DESIGN OF A BRAIN COMPUTER INTERFACE (BCI) SYSTEM BASED ON ELECTROENCEPHALOGRAM (EEG)

by

Ozan GÜNAYDIN

B.S. in Electrical And Electronics Engineering, Sakarya University, 2002

Submitted to the Institute of Biomedical Engineering
in partial fulfillment of the requirements
for the degree of
Master of Science
in
Biomedical Engineering

Boğaziçi University
2010

DESIGN OF A BRAIN COMPUTER INTERFACE (BCI) SYSTEM BASED ON ELECTROENCEPHALOGRAM (EEG)

APPROVED BY:

Prof. Dr. Mehmed Özkan ....
(Thesis Advisor)

Prof. Dr. Ahmet Ademoğlu ....

Prof. Dr. Işıl Bozma ....

DATE OF APPROVAL: 26 April 2010

ACKNOWLEDGMENTS

I would like to gratefully acknowledge the Institute of Biomedical Engineering of Boğaziçi University where this research was carried out.

I am deeply grateful to my thesis advisor, Prof.Dr. Mehmed Özkan, for his academic support, supervising and friendly tolerance during this study. His encouragement led me to finalize an immense research within a limited amount of time.

I would like to thank Prof.Dr. Ahmet Ademoğlu and Prof.Dr. Işıl Bozma for their valuable comments and participation in my thesis committee.

I would also thank Arzu Yılmaz and Barış Tanyeri from Empa Electronics and Robert Owen from Texas Instruments for providing the ICs used in this study.

Many people helped and supported me during this study and I would like to call their names here. My dear sister, Melike Günaydın, helped me to build active electrodes with great patience. My kindly friend, Oğuzcan Çığ, spent a lot of effort to scan a book for me from a far away country. Murat Üngör, a brave friend of mine, helped me to test my measurement circuit with the advantage his hair gives to him. And I am grateful to my friends İlker Böcek, Özlem Yılmaz and Çiğdem Yıldız for providing moral and motivation to complete the thesis.

This thesis is dedicated to my mother and father for their never endy love and support throughout my all life. This thesis is theirs as well as mine.

ABSTRACT

DESIGN OF A BRAIN COMPUTER INTERFACE (BCI) SYSTEM BASED ON ELECTROENCEPHALOGRAM (EEG)

A Brain Computer Interface (BCI), sometimes called a Brain Machine Interface (BMI) is a communication device between the brain and an external device, usually a computer. The main purpose of BCI systems is repairing or assisting human motor-sensory functions by asking the brain to control synthetic devices, computer cursors or robot arms. In order to extract information from the brain, physical source of information must be selected first. Electroencephalography (EEG), Magnetoencephalography (MEG) and Functional Magnetic Resonance Imaging (fMRI) could be the sources of information.

In this thesis, both acquisition hardware and software of a two channel EEG based brain computer interface was designed. EEG based BCI systems are usually implemented by analysis and classification of specific features or patterns in the spontaneous or event related EEG activity. After investigation of the components in EEG, motor imagery related mu and beta rhythms were selected for the information sources of the system.

In order to discriminate left and right hand movement imagery, three different feature extraction methods were developed using: Discrete wavelet transform, power spectrum transform and band pass FIR filters for Mu and Beta rhythms. These features were used as inputs to a two layer feed forward back propagation neural network for classification. Designed system was trained and simulated with the data provided in BCI Competition II. With the direction of the results, a low power system with the TI MSP430 microcontroller using FIR filters and a neural network was implemented.

Keywords: Brain Computer Interface,Motor Imagery,EEG Feature Classification,BCI.

ÖZET

ELEKTROENSEFALOGRAM (EEG) TABANLI BİR BEYİN BİLGİSAYAR ARAYÜZ SİSTEMİ TASARIMI

Bazen Beyin Makina Arayüzü (BMI) olarak da anılan bir Beyin Bilgisayar Arayüzü (BCI), beyin ile genelde bilgisayar olmak üzere harici bir cihaz arasındaki bir haberleşme aracıdır. BCI sistemlerinin amacı; Beynin sentetik cihazları, imleçleri ya da robot kollarını kontrol etmesiyle insanların motor-algılama fonksiyonlarını onarmak veya desteklemektir. Beyinden bu sekilde bilgi çıkarabilmek için öncelikle bilginin fiziksel kaynağı seçilmelidir. Bu uygulamanın potansiyel bilgi kaynakları Elektroensefalogram (EEG), Magnetoensefalogram (MEG) ve Fonksiyonel Manyetik Rezonans Görüntüleme (fMRI) olabilir.

Bu tezde, iki kanallı EEG tabanlı bir beyin bilgisayar arayüzünün hem enstrümantasyon donanımı hem de yazılımı tasarlanmıstır. EEG tabanlı BCI sistemleri genelde olay ilişkili ya da spontane EEG aktivitesi içerisindeki belirli örüntü ya da özelliklerin analizi ve sınıflandırılması ile gerçeklenir. EEG içerisindeki komponentlerin incelenmesi sonucu tasarlanan sistemin bilgi kaynağı olarak hareket hayaline bağlı olan mu ve beta ritimleri seçilmiştir.

Sol ve Sağ el hareket ettirme hayalini ayırabilmek için bu metodlar kullanılarak üç ayrı özellik çıkarma yöntemi geliştirildi: Ayrık dalgacık dönüşümü, Güç spektrumu dönüşümü ve Mu ve Beta ritimleri için bant geçiren FIR filtre. Bu özellikler sınıflandırma amacıyla iki katmanlı bir geriyayılım yapay sinir ağına girdi olarak kullanılmışlardır. Geliştirilen sistem 2. BCI yarışmasına ait veriler ile eğitilmiş ve simüle edilmiştir. Sonuçların ışında TI MSP430 mikrokontrolörü ile FIR filtreler ve yapay sinir ağı kullanılarak düşük güçlü bir sistem gerçeklenmiştir.

Anahtar Sözcükler: Beyin Bilgisayar Arayüzü,Hareket Hayali,EEG sınıflandırması.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

C	Capacitor
f	Frequency
$g_{[n]}$	Low Pass Filter
$h_{[n]}$	High Pass Filter
Hz	Hertz
R	Resistor
s	Second
t	Time
T-F	Time Frequency
$x_{(t)}$	Signal In Time Domain
$X_{(f)}$	Signal In Frequency Domain
W	Weight Vector of An Artificial Neural Network
V	Voltage
β	Beta Rhythm
μ	Mu Rhythm
α	Alpha Rhythm
θ	Theta Rhythm
δ	Delta Rhythm
μV	Micro Volt
μs	Micro Second
Ω	Ohm
$\Psi_{(t)}$	Mother Wavelet

LIST OF ABBREVIATIONS

A3	Discrete Wavelet Transform Level 3 Approximation
AC	Alternating Current
ADC	Analog to Digital Converter
ALS	Amyotrophic Lateral Sclerosis
ANN	Artificial Neural Network
AR	Auto Regressive
BBCI	Berlin Brain Computer Interface
BCI	Brain Computer Interface
BMI	Brain Machine Interface
CMOS	Complementary Metal Oxide Semiconductor
CMRR	Common Mode Rejection Ratio
CWT	Continuous Wavelet Transform
D1	Discrete Wavelet Transform Level 1 Detail
D2	Discrete Wavelet Transform Level 2 Detail
D3	Discrete Wavelet Transform Level 2 Detail
db10	Daubechies 10th Order Wavelet
DC	Direct Current
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram
ECoG	Electrocorticography
EEG	Electroencephalogram
ERD	Event Related Desynchronization
ERP	Event Related Potentials
ERS	Event Related Synchronization
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
fMRI	Functional Magnetic Resonance Imaging

IC	Integrated Circuit
IIR	Infinite Impulse Response
INA	Instrumentation Amplifier
LCD	Liquid Crystal Display
MI	Movement Imagery
MACS	Multiply And Accumulate
MLP	Multilayer Perceptron
MSP	Mixed Signal Processor
OPA	Operational Amplifier
RFI	Radio Frequency Interference
SMR	SensoriMotor Rhythm
SNR	Signal To Noise Ratio
TFR	Time Frequency Representation
TLU	Threshold Logical Unit
TI	Texas Instruments
WPT	Wavelet Packet Transform
WTEC	World Technology Evaluation Center
VLSI	Very Large Scale Integration
VR	Virtual Reality

1. INTRODUCTION

1.1 General Background

Electroencephalography (EEG) is the recording of electrical activity along the scalp produced by the firing of neurons within the brain [1]. Detection of patterns in spontaneous EEG is a complicated task. EEG signals are detected from the scalp and contain noise as a result of electrical interference and movement of electrodes. Signal can also be corrupted by eye blinks and other muscular activities [2].

The investigation of the brain's electrical activity started in 19th century and first successful work by Caton was accomplished on animal subjects [3]. With continious work within the field, Berger was first who could be able to detect the electrical activities of human brain above the scalp [4].

With the advances in monitoring techniques, new application areas has emerged. One of these areas has been the detection and interpretation of brain's activities. These interpretations introduced some attempts to establish communication between computers and brain signals. From that point of view, a brain computer interface can be defined as a kind of communication system. This communication system might provide a new non-muscular channel for sending messages and commands to the external world [5]. EEG is the most common source used in BCI systems because of its low cost, non-invasive nature, easy implementation and good time resolution.

According to the used EEG components, Brain Computer Interfaces can be classified as visual evoked potentials [6], slow cortical potentials [7], μ and β rhythms [8], cortical neurons [9] and P300 potentials [10].

EEG based BCI systems are usually implemented by signal acquisition, filtering, analysis and classification of specific features or patterns in the spontaneous or event related EEG activity. In this study, after the investigation of the components in EEG, motor imagery related mu and beta rhythms were selected for the information sources of the system. After the selection of information source; EEG hardware instrumentation techniques, feature extraction and classification methods were studied and the method of implementation was determined. Finally, the designed classification method was tested and discussed with the data provided in BCI Competition II [11, 12]. With the direction of the results a low power two channel brain computer interface was implemented with TI MSP430 Microcontroller.

In order to extract different frequency bands in EEG activity, spectral analysis must be employed first. Conventionally, Fourier transform has been used for spectral analysis of EEG signal. Since it does not have any time information, Fourier transform is not suitable for non stationary signal analysis like EEG if time information is also needed with frequency information. Both Fourier and Wavelet transforms were evaluated as feature extraction methods in this study. With direction of the results a low power system was implemented with FIR filters as feature extractors to meet the requirements with limited processing power and memory. Details of discrete wavelet decomposition extracts the features in the mu and beta frequency bands which includes information related to motor imagery tasks. These features were used as inputs to a two layer feed forward backpropagation neural network for classification of motor imagery tasks like right or left hand movement imagery. These mu and beta frequency features were obtained using FIR band pass filters on the MSP430 microcontroller later.

1.2 Objective

The main objective of this thesis is to develop and implement a low power EEG based brain computer interface consisting of acquisition hardware, feature extraction and classification of patterns of motor imagery tasks into one of two classes: Right hand or left hand movement imagery. For this purpose, first an electronic circuit was implemented in the lab that detects, amplifies, filters and digitizes brainwave signals. After that, two different feature extraction methods were implemented in Matlab applying db10 level 4 discrete Wavelet transform and FFT transform to the EEG trials. Two different results were obtained; One uses statistics of wavelet coefficients in the mu and beta frequency bands of this decomposition and other frequency spectrum variables as features. These features were then applied as inputs to a two layer backpropagation feedforward neural network for classification of mental tasks. With the direction of these results a low power BCI system was implemented which uses FIR band pass filters as feature extractors instead of Wavelet Decomposition to meet resource requirements.

1.3 Outline of the Thesis

The remaining chapters are organized as follows. Chapter 2 introduces the electrophysiology of the brain briefly; EEG, mu rhythm and details of brain computer interfaces are discussed here. Feature extraction and classification methods; Wavelet transform and Artificial Neural networks are discussed in Chapter 3.

Chapter 4 explains the feature extraction and classification methodology used in this study. It covers details of motor imagery trials, wavelet decomposition, artificial neural network implementation and the designed method for the low power application. Data acquisition and processing hardware design is presented in Chapter 5. Finally, Chapter 6 includes conclusions on the results.

2. EEG AND BRAIN COMPUTER INTERFACES

2.1 EEG

Electroencephalogram (EEG) is defined as the electrical potentials measured on scalp and it reflects the brain activity. This non invasive acquired signal has helped in understanding the brain functions and is used widely in diagnosis and during therapy. Recordings are made at sites distant from the source of the electrical activity. EEG, like ECG, is based on the theory of volume conduction, which describes the flow of the ionic current generated by the nerve cells through the extracellular space. Whether obtained from scalp, cortex or depths of the brain, the recorded potentials represent the summed ionic currents of the many thousands of neurons located under the recording electrode. The net ionic current is recorded as voltage across the resistance of the extracellular space [13].

Many times a stimulus is used to trigger amplitude changes in the EEG which are called Event Related Potentials (ERP). The human EEG can be decomposed into 5 different bands: The delta ($\delta :$ 0-4Hz), theta (θ :4-8Hz), alpha ($\alpha :$ 8-13Hz), beta ($\beta :$ 14-30Hz) and gamma ($\gamma >$30Hz) bands. When amplitude decrease of rhythmic activity is short lasting it is called “Event Related Desynchronization” (ERD). Inversely, an increase in rhythmic EEG activity is called “Event Related Synchronization” (ERS). It is assumed that the ERD and ERS reflect the activation of the underlying neural circuit on the measurement space. When the neural circuit is activated then the synchrony between neurons is decreased where it is reflected as ERD. In the opposite case where the neural circuit is deactivated the neurons start to have a coherent activity, which in turn induces ERS [14].

Areas of Brain Cerebral Cortex are shown in Figure 2.1 [15].

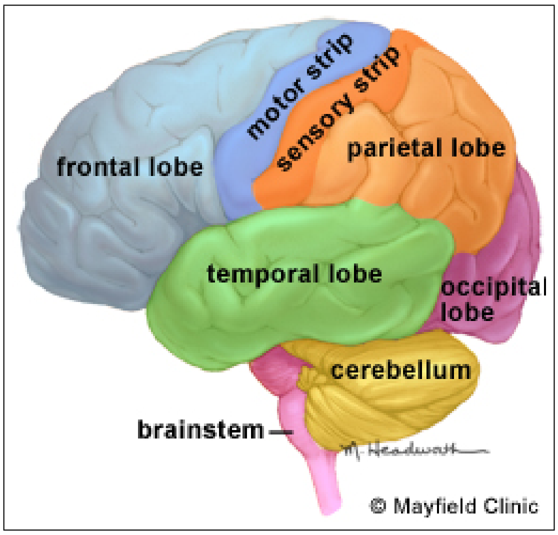


Figure 2.1 Cerebral Cortex Lobes [15].

2.2 Mu Rhythm

The mu rhythm is traditionally defined as an 8-12 Hz rhythm (in α range) recorded over sensorimotor cortex that decreases, or desynchronizes, with movement [16, 17, 18]. Chatrian noted that it also decreased during motor imagery [19]. While the mu rhythm was initially thought to occur in only a minority of individuals [19, 20], computer-based signal processing (e.g., spectral analysis) reveals that it occurs in most normal adults [18, 21, 22]. Furthermore, it is now clear that the mu rhythm is not a single EEG component, but rather a class of rhythms differing from each other in topography, frequency, and/or precise relationship to movement [23, 24].

EEG patters for different waveforms are shown in Figure 2.2.

2.3 Brain Computer Inferfaces

A brain computer interface can be defined as a kind of communication system. This communication system might provide a new non-muscular channel for sending messages and commands to the external world [5].

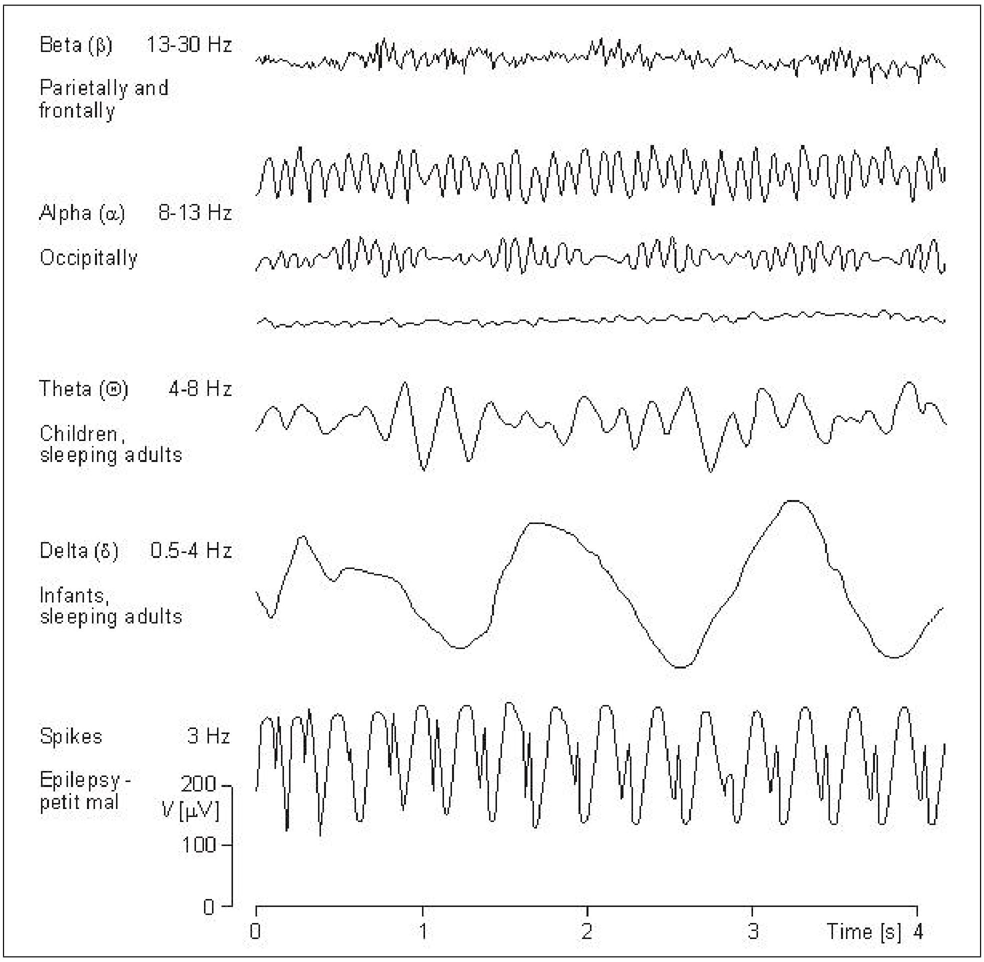


Figure 2.2 EEG patters in different frequency bands. Mu rhythm is in the Alpha frequency band.

Basic building blocks of brain computer interface systems can be defined as:

1. Signal Acquisition
2. Signal Processing : Feature Extraction
3. Signal Processing : Pattern Classification
4. Output Application and Feedback

Block diagram of a typical BCI system is shown in Figure 2.3.

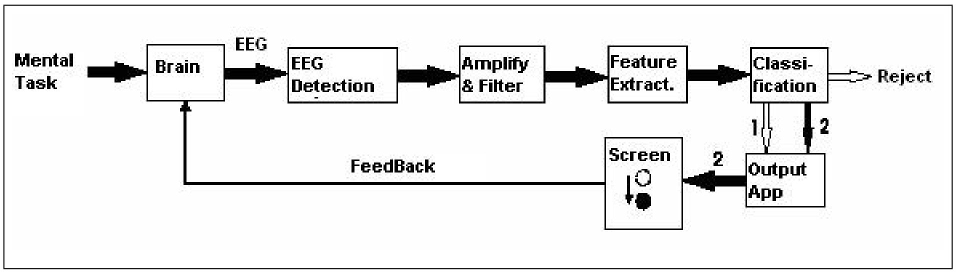


Figure 2.3 A Typical Brain Computer Interface System. Cursor Control, BioFeedback or Control of a Robot arm are examples of Output Applications.

2.3.1 Signal Acquisition

Electrical activity of the brain is measured by electrodes over the scalp. Electrodes establish connections between the scalp and EEG recording device by converting ionic current into electrical current. Electrolytic gel is applied between the scalp and the electrodes to prevent attenuation of the signal. An electrode placement system accepted as international standard called 10-20 System is used to be able to compare the measurements taken [25]. Every electrode position has a letter and a second letter or number to define the hemisphere location in this system. Odd numbers indicate left hemisphere, even ones indicate right and 'Z' letter indicates placement on the center line. The letters F, T, C, P and O stand for Frontal, Temporal, Central, Parietal and Occipital, respectively. The "10" and "20" refer to the fact that the actual distances between adjacent electrodes are either %10 or %20 of the total frontback or right-left distance of the skull. This placement system is shown in Figure 2.4 [26].

EEG measurement can be done in one of two ways:

1. Bipolar Recording: Each channel (i.e., waveform) represents the difference between two adjacent electrodes. The entire montage consists of a series of these channels. For example, the channel "Fp1-F3" represents the difference in voltage between the Fp1 electrode and the F3 electrode. The electrodes are connected in series to an equal number of amplifiers. For example, amplifier 1 measures the difference between electrodes A and B, amplifier 2 measures the difference

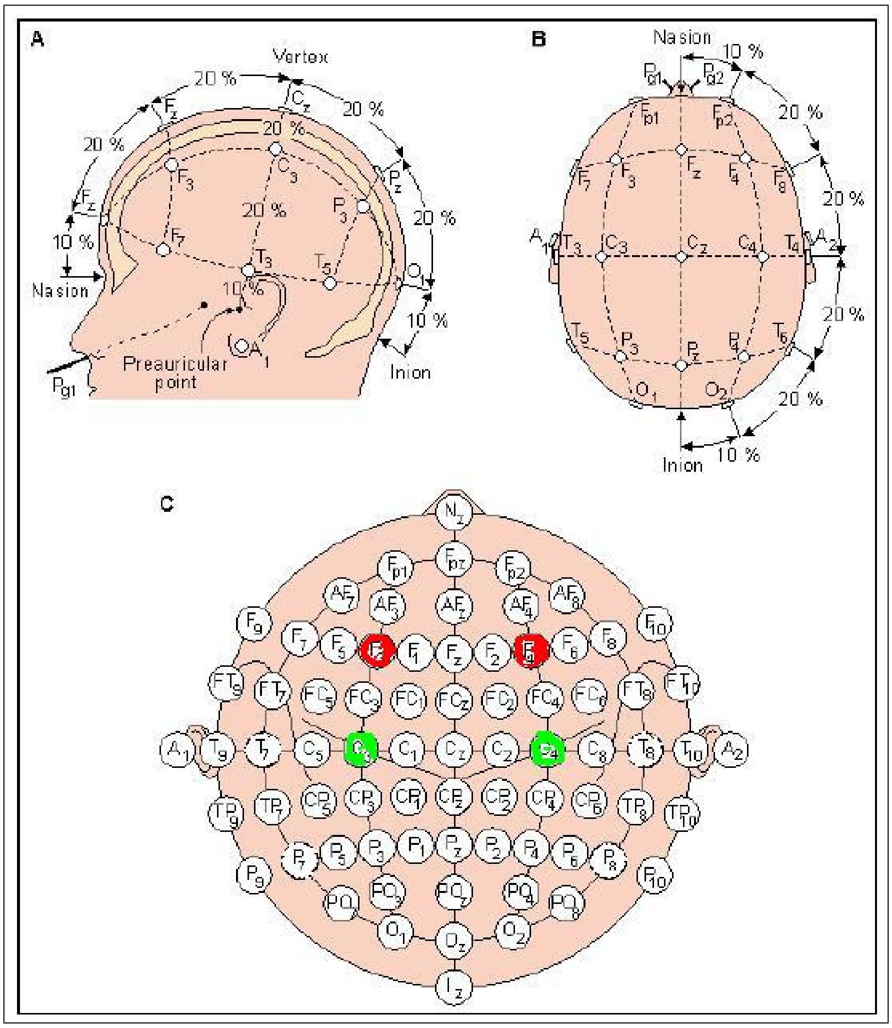


Figure 2.4 10-20 System for EEG Electrode Placement. C3-F3 and C4-F4 pairs are used as two channels in this study [26].

between B and C, and so on. Bipolar recording is used in this study as shown in Figure 2.4.

2. Referential Recording: Each channel represents the difference between a certain electrode and a designated reference electrode. There is no standard position at which this reference is always placed; It is, however, at a different position than the "recording" electrodes. Midline positions are often used because they do not amplify the signal in one hemisphere vs. the other. Another popular reference is "linked ears" which is a physical or mathematical average of electrodes attached to both earlobes or mastoids.

The electromagnetic waves generated by the brain are very small in magnitude (μV range) at the scalp. This requires that a brainwave with a V_{p-p} amplitude of 100 μV be amplified on the order of 80 dB before a useful signal can be seen. The signal must also be filtered of any unwanted frequencies. EEGs commonly use high pass, low pass and notch filters to remove noise generated from muscle movement and other physiological factors, as well as the common electricity lines at 50 Hz [27]. The amplifiers have to provide high common-mode rejection ratio, high gains with very small power. After amplification and filtering of each channel used, the signal is converted to a digital representation for signal processing and storage.

2.3.2 Signal Processing: Feature Extraction and Classification

In nature the signals which we see such as images and hear such as sound and voice as well as EEG, have high dimensionality and contain temporal information. Our brain accomplishes structural information processing to extract characteristic properties of the observed system and maps them to pre-known categories where these steps correspond to feature extraction and classification. Therefore Pattern Recognition can be seen as a combination of these two important steps [28].

Because of these high dimensions of original signals, instead of concentrating on the details of the original space the signal is reduced to a small subset that represents the vital information. This small subset is called the feature set or feature vectors. After digitizing the signal, one or more EEG control channels are derived from a linear combination of a selected set of the ear-referenced or bipolar channels provided by the amplifier. This process is known as Spatial Filtering. Most commonly, each of the EEG control channel derived is EEG activity at a location over sensorimotor cortex.

Spectral analysis is the conventional way of feature extraction which includes the frequency decomposition of the original signal. The voltages at these decomposed frequencies come from signal extraction become the independent variables in the feature space. In the case of the important features appearing as a transient phenomenon, one will face the inability to capture local information with any of the previous methods like Fourier. In order to overcome this problem, it is a good choice to analyze the signal in time-frequency plane (T-F) which have time localization of frequency components like Wavelet Transform.

After the feature extraction phase, a classification must be done based on these features to discriminate between mental tasks. In brain computer interfaces, learning for classification comes in one of two ways or combination of them:

1. Subject Learning : Subjects learn to control their brain activity in a predetermined fashion that can be robustly detected and converted into a computer command. They require subject training through biofeedback and they display a low bandwidth for effective communication. In that approach a formulation is generated to map the weights of the features channels into control commands.
2. Machine Learning : Computer algorithms are needed to train the system for signal classification. Artificial Neural networks are an example of classification based of feature channels. Output classes are the representations of the corresponding mental tasks in the system. That approach requires minimum subject training but neural network training.

In this study, machine learning approach is used, since it does not require extensive subject training. A feed forward artificial neural network will be used for classification of patterns. Network will be trained with the features extracted from the motor imagery trials taken from subjects. A series of methods will be applied for local feature extraction, dimension reduction and classification to be able to use the oscillatory activities of a motor imagery EEG activity for a BCI task.

2.3.3 Output Application And Feedback

Output application of the system is usually a control task such as computer cursor control, biofeedback or robot arm control. Biofeedback is an important factor for subject training. Output of the low power system in this study is classification result displayed on a segmented LCD.

2.3.4 BCI Literature Survey

This part provides an insight into a representative variety of BCI systems that are currently being pursued in research labs. A distinctive feature in BCI studies is the paradigm used for the interaction between user and computer. On one hand there are systems that require an active and voluntary strategy for generating a specific regulation of an EEG parameter such as the motor-related mu-rhythm or the self-regulation of slow cortical potentials (SCP). On the other hand there are passive paradigms, where participants only have to passively view an item for selection. Those systems detect the evoked responses such as P300 as presented in subsection 2.3.4.1 or make use of steady-state evoked potentials (SSVEP) as presented in subsection 2.3.4.3.

Finally, one distinction between BCI labs is based on the realization of the system. Most groups, as introduced in subsections 2.3.4.1,2 and 6, use extensive subject training. So, users have to adapt their brain signals to a fixed decoding algorithm, that is, the learning is on the subject side. Over the past five years, the Berlin group has established a paradigm change, where learning is now done by the computer, following the motto "let the machines learn". Now several groups have adopted this principle. Examples for this approach are discussed in subsections 2.3.4.3,4 and 5. Note that even if a pure machine learning approach was intended, the subject will inevitably learn once feedback has started, so in principle BCI systems will always have both aspects: subject and machine training[29]. But this may be omitted for only two class control applications as in this study which only uses machine learning.

In this section six major BCI labs are introduced. Subsection 1 outlines the Albany BCI, where a user is trained to manipulate his mu and B rhythms to control a cursor in 1- or 2D. Furthermore, BCI control based on the P300 paradigm is shown. Similar to Albany, the Tübingen BCI, outlined in subsection 2, train their subjects to adapt to the system using slow cortical potentials. The group uses BCI as a means for communication of ALS patients with the outside world and as the design of this interaction. Further BCI systems discussed in the section are P300 and mu rhythm based BCIs, and interesting new BCI paradigm based on auditory stimulation and the use of invasive techniques like ECoG for BCI.

In subsection 3 the main research directions of the Graz BCI are depicted. This is the lab which provided the EEG measurements of BCI Competition II used in this study. The group is broadly exploring the whole BCI field from sensors, feedback strategies, and cognitive aspects to novel signal processing methods, with excellent results. The Graz BCI is shown to be not only of use for patients but also it contributes to general man-machine interaction as demonstrated for a moving in a VR environment. Typically, only a few electrodes and machine learning techniques combined with user adaptation are employed to achieve BCI control[29].

Subsection 4 introduces the Berlin BCI. Compared to training times of weeks or even months in other BCIs, the BBCI allows for subject control after 30 minutes. This drastic decrease in training time became possible by virtue of advanced machine learning and signal processing technology. The subsection presents online feedback studies based on the physiological signals' preparatory potential and mu rhythm modulation. The study shows that after less than one hour, five of six untrained subjects were able to achieve high performances when operating a variety of different feedbacks[29].

Similar to Berlin approach, the Martigny BCI introduced in subsection 5 tries to relocate the effort from the subject training to the machine by using machine learning techniques and online adaptation to realize a BCI. In particular, online adaptation is an important direction to compensate for the intrinsic nonstationarities found in EEG signals[29].

Finally, the ideas of the Vancouver BCI are introduced in subsection 6. The main focus here is to establish an asynchronous BCI for patients, that is, a system that detects whether a user is intending something or not. To achieve this goal, the authors also use machine learning techniques that adapt the machine to the user.

2.3.4.1 Noninvasive BCI Research at the Wadsworth Center. The primary goal of the Wadsworth Center brain-computer interface program is to develop electroencephalographic (EEG) BCI systems that can provide severely disabled individuals with an alternative means of communication and/or control. They have shown that people with or without motor disabilities can learn to control sensorimotor rhythms recorded from the scalp to move a computer cursor in one or two dimensions and have also used the P300 event related potential as a control signal to make discrete selections[29]. They are now evaluating the practicality and effectiveness of a BCI communication system for daily use by such individuals in their homes. With Sensorimotor Rhythm Based Cursor Control approach users learn during a series of training sessions to use SMR rhythm amplitudes in mu and/or B frequency bands over left and/or right cortex to move a cursor on a video screen in one or two dimensions. This is not a normal function of this brain signal, but rather the result of training. One example of SMR application at Wadsworth can be seen in Figure 2.5 [30, 31].

