

# General-Purpose Brain-Computer Interface

by

Nazih Shoura

A thesis submitted in conformity with the requirements  
for the degree of Mechatronics Engineering BEng (Hons)

Department of Electrical and Electronic Engineering  
University of Nottingham

© Copyright by Nazih Shoura 2014

# General-Purpose Brain-Computer Interface

Nazih Shoura

Mechatronics Engineering BEng (Hons)

Department of Electrical and Electronic Engineering  
University of Nottingham

2014

## Abstract

This research explores the effectiveness of different combinations of preprocessing and feature extraction technics in identifying mental activities representing right/left hand movement using Electroencephalography (EEG) data. The EEG signals used were recorder for a subject during left and right hand movement through 19 channels (electrodes) at 500Hz. However, only channel C4 and C3 were used in the classification.

A three-layered feedforward artificial neural network, which implements the backpropagation of error learning algorithm, was used to preform the classification in the four methods that were tested. The first one obtained a success rate of 68.4% by feeding the signals resulted from separating Mu brainwaves using FIR filter. Preprocessing in the rest of the methods was done by employing the FIR filter to clean the EEG from deferent kind of artifacts. The absolute values of the Fourier transform of the EEG were used as the input to the classifier in the second method resulting in an accuracy of 81.6%. Whereas in the third method, the classifier used the energy, variance, and waveform of the wavelet decomposition for ten levels to achieve a 63.2% success rate. Finally, the forth method revealed the superior success rate of 94.7% by classifying the coefficient of the ten levels decomposition.

## Acknowledgments

I wish to express my sincere gratitude to my supervisor Dr. Anandan. You have been like a father to me. The wisdom you gifted is priceless.

To my father who never showed me how to give up.

To my mother who's heart whelming support was with me every step in the way.

To that stupid girl who triggered a never-ending series of changes in me.

LOVE YOU ALL!

# Table of Contents

<b>Acknowledgments .....</b>	<b>ii</b>
<b>Chapter 1 Preliminary .....</b>	<b>1</b>
<b>    1    Introduction .....</b>	<b>1</b>
<b>    2    Motivation .....</b>	<b>2</b>
<b>    3    Objectives .....</b>	<b>3</b>
<b>Chapter 2 Literature Review.....</b>	<b>4</b>
<b>    1    Electroencephalography principles .....</b>	<b>4</b>
<b>        1.1    The EEG signals.....</b>	<b>4</b>
<b>        1.2    EEG wave groups.....</b>	<b>7</b>
<b>    2    EEG Signal Preprocessing.....</b>	<b>10</b>
<b>        2.1    Introduction .....</b>	<b>10</b>
<b>        2.2    EEG artifacts removal using Independent component analysis .....</b>	<b>12</b>
<b>        2.3    Artifact removal using peak elimination.....</b>	<b>13</b>
<b>        2.4    Recognition of blinking artifact by means of neural network.....</b>	<b>14</b>
<b>        2.5    Artifact rejection using bandpass Finite impulse response filters .....</b>	<b>15</b>
<b>    3    EEG feature extraction and Classification .....</b>	<b>16</b>
<b>        3.1    Feature extraction.....</b>	<b>16</b>
<b>        3.2    Classification methods.....</b>	<b>17</b>
<b>            3.2.1    Linear classifiers .....</b>	<b>17</b>
<b>            3.2.2    Linear Discriminant Analysis .....</b>	<b>17</b>
<b>            3.2.3    Support Vector Machine .....</b>	<b>19</b>
<b>            3.2.4    Artificial Neural Networks.....</b>	<b>20</b>
<b>            3.2.5    Nonlinear Bayesian classifiers .....</b>	<b>24</b>
<b>            3.2.6    Nearest Neighbor classifiers .....</b>	<b>25</b>
<b>Chapter 3 Methodology and Results.....</b>	<b>26</b>
<b>    1    Summary.....</b>	<b>26</b>
<b>    2    Tools and data used .....</b>	<b>27</b>
<b>        2.1    The EEG data set.....</b>	<b>27</b>

<b>2.2 Matlab .....</b>	<b>28</b>
<b>2.3 EEGLAB .....</b>	<b>28</b>
<b>2.4 Neural Networks toolbox .....</b>	<b>29</b>
<b>3 Procedure common in all classification methods attempted .....</b>	<b>29</b>
<b>3.1 Channel Selection.....</b>	<b>29</b>
<b>3.2 Signal segmentation.....</b>	<b>29</b>
<b>3.3 Classification .....</b>	<b>30</b>
<b>4 Classification methods attempted .....</b>	<b>31</b>
<b>4.1 First method .....</b>	<b>31</b>
4.1.1 Preprocessing.....	31
4.1.2 Feature extraction .....	33
4.1.3 Results .....	34
<b>4.2 Second method .....</b>	<b>35</b>
4.2.1 Preprocessing.....	35
4.2.2 Feature extraction .....	38
4.2.3 Results .....	39
<b>4.3 Third method .....</b>	<b>40</b>
4.3.1 Preprocessing.....	40
4.3.2 Feature extraction .....	40
4.3.3 Results .....	40
<b>4.4 Fort method .....</b>	<b>41</b>
4.4.1 Preprocessing.....	41
4.4.2 Feature extraction .....	41
4.4.3 Results .....	41
<b>Chapter 4 Discussions .....</b>	<b>42</b>
<b>4.1 First method .....</b>	<b>42</b>
<b>4.2 Second method .....</b>	<b>43</b>
<b>4.3 Third method .....</b>	<b>45</b>
<b>4.4 Forth method.....</b>	<b>47</b>
<b>Chapter 5 Conclusions.....</b>	<b>48</b>
<b>Chapter 6 Future development.....</b>	<b>49</b>
<b>References.....</b>	<b>50</b>

<b>Appendices.....</b>	<b>54</b>
<b>1 Appendix A: Backpropagation.....</b>	<b>54</b>
<b>2 Appendix B: Fourier transform.....</b>	<b>58</b>
<b>3 Appendix C: Wavelet transform.....</b>	<b>65</b>

## List of figures

Figure 1. Electrodes placement.....	4
Figure 2. Ten seconds of EEG signals for a subject preforming a multiplication task.....	5
Figure 3. Recording of EEG .....	6
Figure 4. Alpha and Beta waves and their frequency spectrum. ....	7
Figure 5. Theta waves and its frequency spectrum.....	8
Figure 6. Delta waves and its frequency spectrum.....	8
Figure 7. Mu waves and its frequency spectrum on left, Alpha waves and its frequency spectrum on the right. ....	9
Figure 8. The Cerebral hemispheres showing towards the front the motor areas.....	9
Figure 9. Movement of the eyelid or the eye (EOG) .....	11
Figure 10. Proposed system scheme. ....	15
Figure 11. Linear Discriminant separating hyperplane.....	17
Figure 12. SVM hyperplane.....	19
Figure 13. Illustration of a perceptron neural network with three layers.....	21
Figure 14. Illustration of backpropagation algorithm where solid lines represent forward propagation and dashed lines represents back propagation. ....	22
Figure 15. A summary of the methodologies.....	26
Figure 16. Frequency spectra of left hand movement.....	27
Figure 17. Frequency spectra of right hand movement. ....	28
Figure 18. FIR lowpass filter at 8 Hz.....	31

Figure 19. FIR high filter at 12 Hz .....	32
Figure 20. Frequency spectra of left hand movement after applying a FIR bandpass filter from 8Hz to 12Hz. ....	32
Figure 21. frequency spectra of right hand movement after applying a FIR bandpass filter from 8Hz to 12Hz. ....	33
Figure 22. Confusion of first method.....	34
Figure 23. FIR lowpass filter at 1Hz.....	35
Figure 24. FIR high filter at 90 Hz. ....	36
Figure 25. FIR Notch filter between 48 and 52 Hz.....	37
Figure 26. frequency spectra of left hand movement after applying a FIR bandpass filter from 1 Hz to 90 Hz and a FIR Notch filter between 48 and 52 Hz .....	37
Figure 27. Frequency spectra of right hand movement after applying a FIR bandpass filter from 1Hz to 90Hz and a FIR Notch filter between 48 and 52 Hz. ....	38
Figure 28. Confusion of the second method. ....	39
Figure 29. Confusion of the forth method. ....	40
Figure 30. Confusion of fifth method. ....	41
Figure 31. Noise and ringing artifacts caused by the filtering process. ....	42
Figure 32. Confusion for the classification using the absolute values of the FFT of the EEG data when segmented for ten seconds.....	44
Figure 33. Scaling and wavelet functions. ....	45
Figure 34. Amplitudes of the frequency spectra of the Fig.33 function .....	46

# Chapter 1

## Preliminary

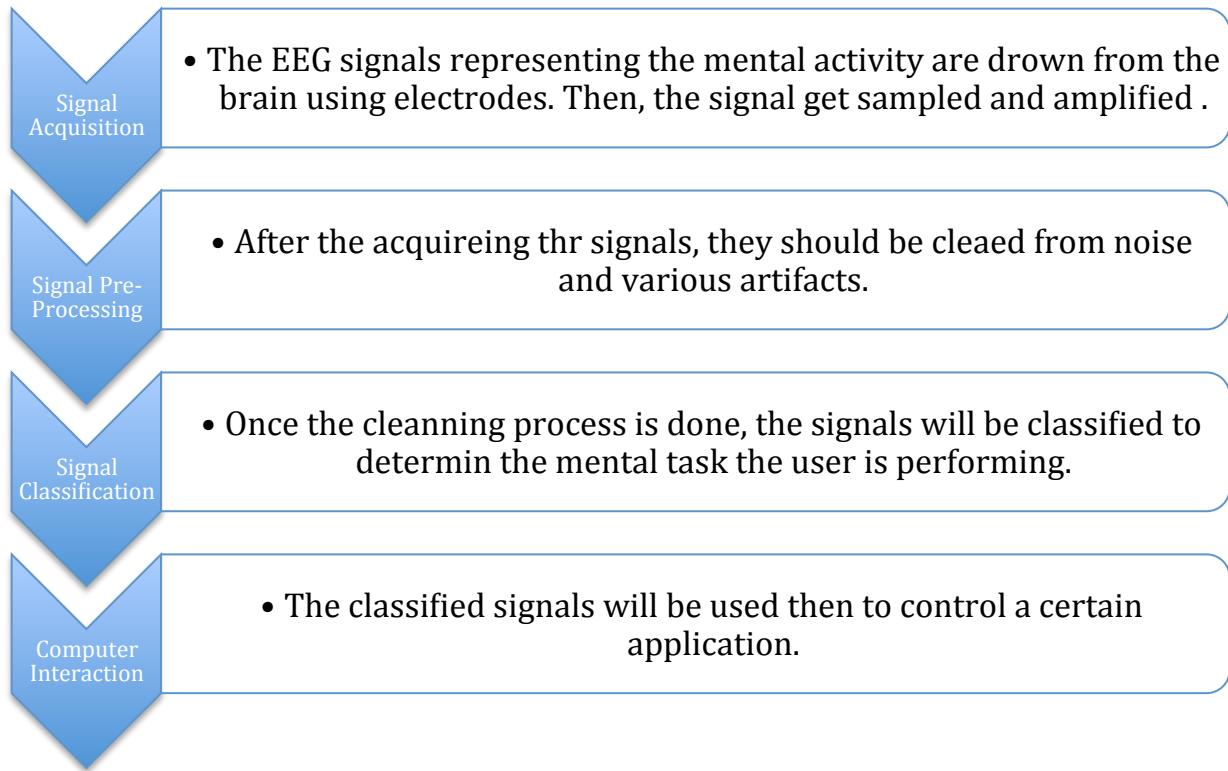
### 1 Introduction

Brain activity produces electrical signals that can be detected from the scalp known as electroencephalogram (EEG). A Brain–Computer Interfaces (BCIs) system can use these signals to provide a direct communication channel between the brain and the external world by transforming them from a mere reflections of brain activity into computer commands that convey the user's intent to the outside world. The process of transforming the signals is done by automatic classification of EEGs.

EEGs are very complex biological signals; the brain's intentions should be discernible in the spontaneous EEG. However, in practice, the high noise levels due to the low conductivity of the human skull, the vast number of electrically active neuronal elements, the complex geometry of the brain and head, and the disconcerting trial-to-trial variability in brain operations severely limit the information discernible. However, different methods have been developed to identify and partially remove the various EEG artifacts. Nonetheless, These problems makes automatically classifying EEGs a rather challenging task.

Nowadays, automatic classification of EEGs is performed using adaptive tools. One of these adaptive tools, which is widely used, is artificial neural networks (ANNs). ANN is basically a set of structured nodes, each one simulating, in a very simplified way, the behavior of a biological neuron. The rules governing neuron behavior are very simple but they result in a powerful classification tool. An ANN, therefore, acts like an “intelligent” filter, capable of recognize given patterns, which can also deviate from the original examples learnt by the network.

The common structure used in constructing Brain Computer Interfaces is demonstrated in the following:



## 2 Motivation

Patients with severe movement disorders need alternative means to transfer their thoughts. These mainly concerns patients who suffer from brainstem stroke, severe polyneuropathy, or end-stage amyotrophic lateral sclerosis, lack control over their muscles, and are not predisposed to use established augmentative technologies due to the fact that these kind of technologies require at least some degree of muscle function. Moreover, paralyzed people or patient with other movement disorders also need alternative ways to control and function. But, as mentioned above, modern disposable means of augmentative communication need at least low level of muscle control to operate them. Using a single group of muscle to carry out functions of another group (for example, use some of the ocular muscles to operate the speech) pass over different sorts of interruptions in normal passageways (for example, activate muscles on shoulders to manipulate the muscles of the hands and forearms [1.3]). Therefore, these techniques prove useless for

mentioned cases. The only acceptable solution for these people is to transfer their intents without relying on physical movement. Hence, they require a method of communicating their wants and wishes that does imply the usage brain's activities pathways of peripheral muscles and nerves.

This project has the potential to help such people by offering the patients alternative means of communication that does not need any muscles functionality. The users will have the ability to send “on” and “off” commands by only imagining moving there right and left hand. This way, they can answer simple yes or no questions and operate basic machines.

### 3 Objectives

The use of EEG signals as a vector of communication between men and machines represents one of the current challenges in signal theory research. The principal element of such a communication system, more known as “Brain Computer Interface”, is the interpretation of the EEG signals related to the characteristic parameters of brain electrical activity.

The role of signal processing is crucial in the development of a real-time Brain Computer Interface. Until recently, several improvements have been made in this area, but none of them have been successful enough to use them in a real system. The goal of creating more effective classification algorithms, have focused numerous investigations in the search of new techniques of feature extraction.

The main objective of this project is to implement different methods of preprocessing and feature extraction that enable the classification EEG signals between left and right hand movements and explore the theories behind them as well as analyze there performance.

## Chapter 2

### Literature Review

# 1 Electroencephalography principles

## 1.1 The EEG signals

It is a well known fact that brain activities generates an ionic current that flows through the brain neurons - due to the electrical nature that the human nervous system has. Voltage fluctuations will emerge to the scalp as a result of this current [2.1]. Recording those fluctuations can be done by affixing electrodes to different sites on the scalp according to a certain configuration, and measuring the potential difference between pairs of the electrodes used. The resulting voltage signals are at the microvolt level. Therefore, at this stage, the signals are usually carried to amplifiers to be magnified approximately ten thousand times to produce what is called Electroencephalography (EEG). Sites for recording EEG are illustrated in Fig.1 waveforms consisting of a 10 second recording of EEG from six recording channels are shown.

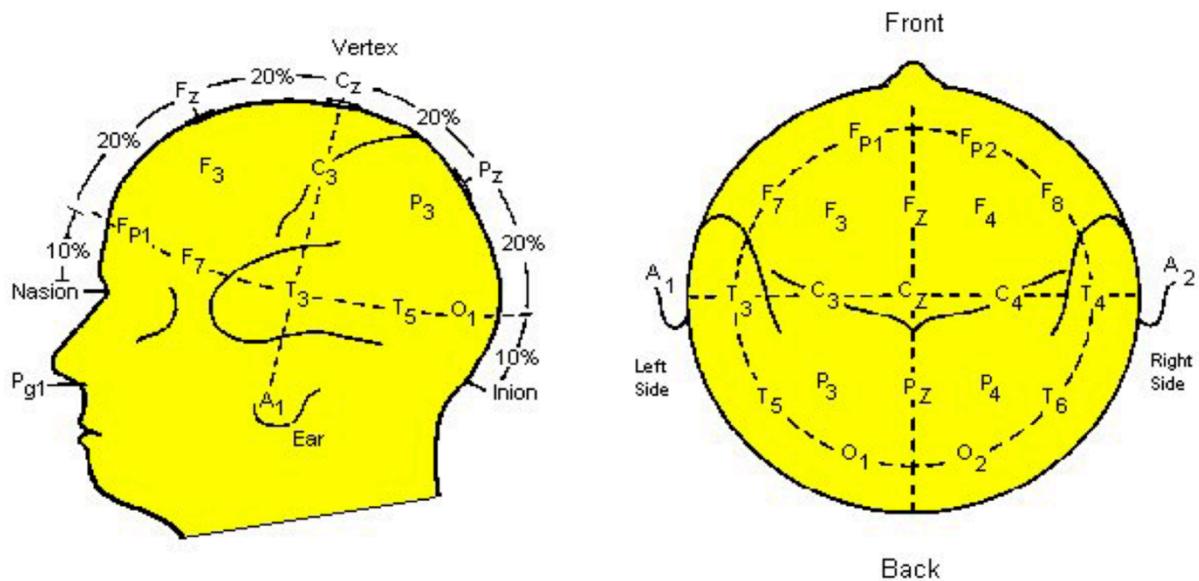


Figure 1. Electrodes placement.

To distinguish the sites on the scalp, the first letter of lobe name was assigned to the respective site where the letters ‘T’, ‘P’, ‘C’, ‘O’, and ‘F’ stand for Temporal, Parietal, Central, Occipital and Frontal. Also, a number or the letter ‘z’ is used to identify the hemisphere. Odd numbers (1,3,5,7) and Even numbers (2,4,6,8) are used to refer to the left and right hemisphere respectively, whereas the sites on the midline are referred to by the letter ‘z’.

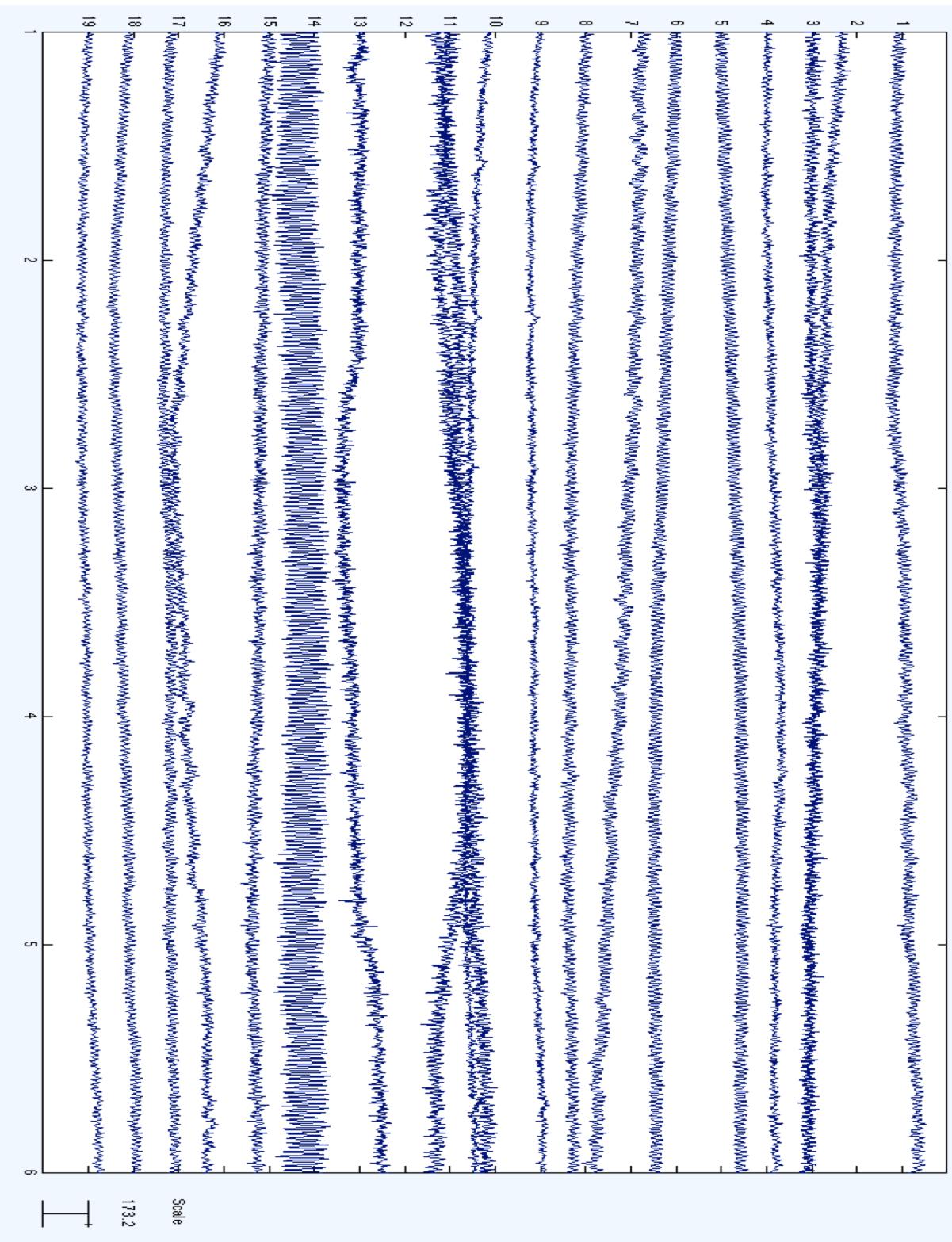


Figure 2. Ten seconds of EEG signals for a subject preforming a multiplication task.

Since the signal on the surface of the scalp is quite weak, the conductivity of the electrodes is of paramount importance. That is the reason why the electrodes are made of materials that has high conductivity such as silver chloride or gold. Moreover, a conductive gel is put on the subject scalp to maximize the signal to noise ratio. Fig 3 shows the electrodes being used to record the EEG.

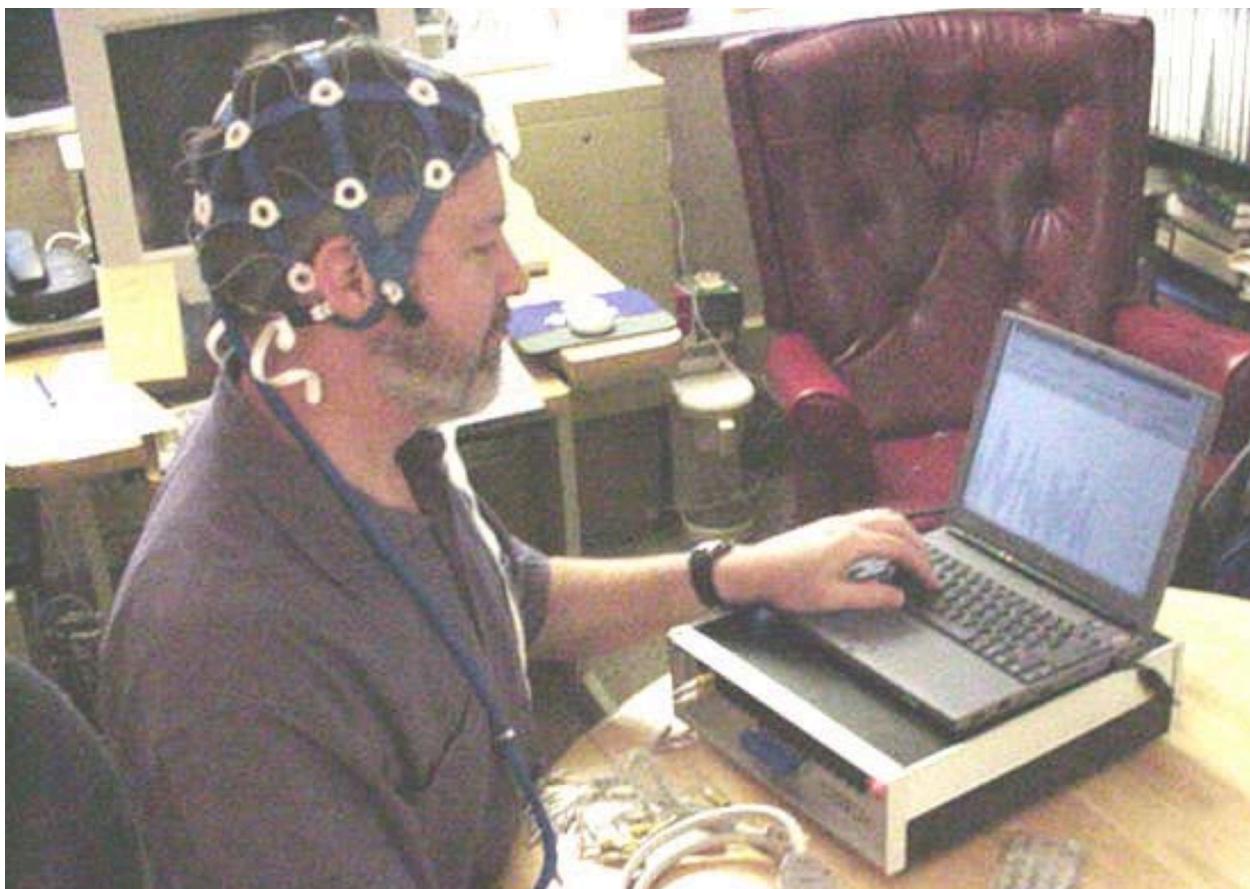


Figure 3. Recording of EEG.

