Abstract

ADCOCK, DAVID BROOKS. Rapid Prototyping of a Single-Channel Electroencephalogram-Based Brain-Computer Interface. (Under the direction of Edward Grant.)

This work describes the design, construction and implementation of a single-channel, electroencephalogram-based (EEG) brain-computer interface (BCI) for the prediction of a single-degree-of-freedom kinematic variable. The system employs a custom-built EEG amplifier to increase noise rejection and decrease the overall cost of the BCI. The EEG amplifier output is read into Matlab synchronously with an analog elbow-angle measurement taken from the test subject's left arm. Sampling is done at 300Hz using a 12-bit National Instruments PCI-6025E data acquisition card. Data is software filtered, processed, and logged in Matlab in real-time on a standard PC. At the end of an initial data acquisition period, a feed-forward backpropagation artificial neural network (ANN) is briefly trained off-line to predict subject elbow angle based solely on recorded EEG activity. Upon resuming recording, the system is accurately able to predict the test subject's elbow angle in real-time. If employed in a robotic system, this BCI would have applications in rehabilitation robotics, search and rescue, tele-robotics and exoskeleton research.

Rapid Prototyping of a Single-Channel Electroencephalogram-Based Brain-Computer Interface

by
D. Brooks Adcock

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the requirements for the degree of
Masters of Science

Biomedical Engineering

North Carolina State University
Raleigh NC, 27695
November 2006

approved by:

Lianne Cartee
Co-Chair of Advisory Committee

John Muth

Edward Grant
Chair of Advisory Committee

Dedication

I dedicate my thesis to my partner in life, Liz Kirby. Liz's love and support, as well as her example as a scientist have provided me with the guidance and motivation needed to complete this project.

Biography

D. Brooks Adcock Jr. was born October 5, 1982 in Durham, NC. He received bachelor degrees in Electrical and Computer Engineering from North Carolina State University in 2005. During this time he certified as an Emergency Medical Technician and rose to the rank of Capitan and Board Member at a local EMS squad. Based his extremely positive experience in applied medicine, Brooks pursued a Masters degree in Biomedical Engineering from NCSU with the Center for Robotics and Intelligent Machines. Brooks' research interests are in blurring the distinction between human and machine intelligence.

Acknowledgments

I have been blessed by being surrounded by encouraging and supportive people. I would particularly like to thank. . .

- My advisor, Dr. Edward Grant for allowing me to operate on the lunatic fringe. I still can't believe he let me do this project. No other advisor would have supported me this much in my eccentric research interests. For this I will always be grateful.
- My committee, Dr Grant, Dr. Cartee and Dr. Muth. I did this project on a tight time frame and my committee was generously accommodating and tolerant.
- Nancy McKinney, Peggy Olive and Debra Paxton. These people went above and beyond to get various documentation through at lightning speed. Without such diligent people I could not have finished. Thank you.
- My subjects. These people sacrificed their time and a bit of hair to further my research. Thanks!
- The CRIM team, Leo Mattos, Kyle Luthy, Carey Merritt, Meghan Hegarty, Matt Craver.
- My family, my significant other, Liz, and the kittenz for their love.
- My Mom for giving so much time to help me edit.
- My Dad for being volunteering his time.
- My biomedical role-models, Daniel Cline (EMT-P), Dr Brent Myers (MD) for teaching me the importance of medicine.

Contents

List of Figures . . . . **viii**

List of Tables . . . . **ix**

Common BCI Abbreviations . . . . **x**

1 Introduction **1**
- **1.1** Motivation 2
- **1.2** Project Goals 3
- **1.3** Thesis Outline 3

2 Review of Brain-Computer Interface Literature **5**
- **2.1** Introduction 5
- **2.2** Anatomical Disambiguation 5
- **2.3** Signal Recording 7
- **2.4** Feature Extraction 7
- **2.5** Feature Analysis 8

3 Exploited Principles of Neurobiology **10**
- **3.1** Introduction 10
- **3.2** Indicators of Brain Activity 10
  - 3.2.1 Bioelectricity 10
  - 3.2.2 Hemodynamic Response 12
- **3.3** Motor Pathways 14
  - 3.3.1 Prefrontal Cortex 14
  - 3.3.2 Posterior Parietal Cortex 15
  - 3.3.3 Premotor Area (PMA) and Supplementary Motor Area (SMA) 15
  - 3.3.4 Primary Motor Cortex (M1) 15
  - 3.3.5 Spinal Motor Pathways 16
  - 3.3.6 Application 17

4 Design of a Custom Electroencephalogram Amplifier **18**
- **4.1** Introduction 18
- **4.2** Electroencephalogram Design Goals 19
- **4.3** Power Supply 20
- **4.4** Stage 1: Pre-Amplifier 21
- **4.5** Stage 2: Instrumentation Amplifier 23
- **4.6** Stage 3: Band-Pass Filter 23
- **4.7** Stage 4: Post-Amplification 26
- **4.8** Additional Noise Reduction 26
  - 4.8.1 Overview 26
  - 4.8.2 Driven Right Leg 26
  - 4.8.3 Wire Braiding 28
  - 4.8.4 Printed Circuit Board Layout 28
  - 4.8.5 Faraday Cage 28
  - 4.8.6 Lead Selection 29
  - 4.8.7 Skin Preparation 30
  - 4.8.8 Software Filtering 30

5 Data Acquisition **32**

5.1 Introduction 32
5.2 Electroencephalogram Data Acquisition 33
5.3 Arm-Angle Data Acquisition 34

6 **Brain Computer Interface Software** **35**
6.1 Introduction 35
6.2 Design Goals 36
6.3 Data Acquisition Procedure and Input Filtering 38
  6.3.1 Data Buffering and Display Threads 38
  6.3.2 Overlapped Band-Pass Filtering 38
  6.3.3 Arm Channel / Predicted Channel Acquisition 41
6.4 Feature Extraction 42
  6.4.1 Component Wave Extraction 42
  6.4.2 Power Density and Autoregressive Coefficients 42
6.5 Learning Paradigms 46
  6.5.1 Bayesian Networks 46
  6.5.2 Linear Discriminant Analysis 46
  6.5.3 Support Vector Machines 47
  6.5.4 Mutual Information 47
  6.5.5 Neural Networks 48
6.6 Neural Network Architecture 48
  6.6.1 Input Size 49
  6.6.2 Hidden Layer Size 49
  6.6.3 Training Algorithm 50
6.7 Output Smoothing 51

7 **Experiments and Results** **53**
7.1 Introduction 53
7.2 Experimental Setup and Procedure 53
7.3 Subject Trials 56
  7.3.1 Subject 1 56
  7.3.2 Subject 2 57
  7.3.3 Subject 3 59

8 **Conclusions** **63**
8.1 Introduction 63
  8.1.1 System Safety 63
  8.1.2 Electroencephalogram Predictive Base 63
  8.1.3 System Cost 63
  8.1.4 Computer Platform 64
  8.1.5 Predictive Capacity 65
  8.1.6 Final Remarks 65
8.2 Future Work 65
  8.2.1 Immediate Improvements 66
  8.2.2 Hardware Improvements 66
  8.2.3 Software Improvements 66
  8.2.4 Software Output 67
  8.2.5 Suggested Experiments 67

References **73**

Appendix A: EEG Circuit Schematic and Layout **74**

Appendix B: BCI Software Code **76**

Appendix C: IRB Application **97**

List of Figures

1 Brodmann’s Areas 6
2 System Block Diagram 9
3 Action Potential 11
4 Invasive and Non-invasive EEG Sensors 11
5 SQUID 13
6 fMRI Image 13
7 The Brain’s Motor Centers 15
8 Homunculus 16
9 EEG Brain Activation During Motor Movement 17
10 International 10-20 EEG System 18
11 Schematic: Power Supply and Layout 21
12 Schematic: EEG Front-end Circuits 22
13 Chebeshev and Butterworth Frequency Response 24
14 Schematic: EEG Circuit 25
15 Schematic: DRL Circuits 27
16 60Hz Noise Rejection by DRL 27
17 Example of Signal Aliasing 32
18 Photo of Elbow Angle Sensor 35
19 BCI Graphical User Interface 37
20 Software Filter Error 40
21 Recorded Alpha and Beta Waves 42
22 Example of a Hilbert Transformation 43
23 Power band trace computed from an EEG 44
24 Instability of the Hilbert Transform 45
25 Spectral Power Density Plot 45
26 An Artificial Neural Network 48
27 An Adaptive Filter Network 49
28 A Neural Network Prediction of Arm Movement 52
29 Output smoothing 53
30 EEG Electrode Placement 55
31 Subject 1: Graphical Results 58
32 Subject 2: Graphical Results 60
33 Subject 3: Graphical Results 62

List of Tables

1 Abbreviations Used In This Paper . . . x
2 Filter PRO Specifications . . . 25
3 Neural Network Trials . . . 51
4 Vulnerable Populations . . . 54
5 Prohibited Medical Conditions . . . 54
6 A Neural Network training iteration performed on Subject 1’s data. . . . 57
7 A Neural Network training iteration performed on Subject 2’s data. . . . 59
8 A Neural Network training iteration performed on Subject 3’s data. . . . 61
9 Expenses . . . 64
10 Computer Configuration used for this project . . . 64

Common BCI Abbreviations

Table 1: Abbreviations Used In This Paper

AC	Alternating Current
AR	Autoregressive
BCI	Brain Computer Interface
BMI	Brain Machine Interface
CMRR	Common-Mode Rejection Ratio
DAQ	Data Acquisition
dB	Decibel
DC	Direct Current
DRL	Driven Right Leg
EEG	Electroencephalogram
EM	Electromagnetic
EOG	Electroocculogram
FFT	Fast Fourier Transform
fMRI	Functional Magnetic Resonance Imaging
GUI	Graphical User Interface
ICA	Independent Component Analysis
IRB	Institutional Review Board
LDA	Linear Discriminant Analysis
MEG	Magnetoencephalogram
PCA	Principle Component Analysis
PET	Positron Emission Tomography
RF	Radio Frequency

1 Introduction

The defining characteristic of humankind has been its ability to evolve technologically as opposed to physically. Anthropologists state that as far back as seven million years ago, the ancestors of humans were creating primitive tools to extend their intrinsic ability [1]. Modern tools have gone so far as to allow us to rectify many of our own pathologies. Through technology, humans have all but absolved themselves from the Darwinian rat-race [2].

Despite the tremendous advances humans have made augmenting themselves, certain technological borders have barely been breached. We adorn ourselves with various devices that allow us to accomplish remarkable things. However, few technologies have focused introspectively. The true goldmine of human technological evolution will be in intrinsic human enhancement. Ambulatory medical devices that enhance or repair a human's natural ability have barely been explored but offer great promise [2].

Pathology has driven research in intrinsic human improvement. The increased incidence of heart disease in the United States has motivated the development of artificial hearts and pacemakers that restore cardiac function [2]. The deaf and blind have recently experienced relief with implantable cochlear and retinal stimulators. Brain implants show promise for relieving central nervous system (CNS) deficits [2].

Pathology aside, these technologies, if proven safe and reliable, will be able to take mankind over new horizons. Current cochlear implants, for example restore hearing in the natural 20-20kHz range, but have the potential to adjust human hearing to any frequency range. Likewise, retinal stimulators could potentially allow people to see selectively in infrared or night-vision. Optical overlays could even be digitally inserted. This could allow surgeons to see inside the body without having to make any extra incisions.

1.1 Motivation

Among the most exciting potential for human-augmenting technologies are enhancements to the human central nervous system (CNS). These technologies have the ability to enhance the lives of those with and without pathologies alike.

Traumatic and pathological injuries to the CNS create a sophisticated problem. Affronts to the CNS such as head trauma, spinal trauma, stroke, Parkinson's disease, and Huntington's disease usually result in irreparable damage [3, 4]. The human CNS, unlike the peripheral nervous system (PNS) does not readily repair itself. This implies that in order to fully recover from a severe CNS injury such as a stroke or spinal cord injury, artificial augmentation is needed. In the United States there is a large market for such augmenting devices [5]. According to the Spinal Cord Injury Information Network there are approximately 11,000 new spinal cord injuries (SCIs) each year [5]. This network claims that there may be 300,000 people in the United States alone currently living with some sort of SCI.

Augmentation of an injured CNS involves three steps. First, the CNS area that provides input to the injured area is identified and is used as input to the augmentation device. The processing for the injured area must be mimicked by the augmenting device. Finally, the effector that is controlled by the injured CNS area must be actuated. This actuation may involve replacing the end-effector or actuation device of the CNS or PNS beyond the site of injury [6].

The risks associated with direct neural interfacing are significant and thus make elective neural interfacing unattractive. However, a person with a pathology can afford to take greater medical risks, thus making such an interface a more attractive proposition. A person choosing to augment their CNS electively, would have to be assured that the procedure would be safe, which is now possible using several non-invasive interfacing procedures. These are an attractive option for the injured and uninjured alike [7].

This research explores the creation of a safe, non-invasive brain-computer interface (BCI). This interface is designed to control a one-degree-of-freedom kinematic variable using only scalp electroencephalography (EEG). The application of this interface will lend mobility to someone who is completely or partially paralyzed. The system will be used in the rehabilitation of stroke or sports injury patients. Outside of pathological augmentation, the system can be used to control an exoskeleton, allowing a person to lift multiple times their weight [8]. Although the potential applications are many, the primary goal of the research was to create a platform on which further research at the NCSU Center for Robotics and Intelligent Machines (CRIM) could be based.

1.2 Project Goals

The goal of this project was to create an EEG-based BCI platform for further research at the NCSU CRIM. The system was designed according to the following criteria.

1. The system must be non-invasive and safe for use by an untrained person.
2. The system must be scalp electroencephalogram based.
3. The system must be cost-effective.
4. The software must run on a standard PC.
5. The system must predict a one-degree-of-freedom kinematic variable from EEG activity alone.

1.3 Thesis Outline

This thesis describes the construction of an EEG-based BCI. In Chapter 2, the current BCI literature is reviewed so that the reader can put this work in context of the current research. Chapter 3 reviews neurobiology as it applies to BCI research, to provide the reader with an understanding of exactly what being attempted. This information is crucial to understanding the instruments and algorithms developed to predict arm movement from brain activity. The design of the BCI is then explored in the order of the signal processing steps. This signal order is followed so that the reader can understand the logical progression of the individual signal processing steps.

Chapter 4 describes the design of the EEG used for acquiring brain signals. Motivation is provided for the EEG design and its features are explored in signal order. Chapter 4 ends with a review of additional EEG signal processing tools for BCI applications.

Chapter 5 describes the methods used to acquire and store the recorded signals. EEG software is discussed in Chapter 6. Data gathering is described, followed by input filtering. Methods for computing neural synchronization are then explained. Finally, the learning algorithms considered for this project are explored.

The verification procedures and experiments are delineated in Chapter 7. Chapter 8 draws conclusions about the system and introduces ideas for future work to improve this device.

2 Review of Brain-Computer Interface Literature

2.1 Introduction

Brain-computer interface (BCI) research is a permutation of functional neuroanatomy research [9–11]. The two fields are so closely related that they are often indistinguishable. As a general rule, studies of the applied use of neural activity can be categorized as BCI research. Studies that explore the temporal and spatial relationships between neural events and external events fall into the category of functional neuroanatomy [9, 12]. Many laboratories such as the McKeown laboratory and the Georgopoulos span both categories and are renowned by both the BCI and neuroanatomy communities [12].

In this chapter, categories of research are explored in signal pathway order. First, research in the identification of the spatial-temporal location of signals is discussed [12]. The preferences of various laboratories for recording these signals is then explored. Practices for extracting relevant data features from recorded signals are reviewed [13, 14]. Finally, studies using different feature analysis techniques are presented.

2.2 Anatomical Disambiguation

The first information that a BCI researcher needs to know is what spatial areas of the brain are responsible for the phenomenon under study. Korbinian Brodmann was one of the first people to make a scientific attempt to categorize the areas of the brain. He used tissue samples to segregate the brain into its distinct components. Brodmann's areas, named for him, are now being confirmed to be of actual significance by more advanced scientific methods such as fMRI, PET, EEG and MEG. [12] Two of the biggest names in the field of anatomical disambiguation are Martin J. McKeown, M.D. and Apostolos P. Georgopoulos, M.D., Ph.D. Both researchers study the cortical activity associated with motor movement [9–11].

Martin McKeown has made extensive use of independent component analysis (ICA) to extract relevant spatial-temporal information from various brain recording modalities including fMRI and EEG

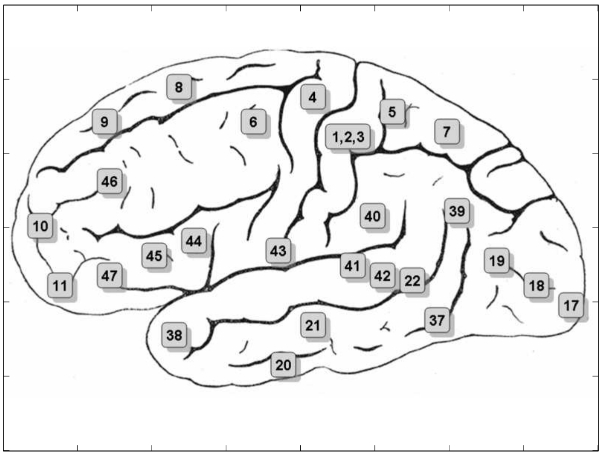


Figure 1: Brodmann's Areas are groups of similar cells. These areas tend to be unique in function. (figure from [15])

[10, 11]. McKeown's work has both proved the merit of ICA and demonstrated the coupling between the motor areas of the brain and body movements [10, 11].

Apostolos Georgopoulos [12] has gained widespread recognition for his analysis of cortical mapping. He is reported to have demonstrated that neurons of the primary motor cortex (M1) encode for the force and direction of a limb. Further research in noninvasive recording techniques such as MEG has shown the efficacy of such methods for clinical use [9].

Other researchers have tackled the problem of identifying relevant spatial-temporal locations as well. Chih Hung has done work on using ICA to decompose multiple-lead EEG records into the most relevant components [16]. This work demonstrates the importance of the C3 and C4 leads in the International 10-20 EEG System for identifying motor movement [16]. Gert Pfurtscheller has explored using autoregressive parameters to segregate left and right motor imagery [17]. This is a somewhat novel approach to signal decomposition. Justin Sanchez in Miguel Nicolelis' laboratory at Duke University demonstrated that statistical culling of data from implanted neuron-recording arrays can actually improve performance by selectively identifying the specific neurons that encode for a movement [18].

2.3 Signal Recording

The Nicolelis laboratory at Duke University is an excellent example of how a research center selects a recording technique and researches around that paradigm. The Nicolelis laboratory is known for its work in implantable electrodes in primates [13]. The technique around which their research is based involves implanting electrodes into the primary motor cortex and other supporting cortical areas [13, 18–20]. The Nicolelis laboratory has done quite a bit of work improving these electrodes [20], improving acquisition from these electrodes [18], as well as optimizing the algorithms used to make kinematic predictions of primate arm movement [13].

The Pfurtsheller laborotory from the University of Graz, Austria, has taken an alternative recording approach. Pfurtsheller has focused on improving non-invasive methods [21, 22]. The apparent goal of the Pfurtsheller laboratory has been to create a BCI that is available to the general public. Significant research has been put into making systems that do not require excessive biomedical equipment [14, 23]. Such non-invasive systems are extremely valuable.

2.4 Feature Extraction

Second in importance to knowing the spatial-temporal location of relevant brain signals, knowing the encoding scheme used by that specific area of the brain. For example, the light sensing retinal ganglia cells of the eye are excited by the absence rather than presence of light [12]. These encoding modalities dictate how the brain signal is extracted so that coherent information can be given to the feature analyzer.

Two of the most common feature extraction techniques are the Hilbert transform and autoregressive (AR) coefficients. Hilbert transformation provides the signal envelope and AR coefficients allow calculation of the band energy ratio. Papers from the Nicolelis and Pfurtsheller laboratories use both techniques [14, 19, 23, 24].

Shiliang Sun analyzed: signal RMS power, spectral centroid, bandwidth, zero-crossing rate, spectral roll-off frequency, band energy ratio, and delta spectrum magnitude as feature extraction techniques for binary classification [25]. Sun supports spectral centroid as the superior method for binary classification [25].

2.5 Feature Analysis

BCI systems use learning algorithms and feature analysis techniques. Evaluation of learning algorithms has been done by Rezaei [26] and Hung [16]. Hung's report of Bayesian networks as an efficient method of categorization has resulted in their popularity and use by several groups [16, 24, 27, 28]. For analysis of non-categorical data, a method must be used that allows numerical prediction. Nicolelis *et al.* report using dynamic-configuration recurrent neural networks for exactly this purpose [13].