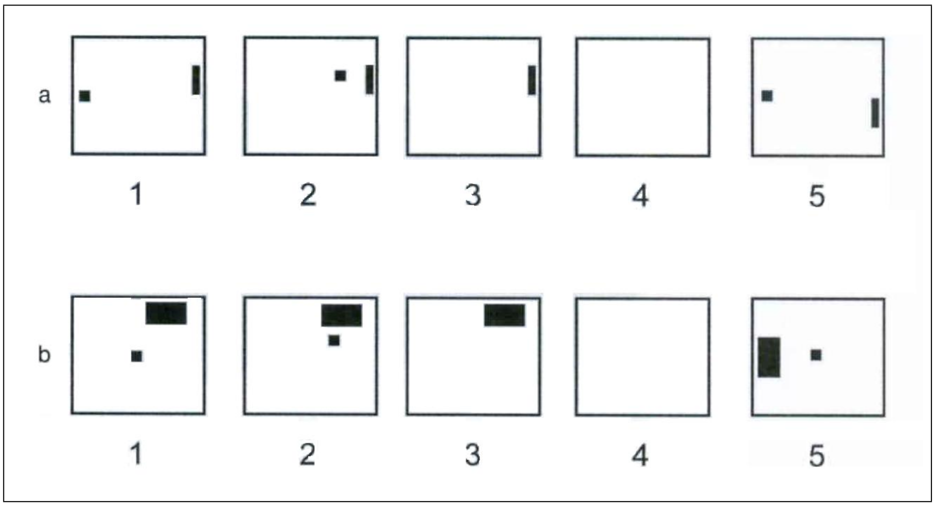


Figure 2.5 a) One Dimensional four target SMR control task [30]. B) Two dimensional eight target SMR control task [31].

In addition to refining and improving SMR and P300BCI performance , Wasdworth is also focused on developing clinically practical BCI systems. The most pressing needs for a successfull home BCI system are developing a more compact system, making the system easy to operate for a caregiver, and providing the user with effective and reliable communcation applications. Their current system includes a laptop computer, a flat panel display, an eight channel electrode cap, and an amplifier with a built in A/D board. Regression analysis is used with the SMR system and classification for the P300 system. The regression approach is well suited to the SMR cursor application since it provides continuous control in one or more dimensions and generalizes well to novel target confs.In contrast, the classification approach is well suited to the P300 where the target is treated as one class and all others are treated as the other class[29].

2.3.4.2 Tubingen BCI Research Group. An important clinical application of BCIs is to enable communication or environmental control in severely paralyzed patients. The BCI "Thought-Translation Device (TTD)" allows verbal communication through the voluntary self-regulation of brain signals (e.g. slow cortical potentials (SCPs)), which is achieved by operant feedback training. Humans' ability to self-regulate their SCPs is used to move a cursor toward a target that contains a selectable letter set. Two different approaches were followed to develop Web browsers that could be controlled with binary brain responses. Implementing more powerful classification methods including different signal parameters such as oscillatory features improved their BCI considerably. It was also tested on signals with implanted electrodes.

Most BCIs provide the user with a visual feedback interface. Visually impaired patients require an auditory feedback mode. A procedure using auditory feedback of multiple EEG parameters was evaluated. Properties of the auditory systems are reported and the results of two experiments with auditory feedback are presented. Clinical data of eight ALS patients demonstrated that all patients were able to acquire efficient brain control as one of the three available BCI systems (SCP,μ rhythm and P300), most of them used the SCP-BCI[29].

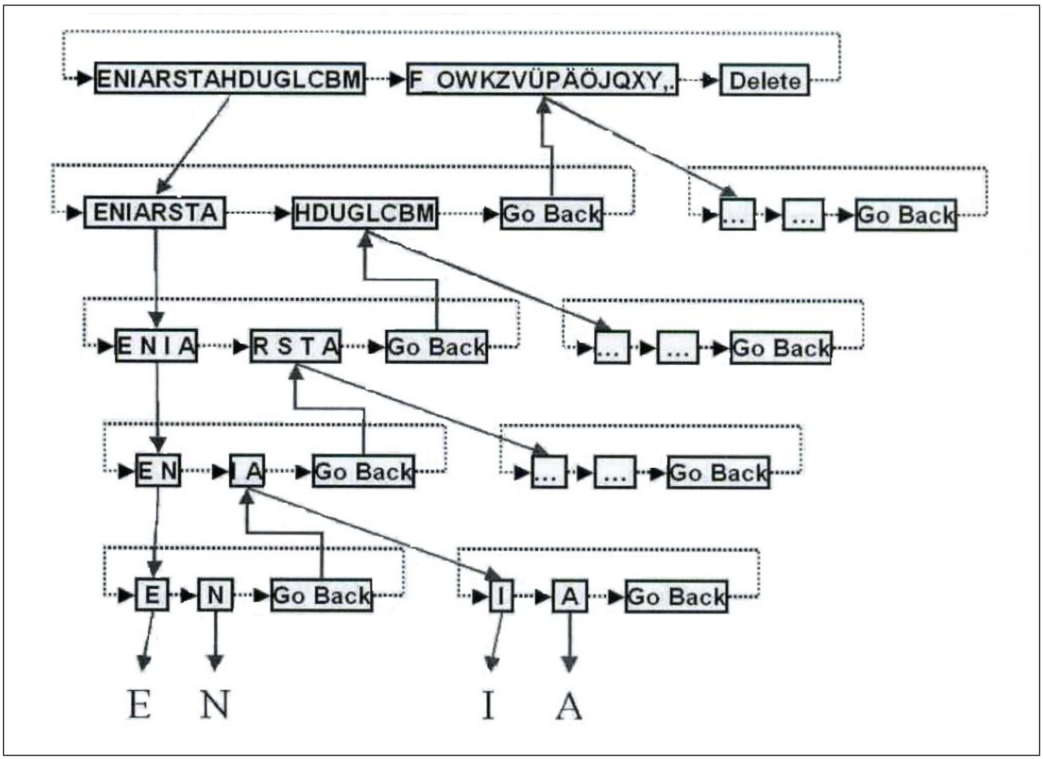


Figure 2.6 Schematic structure of the language support program of Tubingen group. Boxes show letter sets offered during one trial; solid arrows show the subsequent presentation when a select response is produced; dotted arrows show the presentation following a reject response. When the level of single letters is reached, selection leads to the presentation of this letter at the top of the screen. Texts can thus be generated by adding letter to letter. At all except the uppermost level, failure to select one of the two choices results in the presentation of a "go back" option taking the user back to the previous level. At the top level, double rejection and selection of the delete function results in the deletion of the last written letter [29].

2.3.4.3 Graz Brain Computer Interface. The Graz-BCI system uses EEG signals associated with motor imagery, such as oscillations of *beta* or *mu* rhythms or visual and somatosensory steady-state evoked potentials as input signal. Special effort is directed to the type of motor imagery, the use of complex band power features, the selection of important features, and the use of phase-coupling and adaptive autoregressive parameter estimation to improve single-trial classification. A new approach is also the use of steady-state somatosensory evoked potentials to establish a communication with the help of the tactile stimuli. In addition, different Graz-BCI applications are reported: control of neuroprostheses, control of a spelling system, and first steps toward an asynchronous (uncued) BCI for navigation in a virtual enviroment [29].

2.3.4.4 Berlin BCI:Machine Learning Based Detection. The Berlin Brain-Computer Interface (BBCI) project develops an EEG-based BCI system uses machine learning techniques to adapt to the specific brain signatures of each user. This concept allows to achieve high quality feedback already in the very first session without subject training. The first kind of experiments analyzes the predictability of performing limbs from the premovement (readiness) potentials including successful feedback experiments. The limits with respect to the spatial resolution of the somatotopy are explored by contrasting brain patterns of movements of (1) left vs. right foot, (2) index vs. little finger within one hand, and (3) finger vs. wrist vs. elbow vs. shoulder within one arm. A study of phantom movements of patients with traumatic amputations shows the potential applicability of this BCI approach. In a complementary approach, voluntary modulations of sensorimotor rhythms caused by motor imagery (left hand vs. right hand vs. foot) are translated into a proportional feedback signal. They report results of a recent feedback study with six healthy subjects with no or very little experience with BCI control: Half of the subjects achieved an information transfer rate above 35 bits per minute (bpm). Furthermore, one subject used the BBCI to operate a mental typewriter in free spelling mode. The overall spelling speed was 4.5 letters per minute including the time needed for the correction of errors. These results are encouraging for an EEG based BCI system in untrained subjects that is independent of peripheral nervous system activity and does not rely on evoked potentials [29].

2.3.4.5 The IDIAP BCI:Asynchronous Multiclass Approach. Their work is on a self-paced asynchronous BCI that responds every 0.5 seconds. A statistical Gaussian classifier tries to recognize three different mental task; it may also respond "unknown" for uncertain samples as the classifier incorporates statistical rejection criteria. They report experience with different subjects. They also describe three-brain actuated applications: a virtual keyboard, a brain game, and a mobile robot. The are focused on real-time control of brain actuated robots [29].

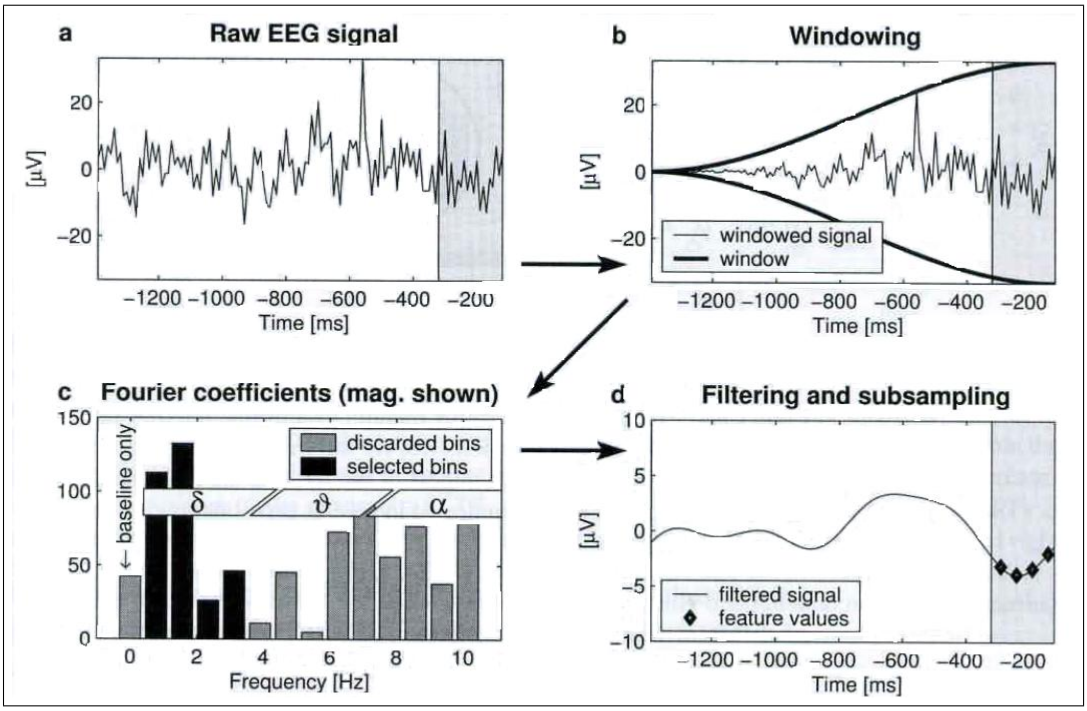


Figure 2.7 This example of BerlinBCI shows the feature calculation in one channel of a premovement trial [-1400 - 120] ms with keypress at t = 0 ms. The pass-band for the FT filtering is 0.4-3.5Hz and the subsampling rate is 20 Hz. Features are extracted only from the last 200 ms where most information on the upcoming movement is expected[29].

2.3.4.6 Canada British Columbia University BCI.

The concept of self-paced control has recently emerged from within the general field of BCI research. The use of assistive devices in real-world environments is best served by inferfaces operated in an asynchronous manner. This self-paced or asynchronous mode of device control is more natural than the more commonly studied synchronized control mode whereby the system dictates the control of the user. The Neil Squire Society develops asynchronous direct brain switches for self-paced control applications [29].

Their latest switch design operated with a mean activation rate of 73 percent and false positive error rates of 2 percent.

2.3.4.7 Commercial Translation of BCI Systems.

The extent to which industry in Europe and Japan has embraced BCI-related research goals and the development of requisite technologies for BCIs is impressive. This high degree of industry commitment was perhaps most evidenced in Germany by institutional entities having the specific missions of actively promoting academic-industrial research interactions, garnering support for BCI research from industry sources, and transitioning the resulting BCI and BCI-related systems to industry for commercialization. Such entities house advanced technologies and equipment made available to startups with limited resources; research collaborations and partnerships could result in spin offs that accelerate the entry of new BCIs and BCI technologies into the marketplace. The EU 6th Framework research programs strongly encourage and to some degree require industrial involvement. Corporations involved in commercialization of BCI systems and/or BCI-related products are essentially able to participate in EU-sponsored research (with some restrictions) as a collaborator along with any other university or institute unit and are eligible to receive funds to conduct their respective component of the overall research project. Equally impressive was the degree to which BCI-related research issues were integrated into the agendas of major Japanese research institutes and corporations and the extent of government support of those private, and sometimes profit-making, entities. In general, the panel saw creative and highly flexible academic-industry collaborations that promoted the transition from laboratory based to commercialized BCIs [32].

BCI research and BCI technologies had reached the stage of translation to industry and commercialization. In the United States, commercialization of BCIs is just beginning to occur, e.g., Cyberkinetics Figure 2.8, which combines technology from Brown University and the University of Utah, for a BCI system that allows the user to move cursors on a computer screen using 2-D kinematic information extracted from motor cortical population single-unit recordings.

In Europe, the WTEC panel found specific mechanisms for joint academic / scientific and industrial collaborations leading to the translation of BCI research, incorporation of BCI technology into small companies, and the creation of spin-offs from research efforts. For example, industrial entities can participate in EU-sponsored research as just another project that receives part of the research budget, i.e., a company can propose to partner with research members of an EU project to develop and shape

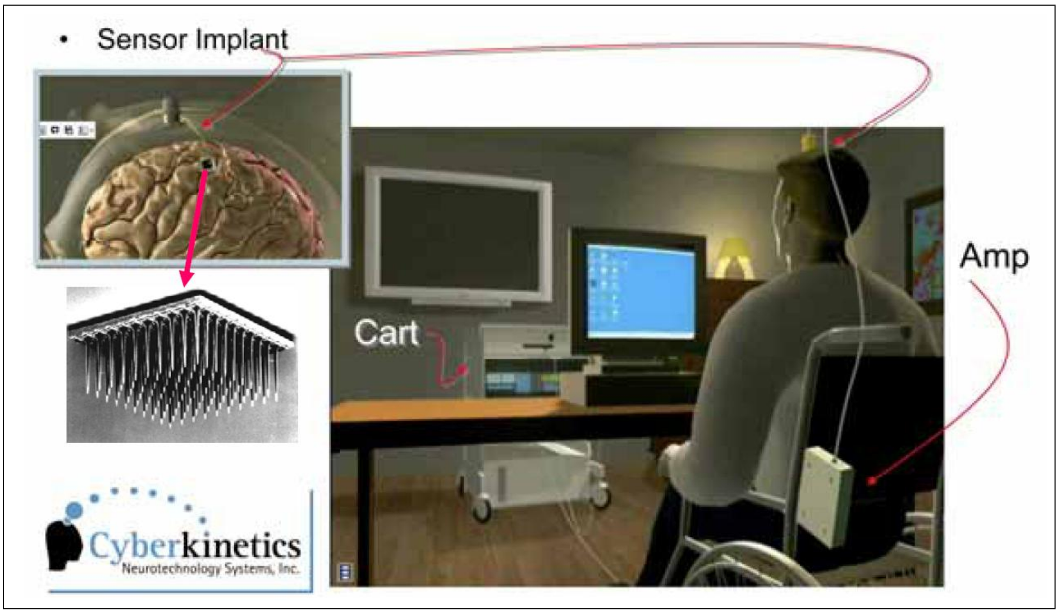


Figure 2.8 Cyberkinetics, a BCI company in the United States (courtesy John Donoghue, Brown University)[32].

a given technology to fit the research requirements of the global project. The only requirement is that each commercial entity provide 50 percent of the costs of its project. Researchers benefit when industry is an integrated member of a large project because it maximizes research needs and available technology. It also benefits the company because it essentially guarantees a customer base; often, industry-related projects are producing technologies ultimately sold to other research-related projects. Scientific progress is achieved through a closer relationship between researchers and the sources of their technology, which allows a faster evolution of next-generation technology. EU projects can require industrial involvement, so relevant businesses often are actively pursued. Example outcomes of the EU encouragement of industry participation include (1) Multi Channel Systems GmbH (MCS), a leading worldwide supplier of multi site electrodes and multichannel recording/stimulation systems for brain slices/cultures and a partner in many EU projects; and (2) g.tec, a worldwide supplier of multichannel EEG amplifiers that grew out of activities of the University of Graz BCI Laboratory and is also now a partner in many EU projects. A g.tec amplifier can be seen in Figure 2.9.

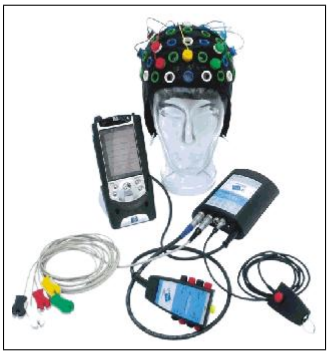


Figure 2.9 G.tec Company EEG Amplifier: For a variety of BCI technologies, g.tec is a source of one of the best head caps used in the field involving wet electrode recordings[32].

In Germany, the panel was introduced to institutional infrastructures that actively promote interactions between academia and industry. For example, the Fraunhofer Institute (Berlin) for Computer Architecture and Software Technology pursues the development of BCI research and BCI technology both for medical and commercial applications (e.g., gaming, auto industry). The Fraunhofer Institute in Berlin is one of four throughout Germany. The director of a given research group in the Fraunhofer FIRST Berlin holds an 80-percent appointment in Potsdam University (Berlin) and a 20-percent position in the Fraunhofer Institute. Support is derived from any source, but the university pathway allows funding for basic research, whereas the institute pathway provides an avenue for industrial support. At least 30 percent of the funding through the Fraunhofer Institute must be provided by industrial sources. So for example, the Intelligent Data Analysis Group (IDA) directed by Prof. Dr. Klaus- Robert Müller engages in a wide range of theoretical research in machine learning and signal processing and develops new algorithms for real-world data analysis. The group also receives funding from the automobile industry to develop pop-up displays for the driver when periods of cognitive overload or high-attention demand occura form of nonmedical BCI. It also receives support from the gaming industry to develop brain-driven video games. Through active collaborations with the Charité University of Medicine Berlin, one of the premiere medical universities in Germany, the group is able to conduct experiments for clinical applications of BCIs. Through additional fundamental work on the neurophysiological underpinnings of BCI signals, the Charité group develops new experimental paradigms to point the IDA team to new directions of analytical development. This is an exciting state- and local-sanctioned infrastructure to support the highly interdisciplinary interactions at the fundamental, clinical, and industrial levels necessary for the development of BCIs [32].

2.3.5 Summary of the Existing Work on the Subject

Summary of the used signals, techniques and methods in BCI studies are listed below.

1. Number of channels used: 2 - 128.
2. Applications: Robot Arm Control, Cursor Control, Virtual Keyboard, Neural Prosthesis, Gaming, Virtual Reality.
3. Feature Signals: EEG: Mu and Beta Rhythms, Slow Cortical Potentials (SCP), Evoked Potentials, P300 Potentials; EcOG (Electrocorticogram); Magnetoencephalography (MEG); Functional Magnetic Resonance Imaging (fMRI); Functional Near Infrared Spectroscopy (fNIRS).
4. Feature Extraction Methods: Fourier Transform, Wavelet Transform, FIR and IIR filter features, Local Field Potentials, Principal Component Analysis (PCA).
5. Classification Methods: Local Discriminant Analysis (LDA), Regression Analysis, Neural Networks, Least Square Regression.
6. Achievements: BCI applications are commercially available from companies such as Cyberkinetics and G.tec applications with keyboard control or gaming. There are many application implementations taking place in academic environments. For example, robot arm control is implemented successfully. Two dimensional cursor control with subject training is an another example from Academic work.
7. Databases: Four BCI competitions have been organized until now. Databases are available on the Web: http://www.bbci.de/competition/.

3. FEATURE EXTRACTION AND CLASSIFICATION METHODS

In the first phase of this study, two different feature extraction methods were evaluated in Matlab applying db10 level 4 discrete Wavelet transform and FFT transform for feature extraction. After getting the results, as the second phase, for implementation with a microcontroller a resource efficient method was developed which FIR band pass filters utilized instead of Wavelet transform to extract sub band information.

Through wavelet transform, the EEG signal were decomposed into the frequency sub bands using Discrete wavelet transform and a set for statistical features was extracted from the sub-bands to represent the distribution of wavelet coefficients according to the characteristics of motor imagery EEG signals. With Fourier transform approach, trials were transformed into frequency domain to extract features. Finally an Artificial neural network was utilized to classify computed features into different categories that represent the left or right hand movement imagery. Same classification method was used with extracting features by FIR band pass filters in the low power system implementation. In this chapter, theory of FIR filters, discrete wavelet transform and artificial neural networks will be discussed.

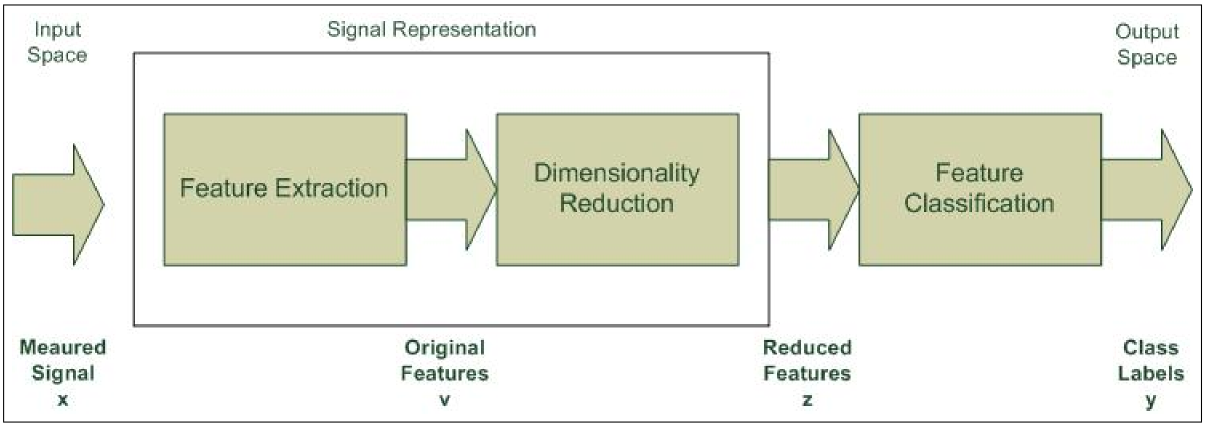


Figure 3.1 The Processing stages of Classification

3.1 Filtering Basics

A filter is a system that performs mathematical operations on a signal to reduce or enhance certain aspects of that signal. To reduce or extract some of the properties of the real world signals, calculating and classification of frequency components in that signal is common. To calculate and process that frequency based information, filters are used. Filters can be implemented as analog filters by electronic circuits or digital filters implemented in digital domain. Broad classification of filters can be seen in Figure 3.2 [33].

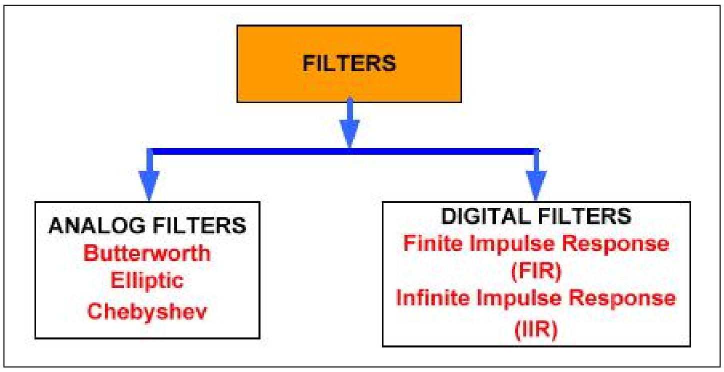


Figure 3.2 Classification Of Filters[33]

Specification and differences of analog and digital filters are summarized below[33].

Analog Filters:

1. Mature and well developed design methodologies available
2. Accuracy limited, use components that are subjected to changes over temperature
3. Small change in filter specifications leads to complete change in hardware
4. Testing and verifications are time consuming
5. Storage and portability a cause for concern
6. Inherently expensive to improve accuracy

Digital Filters:

1. Design is simple, borrows all concepts from its analog counterpart
2. Modifying characteristics requires small change in software with no hardware changes necessary
3. Interface to digital microcomputers is extremely simple
4. Extremely accurate. At least a 1000 times better accuracy than its analog counterpart
5. 6dB increase in gain with every bit of increase in resolution on fixed point machines such as MSP430
6. Inherently expensive to improve accuracy

Considering only this comparison digital filters can be seen the only choice because of their advantages and easy to use. But analog electronic filters are still needed in signal processing before analog digital conversion. The real world signals have a lot of noise components with different frequency and amplitude levels. At the same time the real world signals are usually very small in amplitude which can change from 1 μV to hundreds of mVs. Since analog signals are very small in amplitude and contain too much noise, they must be amplified and filtered before analog to digital conversion to reach a ADC detectable amplitude level and to prevent anti aliasing. A filtering stage is also needed before the first amplification stage to reject DC components and not to amplify these components. Analog filters are used to meet these requirements in signal processing.

Characteristics of filters are summarized in Figure 3.3 [33].

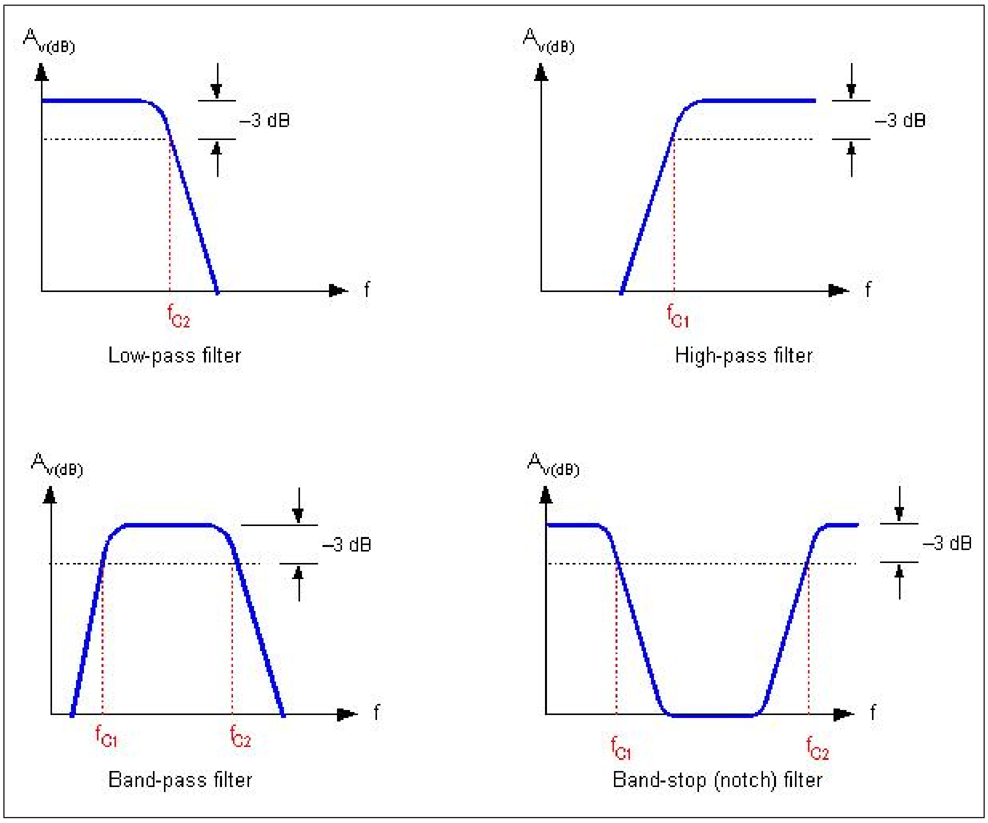


Figure 3.3 Characteristics Of Filters

3.2 Digital Filters

3.2.1 Digital Filter Basics

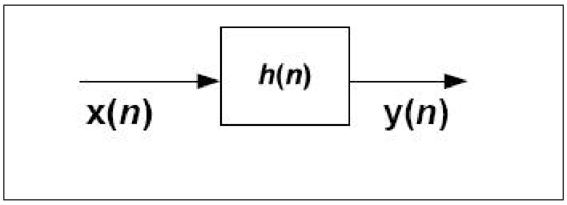


Figure 3.4 Digital Filter

1. Input sample x(n) operated by filter h=[h(0), h(1).h(M)] to give output sample y(n), every sampling instant, defined by sampling frequency

2. Mathematically a convolution of input vector x and filter vector h of order M

$$y = x \otimes h = \sum_{i=0}^{M-1} h(i) \cdot x(n-i) \tag{3.1}$$

3. Current output sample depends on present and previous samples of input and/or output samples

4. Filter h is almost always a real number, and is usually converted into the nearest integer on fixed-point numbers such as microcontrollers

3.2.2 Digital FIR Filters

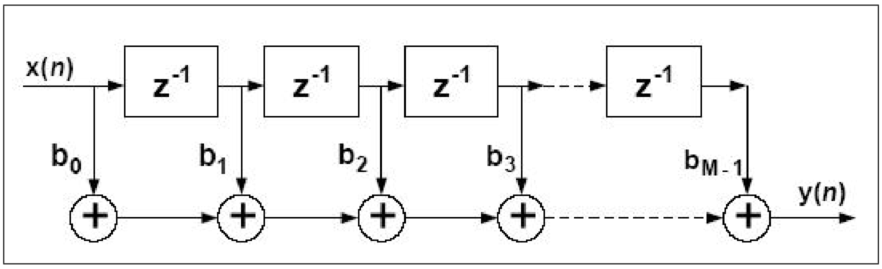


Figure 3.5 Digital FIR Filter

$$y(n) = \sum_{i=0}^{M-1} b(i) \cdot x(n-i) \tag{3.2}$$

1. Impulse Response is finite Time domain representation of the filter b has finite length and equal to the length of vector h

2. Inherently stable Output depends only on the present and previous samples of the input Output always bounded by input, if input stops, output immediately follows

3. Can exhibit linear phase across all frequencies Linear phase property induces symmetry for coefficients b, thus reducing CPU overhead Does not introduce phase distortion

Digital finite impulse response (FIR) filters form the basis for numerous digital signal processing applications. The basic operation needed to implement a FIR filter is the signed multiply-and-accumulate (MACS), which is traditionally performed using a hardware multiplier peripheral in any DSP device. Some of the MSP430 devices have an integrated hardware multiplier that can perform this MACS operation allowing these devices to run the FIR filter algorithm more efficiently than devices without a built-in hardware multiplier.

In addition to the MACS operation the processor handles the task of moving the digital samples and filter coefficients from memory to the MAC hardware, retrieving the results and storing them into memory. In a real-time digital filter algorithm, the computation and memory-move operations have to be completed within one sample period. The number of computations to be performed within one sample period depends on the number of taps of the filter, i.e, the order of the filter. The order of the filter is determined by the required filter performance characteristics. When higher order filters are combined with faster sampling rates, the demand on the processor becomes very high. This limits typical MCUs to handle a real-time FIR filter algorithm only at low sample rates and with a reduced number of filter taps [34]. Because of these limitations sampling rate was limited at 128 Hz in the MCU implementation in parallel with 3.5 seconds recording time to stay in the limits of the on chip memory in this study.

3.3 Discrete Wavelet Transform Analysis

The earliest form of function representation using orthogonal basis functions is undoubtedly the Fourier series for continuous and periodic signals:

$$x(t) = \sum_{-\infty}^{\infty} c_k e^{jk(2\pi/T)t} \tag{3.3}$$

$$c_k = \frac{1}{T}\int_T x(t)e^{-jk(2\pi/T)t} \tag{3.4}$$

where x(t) is the signal to be analyzed, T is the period of the signal, and c_k 's are the Fourier coefficients, representing the spectral components of x(t). The complex exponential functions at different discrete frequencies of $2\pi jk/T$ are not compactly supported in time since they extend to infinity. As noted above, this makes the Fourier representation inadequate in analyzing non stationary signals. Fourier representation can not provide any information regarding the time localization of these spectral components. This is not a problem for analyzing stationary signals, since all spectral components exist at all times. But for non stationary signals, whose spectral content change in time, Fourier representation is not appropriate. Unfortunately, most signals encountered in practice, regardless of their source, are non stationary in nature.

