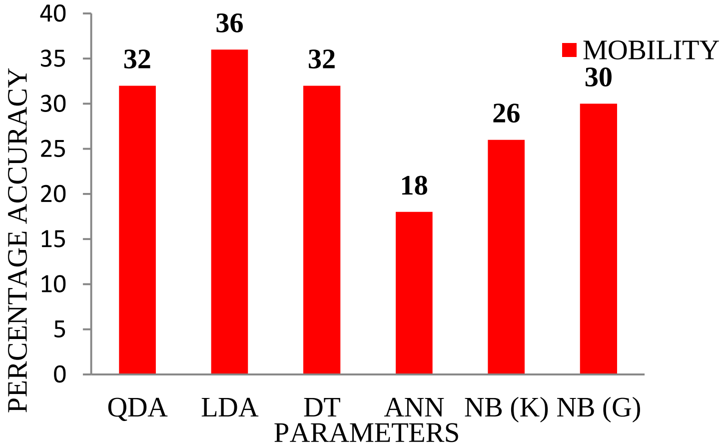


Fig.32. Results of classifiers when Mobility is used as a feature along with PCA

In this result, LDA classifies the data but still at very low percentage.

6.4 Results (Power Spectral Density):

In this result as shown in Fig.33, all the above mentioned classifiers are implemented on the EEG data when mobility is used as a feature.

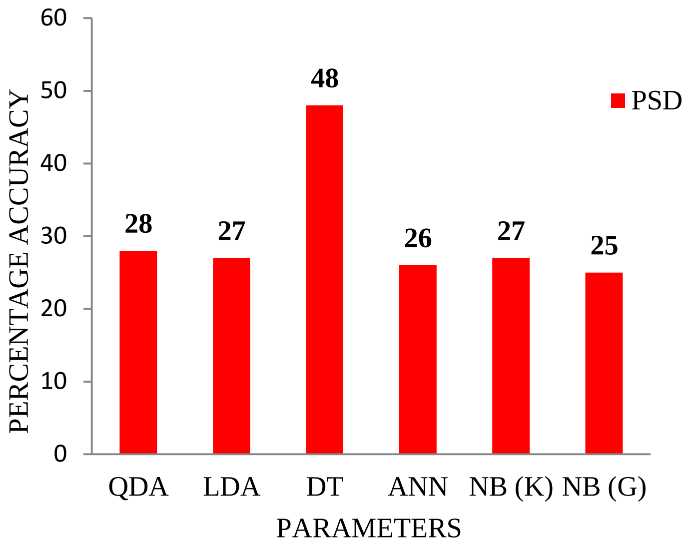


Fig.33. Results of classifiers when PSD is used as a feature along with PCA

Here D.T classfication performance is better than other classifiers.

CHAPTER 7: INTEGRATION OF TIME DOMAIN & FREQUENCY DOMAIN FEATURES

Integration of time domain & frequency domain features is a novel approach to accomplish the tasks mentioned in the objectives. In this approach the time domain feature (Standard Deviation) and frequency domain feature (Power Spectral Density) have been integrated before feeding to the classifiers. The block diagram of novel approach is shown in Fig.34.

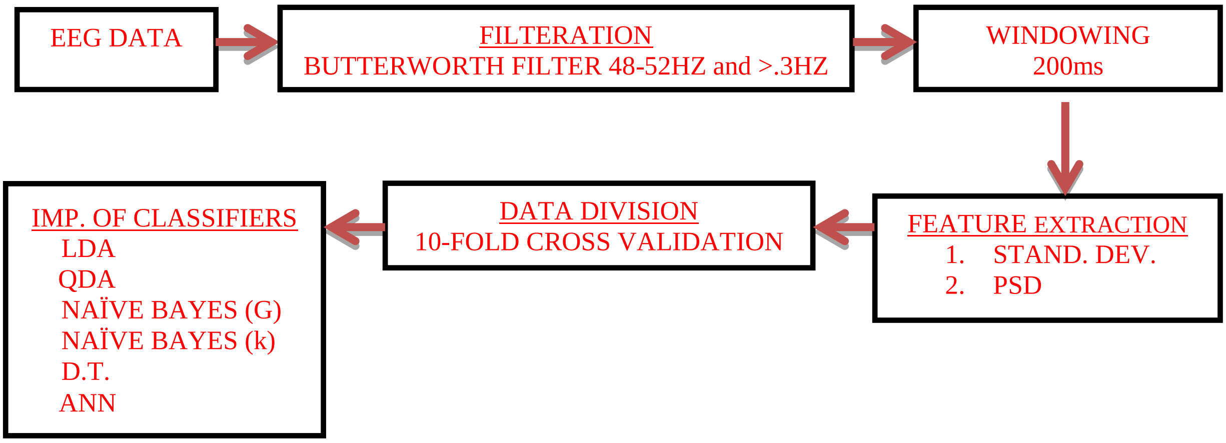


Fig.34. Block diagram of integration of time domain & frequency domain features

In this approch the EEG data, filteration ,windowing, data division, and classifiers are the same , the only difference lies in the integration of features.

7.1 Integration Of Features:

Indiviual time domain or frequency domain feature processing is a common practice in BCI. In this novel approach, we try to intregrate the two features of different domains. Now the classifers have more flexibility to discriminate among the classes. The dimensionality of the fed data to the classifiers has been changed. Now the each trail has dimensions of 60 rows and 19 columns. The dimensions of each class is 180 rows and 19 columns. The dimensions of total data is 720 rows and 19 columns. The feature eraction can be visulized in Fig.35.

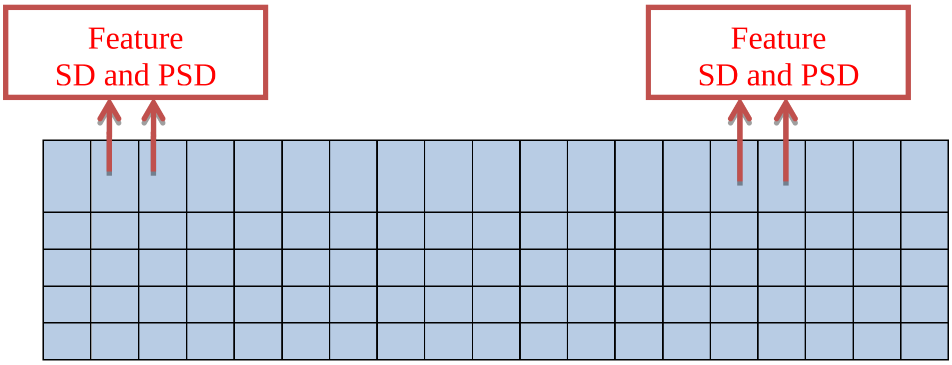


Fig.35. Feature exaction in novel approach

7.2 Results:

In this result as shown in Fig.36, all the above mentioned classifiers are implemented on the EEG data when Standard Deviation and Power Spectral Density is used as a feature.