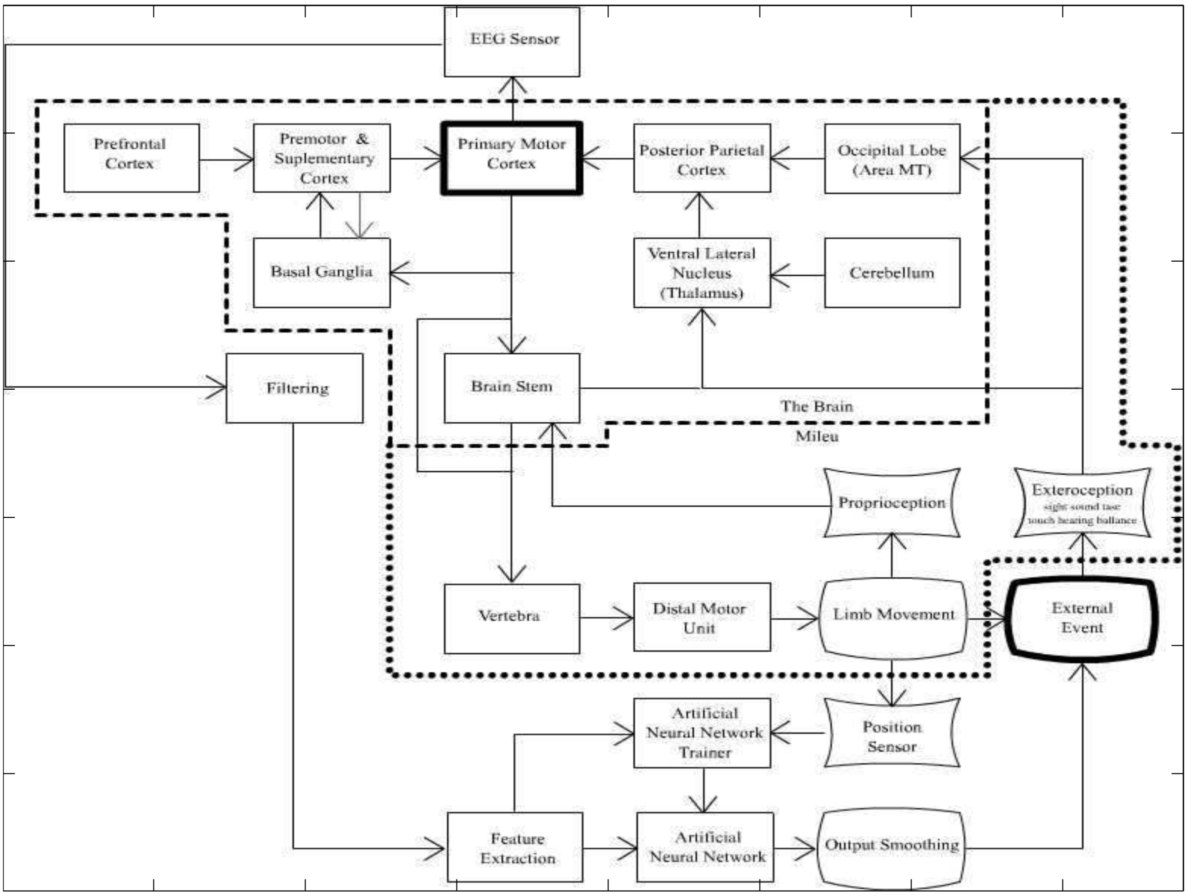


Figure 2: Categorized brain signals are filtered then used for feature extraction. Features are fed to learning algorithms that predict desired variables. Output smoothing removes obvious artifact from the predictions

3 Exploited Principles of Neurobiology

3.1 Introduction

Neuroscience studies of functional anatomy provide BCI researchers with the tools needed to interpret brain activity. There is a fine line between functional anatomy studies and BCI studies. Functional neuroscientists use medical imaging and learning algorithms to identify spatial-temporal brain patterns correlating to known external events. Alternatively, BCI researchers use medical imaging and learning algorithms to interpret spatial-temporal brain patterns to drive external events. This chapter briefly discusses the neural anatomy and physiology that is interpreted by the BCI to drive external events.

3.2 Indicators of Brain Activity

3.2.1 Bioelectricity

The excitable cells of the body use electricity and ionic currents to transmit and store information. Excitable cells can be found all over the body including in brain, heart and muscle tissue. In the brain, neurons and glia are the primary excitable cells. Neurons are highly adaptive cells capable of learning. Glia are the cells that support neuronal activity. The electrical activity of neurons and glia can be measured as a means of interpreting brain activity. When neurons are at rest, they typically exhibit a resting electrical potential of around $-70mV$ [29]. When electrically excited, neurons permit ions to rapidly cross the cell wall [29]. The voltage level rises to a peak around $40mV$.

Electroencephalography Electroencephalogram (EEG) electrodes can measure the electrical activity of neurons. Electrodes can be invasively implanted or non-invasively adhered to the scalp. A single electrode measures the activity of a group of neurons. When local neurons fire in phase, their post-synaptic potentials create a large enough ionic potential field to be measured by electrodes [31]. The greater the degree of synchronization, the greater the potential.

It is typically the post-synaptic potential rather than the action potential (AP) spike that is read.

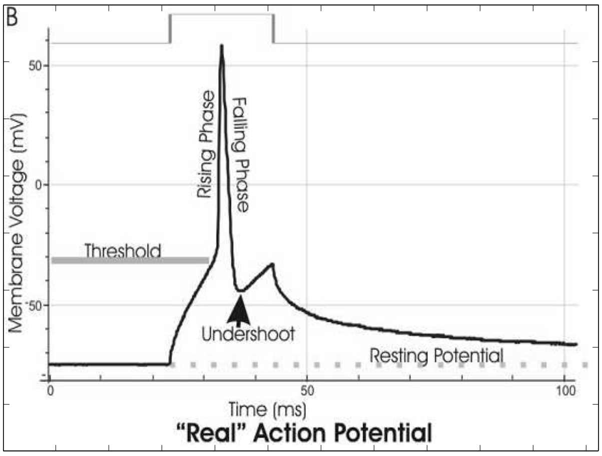


Figure 3: Electrical impulses generated by cells (action potentials) are the basis of nerve function. They can be recorded by brain interfaces to predict thought.
(figure from [30])

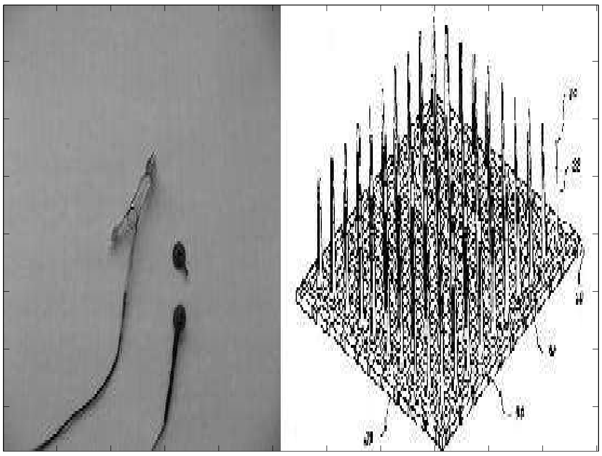


Figure 4: Invasive and non-invasive sensors can be used for EEG recording.(figure (right) from [32])

AP spikes are short lived and are thus less likely to overlap. In order to read the actual spike, nerve sleeve electrodes are needed [33]. Using these electrodes is highly invasive.

EEG is a popular method for BCI [14, 16, 17, 21, 23–28, 34, 35]. EEG has the advantage of having arbitrarily high temporal resolution. Spatial resolution can be theoretically increased to the single neuron level. Unfortunately in EEG, invasiveness accompanies spatial resolution. Electrodes placed in the brain can provide excellent data [13, 18–20]. Non-invasive scalp EEG signals provide only a rough view of the underlying neural activity. It has been said that interpreting scalp EEGs is like trying to listen into a single conversation in an sports stadium by sitting on the roof with a stethoscope.

Magnetoencephalography The orthogonal correlate to the electric field is the magnetic field. Movement of electrons or ions results in the creation of magnetic fields. In neuronal dendrites, ionic current flows are sufficient to create magnetic fields in the femtotesla range [36]. Magnetic fields are not distorted by cranial tissue as are electric fields [9]. MEG is superior to scalp EEG when evaluated on the basis of signal quality [9].

MEG was the initial, albeit short-lived, choice for brain signal acquisition. Initial research into scalp EEG quickly unveiled its shortcomings in spatial resolution. MEG offered better spatial resolution with temporal resolution equal to EEG. Georgopoulos' paper on MEG interfacing made the choice look promising [9]. But, it was soon learned that MEG is prohibitively expensive. MEG acquisition requires the use of superconducting quantum interference devices (SQUIDs) to read the MEG signal. The further requirement of an electrically sterile aluminum and mu-metal enclosed room was also a deterrent to using MEG.

3.2.2 Hemodynamic Response

Two standard methods for evaluating functional neuroanatomy use hemodynamic responses to identify cortical areas of increased function during a given task [12]. Research with these tools has largely provided us with the knowledge of how the human cortex is organized [12]. Primarily used for

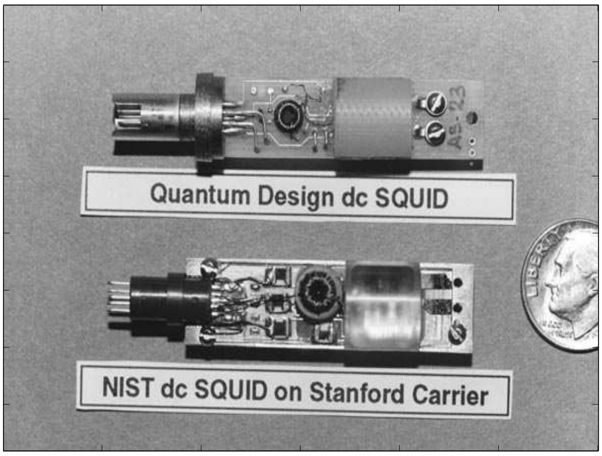


Figure 5: SQUID's are sensitive magnetic field sensors used for MEG.(figure from [37])

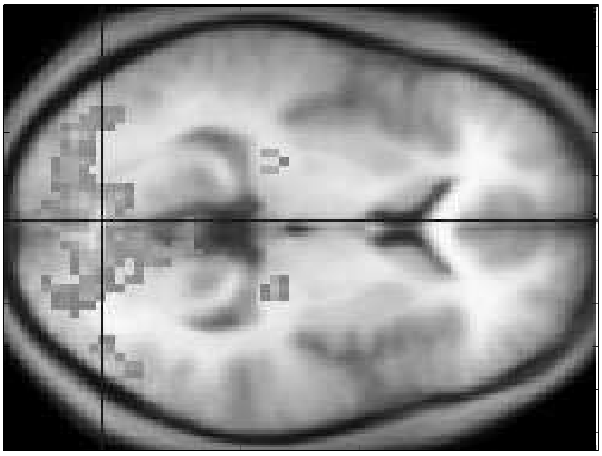


Figure 6: fMRI can be used to acquire high spatial resolution, low temporal resolution brain activity maps. (figure from [40])

clinical medical diagnostics and functional neuroscience research, these tools can also be used for BCIs but are currently too unwieldy to be practical outside of a clinical environment [12, 38].

Functional Magnetic Resonance Imaging fMRI is a high spatial resolution method for measuring brain activity. A MRI machine measures brain activity by identifying the increase in blood flow that occurs as a result of increased neural activity [12]. The fact that blood is diamagnetic when oxygenated and paramagnetic when deoxygenated makes this possible [39]. Although fMRI has a spacial resolution of about 3-6mm, its temporal resolution is in the order of seconds [39]. Honda has developed a fMRI based BCI that functions despite this limitation [38]. For most BCI applications, however, temporal resolution, size, cost, and magnetic exposure risks make fMRI impractical.

Positron Emitission Tomography An alternative to fMRI is positron emission tomography (PET). In PET, radioactive markers are injected into the blood supply. As the marker decays, positrons are emitted. Differential positron emission indicates the relative amount of blood located in a given area. PET is comparable to fMRI for BCI applications. PET trades magnetic field exposure for radioactive exposure. Where fMRI is simply not an option for subjects with ferrous metals in their body, PET can still be used.

EEG, MEG, fMRI and PET were all considered as potential means of brain monitoring for this project. The use of MEG, fMRI, or PET would have precluded making a portable, cost-effective system for use by future researchers in the CRIM laboratory. Scalp EEG, although lacking in spatial resolution, meets all the design criteria this project. Should higher spatial resolution be needed for future work, MEG, fMRI, PET, or invasive EEG might be used.

3.3 Motor Pathways

Brain-computer interfaces emulate the natural motor pathways of the central nervous system. A complete understanding of the functional anatomy of the motor pathways is required for creating a functional BCI. Studies using fMRI, PET and implanted electrodes reveal that certain brain areas can be used to predict motor movements [9, 12, 13]. The raw data is typically analyzed with statistical methods such as principle component analysis (PCA), independent component analysis (ICA) or linear discriminant analysis (LDA) to reveal how strongly each area of the brain correlates to motor activity [16, 34, 41]. It is consistently demonstrated that despite individual variations amongst subjects, specific brain areas are related to certain functions.

3.3.1 Prefrontal Cortex

The prefrontal cortex is shown to be involved in the highest, most abstract levels of motor planning. This area is bilaterally connected via the corpus callosum.

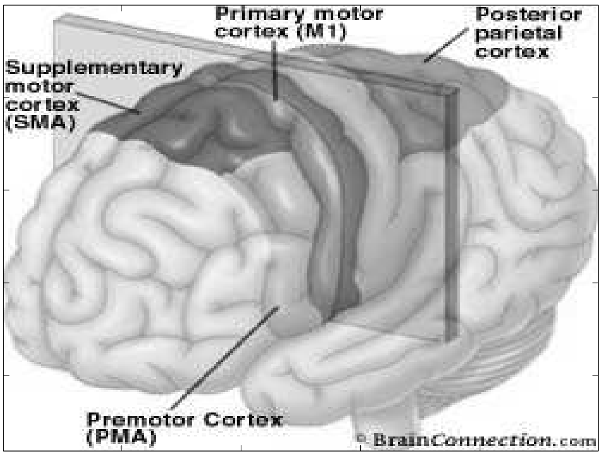

Primary motor cortex (M1)
Posterior parietal cortex
Supplementary motor cortex (SMA)
Premotor Cortex (PMA)
© BrainConnection.com

Figure 7: The brain's motor areas connect to allow coordinated movement.
(figure contains copyrighted material from [42])

3.3.2 Posterior Parietal Cortex

The posterior parietal cortex (PPC) provides the motor areas of the brain with the environmental information needed to formulate motor movements. The PPC consists of two areas, Brodmann's area 5 and area 7. Brodmann's Area 5 of the PPC receives information from the primary sensory cortex (S1) relating current body position and tactile information. Brodmann's Area 7 of the PPC receives visual feedback from the occipital lobe. See Figure 1. [12]

3.3.3 Premotor Area (PMA) and Supplementary Motor Area (SMA)

The PMA and SMA (Brodmann's Area 6) deal with motor planning. Studies with monkeys show that intended movements are queued in these areas here until they are executed. Artificial electrical stimulation of these areas results in complex motor movements. The PMA and SMA take inputs from the basal ganglia. The PMA feeds to the reticulospinal motor neurons which synapse with proximal motor units. The SMA feeds directly to distal motor units. [12]

3.3.4 Primary Motor Cortex (M1)

The primary motor cortex (M1) is believed to be the most direct link to motor movements. Artificial electrical stimulation is shown to instigate simple motor movements at a minimum electrical threshold.

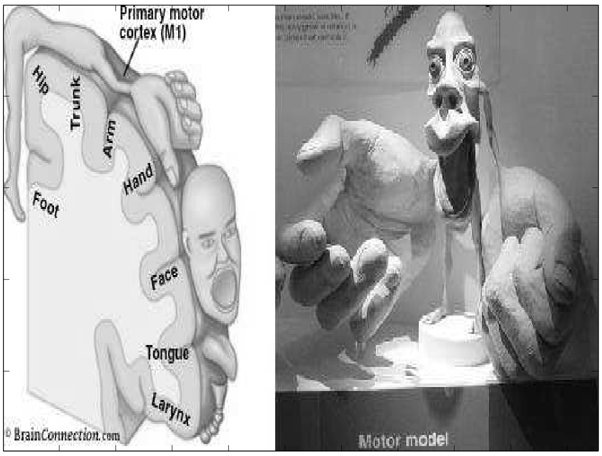


Figure 8: The Homunculus represents what a human would look like if each body part was proportional to the size of the brain area responsible for its control.
(figures contain copyrighted material from [42, 43])

Layer V pyramidal neurons (Betz cells) are the primary transmitters of motor commands to lower motor neurons from M1. Inputs to Betz cells are from other cortical areas such as the PMA and SMA. Inputs also come from the cerebellum and pons via the ventral lateral nucleus of the thalamus. The most recent thoughts about Betz cells are that they encode for the force and direction of motor movements. This implies that decoding to specific motor units is done at the level of the brain stem or spinal cord. [12]

It is also of note that M1 adheres to a consistent cortical mapping among species. The human M1 is consistently mapped according to the “homunculus” [12]. Artificial stimulation of a given location area mapped by the homunculus results in a motor movement of that area. This consistency can be employed to the advantage of the BCI researcher.

3.3.5 Spinal Motor Pathways

Two pathways travel from the cortex of the brain to the effectors of the body. These pathways are the lateral and ventromedial pathways. The lateral pathways are under direct cortical control, whereas the ventromedial are filtered by the brain stem. The lateral pathways carry information about conscious movements and the ventromedial pathways mitigate posture and balance. Both pathways descustate or cross from left to right. For this reason, the right motor cortex controls the left side of the body and the left side of the brain controls the right side of the body. Finally at the level of the spinal vertebra, the

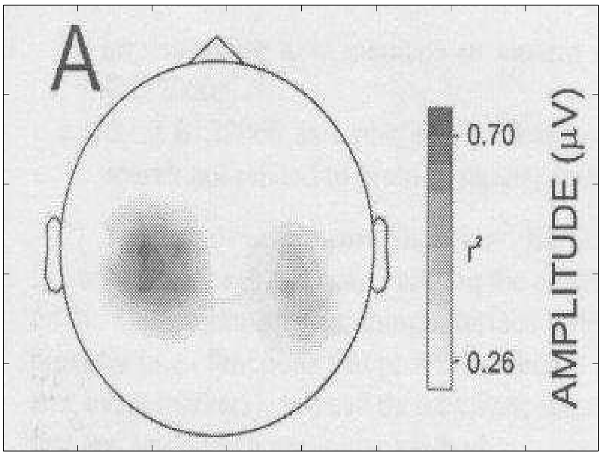


Figure 9: EEG recordings of brain activity during left- and right-sided motor movements. (figure from [35])

pathways synapse with lower motor neurons.

3.3.6 Application

It is consistently shown that M1 provides the best predictor of motor movement [16]. Other areas such as the PMA and SMA can enhance predictive abilities of a system when used in conjunction with M1 [13]. Nicolelis places implantable electrodes in M1, the PMA and other locations to improve predictive quality [13]. The disadvantages of using locations other than M1 are that the encoding of the motor movement becomes increasingly abstract. The least abstract reading would be at the level of motor unit nerves themselves, but this would no longer be a brain interface but more like an electromyogram interface.

Placement of EEG electrodes over the hand-arm motor region of M1 results in the best readings of arm activity. Conveniently, this location corresponds to the C3,C4 site on the International 10-20 EEG system. Reading is taken on the contralateral side of the arm to be predicted. In this BCI, left arm movement was predicted, thus the right scalp lead, C4, was used for EEG acquisition.

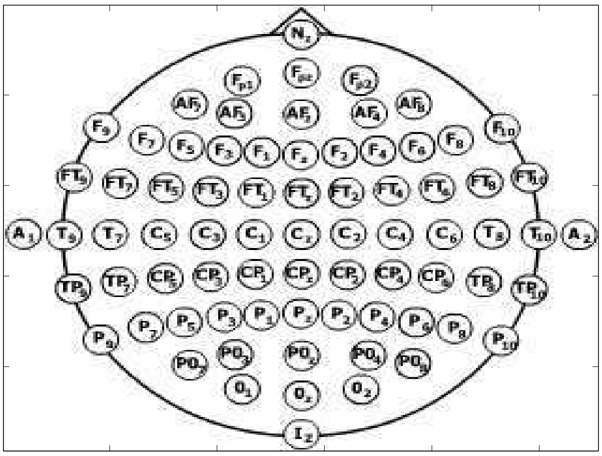


Figure 10: The International 10-20 EEG System is a method of EEG lead placement.

4 Design of A Custom Electroencephalogram Amplifier

4.1 Introduction

Integral to the BCI system is how brain activity is measured. The human central nervous system is an orchestra of interconnected neurons, glia, and support tissue that broadcast multiple indicators of mental activity. A spectrum of tools exists for measuring these indicators ranging from bulky, costly but accurate fMRI, to the portable, inexpensive, and inaccurate scalp EEG. All of these tools create a window into the mind of a test subject [12], yet none of the tools serves as a ubiquitous interface for all BCI applications. PET and fMRI, for example, both provide a moderate-to-high resolution, real-time view of brain activity, but neither is portable enough to be used outside of a clinical environment. Scalp EEGs are highly portable but are capable only of detecting gross brain activity. This inadequacy may be overcome by replacing scalp EEG with scalp MEG.

Scalp MEG is similar to scalp EEG with the exception that the sensors read magnetic as opposed to electric fields emitted through the skull. The insusceptibility of magnetic fields to convolution by cranial tissue makes scalp MEG readings far more accurate than scalp EEG [9]. Yet current sensors for MEG are prohibitively expensive. Groups like the Nicolelis laboratory at Duke University have chosen an alternate solution using implantable EEG electrodes that are physically embedded in the brain [13]. This solution is highly effective but precludes use by the general population. Furthermore, communication from a brain implant is limited to the area in which it is implanted. More often than not, signals from multiple areas of the brain are desirable. The Nicolelis laboratory deliberately targets carefully chosen brain areas; a general BCI user may require a more adaptable solution [13]. The research of Pfurtscheller *et al.*, reinforces the merit of an adaptable system. Pfurtscheller et al. cite their work showing that multiple-lead, subject-customizable, EEG-based BCIs achieve higher accuracy than single-lead EEG-based BCIs [24].

In order to be practically applicable, a BCI system must not be overly expensive, bulky, invasive, or specialized. These constraints are in direct opposition to recording quality, which is also critical for a functional BCI application. Until technologies such as fMRI and MEG become more portable and affordable, the only brain-interface solution that has any practical application for applied BCI is scalp EEG. At roughly 2000 US dollars per system, EEGs are the cheapest solution but are hardly a bargain. EEG experts such as Stephen Luck assert that even with a top-of-the-line system, experiments must be performed in an electrically sterile environment [44]. Such requirements are impractical for most non-clinical BCI applications.

The need for a low-cost, highly-portable, noise-impervious EEG system motivates the need to construct a custom EEG amplifier. The advantages of a custom EEG amplifier are numerous. Organizations such as OpenEEG have shown that a custom EEG amplifier can be built for around 200 US dollars. Basic biosignal amplifiers such as EEG amplifiers are not large circuits and can be custom designed to be highly portable. Furthermore, custom filters can be implemented to attack noise bands. A properly designed EEG amplifier could be highly effective for multiple BCI applications.

4.2 Electroencephalogram Design Goals

The BCI design for this project required a low-cost, high-accuracy, noise-impervious EEG amplifier. Future projects may also require some sort of general-purpose EEG, EMG, or ECG amplifier.

Hence, the goal of this phase of the project is to create a general-purpose EEG amplifier with the following features...