## 1.2 EEG wave groups

By observing EEG of subjects while performing different tasks, it was found that different brain waves are linked to a different state of consciousness and actions. These waves are categorized by their frequencies. Although these brain waves are never generated alone, consciousness and actions make a certain band of frequency more noticeable than the others. Particularly, Six different types of waves [2.3] can be defined in the signals of the brain:

Alpha: The Alpha waves are located in the frequency range of 8 and 12 Hz, with voltage amplitude of 30-50  $\mu$ V (Fig.4 left). Best defined over the occipital and frontal cortex. They are associated with inattention and can be reduced or eliminated by hearing unfamiliar sounds, mental concentration or anxiety.

Beta: The beta waves are located in the frequency band of 12 to 30 Hz, with low voltage amplitude of 5-30  $\mu$ V (Fig.4 right). Best defined over the frontal and central cortex. They are associated with focused concentration and active thinking. During intense mental activity, when the mental activity becomes intense, frequencies of beta waves can reach about 50 Hz.

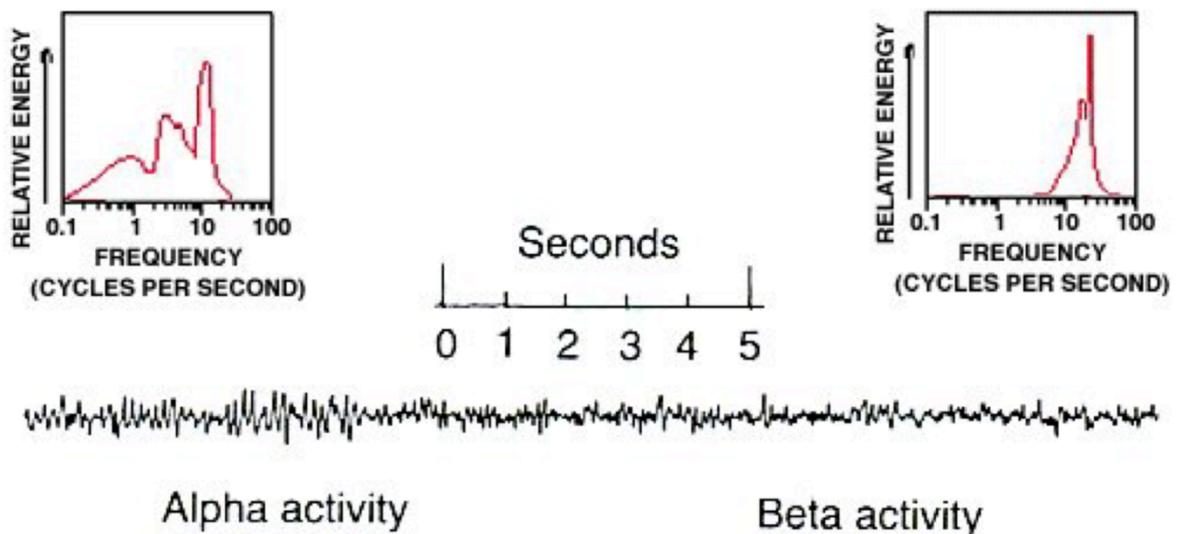


Figure 4. Alpha and Beta waves and their frequency spectrum.

Theta: They are in the frequency band of 4 to 7 Hz, their voltage amplitude is usually greater above 20  $\mu$ V (Fig.5). They are associated with emotional stress, particularly disappointment or frustration. Also associated with deep meditation, creative inspiration, unconscious material, and daydreaming.

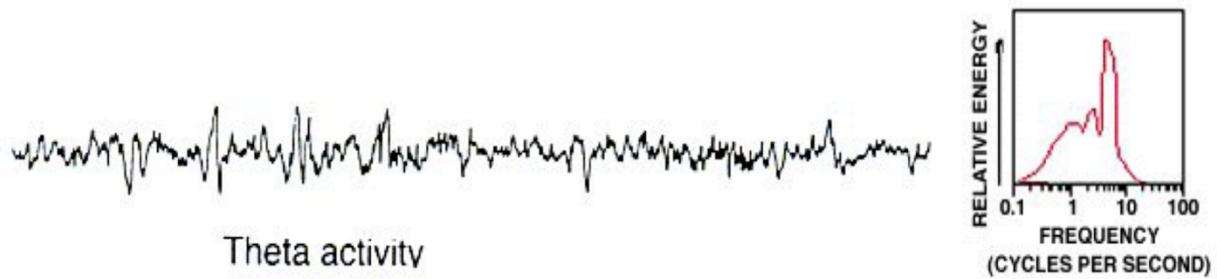


Figure 5. Theta waves and its frequency spectrum.

Delta: The Delta waves are in the frequency band of 0.5 to 3 Hz, the amplitude of their voltage is variable (Fig.6). Associated with deep sleep. It is thought that the occurrence of delta waves during state of awakens indicates physical defects in the human brain.

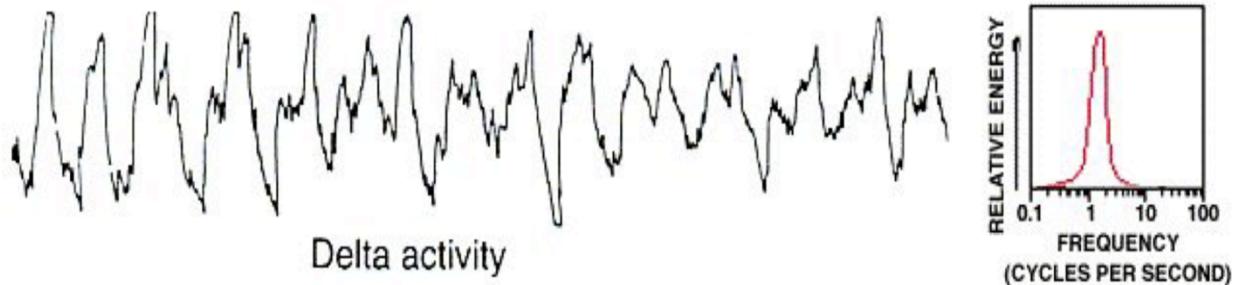


Figure 6. Delta waves and its frequency spectrum.

GAMMA: The Gamma waves are located in the frequency range of 26 Hz onward. They are associated with consciousness mechanism - the attachment of the coherent percepts that are capable of acting in a re-entrant.

Mu: The Mu waves are found in the frequency range of 8 to 12 Hz. Best defined over the motor cortex (Fig.7). They are associated with motor activities. This includes actual movement and imaginary movement. Although mu waves and alpha waves share the same frequency band (Fig. 8), Alpha waves is recorded over occipital cortex. Most Brain computer interfaces uses Mu waves because by applying mental effort like raising the level of attention or by recalling strongly stimulating image.

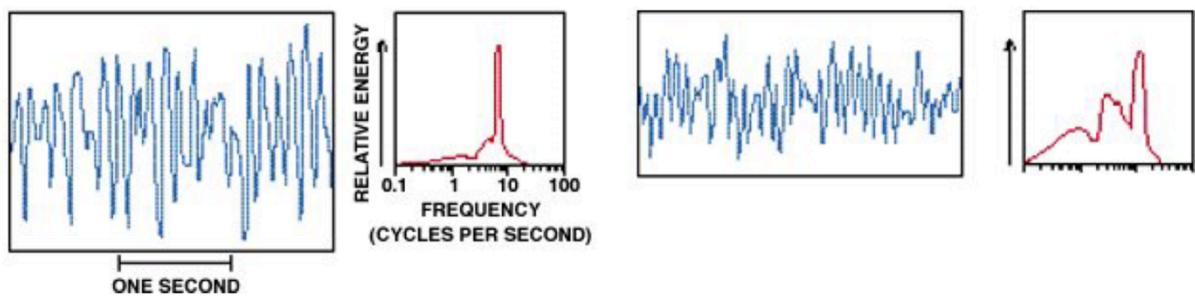


Figure 7. Mu waves and its frequency spectrum on left, Alpha waves and its frequency spectrum on the right.

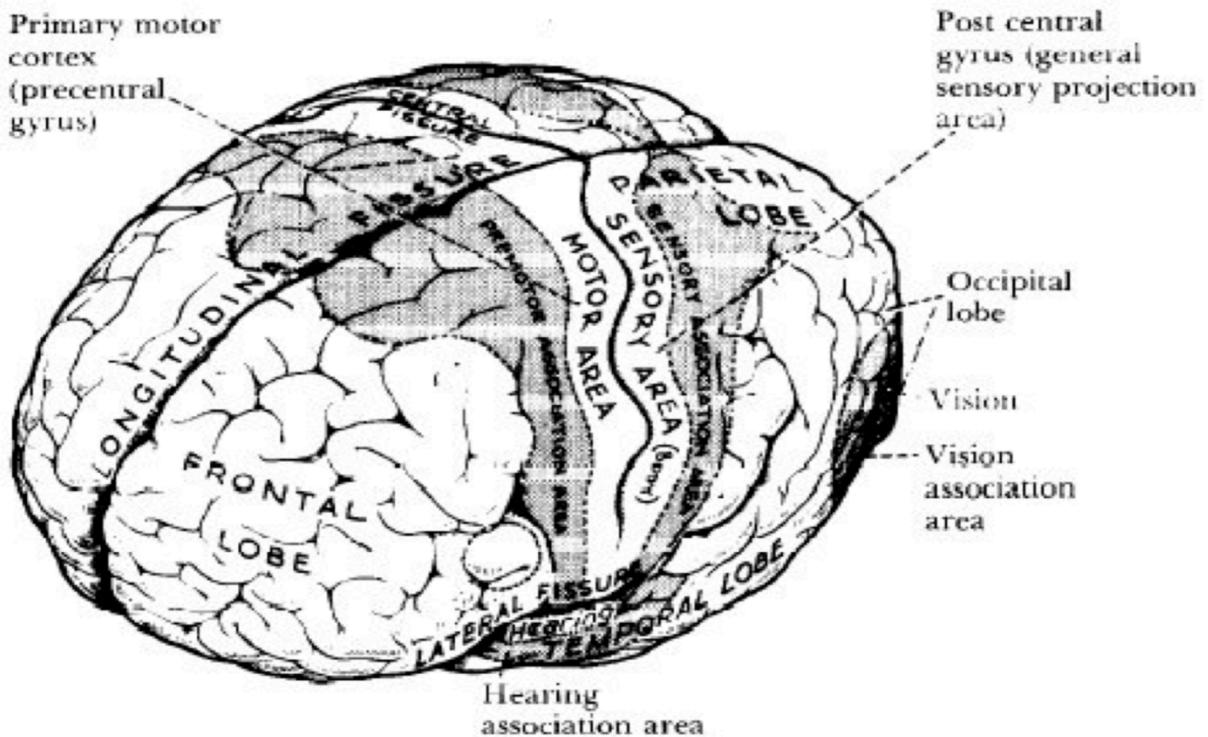


Figure 8. The Cerebral hemispheres showing towards the front the motor areas.

## 2 EEG Signal Preprocessing

### 2.1 Introduction

A critical problem in analyzing EEG is separating the brain waves from the Interference waves, also called artifacts, which are electrical potential not resulted from brain activities added to the EEG while recording it. Two sources are the main cause of artifacts:

1. The equipment used in recording the EEG where noise can be presented from the amplifiers, the electrodes, and the leads.
- 2- Electrical activity from various muscles like neck, jaw and eye muscles can be easily confused with the other brain waves. This is due to the fact that the muscles produce relatively large signals and are much closer to the surface than the neuron that carries the signals of interest, which are deep in the human brain. Moreover, the muscles produce relatively large signals compared to the brain neuron signals, which are attenuated while passing through the skin and the skull.

It was shown in previous works that the artifact the effect the EEG the most is the one caused by the eyeball and eye blinks movements. Movement in the eyelids and eyeball changes the existing potential field between the eyes retina and cornea, which is about 100mV [2.4]. The effect of this change is mainly in the signals drawn from the electrodes placed on the most frontal lobe. That is, the electrode located on Fp1 and Fp2. However, F3, F4, F7 and F8 are also affected but with less severity. Many low and high frequencies are included in the EEG because of that potential change -depending on the duration of the movement and amplitude of the potential change.

By observing the corrupted EEG, some criteria for the recognizing this type of artifact can be noted [2.5]. Those criteria can aid the search for a suitable filtering technique. The following are some corrupted EEG criteria

- Fp1 and Fp2 channels present high amplitude of delta wave.
- Similarity between the signals drawn from Fp1 and Fp2.
- Posteriorly, the amplitude of the delta wave declines rapidly (Much higher amplitude of the delta wave drawn from Fp1 and Fp2 than other channels).

There are two methods for filtering eye blink artifacts. The first one is the rejection methods, and the other is the subtraction methods [2.6]

- Rejection methods works by filtering the corrupted EEG, by detecting the corruption either automatically or by visual detection. The success of this method mainly depends on how well is the detection.
- Subtraction methods where it's assumed that the EEG produced by the measurement is a linear combination of the signals produced by the movement of the eyelid or the eye (Fig. 9), which is called electrooculogram (EOG), and an original EEG. Therefore, the original EEG can be obtained by separately recording the EOG and subtracting it from the measured EEG.

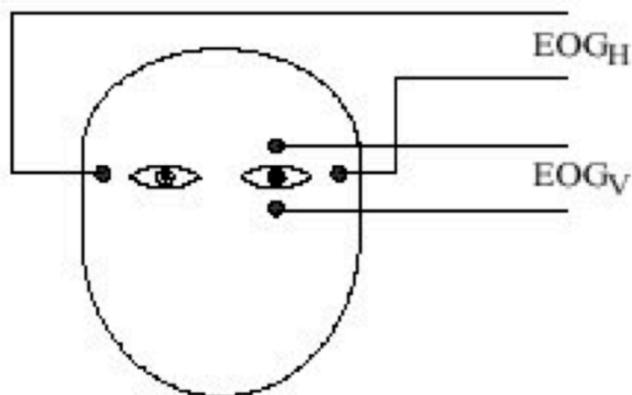


Figure 9. Movement of the eyelid or the eye (EOG).

These artifacts can cause errors in the classification of the EEG therefore reducing its usability. Nevertheless, by analyzing the EEG signals via various means, the artifacts can be removed to produce more useful signals. The recent methods to separate the brain activity from the blink and other muscle artifacts are biased on the concept of BSS or blind source separation. Four methods are explained in the following:

## 2.2 EEG artifacts removal using Independent component analysis

Independent component analysis abbreviated as ICA is a fairly recent technique for blind source separation. It's biased on the assumption that there exists a number  $n$  of signals made of a linear combination of  $m$  signals that are from independent source and are unknown. The instantaneous observation made on an a data vector with  $n$  dimensions  $x(i) = [x_1(i) \dots x_n(i)]^T$  is expressed as follows [2.7]:

$$x_k(i) = \sum_{j=1}^m a_{kj} s_i(i); k = 1 \dots n$$

$$\mathbf{X}(i) = \mathbf{AS}(i)$$

Where the mixing matrix  $\mathbf{A} = [a_{kj}]$  and the signals  $s(i) = [s_1(i) \dots s_m(i)]$  are unknown.

Other conditions for the existence of a solution are the following:

1-  $n = m$  (there are at least as many mixtures as the number of independent sources)

2- One source at most may be Gaussian.

The ICA obtains, under these assumptions, a solution of the following form:

$$\hat{\mathbf{S}}(i) = \mathbf{B}\mathbf{x}(i)$$

$\mathbf{B}$  is a factor known as the separating matrix.

This technique is advantageous over others in the following ways:

1. ICA separates the artifacts corrupted EEG signals into components that are independent by analyzing the nature of data, and doesn't need any artifacts free reference channels.
2. Artifact removal using this technique can conserve all the trials that have been recorded, which makes this technique preferable over other techniques based on rejection where there are only a limited amount of data in availability, or when muscle movements and blinks happens in a frequently manner.
3. This method is advantageous over regression methods in that it is characterized by data preservation.

However, ICA has certain limitations, which come in the following forms:

1. When the amount of data in the training set is small, the assumption that the constituents used by ICA are temporally independent cannot be satisfied.
2. ICA works under the assumption that the artifactual and neural activity sources contributing to EEG signals don't change over time, which is, in general, not true.
3. This method is tiresome since it entails the use of many complex computations as compared to rejection approach and also due to the fact that EEG brain computer interface is inherently real-time in nature.

### 2.3 Artifact removal using peak elimination

It was shown in previous work [2.5] that the introduction of artifacts in the EEG signals leads to a substantial energy rise in  $F_{p1}$  and  $F_{p2}$  which are forehead locations. Using small overlapping windows, this method analyses these two channels to check if an established threshold is surpassed by the energy of the signals. If this occurs, then the samples are considered artifact and they get rejected.

Even though this method is simple, the outcome is satisfying enough for it to be considered it as a first option for filtering artifact. Its implementation is considered easy enough to be carried on signal-processing platform that are fairly low in complexity. Nevertheless, a downside of this method is the fact that non-corrupted data can be mistakenly recognized as artifact and get rejected.

## 2.4 Recognition of blinking artifact by means of neural network

According to Bogacz and colleagues proposed method [2.12], the artifacts in the EEG signals can be found by using a method those functions on the basis of the neural system. In this method, the neural network's input was not a raw sampled EEG signal, but it was the coefficients that were calculated for an interval of one second of the EEG signal to express some distinguishing features of the blinking artifacts. In terms of sensitivity and correlation, 14 coefficients were chosen out of 41 that were designed based on the knowledge of the authors about the artifact recognition. A training set containing a large amount of data taken from over 27000 windows describing various kinds of pathological and proper waves, blinking artifacts, and artifacts aroused by other sources like facial and jaw muscles. Subsequently, three algorithms were tried and compared: RBF networks, back propagation networks, and, k-neighbors and. A classification error of 1.40%, which was the lowest among the other classification algorithms, was obtained by using back propagation [2.12]. Two factors were the reason behind the high accuracy obtained by this method:

- 1- A training set containing a large amount of data containing various kinds of EEG signals.
- 2- The coefficients that are used as the input to the neural network express the artifacts characteristic features, since they encode great amount of domain expert's knowledge.

## 2.5 Artifact rejection using bandpass Finite impulse response filters

The method uses a bandpass finite impulse response filter (FIR), followed by a particular eye blink threshold, in the purpose of filtering out eyeball and eye blink muscle artifacts. This method consists of the following [2.13] (Fig.10):

1. In order to remove slow baseline drift, the EEG samples obtained get passed through a digital bandpass filter.
2. During the training session, the blink threshold ( $V_t$ ) is determined.
3.  $V_t$  and the absolute value of the sample are compared.
4. In case the value exceeds  $V_t$ , then  $N$  samples are removed from the vicinity of zero crossing ( $N/2$  from both sides of threshold crossing).
5. The final step is filling the gap caused by the removal of the artifacts by shifting the following  $N$  samples.

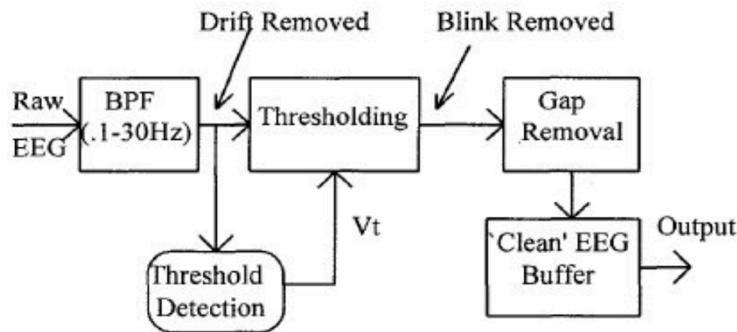


Figure 10. Proposed system scheme.

This method is advantageous in that it can work even with the present of baseline drift artifacts. Moreover, it's simple enough for it to be implemented on a digital signal processor that is low in processing power therefore lowering the cost. Nevertheless, this method can fail when the rate of which the blink occurs is high which is unnatural.

### 3 EEG feature extraction and Classification

#### 3.1 Feature extraction

Different approaches have been made with the intention of designing BCI like amplitudes of EEG signals [2.14], Adaptive AutoRegressive parameters and AutoRegressive [2.17] Band Powers [2.15] Power Spectral Density values [2.16], inverse model-based features [2.19] and Time-frequency features [2.18]. The following features properties are among the critical ones that must be considerate when designing a BCI:

- Excessive dimensionality: Feature vectors are often have high number of dimensions in BCI systems, for example [2.20]. Definitely, some features are usually obtained from numerous time segments or from several channels before being concatenated into a separate feature vector
- Noise and outliers: as a rule BCI features contain outliers or are noisy due to the fact that EEG signals have a reduced signal-to-noise ratio;
- The size of the training sets: The training sets are comparatively small as the process of training usually takes a lot of time and it's demanding for the subjects.
- Non-stationarity: Features of the BCI are non-stationary because EEG signals may swiftly change over time and more specially over sessions.

## 3.2 Classification methods

A number of classifying methods are commonly used to classify the acquired features:

### 3.2.1 Linear classifiers

Linear classifiers are algorithms that employ linear functions to perform the classification. They are among the most commonly used algorithms for BCI designing. Support Vector Machine and Linear Discriminant Analysis are the two main kinds of linear classifier.

### 3.2.2 Linear Discriminant Analysis

In order to separate the classes Linear Discriminant Analysis uses hyperplanes exploiting the features that represent the each classes where the feature vector regulates on which side of the hyperplane the class must be [2.21] (Fig 11).

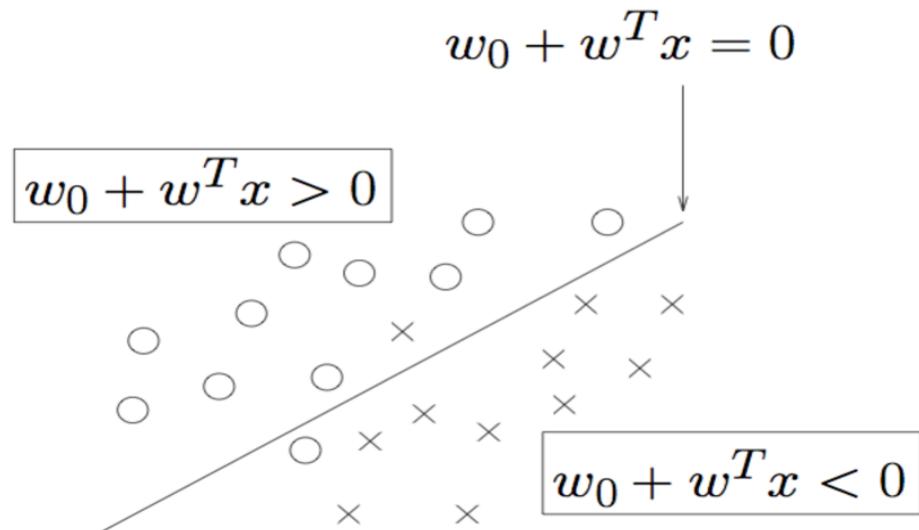


Figure 11. Linear Discriminant separating hyperplane.