STFT allows analysis of non stationary signals by segmenting them into stationary short pieces, and computing the Fourier representation of each piece:

$$S(\tau, f) = \int x(t)\omega^*(t-\tau)e^{-2j\pi ft}dt \tag{3.5}$$

$$x(t) = \int_\tau \int_f S(\tau, f)\omega^*(t-\tau)e^{2j\pi ft}d\tau dt \tag{3.6}$$

where $\omega(t)$ is the windowing function, f and t are frequency and translation (time) parameters respectively, * is the complex conjugate operator, and $S(\tau, f)$ is STFT of x(t) at frequency f and translation t. For each frequency f, time localization is obtained through segmenting x(t) by $\omega(t-\tau)$, the windowing function centered at $t = \tau$ [35].

The time - frequency resolution depends entirely upon the choice of the windowing function $\omega(t)$ which gives Δt and Δf values. It is important to consider the bounds on temporal resolution Δt and frequency resolution Δf of STFT. The resolution in time and in frequency can not be arbitrarily small; their product is lower bounded by the time - bandwidth uncertainty principle or the Heisenberg inequality [36]:

$$\Delta t \cdot \Delta f \geq \frac{1}{4\pi} \tag{3.7}$$

This means that one must trade time resolution for frequency resolution, or vice versa.

STFT uses the same window for the analysis of the entire signal. The problem with STFT is that it provides constant resolution for all frequencies. If the signal has high frequency components for a short time, a narrow window would be suitable for good time resolution. However, a narrow window means wider frequency bands, resulting in poor frequency resolution. If the signal also features low frequency components of longer time interval, then a wider window must be used to obtain good frequency resolution [37]. This is the existence reason of WT. It provides varying time and frequency resolutions by using windows of different lenghts:

$$W(a,b) = \frac{1}{\sqrt{a}} \int x(t) \Psi^*(\frac{t-b}{a}) dt \tag{3.8}$$

$$x(t) = \frac{C_\Psi}{a^2} \int_{a>0} \int_b W(a,b) \Psi^*(\frac{t-b}{a}) da \cdot db \tag{3.9}$$

where a $>$ 0 and b are scale and translation parameters, respectively, Ψ is the mother wavelet, C_Ψ is a constant that depends on Ψ, and W(a,b) is the continuous wavelet transform of x(t). We can interpret Eq. 3.8 as an inner product of x(t) with the scaled and translated versions of the basis functions Ψ :

$$W(a,b) = \int x(t)\Psi^*_{(a,b)}(t)dt \tag{3.10}$$

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\Psi(\frac{t-b}{a}), a > 0, b \in \Re \tag{3.11}$$

Scaled and translated versions of the basis functions are obtained from one prototype function, the mother wavelet. In principle, the CWT produces an infinite number of coefficients, thus it provides a redundant representation of the signal.

The DWT provides a highly efficient wavelet representation that can be implemented with a simple recursive filter scheme and the original signal reconstruction can be obtained by an inverse filter [38]. The procedure of multi-resolution decomposition of a signal x[n] is schematically shown in Figure 3.3.

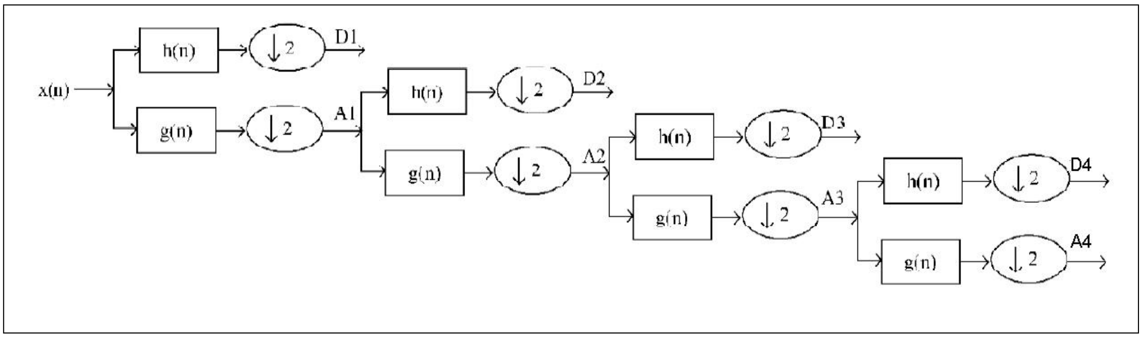


Figure 3.6 Decomposition of DWT; h[n] is the high pass filter, g[n] is the low pass filter.

The number of levels of decomposition is chosen on the basis of the dominant frequency components of the signal. According to the motor imagery EEG signals itself, level 4 and the wavelet of Daubechies order 10 was selected to extract μ and β rhytms as features [38]. As a result, the EEG signal is decomposed into the details D1-D3 and approximation A3. The ranges of different frequency bands for a 128 Hz sampled signal are shown in Table 3.1.

Table 3.1
Frequencies correspond to different levels of decomposition for Daubechies order 10 wavelet with a sample rate 128 Hz

Decomposed Signal	Frequency range Hz	Level
D1	32-64	1
D2	16-32	2
D3	8-16	3
A3	0-8	3

The extracted wavelet coefficients show the distribution of the motor imagery signal in time and frequency. It can be seen from the table that the component D3 decomposition is within the μ rhythm. Statistics over the set of wavelet coefficients were computed so as to reduce the total dimensions of the feature vectors. Statistical features of each sub band can be a combination of the following:

1. Mean of the coefficients of sub band
2. Median of the coefficients of sub band
3. Maximum value of the coefficients of sub band
4. Minimum value of the coefficients of sub band
5. Standard deviation of the coefficients of sub band
6. Mean absolute deviation of the coefficients of sub band
7. Mean of the absolute values of the coefficients of sub band

3.4 Classification with Artificial Neural Networks

3.4.1 Threshold Logical Unit

According to a simplified account, the human brain consists of about ten billion neurons, and a neuron is, on average, connected to several thousand other neurons. Through these connections, neurons both send and receive varying quantities of energy. One very important feature of neurons is that they do not react immediately to the reception of energy. Instead, they sum their received energies, and they send their own quantities of energy to other neurons only when this sum has reached a certain critical threshold. The brain learns by adjusting the number and strength of these connections. Even though this picture is a simplification of the biological facts, it is sufficiently powerful to serve as a model for the neural net [39].

The first step toward understanding neural networks is to abstract from the biological neuron, and to focus on its character as a threshold logic unit (TLU). A TLU is an object that inputs an array of weighted quantities, sums them, and if this sum meets or surpasses some threshold, outputs a quantity. Let's label these features. First, there are the inputs and their weights: X1,X2, ..., Xn and W1, W2, ...,Wn. Then, there are the Xi*Wi that are summed, which yields the activation level a, which can been seen in Eq. 3.12.

$$a = (X1 \times W1) + (X2 \times W2) + ... + (Xi \times Wi) + ... + (Xn \times Wn) \tag{3.12}$$

The threshold is called λ . Lastly, there is the output: y. When a $>=$ λ , y=1, else y=0. Notice that the output does not need to be discontinuous, since it could also be determined by a squashing function, s (or sigma), whose argument is a, and whose value is between 0 and 1. Then, y=s(a). A Threshold logical unit can be seen in Figure 3.7 [39].

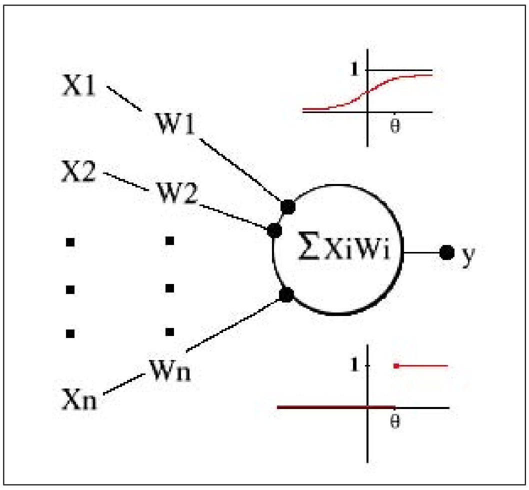


Figure 3.7 Threshold logic unit, with sigma function (top) and cutoff-step function (bottom).

A TLU can classify. For example, If a TLU that has two inputs, whose weights equal 1, and whose λ equals 1.5. When this TLU inputs <0,0>, <0,1>, <1,0>, and <1,1>, it outputs 0, 0, 0, and 1 respectively. This TLU classifies these inputs into two groups: the 1 group and the 0 group. In so far as a human brain that knows about logical conjunction (Boolean AND) would similarly classify logically conjoined sentences, this TLU knows something like logical conjunction [39].

This TLU has a geometric interpretation that clarifies the classification. Its four possible inputs correspond to four points on a Cartesian graph. From X1*W1+ X2*W2 = λ, in other words, the point at which the TLU switches its classificatory behavior, it follows that X2 = -X1 + 1.5. The graph of this equation cuts the four possible inputs into two spaces that correspond to the TLU's classifications. This is an instance of a more general principle about TLUs. In the case of a TLU with an arbitrary number of inputs, N, the set of possible inputs corresponds to a set of points in N-dimensional space. If these points can be cut by a hyperplane, in other words, an N-dimensional geometric figure corresponding to the line in the above example, then there is a set of weights and a threshold that define a TLU whose classifications match this cut [39]. N dimensional classification problem can be seen in Figure 3.8.

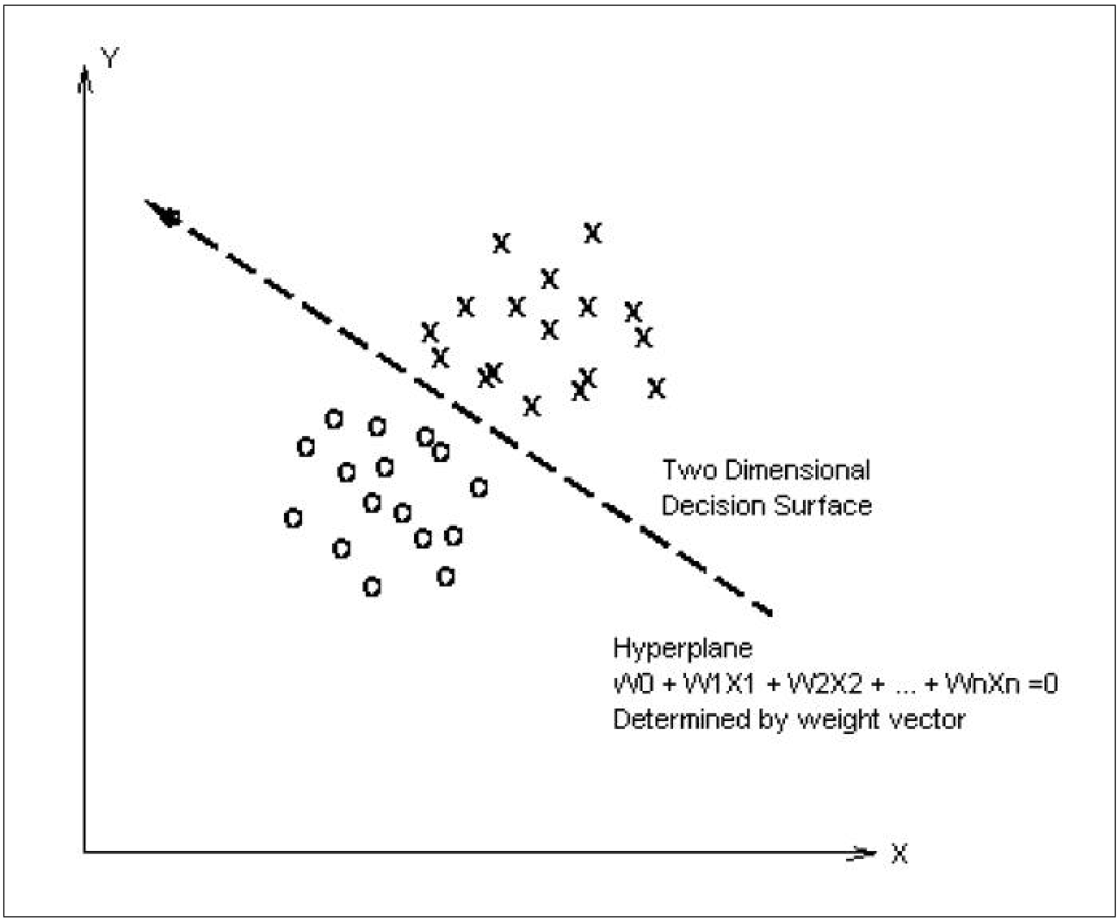


Figure 3.8 Pattern Classification Problem in n-dimensional space[39].

3.4.2 Perceptron

Since TLUs can classify, they have a memory. Neural nets are also supposed to learn. Their learning mechanism is modeled on the brain's adjustments of its neural connections. A TLU learns by changing its weights and threshold. Actually, the weight-threshold distinction is somewhat arbitrary from a mathematical point of view. Recall that the critical point at which a TLU outputs 1 instead of 0 is when the SUM (Xi * Wi) $>=\lambda$. This is equivalent to saying that the critical point is when the SUM (Xi * Wi) + (-1 * λ) $>=$ 0. So, it is possible to treat -1 as a constant input whose weight, λ, is adjusted in learning, or, to use the technical term, training. In this case, y=1 when SUM(Xi * Wi)+ (-1 * λ) $>=$ 0, else y=0. [39].

During training, a neural net inputs:

1. A series of examples of the items to be classified
2. Their proper classifications or targets

Such input can be viewed as a vector: <X1, X2, ...,Xn, λ, t>, where t is the target or true classification. The neural net uses these to modify its weights, and it aims to match its classifications with the targets in the training set. More precisely, this is supervised training, as opposed to unsupervised training. The former is based on examples accompanied by targets, whereas the latter is based on statistical analysis. Weight modification follows a learning rule. One idealized learning algorithm can be seen below [39].

```
fully_trained = FALSE
DO UNTIL (fully_trained):
fully_trained = TRUE
FOR EACH training_vector = <X1, X2, ..., Xn, theta, target>::
# Weights compared to theta
a = (X1 * W1)+(X2 * W2)+...+(Xn * Wn) - theta
y = sigma(a)
IF y != target:
fully_trained = FALSE
FOR EACH Wi:
MODIFY_WEIGHT(Wi) # According to the training rule
IF (fully_trained):
BREAK
```

There are many training rules. One possible rule is based on the idea that weight and threshold modification should be determined by a fraction of (t - y). This is accomplished by introducing σ ($0 < \sigma < 1$), which is called the learning rate. The change in Wi equals (σ * (t - y) * Xi). When alpha is close to 0, the neural net will engage in more conservative weight modifications, and when it is close to 1, it will make more radical weight modifications. A neural net that uses this rule is known as a perceptron, and this rule is called the perceptron learning rule [39].

3.4.3 Activation Function

A biological neuron initiates an action potential whenever it is stimulated beyond a threshold level. As in its biological model, the unit in an artificial neural network also produces activity level if the "total input" received from other units is greater than or equal to a threshold. The total input of a unit is a scalar quantity, and is usually taken to be a linear function of the activity level of the units that provide input to this unit. If x is being the total input of a unit, and λ is the threshold for activation of the unit, then the output y of the unit is y = 1 if x >= λ else y = 0 where x = $\sum y_i w_i$ and y_i are the activation level of units giving their output as input to this unit [40].

The activation function, as given in the equations above, is the step function. The step function has two discrete states: 0 or 1. However, in most of the neural network applications, the sigmoid function is used rather than the step function. The sigmoid function is a smoothed version of the step function. The reasons of use of sigmoid function are that it is continuous and thus differentiable [40]. Sigmoid function can be seen in Figure 3.9.

3.4.4 Backpropagation

Backpropagation is one of the most popular and successful learning methods for multilayer feed forward networks. There are two distinct phases of the operation

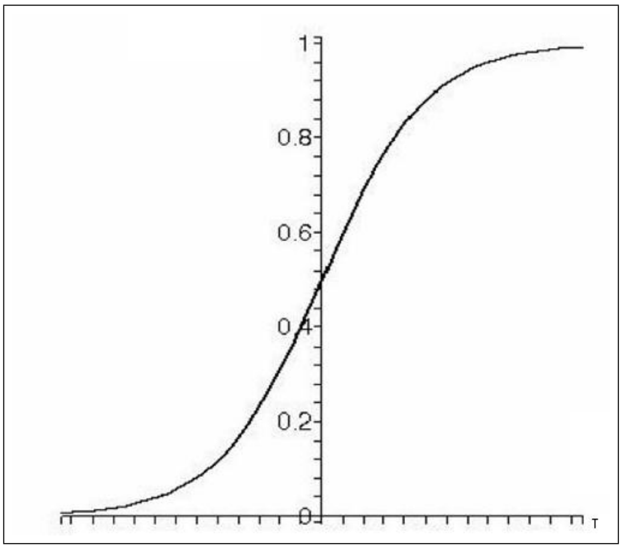


Figure 3.9 Sigmoid Activation Function, gives the nonlinear processing capability to the neuron.

of backpropagation learning: the forward and the backward phase. In the forward phase, the input signal propagates through the network layer from the input layer to the output layer, eventually producing some response at the output of the network. The actual response produced is compared with a desired response, generating error signals that are then propagated in the backward direction through the network. In this backward phase of operation, the free parameters of the network are adjusted so as to minimize the sum of squared errors. Backpropagation learning has been applied successfully to some difficult problems such as speech recognition [41].

Backpropagation is supervised learning since during training, correct output is provided to the network. It uses gradient descent on the error and it modifies the weights so that the error gets smaller and smaller. The activation function of the neurons is usually sigmoidal, a sort of threshold function, but differentiable [40].

Many researchers believe that backpropagation can be an explanation how biological neuron learns. The most important contradiction to this assumption is that synaptic connections in biological neurons transmit information (signal) in one direction only while during learning in backpropagation reverse propagation of error is needed from the output layer to the input layer [42].

4. CLASSIFICATION IMPLEMENTATION AND RESULTS

In this BCI study sensorimotor Rhythms of the EEG is needed for motor imagery BCI operation. First a signal acquisition system was developed in the laboratory with the use of ICs from Texas Instruments. During hardware design, Texas Instruments documentation and Modular EEG design of OpenEEG Project was used [43, 44]. Hardware design and a low power implementation is explained in the next chapter.

In this chapter, designed and evaluated feature extraction and classification methods is discussed with the results. Feature channels for BCI control was extracted from the SMR rhythms with two distinct methods, discrete wavelet transform and fast Fourier transform, respectively. The statistics of these decompositions were used as inputs to a two layer feedforward backpropagation neural network for classification of motor imagery tasks. These designs were developed and implemented in Matlab which was tested with data from BCI Competition II [12]. With direction of the results an efficient algorithm for low power microcontroller application was designed based on FIR band pass filters instead of wavelet transform to extract the sub band features.

The block diagram of the designed processing solution for BCI motor imagery classification is displayed in Figure 4.1.

The remaining sections are organized as follows. First data set provided by BCI Competition is explained. Then, dimensionality reduction and feature extraction of motor imagery trials with Wavelet transform and FFT are explained. Neural network and classification implementation comes ofter that. In the last section designed system for low power embedded application is described.

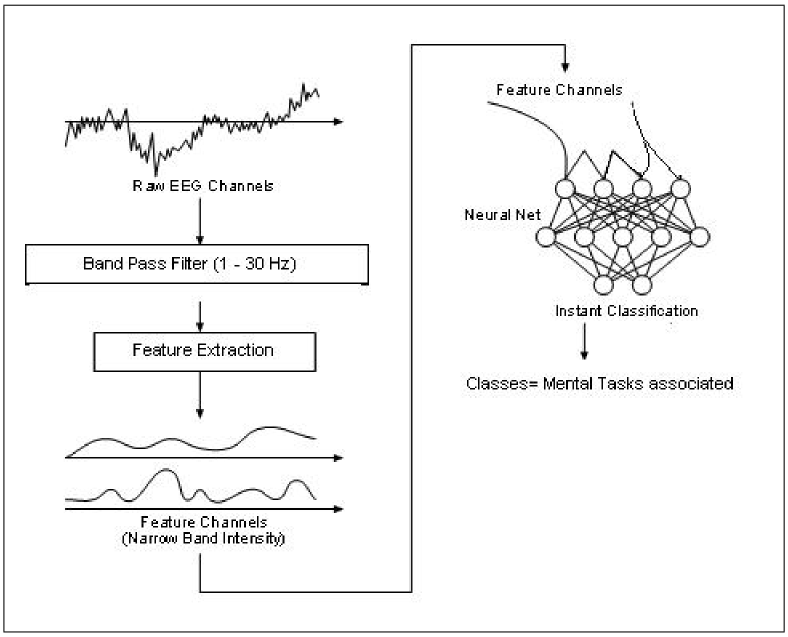


Figure 4.1 Block Diagram Of the BCI processing system

4.1 Motor Imagery Trials And Data Set Description

The design, training and test of the feature extraction and classification system was done with the measurements taken for BCI Competition II. This chapter explains the methodology of the experiment.

This data set is provided by the Department of Medical Informatics, Institute for Biomedical Engineering, University of Technology, Graz, Austria. This data set was recorded from a healthy subject (female, 25 yrs) during a feedback session. The subject sat in a comfortable chair with armrests. The task was to control a feedback bar in one dimension by imagination of left- or right-hand movements. The order of left and right cues was random. The experiment included seven runs with 40 trials each. All runs were conducted on the same day with breaks of several minutes in between. The data set consists of 280 trials of 9-s length. The first 2 s were quiet. At t = 2 s, an acoustic stimulus indicated the beginning of the trial, and a cross (+) was displayed for 1 s. Then, at t = 3 s, an arrow (left or right) was displayed as a cue stimulus. The subject was asked to use imagination as described above to move the feedback bar into the direction of the cue. The feedback was based on AAR parameters calculated from channels C3 and C4. The AAR parameters were combined with a discriminant analysis into one output parameter (similar to [45] and [46]). The recording was made using a g.tec amplifier and AgAgCl electrodes. Three bipolar EEG channels were measured over C3, Cz, and C4. EEG was sampled with 128 Hz and was filtered between 0.5 and 30 Hz. Similar experiments are described in [45] [47]. The trials for training and testing were randomly selected to prevent any systematic effect due to the feedback [12]. Electrode positions and timing scheme of the experiment can be seen in Figure 4.2.

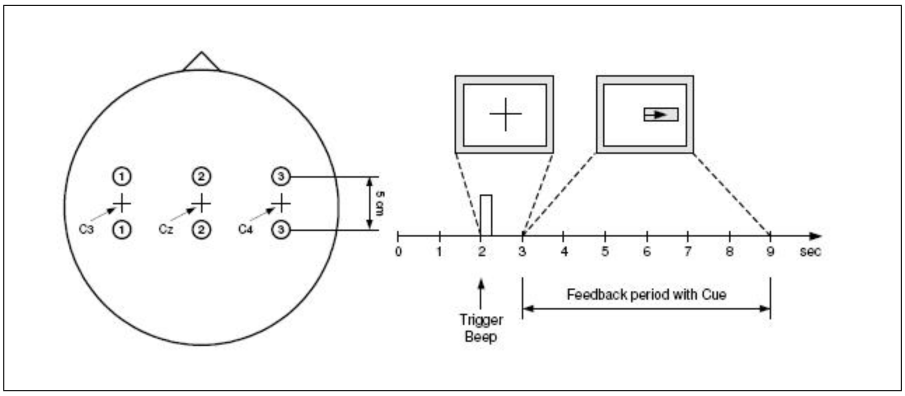


Figure 4.2 The Competition Data Set Electrode positions (left) and timing scheme (right)[12].

Central brain oscillations in the mu rhythm in the range of 7-12Hz and beta above 13Hz bands are strongly related to sensorimotor tasks. Sensory stimulation, motor behavior, mental imagery can change the functional connectivity cortex which results in an amplitude suppression or in an amplitude enhancement .This phenomenon was also called event related desynchronization (ERD) and event-related synchronization (ERS). Left and right hand movement imagery is typically accompanied with ERD in the mu and beta rhythms and has the characteristic of contralateral dominance [38].

The power spectrum on C3 and C4 of the training set are shown in Figure 4.3. It indicates that the power spectrum mainly distribute in the range of 8 - 13 Hz and 19 - 24 Hz. In addition, the power of mu and beta rhythms evoked by right hand movement imagery is lower than that of left hand movement imagery for channel C3, and it is contrary for channel C4 which is consistent with the principle of contralateral dominance. This led us to use wavelet decomposition to extract the differences between the two motor imagery tasks [38].

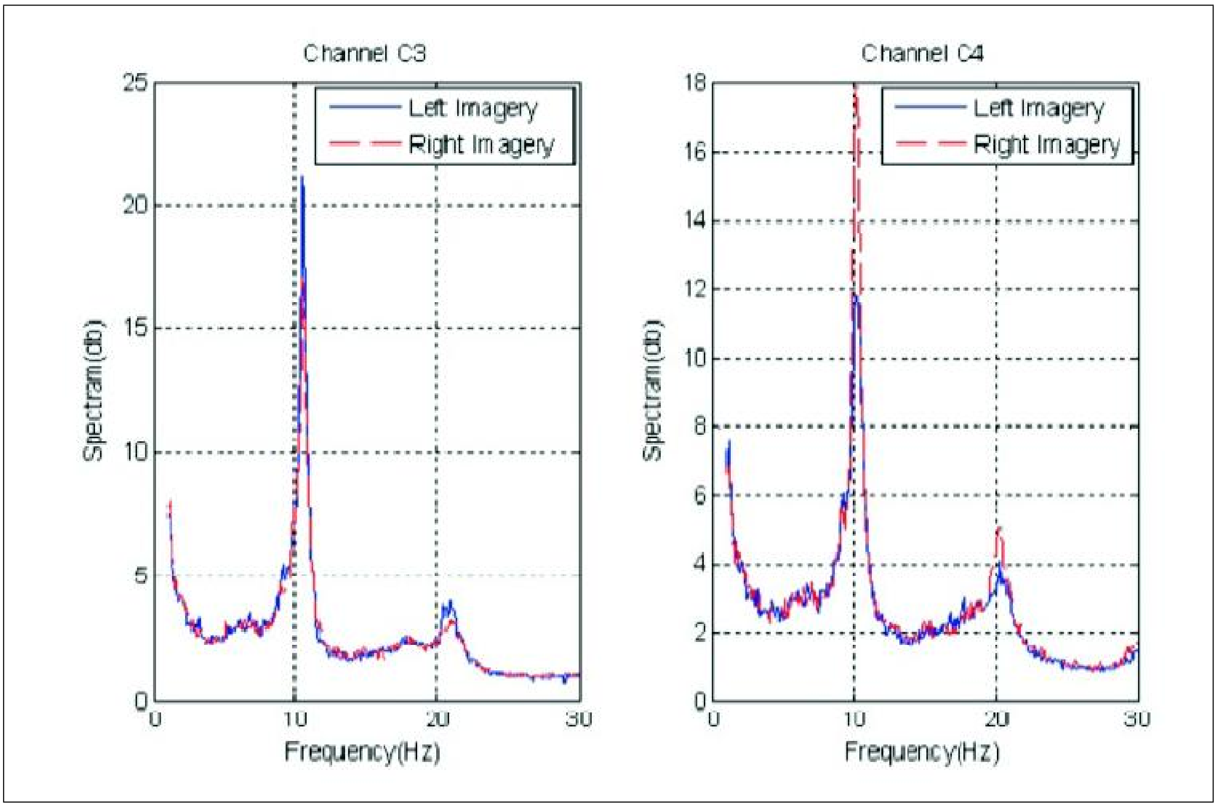


Figure 4.3 Average power spectrums on Channel C3 and C4[38].

4.2 Feature Extraction with Discrete Wavelet Transform

As seen in Table 3.1, frequency components of D2 and D3 detail levels of the wavelet decomposition are in the beta and mu bands, respectively. Daubechies order 10 wavelet with 4 decomposition levels were used to decompose all of the 280 trials into the three frequency bands. The extracted wavelet coefficients show the distribution of the motor imagery signal in time and frequency. The original signal is sampled with 128 Hz rate and longs 9 seconds. It is down sampled by 2 in each decomposition level, so D2 level contains 288 and D3 contains 144 coefficients since the complete single trial composes of 1152 coefficients (128 Hz * 9 seconds).

Statistics over the set of wavelet coefficients of sub bands D2 and D3 where computed for each trial so as to reduce the total dimension of the feature vectors. The statistical features of each sub band are selected from the explained ones as follows:

1. Standard deviation of coefficients in the sub band
2. Mean of the absolute values of the coefficients in the sub band
3. Maximum values of the coefficients in the sub band
4. Mean absolute deviation of the coefficients in the sub band

These features represent the frequency distribution and the amount of changes in frequency distribution. Thus 16 statistical features of wavelet coefficients are obtained for two channels of C3 and C4 for each trial (4 statistical features of coefficients listed above for D2 level and same 4 again for D3 level for each two channels).

Example decompositions of one right hand movement imagery and one left hand movement imagery task are given in Figures 4.4 to 4.11. Figures 4.4 to 4.7 represent right hand movement imagery, 4.8 to 4.11 represent left hand movement imagery.

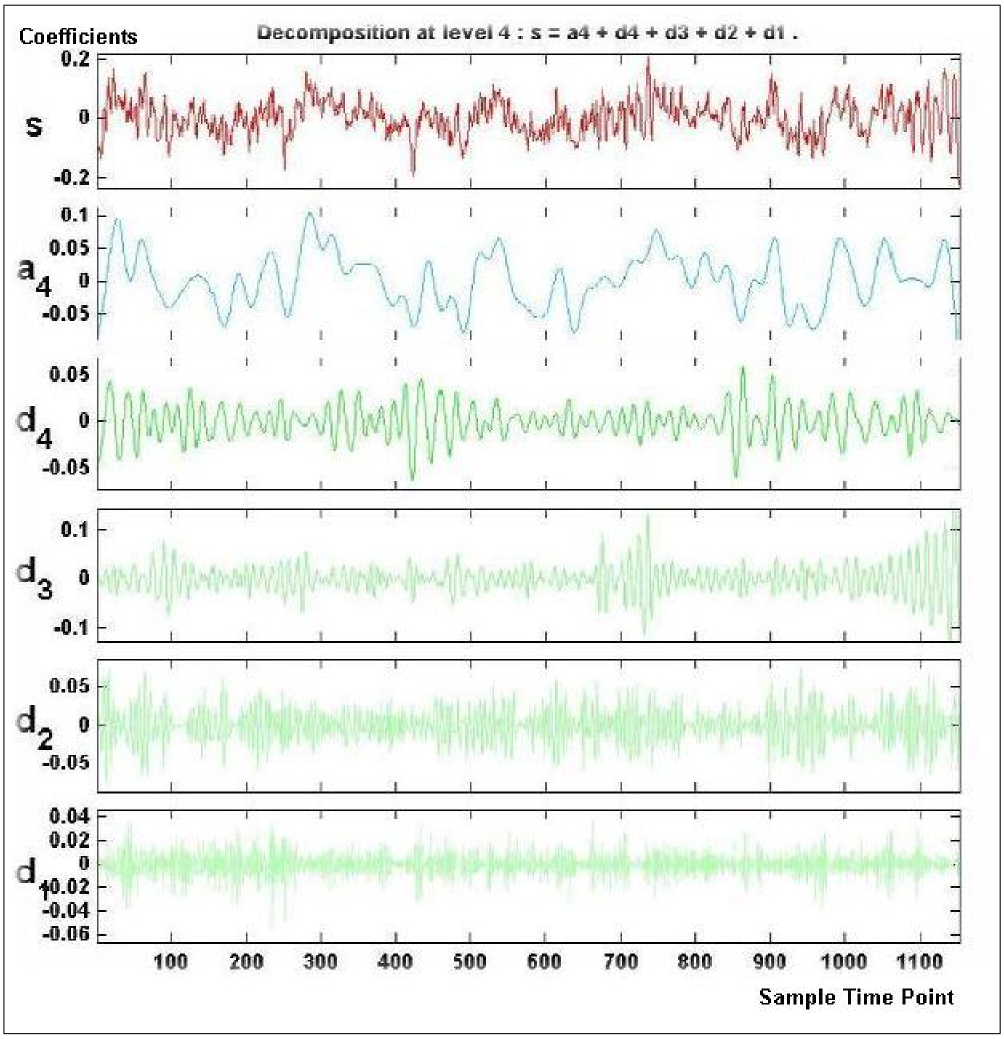
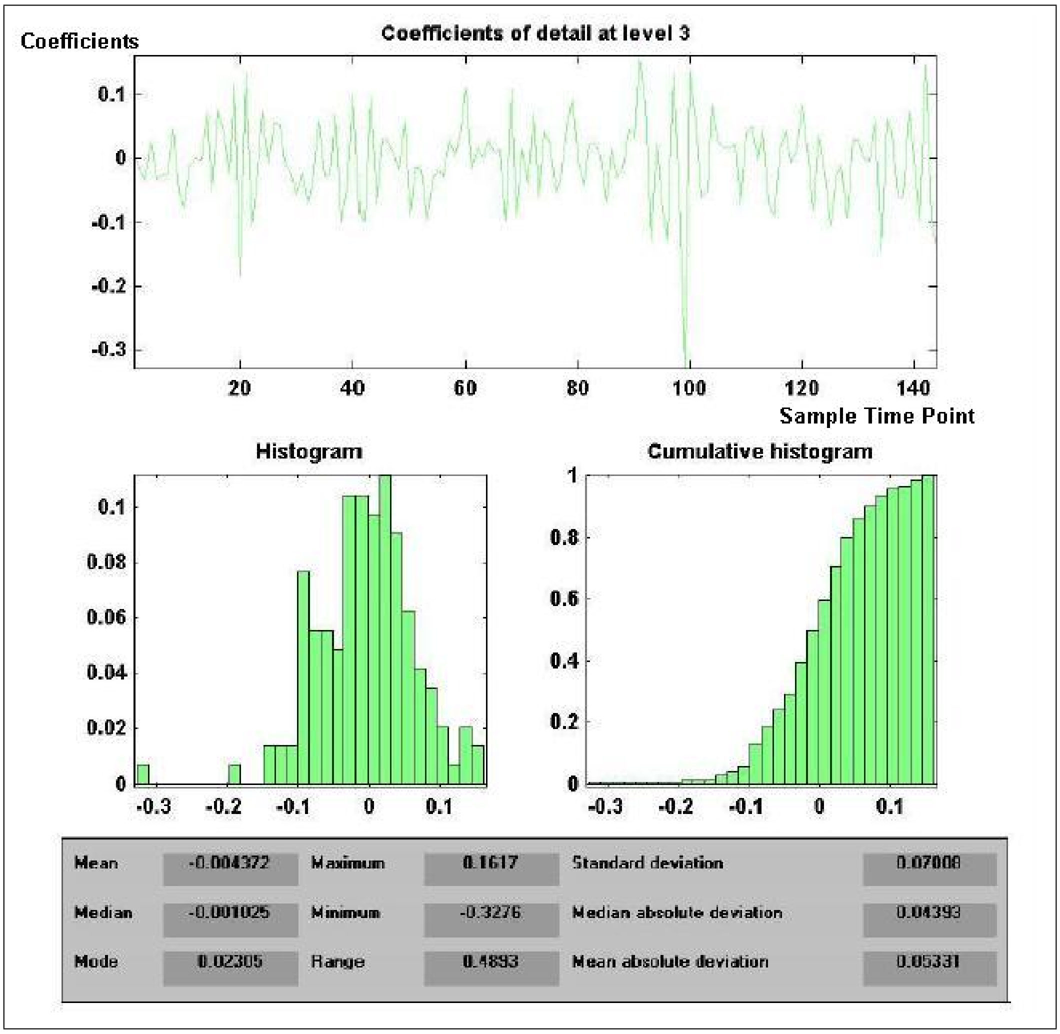


Figure 4.4 db10 level 4 decomposition of trial over channel C3 during a right hand movement imagery task. D3 represents mu, D2 represents beta rhythms. Beginning of the motor imagery is time point 400 (after t=3 in 9 second trial).