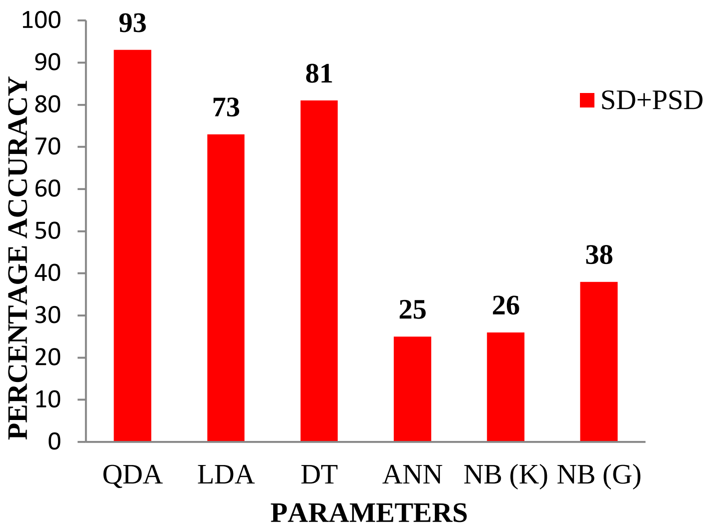


Fig.36. Results of classifiers when SD+PSD is used as a feature.

Some important conclusions of this result are following,

a. By integrating the frequency domain and time domain features the percentage accuracy has been increased drastically.
b. Individual SD accuracy (78%)
c. Individual PSD accuracy (82%)
d. Novel approach accuracy (93%)
e. The maximum change in accuracy is about 11%.

CHAPTER 8: SOFTWARE IMPLEMENTATION

The techniques that have been explained in previous chapters are implemented on MATLAB. The results and conclusions are being drawn on the basis of results received by the processing in MATLAB. The programs have been developed for individual approach.

8.1 Time Domain Analysis of EEG Signals

In this technique the EEG signals has been analyzed in time domain. In this approach first the data has been imported into the MATLAB for further processing. There are total four classes and each class has three trails. Each trial has 3008 rows and 19 columns. The step by step procedure adopted for this technique is following,

8.1.1 Filtration

To remove unwanted signals filtration has been done in MATLAB. For filtration a 6 order Butterworth Band Stop and High Pass filter has been implemented. For high pass the cutoff frequency is 0.3Hz and for band stop filter the cutoff frequency is from 48Hz to 52Hz.

By implementing Butterworth filter, the dimensionality of the individual data is still same only mentioned frequency components have been eliminated.

Commands to perform filtration using Butterworth filter.

```
[p,q]=butter(6,0.0006,'high');%high pass at.3Hz for removal DC baseline
[t,u]=butter(6,[0.096 0.104],'stop');%removes 48-52Hz AC Frequency
```

To perform filtration on the individual trail, following commands should be given,

```
% Filtration of Left Hand Backward Motion -1
% Here x1 is a variable used for LHBM-1for removal of DC baseline.
 v=filtfilt(p,q,x1);
% To remove AC frequency.
 y1=filtfilt(t,u,v);

Same code is implemented on all the remaining trails for above mentioned
artifacts.
 %  filteration of Left Hand Backward Motion -2
```

```
 v=filtfilt(p,q,x2);
y2=filtfilt(t,u,v);
%  filteration of Left Hand Backward Motion -3
 v=filtfilt(p,q,x3);
y3=filtfilt(t,u,v);
%  filteration of Left Hand Forward Motion -1
 v=filtfilt(p,q,x4);
y4=filtfilt(t,u,v);
%  filteration of Left Hand Forward Motion -2
v=filtfilt(p,q,x5);
y5=filtfilt(t,u,v);
%  filteration of Left Hand Forward Motion -3
v=filtfilt(p,q,x6);
y6=filtfilt(t,u,v);
%  filteration of Right Hand Backward Motion -1
v=filtfilt(p,q,x7);
y7=filtfilt(t,u,v);
%  filteration of Right Hand Backward Motion -2
v=filtfilt(p,q,x8);
y8=filtfilt(t,u,v);
%  filteration of Right Hand Backward Motion -3
v=filtfilt(p,q,x9);
y9=filtfilt(t,u,v);
%  filteration of Right Hand Forward Motion -1
v=filtfilt(p,q,x10);
y10=filtfilt(t,u,v);
%  filteration of Right Hand Forward Motion -2
v=filtfilt(p,q,x11);
y11=filtfilt(t,u,v);
%  filteration of Right Hand Forward Motion -3
v=filtfilt(p,q,x12);
y12=filtfilt(t,u,v);
```

8.1.2 Windowing

To include the one particular action for processing, a windowing method has been implemented. A window of size 200msec has been made. The windowing methods needs following MATLAB commands.

The windowing method implemented on individual trail. Each window has 100 rows and 1 column as shown in fig.37.