LDA accepts normal distribution of the data, with equal covariance matrix for both classes. The hyperplane that separate the classes is gained by getting the projection that minimize the interclasse variance and boost the distance between the means of the two classes [2.40]. To resolve a N-class problem ( $N > 2$ ) numerous hyperplanes can be used. “One Versus the Rest” (OVR) strategy is the strategy commonly used for multiclass BCI, the strategy comprises of splitting each class from the others.

This technique is appropriate for online BCI system as it has a extremely low computational requirement. Furthermore, this classifier normally provides good results and is quite simple to use. Subsequently, LDA has been successfully used in numerous BCI systems, among them are multiclass [2.25], asynchronous [2.39], P300 speller [2.40], motor imagery based BCI [2.41].

The foremost disadvantage of LDA is its linearity that offers modest results when applied to complex nonlinear EEG data [2.27].

A Regularized Fisher’s LDA (RFLDA) can also be used for designing BIC systems. This classifier announces a regularization parameter  $C$  that is able to penalize or allow the classification mistakes on the training set. The obtained classifier can maintain a better accommodate and generalization capabilities outliers. As outliers in EEG data are quite common, this regularized version of LDA results in better classification for BCI than the non-regularized version [2.42].

### 3.2.3 Support Vector Machine

The SVM use the discriminant hyperplane (Figure 12) in identification of classes [2.29]. However, in SVM, the hyperplane that maximizes the margins is used; that is the nearest distance from the training point as shown in the figure below. The maximized margins increase the generalization capabilities [2.29]. SVM as RFLDA uses the regularization parameter C that ensures the outliers are accommodated and hence allows errors on the set for training

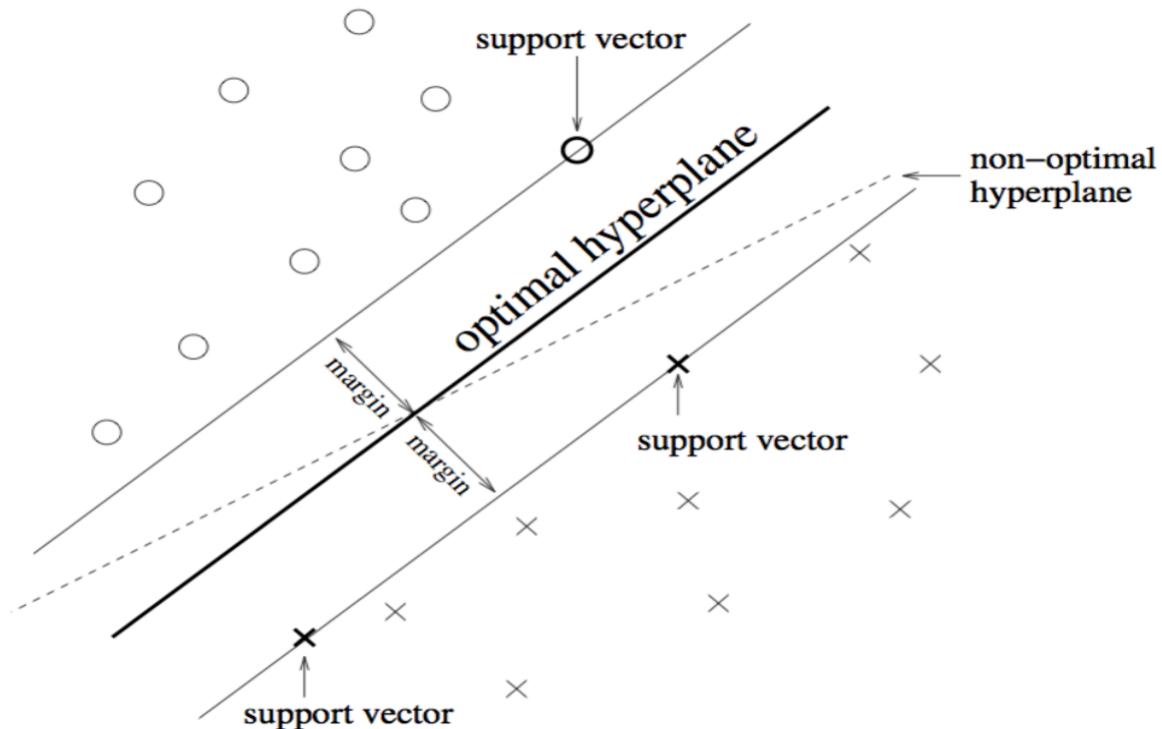


Figure 12. SVM hyperplane.

This SVM makes possible the classification by the use of the linear decision boundaries. It is thus called linear SVM. It has been applied successfully in relatively large number of synchronous BCI problems [2.28]. It is also possible to create a decision boundary that is not linear by increasing the classifier complexity using the “kernel trick”. In implicitly mapping, this involves the use of data to a different space whose characteristics is generally have a higher number of dimensions, using the kernel function  $K(x, y)$ . The kernel in the research of BCI is mostly the Radial or Gaussian Basis Function (RBF) kernel:

$$K(x, y) = e^{\left(\frac{-||x-y||^2}{2\sigma^2}\right)}$$

In this case, the corresponding SVM usually referred to as RBF SVM or Gaussian SVM [2.29] RBF SVM gives great results for the BCI applications [2.25]. Similar to LDA, SVM is used with multiclass BCI problems by the use of OVR strategy [2.29].

The application the SVM is advantageous in several ways. Due its regularization term and the margin maximization, it has an excellent generalization properties [2.30], being insensitive to overtraining [2.30], and a solution to the curse-of-dimensionality [2.29]. The SVM have some hyper-parameters that are defined by hand, i.e. the regularization parameter C, as well as the RBF width  $\sigma$  if employing kernel 2. However, speed is the expense of these advantages.

### 3.2.4 Artificial Neural Networks

The inspiration for the Artificial neural networks came from examination of central nervous. They have been used to build a wide variety of systems to solve problems that are hard to solve using ordinary rule-based programming, speech recognition and including computer vision and they are the most commonly used classifiers in the field of BCI e.g. [2.31].

Artificial Neural Networks are It's an assembly of artificial neurons connected together in a specific architectures to produce nonlinear decision making functions [2.32].

Neural network architectures used in BCI:

#### 3.2.4.1 Multilayer Perceptron

Multilayer Perceptron is the most commonly used neural network in the BCI research. It's a feed-forward network made of a number of layers. Each layer has units or "neurons" which receive their input from the weighted sum of the output of the neurons in the previous layer and send their output to the neurons in the next layer. There are no connections between the neurons that lay in the same layer. The first layer is the input layer, followed by one or more hidden layers, and at the end, there's the output layer [2.32] (Fig 13).

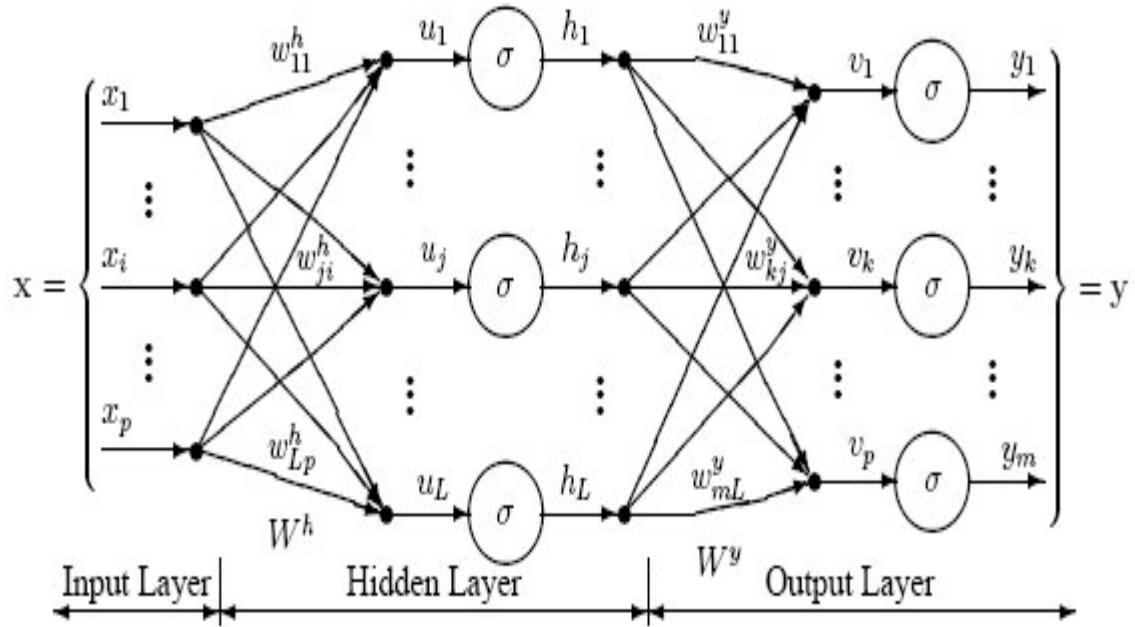


Figure 13. Illustration of a perceptron neural network with three layers.

The number of hidden neurons in the hidden layer significantly affects the accuracy of the classification and the speed of the training (training is explained below in section 3.2.4.1.1). If few neurons were used in the hidden layer there won't be enough neurons to adequately detect the signals. On the other hand, increasing the number of hidden neurons means increasing the training time. Also, using neurons more than actually needed, will cause in overfitting. That is, the training data wont be enough to train the excess neurons in the hidden layer, not to mention the significant decrease in the training speed. Therefore compromise should be made between classification accuracy and speed of the neural network when it comes to choosing the number of the neurons in the hidden neurons. The number of hidden neurons should be between the number of inputs and the number of outputs [2.30].

Neural Networks and thus Multilayer perceptron are, when composed of a sufficient number of layers and neurons, can approximate any continuous function. This gives Neural Networks the flexibility to adapt to a wide variety of problems. Therefore, Multilayer perceptron have been used in most BCI problems such as multiclass [2.31] or binary [2.33] [46], synchronous [2.34] [20] or asynchronous [2.35] [12] BCI. Nevertheless, Multilayer perceptron Neural networks being universal approximators gives these kind of classifiers sensitivity to overtraining, especially signals like EEGs which are non-stationary and noisy by nature, e.g., [2.36] [47]. Therefore, the careful selection of neural network architecture and its regularization should be chosen carefully [2.30][2.44].

### 3.2.4.1.1 Training the neural network

To approximate continuous functions, the weight of each neuron should be set up to an appropriate value, which is the result of a procedure known as training, which adjusts the network's weights, by means defined by the training algorithm. A number of training algorithms have been developed but the most commonly used one is the backpropagation.

#### 3.2.4.1.1.1 Backpropagation

The learning starts by initializing the connections weights to random values. Afterward, two sub processes take place:

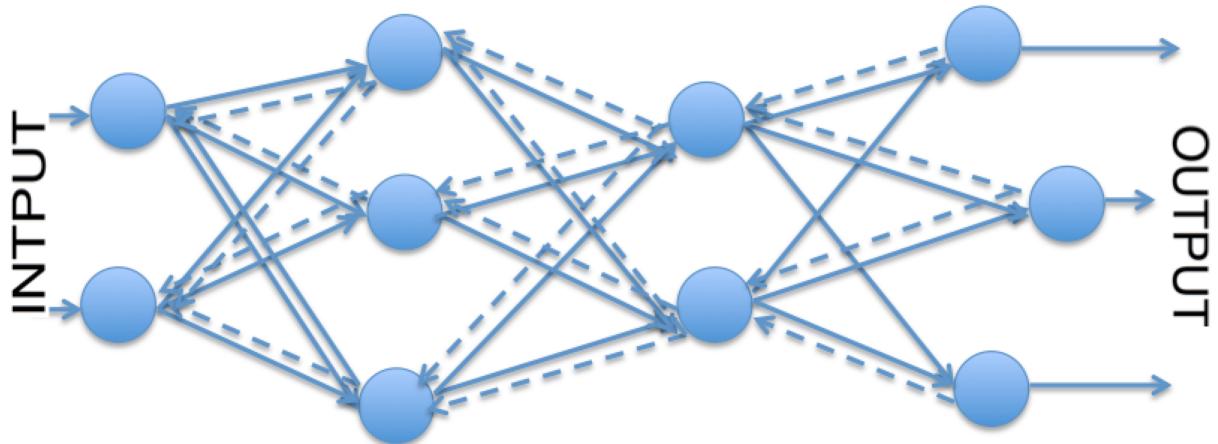


Figure 14. Illustration of backpropagation algorithm where solid lines represent forward propagation and dashed lines represents back propagation.

- 1- Forward propagation (Illustrated by the solid lines in Fig 14): The input data propagates from the input neurons to the output neurons passing through the hidden layer. During this transaction, the weights of the connections are held constant, the status of each layer will affect the next layer, only. Then the error signal used in the second is calculated by comparing the output of the network with the expected output.
- 2- Back propagation (Illustrated by the dashed lines in the above diagram): The error signal obtained from the forward propagation transfers from the output layer to the input layer passing through all the hidden layers in between. During this propagation, the values of connection weights are modified using the error feedback with the aim of bringing the value of the real output closer to the expected output value.

A mathematical detailed description of the backpropagation can be found in appendix A.

### 3.2.4.2 Other Neural Network architectures

In the field of BCI different types of Neural Network architecture are used. Amongst them, the Gaussian classifier is one that deserves a detailed attention as it has been specially created to perform the classification in BCI [2.38]. Each unit of this Neural Network architecture appears to be a Gaussian discriminant function that represents a different class prototype. As its authors state, this Neural Network outdoes Multilayer perceptron architecture when used in the field of BCI, therefore can accomplish rejection of ambiguous samples efficiently [2.38]. As a result, this classifier can be applied with success to mental task classification and motor imagery [2.38], mainly throughout asynchronous experimentations [2.38].

If we do not take Gaussian classifier into account, some other types of NN have been tried in a more peripheral way for BCI purposes:

- RBF Neural Network [2.42] [5];
- Fuzzy ARTMAP Neural Network [2.40] [55].
- Bayesian Logistic Regression Neural Network (BLRNN) [2.43] [8];
- Adaptive Logic Network (ALN) [2.44] [59];
- Dynamic Neural Networks such as Gamma dynamic Neural Network (GDNN) [2.41] [57] or the Finite Impulse Response Neural Network (FIRNN) [20].
- Learning Vector Quantization (LVQ) Neural Network [2.39][53].
- Probability estimating Guarded Neural Classifier (PeGNC) [2.36].

### 3.2.5 Nonlinear Bayesian classifiers

Application of all the non-linear Bayesian classifiers leads to a non-linear decision boundaries. In addition, they are generative enabling them to perform more efficient in the rejection of uncertain samples than the discriminative classifiers. Nevertheless, the non-linear Bayesian classifiers are less common than Neural Networks or linear classifiers in BCI applications.

#### 3.2.5.1 Bayes quadratic

The aim of the Bayesian classification is to assign to a class that has the highest probability to each feature vector its with the highest probability [2.34]. In this case, the rule of Bayes is used in computing the posteriori probability belonging to a given class that a feature vector has [2.34]. Using these probabilities and, the rule of Maximum A Posteriori (MAP), the vector feature of this class can be estimated.

Bayes quadratic assumes a different normal distribution of data. This result is that the quadratic decision boundaries so produced explains the name of the classifier. Although this classifier isn't commonly used in the applications of BCI, it has been applied widely successfully to motor imagery [2.33], and in the classification mental tasks [2.26].

### 3.2.6 Nearest Neighbor classifiers

These classifiers are relatively simple than the rest presented above. They mainly consist of classifying a feature vector depending on its nearest neighbor(s). The neighbor could be a class prototype as in the case Mahalanobis distance or a feature vector of the training set as in of k Nearest Neighbors (kNN). They are generally nonlinear discriminative classifiers.

#### 3.2.6.1 k Nearest Neighbors

It assigns the class that is dominant among its k nearest neighbors within the training set to an unseen point [2.34]. In BCI, metric distance is used to obtain the nearest neighbors, e.g., [2.28]. kNN can approximate with a sufficiently high value of k, and enough number of samples for training, any function which gives it the ability to create nonlinear decision boundaries.

KNN algorithms are not common in the BCI research because of there sensitivity to the curse-of-dimensionality [2.20], which make them be unsuccessful in several BCI experiments [2.28] . But kNN may show efficiency when used in BCI that has feature vectors with low number of dimensions [2.22].

# Chapter 3

## Methodology and Results

### 1 Summary

A summary of the methodology is provided in the following diagram:

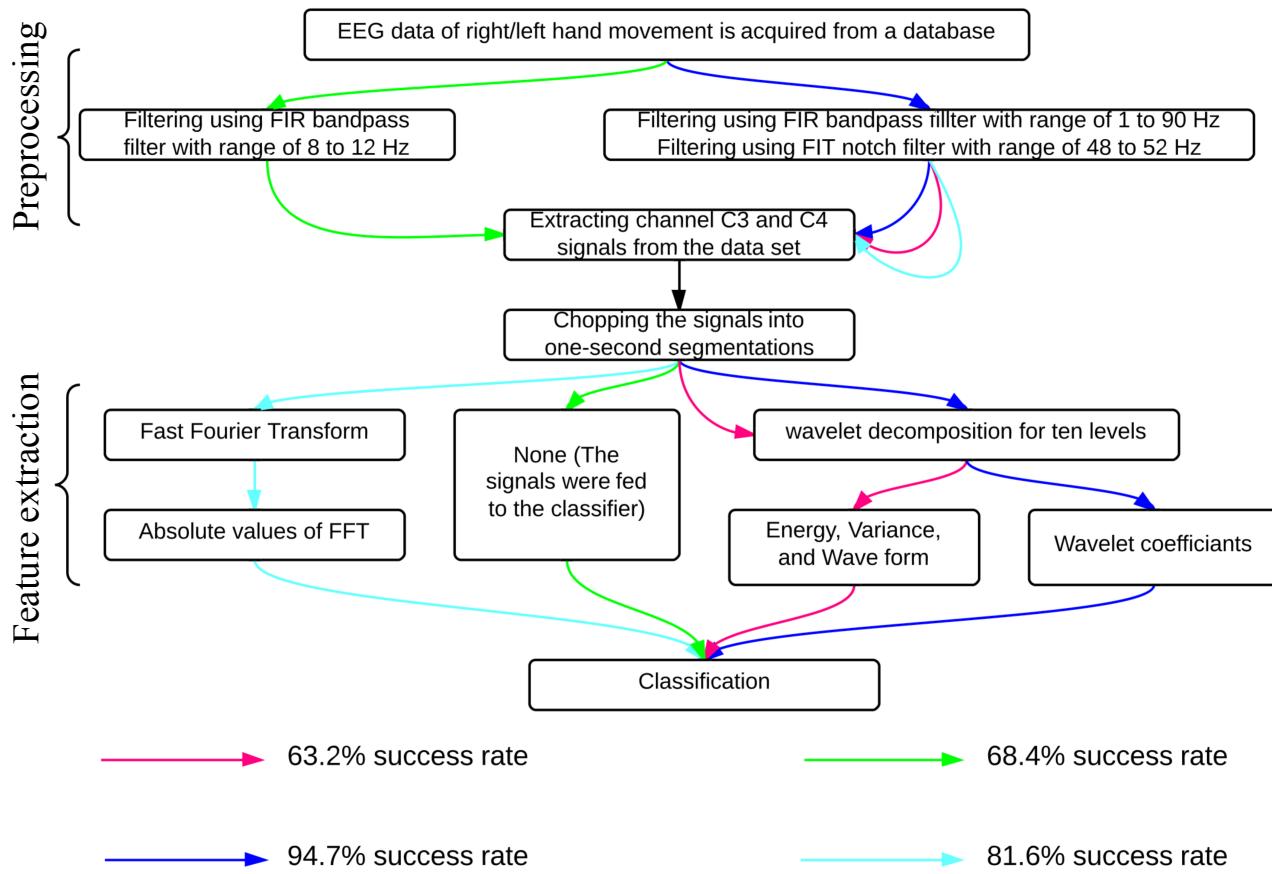


Figure 15. A summary of the methodologies.

## 2 Tools and data used

### 2.1 The EEG data set

The EEG data used in this paper are for a right handed 21 year old male with no known medical conditions. It consists of left and right hand movements recorded with the subject eyes being closed. Nineteen electrodes were distributed on the scalp sites to capture the EEG data. Each channels is represented with a different color in Fig. 16 and Fig. 17.

The Neurofax EEG System, which uses a daisy chain montage, was used to record the EEG signals at 500 Hz. The data was aquired from the following database provided by the National University of Sciences and Technology, Pakistan[3.8].

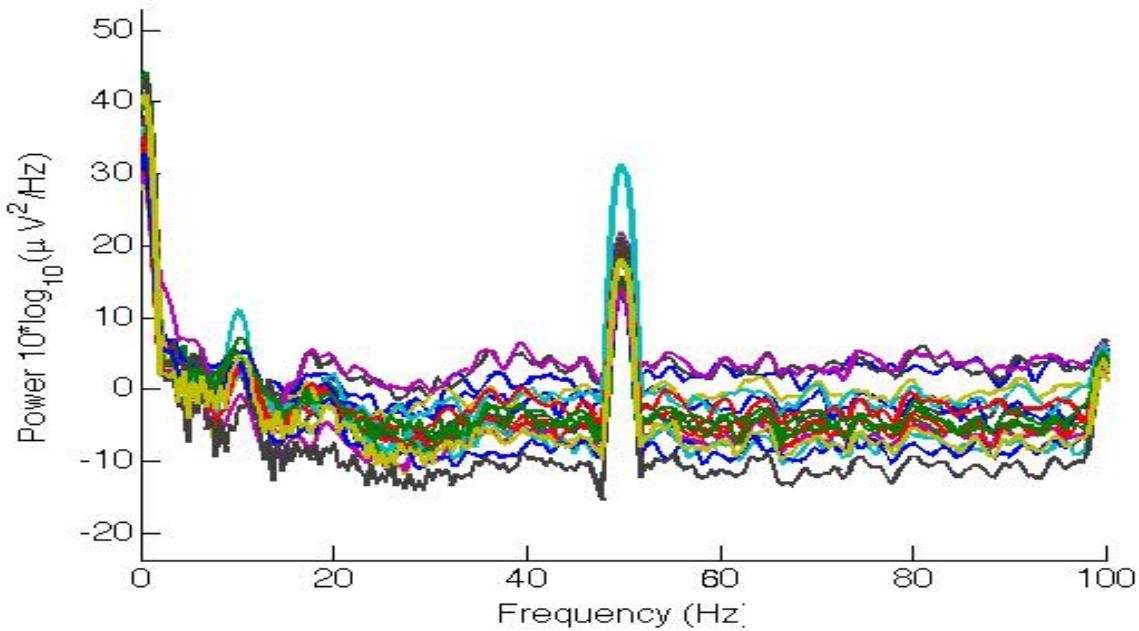


Figure 16. Frequency spectra of left hand movement.