- Maximum safety
- Minimal cost
- Maximum noise rejection
- Driven-right-leg active noise cancellation
- Strict band-pass filtering between 0.1Hz and 40Hz
- Maximum portability
- Battery powered
- Single-channel differential analog input
- Multiple analog outputs with varying net gains

4.3 Power Supply

Power for the EEG amplifier comes from an isolated $\pm 6V_{dc}$ supply. $18V_{dc}$ is supplied by two series 9-Volt batteries. Two $6V_{dc}$ voltage regulators (78x06) are cascaded to produce $+0V_{dc}$, $+6V_{dc}$ and $+12V_{dc}$ voltage outputs relative to battery ground. The $+6V_{dc}$ terminal is selected to be $Ground_{analog}$. The other two outputs thus become $+6V_{dc}$ and $-6V_{dc}$ relative to $Ground_{analog}$.

The battery driven $\pm 6V_{dc}$ voltage configuration provides superior safety, noise rejection and amplifier performance compared to wall power. This technique makes the system more immune to 60Hz noise, and most importantly, eliminating the high voltage power connection greatly reduces the probability of an electrical malfunction injuring the subject. Plus / minus voltage supplies also allow for superior rail-to-rail performance in operational amplifiers.

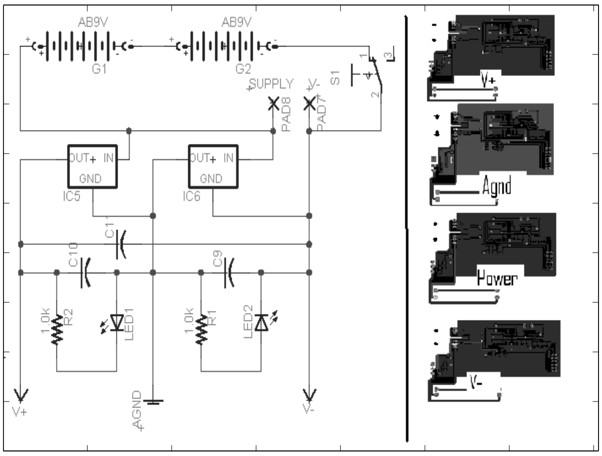


Figure 11: Schematic of the EEG power supply circuit and circuit layout.

Although battery-driven circuits have less intrinsic noise than outlet voltage sources, they are not immune to noise. Natural fluctuations in circuit power consumption result in power voltage fluctuations as voltage regulators adapt to increased power demands. Power voltage fluctuations can be translated to recorded signals via fluctuating amplifier gains. Placing capacitance between the leads of the power supply provides temporary power sources that can respond instantly to fluctuating power demands [45]. These capacitors stabilize the power supply, thus preventing power supply noise from being translated to the signal.

In the PCB layout, power supply traces are routed as far as possible from signal traces in order to prevent EM crosstalk between power and signal. The $Ground_{analog}$ node is expanded into a large plane, filling in gaps between signal traces. This technique grounds external EM interference that would otherwise be induced between signal traces. These noise reduction techniques further stabilize the signal [45].

4.4 Stage 1: Pre-Amplifier

Before the signal entering the body can be amplified by a differential operational amplifier, it must be conditioned. Like any amplifier, a differential op-amp is susceptible to saturation if the input signal is amplified into regions beyond its rails. Saturation most often results in irreconcilable noise as signals are

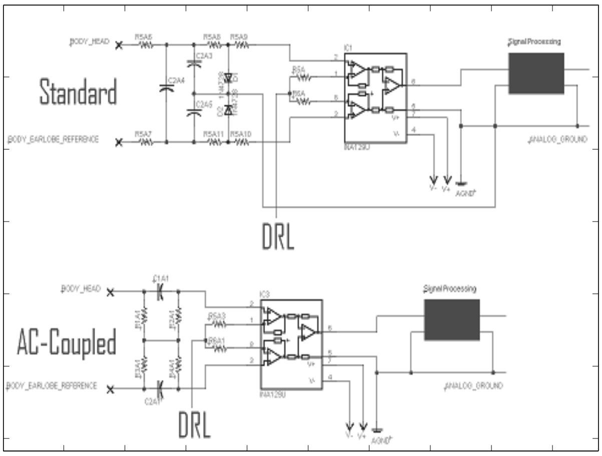


Figure 12: Comparison of AC-Coupled and non-AC-coupled front-ends for biosignal amplifiers

clipped resulting in the addition high-frequency components. Bioelectric signals such as EEG and EKG are extremely small $10\mu V$ to $100mV$. They can be amplified at high gains without risking saturation.

Two common noise signals can be expected on the input lines along with the desired bioelectric signal: high voltage transients and 60Hz noise. High voltage transients are large, slow changes in skin surface potential caused by the accumulation of various types of charge. 60Hz noise is a byproduct of EM coupling between wires in a circuit and the power wires that are abundant in buildings. Other EM spectra are also picked up on wires, however they are far less powerful than 60Hz interference. The general term for such signals is common-mode noise because the interference is imparted equally on all leads. Voltage transients and common-mode noise are both often more pronounced than the desired signal to be read [46]. These noise sources can thus pose a saturation risk [46] for the differential op-amp.

To combat noise-signal sources entering the differential op-amp, front-end circuitry is constructed before the differential op-amp. This circuitry performs two tasks. First, AC-coupling capacitors are placed on each input lead. This removes DC coupled signals such as voltage transients. Secondly, a carefully constructed bias path for the AC-coupling capacitors improves the CMRR, allowing the differential op-amp to eliminate 60Hz noise and other EM interference. Mario Spinelli *et al.* describe such a system in their paper “AC-Coupled Front-End for Biopotential Measurements” [46]. A standard front-end AC-couple has a bias path to ground.

An improved front end eliminates the bias path to ground by creating a reverse bias path. This modification was confirmed experimentally to increase noise rejection greatly.

4.5 Stage 2: Instrumentation Amplifier

The first amplification stage is an INA129U surface-mount, differential, instrumentation amplifier. Like most differential op-amps, the CMRR of the INA129U increases with gain [47, 48].

Table: from INA129 data sheet showing CMRR as a function of Gain

The highest gain possible is selected for the first amplification stage to maximize CMRR. Blocking noise in this way, before it cascades to later stages, means that the noise will not have to be removed later. Gain is set by two equal gain resistors according to the equation $G = 1 + \frac{4.94k\Omega}{R_{gain}}$ [47]. 56Ω Gain resistors are used to set a gain of approximately 100. This results in a CMRR of approximately 125 dB [47]. This amplification is the highest achievable without risking amplifier saturation.

Although the INA129 is available in both PCB surface-mount (SO8) and through-hole (DIP8) packages, surface-mount is preferable for low noise applications [45].

4.6 Stage 3: Band-Pass Filter

The signal passed from the initial amplification stage is a composite of EEG, motion artifact, and electrical interference. The amplitude of the EEG, although significantly amplified, is still only in the single millivolt range. This small EEG signal lies on the spectrum of about 1Hz to 35Hz [27, 49]. Analysis of the signal spectrum at this stage reveals a broad spectrum including a large spike at around 60Hz. This spike is the residual after common-mode noise rejection.

Any signal component outside of the range of 1Hz to 35Hz is undesirable, and is removed. This can be done in software, however few data acquisition units can sample such small signals. Before the signal can be amplified to a point where it can be sampled, undesirable contaminants must be filtered out. Filtering before sampling serves a second purpose. When a signal is filtered, all signal components above half the sampling frequency will result in aliasing. It is thus necessary for all signals above half

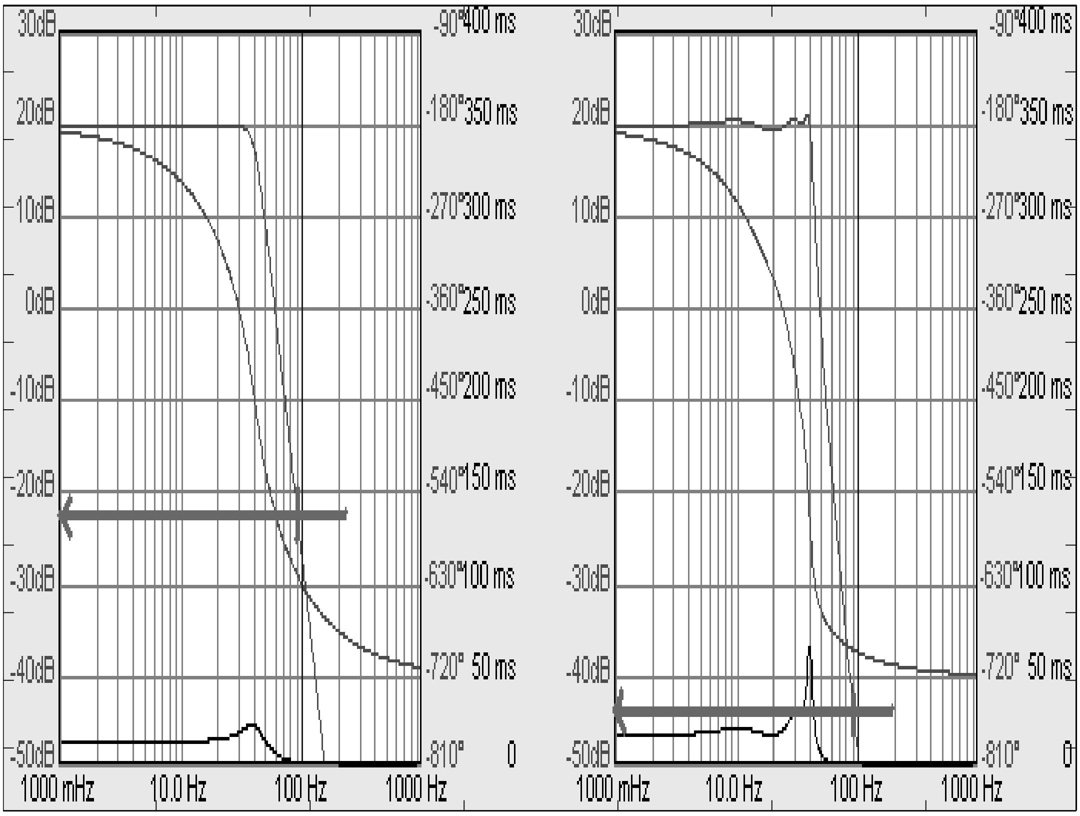


Figure 13: Chebyshev have better greater attenuation (-68dB) at 100Hz than do Butterworth filters (-47dB) for the same order and cutoff frequency

the sampling frequency to be culled to make aliasing negligible. The sampling frequency is commonly referred to as f_s. Half the sampling frequency is commonly referred to as the Nyquist frequency, or f_N.

Many biosignal amplifiers employ first- or second-order Butterworth filters to prepare the signal for sampling because Butterworth filters are maximally flat in their pass band. An alternative to a Butterworth filter is a Chebyshev filter which is not maximally flat in its pass-band. Unfortunately, Chebyshev filters exhibit ripple in the areas immediately around the cutoff frequency. To their advantage, Chebyshev filters exhibit a far steeper cutoff than Butterworth filters of the same order. Butterworth filters exhibit a gradual roll-off.

The filtering specifications of this device call for strict filtering between 0.1 and 40Hz. It is desired to have 60Hz frequencies attenuated by a minimum of 20dB to avoid 60Hz saturation of the amplifier. Signals above f_N need to be attenuated so drastically that they can be negated. 20dB attenuation is equivalent to reducing the signal by a factor of 10. This is sufficient to avoid any aliasing that would occur if frequency components above f_N were large enough to be sampled.

The overall band-pass filter used here is a combination of a 6th-order, low-pass active Chebyshev filter and a passive AC-couple (a capacitor in series). Sixth-order filters are absurdly difficult to design by hand. If a circuit is designed correctly, and the transfer function is computed correctly (both of which are unlikely), components must still be selected to achieve the desired filtering characteristics. Furthermore, component values must be commercially available. Fortunately, Texas Instruments has designed a powerful and free software package called “Filter Pro” that designs filters given desired parameters passed to the function. This program was used and given the following constraints.

Table 2: Specifications given to Texas Instruments’ Filter PRO software for the design of the EEG filter.

Cutoff Frequency	40Hz
Filter Type	Chebyshev
Order	6
Configuration	Multiple Feedback (MFB)
Fully Differential	NO
Resistor Series	E12
Capacitor Series	E6
Gain A	10
Gain B	1
Gain C	1

From the constraints issued to Filter Pro the following filter was designed. When prototyped, the circuit exhibited nearly identical behavior to the Filter Pro prediction. An AC-couple was selected by experimentation. The filter was attached to a spectrum analyzer and a range of AC-couple capacitors were tried until one yielded the desired outcome. Initial designs included an active high-pass filter. It was found on the spectrum analyzer that better performance was obtained by a simple AC-couple.

4.7 Stage 4: Post-Amplification

The final processing stage of the signal pathway is simple amplification. The signal up to this point has been treated as a generic biosignal. Different biosignals require different amplification. EEG, for example, requires far more amplification than ECG because of the fact that ECG signals are, prior to amplification, orders of magnitude larger than EEG. The net gain up to this point is approximately 1000.

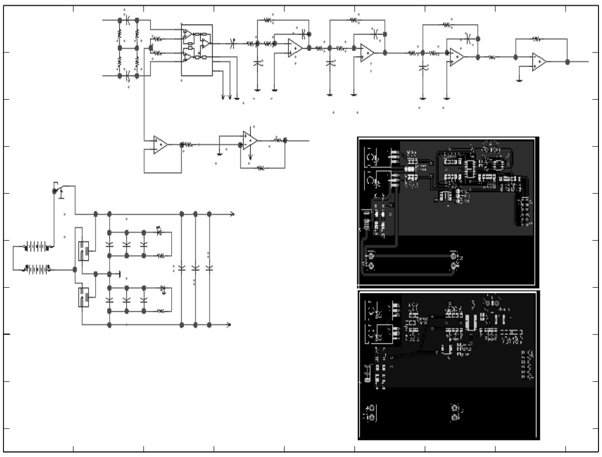


Figure 14: Complete schematic of the EEG circuit.

This is sufficient for ECG, however, EEG requires further amplification by 10 to 100. Such amplification results in an EEG signal with a peak-to-peak voltage of about 2 volts.

The amplifier designed in this project amplifies the signal in this stage by a factor of 10. Output is routed to an output pin. The input to this amplification stage is also routed to a second output pin so that the signal can be analyzed before amplification.

4.8 Additional Noise Reduction

4.8.1 Overview

Even with the construction of a low-noise EEG circuit, many factors can still contribute to noise. EM detection on wires, skin impedance, motion artifact, and PCB noise all degrade signal quality. This section describes techniques to overcome these obstacles to signal integrity.

4.8.2 Driven Right Leg (DRL)

One of the most effective ways of canceling EM interference induced in lead lines is to actively cancel the artifact signal by feeding it back into the lines, out of phase. This is accomplished using a driven right leg (DRL) circuit [49]. There are several designs available for DRL circuits including Webster's classic design [49], an integrating design [46], and a transconductance version [50]. Unfortunately, no

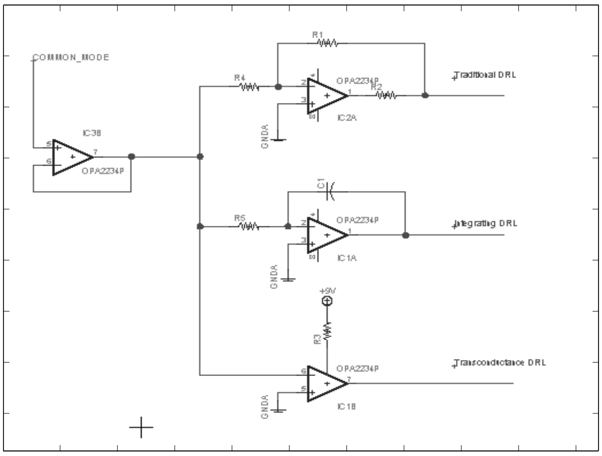


Figure 15: Examples of DRL circuits. Top: standard DRL, Middle: integrating DRL. Bottom: transconductance DRL.

sources were found that thoroughly describe their design criteria.

The general principal of the DRL circuit is to read the common-mode noise signal from between the R_{gain} resistors of the differential instrumentation amplifier. This signal is buffered and added to noise signals from other leads. The signal is then fed back into the body via an amplifier. Experiments showed that Webster's design has superior noise rejection quality. Signal spectral analysis revealed that the act of attaching the DRL electrode to the body virtually eliminates 60Hz noise instantaneously.

For its superior performance, Websters DRL design was implemented to improve the CMRR.

4.8.3 Wire Braiding

A common type of antenna is a wire loop which transduces EM flux. A common problem in instrumentation is that roughly parallel wires that are separated act like antennae for EM interference. To eliminate this problem, wires may be twisted or braided together to the greatest extent possible. This places them in contact and eliminates their ability as a transducer of EM flux. This technique was implemented to reduce noise.

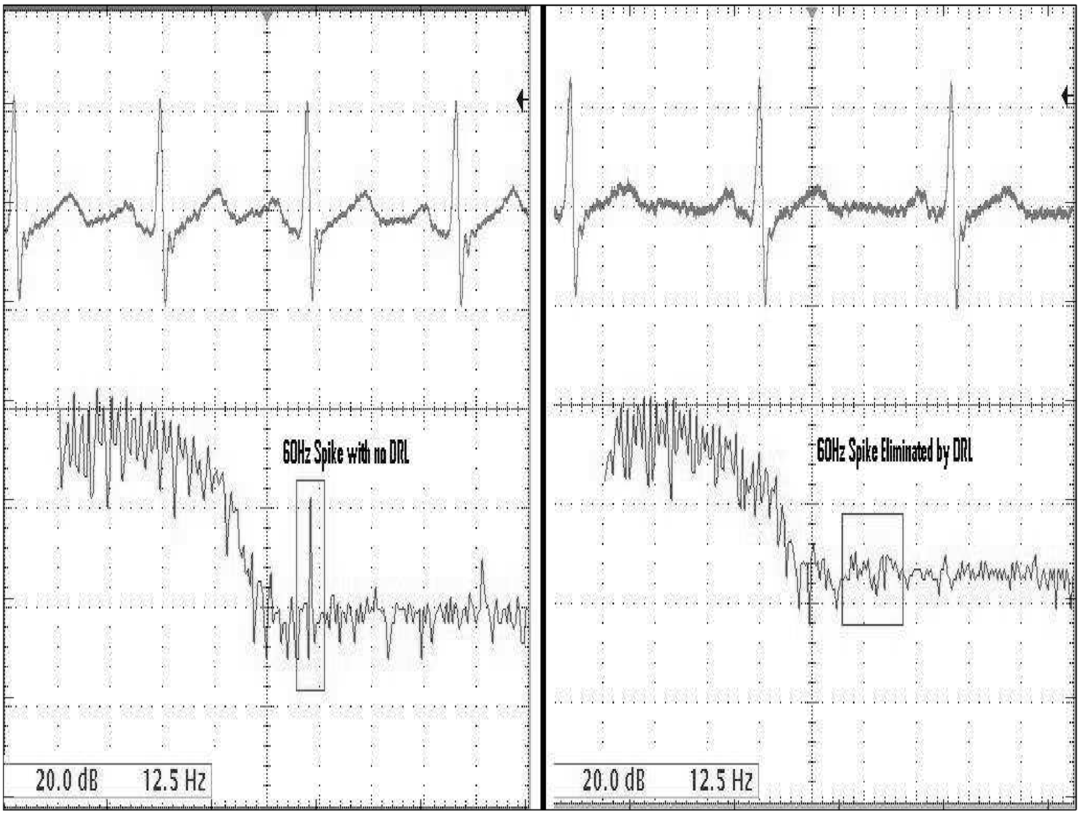


Figure 16: Attaching a DRL circuit eliminates the 60Hz spike in the biosignal trace.

4.8.4 Printed Circuit Board Layout

Circuit layouts have many problems caused by flat traces in close proximity exhibiting mutual capacitance. Traces that are separated by some distance may also transduce EM flux. Carter describes many circuit layout techniques for the minimization of noise artifact [45]. Techniques used in the EEG PCB layout are listed below.

- Traces are orthogonal wherever possible
- The ground plane is large in order to sink, distribute, and negate interference
- Power supply circuitry is as far as possible from signal circuitry
- Signal circuitry is as close together as possible to eliminate distance losses and EM pickup (short traces make poor antennae)
- No trace has a sharp corner and all corners are rounded to prevent high-frequency reflections

- No digital circuitry was placed on the board, reducing high-frequency interference from digital switching

4.8.5 Faraday Cage

A Faraday cage was built into a plastic carrying case. Fine wire-mesh copper screen was used to line the entire inside of the box. The box is groundable through a connection tab on the box's exterior. Experiments showed that this box is highly effective at reducing external EM interference. An experiment was performed where all leads were disconnected from a subject, thus receiving maximum EM interference. EM noise was analyzed under different shielding conditions. The circuit and leads were placed in the grounded cage and the lid slowly closed. As the box was closed, EM was rejected. The Faraday cage is a barrier to outside noise. This cage is not particularly useful because the leads, the primary receptors of interference, must remain outside the box. For this reason the cage was abandoned for the final design.

4.8.6 Lead Selection

Much experimentation was done with leads. Initial attempts to make leads were unsuccessful. Webster's Medical instrumentation illuminates part of the problem. The electrode at the skin interface must be made of specialized conductors. Leads must interface with the skin using certain types of metals that exchange ions for electrons. Silver chloride (AgCl), for example, emits an electron in response to a cation. Conversely, it emits a chloride ion when stimulated by an electron.

$$Ag \rightleftharpoons Ag^{+} + e^{-} \tag{4.1}$$

$$Ag^{+} + Cl^{-} \rightleftharpoons AgCl \downarrow \tag{4.2}$$

Experimentation with various lead constructions revealed that commercial leads yield the best performance. Inexpensive, alligator clips yield the second best performance but are prone to noise. Custom leads are so prone to noise that a usable signal cannot obtained. Shielded leads perform better than unshielded leads. Unfortunately, commercial shielded leads were too expensive for this design iteration.