100x1	100x1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100x1	100x1
-	-	-	-	-	-	-	-	-	-	-	-		-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	--	--	-	-	-	-	-	-	-	-	-	-	--	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	--	-	-
100x1	100x1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100x1	100x1

Fig.37. Windows of Individual trials

Now individual column has 30 windows and in one trail we have 19 columns, so total numbers of rows are 30X19=570. Here we have 12 trails so total windows of whole data are 570X12=6840.

```
% Windowing
 n = 1;
% combining all 12 trails in one matrix. The data has 3008 rows and 228
columns.
 v = [y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 y11 y12];
 size(v)
%selecting first 19 columns of individual trail for making windows and this
loop runs 12 times to include all trails.
for i=1:19:228
% including 3008 rows of first 19 columns of the matrix v as mentioned
above.
     x=v(1:3008,i:i+18);
% selecting individual columns for windowing.
 for j=1:19
% making window of 100 rows of individual column.
q5=x(1:100,j); % 1 window
% saving the windows results in columns of matrix sal. This process
continues 30 times to include all the 3008 rows of individual columns and
saving in the matrix sal.
sal(:,n) = q5;
```

```
n = n+1;
q6=x(101:200,j); % 2 window
sal(:,n) = q6;
n = n+1;
q7=x(201:300,j); % 3 window
sal(:,n) = q7;
n = n+1;
q8=x(301:400,j); % 4 window
sal(:,n) = q8;
n = n+1;
q9=x(401:500,j); % 5 window
sal(:,n) = q9;
n = n+1;
q10=x(501:600,j); % 6 window
sal(:,n) = q10;
n = n+1;
q11=x(601:700,j); % 7 window
sal(:,n) = q11;
n = n+1;
q12=x(701:800,j); % 8 window
sal(:,n) = q12;
n = n+1;
q13=x(801:900,j); % 9 window
sal(:,n) = q13;
n = n+1;
q14=x(901:1000,j); % 10 window
sal(:,n) = q14;
n = n+1;
q15=x(1001:1100,j); % 11 window
sal(:,n) = q15;
n = n+1;
q16=x(1101:1200,j); % 12 window
sal(:,n) = q16;
n = n+1;
q17=x(1201:1300,j); % 13 window
sal(:,n) = q17;
n = n+1;
q18=x(1301:1400,j); % 14 window
sal(:,n) = q18;
n = n+1;
q19=x(1401:1500,j); % 15 window
```

```
sal(:,n) = q19;
n = n+1;
q20=x(1501:1600,j); % 16 window
sal(:,n) = q20;
n = n+1;
q21=x(1601:1700,j); % 17 window
sal(:,n) = q21;
n = n+1;
q22=x(1701:1800,j); % 18 window
sal(:,n) = q22;
n = n+1;
q23=x(1801:1900,j); % 19 window
sal(:,n) = q23;
n = n+1;
q24=x(1901:2000,j); % 20 window
sal(:,n) = q24;
n = n+1;
q25=x(2001:2100,j); % 21 window
sal(:,n) = q25;
n = n+1;
q26=x(2101:2200,j); % 22 window
sal(:,n) = q26;
n = n+1;
q27=x(2201:2300,j); % 23 window
sal(:,n) = q27;
n = n+1;
q28=x(2301:2400,j); % 24 window
sal(:,n) = q28;
n = n+1;
q29=x(2401:2500,j); % 25 window
sal(:,n) = q29;
n = n+1;
q30=x(2501:2600,j); % 26 window
sal(:,n) = q30;
n = n+1;
q31=x(2601:2700,j); % 27 window
sal(:,n) = q31;
n = n+1;
q32=x(2701:2800,j); % 28 window
sal(:,n) = q32;
n = n+1;
```

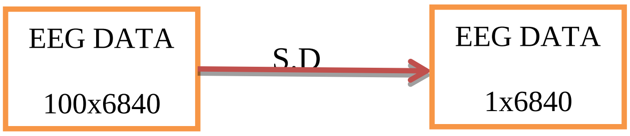
```
q33=x(2801:2900,j); % 29 window
sal(:,n) = q33;
n = n+1;
q34=x(2901:3000,j); % 30 window
sal(:,n) = q34;
n = n+1;
 end
 end
A=sal; after completion of all the loops, the results will be saved in sal,
and has dimensions of 100 rows and 6840 columns.
```

8.1.3 Feature Extraction

Feature represents characteristic of the signal and help in classification. The features have been extracted in time domain. The following mentioned commands will help to find out the features of the signal.

8.1.3.1 Standard Deviation:

After calculating the feature, the dimensions of the data is reduced as,



The standard deviation of individual window will be calculated as,

```
A1=std(A); % sal is saved in variable A
```

8.1.3.2 Mean:

Here again the dimensions will be changed from 100x6840 to 1x6840.

The mean of individual feature will be calculated as,

```
A1=mean(A);
```

8.1.3.3 Variance:

Here again the dimensions will be changed from 100x6840 to 1x6840.

The variance of individual feature will be calculated as,

```
A1=var(A)
```

8.1.3.4 Activity, Mobility and Complexity:

To calculate the activity, mobility and complexity you have to make following program in MATLAB.

```
[ACTIVITY, MOBILITY, COMPLEXITY,m0,m1,m2] = hjorth(A);

[N,K] = size(S);    % number of electrodes K, number of samples N

%m0 = mean(sumsq(S,2));
d0 = S;
%m1 = mean(sumsq(diff(S,[],1),2));
d1 = diff([zeros(1,K);S ],[],1);
d2 = diff([zeros(1,K);d1],[],1);

FLAG_ReplaceNaN = 0;

if nargin<2,
        UC = 0;
end;
if nargin<3;
        if UC==0,

        elseif UC>=1,
                B = ones(1,UC);
                A = UC;
        elseif UC<1,
                FLAG_ReplaceNaN = 1;
                B = UC;
                A = [1, UC-1];
        end;
else
        B = UC;
end;

if ~UC,
        m0 = mean(d0.^2);
        m1 = mean(d1.^2);
```

```
        m2 = mean(d2.^2);
else
        if FLAG_ReplaceNaN;
                d0(isnan(d0)) = 0;
                d1(isnan(d1)) = 0;
                d2(isnan(d2)) = 0;
        end;
        m0 = filter(B,A,d0.^2)./filter(B,A,(~isnan(d0)).^2);
        m1 = filter(B,A,d1.^2)./filter(B,A,(~isnan(d1)).^2);
        m2 = filter(B,A,d2.^2)./filter(B,A,(~isnan(d2)).^2);
end;

ACTIVITY    = sqrt(m0);
MOBILITY    = m1./m0;
tmp         = (m2./m1 - MOBILITY);
COMPLEXITY = tmp;
COMPLEXITY(tmp<0) = NaN;
COMPLEXITY = sqrt(COMPLEXITY);
MOBILITY    = sqrt(MOBILITY);
```