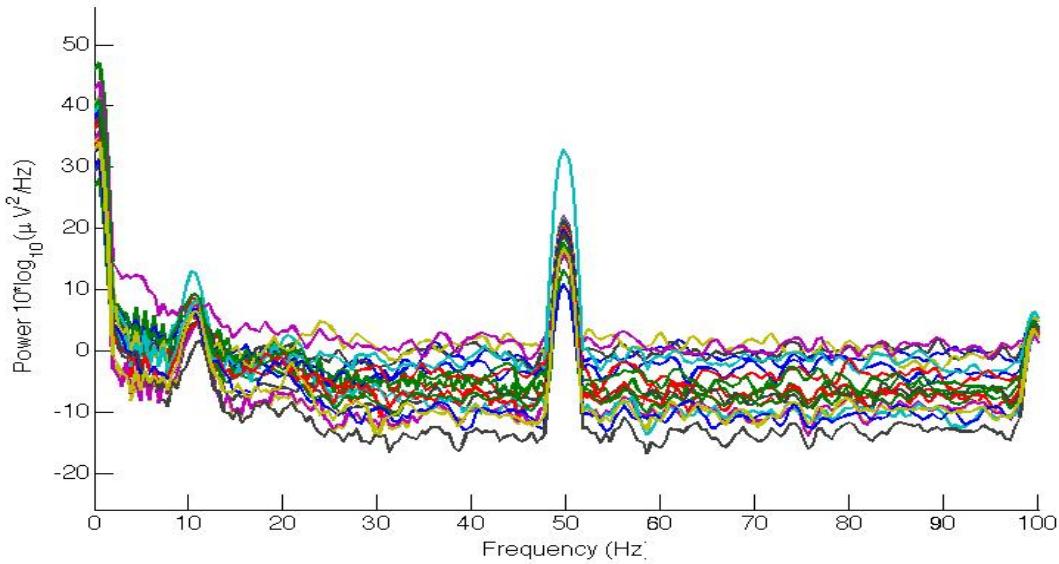


Figure 17. Frequency spectra of right hand movement.

## 2.2 Matlab

Matlab was used throughout this project as the programming environment of choice. MATLAB (matrix laboratory) is an environment developed by MathWorks for numerical computing. it allows, function plotting, manipulations of matrixes, algorithms implementation, building user interfaces, and interfacing with applications coded in other programming languages, such as Java Fortran, C, and C++. Moreover, the MuPAD symbolic engine that other toolboxes use enables symbolic computing, and an additional add-on named Simulink, adds Model-Based Design and graphical simulation for embedded and dynamic systems.

## 2.3 EEGLAB

To perform the EEG preprocessing, EEGLAB was used [3.1]. EEGLAB is a Matlab toolbox that preforms processing for event-related and continuous MEG, EEG and other electrophysiological data. It preforms artifact rejection, independent component and time/frequency analysis, event-related statistics, and several visualization modes of the data.

## 2.4 Neural Networks toolbox

The MATLAB neural networks toolbox was used for all NN experiments. Neural Network Toolbox™ enables us to model complex nonlinear systems. It supports supervised learning with, radial basis, dynamic, and feedforward networks, and unsupervised learning with competitive layers, and self-organizing maps. Designing, visualizing, training, and simulating neural networks are made easy with the usage of the toolbox. It can be used for various applications such as pattern recognition, clustering, time-series prediction, dynamic, and data fitting.

## 3 Procedure common in all classification methods attempted

### 3.1 Channel Selection

Biased on the findings in [3.2], the EEG data drown from many channels appeared to be redundant when it comes to information about motor movement. In [3.3] [3.4], it's shown that the data corresponding to the activity of the brain's neurons that is related to the execution of movements of right and left hand is almost exclusively drown from the C3 and C4 channels of the EEG channels. Hence, analyzing the data coming from channel C3 and C4 and discarding the information in the other channels should be sufficient to obtain satisfactory results.

### 3.2 Signal segmentation

In real word applications the EEG signals should be processed every second so that the system can maintain reasonable responsibility. A problem can arise from processing every second of EEG data separately which is that the amount of data in one second of EEG signals might not be sufficient enough to classify it. In this research, the EEG signals used were chopped into one-second segmentations containing 500 samples. However, to reduce the effect of mentioned problem, each second of the EEG signals was averaged with the previous second (moving average). This way, the data produced is more likely to be classified correctly, and in the same time, the period for which the EEG signals are processed for is still short enough for practical application.

### 3.3 Classification

The training in this project was handled with the aid of the back-propagation learning algorithm explained in the literature review. Considering the findings in section 3.2.4.1, the number of hidden layer set to 10 neurons and the output is a one node with the value of zero representing right hand movement and one representing left hand movement. 70% of the data set was used for the training, 15% for the validation, and 15% for the testing.

## 4 Classification methods attempted

### 4.1 First method

#### 4.1.1 Preprocessing

As mentioned in the literature review, the brain wave that is generated during motor activity is the mu brain wave, which has frequency band of 8Hz to 12Hz. Hence, taking that frequency band and ignoring the rest of the frequencies promises a better classification since the redundant information will be removed.

A FIR filter was used for a bandwidth of 8Hz to 12Hz in order to obtain the required frequencies band. To perform the bandpass filter, a lowpass filter of 8Hz (Fig. 18) was applied followed by a highpass filter of 90Hz (Fig. 19). It was recommended by the EEGLAB tutorial to perform the bandpass filter this way as it minimizes the ringing and filtering noise.

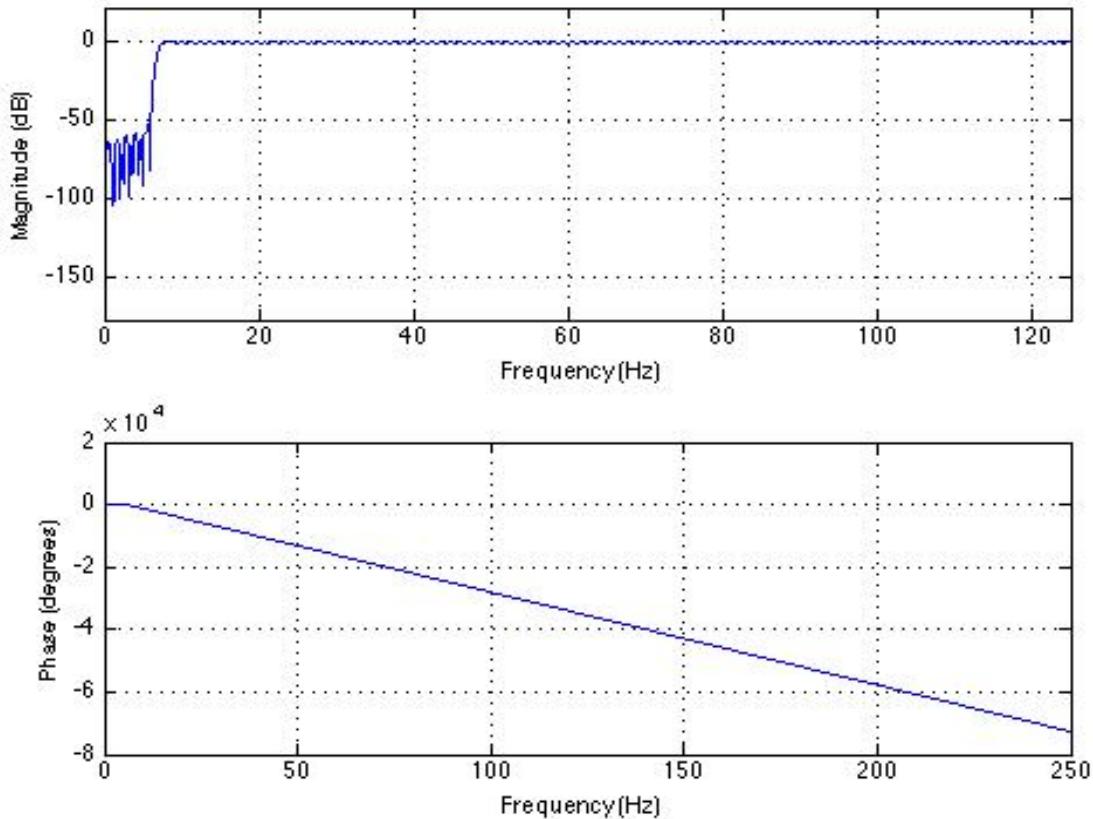


Figure 18. FIR lowpass filter at 8 Hz.

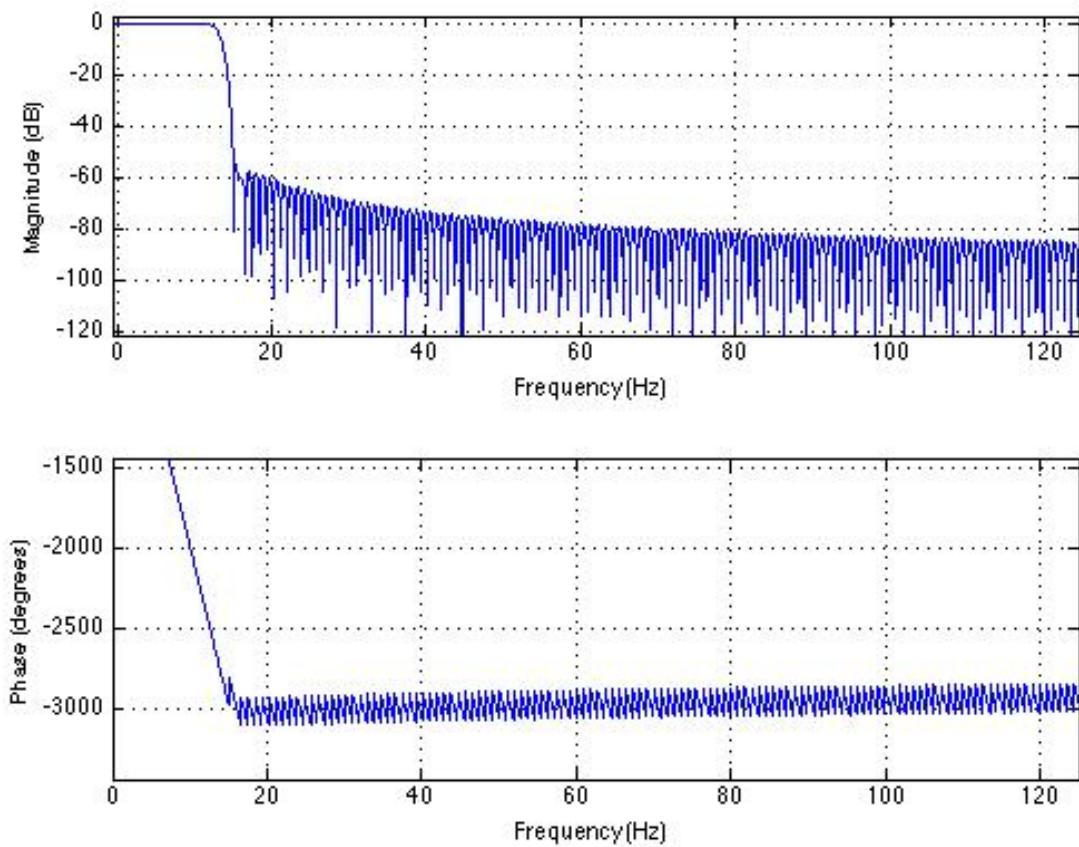


Figure 19. FIR high filter at 12 Hz.

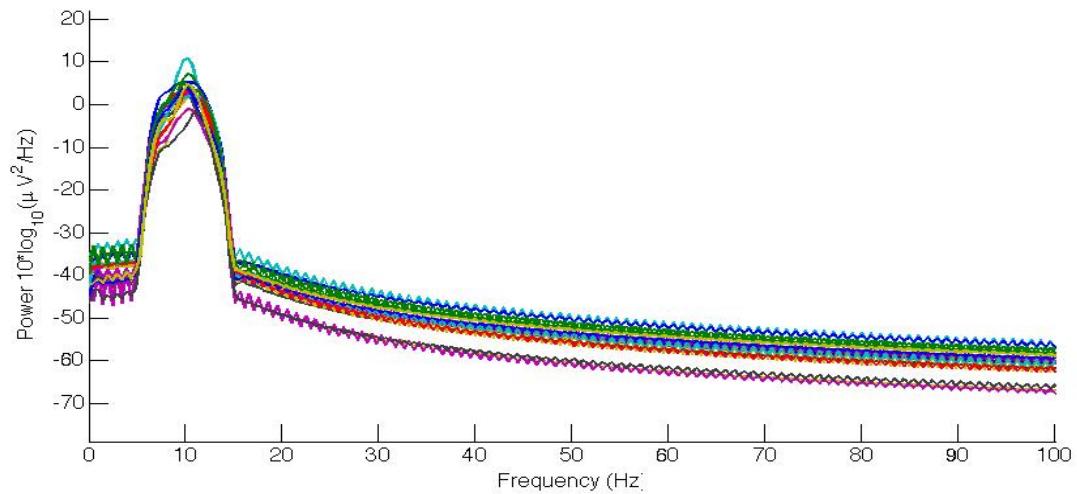


Figure 20. Frequency spectra of left hand movement after applying a FIR bandpass filter from 8Hz to 12Hz.

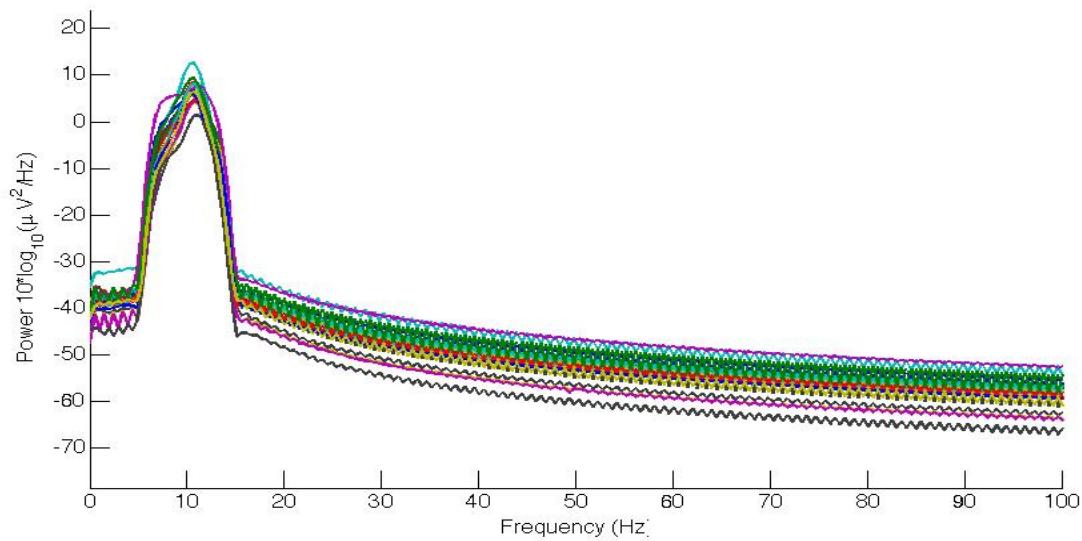


Figure 21. frequency spectra of right hand movement after applying a FIR bandpass filter from 8Hz to 12Hz.

#### 4.1.2 Feature extraction

In this method, no feature extraction was used. The processed signal from channel C3 and C4 was averaged and fed to the classifier.

### 4.1.3 Results

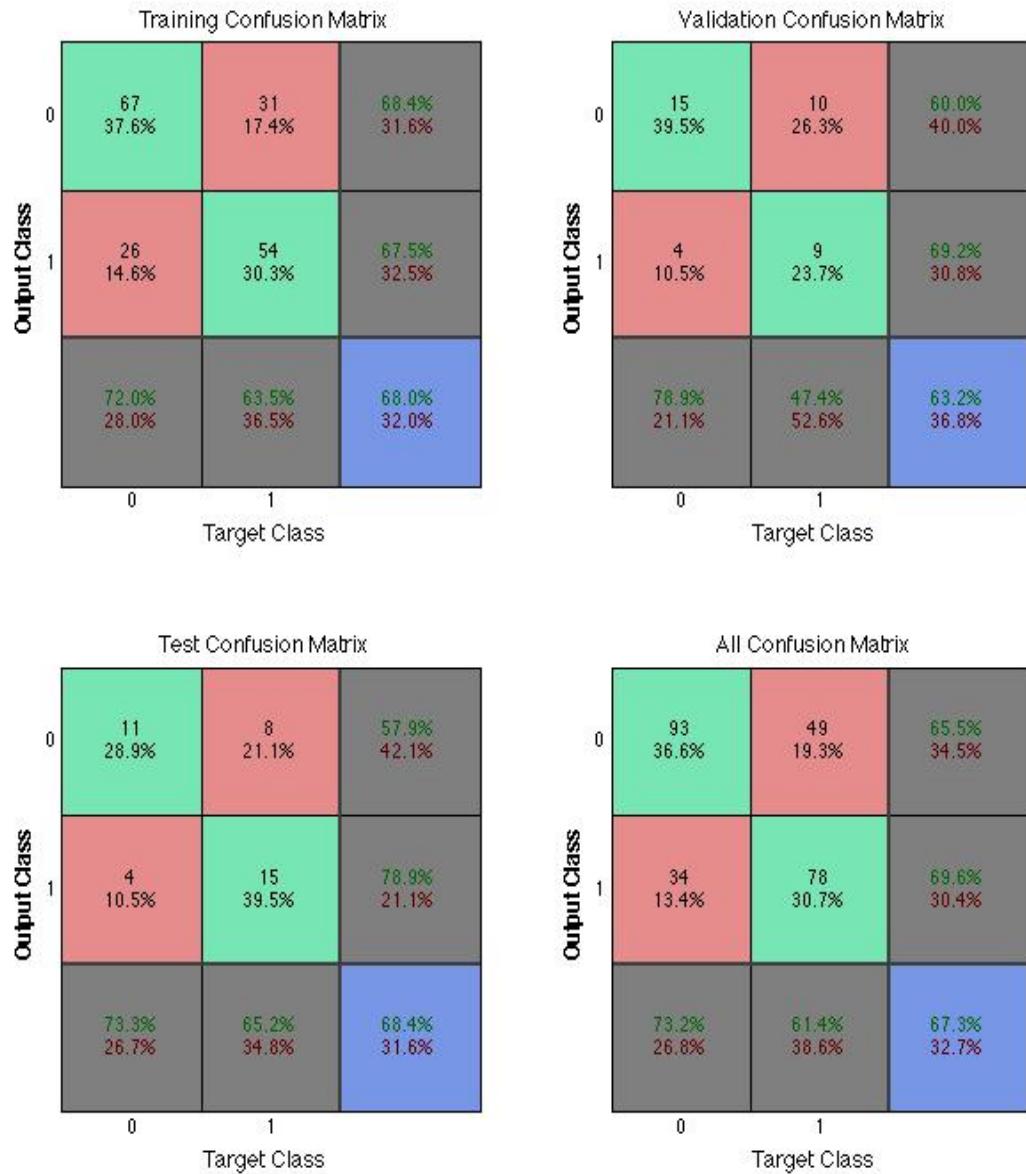


Figure 22. Confusion of first method.

## 4.2 Second method

### 4.2.1 Preprocessing

To remove the shifts of the direct current and to minimize the artifacts at boundaries caused by the filtering processes, a bandpass filter from 1 Hz to 90 Hz was applied. To preform the bandpass filter, a lowpass filter of 1Hz (Fig. 23) was applied followed by a highpass filter of 90Hz (Fig. 24).

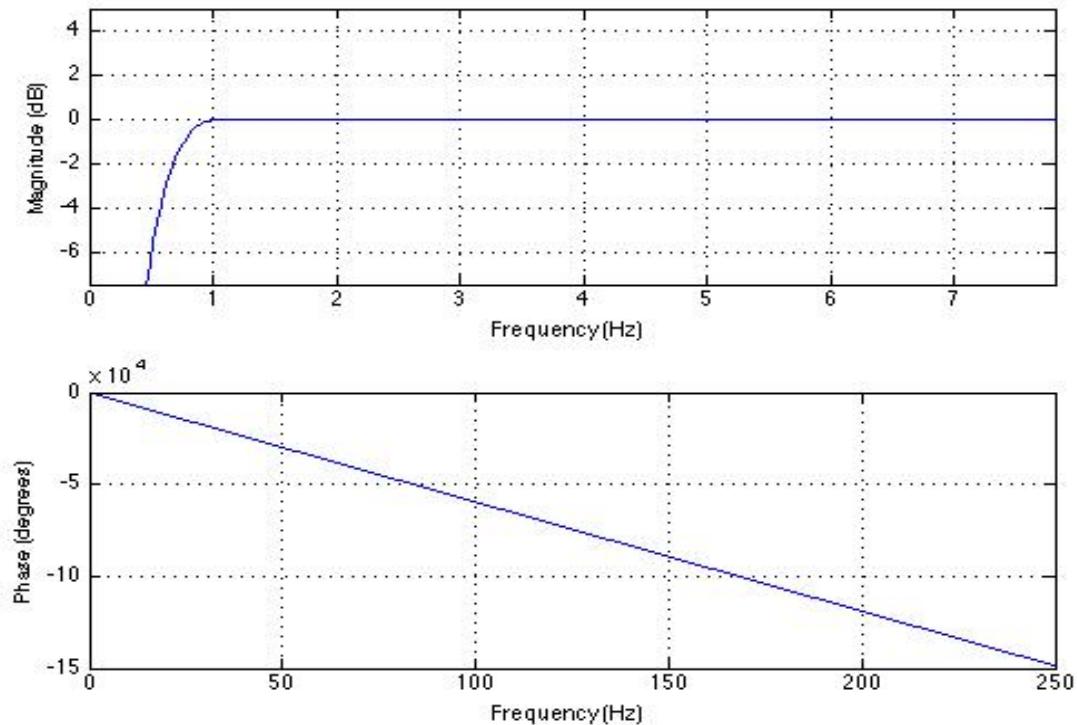


Figure 23. FIR lowpass filter at 1Hz.

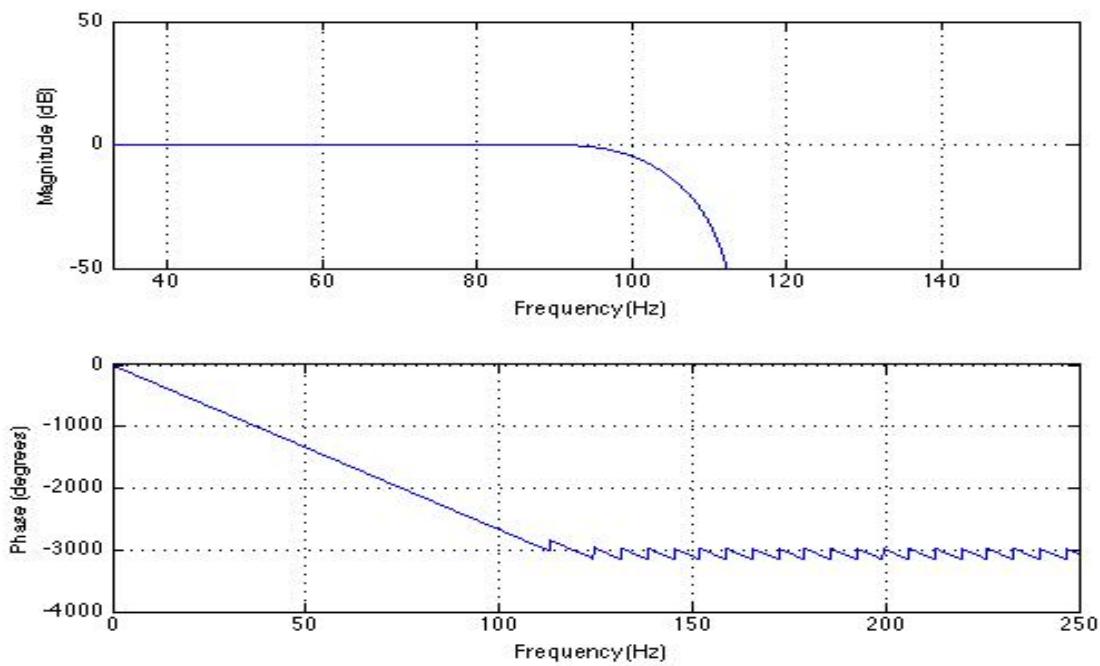


Figure 24. FIR high filter at 90 Hz.

The electrodes on the scalp not only capture the EEG signals, but also capture all electromagnetic pollution around the electrodes, which introduce a lot of noise. The source that contributes to most of the external noise is the power lines, which emit oscillations of electromagnetic signals of 50 Hertz [3.5], which can be seen clearly in Fig. 23 and Fig 24. In order to remove this noise, a FIR Notch filter was used to filter out the frequencies between 48 and 52 Hz (Fig. 25).