Mean	-0.004372	Maximum	0.1617	Standard deviation	0.07008
Median	-0.001025	Minimum	-0.3276	Median absolute deviation	0.04393
Mode	0.02305	Range	0.4893	Mean absolute deviation	0.05331

Figure 4.5 Statistics of wavelet coefficients of sub band D3 of channel C3 for right hand motor imagery task. All available statistics values are shown but only 4 of them were used as listed in Section 4.2.

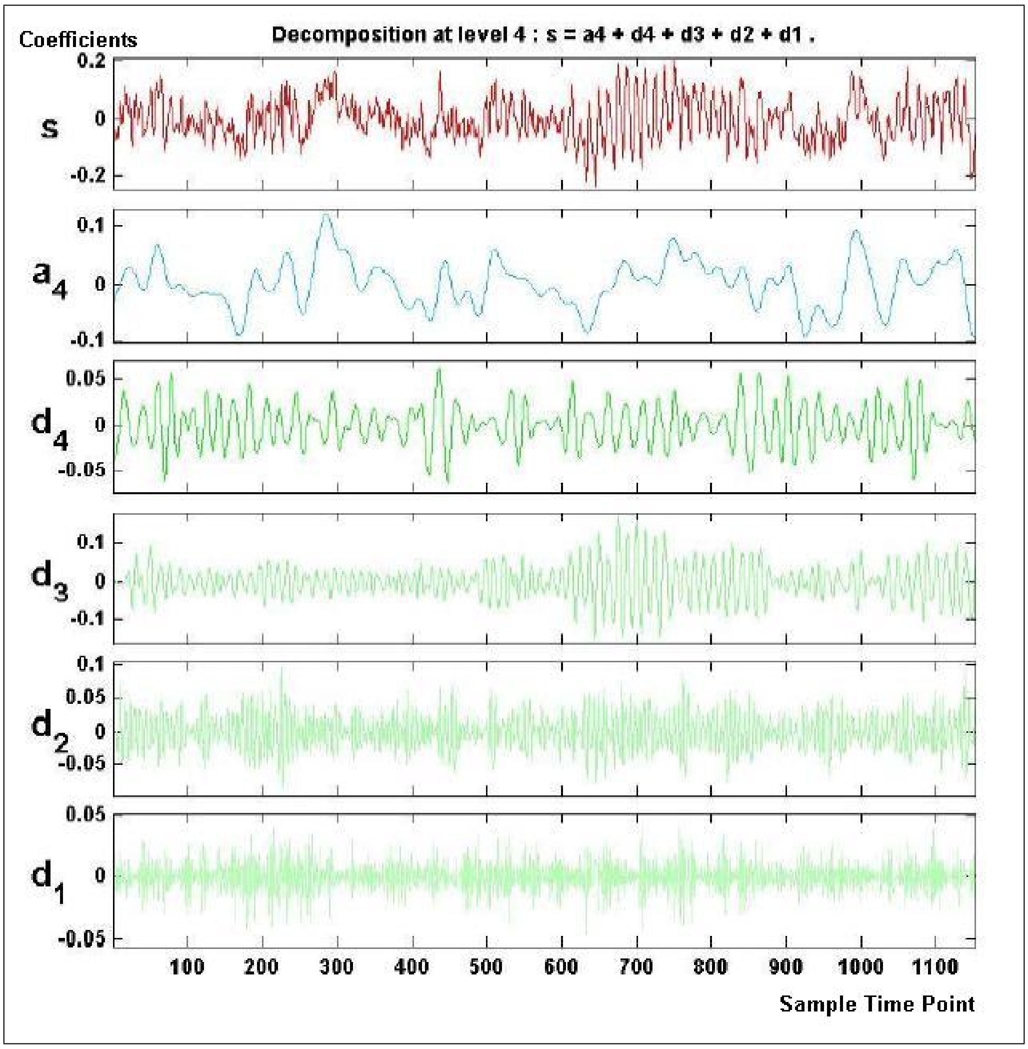
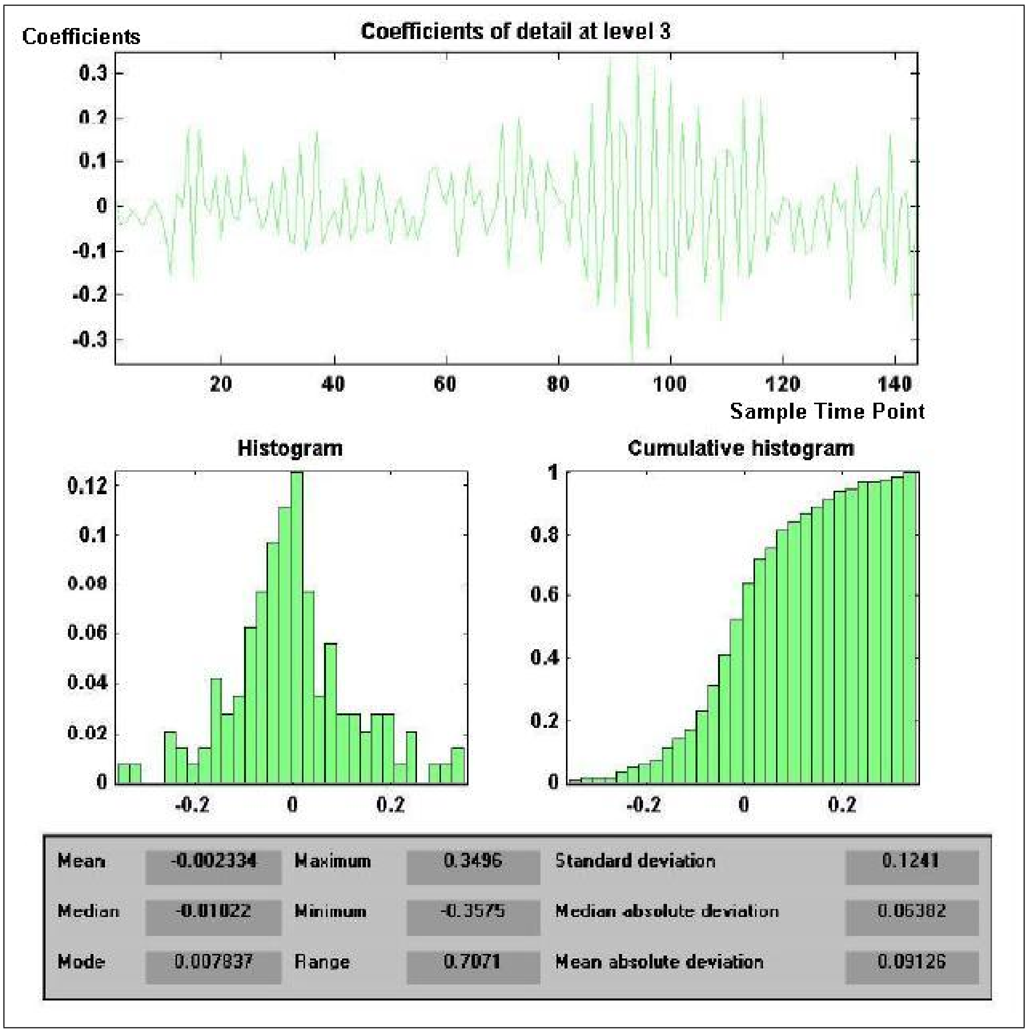


Figure 4.6 db10 level 4 decomposition of trial over channel C4 during a right hand movement imagery task. D3 represents mu, D2 represents beta rhythms. Beginning of the motor imagery is time point 400 (after t=3 in 9 second trial). Desynchronization of mu rhythm during imagery task is apparent in detail 3.



Mean	-0.002334	Maximum	0.3496	Standard deviation	0.1241
Median	-0.01022	Minimum	-0.3575	Median absolute deviation	0.06382
Mode	0.007837	Range	0.7071	Mean absolute deviation	0.09126

Figure 4.7 Statistics of wavelet coefficients of sub band D3 of channel C4 for right hand motor imagery task. All available statistics values are shown but only 4 of them were used as listed in Section 4.2.

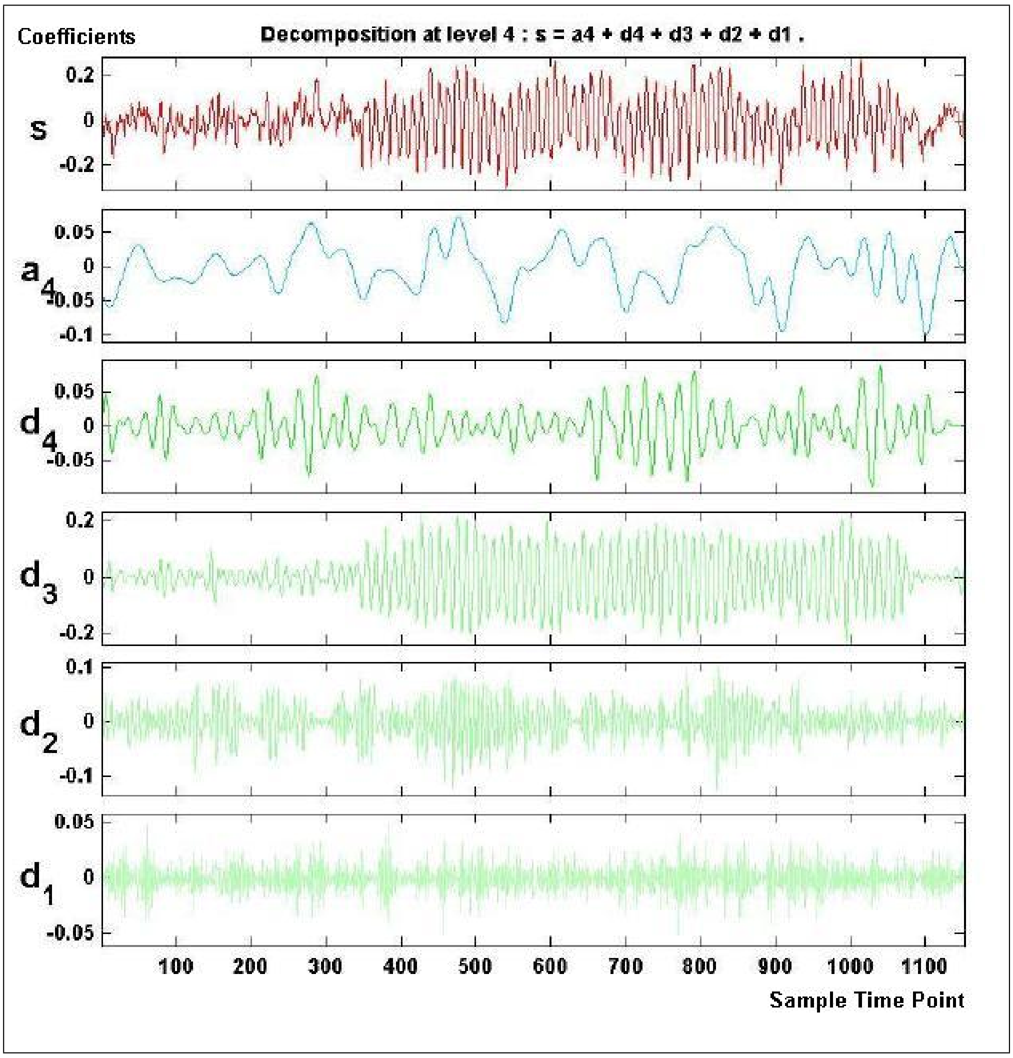
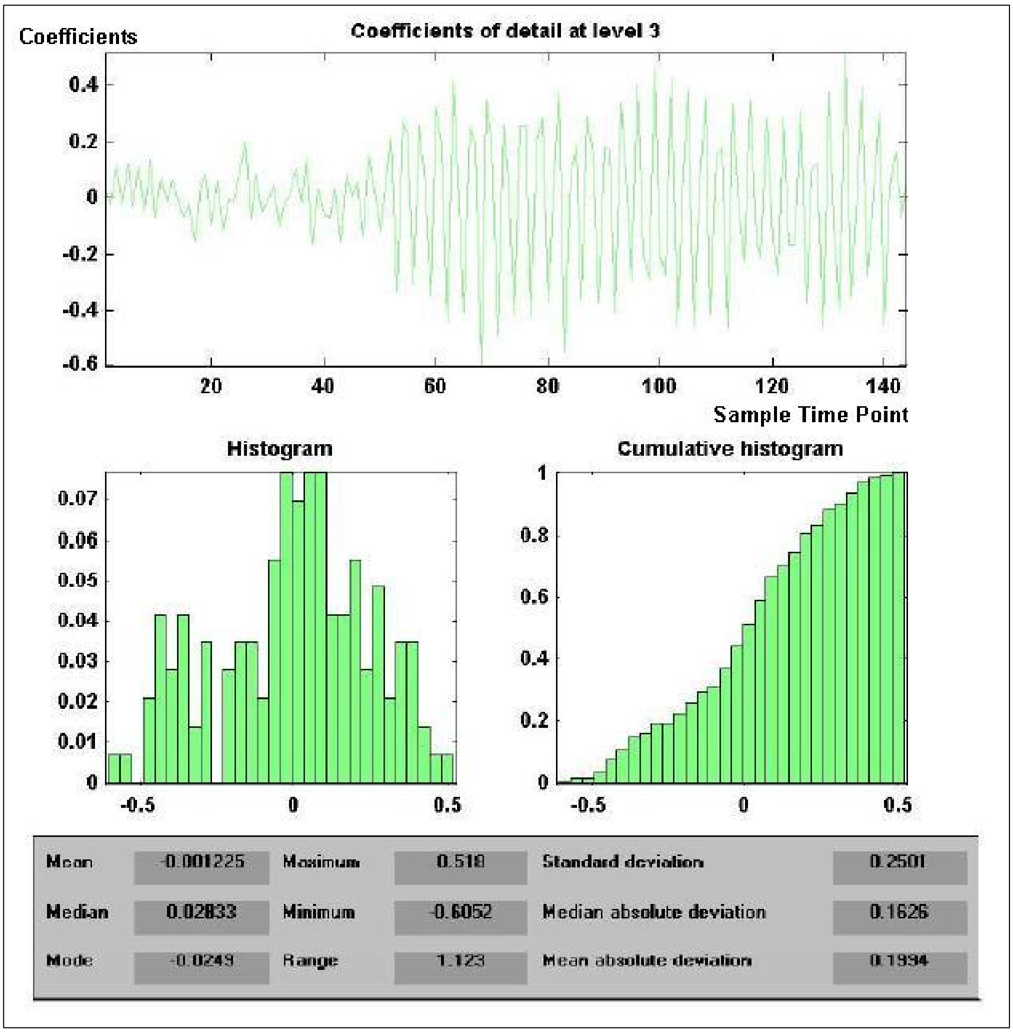


Figure 4.8 db10 level 4 decomposition of trial over channel C3 during a left hand movement imagery task. D3 represents mu, D2 represents beta rhythms. Beginning of the motor imagery is time point 400 (after t=3 in 9 second trial). Desynchronization of mu rhythm during imagery task is apparent in detail 3.



Mean	-0.001225	Maximum	0.518	Standard deviation	0.2501
Median	0.02833	Minimum	-0.6052	Median absolute deviation	0.1626
Mode	-0.0249	Range	1.123	Mean absolute deviation	0.1994

Figure 4.9 Statistics of wavelet coefficients of sub band D3 of channel C3 for left hand motor imagery task. All available statistics values are shown but only 4 of them were used as listed in Section 4.2.

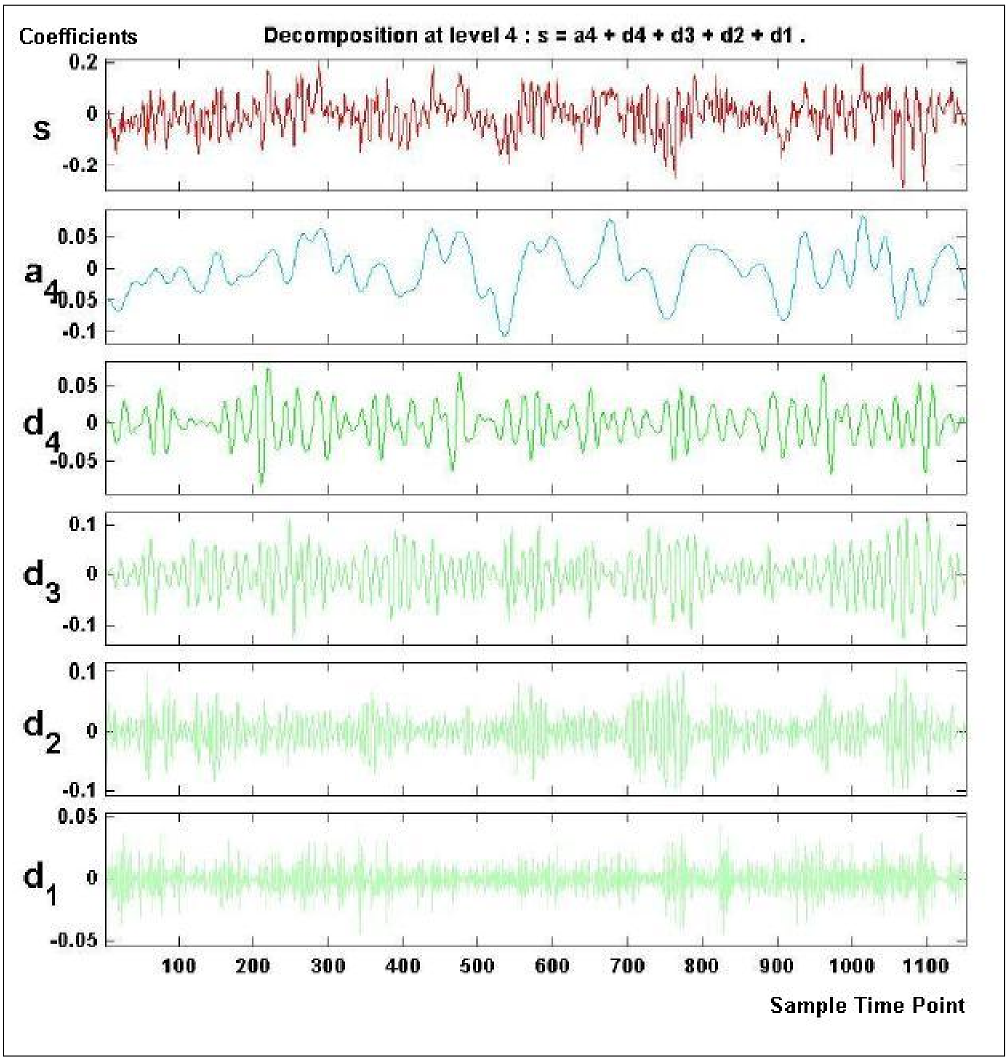
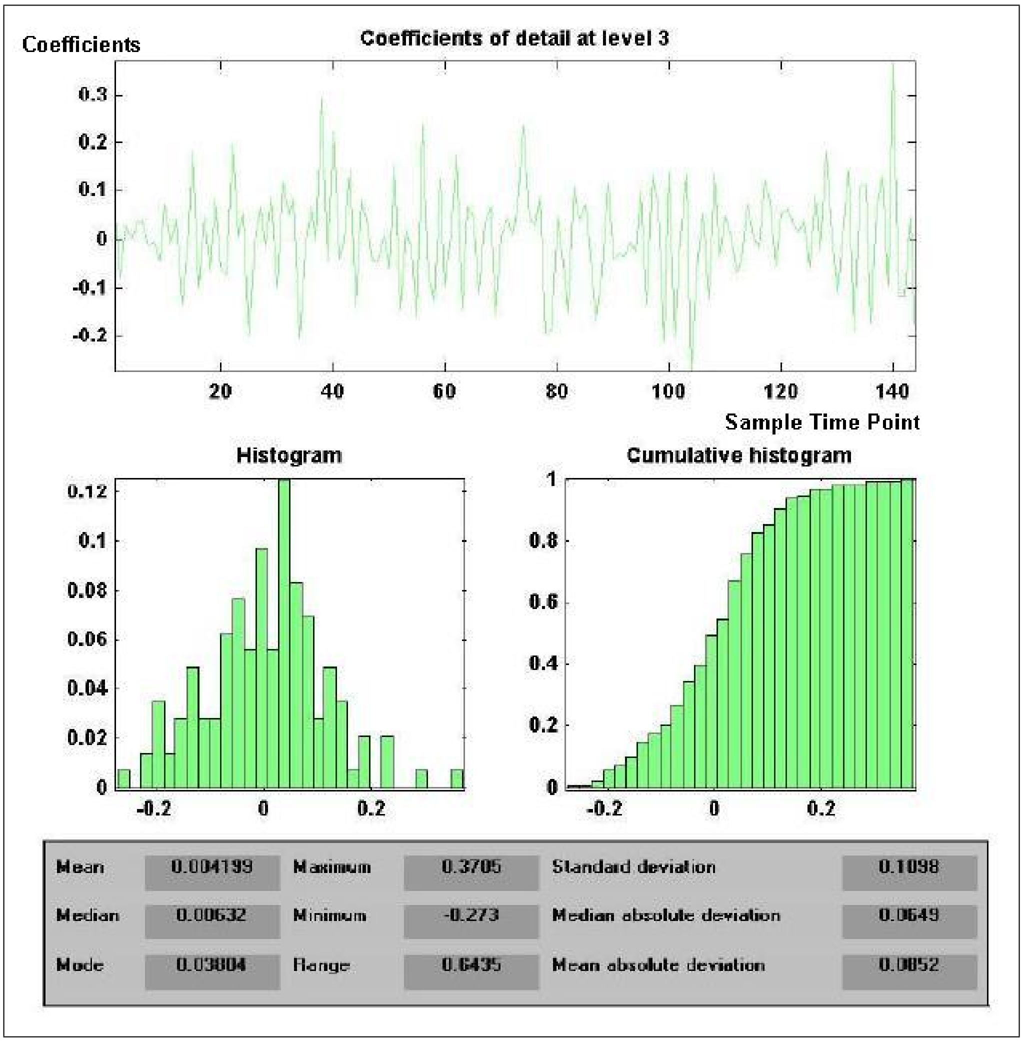


Figure 4.10 db10 level 4 decomposition of trial over channel C4 during a left hand movement imagery task. D3 represents mu, D2 represents beta rhythms. Beginning of the motor imagery is time point 400 (after t=3 in 9 second trial).



Mean	0.004199	Maximum	0.3705	Standard deviation	0.1098
Median	0.00632	Minimum	-0.273	Median absolute deviation	0.0649
Mode	0.03804	Range	0.6435	Mean absolute deviation	0.0852

Figure 4.11 Statistics of wavelet coefficients of sub band D3 of channel C4 for left hand motor imagery task. All available statistics values are shown but only 4 of them were used as listed in Section 4.2.

4.3 Classification

4.3.1 Classification based on Wavelet features

A two layer feed forward backpropagation neural network was used for classification of the trials. Extracted 16 statistical features of wavelet coefficients for each trial were used as inputs to the neural network. The structure of the final designed artificial neural network is shown in Figure 4.12. The network has 20 neurons in the hidden layer with tanh activation function. The final layer has two neurons with logistic activation.

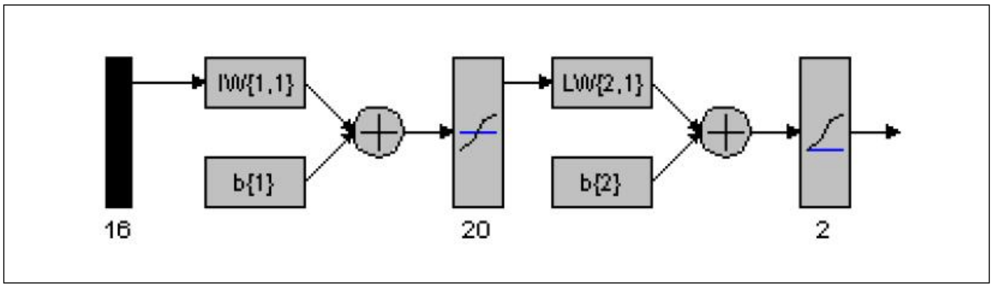


Figure 4.12 Two layer Feed Forward Artifical Neural Network used for classification. First layer uses the tanh activation function, the second a normal logistic activation.

The network was trained with 140 trials of training data set and tested against 140 trials of the testing data set to discriminate between right and left hand movement imagery tasks. Different numbers of hidden neurons and different feature vectors were used for classification. The results are displayed in Table 4.1. All of the trials of the data set correspond to one of the two classes. Because of that all of the false classifications are false positives.

Best classification result was taken from the Network2 and absoluted statistics of feature coefficients of wavelet transform.

Table 4.1
Properties of Neural Networks and Classification Results based on DWT features

	Output Layer	Hidden Layer	Inputs	Train Function	Hidden Units	Vector Coeff.	True Classification Per.
Network1	Logistic	Tanh	16	Trainscg	4	absoluted	75
Network1	Logistic	Tanh	16	Trainscg	4	squared	68
Network2	Logistic	Tanh	16	Trainlm	20	absoluted	89
Network2	Logistic	Tanh	16	Trainlm	20	squared	74

4.3.2 Classification based on Power Spectrum features

To classify based on power spectrum features of C3 and C4 channels, FFT of each train and test trails was computed. The average of power spectrum of all trials was shown in Figure 4.3. The features used for FFT based classification are as follows:

1. Mean of the power in μ band (8 - 12 Hz)
2. Mean of the power in β band (19 - 24 Hz)
3. Maximum value in μ band (8 - 12 Hz)
4. Maximum value in β band (19 - 24 Hz)

The results are displayed in Table 4.2. All of the trials of the data set correspond to one of the two classes. Because of that all of the false classifications are false positives.

Table 4.2
Classification Results based on FFT features

Feature Vector	True Classification Percent
Ratio of mean values between two bands	75
Only Max value in μ band	72
Only Max value in β band	63

4.4 Developed Method for An Embedded Low Power Application

Fast and efficient algorithms are needed for embedded and low power applications because of the limitations listed below:

1. Multiplication is the core operation for filtering
2. Microcontrollers are fixed-point devices, simple in architecture and have limited CPU cycles and memory
3. Real-time signal processing broadens the application space
4. Less CPU cycles implies lower current consumption with increased battery life, especially for hand-held devices
5. Reduced CPU utilization increases usability and accommodates better and advanced features

Because of these limitations a resource efficient model was developed using the results from Wavelet and FFT features classification. In this approach, band pass FIR filters were used instead of Wavelet Decomposition for feature extraction.

Since multiplication is the core operation of filtering, it must be done in an efficient way to save time and power. Hardware multiplier and multichannel Direct Memory Access (DMA) features are the main differences between a Digital Signal Processor (DSP) and a microcontroller (MCU).

Features of a typical Hardware Multiplier:

1. Simple to use
2. HW multiplier includes a multiply and accumulate (MAC) function for filtering

3. ADC samples are integers, coefficients need to be converted from real numbers to integers, via scaling

For this implementation FG4618 was selected from MSP430 MCU family which includes a hardware multiplier with low power features. Since it has a hardware multiplier, a FIR filter can be implemented in software easily. The designed solution has following key functions:

1. Two FIR band pass filters, one for μ band and one for β band to extract the features like wavelet sub band decomposition
2. Designed filter coefficients are scaled to integers to multiply with ADC samples efficiently.
3. Absolute values of filtered sample arrays are computed
4. Mean values of absoluted arrays are computed as features
5. These features are feeded to a trained MLP neural network with predefined weights for classification

Designed filter specifications are given in subsection.