After feature selection, the featured data must be rearranged before feeding to the classifier,

```
n=1;
% For individual columns there are 30 values of 30 windows, so here we have total 228 columns so, this program runs 228 times.
for i=1:30:6840
    r=A1(:,i:i+29);
    sa(:,n)=r; % saving of value in sa variable
    n=n+1;
end
A2=sa; %final save in A2

w1=A2(:,1:19); %extracting first 19 columns and saving in w1 for LHBM-1
w2=A2(:,20:38); %extracting first 19 columns and saving in w2 for LHBM-2
w3=A2(:,39:57); %extracting first 19 columns and saving in w3 for LHBM-3
w4=A2(:,58:76); %extracting first 19 columns and saving in w4 for LHFM-1
w5=A2(:,77:95); %extracting first 19 columns and saving in w5 for LHFM-2
w6=A2(:,96:114); %extracting first 19 columns and saving in w6 for LHFM-3
w7=A2(:,115:133); %extracting first 19 columns and saving in w7 for RHBM-1
w8=A2(:,134:152); %extracting first 19 columns and saving in w8 for RHBM-2
```

```
w9=A2(:,153:171); %extracting first 19 columns and saving in w9 for RHBM-3
w10=A2(:,172:190);%extracting first 19 columns and saving in w10 for RHFM-1
w11=A2(:,191:209);%extracting first 19 columns and saving in w11 for RHFM-2
w12=A2(:,210:228);%extracting first 19 columns and saving in w12 for RHFM-3
```

The below mentioned command arranged the data as,

LHBM-1
LHBM-2
LHBM-3
LHFM-1
LHFM-2
LHFM-3
RHBM-1
RHBM-2
RHBM-3
RHFM-1
RHFM-2
RHBM-3

```
A3=[w1;w2;w3;w4;w5;w6;w7;w8;w9;w10;w11;w12]
```

The output cross ponding to each class is specified as,

```
t1=ones(1,90); % for Left Hand Backward Motion
t2=2*t1; %for Left Hand Forward Motion
t3=3*t1; %for Right Hand Backward Motion
t4=4*t1; %for Right Hand Forward Motion
Y=[t1,t2,t3,t4]; % data dimensions 1x360
```

8.1.4 Data Division

Before classification, 10-fold cross validation technique is implemented to divide the data in to training and testing.

```
rng(0,'twister');
C = cvpartition(Y,'k',10); % dividing output y into 10 folds
```

8.1.5 Classification

Classification helps to find out the user's intent. LDA, QDA, Naive Bayes, ANN, Decision Trees has been implemented on the data. The following commands should be written in MATLAB.

```
yorder = unique(Y);

%here xtr=xtrain= EEG data for training, ytr=ytrain= cross ponding output
used in training, xte=xtest= EEG data for testing, yte=ytest= cross ponding
output used in testing

% FOR LDA ANLYSIS
F=@(xtr,ytr,xte,yte)confusionmat(yte,classify(xte,xtr,ytr,'linear'),'order'
,yorder);
cfMat = crossval(f,X,Y,'partition',C);
cfMat1 = reshape(sum(cfMat),4,4) %arranging confusion matrix in 4x4 matrix.

% % FOR QDA ANALYSIS
F=@(xtrain,ytrain,xtest,ytest)confusionmat(ytest,classify(xtest,xtrain,ytra
in,'quadratic'),'order',yorder);
cfMat = crossval(f,X,Y,'partition',C);
cfMat2 = reshape(sum(cfMat),4,4) %arranging confusion matrix in 4x4 matrix.

% % % % FOR NAIVE BAYES WITH NORMAL DISTRIBUTION
F=@(xtrain,ytrain,xtest,ytest)confusionmat(ytest,predict(NaiveBayes.fit(xtr
ain,ytrain), xtest));
cfMat = crossval(f,X,Y,'partition',C);
cfMat3 = reshape(sum(cfMat),4,4) %arranging confusion matrix in 4x4 matrix.

% % % % FOR NAIVE BAYES WITH KERNAL DISTRIBUTION
F=@(xtrain,ytrain,xtest,ytest)confusionmat(ytest,predict(NaiveBayes.fit(xtr
ain,ytrain,'dist','kernel'), xtest));
cfMat = crossval(f,X,Y,'partition',C);
cfMat4 = reshape(sum(cfMat),4,4) %arranging confusion matrix in 4x4 matrix.

% % % % FOR DECISION TREE for classification
F=@(xtrain,ytrain,xtest,ytest)confusionmat(ytest,predict(ClassificationTree
.fit(xtrain,ytrain), xtest));
cfMat = crossval(f,X,Y,'partition',C)
cfMat5 = reshape(sum(cfMat),4,4) %arranging confusion matrix in 4x4 matrix.
out=cfMat5;
% % % % ANN for classification
% Extraction of 10 training and 10 testing data from 10-fold cross
validation method for one by one implementation of ANN.
trIdx1 = C.training(1); % extraction of first training data
```

```
trIdx2 = C.training(2); % extraction of second training data
trIdx3 = C.training(3); % extraction of third training data
trIdx4 = C.training(4); % extraction of fourth training data
trIdx5 = C.training(5); % extraction of fifth training data
trIdx6 = C.training(6); % extraction of sixth training data
trIdx7 = C.training(7); % extraction of seventh training data
trIdx8 = C.training(8); % extraction of eighth training data
trIdx9 = C.training(9); % extraction of ninth training data
trIdx10 = C.training(10); % extraction of tenth training data
teIdx1 = C.test(1); % extraction of first testing data
teIdx2 = C.test(2); % extraction of second testing data
teIdx3 = C.test(3); % extraction of third testing data
teIdx4 = C.test(4); % extraction of fourth testing data
teIdx5 = C.test(5); % extraction of fifth testing data
teIdx6 = C.test(6); % extraction of sixth testing data
teIdx7 = C.test(7); % extraction of seventh testing data
teIdx8 = C.test(8); % extraction of eighth testing data
teIdx9 = C.test(9); % extraction of ninth testing data
teIdx10 = C.test(10); % extraction of 10th testing data

% first fold
a=X(trIdx1,:); % saving the training value of EEG data of first fold in a
b=Y(trIdx1,:);  % saving the training value of cross ponding output of
                 First fold in b
a=a'; % taking transpose
b=b'; % taking transpose
size(a)% checking size of a
size(b) )% checking size of a
rand('seed', 491218382) % to eliminate the randomization of initial weights
                          and basis values
% feed forward architecture of ANN with three layers, 1st layer has 19
neurons, 2nd has 20 neurons and 3rd has 1 neuron along with activation
function and training algorithm.
net=newff(minmax(q1),[19 20 1],{'tansig','tansig','purelin'},'trainscg');
net.trainparam.epochs=2000; %number of epoches
net1=train(net,a,b); % training of ANN
out1=sim(net1,(X(teIdx1,:))'); %simulation for testing data
out=round(out1);
ldaResubCM1 = confusionmat((Y(teIdx1,:))',out) % calculation for confusion
matrix
% same steps will be repeated for remaining all 9 folds.
```

```
% second fold
a=X(trIdx2,:);
b=Y(trIdx2,:);
a=a';
b=b';
net1=train(net,a,b);
out1=sim(net1,(X(teIdx2,:))');
out=round(out1);
ldaResubCM2 = confusionmat((Y(teIdx2,:))',out)
% third fold
a=X(trIdx3,:);
b=Y(trIdx3,:);
a=a';
b=b';
net1=train(net,a,b);
out1=sim(net1,(X(teIdx3,:))');
out=round(out1);
ldaResubCM3 = confusionmat((Y(teIdx3,:))',out)
% fourth fold
a=X(trIdx4,:);
b=Y(trIdx4,:);
a=a';
b=b';
net1=train(net,a,b);
out1=sim(net1,(X(teIdx4,:))');
out=round(out1);
ldaResubCM4 = confusionmat((Y(teIdx4,:))',out)
% fifth fold
a=X(trIdx5,:);
b=Y(trIdx5,:);
a=a';
b=b';
net1=train(net,a,b);
out1=sim(net1,(X(teIdx5,:))');
out=round(out1);
ldaResubCM5 = confusionmat((Y(teIdx5,:))',out)
% sixth fold
a=X(trIdx6,:);
b=Y(trIdx6,:);
a=a';
b=b';
```

```
net1=train(net,a,b);
out1=sim(net1,(X(teIdx6,:))');
out=round(out1);
ldaResubCM6 = confusionmat((Y(teIdx6,:))',out)
% seventh fold
a=X(trIdx7,:);
b=Y(trIdx7,:);
a=a';
b=b';
net1=train(net,a,b);
out1=sim(net1,(X(teIdx7,:))');
out=round(out1);
ldaResubCM7 = confusionmat((Y(teIdx7,:))',out)
% Eighth fold
a=X(trIdx8,:);
b=Y(trIdx8,:);
a=a';
b=b';
net1=train(net,a,b);
out1=sim(net1,(X(teIdx8,:))');
out=round(out1);
ldaResubCM8 = confusionmat((Y(teIdx8,:))',out)
% Ninth fold
a=X(trIdx9,:);
b=Y(trIdx9,:);
a=a';
b=b';
net1=train(net,a,b);
out1=sim(net1,(X(teIdx9,:))');
out=round(out1);
ldaResubCM9 = confusionmat((Y(teIdx9,:))',out)
% 10th fold
a=X(trIdx10,:);
b=Y(trIdx10,:);
a=a';
b=b';
net1=train(net,a,b);
out1=sim(net1,(X(teIdx10,:))');
out=round(out1);
ldaResubCM10 = confusionmat((Y(teIdx10,:))',out)
```