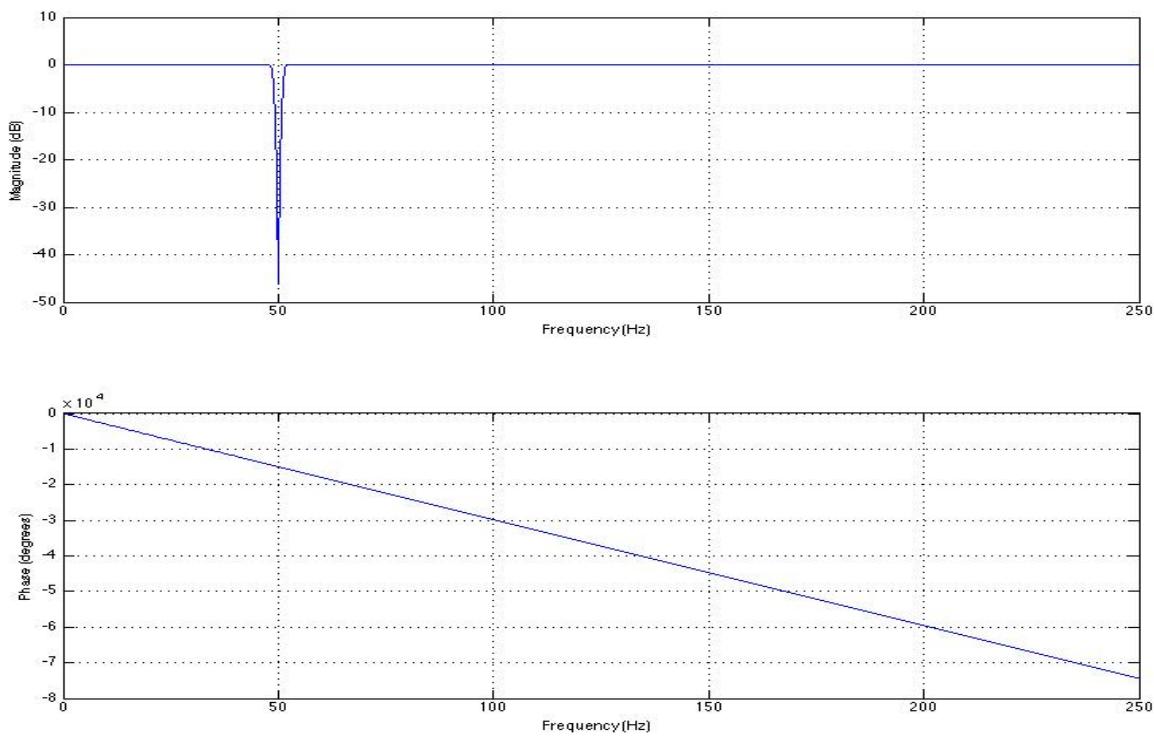


Figure 25. FIR Notch filter between 48 and 52 Hz.

The resultant EEG can be seen in figure 3.6 and figure 3.7 with the 50 Hz line noise removed and the shifts of the direct current the artifacts at boundaries clearly reduced.

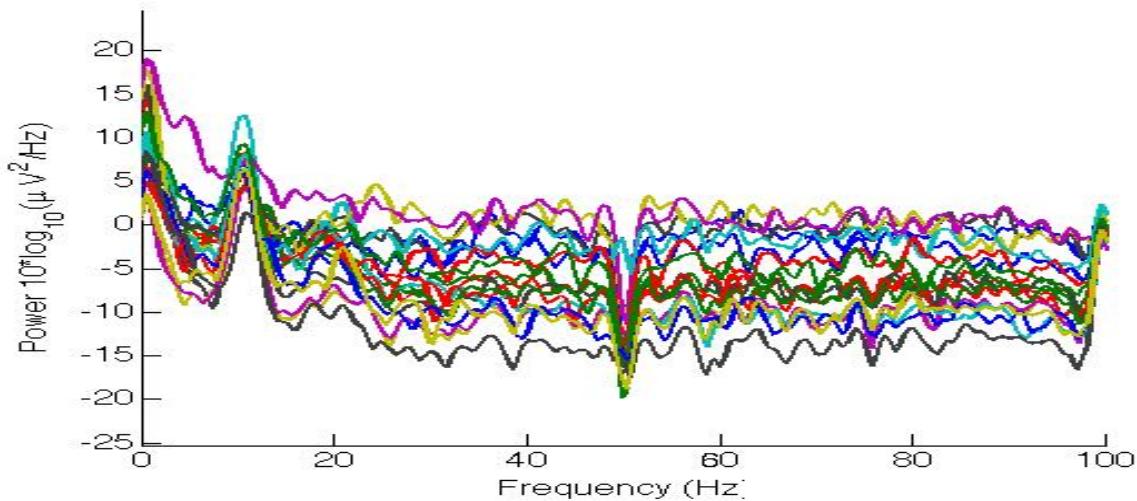


Figure 26. frequency spectra of left hand movement after applying a FIR bandpass filter from 1 Hz to 90 Hz and a FIR Notch filter between 48 and 52 Hz

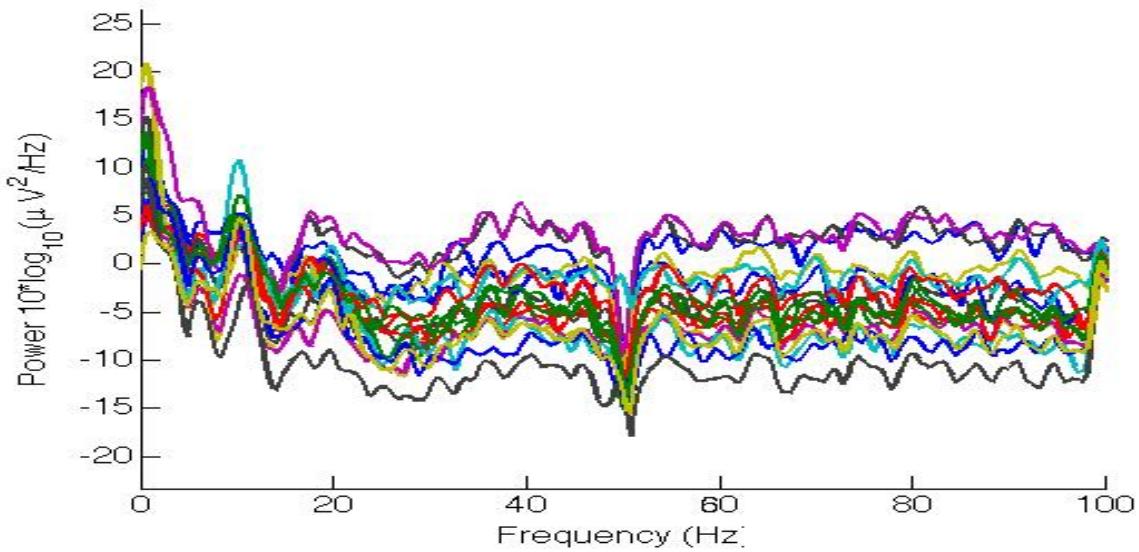


Figure 27. Frequency spectra of right hand movement after applying a FIR bandpass filter from 1Hz to 90Hz and a FIR Notch filter between 48 and 52 Hz.

Then, channel C3 and C4 where chosen to represent the motor movement as descript in section 2.3 and signal was segmented as descript in section 2.1

#### 4.2.2 Feature extraction

After separating the signals drown from channel C3 and C4 from the EEG set. The Fast Fourier transformation (see appendix A) was taken for each channel to construct two features vectors for the two channels. Afterword, the two features vectors obtained from processing channel C3 and C4 were averaged to produce the final feature vector that was fed to the neural network.

### 4.2.3 Results

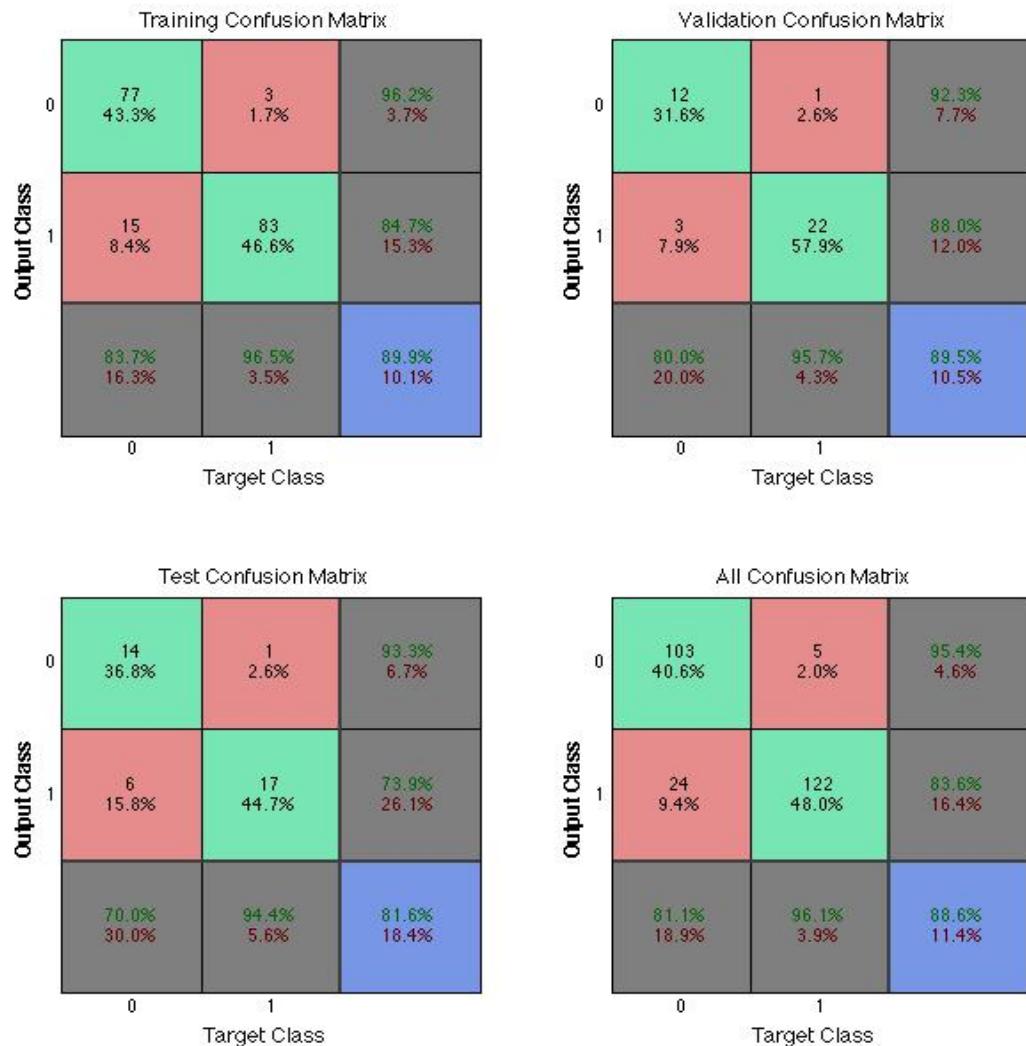


Figure 28. Confusion of the second method.

## 4.3 Third method

### 4.3.1 Preprocessing

The preprocessing used in this method is the same as the one used in the second method.

### 4.3.2 Feature extraction

The Multiscale Wavelet decomposition (see appendix C) was computed for ten levels using Daubechies wavelet and the features, including Energy, Variance, and Waveform Length were obtained from each level resulting in a two feature vectors having 33 feature each for both channels C3 and C4. Afterword, the two features vectors were averaged to produce the final features vectors that was fed to the neural network.

### 4.3.3 Results

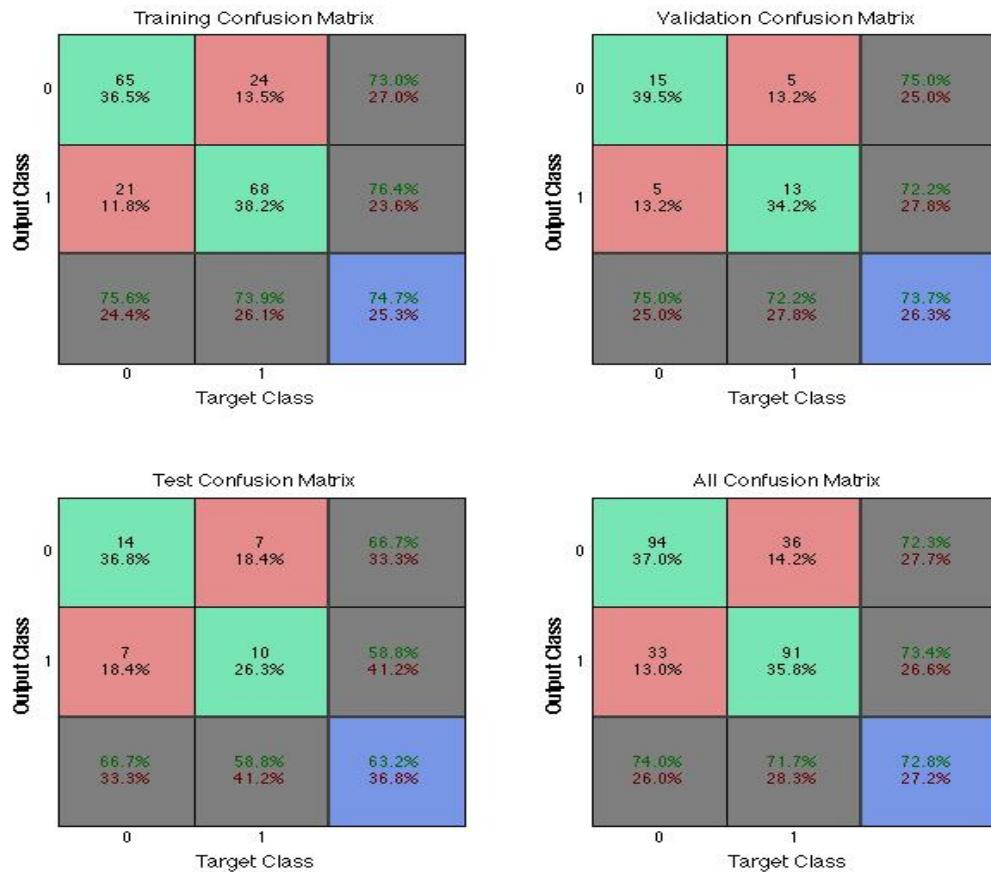


Figure 29. Confusion of the forth method.

## 4.4 Fort method

### 4.4.1 Preprocessing

The preprocessing used in this method is the same as the one used in the second method.

### 4.4.2 Feature extraction

The signals drawn from channel C3 and C4 were separated from the set to represent the motor movement as described in section 2.3. Then coefficients of wavelet decomposition for ten levels (see appendix C) were taken for each channel to construct two feature vectors for the two channels. Afterward, the two feature vectors obtained from processing channel C3 and C4 were averaged to produce the final feature vector that was fed to the neural network.

### 4.4.3 Results



Figure 30. Confusion of fifth method.

## Chapter 4 Discussions

### 4.1 First method

It was discussed in section 1.2 that motor activities including actual and imaginary movement is best defined in mu brainwave which is in the frequency range of 8 to 12 Hz. Therefore, taking out that frequency range and perform the classification over it should promote very good results since the other information that doesn't represent left/right hand movement will be removed and won't disturb the classification process. However, the result came out rather surprising where a success rate of only 68% was obtained. The bad results were later explained by looking at the frequency spectra of the processed signals where it can be clearly seen that a large amount of noise and ringing artifacts caused by the filtering process being introduced to the signals (Fig. 31), which greatly affected the accuracy of the classification.

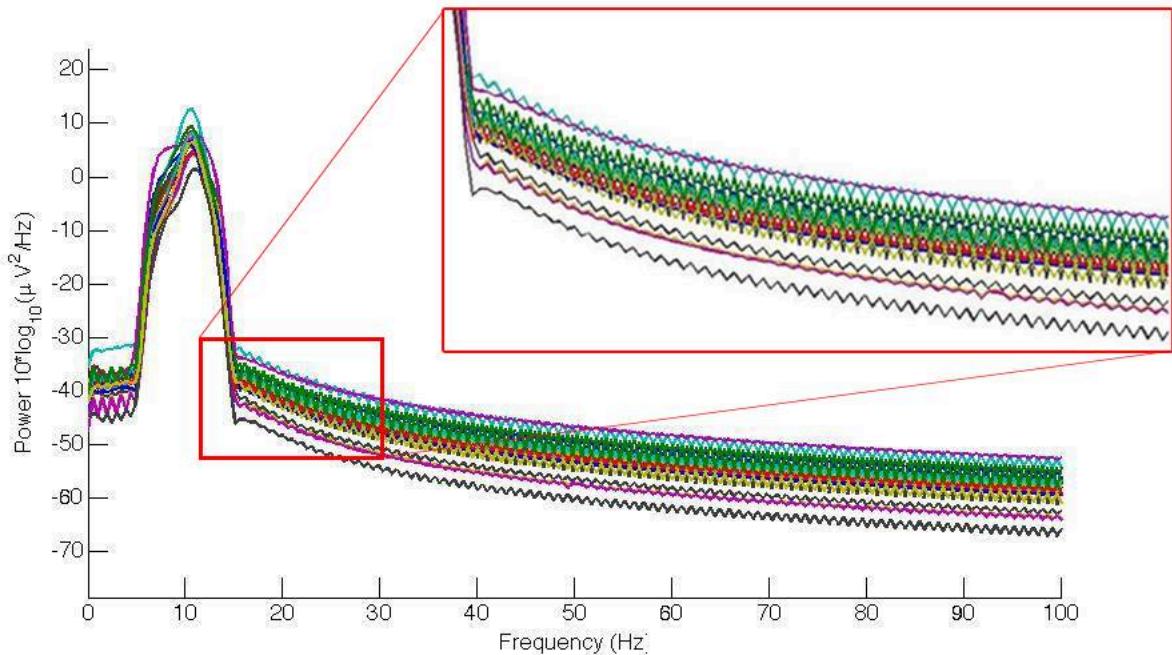


Figure 31. Noise and ringing artifacts caused by the filtering process.

## 4.2 Second method

This second method showed a success rate of 81.6% when it was tested. In this method, the preprocessing was done with the aid of FIR filters to remove the external power lines noise and filter out the shifts of the direct current and minimize the artifacts at boundaries as previously explained in section 4.1.1. It uses the absolute values Fast Fourier Transformation as the feature vector, which generates the frequency spectrum of successive segmentations of the EEG signal.

Biased on my anticipation, FFT should not give good results. The reasoning behind this is that it was shown in [3.6] that the frequency spectrum is observed to vary over time as the Fourier coefficients vary which indicates that the EEG signal is a non-stationary signal. And since FFT it cannot represent the change of the frequency in respect to time, FFT doesn't promises good result. However, the results obtained represent a relatively well performed classification, this is probable due to the frequency content not being changed much since the period the EEG signal was segmented for is quite short.

To verify this finding, an experiment was carried out to test the classification with a longer segmentation period of the EEG signal, where the EEG signal was segmented for ten seconds instead of one without performing the moving average processed mentioned in section 3.1. The experiment resulted in a classification error rate of 47.4%, which is, when compared with the 18.4% error rate of the classification using short period segmentation, clearly shows a significant increase in the classification error rate. Therefore, the proposed reason for the good results of the classification.

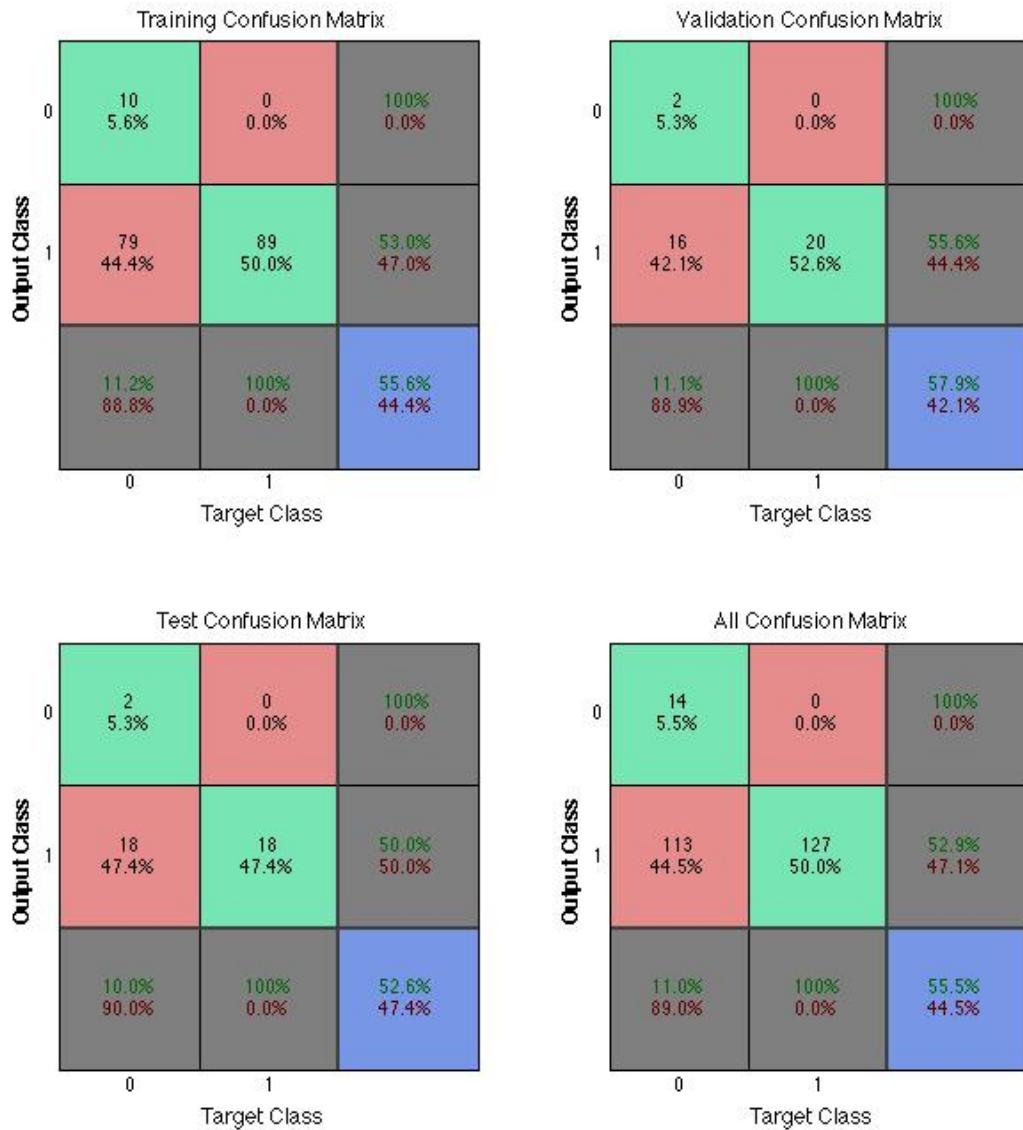


Figure 32. Confusion for the classification using the absolute values of the FFT of the EEG data when segmented for ten seconds.

### 4.3 Third method

Wavelet transformation (Appendix C) is one of the most popular candidates of the time-frequency-transformations. It's capable of providing the time and frequency information simultaneously. Hence giving a time-frequency representation of the signal

In this method, wavelet transformation was employed to extract the features from the EEG signal. The principal motivation behind trying wavelet transformation is my reasoning that better classification over the non-stationary EEG signal may be achieved if the feature extraction method could represent the change of frequency in respect to time.

After preprocessing the EEG signals as mentioned in section 5.2.1, the EEG signals of C3 and C4 channels were decomposed for ten levels using Daubechies' wavelet [3.7] and the energy, variance, and waveform of each level were calculated to form the input to the classifier -the decomposition was done for ten levels as the supervisor of this project advised to be an appropriate number of levels.