4.8.7 Skin Preparation

Skin surface impedance has a tremendous effect on signal quality [51]. Mismatched impedance causes differential electrical transduction between the leads, which translates directly to the voltage measured by the differential amplifier [49]. There are several methods for overcoming this, the simplest of which is to prepare the skin with a commercial electrode preparation gel. NuPrep $^{\text{TM}}$is a slightly abrasive sterilizing gel that is applied to a cotton swab and rubbed over the desired electrode area. This action gently removes the top layer of dead skin and cleans the electrode site. After preparation, the electrodes have an equal, low impedance connection site on the skin.

4.8.8 Software Filtering

Noise can also be subtracted in software. Simple discrete-time Fourier-transform-based filtering can be employed to control selected bands. More complex filtering methods are suggested by Clancy, Hamilton, Erfanian, LevKov and Osdamar [52–56].

These methods of software filtering apply advanced spectral analysis to selectively remove artifact signals such as electroocculogram (EOG) and motion artifact. With the exception of Levkov's subtraction procedure, none of these systems were implemented because of limited computational resources.

The methods described by Levkov are advanced subtraction procedures that are highly effective at increasing the CMRR [56]. A simplified version of Levkov's procedure was investigated. Three electrodes were used, signal (S), reference (R) and ground(G). The desired recorded lead was between signal and reference. Actual recorded leads were signal-to-ground (SG) and reference-to-ground (RG). All leads contained equal common mode noise (v_{cm}). This noise was almost completely rejected by subtracting RG from SG. Multiple leads are required to perform a subtraction procedure, therefore Levkov's methods were not implemented in the final design. Single-lead noise-rejection methods were implented.

$$SG = V_{signal\ to\ ground} + Noise_{common-mode} \tag{4.3}$$

$$RG = V_{reference\ to\ ground} + Noise_{common-mode} \tag{4.4}$$

$$SR = (V_{signal\ to\ ground} + Noise_{common-mode}) - (V_{reference\ to\ ground} + Noise_{common-mode}) \tag{4.5}$$

$$= V_{signal\ to\ ground} - V_{reference\ to\ ground} \tag{4.6}$$

$$= V_{signal\ to\ reference} \tag{4.7}$$

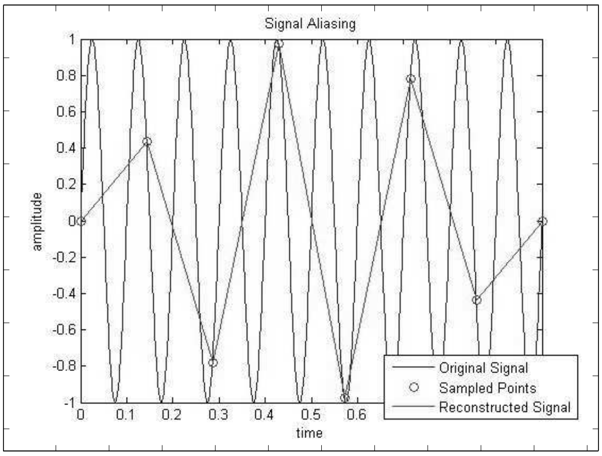


Figure 17: Example of Signal Aliasing.

5 Data Acquisition

5.1 Introduction

Analog data must be acquired by a computer in order to do the processing needed for a BCI. The goal of this BCI was to predict arm-angle based on EEG activity from a single head lead. Both arm-angle and EEG activity are acquired by a National Instruments PCI-6025E DAQ. EEG activity and arm angle are converted to analog electrical signals that are sampled by the DAQ at 300Hz.

The sampling frequency, f_s, of 300Hz is based on the Nyquist criteria of data sampling. This criteria states that the lowest frequency at which a signal may be sampled is the Nyquist frequency. The Nyquist frequency is equal to half the sampling frequency. If the greatest frequency component of a sampled signal exceeds this frequency, aliasing will occur.

In the case of this BCI, the highest frequency component to be read lies in the EEG signal and is approximately 40Hz. When the original EEG biosignal is hardware filtered, attenuation begins at 40Hz. Because no realizable filter is perfect, only frequencies above 100Hz can be considered to be of negligible amplitude. Based on this assumption, the Nyquist frequency is 200Hz. The Nyquist frequency however dictates the minimum frequency at which a signal can be sampled, not the optimal sampling frequency. Faster sampling results in more accurate signal reconstruction. It also results in larger, more unwieldy data sets. Optimal sampling balances sampling accuracy with data storage and processing capability. Most BCI applications report sampling frequencies of around 120Hz [27]. This is inadequate for this application because it is below the Nyquist frequency. Sampling at an arbitrarily chosen rate of 1.5 times the Nyquist frequency, 300Hz, results in better accuracy than sampling at the minimum Nyquist frequency. Sampling at 300Hz also results in data sets of manageable size.

Calculating the data acquisition rate reveals that data management is quite feasible. Sampling at 300Hz results in 18000 samples per minute, per channel. At 12-bit sampling resolution, this equates to 216000 bits (27000 bytes or about 26.4kBytes) per minute, per channel. As two channels (EEG and arm angle) are being sampled, a total of 52.7kB are being sampled per minute. At this rate it would take 19.4 minutes to accrue 1MB of data. This data size is manageable considering that only ten minutes of data samples are required to train a BCI [16]. Most personal computers now have up to 2048MB (2GB) or more RAM. Were RAM to be used strictly for data storage, a personal computer with 2GB of ram could store 27.6 days worth of data. This indicates that the data requirements of this BCI are well within the processing power of even a modest home PC.

5.2 Electroencephalogram Data Acquisition

The EEG signal begins as a bioelectric signal produced by groups of neurons in the brain. The signal causes ions to move locally within tissue. These ions instigate a chemical reaction in the silver chloride (AgCl) EEG electrodes. This chemical reaction results in the emission or absorption of electrons. Electrons flow preferentially along copper lead wires to a differential op-amp where they are transduced into a powered electrical signal. This signal is amplified, filtered in hardware, amplified again, and transmitted as a large analog voltage to a data acquisition device. Low noise data acquisition is made possible by separating the digital sampling circuitry of the DAQ from the sensitive analog circuitry of the EEG sensor. To further make the EEG signal as noise free as possible, the EEG channel on the DAQ is connected physically as far apart from the arm DAQ channel as possible to prevent electrical crosstalk.

The analog signal of the EEG is sampled at a rate of 300Hz with 12-bit resolution. With $\pm 6 Volt$

rails this equates to a voltage sensitivity of about $3mV$. This sensitivity is sufficient for reading the small EEG waves which are amplified to a peak to peak amplitude of around $20-30mV$. Larger EEG waves have peak-to-peak amplitudes of $200-300mV$. Voltage transients have amplitudes in the volt range. These EEG signals are sampled, buffered on the DAQ, and read into Mathworks Matlab7.1 $^{\text{TM}}$for processing and permanent storage.

5.3 Arm-Angle Data Acquisition

The position of the human elbow is regulated by local lower motor neurons and upper motor neuron inhibition of the lower motor neurons. Elbow activity is cortically controlled by the activation of upper motor neurons inhibiting muscular opposition to the activity. In other words, flexion is controlled by the inhibition of extension-lower-motor neurons and extension is controlled by the inhibition of flexion-lower-motor neurons. [12]

Indirectly, elbow angle indicates the relative activity of the upper motor neurons controlling the elbow. For this BCI, a special elbow brace, originally designed by NCSU graduate student Carey Merritt, was modified to sense elbow angle [57]. This was achieved by the addition of a potentiometer sensor to the elbow brace that changes resistance as a function of brace angle.

Power and ground leads of the position sensor are connected to a 9V battery. As arm position changes, the voltage of the variable-output lead of the sensor changes concomitantly. The variable lead of the sensor is connected as analog input to a second channel of the DAQ. This channel is physically connected as far as possible from the EEG channel on the DAQ to prevent crosstalk. The arm-angle channel is sampled at 300Hz and 12-bit resolution. The arm channel is sampled synchronously with the EEG channel. Arm position data is sampled, buffered on the DAQ, and read into Mathworks Matlab7.1 $^{\text{TM}}$for processing and permanent storage.

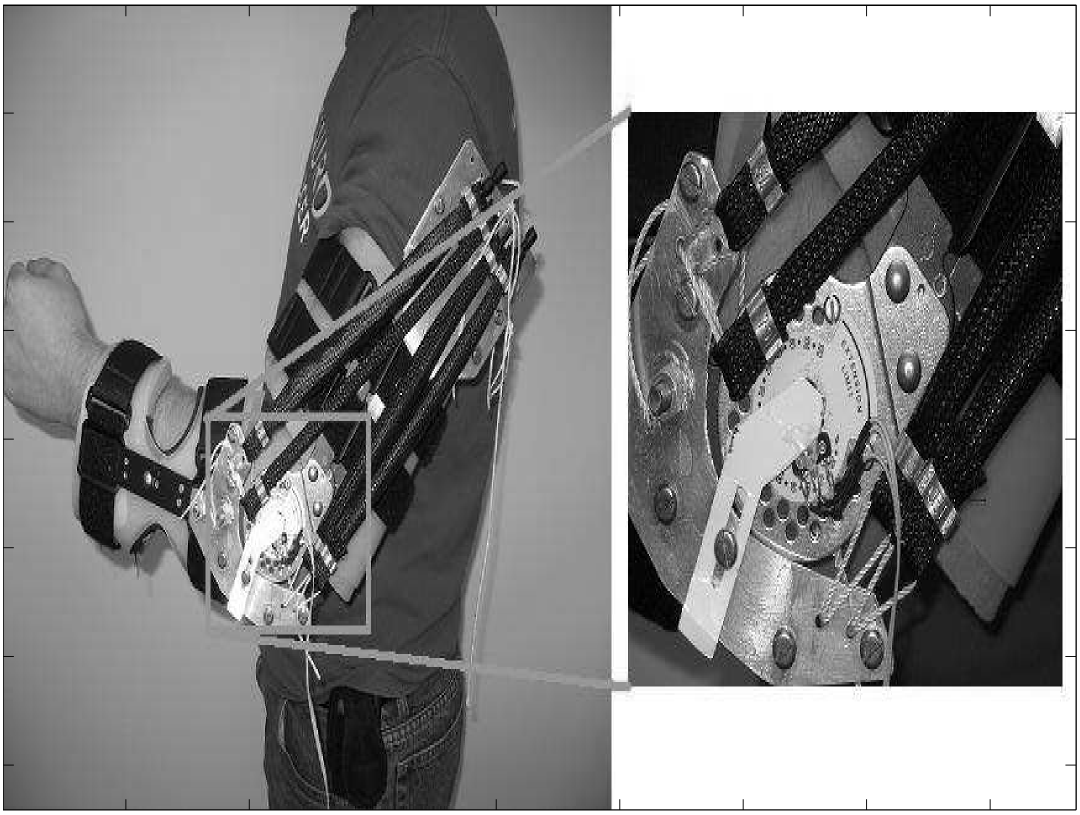


Figure 18: Elbow angle sensor.

6 Brain Computer Interface Software

6.1 Introduction

All the processing of the BCI is done in software in real-time. The BCI hardware is responsible for providing the software with clean, meaningful data. Software then performs all of the signal processing necessary to predict actual arm movement in real-time based on the EEG signal. Signal filtering, alpha- and beta-wave extraction, power-band analysis, and neural network arm-angle-prediction are performed in real-time. Neural network training is performed off-line.

The program has a meaningful graphical user interface that allows the experimenter to acquire, view, analyze, save, and recall data. When a subject is properly connected to the system, the experimenter begins data acquisition and can view any of the acquired or processed channels in real-time. Acquisition normally occurs for a fixed period of ten minutes, but may be interrupted at any time by the experimenter. Acquired data can be saved and used for neural network training at the experimenter's discretion and convenience.

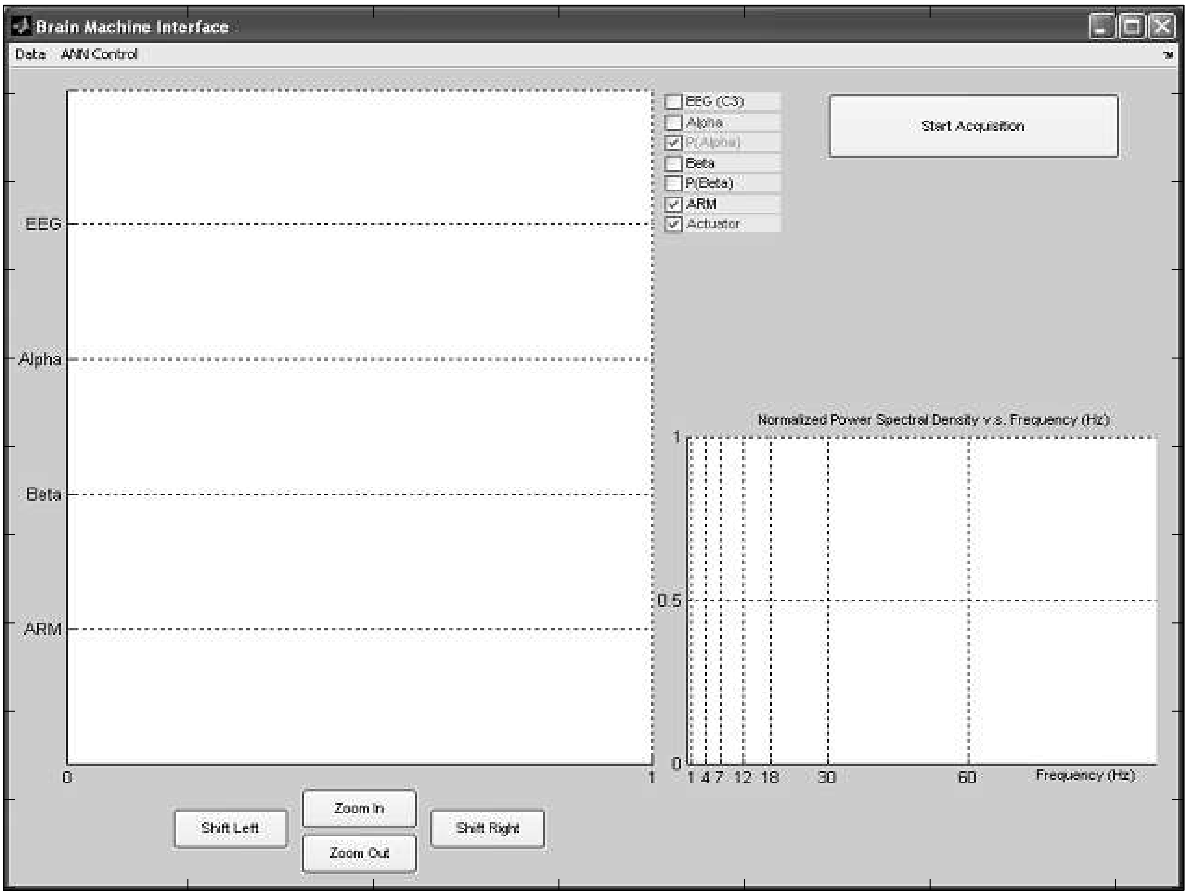
The program was developed in Mathworks Matlab7.1 ^{TM}. Of the available environments in which a signal analysis platform could be built, Matlab provided the most efficient environment for rapid prototyping and development. Matlab Signal Processing, Data Acquisition and Neural Network toolboxes were used.

The tools built into Matlab allowed rapid iterative development. Signal processing requires many complex algorithms such as the fast-Fourier-transform frequency-analysis method and Levenberg-Marquardt neural-network-training method. Matlab has many such methods already built-in and optimized. A significant part of the software design time involved experimenting with different algorithms for signal analysis. Use of Matlab enabled the rapid development of a functional system. See Figure 2.

6.2 Design Goals

The purpose of the software was to provide an arm-angle prediction based on analog EEG data. The software was supplied with the analog EEG signal via a data acquisition device. Actual arm angle was also supplied via the data acquisition device for evaluation of accuracy. The actual arm angle was used to train the learning algorithm. The actual arm angle was NOT used to help render the arm-angle prediction. Use of the actual-arm angle to aid in arm-angle prediction would have defeated the purpose of a brain-actuated system. The software did all of the signal processing necessary to predict arm angle in real-time. Natural physiological differences between subjects makes a fixed prediction algorithm impossible. A computer-learning algorithm was employed to make the arm-angle prediction. This algorithm was trained off-line but yield predictions in real-time. The prediction did not need to be output to any device. All that was required is that the output be displayed on a screen along with the actual arm angle.

To facilitate data analysis, the program allowed data to be viewed, saved, and recalled. Viewing the data, both in real-time and retrospectively, allowed experimenters to assess the quality of recordings



☐ EEG (C3)
☐ Alpha
☑ P(Alpha)
☐ Beta
☐ P(Beta)
☑ ARM
☑ Actuator

Figure 19: BCI Graphical User Interface.

under various conditions. This is critical at the beginning of a trial. Frequently, electrodes needed to be adjusted or repositioned in order to improve the signal-to-noise ratio. Saving and recalling data allowed for data analysis and simulation without connection of the BCI to a live subject.

6.3 Data Acquisition Procedure and Input Filtering

6.3.1 Data Buffering and Display Threads

Two channels of analog data are simultaneously read into the software at a rate of 300Hz. This rate corresponds to approximately three times the highest frequency component to be read. Data is read into a 12-bit National Instruments PCI-6025E DAQ. In software, there are two concurrent timer-based threads. A data acquisition thread gets data from the PCI-6025E buffer and immediately processes the signal. A second display thread puts the acquired and processed data on an on-screen oscilloscope. Both primary threads are timer-based. Matlab does not support threads, thus, multiple processes were put on timers. The acquisition and display timers have periods of 0.1 and 0.12 seconds respectively. This means that collisions will only occur approximately every 1.2 seconds, or every tenth cycle. The display timer is set to drop its action upon collision, whereas the acquisition timer is set to queue its action upon collision. With this configuration, should a collision occur, the display timer will yield to the acquisition timer. The acquisition timer is given priority because it is responsible for maintaining the real-time signal processing. A lapse in the display is preferable to a lapse in the real-time signal processing.

6.3.2 Overlapped Band-Pass Filtering

The EEG signal read by software contains a lot of information that is irrelevant to the prediction of arm movement. Fabiani *et al.* report that alpha waves (8-12Hz) and beta waves (18-28Hz) provide the best information about motor movement [24]. Relevant wave bands can be extracted using software band-pass filtering. The relevant wavebands for the extraction of motor activity are the pure overall EEG (3-35Hz), alpha waves (8-12Hz), and beta waves (18-28Hz).

The procedure for software band-pass filtering follows.

1. Compute the FFT of a the signal to be filtered.
2. Zero all elements of the FFT corresponding to frequencies outside of the pass-band.
3. Compute the inverse FFT of the modified FFT to obtain the filtered signal.

The FFT results in the complex frequency-domain representation of a discrete time-domain signal. A discrete signal of M points will have a FFT with M points as well. Each point corresponds to a frequency component. In the FFT there are two points for each frequency component. An M point FFT thus represents $\frac{M}{2}$ frequency components. The highest frequency component represented by a FFT is always $\frac{sampling frequency}{2}$. It follows that there are $\frac{M}{2}$ frequencies between $0Hz$ and half the sampling frequency. More FFT samples means greater FFT resolution. We compute how many samples are needed to obtain a FFT precision of ρHz for a sampling frequency f_s

$$f_{FFTmax} = \frac{f_s}{2} \tag{6.1}$$

$$\Delta f = \frac{f_{FFTmax}}{\frac{M}{2}} = \rho \tag{6.2}$$

$$M = \frac{f_s}{\rho} \tag{6.3}$$

$$\frac{f_s}{\rho}\, samples = \frac{1}{\rho}\, seconds \tag{6.4}$$

It was desired to software-filter incoming data with at least 1 Hz precision at a real-time rate of 10 Hz. With a sampling rate of 300 Hz and an acquisition timer frequency of 10 Hz, data is acquired by the software 30 samples at a time. Filtering this data on demand, 30 samples at a time, results in a FFT precision of 10 Hz. This means that the filterable bands are 0-10 Hz, 10-20 Hz $\cdots$ 140-150 Hz. To approximate filtering at 3-35 Hz, 8-12 Hz and 18-28 Hz, this precision was unacceptable. To accurately achieve these filtered bands, a FFT resolution of 1Hz or less was required. This implied filtering at least

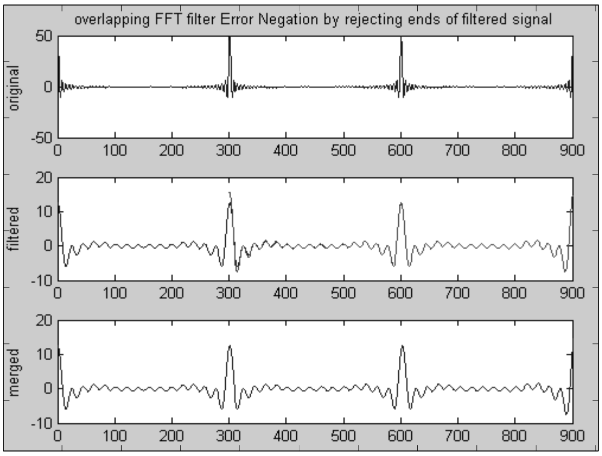


Figure 20: Software bandpass filtering results in errors at the beginning and end of the signal.

one second of data at a time, regardless of sampling rate. Reading buffered samples and processing data once every second is not real-time processing. Another means of processing was employed to keep the processing rate within the design parameters.

Fabiani *et al.* [24] describe in their paper "Conversion of EEG Activity into Cursor Movement by a Brain-Computer Interface(BCI)," a creative method of computing AR coefficients on incoming data. Every 0.1 seconds the Fabiani BCI computes the AR on the past 0.2 seconds of data such that the iterations overlap [24]. This method inspired a similar method for band-pass filtering. Using a similar method, data is acquired 0.1 seconds at a time. However, filtering is immediately done on the past two seconds of data in order to acheive $\frac{1}{2}Hz$ filtering precision. Although the filtering is more precise, the first quarter of the filtered signal is not accurate.

The beginning of the filtered signal overlaps with a signal that was previously filtered. The overlapping areas are nearly identical except at the first quarter of the overlap where the most recently filtered signal deviates. The overlapping portion of the newly filtered signal is safely discarded, resulting in 0.1 seconds of data that has been filtered with $\frac{1}{2}Hz$ precision.

The filtered signals, now cleaned of undesired components, are ready for further processing steps as Hilbert transformations or AR coefficient computation.