For final result, combine the results of all 10 folds.

8.2 Frequency Domain Analysis of EEG Signals

In this technique the EEG signals has been analyzed in frequency domain. In this approach all the steps are same as mentioned in previous technique, the only difference is about the feature extraction in frequency domain.

8.2.1 Feature Extraction:

Feature represents characteristic of the signal and help in classification. The feature Power spectral Density has been extracted in frequency domain. The following mentioned commands will help to find out the feature of the signal.

```
n=1;
for i=1:6840 % total number of windows.
Fs=1010;  %Frequency twice than the sampling frequency
nfft=2^nextpow2(length(x));
Pxx=abs(fft(A(:,i),nfft)).^2/length(A(:,i))/Fs; %fast Fourier transform
Hpsd=dspdata.psd(Pxx(1:length(Pxx)/2),'Fs',Fs); % calculation of PSD
b=avgpower(Hpsd); % averaging of PSD
sad(:,n)=b; % saving in sad
n=n+1;
end
A4=sad;
```

8.3 TIME & FREQUENCY DOMAIN ANALYSIS OF EEG SIGNALS WITH DIMENSIONALITY REDUCTION

In this approach before feature exaction the dimensionality of data has been reduced by using Principal Component Analysis (PCA). The PCA analyzed data has 5 columns and rows are unchanged. The remaining all steps are same; the only difference is about the dimensionality reduction.

8.3.1 Dimensionality Reduction

For dimensionality reduction PCA has been implemented. The PCA reduces the columns from 19 to5. The numbers of rows are not being affected because of this implementation. For PCA following steps must be calculated

- Make sure data is zero mean.

- Compute covariance matrix.
- Perform Eigen decomposition of Covariance matrix.
- Sort eigenvectors in descending order.
- Apply mapping on the data
- Normalize in order to get eigenvectors of covariance matrix

```
function out=pcaasif(X,no_dims)
tic
 if ~exist('no_dims', 'var')
        no_dims = 2;
    end

    % zero mean
    mapping.mean = mean(X, 1);
    X = bsxfun(@minus, X, mapping.mean);

    % covariance matrix
    if size(X, 2) < size(X, 1)
        C = cov(X);
    else
        C = (1 / size(X, 1)) * (X * X')         % if N>D, we better use this
matrix for the eigendecomposition
    end

    % eigendecomposition of C
    C(isnan(C)) = 0;
    C(isinf(C)) = 0;
    [M, lambda] = eig(C);

    % eigenvectors in descending order
    [lambda, ind] = sort(diag(lambda), 'descend');
    if no_dims < 1
        no_dims = find(cumsum(lambda ./ sum(lambda)) >= no_dims, 1,
'first');
        disp(['Embedding into ' num2str(no_dims) ' dimensions.']);
    end
    if no_dims > size(M, 2)
        no_dims = size(M, 2);
```

```
warning(['Target dimensionality reduced to ' num2str(no_dims) '.']);
    end
    M = M(:,ind(1:no_dims));
    lambda = lambda(1:no_dims);

    % mapping on the data
    if ~(size(X, 2) < size(X, 1))
        M = bsxfun(@times, X' * M, (1 ./ sqrt(size(X, 1) .* lambda))');
% normalization
    end
    out = X * M;toc
```