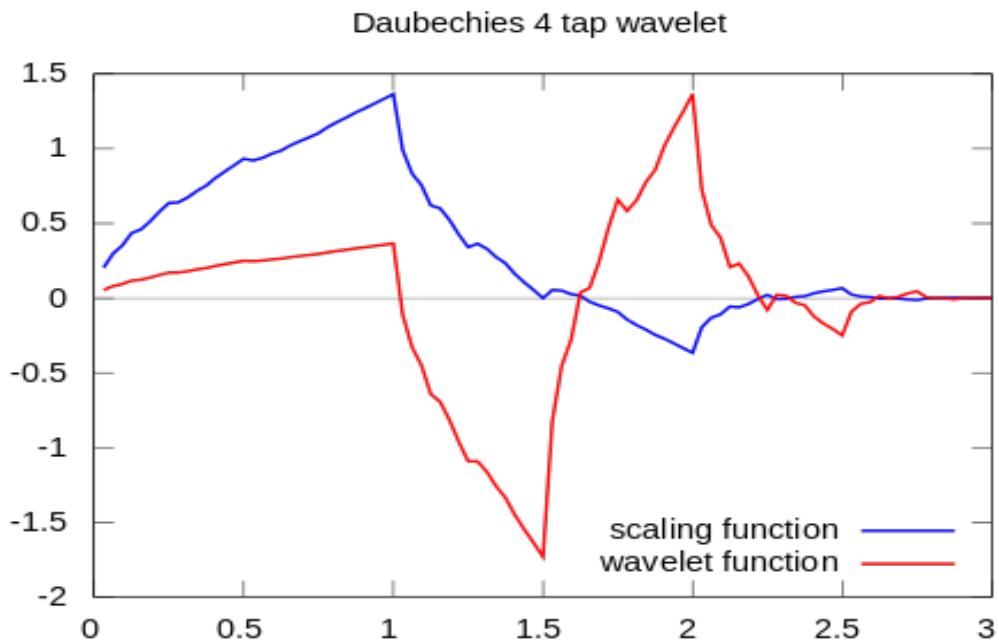


Figure 33. Scaling and wavelet functions.

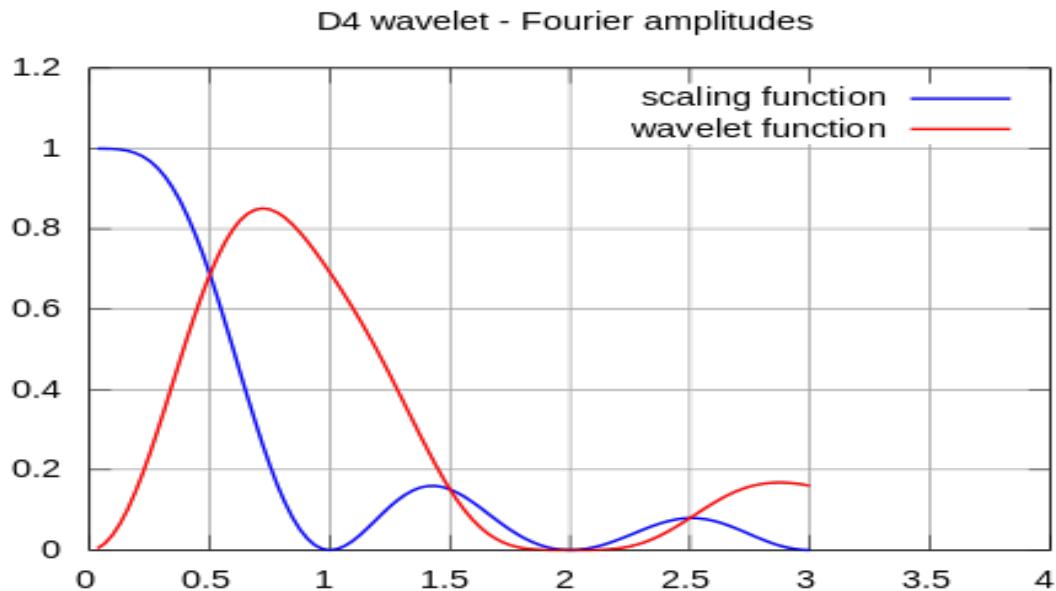


Figure 34. Amplitudes of the frequency spectra of the Fig.33 function

The results of the classification did not meet the expectations where the archived success rate was only 63.2%. The reason behind the low success rate of this method is probably because of the following:

Classification is like a test of the pattern, if the pattern shows features that are close enough to the patterns used in the learning processes, then the pattern in hand will be classified correctly. However, using too many type of features to train the classifier makes the testing of the presented patterns harder to pass and even the patterns that deviate slightly from the training patterns becomes not close enough resulting in misclassifying them. In other words, the usage of too many type features made the classification “tight”.

## 4.4 Forth method

The preprocessing was done in this process corresponds with the preprocessing in the second method. The feature extraction on the other hand is the same as in the third method where Daubechies' wavelet was used to preform the wavelet decomposition at ten levels, only in this method, the coefficients resulted from the decomposition were fed to the classifier as the feature vector with the aim of resolving the problem discussed in the third method. The outcome of this method was, as expected, a superior success rate of 94.7% thanks to the good time-frequency representation of the wavelet decomposition and artifacts removal by the FIR filter.

## Chapter 5 Conclusions

This paper focuses on the classification of EEG signals for right and left fist movements based on a specific set of features. Very good results were obtained using NNs showing that offline discrimination between right and left movement, for executed hand movements, is comparable to leading BCI research. Our methodology is not the best, but is somewhat a simplified efficient one that satisfies the needs for researchers in field of neuroscience.

The role of signal processing is crucial in the development of a real-time Brain Computer Interface. Until recently, several improvements have been made in this area, but none of them has been entirely successful. The goal of creating more effective classification algorithms, have focused numerous investigations in the search of new techniques of feature extraction.

It was seen that feature extraction methods that can represent the change of frequency in term of time such as wavelet transform promises better results when used for classifying EEG signals due to the non-stationary nature of the EEG.

We do not recommend suppressing the 50 Hz line with a notch filter or other sharp filters and better results would probably be obtained if the base line wasn't filtered in our experiments. The reason behind this conclusion is that such filters add phase distortions and delays in addition to the noise and ringing artifacts cased by the filter. This phenomenon was clearly demonstrated by when the FIR filter was used to extract Mu brainwaves in the first method where the filtering process noise greatly affected the signals and decreased the classification success rate.

## Chapter 6

### Future development

If the advances that are ahead of us counterpart the ones happened during the previous few decades, communication between humans and computers in direct neural way may become more developed and its use spread widely. New computers will perhaps one day come with a built in biological sensors and thought-recognition software, and those new technologies become common as a mouse and keyboard are today.

This thesis focused on classifying EEG signals for left and right movements using different combinations of preprocessing and feature extraction. The obtained results were good showing that offline success rate of classifying right and left movement is comparable to leading BCI research. The methodologies in this thesis are probably not the best, but they are somewhat a simplified efficient one that satisfies the needs for future researchers in field of BCI.

Although non-invasive BCIs are already being used for some relatively simple applications, it has been commonly assumed that only invasive BCIs that use implanted electrodes inside the human brain can provide accurate classification for various brain activities and complicated application like controlling a robotic arm with a multidimensional movement can only be successfully done using invasive BCIs, which has the obvious down back of being invasive.

A further development for this project would be to prove classifying EEG signals into more than just two classes using adaptive algorithms applied to electroencephalographic activities recorded from the scalp surface in a non-invasive way. As a results of this development, people with severe disabilities in there motor system can use there thought to control neuroprostheses or robotics arms without the need to implement any electrodes in their brain.

Also, In the near future, I aim to implement the developed algorithm to an online application. That is by using the actual electrodes to get the EEG signal instead of getting them from a database and map the output of the system into a set of two commands to use them to control an actual application. An example of such application is a health systems and computer games.

## References

- [1.1] J. P. Donoghue, "Connecting cortex to machines: recent advances in brain interfaces," *Nature Neuroscience Supplement*, vol. 5, pp. 1085– 1088, 2002.
- [1.2] S. Levine, J. Huggins, S. BeMent, R. Kushwaha, L. Schuh, E. Passaro, M. Rohde, and D. Ross, "Identification of electrocorticogram patterns as the basis for a direct brain interface," *Journal of Clinical Neurophysiology*, vol. 16, pp. 439-447, 1999.
- [1.3] K. L. Kilgore, P. H. Peckham, M. W. Keith, G. B. Thorpe, K. S. Wuolle, A. M. Bryden, and R. L. Hart, "An implanted upper-extremity neuro-prosthesis: Follow-up of five patients," *J. Bone Joint Surg.*, vol. 79A, pp. 533–541, 1997.
- [2.1] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural Science*, 3<sup>rd</sup> ed. New York: Elsevier/North-Holland, 1991.
- [2.2] J. Ochoa, 'EEG signal classification for brain computer interface applications', Ecole Polytechnique Federale De Lausanne, vol 7, pp. 1--72, 2002.
- [2.3] U. Hoffmann, J.-M. Vesin and T. Ebrahimi, "Recent Advances in Brain-Computer Interfaces," IEEE International Workshop on Multimedia Signal Processing (MMSP07), 2007.
- [2.4] George D Fuller, Ph.D. BIOFEEDBACK Methods and Procedures in clinical practice, 1977.
- [2.5] M. Nakamura, T. Sugi, A. Ikeda, R. Kakigi and H. Shibasaki, 'Clinical application of automatic integrative interpretation of awake background. EEG: quantitative interpretation, report making, and detection of artifacts and reduced vigilance level', *Electroencephalography and clinical Neurophysiology*, vol 98, iss 2, pp. 103--112, 1996.
- [2.6] B.W.Jervis, E.C.Ifeachor, and E.M.Allen. The removal o ocular artifacts from the electroencephalogram :a review. *Medical & Biological Engineering & Computing* , 26: 2-12, 1988.
- [2.7] Vigario R, Sarela J, Jousmaki V, Hamalainen M, Oja E. Independent component approach to the analysis of EEG and MEG recordings. *IEEE Trans Biomed Eng*. 47(5): 589-93, 2000.
- [2.8] Jung T-P, Makeig S, Westerfield W, Townsend J, Courchesne E, and Sejnowski TJ, "Removal of eye activity artifacts from visual event-related potentials in normal and clinical subjects" *Clinical Neurophysiology*, 111:10, 1745-58, 2000.
- [2.9] T. Jung, C. Humphries, T. Lee, S. Makeig, M. McKeown, V. Iragui and T. Sejnowski, 'Removing electroencephalographic artifacts: comparison between ICA and PCA', pp. 63-72, 1998.
- [2.10] A. Schl\"ogl and G. Pfurtscheller, 'EOG and ECG minimization based on regression analysis', Institute for Biomedical Engineering, Department of Medical Informatics, Univ. of Techn., Graz, Austria [5.2] P. Flandrin. A time-frequency formulation of optimum detection." *IEEE Trans. Acoust., Speech, Signal Processing*, vol 36, iss 9, pp. 1377--1384, 1988.

- [2.11] F. Cincotti, A. Scipione, A. Tinipéri, D. Mattia, M.G. Marciani, J. del R. Millán, S. Salinari, L. Bianchi, and F. Babiloni. Comparison of different feature classifiers for brain computer interfaces. In Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering, 2003.
- [2.12] R. Bogacz, U. Markowska-Kaczmar, A. Kozik. Blinking artefact recognition in the EEG signal using artificial neural network. In Proc. of 4 Conference on Neural Networks and Their Applications, Zakopane (Poland).
- [2.13] Gupta, S.; Singh, H. Preprocessing EEG signals for direct human-system interface. Intelligence and Systems, 1996., IEEE International Joint Symposia on , 1996 Page(s): 32 –37.
- [2.14] M. Kaper, P. Meinicke, U. Grossekathofer, T. Lingner, and H. Ritter. Bci competition 2003–data set iib: support vector machines for the p300 speller paradigm. *IEEE Transactions on Biomedical Engineering*, 51(6):1073–1076, 2004.
- [2.15] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer. Eeg-based discrimination, between imagination of right and left hand movement. *Electroencephalography and Clinical Neurophysiology*, 103:642–651, 1997.
- [2.16] S. Chiappa and S. Bengio. Hmm and iohmm modeling of eeg rhythms for asynchronous bci systems. In European Symposium on Artificial Neural Networks ESANN, 2004.
- [2.17] W. D. Penny, S. J. Roberts, E. A. Curran, and M. J. Stokes. Eeg-based communication: a pattern recognition approach. *IEEE Transactions on Rehabilitation Engeneering*, 8(2):214–215, 2000.
- [2.18] T. Wang, J. Deng, and B. He. Classifying eeg-based motor imagery tasks by means of time- frequency synthesized spatial patterns. *Clinical Neurophysiology*, 115(12):2744–2753, 2004.
- [2.19] L. Qin, L. Ding, and B. He. Motor imagery classification by means of source analysis for brain computer interface applications. *Journal of Neural Engineering*, pages 135–141, 2004.
- [2.20] J. H. K. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1), 1997.
- [2.21] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Recognition, second edition. WILEY-INTERSCIENCE, 2001.
- [2.22] 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. *IEEE Transactions on Biomedical Engineering*, 51(6), 2004.
- [2.23] S. Solhjoo, A. M. Nasrabadi, and M. R. H. Golpayegani. Classification of chaotic signals using hmm classifiers: Eeg-based mental task classification. In Proceedings of the European Signal Processing Conference, 2005.
- [2.24] B. Obermeier, C. Guger, C. Neuper, and G. Pfurtscheller. Hidden markov models for online classification of single trial eeg. *Pattern recognition letters*, pages 1299–1309, 2001.

- [2.25] 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 Transactions on Neural System and Rehabilitation Engineering*, 11:141–144, 2003.
- [2.26] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.
- [2.27] G. N. Garcia, T. Ebrahimi, and J.-M. Vesin. Support vector eeg classification in the fourier and time-frequency correlation domains. In *Conference Proceedings of the First International IEEE EMBS Conference on Neural Engineering*, 2003.
- [2.28] B. Blankertz, G. Curio and K. M\"uller, 'Classifying single trial EEG: Towards brain computer interfacing', *Advances in neural information processing systems*, vol 1, pp. 157--164, 2002.
- [2.29] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2, 1998.
- [2.30] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition : A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.
- [2.31] C. W. Anderson and Z. Sijercic. Classification of eeg signals from four subjects during five mental tasks. In *Solving Engineering Problems with Neural Networks: Proceedings of the International Conference on Engineering Applications of Neural Networks (EANN'96)*, 1996.).
- [2.32] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.
- [2.33] S. Solhjoo and M. H. Moradi. Mental task recognition: A comparison between some of classification methods. In *BIOSIGNAL 2004 International EURASIP Conference*, 2004.
- [2.34] K. Fukunaga. *Statistical Pattern Recognition*, seconde edition. ACADEMIC PRESS, INC, 1990.
- [2.35] S. Rezaei, K. Tavakolian, A. M. Nasrabadi, and S. K. Setarehdan. Different classification techniques considering brain computer interface applications. *Journal of Neural Engineering*, 3:139–144, 2006.
- [2.36] T. Felzer and B. Freisieben. Analyzing eeg signals using the probability estimating guarded neural classifier. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(4), 2003.
- [2.37] M. Congedo, F. Lotte, and A. L'ecuyer. Classsification of movement intention by spatially filtered electromagnetic inverse solutions. *Physics in Medicine and Biology*, 51(8):1971–1989, 2006.
- [2.38] J. R. Mill\'an, J. Mourio, F. Cincotti, F. Babiloni, M. Varsta, and J. Heikkonen. Local neural classifier for eeg-based recognition of mental tasks. In *IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 2000.
- [2.39] 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 on Biomedical Engineering*, 51(6), 2004.

- [2.40] V. Bostanov. Bci competition 2003–data sets ib and iib: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram. *IEEE Transactions on Biomedical Engeneering*, 51(6):1057–1061, 2004.
- [2.41] G. Pfurtscheller. Eeg event-related desynchronization (erd) and event-related synchronization (ers). 1999.
- [2.42] Advances in Neural Information Processing Systems (NIPS 01), 14:157–164, 2002.
- [2.43] Dtreg.com, 'Multilayer Perceptron Neural Networks', 2014. [Online]. Available: <http://www.dtreg.com/mlfn.htm>. [Accessed: 04- May- 2014].
- [2.44] R. Hecht-Nielsen, 'Theory of the backpropagation neural network', pp. 593--605, 1989.
- [3.1] Sccn.ucsd.edu, (2014). EEGLAB - Open Source Matlab Toolbox for Electrophysiological Research. [online] Available at: <http://sccn.ucsd.edu/eeglab/> [Accessed 26 Apr. 2014].
- [3.2] J. Sleight, P. Pillai, and S. Mohan, "Classification of Executed and Imagined Motor Movement EEG Signals," Ann Arbor: University of Michigan, 2009, pp. 1-10.
- [3.3] L. Deecke, H. Weinberg, and P. Brickett, "Magnetic fields of the human brain accompanying voluntary movements: Bereitschaftsmagnetfeld," *Experimental Brain Research*, vol. 48, pp. 144-148, 1982.
- [3.4] C. Neuper and G. Pfurtscheller, "Evidence for distinct beta resonance frequencies in human EEG related to specific sensorimotor cortical areas," *Clinical Neurophysiology*, vol. 112, pp. 2084-2097, 2001.
- [3.5] Eeginfo-europe.com, (2014). Impedance Measurement - EEG Info Europe. [online] Available at: <http://www.eeginfo-europe.com/neurofeedback/support/technical-infos-faq/impedance-measurement.html> [Accessed 25 Apr. 2014].
- [3.6] N. Hazarika, J. Chen, A. Tsoi and A. Sergejew, 'Classification of EEG signals using the wavelet transform', *Signal processing*, vol 59, iss 1, pp. 61--72, 1997.
- [3.7] "Orthonormal Bases of Compactly Supported Wavelets," I. Daubechies, *Communications on Pure and Applied Mathematics*, Vol. XL1, pp. 090-996, 1988.
- [3.8] Sites.google.com, 'projectbci', 2014. [Online]. Available: <https://sites.google.com/site/projectbci/>. [Accessed: 06- May- 2014].

# Appendices

## 1 Appendix A: Backpropagation

### Derivation of Backpropagation

#### 1 Introduction

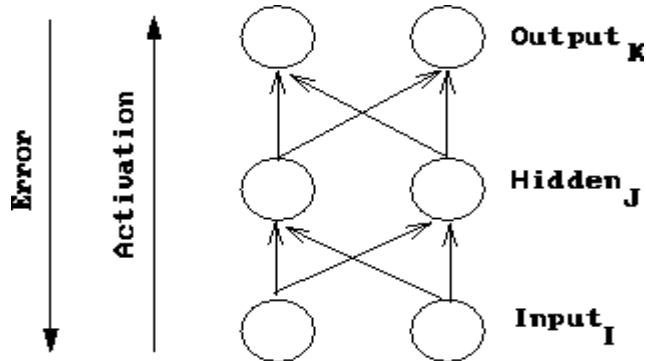


Figure 1: Neural network processing

Conceptually, a network forward propagates activation to produce an output and it backward propagates error to determine weight changes (as shown in Figure 1). The weights on the connections between neurons mediate the passed values in both directions.

The Backpropagation algorithm is used to learn the weights of a multilayer neural network with a fixed architecture. It performs gradient descent to try to minimize the sum squared error between the network's output values and the given target values.

Figure 2 depicts the network components which affect a particular weight change. Notice that all the necessary components are locally related to the weight being updated. This is one feature of backpropagation that seems biologically plausible. However, brain connections appear to be unidirectional and not bidirectional as would be required to implement backpropagation.

#### 2 Notation

For the purpose of this derivation, we will use the following notation:

- The subscript  $k$  denotes the output layer.
- The subscript  $j$  denotes the hidden layer.
- The subscript  $i$  denotes the input layer.

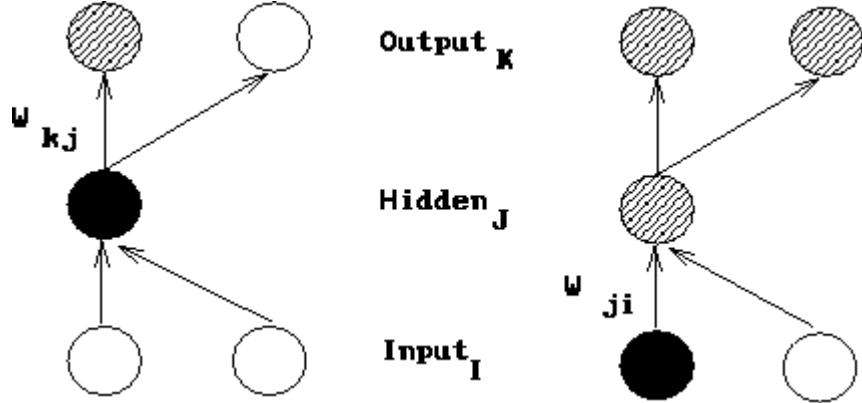


Figure 2: The change to a hidden to output weight depends on error (depicted as a lined pattern) at the output node and activation (depicted as a solid pattern) at the hidden node. While the change to a input to hidden weight depends on error at the hidden node (which in turn depends on error at all the output nodes) and activation at the input node.

- $w_{kj}$  denotes a weight from the hidden to the output layer.
- $w_{ji}$  denotes a weight from the input to the hidden layer.
- $a$  denotes an activation value.
- $t$  denotes a target value.
- $net$  denotes the net input.

### 3 Review of Calculus Rules

$$\frac{d(e^u)}{dx} = e^u \frac{du}{dx} \quad \frac{d(g+h)}{dx} = \frac{dg}{dx} + \frac{dh}{dx} \quad \frac{d(g^n)}{dx} = n g^{n-1} \frac{dg}{dx}$$

### 4 Gradient Descent on Error

We can motivate the backpropagation learning algorithm as gradient descent on sum-squared error (we square the error because we are interested in its magnitude, not its sign). The total error in a network is given by the following equation (the  $\frac{1}{2}$  will simplify things later).

$$E = \frac{1}{2} \sum_k (t_k - a_k)^2$$

We want to adjust the network's weights to reduce this overall error.

$$\Delta W \propto -\frac{\partial E}{\partial W}$$

We will begin at the output layer with a particular weight.

$$\Delta w_{kj} \propto -\frac{\partial E}{\partial w_{kj}}$$

However error is not directly a function of a weight. We expand this as follows.

$$\Delta w_{kj} = -\varepsilon \frac{\partial E}{\partial a_k} \frac{\partial a_k}{\partial \text{net}_k} \frac{\partial \text{net}_k}{\partial w_{kj}}$$

Let's consider each of these partial derivatives in turn. Note that only one term of the  $E$  summation will have a non-zero derivative: the one associated with the particular weight we are considering.

#### 4.1 Derivative of the error with respect to the activation

$$\frac{\partial E}{\partial a_k} = \frac{\partial(\frac{1}{2}(t_k - a_k)^2)}{\partial a_k} = -(t_k - a_k)$$

Now we see why the  $\frac{1}{2}$  in the  $E$  term was useful.

#### 4.2 Derivative of the activation with respect to the net input

$$\frac{\partial a_k}{\partial \text{net}_k} = \frac{\partial(1 + e^{-\text{net}_k})^{-1}}{\partial \text{net}_k} = \frac{e^{-\text{net}_k}}{(1 + e^{-\text{net}_k})^2}$$

We'd like to be able to rewrite this result in terms of the activation function. Notice that:

$$1 - \frac{1}{1 + e^{-\text{net}_k}} = \frac{e^{-\text{net}_k}}{1 + e^{-\text{net}_k}}$$

Using this fact, we can rewrite the result of the partial derivative as:

$$a_k(1 - a_k)$$

#### 4.3 Derivative of the net input with respect to a weight

Note that only one term of the  $\text{net}$  summation will have a non-zero derivative: again the one associated with the particular weight we are considering.

$$\frac{\partial \text{net}_k}{\partial w_{kj}} = \frac{\partial(w_{kj}a_j)}{\partial w_{kj}} = a_j$$

#### 4.4 Weight change rule for a hidden to output weight

Now substituting these results back into our original equation we have:

$$\Delta w_{kj} = \varepsilon \overbrace{(t_k - a_k)a_k(1 - a_k)}^{\delta_k} a_j$$

Notice that this looks very similar to the Perceptron Training Rule. The only difference is the inclusion of the derivative of the activation function. This equation is typically simplified as shown below where the  $\delta$  term represents the product of the error with the derivative of the activation function.

$$\Delta w_{kj} = \varepsilon \delta_k a_j$$

## 4.5 Weight change rule for an input to hidden weight

Now we have to determine the appropriate weight change for an input to hidden weight. This is more complicated because it depends on the error at all of the nodes this weighted connection can lead to.

$$\Delta w_{ji} \propto -\left[ \sum_k \frac{\partial E}{\partial a_k} \frac{\partial a_k}{\partial net_k} \frac{\partial net_k}{\partial a_j} \right] \frac{\partial a_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}}$$

$$= \varepsilon \left[ \sum_k \overbrace{(t_k - a_k) a_k (1 - a_k)}^{\delta_k} w_{kj} \right] a_j (1 - a_j) a_i$$

$$= \varepsilon \overbrace{\left[ \sum_k \delta_k w_{kj} \right]}^{\delta_j} a_j (1 - a_j) a_i$$

$$\Delta w_{ji} = \varepsilon \delta_j a_i$$

## 2 Appendix B: Fourier transform

# APPENDIX B

---

## FOURIER ANALYSIS AND THE FAST FOURIER TRANSFORM (FFT)

This is a brief introduction to the fast Fourier transform (FFT) [1], particularly for the applications in this book.