6.3.3 Arm Channel / Predicted Channel Acquisition

One of the challenges of data acquisition is that the data channels require two different sampling rates. The two environmental signals that are read into the computer via the PCI-6025E data acquisition card are EEG brain activity and actual arm angle. The EEG channel requires a sampling rate of 300Hz. The arm angle requires a sampling rate of 10Hz. Sampling the EEG at a slower rate results in the destruction of the signal from aliasing. Sampling the arm angle at a faster rate creates unnecessarily large data sets.

Fabiani *et al.* reported using an under-sampled kinematic variable to perform training [24]. Fabiani's method of sampling is to acquire the EEG at a high rate, thus getting accurate filtering and power-band data [24]. The kinematic variable is sampled at 10Hz [24]. Algorithm training is done on the 10Hz sampled signal but makes use of the finer data of the EEG to get more accurate frequency information [24].

Arm-position sampling and simulation, as well as neural network training, is performed on a the 10Hz signal. Arm position is read synchronously with EEG. Both signals are read at 300 samples/sec. Arm position is under-sampled to 10 samples/second by discretely averaging 0.1 seconds of acquired data at a time and storing the value. EEG data is stored at the original 300 samples/second. Arm-position prediction is done at 10Hz to match arm acquisition. Each prediction uses the past 15 seconds of high resolution EEG data to predict arm angle.

Under sampling the arm results in an efficient system. Consider the case where arm angle is predicted every 3ms (at 300Hz) as opposed to every 0.1 sec (10Hz). Simulation time would increase by a factor of 30. Training time would increase by at least a factor of 30. The arm-angle prediction would exhibit high frequency components every time the learning algorithm made a poor prediction. The sampling rate would be decreased further but one risks sampling too slowly. Fabiani *et al.* assert that 10Hz is an effective rate and that is confirmed in this research [24].

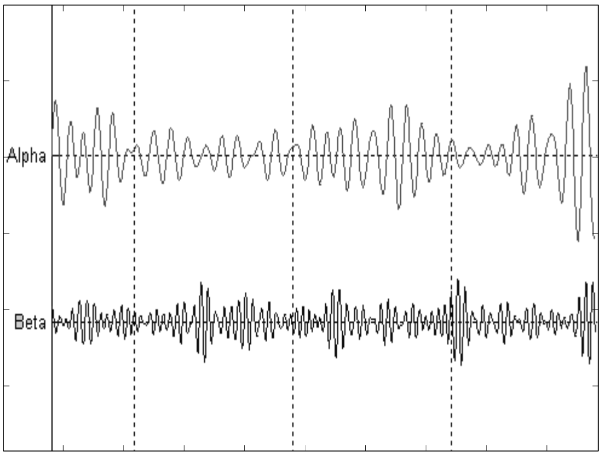


Figure 21: Alpha and beta waves can be extracted from the EEG signal by bandpass filtering.

6.4 Feature Extraction

6.4.1 Component Wave Extraction

As neurons of the central nervous system fire, coordinated firing results in patterns of electricity that can be measured at the scalp. Spectral analysis reveals certain frequency bands to be indicators of cortical activity. Various event-related potential (ERP) studies have shown that alpha and beta waves emitted from the area of the C3 and C4 electrodes (primary motor cortex) on the standard international 10-20 EEG configuration, exhibit high correlation to motor movement [16, 34]. It is thus desired to extract the alpha and beta waves from the net EEG signal.

Alpha and beta wave extraction is accomplished via software FFT-based band-pass filtering. The method of band-pass filtering is described in section 6.3.2. As net EEG data is sampled, it is independently filtered in real-time on the 8-12Hz and 18-28Hz bands. The results of that filtering are stored as two additional computed signal traces. These traces are the alpha and beta waves.

6.4.2 Power Density and Autoregressive Coefficients

Event related (de)synchronization of the alpha and beta bands of the primary motor cortex EEG leads (C3, C4) provide indicators of motor movement [58]. When viewing the electrical activity of the brain, event-related synchronizations appear as waveforms of increased amplitude. The amount of syn-

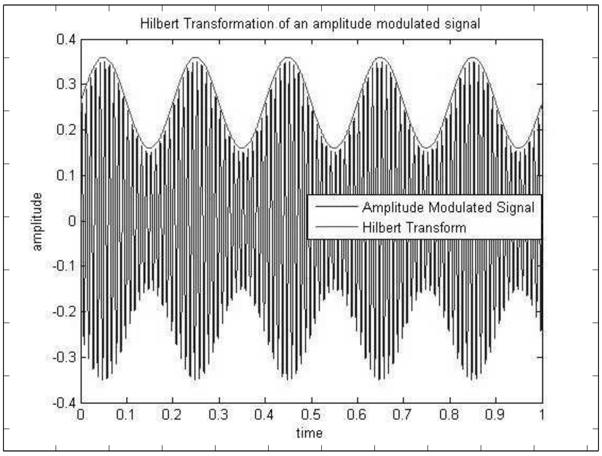


Figure 22: Hilbert transformation of an amplitude modulated signal

chronization is thus related to the signal energy as opposed to simply the signal's instantaneous amplitude. Two methods for extracting the degree of neural synchronization are considered, the Hilbert transformation and AR coefficients. The Hilbert transform provides the signal envelope for the transformed wave, thus giving a measure of its instantaneous power. Autoregressive (AR) coefficients describe the filter coefficients needed to construct a given signal from white noise. As such, the coefficients provide the spectral power density of a signal. Examining the properties of these methods on the relevant spectral ranges for alpha and beta waves yields the degree of neural synchronization in those bands.

The Hilbert transform method of computing neural synchronization is to compute the envelope of the alpha and beta bands. Hung *et al.* use this in their BCI [16]. To acquire a measure of the instantaneous synchronization in the alpha and beta bands, the relevant bands are first extracted from the EEG signal via software band-pass filtering [16]. The Hilbert transformation of the computed alpha and beta waves are acquired by convolving each signal by $\frac{1}{\pi t}$. In Matlab this is easily executed using the built-in $hilbert(x)$ function, part of the Matlab Signal Processing Toolbox. This instantaneous signal envelope provides an indication of the absolute degree of synchronization in the alpha or beta band.

An alternate means of computing neural synchronization is to compute the normalized spectral power density and analyze what percentage of the overall power lies in the desired power bands. Autoregressive (AR) coefficients reveal the filter coefficients required to fabricate a given signal from white

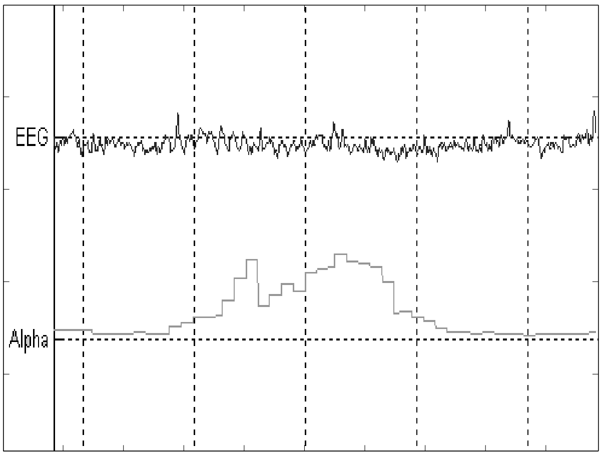


Figure 23: The relative power of the alpha waves is computed as a function of time using AR coefficients

noise. The 255-point frequency response of the AR transfer function reveals the spectral power density. The preceding two steps are done quickly in Matlab with the $pburg$ algorithm from the Matlab Signal Processing Toolbox. This function is used on the EEG signal which is first filtered between 3 and 35Hz to remove obviously undesired signals. The resulting transfer function yielded by the “pburg” algorithm is analyzed by summing the frequency response over the alpha and beta spectra. The sums reveal the relative power of the alpha and beta bands. This method was successfully used by Guger and Pfurtsheller [14].

Comparison of the two methods for computing neural synchronization reveal that AR coefficients provide a more reliable measure. Hilbert transformations are excellent for computing the envelope of amplitude modulated signals. Not all signals provide as nice results. Square waves for example, create divergent signals when using the Hilbert transform. Switching behavior can thus crash subsequent processing stages when the Hilbert transformation yields $\pm infinity$ at switching junctions. Computing AR coefficients provides the additional benefit that it can diagnose 60Hz noise which will be noticeable upon examination of the spectral power density.

Figure of spectral density.

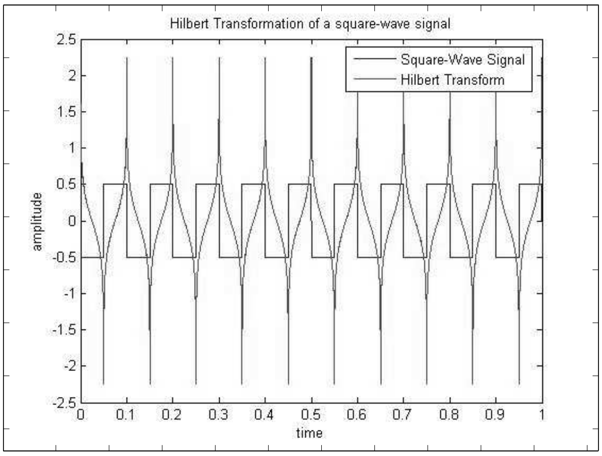


Figure 24: The Hilbert transform diverges under specific signal conditions which can cause errors during further signal processing steps.

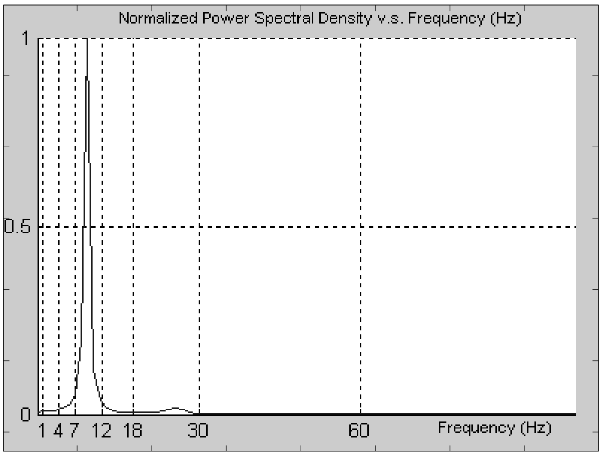


Figure 25: Spectral power density is a more stable measure of neural synchronization.

6.5 Learning Paradigms

This section provides an overview of the most popular learning algorithms in BCI. A brief description of each algorithm is provided along with the motivation for its acceptance or rejection.

6.5.1 Bayesian Networks

A Bayesian network is a state-based, probabilistic graphical structure. A series of states are defined, the connections between which are assigned a bayesian probability. These networks are extremely useful in BCI research for differentiating cognitive activities such as left- and right-handed motion [26]. These networks do not provide a continuum of possible values, so they are of limited use in predicting kinematic variables. One could approximate a kinematic variable by predicting the sign of the derivative of an action (-1 0 +1). This prediction could be integrated over time to acquire an estimate position. Given the lack of an accurate derivative magnitude, this method was of limited use and was not used in this design.

6.5.2 Linear Discriminant Analysis

Linear discriminant analysis (LDA) seeks to separate data into classes according to computed or learned linear discriminants. LDA is used for solving problems such as determining which frequency bands contribute to motor activity [16]. LDA is a common tool in machine learning but, like Bayesian networks, lacks the ability to express a prediction numerically as opposed to categorically. LDA would provide the ability to adjust alpha and beta frequency bands to the precise relevant band of the individual subject. This feature would be useful but was not within the scope of this thesis. LDA was not used in this project iteration.

6.5.3 Support Vector Machines

Support vector machines (SVM) are reported to be highly effective in BCI predictions [16]. SVMs are linear or non-linear data classifiers. Data is classified using hyperplane separators called parallel margin hyperplanes. These planes, parallel and equidistant from the separator plane, maximize data class separation [59]. No information was found to show how SVMs could be used for numerical as opposed to categorical predictions. However, SVMs hold much promise as their training is far simpler than most algorithms. SVM training relies on only the data points (support vectors) adjacent to the margin hyperplanes [59]. The availability of SVM toolboxes for Matlab makes their use even more appealing. Although Hung showed SVMs perform equal to or better than neural networks, no support could be found to implement them as a numerical predictor [16]. SVMs were not incorporated in this iteration of design.

6.5.4 Mutual Information

Mutual information (MI) is a useful tool in BCI. MI provides feedback about how well an input dataset will predict an output. MI addresses the question of mutual entropy, or how much predictive uncertainty is left given a set of knowledge. Given the probability distribution functions, p, of two discrete datasets X and Y, it can be discerned how much information about Y is contained in X by formula (number formula below)

$$Mutual\ Information = \sum_{x}\sum_{y} p(x,y) log_2 \frac{p(x,y)}{p(x)p(y)}$$

This information can be used as feedback on connection quality, electrode placement, and even the quality of the learning algorithm. Rezaei reports using mutual information in their work [26]. The additional amenities provided by MI did not outweigh their implementation cost and they were not used in this project iteration.

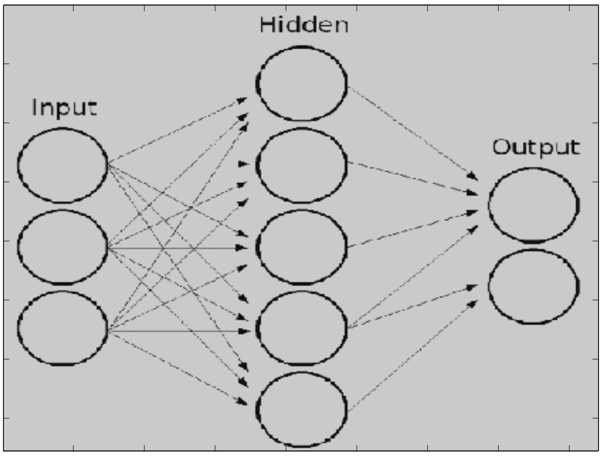


Figure 26: An example of an artificial neural network.(figure from [60])

6.5.5 Neural Networks

Neural networks provide a robust machine-learning tool with a multitude of possible configurations and support in many environments. BCI experts such as Miguel Nicolelis, use neural networks as their learning algorithms [13]. Hung achieved 60-70% prediction accuracy using neural networks in their BCI learning algorithm trials [16]. Neural networks can be configured for numerical or categorical classification as well as linear and non-linear response. Matlab has a standard Neural Network Toolbox with great support and a variety of built-in functions, including multiple learning algorithms. Furthermore, there is something truly concinnous about using artificial neurons to disambiguate the activity of biological neurons. Artificial neural networks were chosen as the machine-learning algorithm because of their ease of implementation and demonstrated sucess with effective BCI prediction in other studies [13, 16].

6.6 Neural Network Architecture

A 15-12-1 sized feed-forward, backpropagation neural network with respective tangent-sigmoid and linear neuron layers was employed as the predictive element of the BCI. Inputs were one input neuron with 14 consecutive tapped delays. The Levenberg-Marquardt learning algorithm was selected.

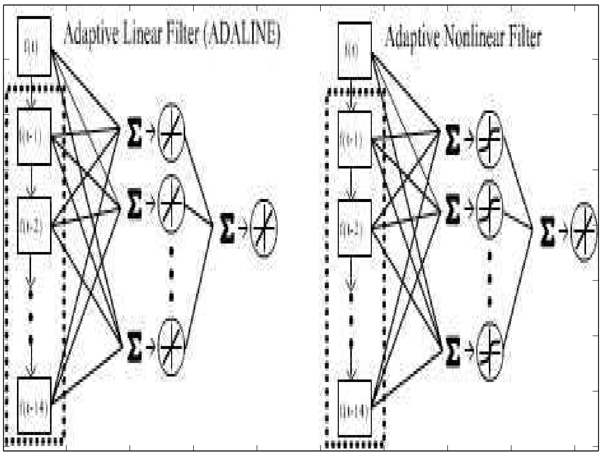


Figure 27: Adaptive filters networks are specific configurations of neural networks. These filters can be linear or non-linear. The non-linear version shown was used in this project.

6.6.1 Input Size

Design of the input was inspired by adaptive linear (ADALINE) neural network filter [27, 61] . An $N-input$ ADALINE filter typically has one input that traverses sampled data. Previous samples are passed backward through $N-1$ tapped delay inputs.

Hung *et al.* describe an ANN for the classification of hand movement that has 128 inputs, ten hidden neurons and one output [16]. The neural network used in this project has one input that traverses the processed EEG data ($EEG_p(t)$). There are 14 tapped delays that sample past data at an interval of one second. In other words, the inputs to the neural network at any given time, t, are the processed EEG samples $\{EEG_p(t)\ EEG_p(t-1)\ EEG_p(t-2) \cdots EEG_p(t-14)\}$.

The input vector of 15 elements, spaced one second apart was converged upon empirically. Starting with 10 input neurons spaced one second apart, network size was increased. Stopping conditions were negligible performance and / or substantially increased training time. An input size of 15 yielded a good balance of performance and training time.

6.6.2 Hidden Layer Size

Whereas ADALINE filters use linear neural network transfer functions in their hidden layers, BCI applications require non-linear learning. For this reason the hidden layer of 12 neurons used tangent-sigmoid transfer functions. These functions take a weighted sum of inputs from the input layer and compute an output in the range of $\Re \in [-1,1]$.

Figure: Tangent Sigmoid Function

The hidden layer size of 12 was converged upon empirically by starting with a small number of hidden neurons and increasing the number until the added training time balance the performance increase. A hidden layer size of 12 met this criteria.

The outputs of these hidden neurons are weighted and summed and passed to the output linear transfer function. The output of this transfer function is $\Re \in [-\infty,\infty]$.

6.6.3 Training Algorithm

Several training algorithms were used. Initial trails with delta-rule gradient decent failed due to lack of accuracy. The Levenberg-Marquardt (LM) algorithm proved to be more accurate. Trials were done on small data sets. Hung reports using training times of ten minutes [16]. Ten minutes of data at 300 samples per second yields 180000 samples. Before the network was optimized, the LM algorithm consumed too much memory and regularly crashed the computer. Because the requirements of the BCI state that the system be manageable on a PC, the algorithm was changed back to the less accurate but less memory costly delta-rule. This time an advanced delta-rule algorithm was used that utilized “momentum” to prevent error minimization from converging at a local minimum. Network size and training methods were semi-optimized and the algorithm was returned to the LM algorithm for final optimization. The data showing the neural network trails is shown below.

The final training method is similar to the method of simulation. Instead of training on 180000 sample iterations, arm angle and the degree of neural synchronization is under-sampled by a factor of 30, to 6000 samples (or 10Hz). High-resolution EEG data is still used to compute the degree of synchronization. Training inputs are identical to the simulation inputs at a given time, t.

$$\{EEG_p(t)\ EEG_p(t-1)\ EEG_p(t-2)\cdots EEG_p(t-14)\}$$

Table 3: The results of the neural network performance trials resulted in convergence upon the current network architecture. [A] is simulation time duration, [B] is the ANN configuration. [C] is the input spacing in samples. [D] specifies the neuron transfer function (tansig is non-linear, lin is linear). [E] is the evauation function used for calculating predictive error. [F] is the training algorithm (traingdx is gradient decent, trainlm is Levenberg-Marquardt). [G] is the final error (mse or sse). [H] specifies how the simulation was terminated (Max Epoch is a natural termination).

A	B	C	D	E	F	G	H
412.7	5 10 1	1	tansig,lin	sse	traingdx	3108	Max Epoch
266.7	5 10 1	1	tansig,lin	sse	traingdx	1238.69	Max Epoch
74	15 12 1	150	tansig,lin	mse	trainlm	0.014	Max Epoch Poor Results
2.5	10 12 1	1	lin,lin	sse	trainlm	286	Fail
50	10 12 1	50	tansig,lin	sse	trainlm	198	Max Epoch
53	10 12 1	50	tansig,tansig	sse	trainlm	154	Max Epoch
21	10 5 1	50	tansig,lin	sse	trainlm	131	Max Epoch
41.4	10 10 1	100	tansig,lin	sse	trainlm	108.63	Max Epoch
61	12 12 1	100	tansig,lin	sse	trainlm	98	Max Epoch
50	10 12 1	150	tansig,lin	sse	trainlm	97	Max Epoch
67	10 15 1	150	tansig,lin	sse	trainlm	95	Max Epoch
51	10 12 1	150	tansig,lin	sse	trainlm	91	Max Epoch
74	15 12 1	150	tansig,lin	sse	trainlm	79	Max Epoch
73	15 12 1	150	tansig,lin	sse	trainlm	79	Max Epoch

The network weights of the 12-15-1 neural network are adjusted by the LM algorithm to minimized the sum-squared error (SSE) between the predicted arm angle and the actual arm angle. Miguel Nicolelis and others use mean-squared error (MSE), but MSE is more forgiving as it averages out accumulated error [19]. MSE was shown in trials to result in a flatline prediction through the mean of the arm data. This prediction minimized the mean error but was not accurate. SSE is unforgiving of any error as errors accumulate with time.

6.7 Output Smoothing

The output of the ANN is a rough prediction of arm movement that generally follows the arm angle. A better system might employ a large, sophisticated ANN or other learning algorithm that could closely fit the predicted arm movement to the actual arm movement. Such a system would be extremely costly to develop.

Another method to obtain an accurate solution is to artificially re-impose the constraints of arm movement after the fact. Arm movement exhibits certain simple characteristics that are hard for a learning

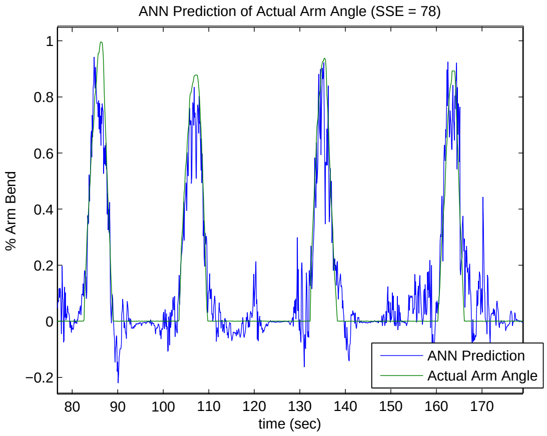


Figure 28: A neural network prediction of arm movement based on neural synchronization.

algorithm to pick up. Arm angle has hard limits on flexion and extension. Arm velocity is limited and typically very smooth. These limits can be imposed by simple filtering after the ANN makes its prediction. A simple ANN is thus allowed a greater degree of freedom. Greater freedom means shorter training times and greater network simplicity.