For implementing PCA on EEG data, following commands should be mentioned in MATLAB,

EEG DATA (individual trail) 3008x19	$\xrightarrow{\text{PCA}}$	EEG DATA (individual trail) 3008x5

```
% PCA of Left Hand Backward Motion -1 with 5 columns
y1=pcaasif(y1,5);

% PCA of Left Hand Backward Motion -2 with 5 columns
y2=pcaasif(y2,5);

% PCA of Left Hand Backward Motion -3 with 5 columns
y3=pcaasif(y3,5);

% PCA of Left Hand Forward Motion -1 with 5 columns
y4=pcaasif(y4,5);

% PCA of Left Hand Forward Motion -2 with 5 columns
y5=pcaasif(y5,5);

% PCA of Left Hand Forward Motion -3 with 5 columns
y6=pcaasif(y6,5);

% PCA of Right Hand Backward Motion -1 with 5 columns
y7=pcaasif(y7,5);

% PCA of Right Hand Backward Motion -2 with 5 columns
y8=pcaasif(y8,5);
```

```
% PCA of Right Hand Backward Motion -3 with 5 columns
y9=pcaasif(y9,5);

% PCA of Right Hand Forward Motion -1 with 5 columns
y10=pcaasif(y10,5);

% PCA of Right Hand Forward Motion -2 with 5 columns
y11=pcaasif(y11,5);

% PCA of Right Hand Forward Motion -3 with 5 columns
y12=pcaasif(y12,5);
```

8.4 Integration of Time Domain & Frequency Domain Feature

Integration of time domain & frequency domain features is a novel. In this approach the time domain feature (Standard Deviation) and frequency domain feature (Power Spectral Density) have been integrated before feeding to the classifiers.


```
% calculate standard deviation

A1=std(A);

% calculate PSD

n=1;
 for i=1:6840 % total number of windows.
 Fs=1010;  %Frequency twice than the sampling frequency
 nfft=2^nextpow2(length(x));
 Pxx=abs(fft(A(:,i),nfft)).^2/length(A(:,i))/Fs; %fast Fourier transform
 Hpsd=dspdata.psd(Pxx(1:length(Pxx)/2),'Fs',Fs); % calculation of PSD
 b=avgpower(Hpsd); % averaging of PSD
 sad(:,n)=b; % saving in sad
 n=n+1;
 end
 A4=sad;
% integration

A= (A1; A4); % dimensions are 360x38
```

CHAPTER 9: COMPARISON OF DIFFERENT TECHNIQUES

The basic aim of the research is to identify a technique which will be best suitable for hardware implementation. To accomplish these tasks, four different approaches have been used as described in the previous chapters. The first approach deals with all the time domain features. The second approach deals with the frequency domain features. The third approach deals with time domain and frequency domain features along with PCA analysis of the data. The fourth approach is integration of time and frequency domain features. Some important results revealed by these approaches about the classifiers results are following,

In first approach, the classifiers have to predict the four classes by using the time domain information of the signal. By considering standard deviation as a feature, the ANN classifier produces better results than all the remaining classifiers. In remaining all the features QDA percentage accuracy is far better than other classifier, the one good reason behind this is the maximum data overlapping. This type of data requires quadratic decision boundaries for classification. By analyzing all the results, the QDA performance with feature activity is more than 90%, and this is maximum result shown in this approach with any mentioned features.

In second approach, the classifiers have to predict the four classes by using the frequency domain information of the signal. By considering Power Spectral Density as a feature, the decision trees predications about the classes are better than other classifiers because we know that EEG data is highly nonlinear and nonlinearity does not affect the performance of decision trees. The percentage accuracy of D.T is slightly greater than 80% which is very much less than the accuracy shown by the QDA in first approach.

In third approach, the PCA analysis of the EEG data has been done before extracting the features and feeding to the classifiers. In this approach the processing time of the data has been reduced drastically because number of columns has been reduced from 19 to 5. The percentage accuracies shown by the classifiers are very much poor and not even a single classifier is able to get accuracy greater than 50%. Here the dimensionality of the data has been reduced but the information of the signal is also lost.

In fourth approach, we try to integrate the features to get increase in the accuracy. In this approach standard deviation and power spectral density are collectively provided to the classifiers. The results have been changed drastically. Here again the QDA performance is better than the remaining all the classifiers because of requirement of quadratic decision boundary.

EEG data is highly non-linear and over lapping, so because of these two reasons, both LDA and Naive Bayes classifiers are fail to produce good results.

So we can summaries above points as,

a. Approach-1 Quadratic Discriminant Analysis (95%)
b. Approach-2- Decision Trees (82%)
c. Approach-3- Decision Trees (48%)
d. Approach-4- Quadratic Discriminant Analysis (93%)

Graphically these results can be represented in Fig.38.

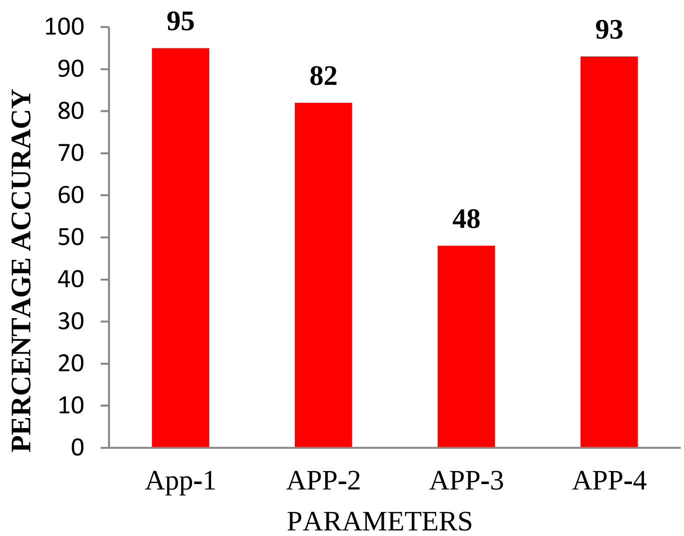


Fig.38. Results of all approaches

By analyzing the results, it is obvious that QDA performance is much better than Decision trees in time domain and integration approach. Decision Trees is best suitable for frequency domain approach but QDA also perform good in frequency domain but not better than Decision Trees.