### B.1 THE STRUCTURE OF THE FFT

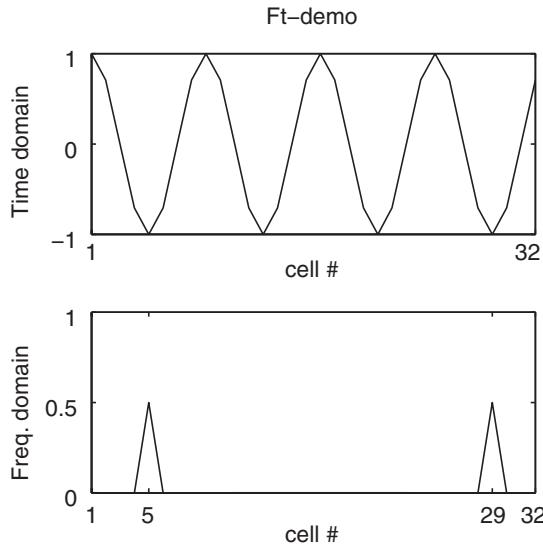
Figure B.1 shows a cosine function and the Fourier transform of this function using the MATLAB command **fft**. For simplicity, we will assume that the cosine function is a time-domain function and that the units are seconds.

The Fourier transform of a cosine is two delta functions:

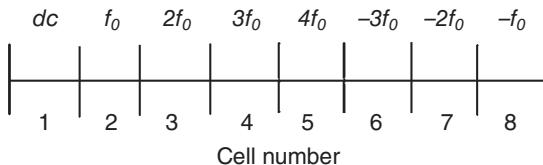
$$\mathcal{F}\{\cos(2\pi f_o t)\} = \pi[\delta(f + f_0) + \delta(f - f_0)].$$

The two delta functions appear in the bottom part of Figure B.1, but not where we expect them. This takes some explanation. The MATLAB command **fft** implements a “fast Fourier transform.” (This is the algorithm that made Fourier transforms practical in calculating convolutions.) The **fft** puts the positive frequencies in the first  $N/2$  positions, and the negative frequencies in the last  $N/2$  positions. This is illustrated in Figure B.2 for a buffer that is eight cells.

The lowest frequency that can be represented is one that takes the entire buffer. So if the buffer is eight cells, the lowest frequency is  $f_0 = 1/8$ . The highest



**FIGURE B.1** A cosine function in the time domain (top) and its FFT in the frequency domain (bottom).



**FIGURE B.2** Diagram of where the `fft` command stores the different frequencies.

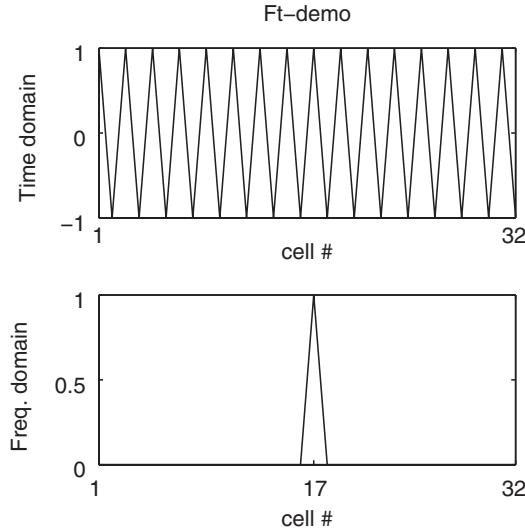
frequency will be  $f_{\max} = 4/8 = 1/2$ . If  $N$  is the total number of cells, then the highest frequency is  $N/2$ . This frequency occupies only one cell in the Fourier domain, as shown in Figure B.3.

Obviously, we can make a similar representation for any frequency that is an integer division of  $N$ .

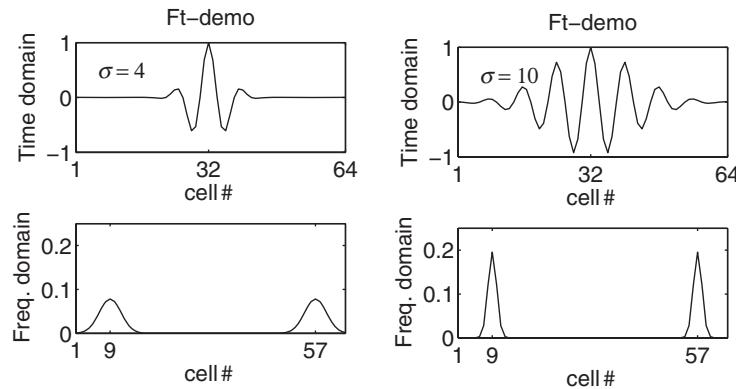
Let us look at some other waveforms, as shown in Figure B.4. Each of these waveforms has a center frequency of  $8/N$ . The first one is in a fairly narrow envelope. In the Fourier domain, it is centered at  $8 + 1 = 9$ , but it is also spread out somewhat. The second pulse has a much broader envelope, but its Fourier transform is very narrow. Is this consistent with what we know about Fourier theory?

Let us look at another example. Figure B.5 shows a sinusoid with a wavelength of 10 cells. Unfortunately, 10 is not an integer division of our buffer size, which is 64. So the Fourier transform is spread out.

If we want more accuracy, a simple solution exists: Do the transforms in a longer buffer. The buffer with 64 cells has an accuracy of  $\Delta f = 1/64$ . If we go to a buffer with 256 cells, our accuracy is  $\Delta f = 1/256$  (Fig. B.6).



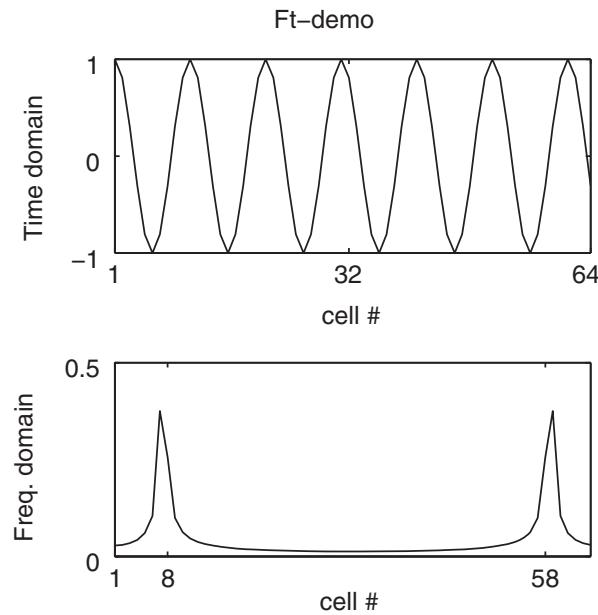
**FIGURE B.3** The FFT of the highest frequency for the 32-cell buffer.



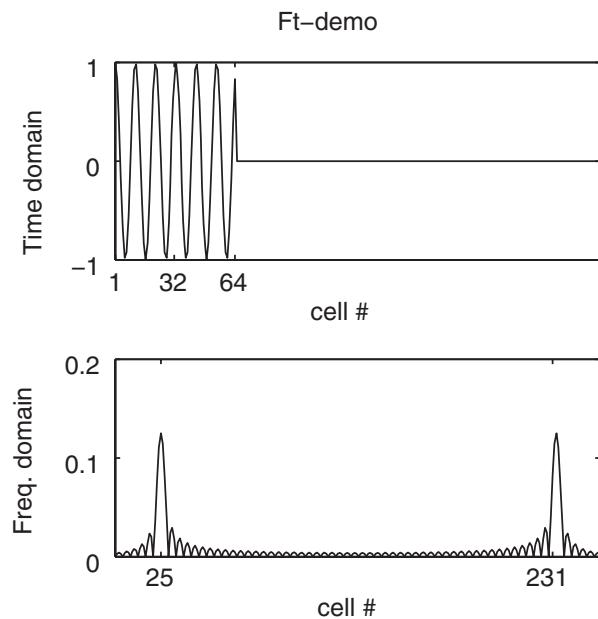
**FIGURE B.4** Two pulses at frequency  $8/N$  contained within Gaussian envelopes.

## B.2 WINDOWING

The Fourier domain waveform of Figure B.6 still looks a little ragged because when we move the sinusoid to the larger buffer, we abruptly truncate the signal at 64. The result is the same as if we had multiplied the time-domain data by a rectangular function. A sharp transition in the time domain leads to ripples in the frequency domain (Fig. B.7, left side). Instead of this abrupt truncation, we can “window” the data, that is, multiply it by a function that results in a smooth transition [2].

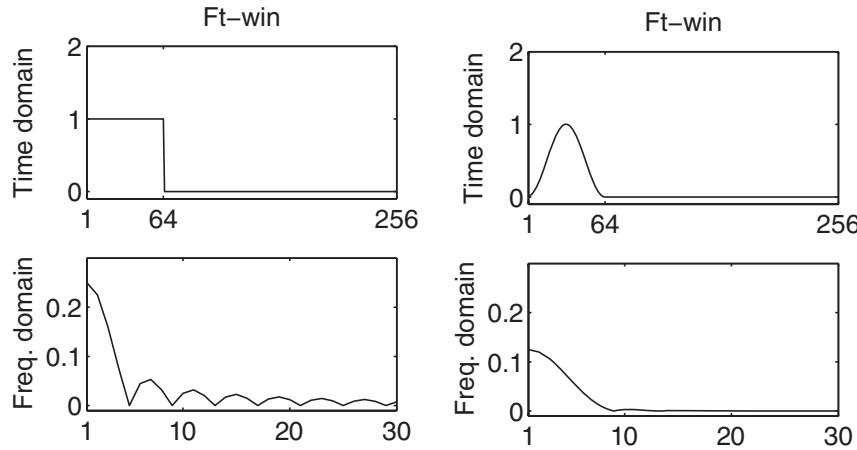


**FIGURE B.5** Fourier transform of frequency 1/10.

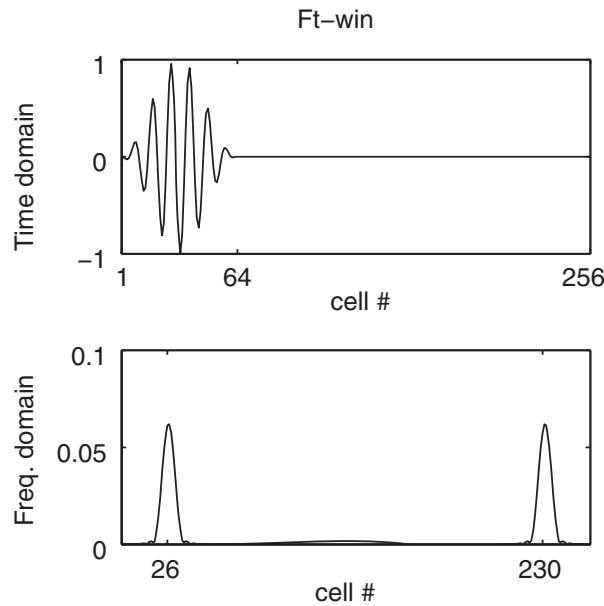


**FIGURE B.6** Fourier transform of the same signal shown in Figure B.5, but with a buffer of 256 cells.

One such window, referred to as the Hanning window, is shown on the right side of Figure B.7. Notice how much smoother the **fft** of the Hanning window is compared to the rectangular function. Many different windows are used to optimize different parameters, but they all tend to have a Gaussian-like shape [3].



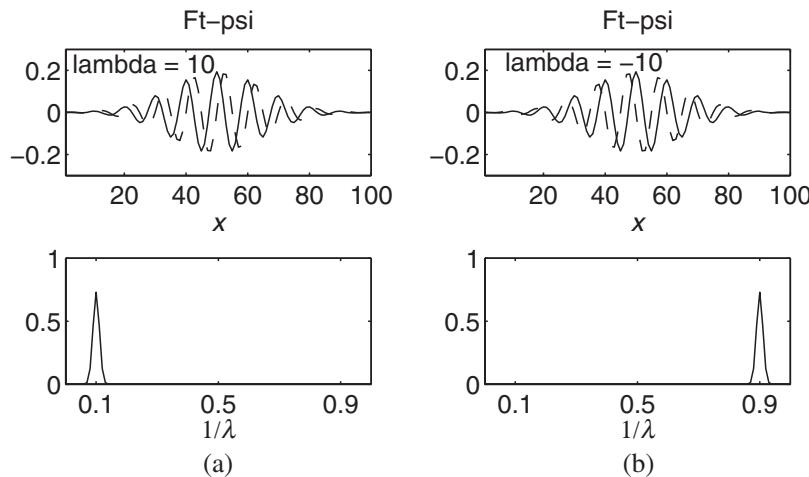
**FIGURE B.7** Fourier transform of a rectangular function (left) and the Hanning window (right).



**FIGURE B.8** A windowed sinusoid and its Fourier transform.

When we window the sinusoidal data before transforming, we get the smoother result shown in Figure B.8.

Until now, we have assumed that the data represent time domain data. However, we can represent waveforms in the spatial domain as well. Suppose Figure B.1 represents a cosine in the space domain where each cell is  $1 \mu\text{m}$ . Then the Fourier transform would take the signal to the inverse wavelength, or  $1/\lambda$ . So for the 64-cell buffer in Figure B.1, the lowest frequency would be  $1/32 \mu\text{m}$ , and the Fourier transform would produce spikes at  $\pm 4/32 \mu\text{m}$ .



**FIGURE B.9** (a) The top diagram shows a spatial Gaussian pulse traveling in the positive direction. The center wavelength is 10; the Fourier transform of the pulse is illustrated on the bottom. Notice that it is centered at 0.1 or  $1/10$ . (b) A similar pulse traveling in the negative direction.

### B.3 FFT OF THE STATE VARIABLE

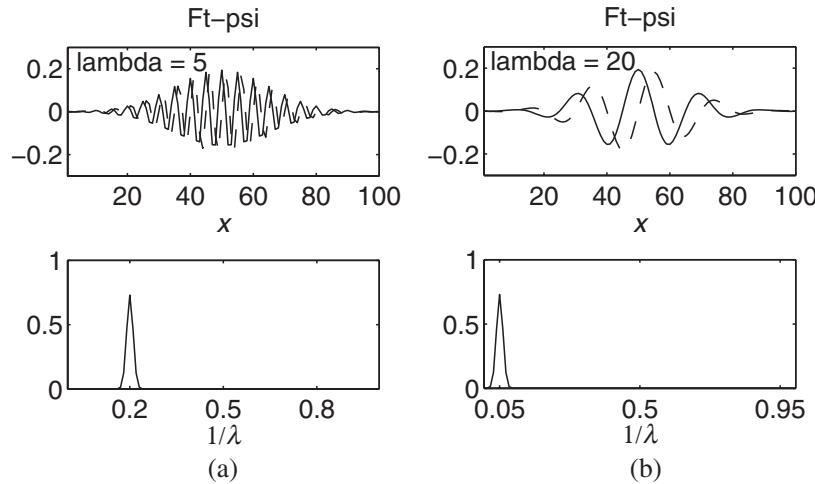
So far, we have been using only real functions in time or space. If Figure B.1 represented the electric field of an electromagnetic pulse we could not tell if it was propagating to the left or right, at least not without seeing the magnetic field as well. In quantum mechanics, we do not have this problem. The state variable  $\psi$  is complex, and the phase between the real and imaginary part dictates its direction.

Figure B.9 shows two complex pulses with their respective Fourier transforms. Pulse (a) is traveling from left to right, as indicated by the fact that its Fourier transform only has positive frequencies. Pulse (b) by contrast, is moving right to left, evident from its negative frequencies.

The position in the frequency domain is easy to understand if we once again think about where the different frequency components are stored in the **fft** buffer, similar to Figure B.2. The difference is that we are using spatial frequencies. The longest wavelength in the 100-cell buffer is 100. The wavelength of 10 is going to be the  $n = 100/10 = 10$  cell, which is the eleventh cell from the right. The pulse moving in the negative direction will be in the tenth cell from the right.

Figure B.10 shows two similar pulses with different wavelengths.

Physicists prefer plots as functions of  $k = 2\pi/\lambda$ . However, with functions of  $k$ , the axis does not show anything that corresponds directly to the waveform.



**FIGURE B.10** Space domain and wavelength domain plots for wavelengths of (a) five cells and (b) 20 cells.

## EXERCISES

- B.1** Use the program `ft_demo.m` to duplicate the results of Figure B.1. Replace the input with the following wavelengths: 64, 32, and 2 cells. Does the `fft` give you the results you expect?
- B.2** Duplicate the results of Exercise B.1 with the following differences: Use  $N = 100$  instead of 64 for the size of the problem space, but still use the wavelengths of 64, 32, and 2 cells. Assume the cells sizes are 1 nm and label the axis in the space and the  $1/\lambda$  domain.
- B.3** Repeat Exercise B.2, but assume you are using time and frequency. Assume the time-domain cells are 1 fs, and label your `fft` domain appropriately.
- B.4** Using the program `ft_psi.m`, initialize a complex wave traveling in the positive direction with a center wavelength of 20 cells inside a Gaussian pulse. Add another domain that plots the `fft` as a function of energy, similar to Figure 3.10.

## REFERENCES

1. E. O. Brigham, *The Fast Fourier Transform and Its Applications*, Englewood Cliffs, NJ: Prentice Hall, 1988.
2. A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1975.
3. E. P. Cunningham, *Digital Filtering: An Introduction*, New York: John Wiley and Sons, 1995.

### 3 Appendix C: Wavelet transform

Discrete Wavelet Transforms \_\_\_\_\_ Draft #2, June 4, 1992

## Discrete Wavelet Transforms: Theory and Implementation

Tim Edwards ([tim@sinh.stanford.edu](mailto:tim@sinh.stanford.edu))

Stanford University, September 1991

### Abstract

Section 2 of this paper is a brief introduction to wavelets in general and the discrete wavelet transform in particular, covering a number of implementation issues that are often missed in the literature; examples of transforms are provided for clarity. The hardware implementation of a discrete wavelet transform on a commercially available DSP system is described in Section 3, with a discussion on many resultant issues. A program listing is provided to show how such an implementation can be simulated, and results of the hardware transform are presented. The hardware transformer has been successfully run at a voice-band rate of 10kHz, using various wavelet functions and signal compression.

## 1 Introduction

The field of Discrete Wavelet Transforms (DWTs) is an amazingly recent one. The basic principles of wavelet theory were put forth in a paper by Gabor in 1945 [4], but all of the definitive papers on discrete wavelets, an extention of Gabor's theories involving functions with compact support, have been published in the past three years. Although the Discrete Wavelet Transform is merely one more tool added to the toolbox of digital signal processing, it is a very important concept for data compression. Its utility in image compression has been effectively demonstrated. This paper discusses the DWT and demonstrates one way in which it can be implemented as a real-time signal processing system. Although this paper will attempt to describe a very general implementation, the actual project used the STAR Semiconductor SPROClab digital signal processing system.<sup>1</sup> A complete wavelet transform system as described herein is available from STAR Semiconductor [7].

## 2 A Brief Discussion of Wavelets

### 2.1 What a Wavelet Is

The following discussion on wavelets is based on a presentation by the mathematician Gilbert Strang [8], whose paper provides a good foundation for understanding wavelets, and includes a number of derivations that are not given here.

A wavelet, in the sense of the Discrete Wavelet Transform (or DWT), is an orthogonal function which can be applied to a finite group of data. Functionally, it is very much like the Discrete Fourier Transform, in that the transforming function is orthogonal, a signal passed twice through the transformation is unchanged, and the input signal is assumed to be a set of discrete-time samples. Both transforms are convolutions.

---

<sup>1</sup>This research was carried out at Stanford University and supported by STAR Semiconductor, Inc. of Warren, New Jersey; and Apple Computer Inc. of Cupertino, California.

Whereas the basis function of the Fourier transform is a sinusoid, the wavelet basis is a set of functions which are defined by a recursive difference equation

$$\phi(x) = \sum_{k=0}^{M-1} c_k \phi(2x - k), \quad (1)$$

where the range of the summation is determined by the specified number of nonzero coefficients  $M$ . The number of nonzero coefficients is arbitrary, and will be referred to as the *order* of the wavelet. The value of the coefficients is, of course, not arbitrary, but is determined by constraints of orthogonality and normalization. Generally, the area under the wavelet “curve” over all space should be unity, which requires that

$$\sum_k c_k = 2. \quad (2)$$

Equation (1) is orthogonal to its translations; i.e.,  $\int \phi(x)\phi(x-k)dx = 0$ . What is also desired is an equation which is orthogonal to its dilations, or scales; i.e.,  $\int \psi(x)\psi(2x - k)dx = 0$ . Such a function  $\psi$  does exist, and is given by

$$\psi(x) = \sum_k (-1)^k c_{1-k} \phi(2x - k), \quad (3)$$

which is dependent upon the solution of  $\phi(x)$ . Normalization requires that

$$\sum_k c_k c_{k-2m} = 2\delta_{0m} \quad (4)$$

which means that the above sum is zero for all  $m$  not equal to zero, and that the sum of the squares of all coefficients is two. Another important equation which can be derived from the above conditions and equations is

$$\sum_k (-1)^k c_{1-k} c_{k-2m} = 0. \quad (5)$$

A good way to solve for values of equation (1) is to construct a matrix of coefficient values. This is a square  $M \times M$  matrix where  $M$  is the number of nonzero coefficients. The matrix is designated  $L$ , with entries  $L_{ij} = c_{2i-j}$ . This matrix always has an eigenvalue equal to 1, and its corresponding (normalized) eigenvector contains, as its components, the value of the  $\phi$  function at integer values of  $x$ . Once these values are known, all other values of the function  $\phi(x)$  can be generated by applying the recursion equation to get values at half-integer  $x$ , quarter-integer  $x$ , and so on down to the desired dilation. This effectively determines the accuracy of the function approximation.

This class of wavelet functions is constrained, by definition, to be zero outside of a small interval. This is what makes the wavelet transform able to operate on a finite set of data, a property which is formally called “compact support.” Most wavelet functions, when plotted, appear to be extremely irregular. This is due to the fact that the recursion equation assures that a wavelet  $\phi$  function is non-differentiable *everywhere*. The functions which are normally used for performing transforms consist of a few sets of well-chosen coefficients resulting in a function which has a discernible shape. Two of these functions are shown in Figure 1; the first is the Haar basis function, chosen because of its simplicity for the following discussion of wavelets, and the second is the Daubechies-4 wavelet, chosen for its usefulness

Wavelet	$c_0$	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$
Haar	1.0	1.0				
Daubechies-4	$\frac{1}{4}(1 + \sqrt{3})$	$\frac{1}{4}(3 + \sqrt{3})$	$\frac{1}{4}(3 - \sqrt{3})$	$\frac{1}{4}(1 - \sqrt{3})$		
Daubechies-6	0.332671	0.806891	0.459877	-0.135011	-0.085441	0.035226

Table 1: Coefficients for three named wavelet functions.

in data compression. They are named for pioneers in wavelet theory [3, 5].<sup>2</sup> The nonzero coefficients  $c_k$  which determine these functions are summarized in Table 1. Coefficients for the Daubechies-6 wavelet, one used in the discussion of the wavelet transformer hardware implementation, are also given in Table 1.