Rough arm-angle predictions of the ANN are made more accurate with simple filtering. The first phase is a simple threshold filter. Because the arm cannot bend backwards it is safe to say that any predicted angle less than zero can be interpreted as zero degrees. Also, all values below .1 may also be zeroed. This eliminates noise about zero, yet only minimally distort actual arm movements. This simplification of the signal allows an averaging filter to perform wonders on high frequency transient signals around zero. These transients get averaged into small ripples. Larger arm movements also get purged of their high frequency components in this process. A final round of threshold filtering removes the residual distortions left behind by the averaging filter.

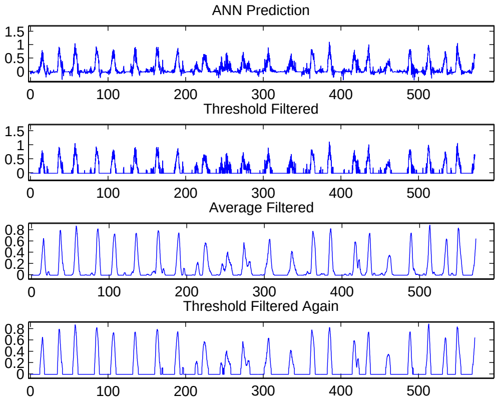


Figure 29: The progression of output smoothing as a signal is thresholded, averaged and re-thresholded.

7 Experiments and Results

7.1 Introduction

This chapter describes the human trials used to confirm the BCI design. In addition to the primary human experiments used to confirm the overall BCI design, many minor unstructured, non-human experiments were performed to optimize specific components. The results of the minor experiments are included in previous sections.

The first subject was run according to the procedure below. However, after completing testing of the first subject, it became apparent that obtaining IRB approval of the procedures might be prudent. Therefore all testing was ceased until IRB approval of the procedures was acquired. IRB documentation can be found in Appendix C.

7.2 Experimental Setup and Procedure

Three human subjects were recruited on the basis of availability. Vulnerable populations were not included. Experimental subjects were first informed of the dangers of the procedure and told not participate if they had any of a specific list of medical conditions. These conditions were provided to the subjects verbally so that no medically identifying information would be present on their consent form. Subjects and experimenter then signed the experimental consent form, with exception of the first subject,

Table 4: These populations are identified by the NCSU IRB as being “Vulnerable” and must be avoided in experimentation unless special permission is given to do otherwise.

minors
fetuses
pregnant women
persons with mental, psychiatric or emotional disabilities
persons with physical disabilities
economically or educationally disadvantaged
prisoners
elderly
students from a class taught by principal investigator

Table 5: People with the listed medical conditions would be at elevated risk if used in this study and were thus prohibited from participation.

Condition	**Reason for Exclusion from Study**
pregnancy	risks apply to fetus as well as subject
history of seizures	subject has a delicate cerebral state
allergy to ointments	non-hypoallergenic ointments are used
history of cranial surgery	subject has a delicate cerebral state
history of stroke	subject has a delicate cerebral state
requires oxygen	oxygen should not be used around unshielded electronics
has a pacemaker or other implant	increased risk of device failure
history of arthritis	required movements will be painful
history of joint injuries or fractures	joint is placed in potentially compromising position
problems sitting or standing	sitting and standing is required
strabismus	uncontrolled eye movements distort EEG
facial spasms	spasms distort EEG signal

who gave verbal consent. Subjects were briefed on the experimental procedure. A bean-bag chair was provided for the subject to lie in. This chair was deliberately chosen for its ergonomics. Settled into the bean-bag chair, subjects were less likely to move or fidget, thus minimizing extraneous neuronal activity in the subject’s primary motor cortex.

As the subject settled into the chair, they were prepared for experimentation. Electrode sites were prepared by cleaning the skin with NuPrep ^{TM}electrode site prep gel. This gel contains a slightly abrasive particulate matter to remove dirt and dead skin, thus reducing electrode impedance. After skin preparation, the signal, reference, and DRL electrodes were prepared. The signal and DRL electrodes were disposable, silver-silver-chloride electrodes filled with Ten20 ^{TM}self-adhering, conductive gel. The reference electrode was a silver-silver-chloride reusable ear-clip electrode filled with Parker Spectra360

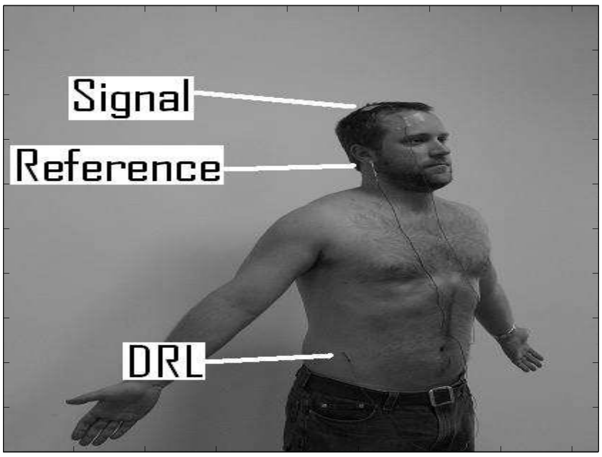


Figure 30: EEG electrode placement for the BCI interface.

TMgel.

The readied electrodes were then placed on the skin. The signal electrode was placed directly coincident with the C4 electrode site of the International 10-20 EEG system. This area lies just over the hand / elbow area of M1. The DRL electrode was placed on the right-lateral-inferior abdomen. This area is far from most bioelectric signal sources such as the heart. These two electrodes are taped in place to prevent accidental removal or disturbance. The reference electrode was clipped to the right earlobe.

Once all electrodes are were place, electrode placement was confirmed and adjusted. Brief analysis of the EEG signal on an oscilloscope confirmed signal quality. A quality signal was identified by the absence of a 60Hz spike in the FFT of the EEG. This implied that the DRL electrode made good contact. Signal and reference connections were confirmed by asking the subject to blink. A pronounced blink in the EEG indicated that signal and reference were properly connected. Poorly placed electrodes were adjusted until a proper connection was established.

After electrodes were placed, several diagnostics were quickly performed. First the DAQ leads were visually confirmed. The BCI software was started and a brief recording was taken. During this recording, all the traces were inspected for anomalies, especially the EEG, arm-angle and alpha power-band. Display gain factors were quickly rectified in software with a software restart. The arm-angle trace and alpha-power-band trace were visually inspected for crosstalk before the brace was placed on the subject. This was done by moving the arm brace while the subject was at rest. No deflection in the alpha-power-band trace during this activity ruled out crosstalk. Finally, a folder was created on the PC to store the trial data.

When the system was configured, data recording began. The subject was told to settle into place, keep eyes closed, minimize facial movement, and to flex the left arm at random. Subjects were instructed that flexion should include bending the elbow, bending the wrist, and closing the hand. This behavior is known to maximize contralateral M1 activity [9, 12].

The initial recording period began when the motion artifacts in the subject's EEG were no longer present. Trials lasted ten minutes. During this time EEG and random arm movements were recorded. After the initial recording period, the recorded data is saved. The learning algorithm was trained using the saved data. The trained neural network was then saved. Upon resuming recording, a real-time prediction of arm angle is provided.

7.3 Subject Trials

7.3.1 Subject 1

Narrative Subject 1 was a 54-year-old, left-handed male. Subject 1 described himself as "left-hand dependent". Initial electrode placement on Subject 1 was suboptimal. Electrode adjustment resulted in an excellent signal quality, demonstrating a clean eye blink artifact upon request. Subject 1 demonstrated visually obvious event-related-desynchronization associated with arm-motor movements. On the data from Subject 1, the learning algorithm was able to achieve a sum-squared error of 86 between the actual and predicted arm-angle. Subject 1's data was analyzed with multiple network configurations. Effectiveness of the training algorithm was confirmed using cross-validation. Subject 1's trial was sufficient to demonstrate effectiveness of the system.

Artificial Neural Network Summary After Subject 1's trial, Subject 1's data was run 5 times off-line to analyze ANN accuracy. Data analysis revealed a mean SSE of 80.4 and a standard deviation of 8.9. The

Table 6: A Neural Network training iteration performed on Subject 1's data.

```
Training
TRAINLM, Epoch 0/100, SSE 7461.26/0.002, Gradient 18566.8/1e-010
TRAINLM, Epoch 10/100, SSE 151.491/0.002, Gradient 31.258/1e-010
TRAINLM, Epoch 20/100, SSE 134.073/0.002, Gradient 9.72035/1e-010
TRAINLM, Epoch 30/100, SSE 115.167/0.002, Gradient 4.77234/1e-010
TRAINLM, Epoch 40/100, SSE 109.537/0.002, Gradient 21.9739/1e-010
TRAINLM, Epoch 50/100, SSE 105.921/0.002, Gradient 20.6039/1e-010
TRAINLM, Epoch 60/100, SSE 98.4473/0.002, Gradient 13.4643/1e-010
TRAINLM, Epoch 70/100, SSE 94.0842/0.002, Gradient 230.201/1e-010
TRAINLM, Epoch 80/100, SSE 90.0255/0.002, Gradient 13.6502/1e-010
TRAINLM, Epoch 90/100, SSE 89.2217/0.002, Gradient 1.41461/1e-010
TRAINLM, Epoch 100/100, SSE 89.0879/0.002, Gradient 7.21374/1e-010
TRAINLM, Maximum epoch reached
```

program output of one of the ANN trials is provided below. The graphical output of network trial reveals that a SSE in the range of 80.4 ± 8.9 is good.

Plotted Results Graphical output of the ANN trial above reveals that a SSE of 89 signifies excellent ANN learning. There was a predicted arm movement for every actual arm movement. Amplitude and duration of the predicted movement deviated slightly from the actual movement but overall curve fit was remarkable. Interestingly there were false positives. The current hypothesis is that these may represent imagined but not executed arm movements.

7.3.2 Subject 2

Narrative Subject 2 was a 28-year-old, right-handed male. Initial electrode placement in Subject 2 was suboptimal. Electrode adjustments resulted in only marginal improvement. Half way through the first trial, the sensor leads broke and the sensor had to be repaired. A successful trial using Subject 2 resulted in a sum-squared error of 332. On-line cross-validation yielded poor results due to a new DC-offset in the sensor that had to be compensated for. The results yielded by off-line analysis still demonstrated sufficient predictive capacity to consider the system functional.

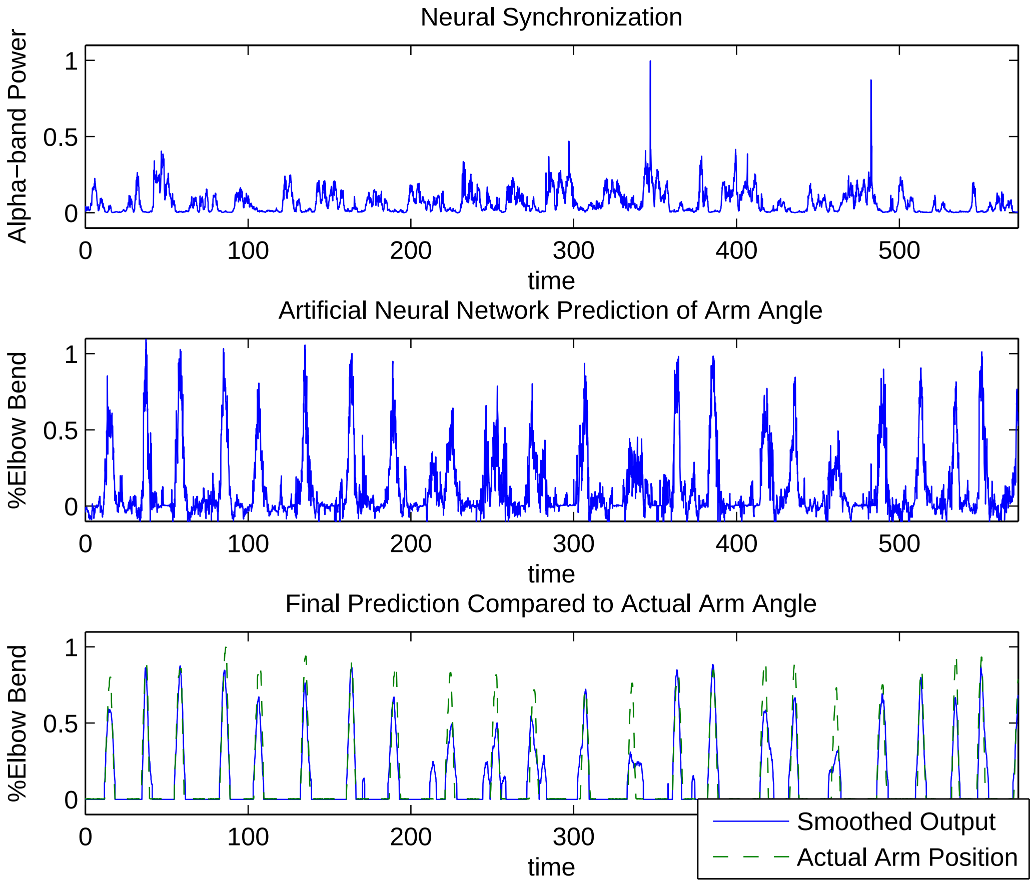


Figure 31: Subject 1: Graphical Results.

Table 7: A Neural Network training iteration performed on Subject 2's data.

```
Training
TRAINLM, Epoch 0/100, SSE 12992.1/0.002, Gradient 24286.1/1e-010
TRAINLM, Epoch 10/100, SSE 421.496/0.002, Gradient 685.112/1e-010
TRAINLM, Epoch 20/100, SSE 351.141/0.002, Gradient 81.2969/1e-010
TRAINLM, Epoch 30/100, SSE 318.885/0.002, Gradient 102.419/1e-010
TRAINLM, Epoch 40/100, SSE 310.737/0.002, Gradient 8.41557/1e-010
TRAINLM, Epoch 50/100, SSE 308.739/0.002, Gradient 2.0468/1e-010
TRAINLM, Epoch 60/100, SSE 307.806/0.002, Gradient 16.3905/1e-010
TRAINLM, Epoch 70/100, SSE 306.306/0.002, Gradient 7.94407/1e-010
TRAINLM, Epoch 80/100, SSE 305.607/0.002, Gradient 5.6504/1e-010
TRAINLM, Epoch 90/100, SSE 304.667/0.002, Gradient 11.5181/1e-010
TRAINLM, Epoch 100/100, SSE 304.165/0.002, Gradient 3.90714/1e-010
TRAINLM, Maximum epoch reached
```

Artificial Neural Network Summary After Subject 2's trial, Subject 2's data was run 5 times off-line to analyze ANN accuracy. Data analysis revealed a mean SSE of 305.6 and a standard deviation of 11.9. The program output of one of the ANN trials is provided below. Examination of the graphical output reveals that a SSE of 300+ is of substantially reduced quality when compared a SSE less than 100.

Plotted Results The plotted results of ANN training for Subject 2 reveal that predictive quality was substantially reduced. This is not surprising given the reduced EEG connection quality. Still, most actual arm movements were accompanied by a prediction. Periods of inactivity seem to be effected most by EEG connectivity. Many false positives existed in Subject 2's data.

7.3.3 Subject 3

Narrative Subject 3 was a 28-year-old, left-handed male. Subject 3's electrode placement was sub-optimal and was not improved by electrode adjustment. Sum-squared error for Subject 3 was 412. Two factors inhibited cross-validation. Electrode quality yielded a poor signal to begin with. An improperly compensated DC-offset from Subject 3 also factored into the neural network's inability to render a proper prediction.

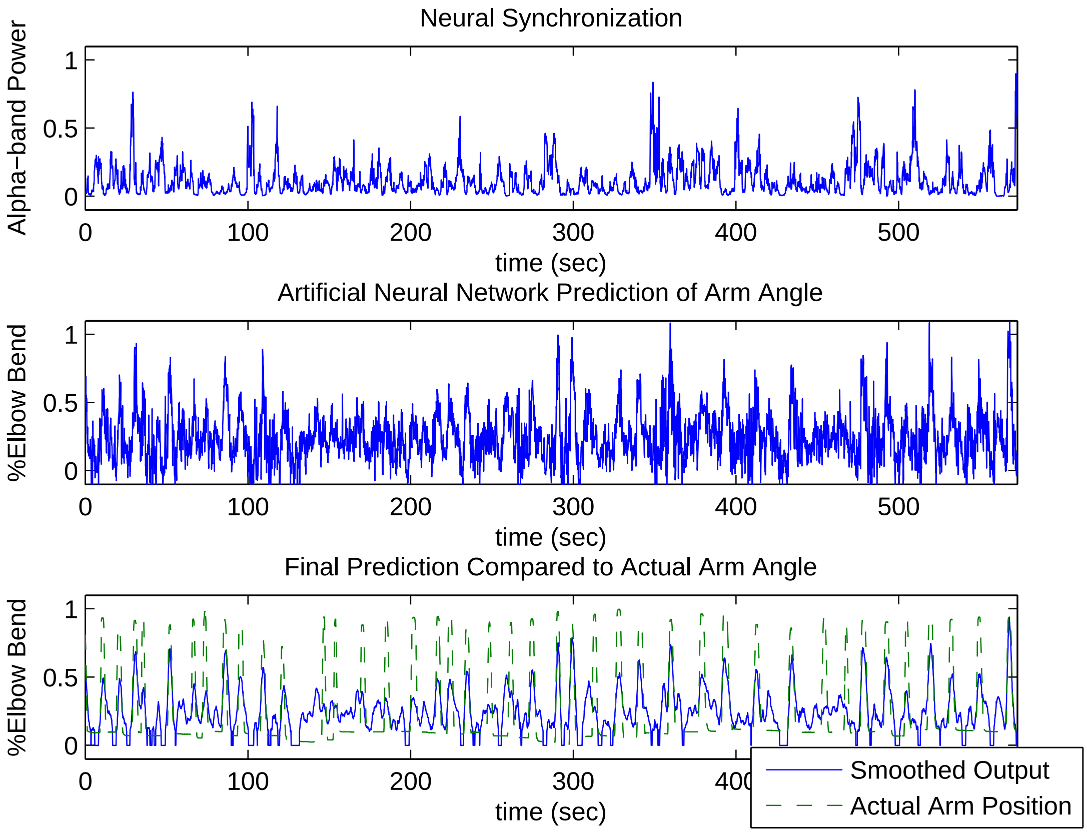


Figure 32: Subject 2: Graphical Results.

Table 8: A Neural Network training iteration performed on Subject 3's data.

```
Training
TRAINLM, Epoch 0/100, SSE 27528.3/0.002, Gradient 36084.9/1e-010
TRAINLM, Epoch 10/100, SSE 489.168/0.002, Gradient 171.401/1e-010
TRAINLM, Epoch 20/100, SSE 464.678/0.002, Gradient 141.937/1e-010
TRAINLM, Epoch 30/100, SSE 441.108/0.002, Gradient 174.926/1e-010
TRAINLM, Epoch 40/100, SSE 423.76/0.002, Gradient 72.7482/1e-010
TRAINLM, Epoch 50/100, SSE 418.828/0.002, Gradient 27.9806/1e-010
TRAINLM, Epoch 60/100, SSE 415.146/0.002, Gradient 14.5451/1e-010
TRAINLM, Epoch 70/100, SSE 413.872/0.002, Gradient 2.75456/1e-010
TRAINLM, Epoch 80/100, SSE 413.031/0.002, Gradient 2.30224/1e-010
TRAINLM, Epoch 90/100, SSE 412.515/0.002, Gradient 1.46729/1e-010
TRAINLM, Epoch 100/100, SSE 412.262/0.002, Gradient 2.22809/1e-010
TRAINLM, Maximum epoch reached
```

Artificial Neural Network Summary After Subject 3's trial, Subject 3's data was run 5 times off-line to analyze ANN accuracy. Data analysis revealed a mean SSE of 429 and a standard deviation of 21.9. The program output of one of the ANN trials is provided below. The results are expected to be worse than those of Subject 1 and Subject 2 based on connection quality. As expected, graphical output revealed a significantly diminished predictive capacity for Subject 3.

Plotted Results Subject 3's graphical results revealed a poor quality predictive capacity. Actual arm movements were accompanied by weak predictions at best. This finding is in accordance with the expected results based on the EEG connection.

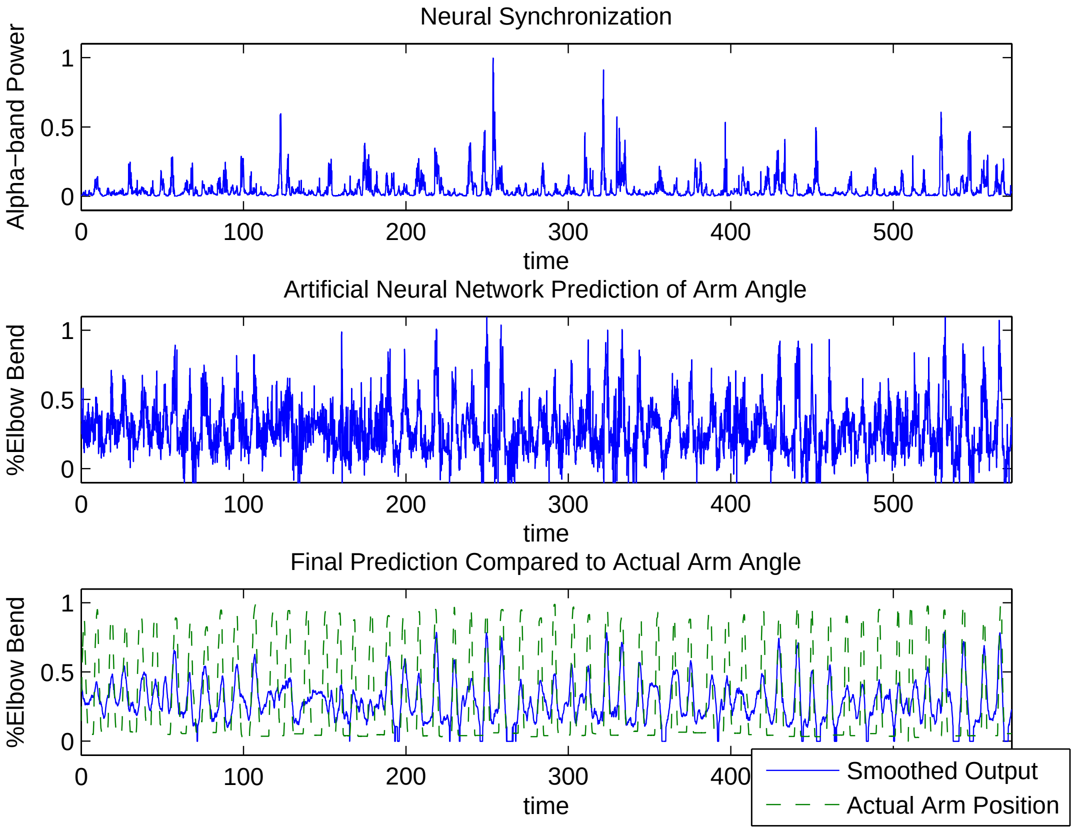


Figure 33: Subject 3: Graphical Results.