The final comparison proves the superiority of QDA. So QDA is the best technique for the classification of upper limb motion EEG data.

CHAPTER 10: HARDWARE IMPLEMETATION

A 2-DOF robotic manipulator for upper limb has been designed in College of Electrical and Mechanical Engineering. This manipulator can open and close the hand. The real task is to control the motion of manipulator by using EEG signals that have been classified by the QDA as mentioned in previous chapter. The methodology adopted to control the motion is shown in Fig.39.

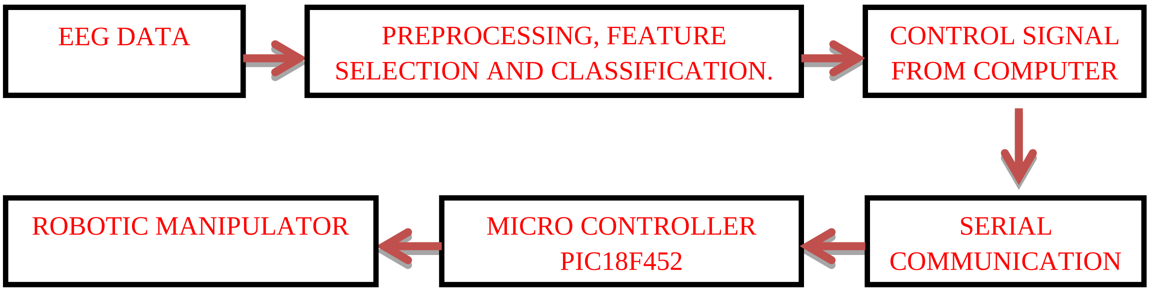


Fig.39. Control methodology of robotic manipulator

In this methodology the most efficient feature along with the classifier has been implemented to acquire the control signal from the computer. The signal from the computer is transferred serially to the micro controller PIC18F452. The control of the manipulator is connected to the microcontroller. The control signals perform the following actions as shown in Table-1

Table-1: Actions performed by the robotic manipulator

Sr. No.	**Signal**	**Signal transmitted to controller**	**Action by robotic manipulator**
01	Left Hand Backward Motion	1	Opening of the hand.
02	Right Hand Backward Motion	2	Opening of the hand.
03	Left Hand Forward Motion	3	Closing of the hand.
04	Right Hand Forward Motion	4	Closing of the hand.

10.1 Graphical User Interface

A GUI has been developed to facilitate the user to control the motion of a robotic manipulator. The GUI can be visualized in Fig. 40.

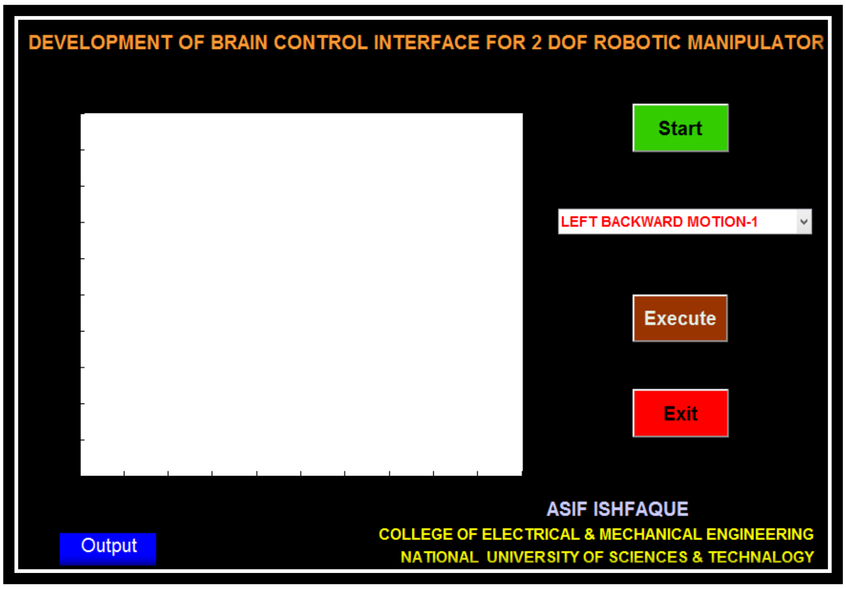


Fig.40. GUI of robotic manipulator

In this GUI the user first presses the Start button and options will appear at the Pop-up Menu. After that, the user will select the trail of motion from the Pop-up Menu and then press Execute; the control signal will be generated and then sent to the Microcontroller. The Exit will terminate the program. The Pop-up Menu has total twelve options three trails of Left Hand Backward Motion, three trails of Left Hand Forward Motion, three trails of Right Hand Backward Motion and tree trails of Right Hand Forward Motion.

The final implemented design can be visualized in Fig.41. In this implementation the control signals are provided to the manipulator for opening and closing of the hand. A bottle is grasped by the manipulator on the basis of signals provided to him.

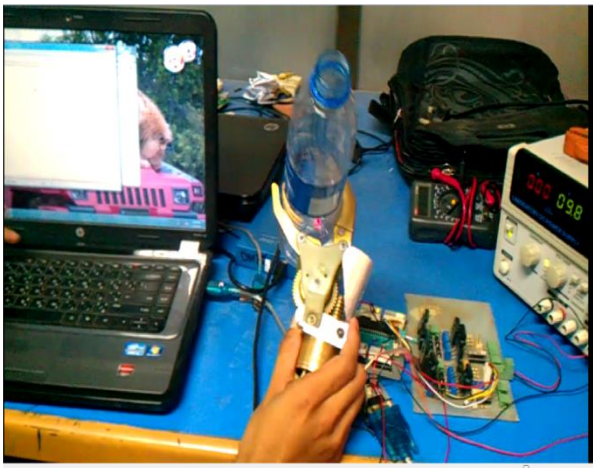


Fig.41. Final implemented design

REFERENCES

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-Computer Interfaces for Communication and Control," *clinical neurophysiology*, vol. 113, pp. 767-791, 2002.