4.4.1 Designed FIR Filters

4.4.1.1 Band Pass FIR Filter 1. Filter specifications:

1. Filter length = 23
2. Sampling frequency = 128 Hz
3. Lower Cut-off frequency = 7 Hz
4. Upper Cut-off frequency = 12 Hz

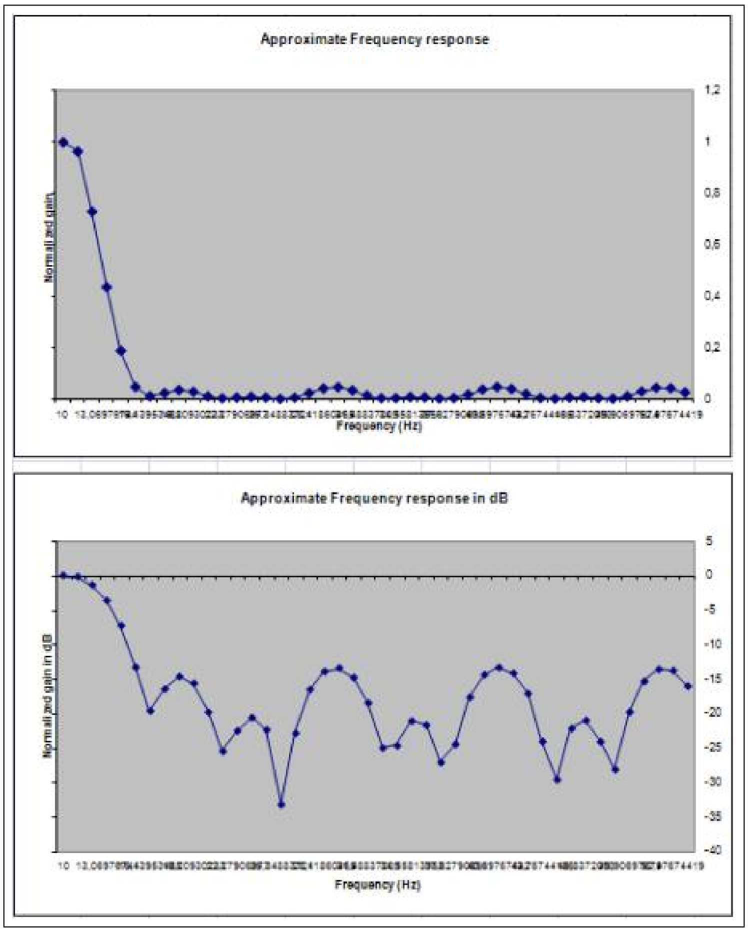


Figure 4.13 Frequency Response of Band Pass Filter 1 (BPF1)

4.4.1.2 Band Pass FIR Filter 2. Filter specifications:

1. Filter length = 27
2. Sampling frequency = 128 Hz
3. Lower Cut-off frequency = 17 Hz
4. Upper Cut-off frequency = 23 Hz

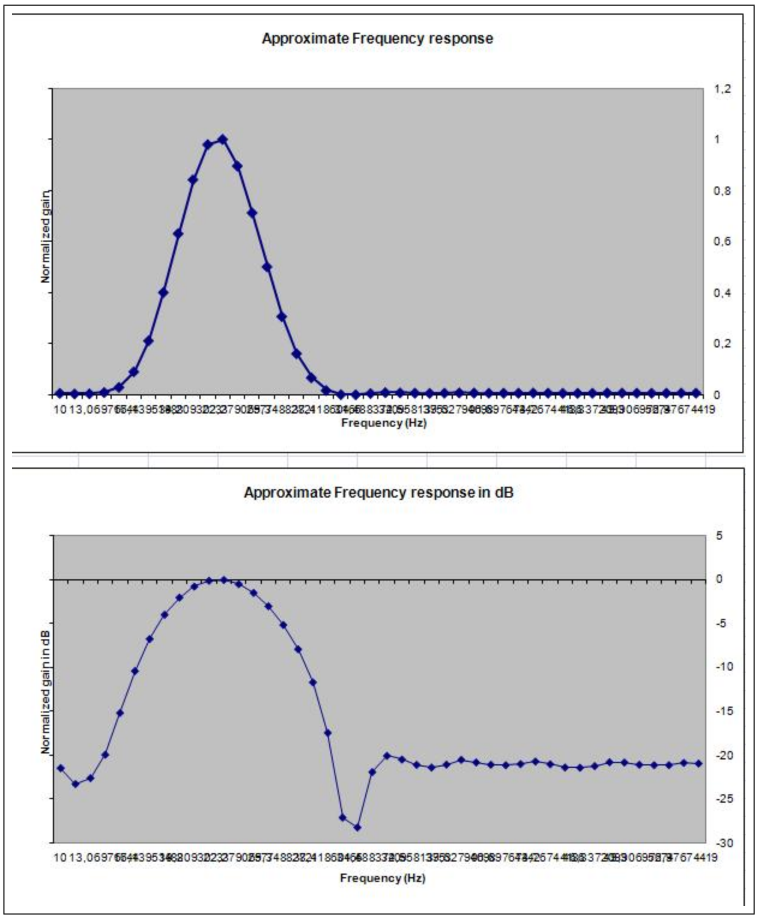


Figure 4.14 Frequency Response of Band Pass Filter 2 (BPF2)

4.4.2 Results of Classification

Samples of calculated features can be seen in the Table 4.3.

Classification results displayed in Table 4.4.

Hardware implementation of this developed method in laboratory is discussed in the next chapter.

Table 4.3
First 30 features from training set as an example. Integer columns show means of the filtered and absoluted samples, floating point colums show the scaled data between [-1,1] for neural network input.

	C3 BPF1	c3 BPF1 SCL	C4 BPF1	C4 BPF1 SCL	C3 BPF2	C3 BPF2 SCL	C4 BPF2	C4 BPF2 SCL	CLASS
t1	260	-0,57	225	-0,67	98	-0,75	142	-0,45	-1
t2	325	-0,4	320	-0,41	118	-0,61	183	-0,18	1
t3	593	0,31	258	-0,58	135	-0,5	152	-0,39	1
t4	175	-0,8	256	-0,58	136	-0,49	172	-0,25	1
t5	205	-0,72	279	-0,52	134	-0,51	206	-0,03	1
t6	518	0,12	346	-0,34	212	0,01	226	0,11	-1
t7	641	0,44	274	-0,54	185	-0,17	164	-0,31	-1
t8	345	-0,35	414	-0,16	181	-0,19	290	0,53	1
t9	576	0,27	325	-0,4	188	-0,15	184	-0,17	1
t10	733	0,69	305	-0,45	242	0,21	194	-0,11	-1
t11	621	0,39	422	-0,14	246	0,24	237	0,18	-1
t12	438	-0,1	334	-0,38	183	-0,18	229	0,13	1
t13	782	0,82	301	-0,46	219	0,06	148	-0,41	-1
t14	309	-0,44	404	-0,19	161	-0,33	315	0,7	1
t15	658	0,49	369	-0,28	229	0,13	219	0,06	1
t16	568	0,25	381	-0,25	194	-0,11	283	0,49	1
t17	491	0,04	370	-0,28	236	0,17	243	0,22	1
t18	423	-0,14	333	-0,38	199	-0,07	231	0,14	-1
t19	617	0,38	583	0,29	190	-0,13	208	-0,01	-1
t20	285	-0,51	244	-0,62	120	-0,6	152	-0,39	-1
t21	341	-0,36	284	-0,51	151	-0,39	177	-0,22	-1
t22	381	-0,25	348	-0,34	135	-0,5	191	-0,13	1
t23	203	-0,73	194	-0,75	120	-0,6	138	-0,48	-1
t24	142	-0,89	378	-0,26	109	-0,67	226	0,11	1
t25	312	-0,43	201	-0,73	175	-0,23	158	-0,35	-1
t26	473	-0,01	404	-0,19	219	0,06	291	0,54	1
t27	251	-0,6	229	-0,66	151	-0,39	188	-0,15	1
t28	514	0,1	394	-0,22	197	-0,09	191	-0,13	1
t29	773	0,79	518	0,11	263	0,35	252	0,28	1
t30	414	-0,16	390	-0,23	190	-0,13	274	0,43	-1

Table 4.4
Classification Results based on developed FIR filter based features to be implemented in embedded hardware

	Output Layer	Hidden Layer	Inputs	Hidden Units	True Classification Percent
Network1	Linear	Tanh	4	4	61
Network2	Linear	Tanh	4	8	65
Network3	Linear	Tanh	4	20	69
Network3 with purification	Linear	Tanh	4	20	72

5. HARDWARE IMPLEMENTATION AND RESULTS

As discussed in previous chapters, sensorimotor Rhythms of the EEG is needed for motor imagery BCI operation. First a signal acquisition system was developed in the laboratory with the use of ICs from Texas Instruments. During hardware design, Texas Instruments documentation and Modular EEG design of OpenEEG Project was used [43, 44]. Then the developed method which is presented in previous chapter was implemented on this hardware. Hardware design and the low power implementation is explained in this chapter.

5.1 Signal Acquisition

The device was implemented in the lab with the use of Texas Instruments ICs and microcontrollers to create a circuit that amplifies, filters and digitizes brainwave signals. The design includes 1 INA333 Instrumentation Amplifier, 3 OPA333 Operational Amplifiers for each channel and 1 MSP430 Microcontroller which is in the experimenter board. Since low power and battery life are important considerations for portable or medical equipment, low power devices were particularly preferred for the design. Requirements of the acquisition system:

1. Detect brainwave signals greater than 10 μ V
2. Process signals, filter frequencies between 0.16 and 30 Hz
3. Amplify signal to a ADC detectable level (volts)
4. Convert analog signal to digital (12 bit resolution, 128 Hz sampling rate)
5. Process the signal as described in the section 4.4 “Developed Method for An Embedded Low Power Application”

After analyzing the above requirements , the design was developed to detect, amplify and filter the brainwaves as shown in Figure 5.1.

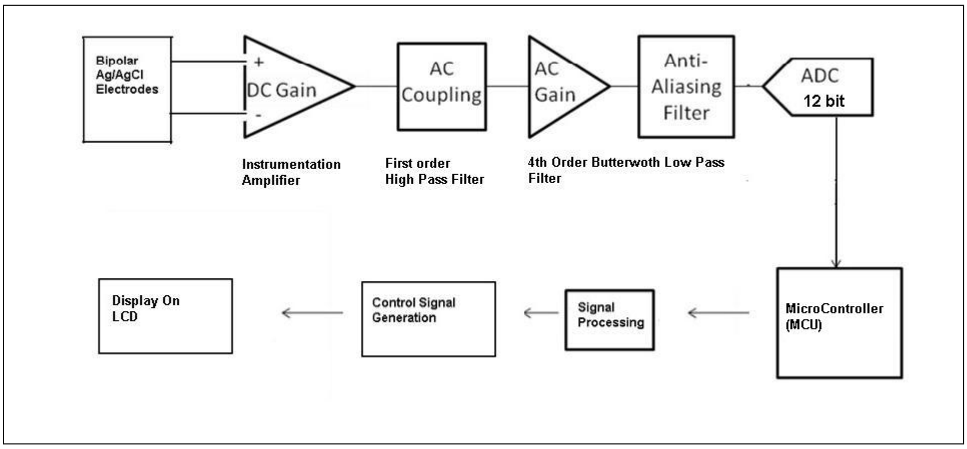

Bipolar Ag/AgCl Electrodes
+
DC Gain
-
Instrumentation Amplifier
AC Coupling
First order High Pass Filter
AC Gain
Anti-Aliasing Filter
4th Order Butterwoth Low Pass Filter
ADC 12 bit
MicroController (MCU)
Signal Processing
Control Signal Generation
Display On LCD

Figure 5.1 Block Diagram Of the Acquisition System

Firstly, to detect the brainwaves, Ag-AgCL electrodes were used. Their outputs were coupled into the input terminals of an instrumentation amplifier. Two pairs of electrodes were placed inside a bathing cap over C3 - F3 and C4 - F4 according to 10-20 system to have two EEG channels for SMR rhythms as displayed in Figure 2.4.

The differential amplifier used is the INA333 Texas Instruments Instrumentation Amplifier. This device was chosen because of its high CMRR, low noise, low offset and low power characteristics. Internal structure of INA333 is shown in Appendix C. Common mode rejection ratio versus Frequency characteristics is also shown in Figure 5.2.

OPA333 Texas Instruments operation amplifier is used for further filtering and amplification stages. OPA333 is a microPower operational amplifier with low voltage, low noise and single supply operation from ZeroDrift Series.

After the acquisition and first level amplification of the signal, it must be filtered to attenuate unwanted frequency bands. This was done with a 4th order Butterworth Sallen-Key low pass filter with a cut-off frequency at 30 Hz which was designed with Texas Instruments FilterPro Analog design software. After filtering and a final stage

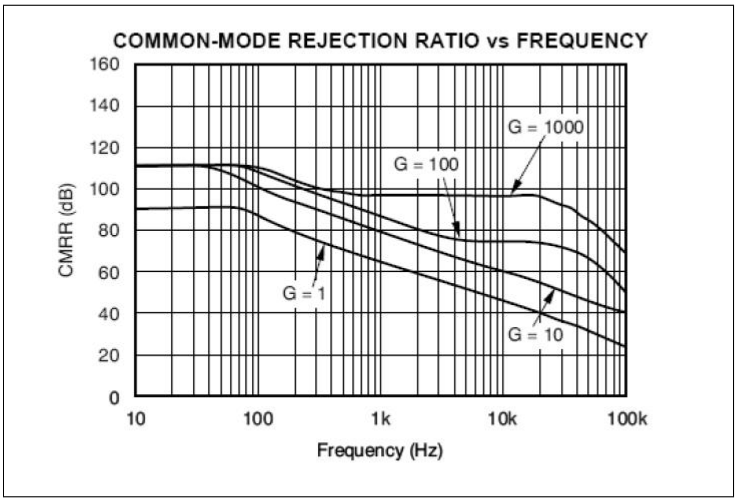


Figure 5.2 CMRR vs frequency, INA333 DataSheet.

amplification, the signal is converted digital domain with an ADC embedded in MSP430 microcontroller. The microcontroller is responsible of both ADC conversion and signal processing.

This analog circuit was firstly simulated in TINA-TI software and then implemented on breadboard.

5.1.1 Instrumentation Stage

INA333 Instrumentation amplifier detects and amplifies the difference between two electrodes connected to its terminals. The electrode positions can be C3 and C4. Since the offset voltages put some DC content into the signal, the gain of the first amplifier is low (G=10) not to amplify the DC component. After INA stage, a first order high pass filter with a cut off frequency of 0.16 Hz is included to reject the DC component for AC coupling. Instrumentation amplifier stage can be seen in Figure 5.3.

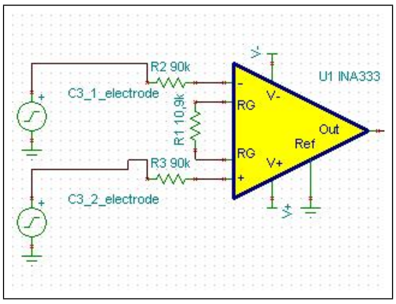


Figure 5.3 First Stage : Instrumentation Amplifier, INA333.

5.1.2 Filtering

For anti-aliasing and detecting a clean EEG signal, frequency components above 30 Hz must be filtered. A fourth order Butterworth Sallen Key Active low pass filter was implemented for that purpose. The Butterworth filter has a flat response in the passband and fast and sharp attenuation in the stop band. In our case, the most important unwanted frequency is 50 Hz mains signal, and the designed cutoff frequency is 30 Hz with a gain of 40dB. Since the order of the Butterworth filter determines the sharpness of the stop band, the higher the order of the filter, the steeper the slope of the stop band. Fourth order is good enough to cancel the 50 Hz mains signal interference. Filter was simulated with a Spice simulation tool (TINA-TI Spice Simulation Software) [48]. The schematic of the filter circuit can be seen in Figure 5.4.

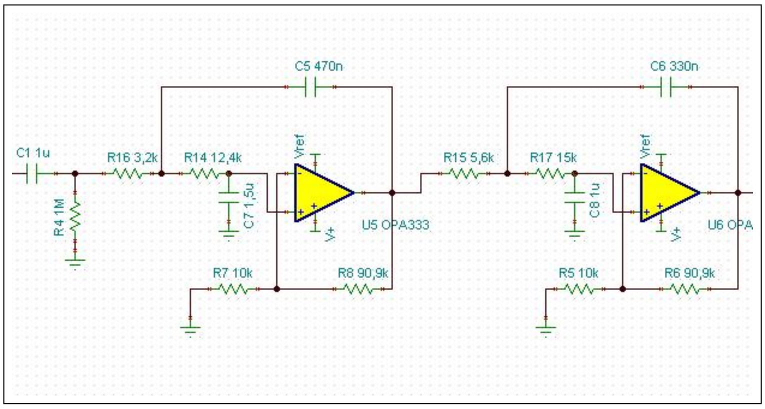


Figure 5.4 Fourth Order Sallen Key Butterworth Filter with a first order high pass filter before input.

Bode Plot of the designed filter can also be seen in Figure 5.5.

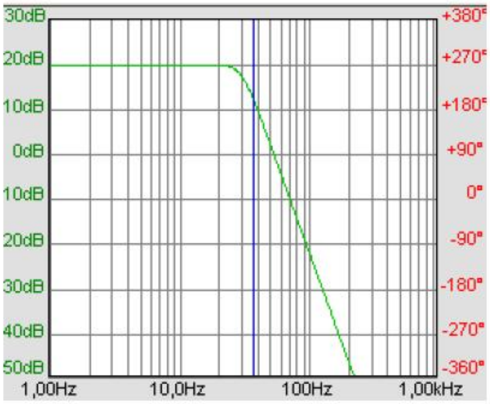


Figure 5.5 Frequency vs Gain (Bode plot) curve of the designed Filter Butterworth Low pass filter

The gain of the final amplification stage was designed to increase the signal level to volts for ADC. The gain of the final amplification stage is 10.

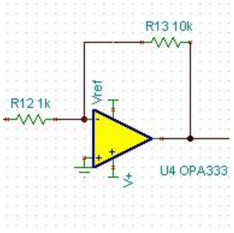


Figure 5.6 Final Amplification Stage

$$G = INAGain(10) * ActiveFilterGain(100) * FinalStageGain(10) = 10000 = 80dB \tag{5.1}$$

After filtering and a final stage amplification, the signal is converted into digital domain with an ADC embedded in MSP430 microcontroller. After the conversion, the processing method described described in the section 4.4 is implemented on the microcontroller. The complete schemantic of the analog circuit implemented can be seen in Appendix A.

Using a function generator, the input was varied between high and low frequencies to determine if the circuit was only passing the required low frequencies. Using a oscilloscope, the output of the circuit was measured. The Butterworh filter successfully filtered out the signals above 30 Hz. Low noise and high precision properties of Texas Instruments ICs helped the detection and amplification.

5.2 Signal Processing Software

The block diagram of the used processor MSP430FG4618 is shown in Figure 5.7.

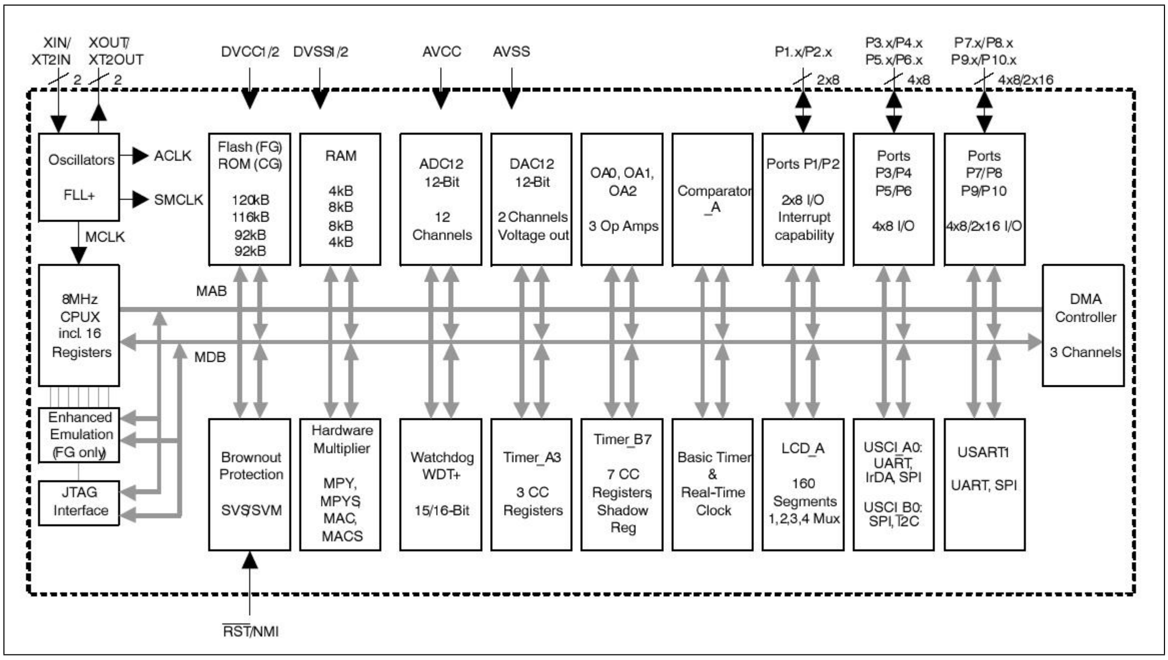

XIN/ XT2IN
XOUT/ XT2OUT
2
2
DVCC1/2
DVSS1/2
AVCC
AVSS
P1.x/P2.x
2x8
P3.x/P4.x P5.x/P6.x
4x8
P7.x/P8.x P9.x/P10.x
4x8/2x16
Oscillators FLL+
ACLK
SMCLK
MCLK
Flash (FG) ROM (CG) 120kB 116kB 92kB 92kB
RAM 4kB 8kB 8kB 4kB
ADC12 12-Bit 12 Channels
DAC12 12-Bit 2 Channels Voltage out
OA0, OA1, OA2 3 Op Amps
Comparator _A
Ports P1/P2 2x8 I/O Interrupt capability
Ports P3/P4 P5/P6 4x8 I/O
Ports P7/P8 P9/P10 4x8/2x16 I/O
8MHz CPUX incl. 16 Registers
MAB
MDB
DMA Controller 3 Channels
Enhanced Emulation (FG only)
JTAG Interface
Brownout Protection SVS/SVM
Hardware Multiplier MPY, MPYS MAC, MACS
Watchdog WDT+ 15/16-Bit
Timer_A3 3 CC Registers
Timer_B7 7 CC Registers Shadow Reg
Basic Timer & Real-Time Clock
LCD_A 160 Segments 1,2,3,4 Mux
USCI_A0: UART, IrDA, SPI USCI_B0: SPI, I2C
USART1 UART, SPI
RST/NMI

Figure 5.7 Block Diagram of The MSP430FG4618 microcontroller

After the amplification stage outputs of two channels are used as inputs to ADC inputs to the MCU. C3 channel is entered into A0 and C4 channel is entered into A1 analog input pins of the microcontroller. Since the supply voltages of the analog board are (-1.5 V, 1.5V), with internal +1.5V reference voltage of the MCU, an external -1.5V reference is used for conversion. The complete digital board of the system can be seen in Appendix B.

Texas Instruments MSP430FG461x/F20xx Experimenter's Board is used with the system since it has many peripherals needed for this project. Board includes an LCD display, two push buttons, 3 LEDs and headers / jumpers to access the pins of the microcontroller. Picture of the board can be seen in Figure 5.8.

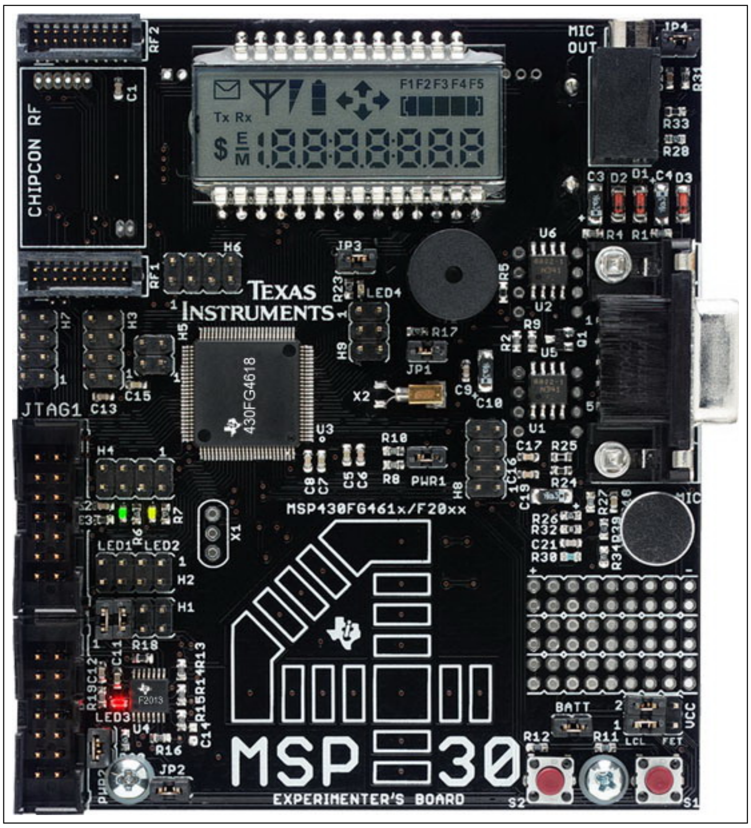


Figure 5.8 Picture of MSP430FG461x/F20xx Experimenter's Board

OPA333 and INA333 assembly packages were soldered onto small circuit boards for insertion onto the breadboard. Picture of the complete circuit is presented in Figure 5.9.

5.2.1 Software Description

The flowchart of the developed software can be seen in Figure 5.10.

After Power on Reset, code initializes the peripherals; Main clock freq, ADC, LCD registers and Digital I/O ports. Writes 'Hello' to LCD registers and enters low

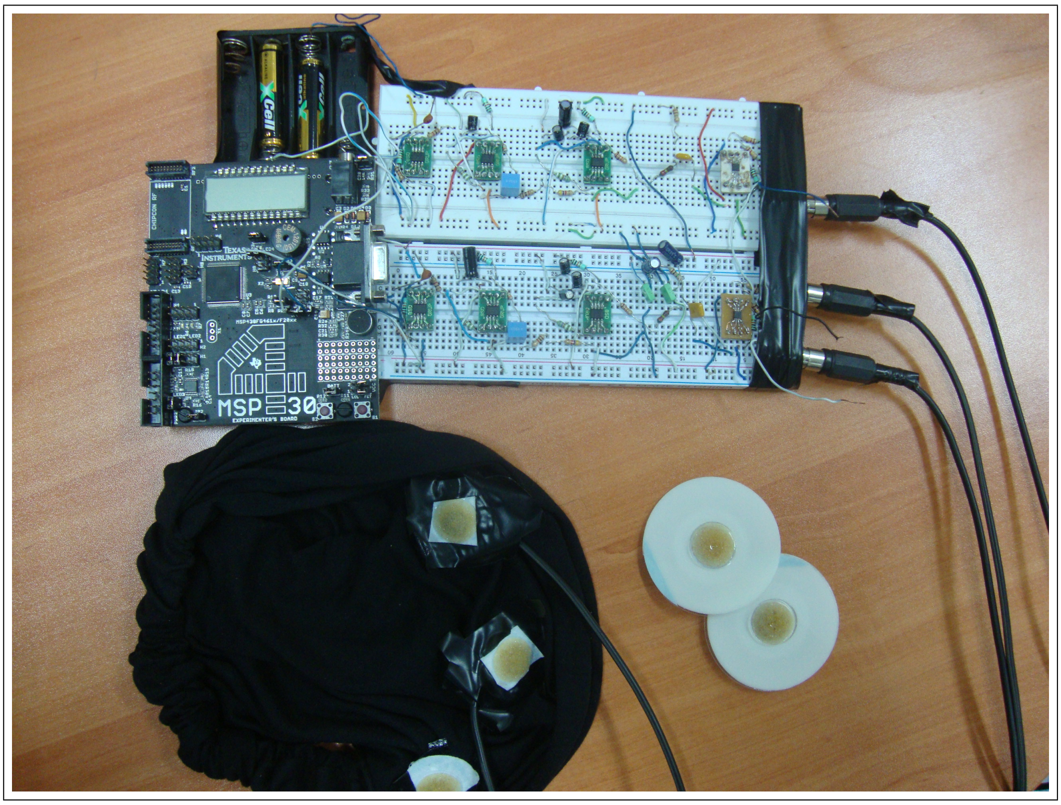


Figure 5.9 Picture of the complete circuit implemented in the lab.

power mode 3 (CPU Off / 32 kHz ACLK On, 1.3 μ A) waiting for interrupts. ADC uses TimerA triggering, +1.5V internal, -1.5V external references which are the supply ranges of instrumentation and operational amplifiers in the same time. CPU operates with 3V battery, so with a voltage divider, supply voltage and reference values are generated. When Button 1 is pressed, Port1 Interrupt wakes up the CPU an in the Port1 Interrupt Service Routine Basic timer is initialized to give about 1 second delay while displaying 'Ready' on LCD. After servicing the ISR the cpu automatically enters back to LPM3 awakened by Basic timer interrupt after 1 second. 'Start' is written to the LCD in the Basic timer interrupt Routine and TimerA is initialized to start the ADC sampling with 128Hz sampling rate. After 448 samples are stored (448 comes from memory limitations and it means 3.5 seconds sampling) TimerA is stopped and stored values are filtered. 4 features are calculated and scaled for neural network input. After neural net classification results are displayed on LCD, then 'Hello' is displayed again and system enters back to low power mode. The complete source code is provided in Appendix D.

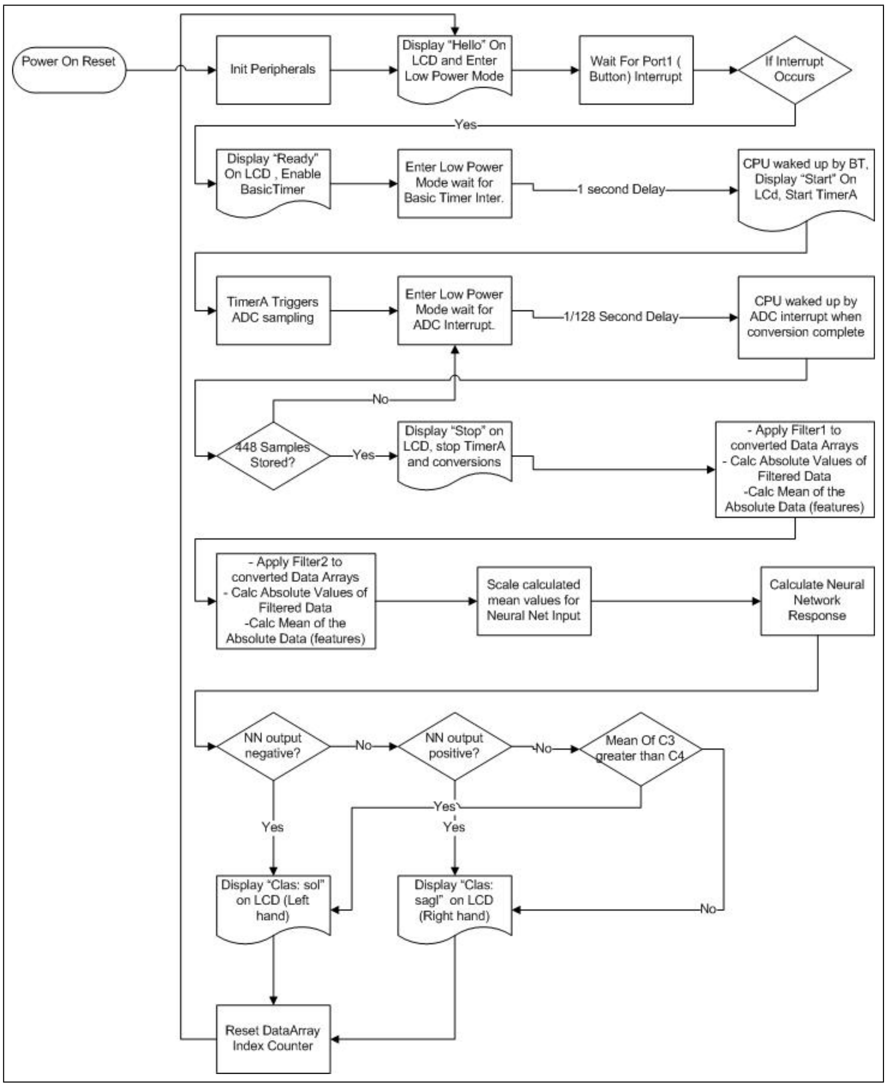

Power On Reset
Init Peripherals
Display "Hello" On LCD and Enter Low Power Mode
Wait For Port1 (Button) Interrupt
If Interrupt Occurs
Yes
Display "Ready" On LCD , Enable BasicTimer
Enter Low Power Mode wait for Basic Timer Inter.
1 second Delay
CPU waked up by BT, Display "Start" On LCd, Start TimerA
TimerA Triggers ADC sampling
Enter Low Power Mode wait for ADC Interrupt.
1/128 Second Delay
CPU waked up by ADC interrupt when conversion complete
No
448 Samples Stored?
Yes
Display "Stop" on LCD, stop TimerA and conversions
- Apply Filter1 to converted Data Arrays - Calc Absolute Values of Filtered Data -Calc Mean of the Absolute Data (features)
- Apply Filter2 to converted Data Arrays - Calc Absolute Values of Filtered Data -Calc Mean of the Absolute Data (features)
Scale calculated mean values for Neural Net Input
Calculate Neural Network Response
NN output negative?
No
NN output positive?
No
Mean Of C3 greater than C4
Yes
Yes
Yes
Display "Clas: sol" on LCD (Left hand)
Display "Clas: sagl" on LCD (Right hand)
No
Reset DataArray Index Counter

Figure 5.10 The flowchart of the developed software

5.3 Measurements taken from the Developed Hardware

An EEG Simulator from Netech company [49] was used to test EEG acquisition system. The compact, microcontroller based instrument has five separate floating outputs and simulates Alpha-Beta Rhythm ABR, Sine, Square and Triangle waveforms with selectable frequencies and amplitudes. Outputs of the amplifier channels were measured via a digital storage oscilloscope and transferred to a PC for permanent storage. The complete measurement system can be seen in Figure 5.11.