8 Conclusions

8.1 Introduction

The original design criteria of the project state that the project must be safe, EEG-based, inexpensive, PC-based, and able to predict a one-degree-of-freedom kinematic variable from EEG activity alone. From the specific design procedures and experiments described above, the system is analyzed against the original design criteria.

8.1.1 System Safety

The BCI System has been officially deemed safe by the North Carolina State University Internal Review Board. The IRB stated in their feedback on the original application,

> The risks posed are minimal and less than those associated with everyday electronic device use. The researchers are taking adequate steps to minimize risk and are prepared to address any adverse reactions in subjects. I would also say that I think the potential risks are over stated in the informed consent.

This confirms that the multitude of steps taken to protect research subjects were fruitful. The implications for this are that the BCI system can be used as a safe research and demonstration tool by untrained laboratory personnel.

8.1.2 Electroencephalogram Predictive Base

The final BCI product was scalp EEG based. A custom EEG amplifier was constructed and demonstrated to be effective for the prediction of motor activity.

8.1.3 System Cost

The complete system cost for the BCI system was around 200 US dollars. The entire project was completed without funding. The minimal cost of the system bodes well for its widespread use as it

Table 9: Expenses

Electronic Components	$104.97
Circuit Manufacturing	$98.30
EEG Supplies	$114.63
Total	$317.90

Table 10: Computer Configuration used for this project

Processor	2.80GHz Pentium(R)
RAM	1.00 GB
Hard Disk	30GB Western Digital
DAQ	National Instruments PCI-6025E
Software	Mathworks Matlab v7.1 +Data Acquisition Toolbox +Signal Processing Toolbox +Neural Network Toolbox

may be employed by students or researchers of even the most modest means. There are however two items that may not be available to all researches that were integral to this project. Mathworks Matlab ^{TM}mathematical analysis software, and a National Instruments PCI-6025E data acquisition card were the backbone of this project. These items were freely available through NCSU. Some research institutions may not have access to them. For this reason the future work done on this project might focus on migrating to an alternate software platform such as C++. It is suggested that this future platform employ the Dataq DI-158U USB DAQ. This DAQ is available for less that 100 US dollars. Other, less capable versions are available for as little as 25 US dollars.

8.1.4 Computer Platform

The system runs on a standard desktop PC. The only features that distinguished this PC from a PC that might be found in anyone's home is Mathworks Matlab ^{TM}mathematical analysis software, and a National Instruments PCI-6025E data acquisition card. As mentioned previously, future iterations should transfer to a C++ base with a Dataq USB DAQ device. This migration would not only render the project cost effective, but would guarantee that the system would be fully integrable into a standard home PC.

8.1.5 Predictive Capacity

Human subject trials demonstrated the predictive capacity of the system. The best sum-squared error demonstrated on IRB approved trails was 79, on ten minutes of training data. Other IRB approved subjects were not as successful due to a component failure and electrode lead connectivity. The component stability and electrode lead issues need to be addressed. Specific suggestions are mentioned in the future work section of this chapter.

The purpose of this experiment was not to analyze the precise effectiveness of the system using multiple subjects. The purpose was to simply demonstrate that it could work. This was done with a great deal of success. One suggested area of future work is to analyze statistically predictive effectiveness of the system using people who exhibit physiological differences. An interesting study might be to examine statistical predictive quality of left- versus right-handed people.

8.1.6 Final Remarks

It is the desire of the researcher that this BCI system be a useful tool for future research in the NCSU Center for Robotic and Intelligent Machines. Having been demonstrated as a functional tool, the BCI will allow research in intelligent human-computer interaction. Hopefully, this field will be explored further at NCSU.

8.2 Future Work

The first rule of engineering states that truly optimal design of any system requires infinite time and money. Certain design options for the BCI were triaged to meet time and budgetary constraints of the project. The design options marked for future work are listed below.

8.2.1 Immediate Improvements

Some immediate improvements need to be made to the EEG leads and DAQ. Improper lead placement repeatedly appeared as a major obstacle. This can be rectified by replacing the current signal and DRL leads with disposable adhesive “sticker” leads. These stickers form a reliable connection. However the connection can be of no use if the DAQ signal is corrupted. The NI PCI-6025E card takes excessive adjustment to negate noise. Switching to a USB-based unit where digital to analog conversion is performed locally as opposed to in the computer, would significantly improve data acquisition. The Dataq DI-158U is such a unit is currently unstable in Matlab.

8.2.2 Hardware Improvements

Future iterations of this project could focus on creating a more capable EEG. The first improvement to the EEG could be the addition of multiple channels. Channel readings from above Brodmann’s areas 5,6,7 could be added to the current lead over Brodmann’s area 4. This could improve predictive quality according to [24]. Multiple leads might also allow studies of other mental tasks, which are cortically multifocal.

Minor improvements to the EEG should include implementing an on-board DSP chip to perform as much signal processing as possible. This could both reduce overall circuitry needed as well as improve the signal-to-noise ratio (SNR).

Improvements to the EEG leads could also help improve the SNR. The use of active, shielded leads with a better connection to the EEG board would significantly reduce noise. As mentioned, leads that clip to adhesive pads rather than stick directly to the skin should be evaluated.

8.2.3 Software Improvements

Software improvements should include improvements to the graphical user interface (GUI). The user should have more control over the filtering, learning algorithm, etc. Currently these items can only be modified when the program is not executing.

Migrating to a language such as C++ could improve software performance. The trade-off of migrating away from Matlab is that it would be difficult to experiment with novel algorithms. Matlab is malleable at the expense of execution speed. A C++ program would be fast and could run on any PC but would not be malleable.

8.2.4 Software Output

The software currently provides limited ability to report data. Data can be saved, but two specific features are desirable. The software should provide the predicted arm angle as an analog output to be connected to a robotic arm for demonstration purposes. The software should also provide more information about connection and prediction quality.

8.2.5 Suggested Experiments

Several experiments would be productive in categorizing the behavior of the BCI. Human physiological differences affect factors like skin impedance and optimal electrode placement. A study should be done categorizing these differences and statistically demonstrating the BCI’s performance in response to each physiological factor.

References

[1] R. Jurmain, L. Kilgore, W. Trevathan, and H. Nelson. *Essentials of Physical Anthropology*. Wadsworth Pub Co, Belmont, CA, 2003.

[2] Wikipedia. Human enhancement. http://en.wikipedia.org/wiki/Human_Enhancement.

[3] Boron and Boulpaep. *Medical Physiology*. Elsevier Saunders, 170 S Independence Mall W 300E, Philadelphia, PA 19106 USA, updated edition edition, 2005.

[4] Shade, Rothenberg, Wertz, Jones, and Collins. *EMT-Intermediate Textbook*. Mosby, Inc, 11830 Westline Industrial Drive, St. Louis Missouri 64146, second edition, 2002.

[5] Spinal Cord Injury Information Network. Spinal cord injury facts and figures at a glance. http://www.spinalcord.uab.edu/show.asp?durki=21446.

[6] Wikipedia. Brain-computer interface. http://en.wikipedia.org/wiki/Brain-computer_interface.

[7] M.M. Moore. Real-world applications for brain-computer interface technology. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2), June 2003.

[8] Wikipedia. Hal 5. http://en.wikipedia.org/wiki/HAL_5.

[9] A.P. Georgopoulos. Magnetoencephalographic signals predict movement trajectory in space. *Exp Brain Research*, 10, 2005.

[10] M.J. McKeown and R. Radtke. Phasic and tonic coupling between EEG and MEG demonstrated with ICA. *Journal of Clinical Neuroplasticity*, 18, 2001.

[11] M.J. McKeown, L.K. Hansen, and T.J. Sejnowski. ICA analysis of fMRI: What is signal and what is noise? *Current Opinion in Neurobiology*, 13, 2003.

[12] M.F. Bear, B.W. Connors, and M.A. Paradiso. *Neuroscience: Exploring The Brain*. Lippincott, Williams Wilkins, 351 West Camden Street, Baltimore ,MD 21201-2436 USA, second edition, 2001.

[13] J.M. Carmena and M.A. Lebedev. Learning to control a BMI for reaching and grasping by primates. *PLoS Biology*, 1(2), October 2003.

[14] C. Guger and G. Pfurtscheller. Design of an EEG-based brain-computer interface (BCI) from standard components running in real-time under Windows. University of Technology Graz, Brockmanngasse 41, 8010 Graz, Austria.

[15] Wikipedia. Image:gray726-brodman.png. http://en.wikipedia.org/wiki/Image:Gray726-Brodman.png. License: This image is in the public domain because its copyright has expired.

[16] C.I. Hung and P.L. Lee. Recognition of motor imagery EEG using ICA and machine classifiers. *Neuroscience Letters*, 382, 2005.

[17] G. Pfurtscheller and C. Neuper. Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. *IEEE Transactions on Rehabilitation Engineering*, 6(3), September 1998.

[18] J.C. Sanchez and J.M. Carmena. Ascertaining the importance of neurons to develop better brain-machine interfaces. *IEEE Transactions on Biomedical Engineering*, 51(6), June 2004.

[19] J.C. Sanchez and J.C. Principe. Simultaneous prediction of four kinematic variables for a BMI using a single recurrent neural network. In *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, 2004.

[20] I. Obeid and J.C. Morizio. Two multichannel integrated circuits for neural recording and signal processing. *IEEE Transactions on Biomedical Engineering*, 50(2), February 2003.

[21] G. Pfurtscheller and G.R. Muller-Putz. 15 years of BCI research at Graz University of Technology: Current projects. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2), June 2006.

[22] G. Pfurtscheller and C. Neuper. Current trends in Graz BCI research. *IEEE Transactions on Rehabilitation Engineering*, 8(2), June 2000.

[23] C. Guger and A. Schlogl. Rapid prototyping of an EEG-based brain-computer interface. Article from the Pfurtsheller Lab, 1998.

[24] G.E. Fabiani, D.J. McFarland, J.R. Wolpaw, and G. Pfurtscheller. Conversion of EEG activity into cursor movement by a brain-computer interface(BCI). *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12(3), September 2004.

[25] S. Sun and C. Zhang. Assessing features for EEG signal categorization. Article from Stake Key lab of Intelligent Technology, 2005.

[26] S. Rezaei and K. Tavakolion. Different classification techniques considering BCI applications. *Journal of Neural Eng*, 3, 2006.

[27] G.R. Muller-Putz, R. Scherer, and G. Pfurtscheller. EEG-based neuroprosthesis control: A step toward clinical practice. *Annals of Biomedical Engineering*, 33, 2005.

[28] J.R. Millan and F. Renkens. Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering*, 51(6), June 2004.

[29] Wikipedia. Action potential. http://en.wikipedia.org/wiki/Action_Potential.

[30] Wikipedia. Image:action potential vert.png. http://en.wikipedia.org/wiki/Image:Action _ potential _ vert.png. License: Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. Subject to disclaimers.

[31] Wikipedia. Electroencephalography. http://en.wikipedia.org/wiki/Electroencephalography.

[32] Wikipedia. Image:utah array pat5215088.jpg. http://en.wikipedia.org/wiki/Image:Utah_array_pat5215088.jpg. License: This image originated from a United States patent which is in the public domain.

[33] T. Suzuki and D. Zeigler. Flexible neural probes with micro-fluidic channels for stable interface with the nervous system. In *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, 2004.

[34] S. Makeig and S. Enghoff. A natural basis for efficient brain-actuated control. *IEEE Trans on Rehabilitation Engineering*, 8, 2000.

[35] G. Schalk and D.J. McFarland. BCI2000: A general-purpose BCI system. *IEEE Transactions on Biomedical Engineering*, 51(6), June 2004.

[36] Wikipedia. Magnetoencephalography. http://en.wikipedia.org/wiki/Magnetoencephalography.

[37] Wikipedia. Image:squid prototype.jpg. http://en.wikipedia.org/wiki/Image:Squid_prototype.jpg. License: This image is copyrighted. However, the copyright holder has irrevocably released all rights to it, allowing it to be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited in any way by anyone for any purpose, commercial or non-commercial, with or without attribution of the author, as if in the public domain.

[38] Slashdot. Honda robot controlled by brain waves.
http://hardware.slashdot.org/article.pl?sid=06/05/25/1619249.

[39] Wikipedia. Functional magnetic resonance imaging. http://en.wikipedia.org/wiki/FMRI.

[40] Wikipedia. Image:fmri.jpg. http://en.wikipedia.org/wiki/Image:FMRI.jpg. License: This file has been released into the public domain by the copyright holder, its copyright has expired, or it is ineligible for copyright. This applies worldwide.

[41] L.I. Smith. *A Tutorial on Principle Components Analysis*, 2002.

[42] www.brainconnection.com. The anatomy of movement. http://www.brainconnection.com/topics/?main=anat/moto anat. License: This content is copyrighted material that may be used for non-commercial, research purposes only. Please see http://www.brainconnection.com/gen/?main=terms#permitted for detailed copyright information.

[43] Wikipedia. Image:sensory and motor homunculi.jpg. http://en.wikipedia.org/wiki/Image:Sensory _ and _ motor _ homunculi.jpg. License: This image has been released into the public domain.

[44] S.J. Luck. *An Introduction to the Event-Related Potential Technique (Cognitive Neuroscience)*. MIT Press, 2005.

[45] B. Carter. *Chapter 17: Circuit Board Layout Techniques*.

[46] M.A. Mayosky E.M. Spinelli. AC-coupled front-end for biopotential measurements. *IEEE Transactions on Biomedical Engineering*, 50(3), March 2003.

[47] Texas Instruments. INA 128, INA 129. sbos051b. http://www-s.ti.com/sc/ds/ina129.pdf.

[48] R.P. Areny and J.G. Webster. Common mode rejection ratio in differential amplifiers. *IEEE Transactions on Instrumentation and Measurement*, 30(4), August 1991.

[49] J.G. Webster. *Medical Instrumentation: Application and Design*. John Wiley and Sons, INC, 605 Third Ave, NYC ,NY 10158 USA, third edition edition, 1998.

[50] N.H. Martinez E.M. Spinelli. A transconductance driven-right-leg circuit. *IEEE Transactions on Biomedical Engineering*, 46(12), December 1999.

[51] S.R.Wiese and J.G. Webster. Electrocardiographic motion artifact versus electrode impedance. *IEEE Transactions on Biomedical Engineering*, 52(1), January 2005.

[52] O. Ozdamar and J. Bohorquez. Signal-to-noise ratio and frequency analysis of continuous loop averaging deconvolution (CLAD) of overlapping evoked potentials. *Acoustical Society of America*, 119(1), January 2006.

[53] E.A. Clancy and E.L. Morin. Sampling, noise-reduction and amplitude estimation issues in surface electromyography. *Journal of Electromyography and Kinesiology*, 12, 2002.

[54] P.S. Hamilton and M.G. Curley. Comparison of methods for adaptive removal of motion artifact. *IEEE Computers in Cardiology*, 27, 2000.

[55] A. Erfanian and B. Mahmoudi. Real-time ocular artifact suppression using recurrent neural network for electroencephalogram-based brain-computer interface. *Medical and Biological Engineering and Computing*, 43, 2005.

[56] G. Mihov C. Levkov. Removal of power-line interference from the ECG: a review of the subtraction procedure. *Biomedical Engineering Online*, 4(50), August 2005.

[57] C. Merritt. A pneumatically actuated brace designed for upper extremity stroke rehabilitation. Master's thesis, North Carolina State University, 2003.

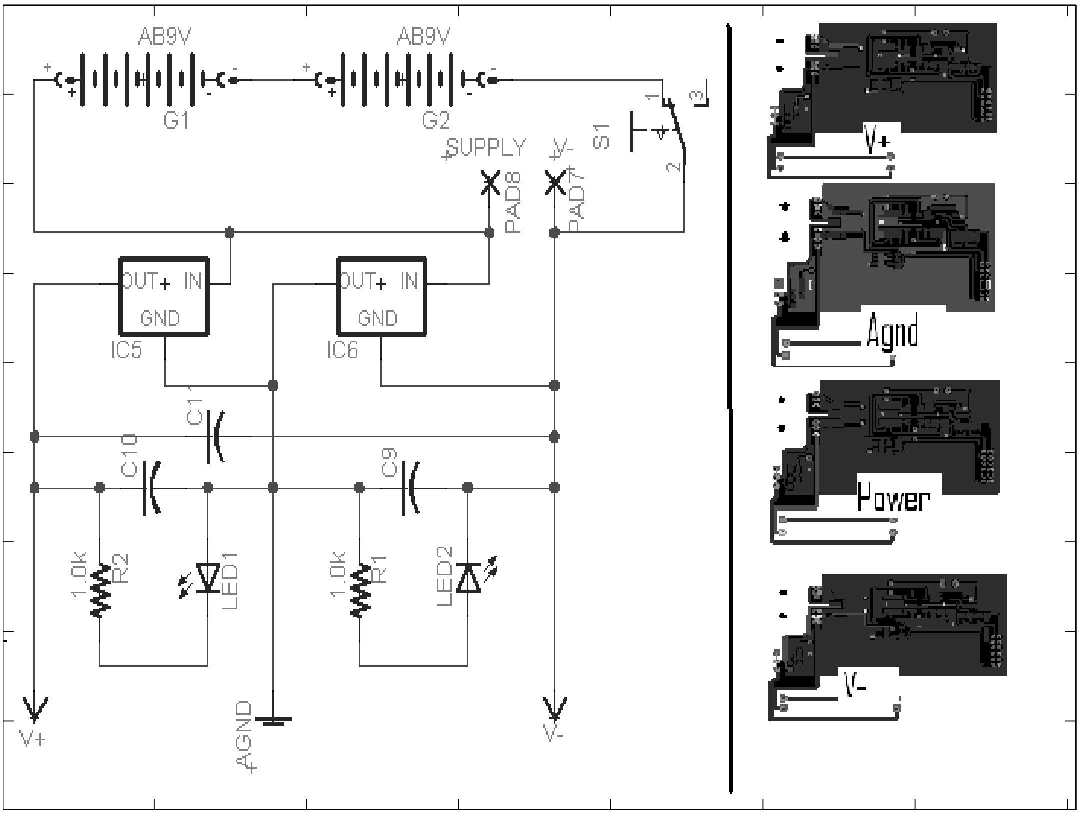
[58] M.J. Fu and J.J. Daly. Assessment of EEG event-related desynchronization in stroke survivors performing shoulder-elbow movements. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 2006.

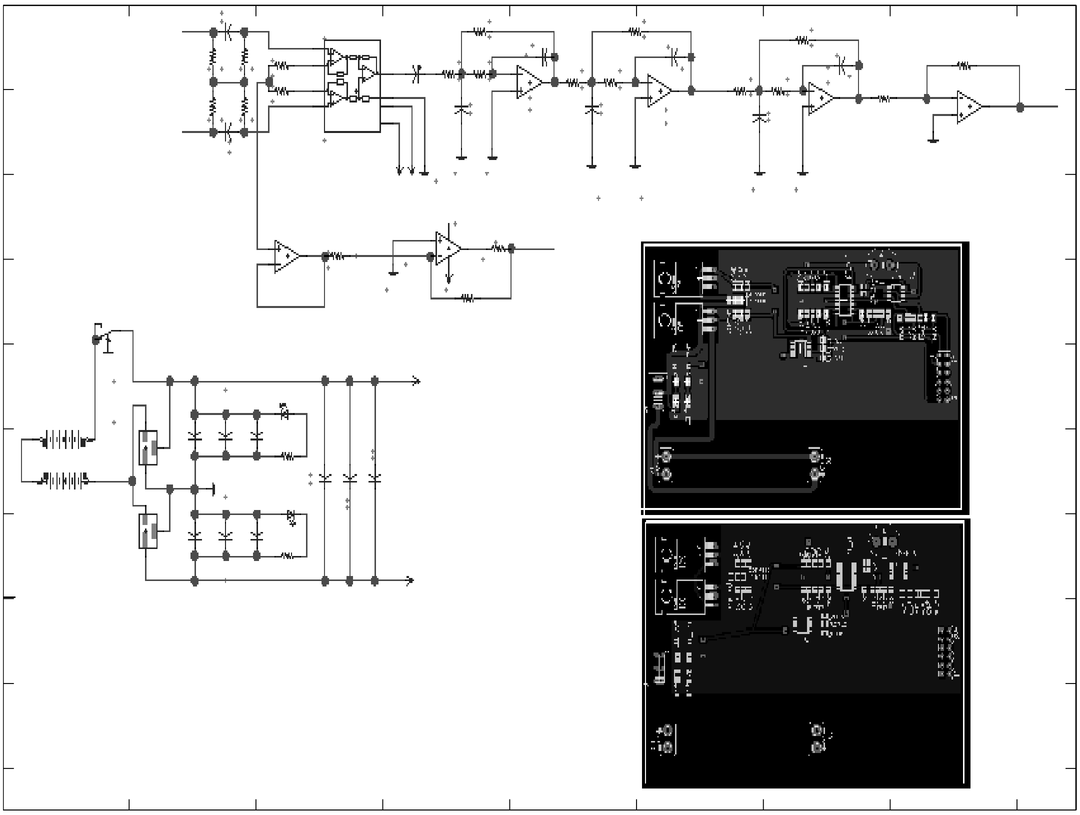
[59] Wikipedia. Support vector machine. http://en.wikipedia.org/wiki/Support_vector_machine.