[2] C. J. Lin and M. H. Hsieh, "Classification of mental task from EEG data using neural networks based on particle swarm optimization," *Neurocomputing*, vol. 72, pp. 1121-1130, 2009.

[3] O. G. Lins, T. W. Picton, P. Berg, and M. Scherg, "Ocular artifacts in recording EEGs and event-related potentials II: Source dipoles and Source Components," *Brain Topogr*, vol. 6, pp. 65-78, 1993.

[4] A. T. Boye, U. Q. Kristiansen, M. Billinger, O. F. d. Nascimento, and D. Farina, "Identification of movement-related cortical potentials with optimized spatial filtering and principal component analysis," *Biomedical Signal Processing and Control*, vol. 3, no. 4, p. 300–304, Oct. 2008.

[5] K. Li, R. Sankar, Y. Arbel, and E. Donchin, "Single trial independent component analysis for P300 BCI system," in , USA, 2009, pp. 4035-4038.

[6] N. P. Castellanos and V. A. Makarov, "Recovering EEG brain signals: Artifact suppression with wavelet enhanced independent component analysis.," vol. 158, pp. 300-312, 2006.

[7] J. F. Gao, Y. Yang, P. Lin, P. Wang, and C. X. Zheng, "Automatic removal of eye-movement and blink artifacts from EEG signals," *Brain Topogr*, vol. 3, pp. 105-114, 2010.

[8] A. Flexer, H. Bauer, J. Pripfl, and G. Dorffner, "Using ICA for removal of ocular artifacts in EEG recorded from blind subjects.," *Neural Networks*, vol. 18, pp. 998-1005, 2005.

[9] R. Yang, D. A. Gray, B. W. Ng, and H. Mingyi, "Comparative Analysis of Signal Processing in Brain Computer Interface," in , China, 2009, pp. 580-585.

[10] S. Lemm, C. Schafer, and G. Curio, "Probabilistic modeling of sensorimotor mu rhythms for classification of imaginary hand movements," in , 2004, pp. 1077-1080.

[11] G. Pires, M. C. Branco, and U. Nunes, "Visual P300-Based BCI to Steer a Wheelchair: A Bayesian Approach.," in , Canada, 2008, pp. 658-661.

[12] D. Garrett, D. A. Peterson, C. W. Anderson, and M. H. Thaut, "Comparison of linear, nonlinear, and feature selection methods for EEG signal classification," *IEEE Transaction, Neural System Rehabilitation Eng.*, vol. 11, pp. 141-144, 2003.

[13] R. Scherer, G. R. Muller, C. Neuper, B. Graimann, and G. Pfurtscheller, "An asynchronously controlled EEG-based virtual keyboard: Improvement of the spelling rate," *IEEE Transactions , Biomedical Eng.*, vol. 51, pp. 979-984, 2004.

[14] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K. R. Mulller, "Single-trial analysis and classification of ERP components—A tutorial," *Neuroimage*, vol. 56, pp. 814-825, 2011.

[15] C. Vidaurre, N. Kramer, B. Blankertz, and A. Schlogl, "Time domain parameters as a feature for EEG-based brain-computer interfaces," *Neural Networks*, vol. 22, pp. 1313-1319, 2009.

[16] A. Schlogl, F. Lee, H. Bischof, and G. Pfurtscheller, "Characterization of four-class motor imagery EEG data for the BCI-competition 2005," *J.Neural Eng.*, vol. 2, p. L14, 2005.

[17] G. N. Garcia, T. Ebrahimi, and J. M. Vesin, "Support Vector EEG Classification in the Fourier and Time-Frequency Correlation Domains," in *NER'03*, Capri Island, 2003, pp. 591-594.

[18] B. Blankertz, G. Curio, and K. Müller, "Classifying Single Trial EEG: Towards Brain Computer Interfacing," in *ANPS'02*, Vancouver, 2002, pp. 157-164.

[19] Z. Haihong, G. Cuntai, and W. Chuanchu, "Asynchronous P300-based brain--computer interfaces: A computational approach with statistical models," *Biomed. Eng.*, vol. 55, pp. 1754-1763, 2008.

[20] J. F. Borisoff, S. G. Mason, A. Bashashati, and G. E. Birch, "Brain-computer interface design for asynchronous control applications: Improvements to the LF-ASD asynchronous brain switch," *Biomed. Eng.*, vol. 51, pp. 985-992, 2004.

[21] T. Kayikcioglu and O. Aydemir, "A polynomial fitting and k-NN based approach for improving classification of motor imagery BCI data.," *Pattern Recogn. Lett.*, vol. 31, pp. 1207-1215, 2010.

[22] E. Haselsteiner and G. Pfurtscheller, "Using time-dependent neural networks for EEG classification," *Rehabil. Eng.*, vol. 8, pp. 457-463, 2000.

[23] D. Coyle, T. M. McGinnity, and G. Prasad, "Improving the separability of multiple EEG features for a BCI by neural-time-series-prediction-preprocessing.," *Biomed. Signal Process. Control* , vol. 5, pp. 196-204, 2010.

[24] M. K. Hazrati and A. Erfanian, "An online EEG-based brain-computer interface for controlling hand grasp using an adaptive probabilistic neural network," *Med. Eng. Phys.*, vol. 32, pp. 730-739, 2010.

[25] G. Pfurtscheller , C. Neuper , A. Schlogl, and K. Lugger , "Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters," *Rehabil. Eng.*, vol. 6, pp. 316-325, 1998.

[26] W. Ting, Y. Guo-zheng, y. Bang-hua, and S. Hong, "EEG feature extraction based on wavelet packet decomposition for brain computer interface," *Measurement*, vol. 41, pp. 618-625, 2008.

[27] Pfurtscheller, G. and Neuper, C., "Motor Imagery and Direct Brain Computer Communication," *Proceedings of the IEEE*, vol. 89, no. 7, pp. 1123-1134, 2001.

[28] Scott, P. D., "Hidden Markov Models (Lecture Notes CE-802)," *Department of Computer Science, University of Essex*, 2008.