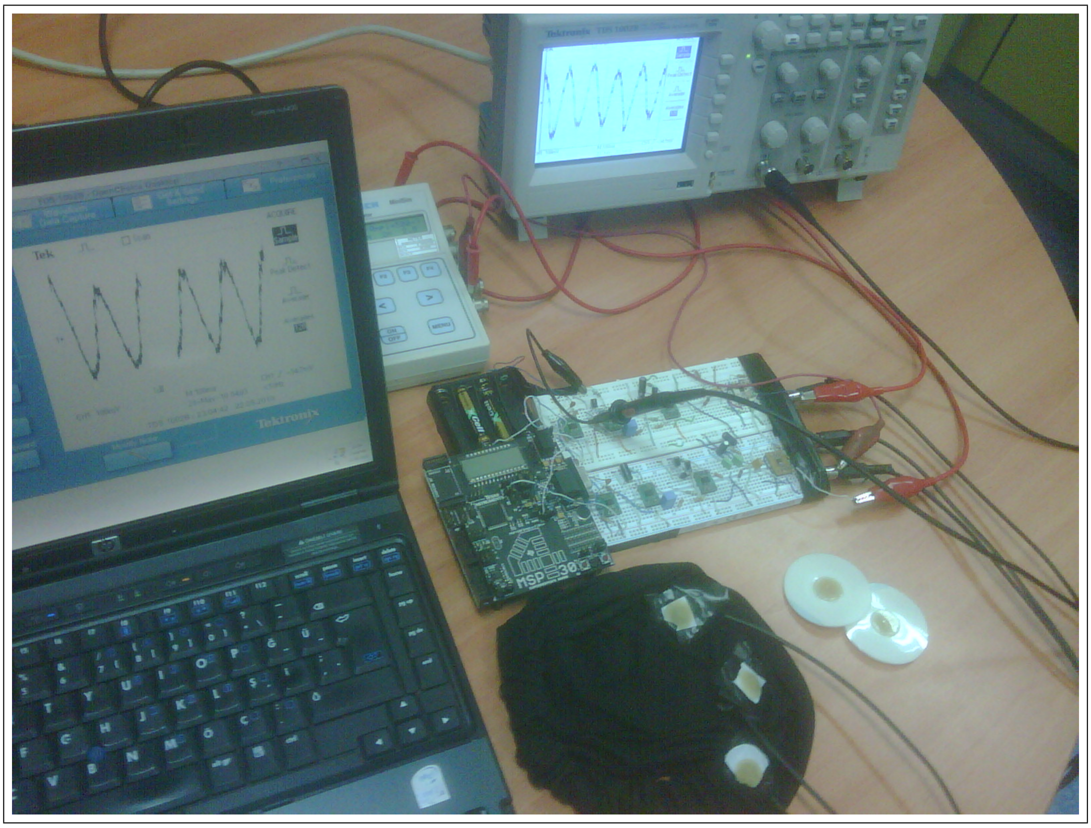


Figure 5.11 Picture of the complete measurement system.

The developed system was tested with the EEG simulator by applying sine and square waves with 2 mV, 500 μ V and 100 μ V amplitudes with 2 and 5 Hz frequencies to prove the system is filtering and amplifying the signal as needed. The outputs of the Instrumentation amplifier, the Butterworth filter and the complete circuit are provided for each EEG simulator input to see the filtering and amplification effects in Figures between 5.12 and 5.47. After providing simulator responses, measurements taken from human subjects are provided in Figures between 5.48 and 5.55. The classification output of the system is based on the method described in Section 4.4.

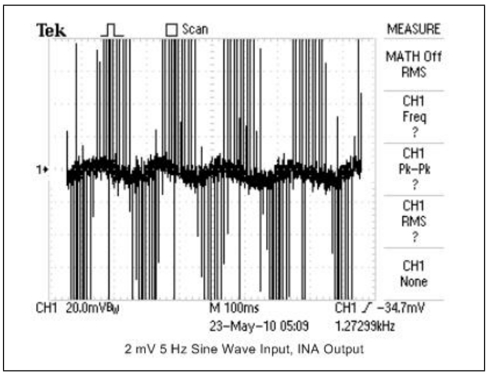


Figure 5.12 Instrumentation Amplifier output of the circuit. Input 2mV amplitude, 5Hz Sine wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

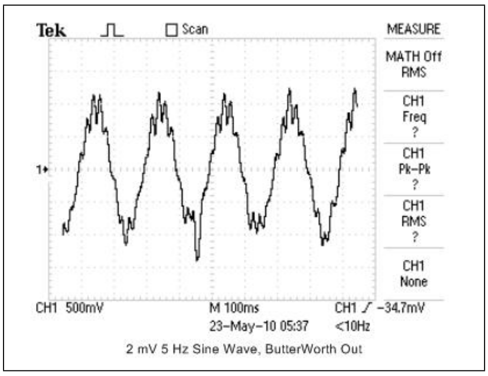


Figure 5.13 Butterworth filter output of the circuit. Input 2mV amplitude, 5Hz Sine wave. Vertical divisions 500mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components.

There are three measurements for each input. First one is the instrumentation amplifier output, second one is the Butterworth Filter output and the last one is the output of the complete circuit. Figure 5.12 presents the instrumentation amplifier output of the circuit with 2mV 5Hz Sine wave input. Output includes both the input signal with INA gain and dc components as noise. Figure 5.13 presents the Butterworth Filter output of the circuit with 2mV 5Hz Sine wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected.

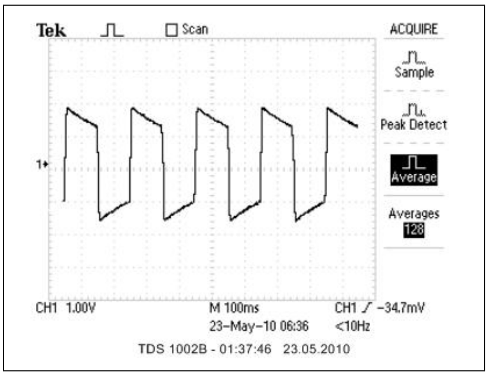


Figure 5.14 Output of the circuit. Input 2mV amp, 5Hz Sine wave. Vertical divisions 1V,horizontal divisions 100ms.Total gain of 10000 can be seen with saturations because of 3V supply voltage.

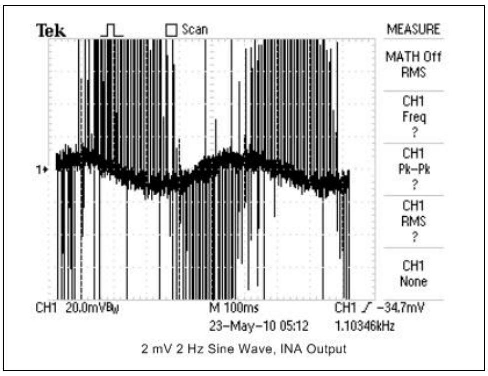


Figure 5.15 Instrumentation Amplifier output of the circuit. Input 2mV amplitude, 2Hz Sine wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

Figure 5.14 presents the output of the complete circuit with 2mV 5Hz Sine wave input. Total gain of 10000 (80dB) can be seen with saturations because of 3V supply voltage. Figure 5.15 presents the instrumentation amplifier output of the circuit with 2mV 2Hz Sine wave input. Output includes both the input signal with INA gain and dc components as noise.

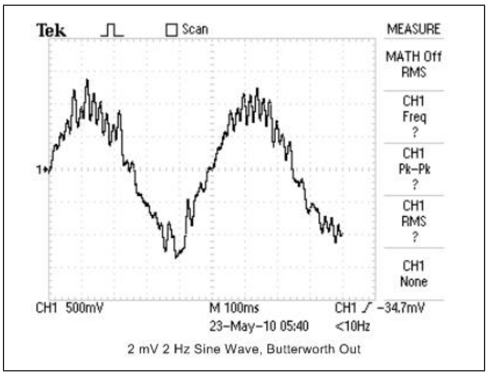


Figure 5.16 Butterworth filter output of the circuit. Input 2mV amplitude, 2Hz Sine wave. Vertical divisions 500mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components.

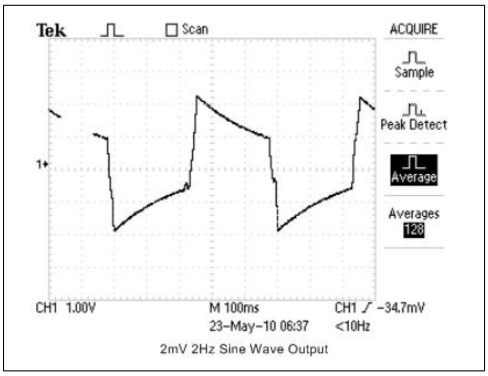


Figure 5.17 Output of the circuit. Input 2mV amplitude, 2Hz Sine wave. Vertical divisions 1V, horizontal divisions 100ms. Total gain of 10000 can be seen with saturations because of 3 V supply voltage.

Figure 5.16 presents the Butterworth Filter output of the circuit with 2mV 2Hz Sine wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected. Figure 5.17 presents the output of the complete circuit with 2mV 2Hz Sine wave input. Total gain of 10000 (80dB) can be seen with saturations because of 3V supply voltage.

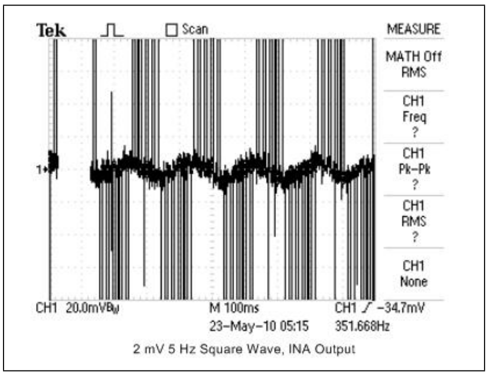


Figure 5.18 Instrumentation Amplifier output of the circuit. Input 2mV amplitude, 5Hz Square wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

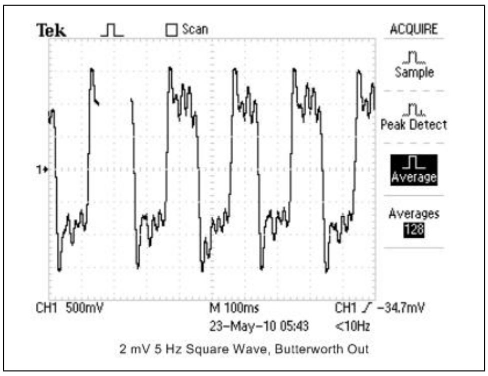


Figure 5.19 Butterworth filter output of the circuit. Input 2mV amplitude, 5Hz Square wave. Vertical divisions 500mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components.

Figure 5.18 presents the instrumentation amplifier output of the circuit with 2mV 5Hz Square wave input. Output includes both the input signal with INA gain and dc components as noise. Figure 5.19 presents the Butterworth Filter output of the circuit with 2mV 5Hz Square wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected.

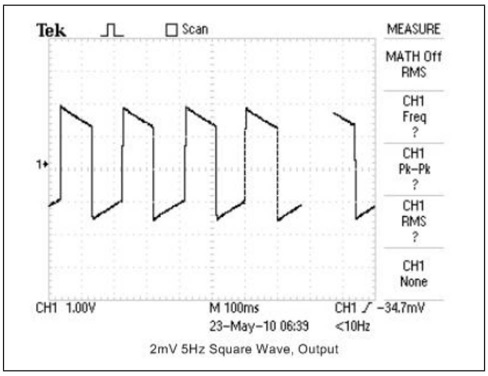


Figure 5.20 Output of the circuit. Input 2mV amplitude, 5Hz Square wave. Vertical divisions 1V, horizontal divisions 100ms. Total gain of 10000 can be seen with saturations because of 3 V supply voltage.

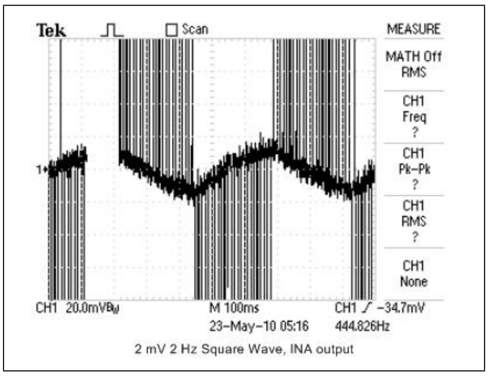


Figure 5.21 Instrumentation Amplifier output of the circuit. Input 2mV amplitude, 2Hz Square wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

Figure 5.20 presents the output of the complete circuit with 2mV 5Hz Square wave input. Total gain of 10000 (80dB) can be seen with saturations because of 3V supply voltage. Figure 5.21 presents the instrumentation amplifier output of the circuit with 2mV 2Hz Square wave input. Output includes both the input signal with INA gain and dc components as noise.

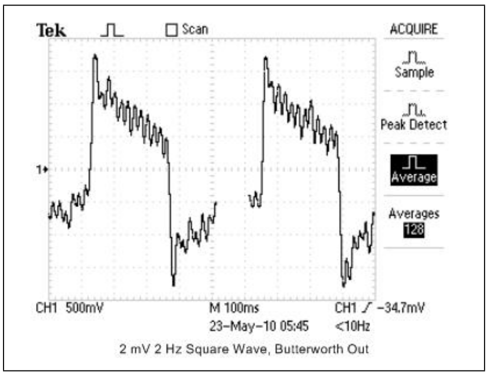


Figure 5.22 Butterworth filter output of the circuit. Input 2mV amplitude, 2Hz Square wave. Vertical divisions 500mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components.

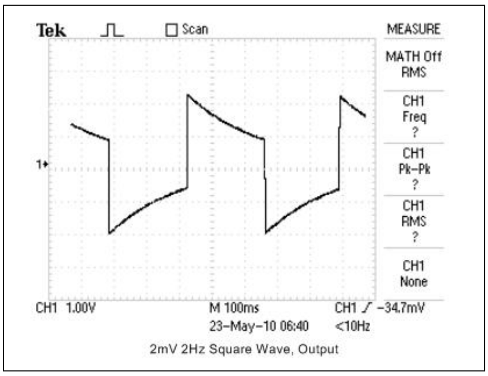


Figure 5.23 Output of the circuit. Input 2mV amplitude, 2Hz Square wave. Vertical divisions 1V, horizontal divisions 100ms. Total gain of 10000 can be seen with saturations because of 3 V supply voltage.

Figure 5.22 presents the Butterworth Filter output of the circuit with 2mV 2Hz Square wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected. Figure 5.23 presents the output of the complete circuit with 2mV 2Hz Square wave input. Total gain of 10000 (80dB) can be seen with saturations because of 3V supply voltage.

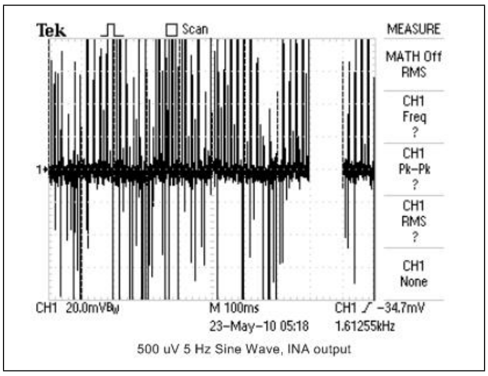


Figure 5.24 Instrumentation Amplifier output of the circuit. Input 500uV amplitude, 5Hz Sine wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

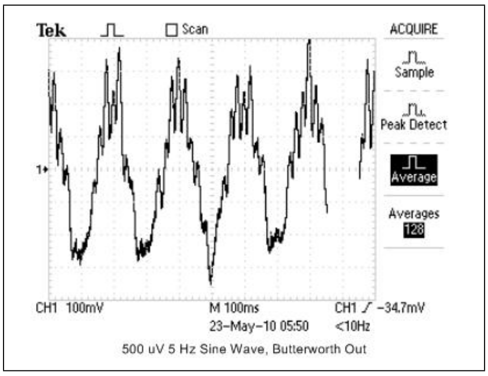


Figure 5.25 Butterworth filter output of the circuit. Input 500uV amplitude, 5Hz Sine wave. Vertical divisions 100mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components.

Figure 5.24 presents the instrumentation amplifier output of the circuit with 500uV 5Hz Sine wave input. Output includes both the input signal with INA gain and dc components as noise. Figure 5.25 presents the Butterworth Filter output of the circuit with 500uV 5Hz Sine wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected.

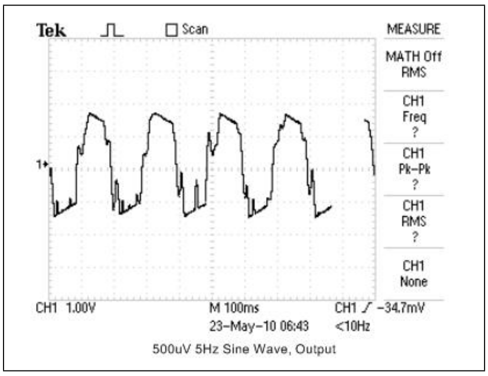


Figure 5.26 Output of the circuit. Input 500uV amplitude, 5Hz Sine wave. Vertical divisions 1V, horizontal divisions 100ms. Total gain of 10000 can be seen with saturations because of 3 V supply voltage.

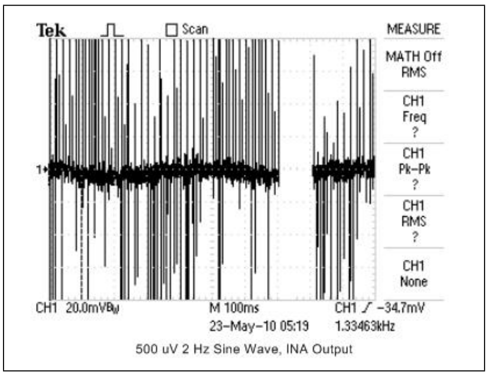


Figure 5.27 Instrumentation Amplifier output of the circuit. Input 500uV amplitude, 2Hz Sine wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

Figure 5.26 presents the output of the complete circuit with 500uV 5Hz Sine wave input. Total gain of 10000 (80dB) can be seen with saturations because of 3V supply voltage. Figure 5.27 presents the instrumentation amplifier output of the circuit with 500uV 2Hz Sine wave input. Output includes both the input signal with INA gain and dc components as noise.

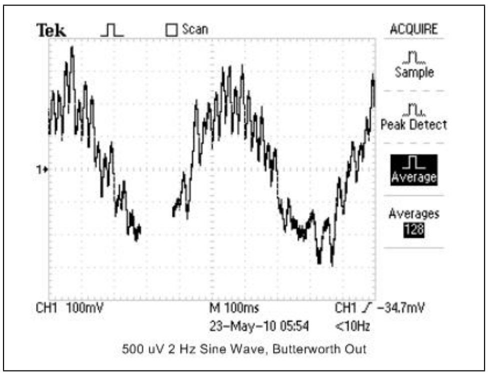


Figure 5.28 Butterworth filter output of the circuit. Input 500uV amplitude, 2Hz Sine wave. Vertical divisions 100mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components.

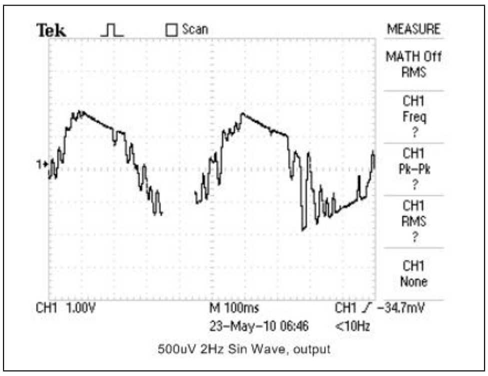


Figure 5.29 Output of the circuit. Input 500uV amplitude, 2Hz Sine wave. Vertical divisions 1V, horizontal divisions 100ms. Total gain of 10000 can be seen with saturations because of 3 V supply voltage.

Figure 5.28 presents the Butterworth Filter output of the circuit with 500uV 2Hz Sine wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected. Figure 5.29 presents the output of the complete circuit with 500uV 2Hz Sine wave input. Total gain of 10000 (80dB) can be seen with saturations because of 3V supply voltage.

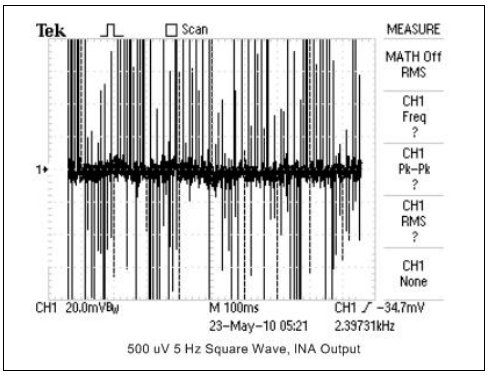


Figure 5.30 Instrumentation Amplifier output of the circuit. Input 500uV amplitude, 5Hz Square wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

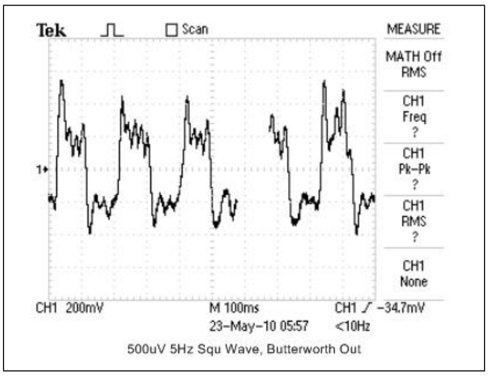


Figure 5.31 Butterworth filter output of the circuit. Input 500uV amplitude, 5Hz Square wave. Vertical divisions 200mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components.

Figure 5.30 presents the instrumentation amplifier output of the circuit with 500uV 5Hz Square wave input. Output includes both the input signal with INA gain and dc components as noise. Figure 5.31 presents the Butterworth Filter output of the circuit with 500uV 5Hz Square wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected.

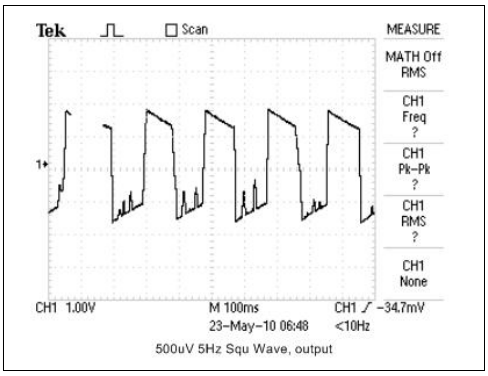


Figure 5.32 Output of the circuit. Input 500uV amplitude, 5Hz Square wave. Vertical divisions 1V, horizontal divisions 100ms. Total gain of 10000 can be seen with saturations because of 3 V supply voltage.

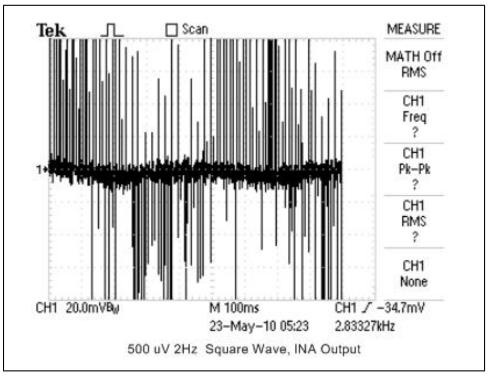


Figure 5.33 Instrumentation Amplifier output of the circuit. Input 500uV amplitude, 2Hz Square wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

Figure 5.32 presents the output of the complete circuit with 500uV 5Hz Square wave input. Total gain of 10000 (80dB) can be seen with saturations because of 3V supply voltage. Figure 5.33 presents the instrumentation amplifier output of the circuit with 500uV 2Hz Square wave input. Output includes both the input signal with INA gain and dc components as noise.

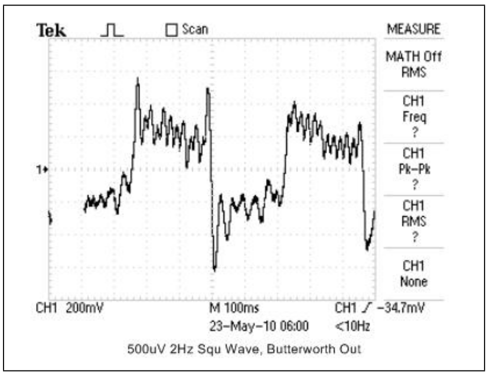


Figure 5.34 Butterworth filter output of the circuit. Input 500uV amplitude, 2Hz Square wave. Vertical divisions 200mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components.

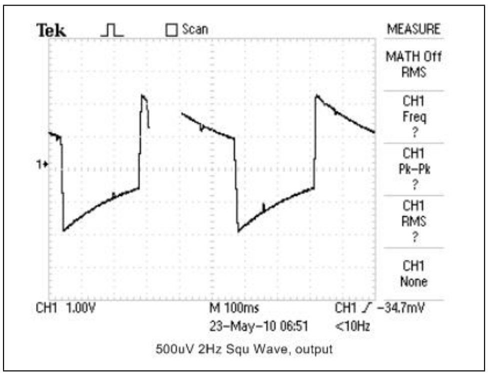


Figure 5.35 Output of the circuit. Input 500uV amplitude, 2Hz Square wave. Vertical divisions 1V, horizontal divisions 100ms. Total gain of 10000 can be seen with saturations because of 3 V supply voltage.

Figure 5.34 presents the Butterworth Filter output of the circuit with 500uV 2Hz Square wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected. Figure 5.35 presents the output of the complete circuit with 500uV 2Hz Square wave input. Total gain of 10000 (80dB) can be seen with saturations because of 3V supply voltage.

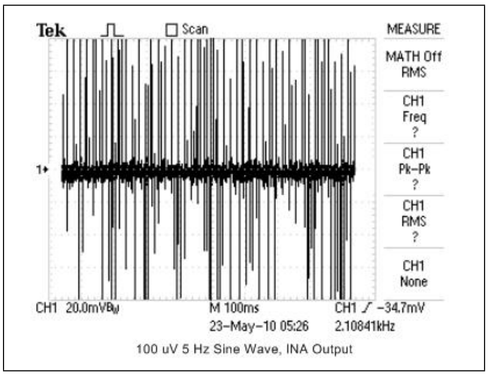


Figure 5.36 Instrumentation Amplifier output of the circuit. Input 100uV amplitude, 5Hz Sine wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

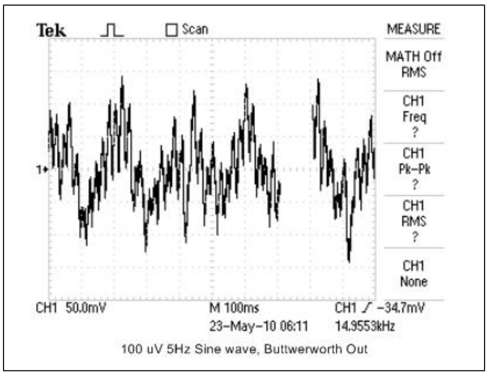


Figure 5.37 Butterworth filter output of the circuit. Input 100uV amplitude, 5Hz Sine wave. Vertical divisions 50mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components.

Figure 5.36 presents the instrumentation amplifier output of the circuit with 100uV 5Hz Sine wave input. Output includes both the input signal with INA gain and dc components as noise. Figure 5.37 presents the Butterworth Filter output of the circuit with 100uV 5Hz Sine wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected.

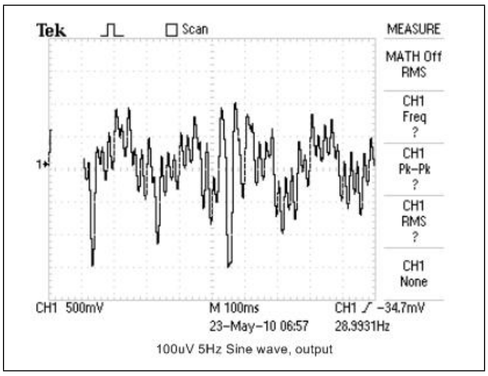


Figure 5.38 Output of the circuit. Input 100uV amplitude, 5Hz Sine wave. Vertical divisions 500mV, horizontal divisions 100ms. Total gain of 10000 can be seen with a motion artifact in signal caused by capturing from oscilloscope.

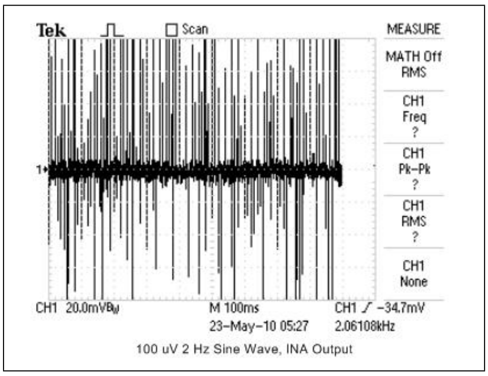


Figure 5.39 Instrumentation Amplifier output of the circuit. Input 100uV amplitude, 2Hz Sine wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

Figure 5.38 presents the output of the complete circuit with 100uV 5Hz Sine wave input. Total gain of 10000 (80dB) can be seen with a motion artifact caused by pressing a key to capture from oscilloscope can be seen in the waveform. Figure 5.39 presents the instrumentation amplifier output of the circuit with 100uV 2Hz Sine wave input. Output includes both the input signal with INA gain and dc components as noise.

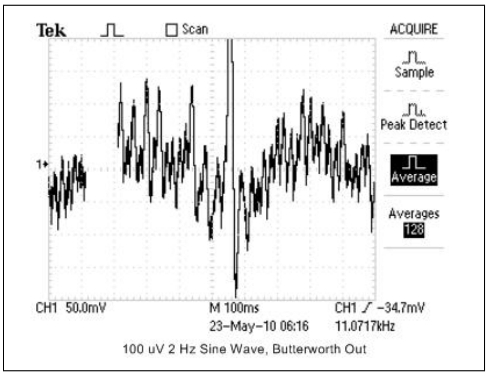


Figure 5.40 Butterworth filter output of the circuit. Input 100uV amplitude, 2Hz Sine wave. Vertical divisions 50mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components and a motion artifact caused by capturing from oscilloscope.

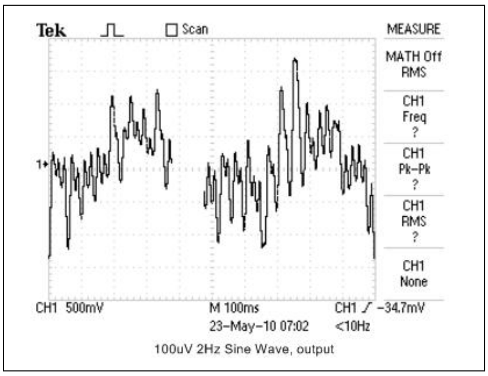


Figure 5.41 Output of the circuit. Input 100uV amplitude, 2Hz Sine wave. Vertical divisions 500mV, horizontal divisions 100ms. Total gain of 10000 can be seen.

Figure 5.40 presents the Butterworth Filter output of the circuit with 100uV 2Hz Sine wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected. Figure 5.41 presents the output of the complete circuit with 100uV 2Hz Sine wave input. Total gain of 10000 (80dB) can be seen in the waveform with no saturation because the input signal is in the acceptable range.

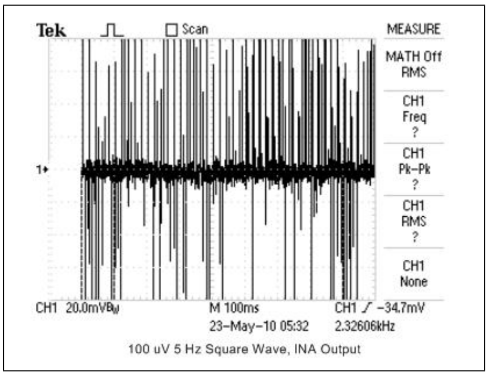


Figure 5.42 Instrumentation Amplifier output of the circuit. Input 100uV amplitude, 5Hz Square wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

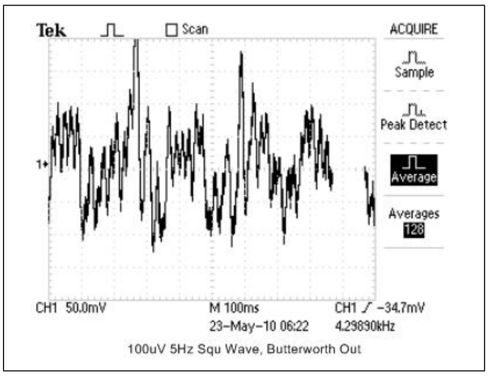


Figure 5.43 Butterworth filter output of the circuit. Input 100uV amplitude, 5Hz Square wave. Vertical divisions 50mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components.