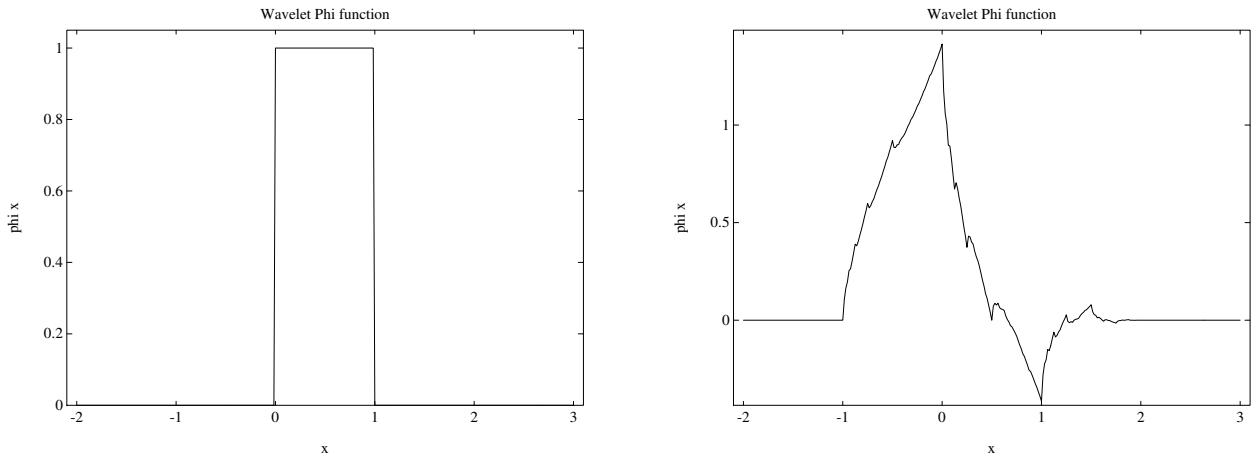


Figure 1: The Haar and Daubechies-4 wavelet basis functions.

The Mallat ‘‘pyramid’’ algorithm [6] is a computationally efficient method of implementing the wavelet transform, and this is the one used as the basis of the hardware implementation described in Section 3. The lattice filter is equivalent to the pyramid algorithm except that a different approach is taken for the convolution, resulting in a different set of coefficients, related to the usual wavelet coefficients  $c_k$  by a set of transformations. A proof of this relation is given in Appendix A, using results from [1].

## 2.2 The Pyramid Algorithm

The pyramid algorithm operates on a finite set of  $N$  input data, where  $N$  is a power of two; this value will be referred to as the *input block size*. These data are passed through two convolution functions, each of which creates an output stream that is half the length of the original input. These convolution functions are filters; one half of the output is produced by

<sup>2</sup>A Daubechies- $n$  wavelet is one of a family of wavelets derived from certain solutions to the wavelet equations that cause these functions to make a best fit of data to a polynomial of degree  $n$ . The Haar wavelet makes a best fit of data to a constant value.

the “low-pass” filter function, related to equation (1):

$$a_i = \frac{1}{2} \sum_{j=1}^N c_{2i-j+1} f_j, \quad i = 1, \dots, \frac{N}{2}, \quad (6)$$

and the other half is produced by the “high-pass” filter function, related to equation (3):

$$b_i = \frac{1}{2} \sum_{j=1}^N (-1)^{j+1} c_{j+2-2i} f_j, \quad i = 1, \dots, \frac{N}{2}. \quad (7)$$

where  $N$  is the input block size,  $c$  are the coefficients,  $f$  is the input function, and  $a$  and  $b$  are the output functions. (In the case of the lattice filter, the low- and high-pass outputs are usually referred to as the odd and even outputs, respectively.) The derivation of these equations from the original  $\phi$  and  $\psi$  equations can be found in [3]. In many situations, the odd, or low-pass output contains most of the “information content” of the original input signal. The even, or high-pass output contains the difference between the true input and the value of the reconstructed input if it were to be reconstructed from only the information given in the odd output. In general, higher-order wavelets (i.e., those with more non-zero coefficients) tend to put more information into the odd output, and less into the even output. If the average amplitude of the even output is low enough, then the even half of the signal may be discarded without greatly affecting the quality of the reconstructed signal. An important step in wavelet-based data compression is finding wavelet functions which causes the even terms to be nearly zero.

The Haar wavelet is useful for explanations because it represents a simple interpolation scheme. For example, a sampled sine wave of sixteen data points (note that this is a power of two, as required) is shown in Figure 2. After passing these data through the filter functions,

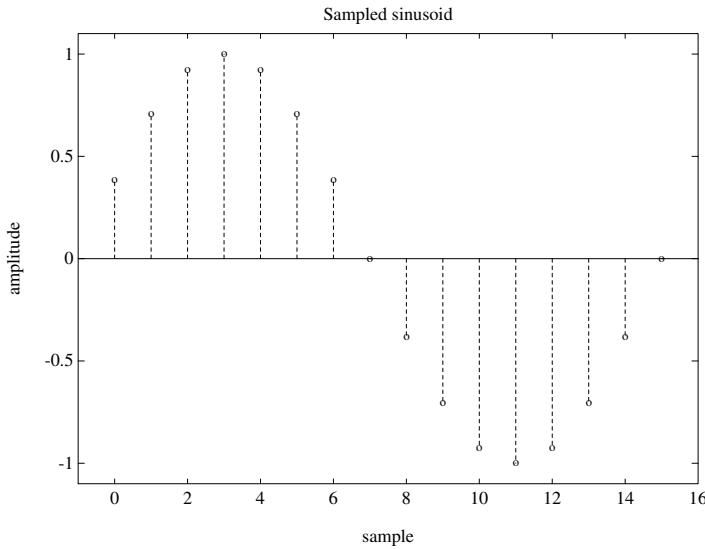


Figure 2: Sampled sinusoid.

the output of the low-pass filter consists of the *average* of every two samples, and the output of the high-pass filter consists of the *difference* of every two samples (see Figure 3). The

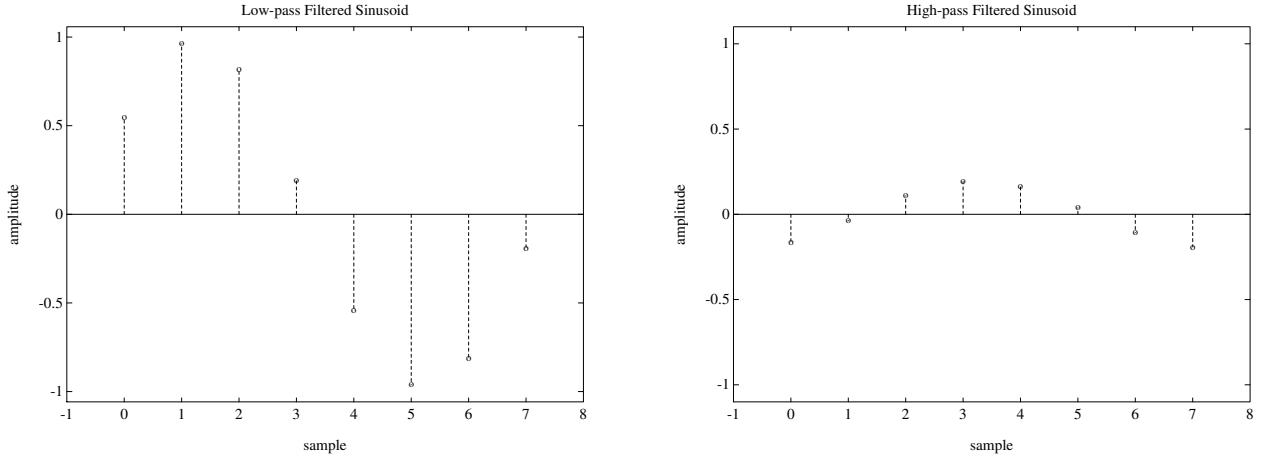


Figure 3: Low- and high-pass outputs from wavelet decomposition of the sampled sinusoid.

high-pass filter obviously contains less information than the low-pass output. If the signal is reconstructed by an inverse low-pass filter of the form

$$f_j^L = \sum_{i=1}^{N/2} c_{2i-j} a_i, \quad j = 1, \dots, N, \quad (8)$$

then the result is a duplication of each entry from the low-pass filter output. This is a wavelet reconstruction with  $2\times$  data compression. Since the perfect reconstruction is a sum of the inverse low-pass and inverse high-pass filters, the output of the inverse high-pass filter can be calculated; it looks like that shown in Figure 4. This is the result of the inverse high-pass

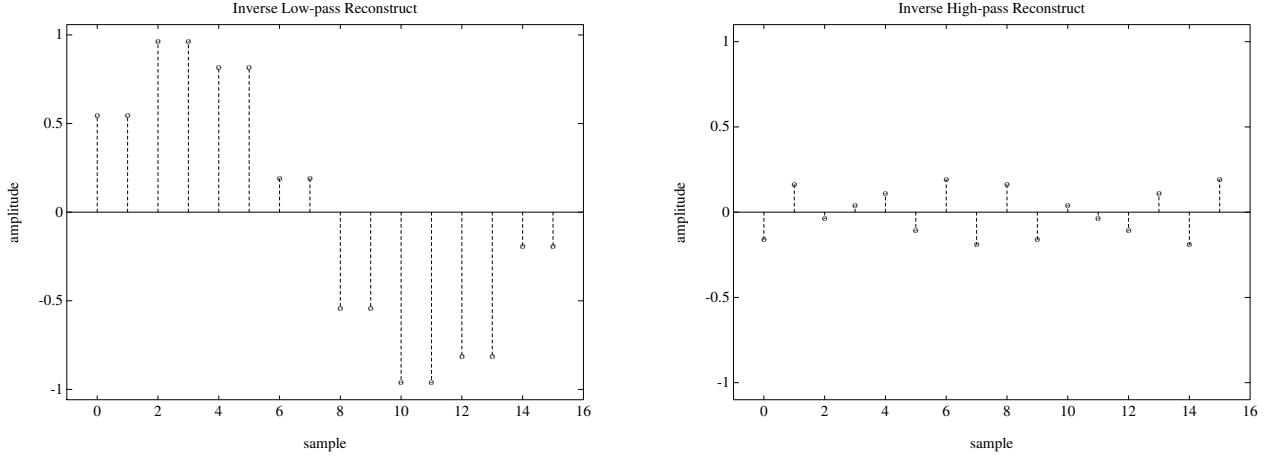


Figure 4: Inverse low- and high-pass filter outputs in sampled sinusoid reconstruction.

filter function

$$f_j^H = \sum_{i=1}^{N/2} (-1)^{j+1} c_{j+1-2i} b_i, \quad j = 1, \dots, N. \quad (9)$$

The perfectly reconstructed signal is

$$f = f^L + f^H, \quad (10)$$

where each  $f$  is the vector with elements  $f_j$ . Using other coefficients and other orders of wavelets yields similar results, except that the outputs are not exactly averages and differences, as in the case using the Haar coefficients.

## 2.3 Dilation

Since most of the information exists in the low-pass filter output, one can imagine taking this filter output and transforming it again, to get two new sets of data, each one quarter the size of the original input. If, again, little information is carried by the high-pass output, then

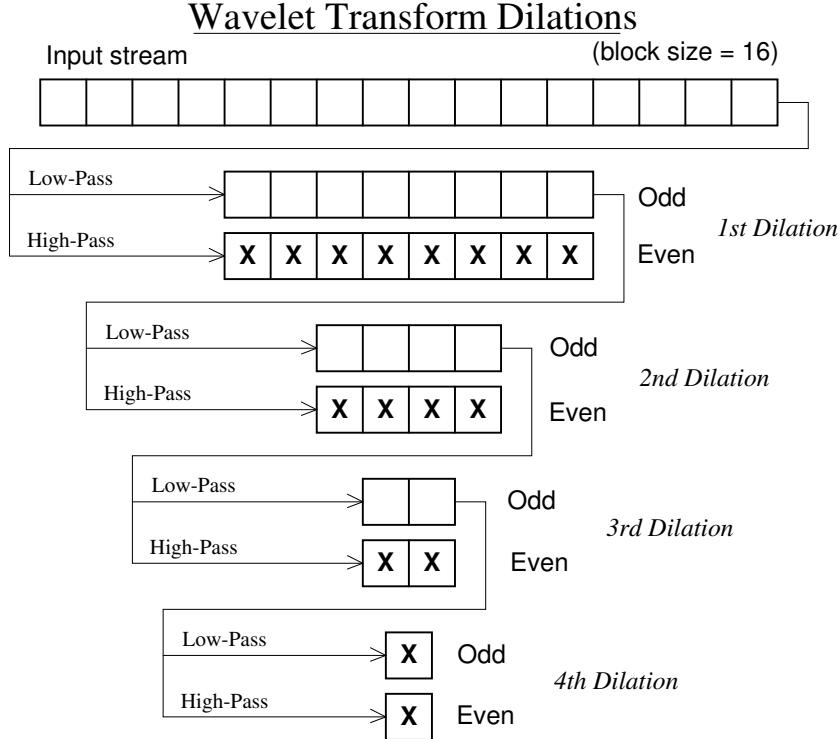


Figure 5: Dilations of a sixteen sample block of data.

it can be discarded to yield  $4 \times$  data compression. Each step of retransforming the low-pass output is called a *dilation*, and if the number of input samples is  $N = 2^D$  then a maximum of  $D$  dilations can be performed, the last dilation resulting in a single low-pass value and single high-pass value. This process is shown in Figure 5, where each ‘x’ is an actual system output. In essence, the different dilations can be thought of as representing a frequency decomposition of the input. This decomposition is on a logarithmic frequency scale as opposed to the linear scale of the Fourier transform; however, the frequency decomposition does not have the same interpretation as that resulting from Fourier techniques. The frequency dependence can be seen by looking at low-pass outputs of two different waveforms through all dilations, and their reconstruction by Haar interpolation at each stage if all high-pass filter information up to that stage is discarded. The graphs of Figures 6 and 7 were created using only low-pass and inverse low-pass filtering, but no high-pass filtering. Clearly, the low-frequency content of the first signal is maintained after many dilations, whereas the high-frequency content of the other is lost immediately. Note that this is a “frequency” decomposition of the block of fixed

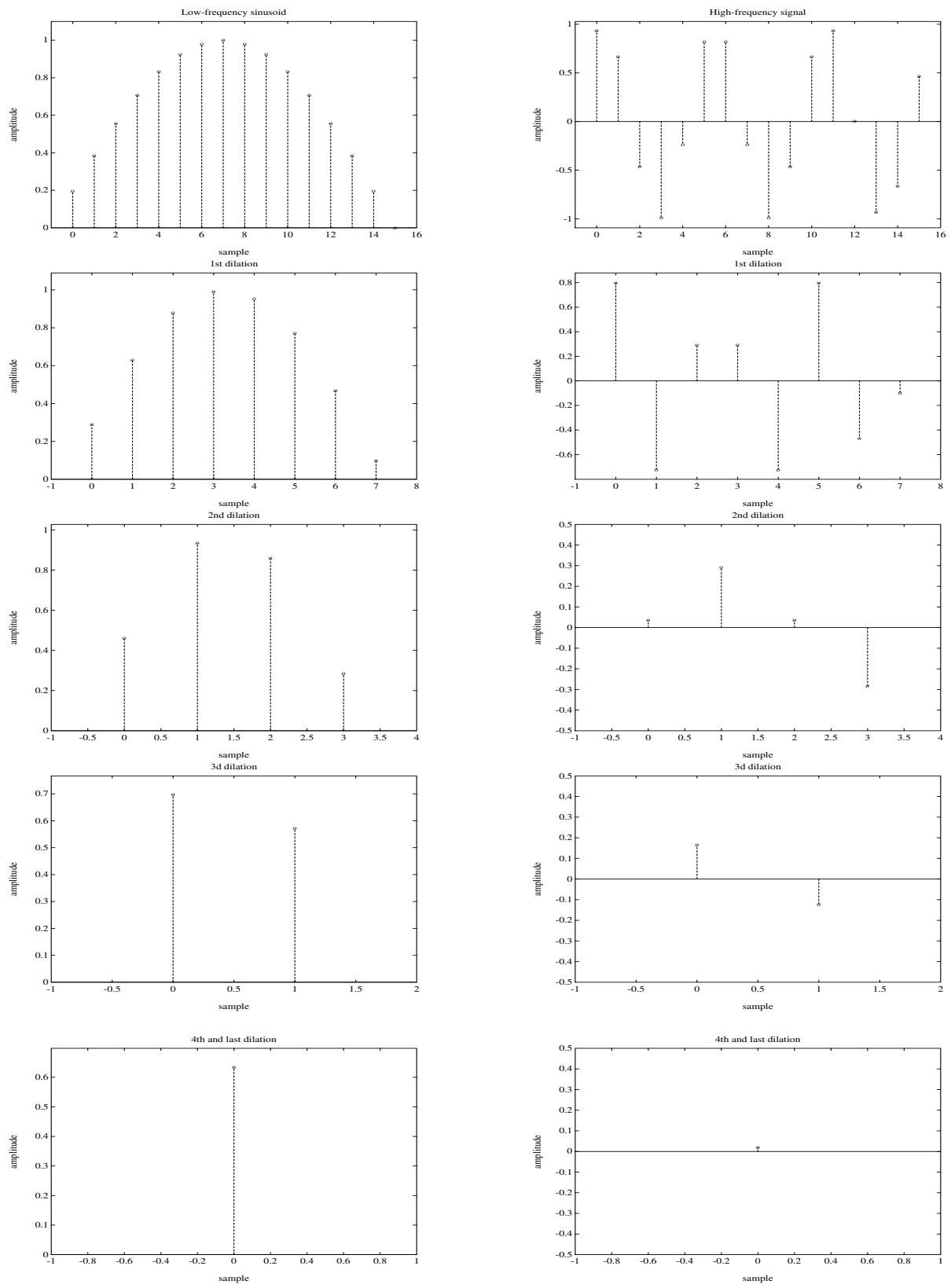


Figure 6: Decompositions of a sixteen sample block of data, using Haar wavelet coefficients.

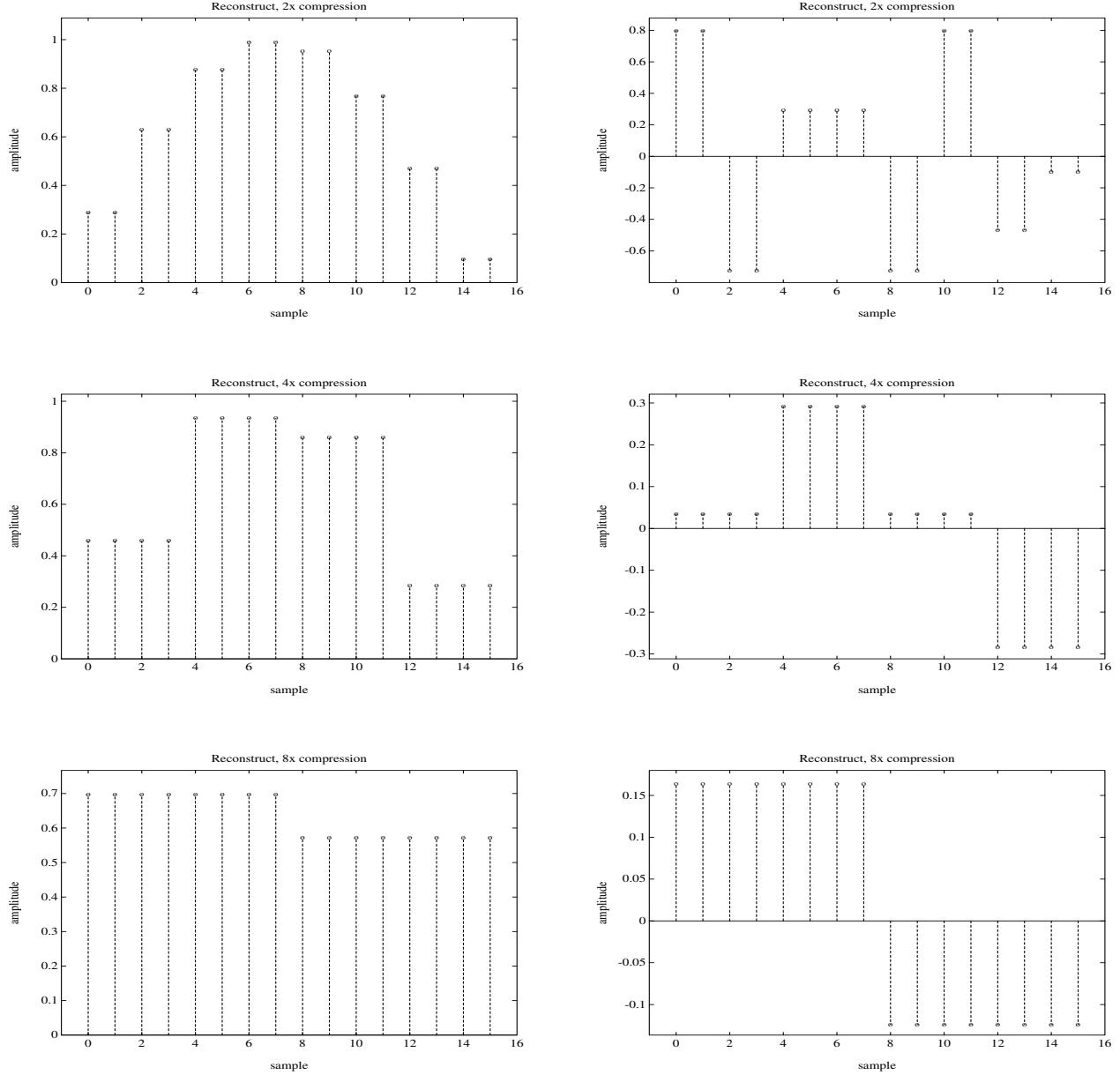


Figure 7: Reconstructions of a sixteen sample block of data, using Haar interpolation and data compression.

length  $N$ , so the lowest possible frequency which can be represented by the decomposition is clearly limited by the number of samples in the block, as opposed to the Fourier treatment in which the decomposition includes all frequencies down to zero due to its infinite support. The rule, therefore, is that successive dilations represent lower and lower frequency content by halves. It is also clear that high rates of compression may require large block sizes of the input, so that more dilations can be made, and so that lower frequencies can be represented in the decomposition.

One should note that careful handling of mean values and low frequencies is required for effective compression and reconstruction. This is particularly true for image data, where small differences in mean values between adjacent blocks can give a reconstructed image a “mottled” appearance.

### 3 Implementing The Wavelet Transformer

The design of the Wavelet Transformer was carried out in six stages, as follows:

1. Simulate the wavelet lattice filters in MATLAB to understand the basic operation of the wavelet transformation, and write *C* code to determine the proper implementation for a multi-dilation wavelet transformer. Ideally, the basic block-diagram structure of the transformer should remain the same for varying orders of wavelet functions, varying wavelet coefficients, varying input block sizes, and should also maintain the same structure between the decomposition and recomposition filters.
2. Design a simple lattice filter and confirm its function.
3. Create modules or subroutines to perform the operations of the complete wavelet transformer.
4. Create the complete transformer and confirm its function. A possible simple implementation is to have one filter to decompose an input into its wavelet representation, with its output feeding directly into another filter in order to recompose the signal back into its original form. The output can be checked against the input for various signals such as sine waves, square waves, and speech.
5. Optimize the wavelet transformer routines such that the total computation time per input sample-time is as small as possible.
6. Write software to automatically design lattice filters, given certain parameters such as wavelet coefficients, input block size, and number of dilations to decompose the input.

#### 3.1 Simulation

The first stage involved understanding the computation involved in a multi-dilation wavelet transform, and to determine the best structure for the SPROC chip, a digital signal processing chip utilizing parallel processing and pipelining for efficiency. The SPROC chip is basically a RISC processor with an instruction set geared toward DSP applications. MATLAB and *C* were chosen as simulation environments.

Although it seemed fairly certain that the final version of the wavelet transformer would be a lattice filter, matrix methods (as found in Strang [8]) were studied in order to gain a basic