Figure 5.42 presents the instrumentation amplifier output of the circuit with 100uV 5Hz Square wave input. Output includes both the input signal with INA gain and dc components as noise. Figure 5.43 presents the Butterworth Filter output of the circuit with 100uV 5Hz Square wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected.

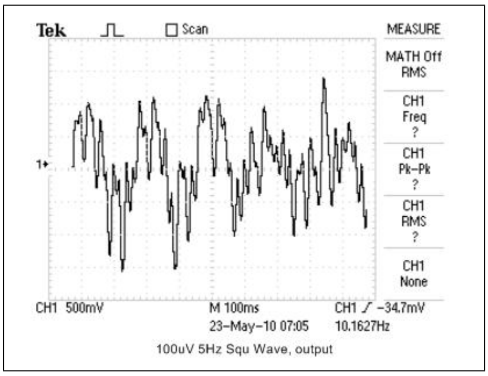


Figure 5.44 Output of the circuit. Input 100uV amplitude, 5Hz Square wave. Vertical divisions 500mV, horizontal divisions 100ms. Total gain of 10000 can be seen.

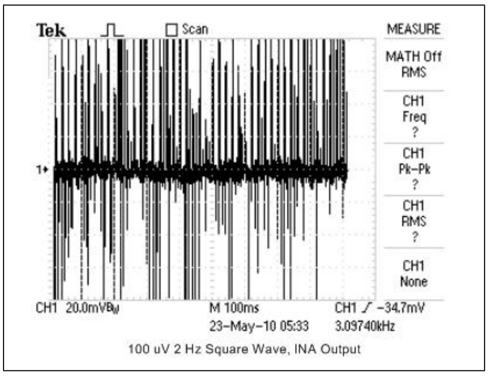
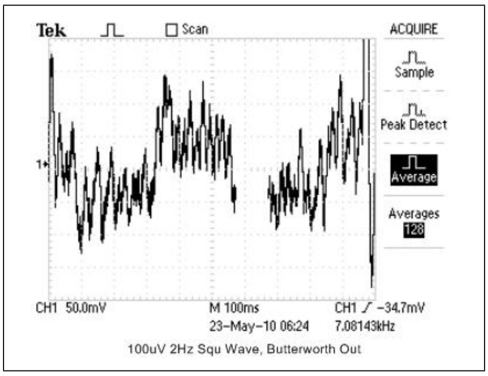


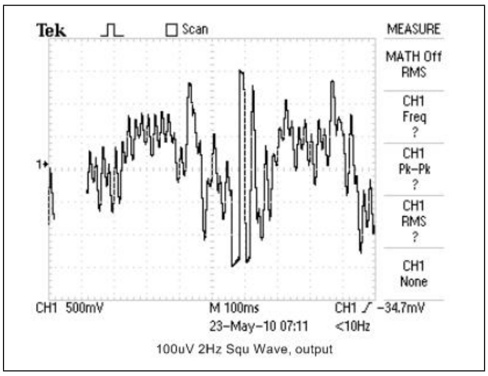
Figure 5.45 Instrumentation Amplifier output of the circuit. Input 100uV amplitude, 2Hz Square wave. Vertical divisions 20mV, horizontal divisions 100ms. INA gain of 10 can be seen with dc components as noise.

Figure 5.44 presents the output of the complete circuit with 100uV 5Hz Square wave input. Total gain of 10000 (80dB) can be seen in the waveform. Figure 5.45 presents the instrumentation amplifier output of the circuit with 100uV 2Hz Square wave input. Output includes both the input signal with INA gain and dc components as noise.



CH1 50.0mV M 100ms CH1 ∫ −34.7mV
23−May−10 06:24 7.08143kHz
100uV 2Hz Squ Wave, Butterworth Out

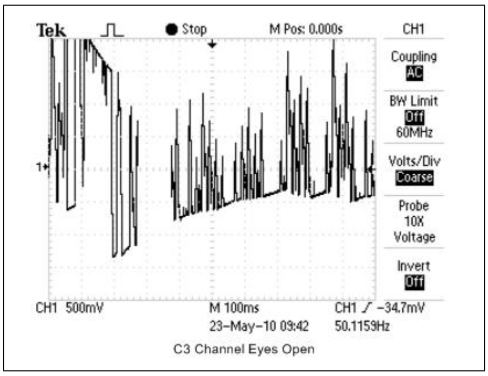
Figure 5.46 Butterworth filter output of the circuit. Input 100uV amplitude, 2Hz Square wave. Vertical divisions 50mV, horizontal divisions 100ms. INA + But Filter gain of 1000 can be seen with rejected high and low frequency noise components.



CH1 500mV M 100ms CH1 ∫ −34.7mV
23−May−10 07:11 <10Hz
100uV 2Hz Squ Wave, output

Figure 5.47 Output of the circuit. Input 100uV amplitude, 2Hz Square wave. Vertical divisions 500mV, horizontal divisions 100ms. Total gain of 10000 can be seen with a motion artifact caused by capturing from oscilloscope.

Figure 5.46 presents the Butterworth Filter output of the circuit with 100uV 2Hz Square wave input. The total gain of 1000 of INA and Butterworth Filter stages can be seen with above 30Hz and below 0.16Hz frequency components rejected. Figure 5.47 presents the output of the complete circuit with 100uV 2Hz Sqaure wave input. Total gain of 10000 (80dB) can be seen with a motion artifact and no saturation because the input signal is in the acceptable range.



● Stop M Pos: 0.000s

CH1

Coupling AC

BW Limit Off 60MHz

Volts/Div Coarse

Probe 10X Voltage

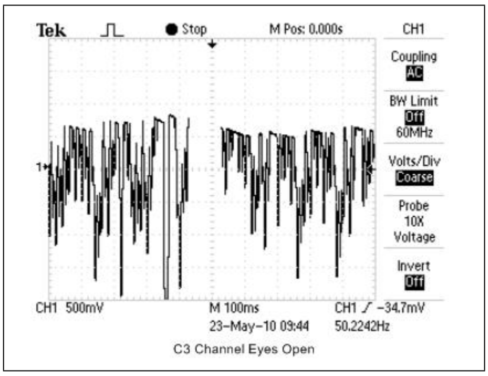
Invert Off

CH1 500mV M 100ms CH1 ∫ −34.7mV

23–May–10 09:42 50.1159Hz

C3 Channel Eyes Open

Figure 5.48 C3 channel measurement from a human subject. Eyes Open.



● Stop M Pos: 0.000s

CH1

Coupling AC

BW Limit Off 60MHz

Volts/Div Coarse

Probe 10X Voltage

Invert Off

CH1 500mV M 100ms CH1 ∫ −34.7mV

23–May–10 09:44 50.2242Hz

C3 Channel Eyes Open

Figure 5.49 C3 channel measurement from a human subject. Eyes Open.

Following figures present measurements taken from human subjects with the developed hardware in the laboratory. Figure 5.48 and 5.49 represent C3 channel measurements from a human subject with eyes open.

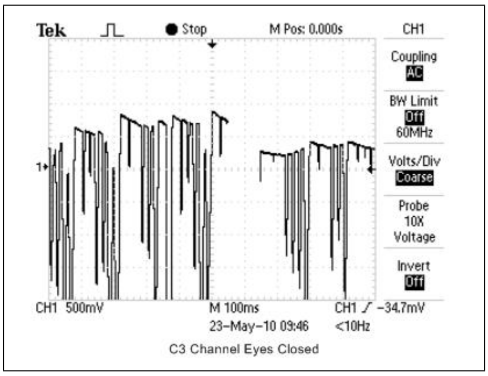


Figure 5.50 C3 channel measurement from a human subject. Eyes Closed.

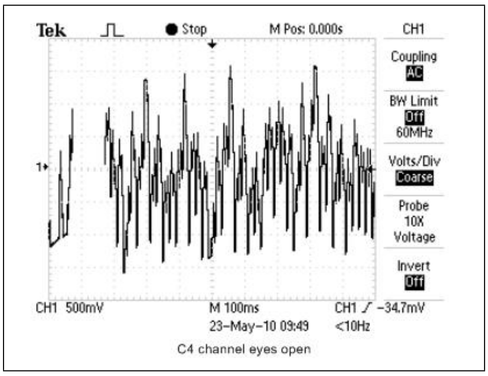


Figure 5.51 C4 channel measurement from a human subject. Eyes Open.

Figure 5.50 presents a waveform from a humen subject with eyes closed. And Figure 5.51 represents C4 channel measurements from a human subject with eyes open.

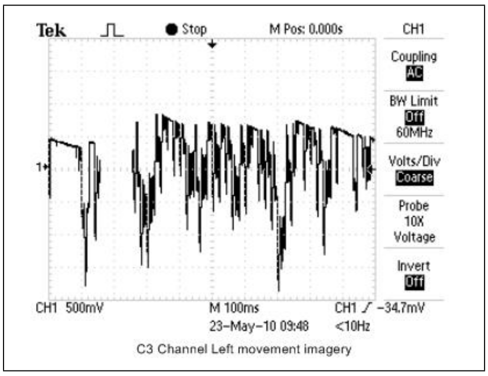


Figure 5.52 C3 channel measurement from a human subject during left hand movement imagery task with a output classification of left hand.

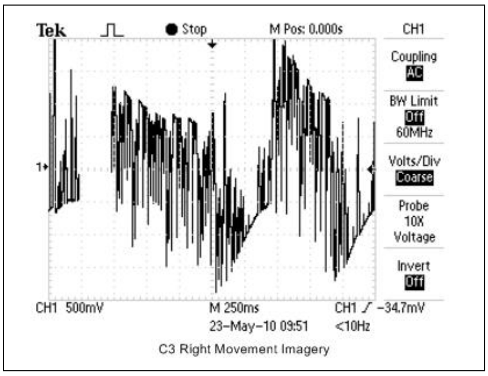


Figure 5.53 C3 channel measurement from a human subject during right hand movement imagery task with a output classification of right hand.

Figures 5.52 and 5.53 are recordings of left and right hand movement imagery tasks.

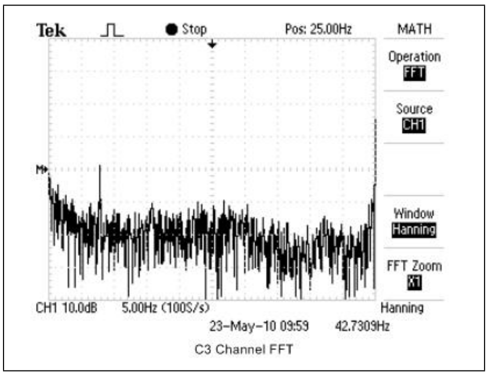


Figure 5.54 C3 channel recording FFT during a left hand movement imagery task with a true classification. Mu rhtyhm power can be seen as 8 - 9 Hz band.

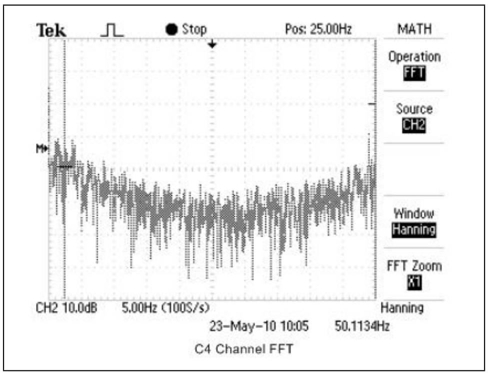


Figure 5.55 C4 channel recording FFT during a left hand movement imagery task with a true classification.

FFTs of the C3 and C4 left hand movement imagery tasks are displayed in Figures 5.54 and 5.55.

5.4 Discussions And Suggestions for the Design

A comparison between the Competition Data set signal of right hand movement imagery and the signal acquired with the developed hardware is provided in Figure 5.56.

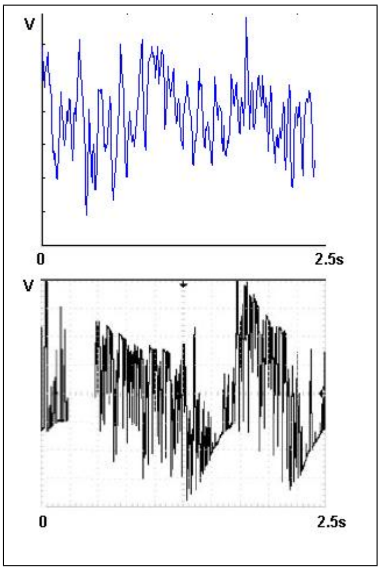


Figure 5.56 Comparison between the Competition Data set signal and the signal measured with the developed hardware of Right Hand movement imagery. Blue one represents the data set signal and black one represents the signal measured in the lab.

The developed method presented in Section 4.4 is successfully implemented with the developed hardware presented in this chapter. The system could be able to detect two channel EEG and provide the classification results as designed. As presented in the results section of the previous chapter, best results were obtained using Wavelet based features with an artificial neural network classification. The classification performance of the FIR band pass filter based approach used in this design can be increased to the levels of Wavelet based approach by implementing the suggestions provided below.

Suggestions:

1. Source code generates 40KB program code and uses 5KB RAM which are not available in many of the members of MSP430 family excluding MSP430X architecture and new MSP430x5xx series.

2. Hardware multiplier and DMA gives DSP like capabilities to FG4618. But many of the MSP430 devices lack MPY or DMA can be very slow on multiplication. Because of that, if a low power system with a MCU is aimed, a device with a MPU should be selected.

3. Different feature extraction bands can be applied by only changing the filter coefficients. ScopeFIR [50] software can be used to design and scale filter coefficients.

4. New features can be extracted by adding new calculations from filtered data such as maximum, minimum or median values.

5. Neural network performance can be tested in a memory free environment. Neural network architecture can be easily customized by changing hidden neuron and learning rate variables.

6. Active electrodes can be used instead of passive Ag-AgCl electodes for permanent usage without applying gel and easy hair penetration.

7. Sigma Delta ADC can be used instead of SAR ADC to reduce the analog board size. Sigma Delta ADCs provide higher resolution with oversampling, hence only an instrumentation amplifier and a first order low pass filter can be used before input to the ADC for each channel. In addition, an analog front end IC may be used instead of discrete components.

6. CONCLUSIONS

Extraction and classification of patterns from spontaneous EEG activity is a complicated task. Since EEG signal contains a lot of noise components such as mains interference, muscular and eye artifacts, filtering of the raw EEG is an important component of this process. Whether an electronic or digital filter used, band pass filtering must be done very smoothly and the filter must have sharp characteristics in the pass band. For a brain computer interface based on motor imagery as in this study, sensorimotor rhytms, especially the mu rhythm has very high importance. Since mu rhythm desynchronizes with the motor imagery, using spatial filtering and combining different channels like C3 and C4 in the analysis, discrimination between motor imagery tasks can be done efficiently.

The most important part of the design is feature extraction. In order to make true classifications, the feature set must provide vital information. Time frequency analysis methods provide decomposition of frequency bands of the signal with time localization. This is very important for non stationary signals like EEG. With the help of time frequency analysis methods like wavelet transform good features are obtained in the desired frequency bands like beta and mu bands.

In this study, discrete wavelet transform was applied to the data set provided for BCI Competition II to extract the features in beta and mu bands. After extracting the features by means of wavelet coefficients, there features were used as inputs to an artificial neural network for motor imagery task classification. After different network and feature combinations, %89 true classification was achieved on test data set. Classification based on power spectrum has lower classification rates. But more efficient features can be computed to increase the level of accuracy.

The most important part of this study is implementation of an efficient algorithm on a microncontroller to implement a low power embedded system. A System was designed uses efficient FIR filters instead of Wavelet sub band decomposition and implemented on the MSP430 microcontroller. Classification rates of this system can be increased with optimized neural network implementations and features.

As future studies, following tasks may be considered:

1. Since the system is low power and embedded it can be assembled on the head of a subject. The commands can be sent wireless.
2. A serial connection with a PC can be done to display a feedback and to implement more powerful algorithms in real time.
3. Wireless module on the experimenter board can be used to transmit commands wirelessly to a robot arm or another effector.
4. A more powerful DSP such as TMS or ARM can be use to use more channels and higher sampling rates to have better resolution.

APPENDIX A. ANALOG BOARD SCHEMANTIC

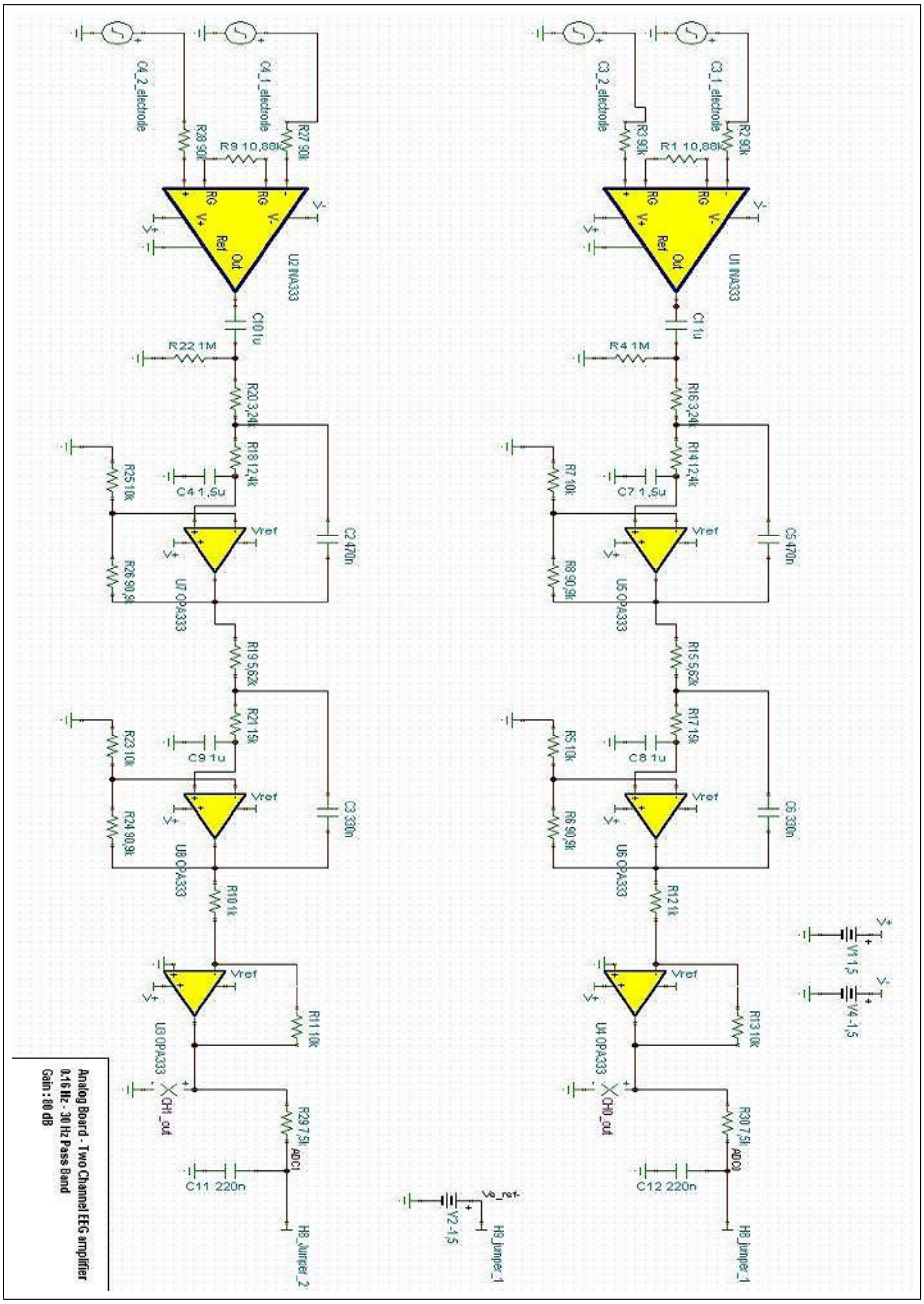


Figure A.1 Analog Board

APPENDIX B. DIGITAL BOARD SCHEMANTIC

The circuit schemantic of TI Experimenter board is presented in Figure B.1.

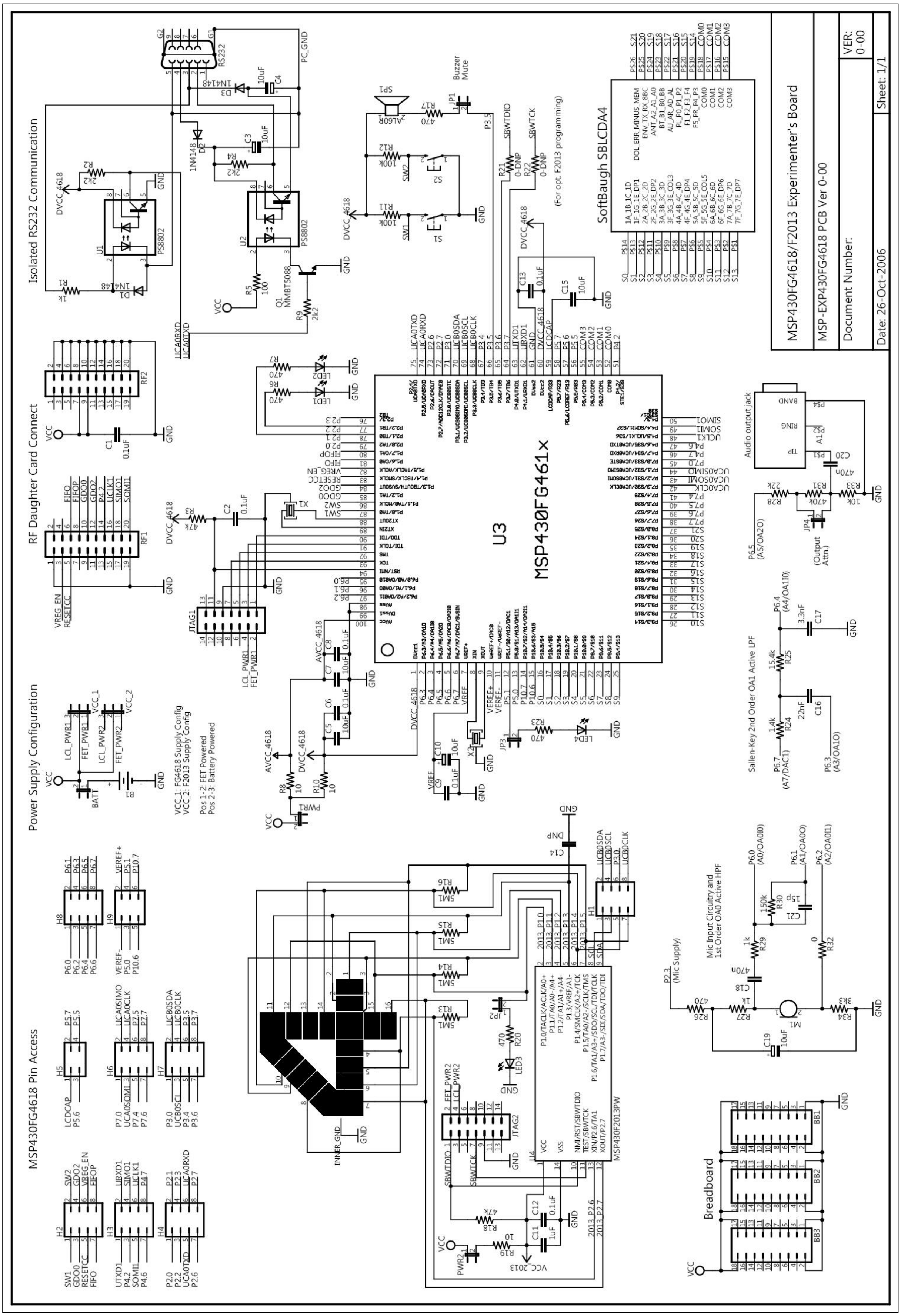

Isolated RS232 Communication
RF Daughter Card Connect
Power Supply Configuration
MSP430FG4618 Pin Access
MSP430FG4618/F2013 Experimenter's Board
MSP-EXP430FG4618 PCB Ver 0-00
Document Number:
Date: 26-Oct-2006
Sheet: 1/1
VER: 0-00
SoftBaugh SBLCDA4
U3
MSP430FG461x
MSP430F2013PW
Breadboard
Sallen-Key 2nd Order OA1 Active LPF
Mic Input Circuitry and 1st Order OA0 Active HPF
Audio output jack
Buzzer Mute
(For opt. F2013 programming)
VCC_1: FG4618 Supply Config
VCC_2: F2013 Supply Config
Pos 1-2: FET Powered
Pos 2-3: Battery Powered
RS232
PS8802
JTAG1
JTAG2
RF1
RF2
(Output Attn.)

Figure B.1 Digital Board

APPENDIX C. INA333 Instrumentation Amplifier

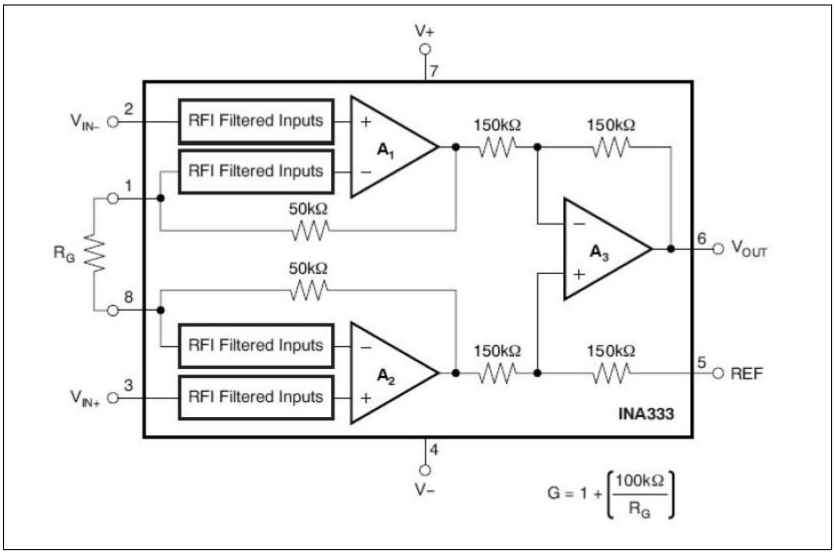


Figure C.1 INA333 Instrumentation Amplifier

APPENDIX D. SOURCE CODE LISTING

```
//******************************************************************************
//   MSP430FG4618 - Two Channel Low power EEG Mu & Beta Rhythm Classification
//   (Texas Instruments MSP430FG461x/F20xx Experimenter's Board)
//    main.c  (requires TI freeware 16*16>32 multiply mul.s43 assembly code)
//
//   Description; Uses analog board and INA333 instrumentation amplifiers to detect
//   the brainwaves, after amplification & anti-aliasing filter and ADC, the code
//   extracts mu and Beta Rhtym features of two channels using
//   FIR band pass filters and classifies the motor imagery, right or left using
//   A MLP backpropagation neural network, displays the result on LCD.
//   Multiplication is implemented in assembly for FIR band pass filters
//   to extract sub frequency band features.
//
//   Operation: After Power on Reset, code inits the peripherals, main clock freq,
//   ADC, LCD registers and digital I/O ports. Writes 'Hello' to LCD registers and
//   enters low power mode 3 (CPU Off / 32 KHz ACLK On, 1.3 μA) waiting for interrupts.
//   ADC uses TimerA triggering, +1.5V internal, -1.5V external references which
//   are the supply ranges of instrumentation and operational amplifiers in the same time.
//   CPU operates with 3V battery, so with a voltage divider, supply voltage and
//   reference values are generated.
//   When Button 1 is pressed, Port1 Interrupt wakes up the CPU an in the Port1
//   Interrupt Service Routine Basic timer is initialized to give about 1 second delay
//   While displaying 'Ready' on LCD. After servicing the ISR the cpu automatically
//   enters back to LPM3 awakened by Basic timer interrupt after 1 second. 'Start'
//   is written to the LCD in the Basic timer interrupt Routine and TimerA is initialized
//   to start the ADC sampling with 128Hz sampling rate. After 448 samples are
//   stored (448 comes from memory limitations and it means  3.5 seconds sampling)
//   TimerA is stopped and stored values are filtered. 4 features are calculated
//   and scaled for neural network input. After neural net classification results
//   are displayed on LCD, then 'Hello' is displayed again and system enters back
//   to the low power mode 3 waiting for button interrupt.
//
//   Notes:
//    -- Source  generates about 40KB program code and uses about 5KB RAM, which
//   are not available in many of the members of MSP430 family excluding MSP430X
//   architecture and new MSP430x5xxx series.
//    -- Hardware multiplier and DMA gives DSP like capabilities to FG4618.But
//    many of the MSP430 devices lack MPY or DMA can be very slow on multiplication.
//    -- Different feature extraction bands can be applied by only changing the
//    filter coefficients. ScopeFIR program can be used to design and scale
//    filter coefficients.
//    -- Neural Network performance can be tested in a memory free environment.
//    Neural network architecture can be easily customized by changing hidden neuron
//    and learning rate variables.
```

```
//
//   Ozan Gunaydin
//   Boğaziçi University
//   Istanbul; ozangunaydin@gmail.com
//   2010
//*****************************************************************************
//*****************************************************************************
//*****************************************************************************

#include <msp430xG46x.h>                         // Specidic Device Header file
#include <stdio.h>
#include <stdlib.h>
#include <intrinsics.h>// Intrinsic functions
#include <stdint.h>       // Integers of defined sizes
#include <time.h>
#include <math.h>


// Data Dependent settings for Neural Network
#define numInputs  4                // Neural Network inputs
#define numPatterns  1              // Input pattern Number, if training utilized it is the number of input pattern

// User defineable Neural Network settings
#define numHidden 20                // Number of hidden neurons
// const int numEpochs = 10000;     // To be used in training mode
const double LR_IH = 0.7;           // input learning rate
const double LR_HO = 0.07;          // output learning rate


// Band pass FIR filter coefficients for Beta Rhythm extraction , 19-24 Hz
static const int coeffsbp2[14] = {
     334, 1153, 775, -2639, -6095, -3382, 7289, 15549, 8170,-12686,-25777, -13271,15302,15135};

// Band pass FIR filter coefficients for Mu Rhythm extraction  ,  8 - 12 Hz
static const int coeffsbp1[12] = {
     9453, -1255, -3809, -7204,-10221, -11525, -10079, -5625, 1038,8175,13618, 7836};


#define LCDMEMS 11 // LCD memories used (3-13)
// Pointer to LCD memory used: allows use of array LCDMem[]
uint8_t * const LCDMem = (uint8_t *) &LCDM3;
// LCD segment definitions (SoftBaugh SBLCDA4)
#define SEG_A   BIT0 //  AAAA
#define SEG_B   BIT1 // F    B
#define SEG_C   BIT2 // F    B
#define SEG_D   BIT3 //  GGGG
#define SEG_E   BIT6 // E    C
#define SEG_F   BIT4 // E    C
```

```
#define SEG_G   BIT5 //  DDDD
#define SEG_H   BIT7 // colon, point

// Patterns for hexadecimal characters
const uint8_t LCDHexChar[] = {
SEG_A | SEG_B | SEG_C | SEG_D | SEG_E | SEG_F,// '0'
SEG_B | SEG_C,// '1'
SEG_C | SEG_F | SEG_E | SEG_G,// 'H'
SEG_F | SEG_E | SEG_D,          // 'L'
SEG_A | SEG_B | SEG_F | SEG_G | SEG_E,// 'P'
SEG_G | SEG_E,                              // 'r'
SEG_F | SEG_B | SEG_C | SEG_D | SEG_G,        // 'y'
SEG_F | SEG_G | SEG_E,// 't'
SEG_F | SEG_E | SEG_D | SEG_C | SEG_B,                     // 'U'
SEG_G | SEG_E | SEG_D | SEG_C,            // 'o'
SEG_A | SEG_B | SEG_C | SEG_E | SEG_F | SEG_G,// 'A'
SEG_A | SEG_F | SEG_G | SEG_C | SEG_D,      // 'S'
SEG_A | SEG_F | SEG_E | SEG_D | SEG_C,              // 'g'
SEG_B | SEG_C | SEG_D | SEG_E | SEG_G,// 'd'
SEG_A | SEG_D | SEG_E | SEG_F | SEG_G,// 'E'
SEG_A | SEG_E | SEG_F | SEG_D,// 'C'
};
const uint8_t LCDBlankChar = 0;


// ADC variables declaration
//
  int A0results[448];    // Data array for ADC samples CH0
  int A1results[448];    // Data array for ADC samples CH1

int A0muResult[448];     // Data array for filtered signals
int A1muResult[448];      // Data array for filtered signals

// Neural Network variables
int patNum = 0;             //Number of input pattern
double errThisPat = 0.0;    //Neural network Error
double outPred = 0.0;       // Neural network output
double RMSerror = 0.0;      // Neural Network RMS error
  // NN the outputs of the hidden neurons
double hiddenVal[numHidden];
  // NN the weights
double weightsIH[numInputs][numHidden];
double weightsHO[numHidden];
  //NN Input data
 double trainInputs[numInputs];

//variables for calculations
unsigned int i,Index;
```

```
int  z,clas, sum1,sum2,output1,output2 =0;
float mean_mu_C3,mean_mu_C4,mean_B_C3,mean_B_C4;    // feature variables

int compare(const void * a, const void * b)         //compare function for sorting
{
  return ( *(int*)a - *(int*)b );
}



// function prototypes
void Init(void);                               // Initializes the peripherals
void InitLCD(void);                            // Clears the LCD memory

int filterbp2(int);                     // Band Pass filter Beta Rhythm
int filterbp1(int);                      // Band Pass filter Mu Rhythm

  // LCD writing functions
void DisplayHello(void);
void DisplayResult1(void);
void DisplayResult2(void);
void DisplayReady(void);
void DisplayGo(void);
void DisplayStop(void);

  //Neural Net Functions
void initWeights();
void calcNet();
//void WeightChangesHO();            //Weight changes for training mode
//void WeightChangesIH();           //Weight changes for training mode
//void calcOverallError();          //Error Calculating for backpropagation
//void displayResults();
double getRand();


long mul16(register int x, register int y);            // 16-bit signed multiplication


// main function
void main(void)
{   Init();                               // Initialize device for the application
    DisplayHello();

    while(1)           // loop forever
   { _BIS_SR(LPM3_bits + GIE);       // Enter Low power mode 3, interrupts enable


    if (Index == 448)               // Stop and calculate class when 448 array filled
    {
      DisplayStop();
      TACTL =  MC_0;            // stop Timer A

      BTCTL |= BIT6;            // stop basic timer

      for(i=0;i<60000;i++)           // delay for Stop message
      {;}



       for (i=0;i<447;i++)        //filtering to extract Mu Rhythm
      {
output1, output2=0;
output1=filterbp1(A0results[i]);
                                output2=filterbp1(A1results[i]);
                                 A0muResult[i]=output1;
                                 A1muResult[i]=output2;
}

           for(i=0;i<447;i++)                 // take absolute value
      {
         A0muResult[i] = abs(A0muResult[i]);
         A1muResult[i] = abs(A1muResult[i]);
        }

      sum1=0;
      sum2=0;
       for (i=0;i<447;i++)                 // Calculate mean values of bands
{

                              //  sum1 = sum1 + A0BResult[i];
                                sum1 = sum1 +  A0muResult[i];
                              //  sum2 = sum2 + A0results[i];
                                sum2 = sum2 + A1muResult[i];
}
      mean_mu_C3 = sum1 / 448;               // first two features: mean of absolute values
      mean_mu_C4 = sum2 / 448;



       for (i=0;i<447;i++)        //filtering to extract Mu Rhythm
{
output1, output2=0;

output1=filterbp2(A0results[i]);
                                output2=filterbp2(A1results[i]);

                                 A0muResult[i]=output1;
                              A1muResult[i]=output2;
}

      for(i=0;i<447;i++)                          // take absolute value
      {
         A0muResult[i] = abs(A0muResult[i]);
         A1muResult[i] = abs(A1muResult[i]);
      }



      sum1=0;
      sum2=0;

      for (i=0;i<447;i++)                        // Calculate mean values of bands
{

                            // sum1 = sum1 + A0BResult[i];
                             sum1 = sum1 + A0muResult[i];
                            // sum2 = sum2 + A0results[i];
                             sum2 = sum2 + A1muResult[i];
}
      mean_B_C3 = sum1 / 448;               // last two features: mean of absolute values
      mean_B_C4 = sum2 / 448;



// Scaling values for neural network input
// Scaling is done by calculating min and max values of the arrays and then
// scaling them to the [-1,1] range.
      // Scaling formula: I = Imin + ((Imax - Imin)*(D-Dmin)/(Dmax -Dmin))

    trainInputs[0] = (((mean_mu_C3 - 100)/ 900)*2)-1;
    trainInputs[1] = (((mean_mu_C4 - 100)/ 900)*2)-1;
    trainInputs[2] = (((mean_B_C3 - 50)/ 400)*2)-1;
    trainInputs[3] = (((mean_B_C4 - 50)/ 400)*2)-1;

    calcNet();             //calculating Neural network response





      if (outPred < 0.0)
      { DisplayResult1();}        // if output is negative,then result is left hand
      else if (outPred > 0.0)
      { DisplayResult2();}      // if output is positive,then result is right hand
      else
      {if(mean_mu_C3> mean_mu_C4){DisplayResult1();} //if output is zero then compare mu rhythm means
      else{DisplayResult2();}}
```

```
            for(i=60000;i>0;i--)        // delay
            {;}
          //   BTCTL |= BIT6;            // stop basic timer


              for(i=60000;i>0;i--)      // delay
            {;}

            Index=0;    // reset index counter

      DisplayHello();

    }   // end of data processing, turn back to the infitine loop
    }  // infinite loop
}//main


// Initialization function
void Init(void)
{
    initWeights();                      // Init Neural Network Weights
    FLL_CTL0 |= XCAP18PF;               // Set load capacitance for xtal
    WDTCTL = WDTPW | WDTHOLD;           // Disable the Watchdog
    while ( LFOF & FLL_CTL0);           // wait for watch crystal to stabilize
    SCFQCTL = 63;                       // 32 x 32768 x 2 = 2.097152MHz

    InitLCD();                          // Clear LCD memory

    P1OUT = 0x00;
    P1DIR = 0xfe;                          // Unused pins as outputs, button pins as inputs
    P2OUT = 0x00;                          // Clear P2OUT register
    P2DIR = 0xff;                          // Unused pins as outputs
    P3OUT = 0x00;                          // Clear P3OUT register
    P3DIR = 0xff;                         // Unused pins as outputs
    P4OUT = 0x00;                          // Clear P4OUT register
    P4DIR = 0xff;                          // Unused pins as outputs
    P5OUT = 0x00;                          // Clear P5OUT register
 //   P5DIR = 0xff;                        // Unused pins as outputs
    P5SEL = BIT4|BIT3|BIT2;

    P6OUT = 0x00;                            // Clear P6OUT register
    P6SEL = 0xff;                             // P6 = Analog selection for ADC input
    P7OUT = 0x00;
    P7DIR = 0xff;
    P1IE |= BIT0;                      //Enable button 1 interrupt
    P1IES |= BIT0;

  do {                                 // Clear interrupts for Port1
  P1IFG = 0; // Clear any pending interrupts ...
  } while (P1IFG != 0); // ... until none remain

// Initialize and enable ADC12
    ADC12CTL0 = ADC12ON + SHT0_8 + REFON;      // ADC12 ON, Reference = 1.5V internal, -1.5V ext
    ADC12CTL1 = SHP + SHS_1 + CONSEQ_3;        // Use sampling timer, TA1 trigger
     ADC12MCTL0 = INCH_0 + SREF_5 ;           // A0 goes to MEM0, Vref+, VeRef-
     ADC12MCTL1 = EOS + INCH_1 + SREF_5 ;     // A1 goes to MEM1,Vref+, VeRef-
    ADC12IE = 0x02;                           // Enable ADC12IFG.1 for ADC12MEM1
    ADC12CTL0 |= ENC;                         // Enable conversions
     _EINT();                                                // Enable global Interrupts
} //init



void InitLCD(void)
{
int i;
for(i = 0; i < LCDMEMS; ++i) { // Clear LCD memories used
LCDMem[i] = 0;
}
P5SEL = BIT4|BIT3|BIT2; // Select COM[3:1] function
LCDAPCTL0 = LCDS4|LCDS8|LCDS12|LCDS16|LCDS20|LCDS24;
// Enable LCD segs 4-27 (4-25 used)
LCDAVCTL0 = 0; // No charge pump, everything internal
LCDACTL = LCDFREQ_128 | LCD4MUX | LCDSON | LCDON;
// ACLK/128, 4mux, segments on, LCD_A on
}



// Interrupt service routine for port 1 inputs
// Only one bit is active so no need to check which
// clear any pending interrupts
// Device returns to low power mode automatically after ISR
// ----------------------------------------------------------------------
#pragma vector = PORT1_VECTOR
__interrupt void PORT1_ISR (void)
{

  DisplayReady();

  do {
    P1IFG = 0;                        // Clear any pending interrupts ...
      } while (P1IFG != 0);           // ... until none remain
```

```
   BTCTL &= ~BIT6;
   BTCNT1 = 0x00;
   BTCNT2 = 0x00;
   BTCTL = BTDIV + BTIP0 + BTIP1 + BTIP2;           // ACLK/(256*256) one second delay
  IE2 |= BTIE;                                // Enable BT interrupt
}

// Interrupt service routine for port ADC12
// After conversion results are stored in data array
// Turn back to low power mode
// ----------------------------------------------------------------------
#pragma vector = ADC12_VECTOR                                  // ADC12 ISR
__interrupt void ADC12ISR (void)
{


  A0results[Index] = ADC12MEM0;             // Move A0 results, IFG is cleared
  A1results[Index] = ADC12MEM1;             // Move A1 results, IFG is cleared
  Index = (Index + 1);                // Increment results index, modulo

  __no_operation();                         // breakpoint to see ADC results
   __bic_SR_register_on_exit(LPM3_bits);  // Exit LPM3 on return

}// ADC12ISR


// Interrupt service routine for Basic timer
// After button press Basic timer provides 1 sec delay for displaying Ready
// Then in ISR , timerA is initialized to start conversion
// ----------------------------------------------------------------------
// Basic Timer Interrupt Service Routine
#pragma vector=BASICTIMER_VECTOR
__interrupt void basic_timer_ISR(void)
{
  DisplayGo();
  IE2 &= ~BIT7;
    TACTL = TASSEL_1 + MC_1 + TACLR;                          // ACLK, Clear TAR, Up Mode
    TACCTL1 = OUTMOD_2;                                      // Set / Reset
    TACCR0 = 255;                                             // 128 samples per second
    TACCR1 = 15;                                             //

}


int filterbp2(int sample)                                     // Band Pass FIR filter for Beta Rhtym
{   static int buflp[32];                                    // Reserve 32 loactions for circular buffering
    static int offsetlp = 0;
    long z;
    int i;
    buflp[offsetlp] = sample;
    z = mul16(coeffsbp2[13], buflp[(offsetlp - 13) & 0x1F]);        //multiplication
    for (i = 0;  i < 13;  i++)
    z += mul16(coeffsbp2[i], buflp[(offsetlp - i) & 0x1F] + buflp[(offsetlp - 26 + i) & 0x1F]);
    offsetlp = (offsetlp + 1) & 0x1F;
    return  z >> 15;                                         // Return filter output
}


int filterbp1(int sample)                                     //  Band Pass FIR filter for mu Rhtym
{   static int buflp[32];                                    // Reserve 32 loactions for circular buffering
    static int offsetlp = 0;
    long z;
    int i;
    buflp[offsetlp] = sample;
    z = mul16(coeffsbp1[11], buflp[(offsetlp - 11) & 0x1F]);
    for (i = 0;  i < 11;  i++)
    z += mul16(coeffsbp1[i], buflp[(offsetlp - i) & 0x1F] + buflp[(offsetlp - 22 + i) & 0x1F]);
    offsetlp = (offsetlp + 1) & 0x1F;
    return  z >> 15;                                         // Return filter output
}



void calcNet(void)                       //Neural Network Calculation function
{
    //calculate the outputs of the hidden neurons
    //the hidden neurons are tanh
    int i = 0;
    for(i = 0;i<numHidden;i++)
    {
  hiddenVal[i] = 0.0;

        for(int j = 0;j<numInputs;j++)
        {
   hiddenVal[i] = hiddenVal[i] + (trainInputs[j]  * weightsIH[j][i]);
        }

        hiddenVal[i] = tanh(hiddenVal[i]);
    }

   //calculate the output of the network
   //the output neuron is linear
   outPred = 0.0;

   for(i = 0;i<numHidden;i++)
   {
    outPred = outPred + hiddenVal[i] * weightsHO[i];
   }
    //calculate the error
   // errThisPat = outPred - trainOutput[patNum];   // for training

}

void initWeights(void)                    // Neural network weights
{

weightsIH[0][0] = -8.951717;
weightsIH[1][0] = 7.653355;
weightsIH[2][0] = 6.631219;
weightsIH[3][0] = 11.094660;
weightsIH[0][1] = 3.310763;
weightsIH[1][1] = 3.215580;
weightsIH[2][1] = -0.963815;
weightsIH[3][1] = 6.746135;
weightsIH[0][2] = 3.641091;
weightsIH[1][2] = -1.118857;
weightsIH[2][2] = -2.280876;
weightsIH[3][2] = -12.060060;
weightsIH[0][3] = 11.269429;
weightsIH[1][3] = 6.643443;
weightsIH[2][3] = -6.614878;
weightsIH[3][3] = 14.733462;
weightsIH[0][4] = 0.743146;
weightsIH[1][4] = 10.348796;
weightsIH[2][4] = -15.558205;
weightsIH[3][4] = 18.596556;
weightsIH[0][5] = -12.275212;
weightsIH[1][5] = -1.859199;
weightsIH[2][5] = -20.228093;
weightsIH[3][5] = -12.789903;
weightsIH[0][6] = -21.430971;
weightsIH[1][6] = -3.899026;
weightsIH[2][6] = -12.496226;
weightsIH[3][6] = 4.471426;
weightsIH[0][7] = 1.179519;
weightsIH[1][7] = 8.862174;
weightsIH[2][7] = 7.262130;
weightsIH[3][7] = 0.380516;
weightsIH[0][8] = -3.315827;
weightsIH[1][8] = 0.415380;
weightsIH[2][8] = -0.804302;
weightsIH[3][8] = -7.525758;
weightsIH[0][9] = 8.997232;
weightsIH[1][9] = 6.043105;
```

```
weightsIH[2][9] = 4.525014;
weightsIH[3][9] = -4.655872;
weightsIH[0][10] = -2.567476;
weightsIH[1][10] = -1.001431;
weightsIH[2][10] = -5.790394;
weightsIH[3][10] = -12.538476;
weightsIH[0][11] = 0.964474;
weightsIH[1][11] = -6.690062;
weightsIH[2][11] = 5.976368;
weightsIH[3][11] = -26.924518;
weightsIH[0][12] = 6.004541;
weightsIH[1][12] = 5.625768;
weightsIH[2][12] = 10.801397;
weightsIH[3][12] = -0.778696;
weightsIH[0][13] = -2.619037;
weightsIH[1][13] = -11.472975;
weightsIH[2][13] = -9.598161;
weightsIH[3][13] = 0.805282;
weightsIH[0][14] = 2.341189;
weightsIH[1][14] = -7.747893;
weightsIH[2][14] = 0.036407;
weightsIH[3][14] = -11.517146;
weightsIH[0][15] = 2.618314;
weightsIH[1][15] = 2.424869;
weightsIH[2][15] = -7.886291;
weightsIH[3][15] = -12.294660;
weightsIH[0][16] = -1.006894;
weightsIH[1][16] = -3.148300;
weightsIH[2][16] = -10.965334;
weightsIH[3][16] = -2.622260;
weightsIH[0][17] = 1.943215;
weightsIH[1][17] = 9.396184;
weightsIH[2][17] = 8.086076;
weightsIH[3][17] = 0.148540;
weightsIH[0][18] = -7.163011;
weightsIH[1][18] = -5.225458;
weightsIH[2][18] = 15.685931;
weightsIH[3][18] = -16.417650;
weightsIH[0][19] = -11.894525;
weightsIH[1][19] = 18.598566;
weightsIH[2][19] = -1.046420;
weightsIH[3][19] = 14.616560;
weightsHO[0] = -1.369972;
weightsHO[1] = 0.576864;
weightsHO[2] = -0.742043;
weightsHO[3] = -0.751341;
weightsHO[4] = 0.608129;
weightsHO[5] = 1.522254;
```

```
weightsHO[6] = -1.743967;
weightsHO[7] = 0.138998;
weightsHO[8] = -0.989855;
weightsHO[9] = -2.040964;
weightsHO[10] = 1.837575;
weightsHO[11] = 1.636948;
weightsHO[12] = 1.512855;
weightsHO[13] = 0.959156;
weightsHO[14] = -1.104560;
weightsHO[15] = -1.112830;
weightsHO[16] = -1.348077;
weightsHO[17] = -0.206748;
weightsHO[18] = -0.431466;
weightsHO[19] = 1.069858;

}



void DisplayHello(void){
        LCDMem[1] = LCDHexChar[9];                    //Display hello on LCD
LCDMem[2] = LCDHexChar[3];
LCDMem[3] = LCDHexChar[3];
        LCDMem[4] = LCDHexChar[14];
        LCDMem[5] = LCDHexChar[2];
            }
void DisplayResult1(void){
      LCDMem[1] = LCDHexChar[3];                   //Display Clas: sol on LCd
LCDMem[2] = LCDHexChar[9];
LCDMem[3] = LCDHexChar[11];
         LCDMem[4] = LCDBlankChar | SEG_H;
         LCDMem[5] = LCDHexChar[15];
            }

void DisplayResult2(void){
  LCDMem[1] = LCDHexChar[12];                   //Display Clas: sag on LCD
LCDMem[2] = LCDHexChar[10];
LCDMem[3] = LCDHexChar[11];
         LCDMem[4] = LCDBlankChar | SEG_H;
         LCDMem[5] = LCDHexChar[15];


            }
void DisplayReady(void){
        LCDMem[1] = LCDHexChar[6];                   //Display ready on LCD
LCDMem[2] = LCDHexChar[13];
LCDMem[3] = LCDHexChar[10];
        LCDMem[4] = LCDHexChar[14];
        LCDMem[5] = LCDHexChar[5];
            }
```

```
void DisplayGo(void){
        LCDMem[1] = LCDHexChar[7];                   //Display start on LCD
LCDMem[2] = LCDHexChar[5];
LCDMem[3] = LCDHexChar[10];
        LCDMem[4] = LCDHexChar[7];
        LCDMem[5] = LCDHexChar[11];
            }

void DisplayStop(void){

        LCDMem[1] = LCDHexChar[4];                   //Display stop on LCD
LCDMem[2] = LCDHexChar[9];
LCDMem[3] = LCDHexChar[7];
        LCDMem[4] = LCDHexChar[11];
        LCDMem[5] = LCDBlankChar;
            }
//end of main.c
```

```
//
//16x16=>32 multiply
//long mul16(register int x, register int y)
//
// Edited by: M Morales, November 2008
// * Updated calling conventions in support of IAR compiler >= 4.x
//******************************************************************************
// THIS PROGRAM IS PROVIDED "AS IS". TI MAKES NO WARRANTIES OR
// REPRESENTATIONS, EITHER EXPRESS, IMPLIED OR STATUTORY,
// INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS
// FOR A PARTICULAR PURPOSE, LACK OF VIRUSES, ACCURACY OR
// COMPLETENESS OF RESPONSES, RESULTS AND LACK OF NEGLIGENCE.
// TI DISCLAIMS ANY WARRANTY OF TITLE, QUIET ENJOYMENT, QUIET
// POSSESSION, AND NON-INFRINGEMENT OF ANY THIRD PARTY
// INTELLECTUAL PROPERTY RIGHTS WITH REGARD TO THE PROGRAM OR
// YOUR USE OF THE PROGRAM.
//
// IN NO EVENT SHALL TI BE LIABLE FOR ANY SPECIAL, INCIDENTAL,
// CONSEQUENTIAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ON ANY
// THEORY OF LIABILITY AND WHETHER OR NOT TI HAS BEEN ADVISED
// OF THE POSSIBILITY OF SUCH DAMAGES, ARISING IN ANY WAY OUT
// OF THIS AGREEMENT, THE PROGRAM, OR YOUR USE OF THE PROGRAM.
// EXCLUDED DAMAGES INCLUDE, BUT ARE NOT LIMITED TO, COST OF
// REMOVAL OR REINSTALLATION, COMPUTER TIME, LABOR COSTS, LOSS
// OF GOODWILL, LOSS OF PROFITS, LOSS OF SAVINGS, OR LOSS OF
// USE OR INTERRUPTION OF BUSINESS. IN NO EVENT WILL TI'S
// AGGREGATE LIABILITY UNDER THIS AGREEMENT OR ARISING OUT OF
// YOUR USE OF THE PROGRAM EXCEED FIVE HUNDRED DOLLARS
// (U.S.$500).
//
// Unless otherwise stated, the Program written and copyrighted
// by Texas Instruments is distributed as "freeware".  You may,
// only under TI's copyright in the Program, use and modify the
// Program without any charge or restriction.  You may
// distribute to third parties, provided that you transfer a
// copy of this license to the third party and the third party
// agrees to these terms by its first use of the Program. You
// must reproduce the copyright notice and any other legend of
// ownership on each copy or partial copy, of the Program.
//
// You acknowledge and agree that the Program contains
// copyrighted material, trade secrets and other TI proprietary
// information and is protected by copyright laws,
// international copyright treaties, and trade secret laws, as
// well as other intellectual property laws.  To protect TI's
// rights in the Program, you agree not to decompile, reverse
// engineer, disassemble or otherwise translate any object code
// versions of the Program to a human-readable form.  You agree
// that in no event will you alter, remove or destroy any
// copyright notice included in the Program.  TI reserves all
// rights not specifically granted under this license. Except
// as specifically provided herein, nothing in this agreement
// shall be construed as conferring by implication, estoppel,
// or otherwise, upon you, any license or other right under any
// TI patents, copyrights or trade secrets.
//
// You may not use the Program in non-TI devices.
//*****************************************************************************
public mul16

RSEG CODE
mul16

#define x1 r9
#define z0 r14
#define z1 r15
#define x  r12
#define y  r13

        push    r9

clr z0
```

```
mov z0,z1
mov z0,x1
tst x
jge xbooth_2
mov #-1,x1
jmp xbooth_2

xbooth_6
add x,z1
addc  x1,z0
xbooth_1
rla x
rlc x1
xbooth_2
rra y
jc xbooth_5
jne xbooth_1
jmp xbooth_4

xbooth_5
sub x,z1
subc  x1,z0
xbooth_3
rla x
rlc x1
rra y
jnc xbooth_6
cmp #0FFFFh,y
jne xbooth_3

xbooth_4
        mov     z1,r12
        mov     z0,r13

        pop     r9

        reta
        end
//end of mul.s43
```

REFERENCES

1. Niedermeyer, E., and L. da Silva, *Electroencephalography, 4th Ed.*, Baltimore, MD: Williams Wilkins, 1999.

2. Anderson, C. W., S. Devulapalli, and E. A. Stolz, "Determining mental state from EEG signals using neural networks," *Scientific Programming*, pp. 171–183, 1995.

3. Caton, R. *British Medical Journal*, 1875.

4. Berger, H., "Uber das elektroenkephalogramm des menschen," *Archiv fur Psychiatrie und Nervenkrankheiten*, Vol. 87, pp. 527–570, 1929.

5. Wolpaw, R. J., N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain computer interfaces for communication and control," *Clinical Neurophysiology*, Vol. 113, pp. 767–791, 2002.

6. Middendorf, M., G. McMillan, G. Calhoun, and K. S. Jones, "Brain computer interfaces based on steady-state visual evoked response," *IEEE Trans. Rehabil. Eng.*, Vol. 8, pp. 211–213, 2000.

7. Birbaumer, N., A. Kubler, N. Ghanayim, T. Hinterberger, J. Perelmouter, J. Kaiser, I. Iversen, B. Kotchoubey, N. Neumann, and H. Flor, "The thought translation device (TTD) for completely paralyzed patients," *IEEE Trans. Rehabil. Eng.*, Vol. 8, pp. 190–192, 2000.

8. McFarland, D. J., L. A. Miner, T. M. Vaughan, and J. R. Wolpaw, "Mu and beta rhythm topographies during motor imagery and actual movement," *Brain Topogr.*, Vol. 12-3, pp. 177–186, 2000.

9. Kennedy, P. R., "The cone electrode: a long-term electrode that records from neurites grown onto its recording surface," *J. Neurosci. Methods*, Vol. 29(3), pp. 181–193, 1989.

10. Donchin, E., K. M. Spencer, and R. Wijesinghe, "The mental prosthesis: assessing the speed of a P300-based brain computer interface," *IEEE Trans. Rehabil. Eng*, Vol. 8, pp. 174–179, 2000.

11. "BCI competition 2," tech. rep., 2003. Available: http://www.bbci.de/competition/ii/.

12. Blankertz, B., K. R. Muller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlogl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schroder, and N. Birbaumer, "The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials," *IEEE Trans. Biomedical Eng*, Vol. 51, 2004.

13. Kandel, E., J. Schwartz, and T. Jessel, *Principles of Neural Science*, New York: Elsevier, 1991.

14. Neuper, C., and G. Pfurtscheller, "Event-related dynamics of cortical rhythms: frequency specific features and functional correlates," *Inter. Journal of Psychophysics*, Vol. 43, pp. 41–58, 2001.

15. Hines, T., "Anatomy of the brain," tech. rep., Mayfield Clinic, Cincincati,OH, 2007. Available: http://www.mayfieldclinic.com/PE-AnatBrain.htm.

16. Gaustaut, H., "Etude electrocorticographique de la reactivite des rhytmes rolandiques.," *Rev. Neurol.*, Vol. 87, pp. 176–182, 1952.

17. Niedermeyer, E., “Alpha rhythms as normal and abnormal phenomena,” *Int. J. Psychophysiol*, Vol. 26, pp. 31–49, 1997.

18. Pfurtscheller, G., and A. Aranibar, “Evaluation of event-related desyncronization (ERD) preceding and following voluntary self-paced movement,” *Electroencephal. clin. Neurophysiol.*, Vol. 46, pp. 138–146, 1979.

19. Chatrian, G. E., M. C. Petersen, and A. Lazarte, “The blocking of the rolandic wicket rhythm and some central changes related to movement,” *Electroencephal. clin. Neurophysiol.*, Vol. 11, pp. 497–510, 1959.

20. Chatrian, G. E., and G. C. Lairy, *The EEG of the Waking Adult: Handbook of Electroencephalography and Clinical Neurophysiology*, Vol. 6A, pp. 46–69. Amsterdam: Elsevier, 1976.

21. Kuhlman, W. N., “Functional topography of the human mu rhythm,” *Electroencephal. clin. Neurophysiol.*, Vol. 44, pp. 83–93, 1978.

22. Pfurtscheller, G., “Mapping of event-related desyncronization and type of derivation,” *Electroencephal. clin. Neurophysiol.*, Vol. 70, pp. 190–193, 1988.

23. Pfurtscheller, G., “Functional topography during sensorimotor activation studied with event-related desynchronization,” *J. Clin. Neurophysiol.*, Vol. 6, pp. 75–84, 1989.

24. Pfurtscheller, G., *Electroencephalography: Basic Principles, Clinical Applications and Related Fields*, pp. 958–967. EEG event-related desynchronization (ERD) and event-related synchronization (ERS), Baltimore: Williams and Wilkins, 4 ed., 1998.

25. Jasper, H., “The ten twenty electrode system of the international federation,” *Electroencephalography and Clinical Neurophysiology*, Vol. 10, pp. 371–375, 1958.

26. “EEG measurement setup,” tech. rep., Wadsworth Center of the New York State Department of Health, New york,USA. Available: http://www.bci2000.org.

27. John, R. K., S. F. Tyner, and J. B. Mayer, *Fundamentals of EEG Technology*, New York: Raven Press, 1983.

28. Ince, N. F., *Analysis and Classification Of EEG with Adapted Wavelets and Local Discriminant Bases.* PhD thesis, Cukurova University, Adana, Turkey, 2005.

29. Dornhege, G., J. Millan, T. Hinterberger, D. McFarland, and K. Muller, *Toward Brain Computer Interfacing*, London: MIT Press, 2007.

30. McFarland, D. J., W. A. Sarnacki, and J. R. Wolpaw, “Brain computer interface (BCI) operation: optimizing information transfer rates,” *Biological Psychology*, Vol. 63, no. 3, pp. 237–251, 2003.

31. Wolpaw, J. R., and D. J. McFarland, “Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans,” *Proceedings of the National Academy of Science of the United States of America*, Vol. 101, no. 51, pp. 17849–17854, 2004.

32. Berger, T., J. Chapin, G. Gerhardt, D. McFarland, J. Principe, W. Soussou, D. Taylor, and P. Tresco, “International assessment of research and development in brain-computer interfaces,” tech. rep., World Technology Evaluation Center, Baltimore,Maryland, 2007. Available: http://www.wtec.org/bci/BCI-finalreport-10Oct2007-lowres.pdf.

33. Venkat, K., “Digital filtering methodologies for MSP430 systems,” in *ATC 2008 MSP430 Advanced Technical Conference*, Texas Instruments, 2008.

34. Murugavel, R., “Digital FIR filter design using the MSP430F16x,” *Texas Instruments Application Report*, Vol. SLAA228, 2004.

35. Gabor, D., “Theory of communication,” *J.Inst. Elec. Eng.*, Vol. 93, pp. 429–457, 1946.

36. Papuolis, A., *Probability, Random Variables, and Stochastic Processes*, New York: McGraw-Hill, 1991.

37. Rioul, O., and M. Vetterli, “Wavelets and signal processing,” *IEEE Signal Processing Magazine.*, Vol. 8, pp. 11–38, 1991.

38. Xu, B. G., and A. G. Song, “Pattern recognition of motor imagery EEG using wavelet transform,” *J. Biomedical Science And Engineering*, Vol. 1, pp. 64–67, 2008.

39. Blais, A., and D. Mertz, “An introduction to neural networks,” *IBM DeveloperWorks Library*, 2001. Available: http://www.ibm.com/developerworks/linux/library/l-neural/.

40. Yilmaz, Y. K., “Detection of P300 components in single trials by an artifical neural network,” Master’s thesis, Bogazici University, Istanbul, Turkey, 1998.

41. Haykin, S., *Neural Networks : A Comphrensive Foundation*, Ontario: Prentice Hall, 1994.

42. Hinton, G. E., “How neural networks learn from experience,” *Scientific American*, Vol. 267, pp. 105–109, 1992.

43. Chatila, A., J. V. Asdlan, J. Bitz, H. Barrow, M. Fimbres, and M. Crowder, “Portable brainwave monitor,” tech. rep., University Of Arizona, Arizona, 2009.

44. “OpenEEG project , open source hardware and software project for EEG analysis, BCI research,” tech. rep. Available: http://openeeg.sourceforge.net.

45. Schlogl, A., K. Lugger, and Pfurtscheller, “Using adaptive autoregressive parameters for a brain-computer-interface experiment,” in *Proc. 19th Int. Conf. IEEE/EMBS*, pp. 1533–1535, IEEE for Engineering in Medicine and Biology Society, 1997.

46. Neuper, C., A. Schlogl, and G. Pfurtscheller, “Enhancement of left-right sensorimotor EEG differences during feedback-regulated motor imagery,” *J. Clin. Neurophysiol.*, Vol. 16, p. 373â382, 1999.

47. Schlogl, A., C. Keinrath, R. Scherer, and G. Pfurtscheller, “Information transfer of an EEG based brain-computer interface,” in *Proc. 1st Int. IEEE EMBS Conf. Neural Eng.*, pp. 641 –644, IEEE for Engineering in Medicine and Biology Society, 2003.

48. “Getting started with TINA-TI,” tech. rep., Texas Instruments, Dallas,Texas, 2007. Available:http://focus.ti.com/docs/toolsw/folders/print/tina-ti.html.

49. “Patient simulators - 330 EEG simulator,” tech. rep., Netech, Farmingdale, NY, 2007. Available:http://www.netech.org/test.asp?cid=9pid=22.

50. “ScopeFIR: FIR filter design software for windows,” tech. rep., IOWEGIAN INTERNATIONAL, 2010. Available:http://www.iowegian.com/scopefir.