[60] Wikipedia. Image:neural network.gif. http://en.wikipedia.org/wiki/Image:Neural _ Network.gif. License: This image has been (or is hereby) released into the public domain by its creator, Joerite. This applies worldwide. In case this is not legally possible, the creator grants anyone the right to use this work for any purpose, without any conditions, unless such conditions are required by law. Subject to disclaimers.

[61] Wikipedia. Neural network. http://en.wikipedia.org/wiki/Neural_Network.

Appendix A: EEG Circuit Schematic and Layout


AB9V
AB9V
G1
G2
SUPPLY
V-
PAD8
PAD7
S1
1
2
OUT+ IN
GND
IC5
OUT+ IN
GND
IC6
C11
C10
C9
1,0k
R2
LED1
1,0k
R1
LED2
V+
AGND
V-
V+
Agnd
Power
V-



Appendix B: BCI Software Code (Matlab)

```
% BCI GUI
% Author: Brooks Adcock
% Center for Robotic and Intelligent Machines
% Version: 3
% Updates: This version abandons multiple recording channels and
% substitutes them with information extracted from a single channel EEG
% signal

function main(varargin)
    clc
    clear
    close all hidden

    global gvars;

    t = timerfind;
    if(length(t)>0)
        deleteTimer(t);
    end

    setupGVars();
    setupGUI( 100 , 50 , 900 , 660 );
    setupDAQ();
    setupMonitorTimer();

%*************************************************************************
% Setup Global Variables
%*************************************************************************
function setupGVars()
    global gvars;
    %Initialize scope data
    gvars.secondsToShow = 10; %sec
    gvars.tshift = 0;

    %initialize recordable data
    gvars.minutesToRecord = 10;
    gvars.fs = 300;
    gvars.rawEEG = zeros(gvars.fs*60*gvars.minutesToRecord,1);
    gvars.recordedData = zeros(gvars.fs*60*gvars.minutesToRecord,2);
    gvars.computedData = zeros(gvars.fs*60*gvars.minutesToRecord,4);
    gvars.simulatedData = zeros(gvars.fs*60*gvars.minutesToRecord,1);
    gvars.recordedTime = zeros(gvars.fs*60*gvars.minutesToRecord,1);
    gvars.recordedIndex = 0;
    gvars.dataMutex = 0;

    % Neural Network
    gvars.nInputs = 15;   %samples
    gvars.simWindow = 15; %How many seconds of data the ANN uses for input
    iLim = 2*ones(gvars.nInputs,2);
    iLim(:,1) = iLim(:,1)*-1;
    iLim(:,2) = iLim(:,2)*1;

    gvars.NN = newff(iLim,[12 , 1],{'tansig','purelin'},'trainlm2');
    gvars.NN = init(gvars.NN);

    gvars.NN.performFcn = 'sse';
    gvars.NN.trainParam.show = 1;
    gvars.NN.trainParam.lr = 0.02;
    gvars.NN.trainParam.epochs = 100;
    gvars.NN.trainParam.goal = 2e0;
    gvars.NN.trainParam.max_fail = 10e10;
    %gvars.NN.trainParam.mem_reduc = 2;

%**************************************************************************
% Setup Graphical User Interface (GUI)
%**************************************************************************
function setupGUI(x,y,w,h)
    global gvars;
    mywindow = figure(...
                    'Name','Brain Machine Interface',...
                    'NumberTitle','off',...
                    'MenuBar','none',...
                    'Interruptible','on',...
                    'DeleteFcn',@callbackWindowClose,...
                    'Units','pixels',...
                    'Position',[x y w h],...
                    'Visible','on');
    myplot = axes(...
                    'Parent',mywindow,...
                    'Units','normalized',...
                    'Position',[.05 .15 .5 .825],...
                    'Color',[1 1 1],...
                    'XGrid','on',...
                    'XLim',[0 1],...
                    'XTick',...
                   round(linspace(0,gvars.fs*60*10,gvars.fs*60*10)),...
                    'YGrid','on',...
                    'YLim',[0 1],...
                    'YTick',linspace(-0,1,6),...
                    'YTickLabel',{'';'ARM';'Beta';'Alpha';'EEG';''},...
                    'Visible','on');
fftplot = axes(...
                    'Parent',mywindow,...
                    'Units','normalized',...
                    'Position',[.58 .15 .4 .4],...
                    'Color',[1 1 1],...
                    'XGrid','on',...
                    'XLim',[0 100],...
                    'XTick',[1 4 7 12 18 30 60],...
                    'YGrid','on',...
```

```
                ’YLim’,[0 1],...
                ’YTick’,(linspace(-0,1,3)),...
                ’Visible’,’on’);
gvars.myplot = myplot;

%**********************************
% Menu
%**********************************
mymenu = uimenu(...
’Parent’            ,mywindow ,...
’Label’             ,’Data’,...
’Interruptible’     ,’on’,...
’Visible’          ,’on’...
);

uimenu(...
’Parent’            ,mymenu ,...
’Label’             ,’Open Data’,...
’Interruptible’     ,’on’,...
’Visible’          ,’on’,...
’Callback’          ,@callbackFileOpenData...
);

uimenu(...
’Parent’            ,mymenu ,...
’Label’             ,’Save Data’,...
’Interruptible’     ,’on’,...
’Visible’          ,’on’,...
’Callback’          ,@callbackFileSaveData...
);
uimenu(...
’Parent’            ,mymenu ,...
’Label’             ,’Show Current Data’,...
’Interruptible’     ,’on’,...
’Visible’          ,’on’,...
’Callback’          ,@localDisplayAll...
);
% ANN menu
mymenu = uimenu(...
’Parent’            ,mywindow ,...
’Label’             ,’ANN Control’,...
’Interruptible’     ,’on’,...
’Visible’          ,’on’...
);
uimenu(...
’Parent’            ,mymenu ,...
’Label’             ,’Open NN’,...
’Interruptible’     ,’on’,...
’Visible’          ,’on’,...
’Callback’          ,@callbackFileOpenNN...
```

```
);
uimenu(...
'Parent'            ,mymenu ,...
'Label'             ,'Save NN',...
'Interruptible'     ,'on',...
'Visible'          ,'on',...
'Callback'          ,@callbackFileSaveNN...
);
uimenu(...
'Parent'            ,mymenu ,...
'Label'             ,'Train on current data',...
'Interruptible'     ,'on',...
'Visible'          ,'on',...
'Callback'          ,@localTrainNN...
);
%**********************************
% Lines
%**********************************
gvars.lines = [];
colorMap = [ .5 0 0 ;... EEG
             0 .5 0;... Alpha
             0 1  0 ;... Alpha Power
             0 0 .5 ;... Beta
             0 0 1 ;... Beta Power
             0 0 0 ;... ARM
             1 0 0];   %Predicted
for i= 1:size(colorMap,1),
    gvars.lines(i) = line(...
             'Parent',myplot,...
             'Color',colorMap(i,:),...
             'Visible','off');
end
gvars.spectrumLine = line(...
             'Parent',fftplot,...
             'Color',[ 0 0 1],...
             'Visible','off');
%**********************************
% Legend
%**********************************
gvars.legend = [];
nameMap = {'EEG (C3)' ;...
           'Alpha'    ;...
           'P(Alpha)' ;...
           'Beta'     ;...
           'P(Beta)'  ;...
           'ARM'      ;...
           'Actuator'};
valueMap = [0;...
            0;...
            1;...
            0;...
            0;...
            1;...
            1];...
for i= 1:size(colorMap,1),

    gvars.legend(i) = uicontrol(...
        ’Parent’    ,mywindow,...
        ’Style’     ,’checkbox’,...
        ’Units’     ,’normalized’,...
        ’Position’  ,[ .56 .95-(i-1)*.025 .1 .021],...
        ’String’    ,nameMap{i},...
        ’Value’     ,valueMap(i),...
        ’ForegroundColor’     ,get(gvars.lines(i),’Color’),...
        ’Visible’   ,’on’);

end

%**********************************
% Buttons
%**********************************
gvars.buttonAcquire = uicontrol(...
                ’Parent’    ,mywindow,...
                ’Style’     ,’pushbutton’,...
                ’Units’     ,’normalized’,...
                ’Position’  ,[ .7 .89 .25 .08],...
                ’String’    ,’Start Acquisition’,...
                ’Value’     ,0,...
                ’Callback’  ,@callbackAcquireButton,...
                ’Visible’   ,’on’);
 gvars.buttonTrain = uicontrol(...
                ’Parent’    ,mywindow,...
                ’Style’     ,’pushbutton’,...
                ’Units’     ,’normalized’,...
                ’Position’  ,[ .7 .8 .25 .08],...
                ’String’    ,’Halt and Train’,...
                ’Value’     ,0,...
                ’Callback’  ,@callbackHaltAndTrain,...
                ’Visible’   ,’off’);

uicontrol(... Zoom Out
    ’Parent’    ,mywindow,...
    ’Style’     ,’pushbutton’,...
    ’Units’     ,’normalized’,...
    ’Position’  ,[ .25 .015 .1 .05],...
    ’String’    ,’Zoom Out’,...
    ’Value’     ,0,...
    ’Callback’  ,@callbackZoomOut,...
    ’Visible’   ,’on’);
uicontrol(... Zoom IN
        'Parent'    ,mywindow,...
        'Style'     ,'pushbutton',...
        'Units'     ,'normalized',...
        'Position'  ,[ .25 .07 .1 .05],...
        'String'    ,'Zoom In',...
        'Value'     ,0,...
        'Callback'  ,@callbackZoomIn,...
        'Visible'   ,'on');
    uicontrol(... Shift Left
        'Parent'    ,mywindow,...
        'Style'     ,'pushbutton',...
        'Units'     ,'normalized',...
        'Position'  ,[ .14 .045 .1 .05],...
        'String'    ,'Shift Left',...
        'Value'     ,0,...
        'Callback'  ,@callbackShiftLeft,...
        'Visible'   ,'on');
    uicontrol(... Shift Right
        'Parent'    ,mywindow,...
        'Style'     ,'pushbutton',...
        'Units'     ,'normalized',...
        'Position'  ,[ .36 .045 .1 .05],...
        'String'    ,'Shift Right',...
        'Value'     ,0,...
        'Callback'  ,@callbackShiftRight,...
        'Visible'   ,'on');

    %*********************************
    % Buttons
    %*********************************
    uicontrol(...
                'Parent'    ,mywindow,...
                'Style'     ,'text',...
                'BackgroundColor'     ,[.8 .8 .8],...
                'Units'     ,'normalized',...
                'Position'  ,[ .64 .56 .3 .02],...
                'String'    ,...
          'Normalized Power Spectral Density v.s. Frequency (Hz)',...
                'Visible'   ,'on');
    uicontrol(...
                'Parent'    ,mywindow,...
                'Style'     ,'text',...
                'BackgroundColor'     ,[.8 .8 .8],...
                'Units'     ,'normalized',...
                'Position'  ,[ .87 .125 .1 .02],...
                'String'    ,'Frequency (Hz)',...
                'Visible'   ,'on');
%*************************************************************************
% Setup DAQ
%*************************************************************************
```

```
function setupDAQ()
    global gvars;



    % Find and Kill any open DAQs
    openDAQ = daqfind;
    for i = 1:length(openDAQ),
      stop(openDAQ(i));
      delete(openDAQ(i));
    end

    % Setup current daq
    %NIDAQ PCI-6025E
%     gvars.ai = analoginput('nidaq','1');
%
%     addchannel(gvars.ai,0); % EEG
%     addchannel(gvars.ai,7); % Arm
%     set(gvars.ai,...
%             'TriggerRepeat', 0,...
%             'TriggerType', 'Manual',...
%             'InputType', 'Differential',...
%             'SampleRate',gvars.fs,...
%             'SamplesPerTrigger', gvars.fs*60*gvars.minutesToRecord,...
%             'TimerPeriod', .1,...
%             'Timerfcn', @callbackDAQAction,...
%             'Stopfcn', @callbackDAQStopFunction);



%     %DATAQ 158U
    daqregister('C:\WINDAQ\MATLAB\dataq.dll')
    gvars.ai = analoginput('dataq');
    ch1 = addchannel(gvars.ai,3); % EEG
        %set(ch1,'SensorRange', [ -1 1])
        %set(ch1,'InputRange',[-6 6])
    ch2 = addchannel(gvars.ai,0); % Arm
        %set(ch1,'SensorRange', [ -1 1])
        %set(ch1,'InputRange',[-10 10])

    set(gvars.ai,        'TriggerRepeat', 0);
    set(gvars.ai,        'SampleRate',gvars.fs);
    set(gvars.ai,        'SamplesPerTrigger',...
      gvars.fs*60*gvars.minutesToRecord);
    set(gvars.ai,        'TimerPeriod', .1);
    set(gvars.ai,        'Timerfcn', @callbackDAQAction);
    set(gvars.ai,        'Stopfcn', @callbackDAQStopFunction);

   % AI triggered on button press
```

```
%**************************************************************************
% Setup Monitor Timer
%**************************************************************************
function setupMonitorTimer()
    global gvars;
    gvars.monitorTimer = timer(’TimerFcn’,@callbackMonitorTimer,...
                               ’Period’,.12,... frames per second
                               ’ExecutionMode’,’fixedRate’,...
                               ’BusyMode’,’drop’);
    %Timer Started by button press

%**************************************************************************
% Callback: Monitor Timer - displays lines
%**************************************************************************
%TODO Rewrite this to take any number of points
function callbackMonitorTimer(varargin)
    global gvars;

    % Dont Bother Displaying until ya got somethin’
    if(gvars.recordedIndex < 10)
        return;
    end

    %OBEY THE MUTEX!
    if( gvars.dataMutex == 1)
        return;
    end
    gvars.dataMutex = 1;

    try
        % Compute Time range
        tdata = localGetTimeAxis();
            tmax = tdata(1);
            tmin = tdata(2);

        % Set the Data
        displayData = localCreateDisplayData([tmin tmax]);

        xdata = displayData(8,:);
        ydata = displayData(1:7,:);
       % Draw the lines
        set(gvars.myplot,’XLim’,[tmin tmax]);
        for i=1:length(gvars.legend),
            if(get(gvars.legend(i),’Value’)==1)
                set(gvars.lines(i),’XData’,xdata);
                set(gvars.lines(i),’YData’,ydata(i,:));
                set(gvars.lines(i),’Visible’,’on’);
            else
                set(gvars.lines(i),’Visible’,’off’);
            end
        end

        % Compute the Power Spectrum
        tmax = (gvars.recordedIndex-1)/gvars.fs;
        if(tmax > 1)
            spectrum = localComputeSpectrum([tmax-.1 tmax]);
            scale = max(max(spectrum(1,:)),.000001);
            spectrum(1,:) = spectrum(1,:)/scale;
            set(gvars.spectrumLine,’XData’,spectrum(2,:));
            set(gvars.spectrumLine,’YData’,spectrum(1,:));
            set(gvars.spectrumLine,’Visible’,’on’);
        end
    catch
        disp(’Error in callbackMonitorTimer’)
        catchErrorFn();
        gvars.dataMutex = 0;
    end
    % Release Mutex back into the wild
    gvars.dataMutex = 0;
%*************************************************************************
% Callback: Window Closing
%*************************************************************************
function callbackWindowClose(varargin)
    global gvars;
    callbackStopAndClear();
    deleteTimer(gvars.monitorTimer);
    callbackDAQStopFunction();
    daqregister(’C:\WINDAQ\MATLAB\dataq.dll’,’unload’)
    delete(gvars.ai);

%*************************************************************************
% Callback: DAQ Action
%*************************************************************************
function callbackDAQAction(varargin)
    global gvars;

    if(gvars.ai.SamplesAvailable <= 0 )
        % No samples yet taken
        return;
    end

    % Obey the MUTEX!
    if( gvars.dataMutex == 1)
        return;
    end
    gvars.dataMutex = 1;

    try
```

```
% Record Bounds of Signal
sa = gvars.ai.SamplesAcquired;
sr = gvars.recordedIndex;
clipSize = sa-sr; % should equal SamplesAvailable property

% Get Samples
[data,tdata] = getdata(gvars.ai,clipSize);

%Record RAW data
gvars.rawEEG(sr+1:sa,1) = data(:,1); %Raw unfiltered EEG
gvars.recordedData(sr+1:sa,2) = data(:,2)-.58;  %ARM
gvars.recordedTime(sr+1:sa) = tdata;        %Time

%Compute and Record Filtered data
%(EEG, alpha , beta, power bands)
%process on 2 seconds giving 1/2Hz resolution
 % Compute the indices needed to do the signal processing
    past_2_sec = (sa - 2*gvars.fs)+1:sa; %600 samples
    past_1_sec = (sa - 1*gvars.fs)+1:sa; %300 samples
    last_1_sec = (1:gvars.fs)+gvars.fs;  %300 samples
    past_10th_sec = sr+1:sa;      %30 samples (usually)

%Don’t try to filter past values
if( past_2_sec(1) > 1 )
  %the past 2 seconds of EEG data
    dataToFilter = gvars.rawEEG(past_2_sec,1)’;

    % SIGNAL FILTERING
    % EEG, Alpha and Beta Wave Extraction
    tempEEG = myBandpassFilter(dataToFilter,gvars.fs,3,35);
    tempAlpha = myBandpassFilter(dataToFilter,gvars.fs,8,12);
    tempBeta = myBandpassFilter(dataToFilter,gvars.fs,18,28);

    %Record data, truncating ends of each signal
    gvars.recordedData(past_1_sec,1) =...
        tempEEG(last_1_sec)’;%EEG
    gvars.computedData(past_1_sec,1) =...
      tempAlpha(last_1_sec)’;%Alpha Wave
    gvars.computedData(past_1_sec,3) =...
      tempBeta(last_1_sec)’;%Beta Wave

    % FEATURE EXTRACTION
    %Compute Alpha and Beta Band Power
    %Compute Power in the 8-12 and 18-30 bands
    power = localBandPower(tempEEG,[8 12 ; 18 30]);
    gvars.computedData(past_10th_sec,2) = ...
     ones(size(past_10th_sec))*mean(power(1,:));   % Alpha Power
    gvars.computedData(past_10th_sec,4) = ...
     ones(size(past_10th_sec))*mean(power(2,:));   % Beta Power
```

```
            % PREDICTION
            %Simulation has a one second delay (1*fs)
            simpoint = sa-1*gvars.fs;

            %Data from time (t-1 = simpoint) is used to render the
            %prediction for time t
            gvars.simulatedData(past_10th_sec) = ...
    ones(size(past_10th_sec))*mean(localSimNN(simpoint,simpoint));

            % OUTPUT SMOOTHING
            gvars.simulatedData(past_10th_sec) = ...
  myThresholdFilter(gvars.simulatedData(past_10th_sec),[-inf .1],0);
                %gvars.simulatedData(sr+1:sa) = ...
 ones(size(sr+1:sa))*mean(gvars.simulatedData(max(1,sa-2*gvars.fs):sa));
                %gvars.simulatedData(sr+1:sa) = ...
  myThresholdFilter(gvars.simulatedData(sr+1:sa),[-inf .1],0);

        end


        %Update current recording index
        gvars.recordedIndex=sa; %Varable used to keep track of progress

    catch
        disp('Error in DAQ Action')
        catchErrorFn();
        gvars.dataMutex = 0;
    end
    gvars.dataMutex = 0;

%**************************************************************************
% Callback: Daq Stop Function
%**************************************************************************
function callbackDAQStopFunction(varargin)
    global gvars;
    %TODO: make this handle the data acquisiton stop
    if isvalid(gvars.ai)
        if ~isempty(gvars.ai)

            stop(gvars.ai)

            %tone([100],.5);
        end
    end
%**************************************************************************
% Callback: Acquire Button
%**************************************************************************
function callbackAcquireButton(varargin)
    global gvars;
    gvars.minutesToRecord = 10;
    gvars.recordedData = zeros(size(gvars.recordedData));
    gvars.computedData = zeros(size(gvars.computedData));
    gvars.simulatedData = zeros(size(gvars.simulatedData));
    gvars.recordedTime = zeros(size(gvars.recordedTime));
    gvars.recordedIndex = 0;
    gvars.dataMutex = 0;


    % Start Monitoring
    assertTimer(gvars.monitorTimer,'on');

    % Start DAQ
    rn = get(gvars.ai,'Running');
    if(isequal(rn,'Off'))
        start(gvars.ai);
        %trigger(gvars.ai);
    end

    set(gvars.buttonAcquire,'String','Halt and Clear Data');
    set(gvars.buttonAcquire,'Callback',@callbackStopAndClear);

    set(gvars.buttonTrain,'Visible','on');

%*************************************************************************
% Callback: Stop Timers / Display and Clear Data
%*************************************************************************
function callbackStopAndClear(varargin)
    global gvars;

    % Start Monitoring
    assertTimer(gvars.monitorTimer,'off');

    % Start DAQ
    rn = get(gvars.ai,'Running');
    if(isequal(rn,'On'))
        stop(gvars.ai);
    end

    % Erase Data
    l = length(gvars.recordedTime);
    gvars.recordedData = zeros(size(gvars.recordedData));
    gvars.computedData = zeros(size(gvars.computedData));
    gvars.simulatedData = zeros(size(gvars.simulatedData));
    gvars.recordedTime = zeros(size(gvars.recordedTime));
    gvars.recordedIndex = 1;

    % Clear Traces
    for i=1:length(gvars.lines),
        set(gvars.lines(i),'Visible','off');
    end
    set(gvars.spectrumLine,’Visible’,’off’);
    % Reset Button
    set(gvars.buttonAcquire,’String’,’Start Acquisition’);
    set(gvars.buttonAcquire,’Callback’,@callbackAcquireButton);

    % Reset Train Button
    set(gvars.buttonTrain,’Visible’,’off’);

%*************************************************************************
% Callback: Halt and Train
%*************************************************************************
function callbackHaltAndTrain(varargin)
    global gvars

        % Start Monitoring
    assertTimer(gvars.monitorTimer,’off’);

    % Start DAQ
    rn = get(gvars.ai,’Running’);
    if(isequal(rn,’On’))
        stop(gvars.ai);
    end

    % Update this button
    set(gvars.buttonTrain,’String’,’Halt Training’);
set(gvars.buttonTrain,’Visible’,’off’);

    % Update Acquisition Button
    set(gvars.buttonAcquire,’String’,’Start Acquisition’);
    set(gvars.buttonAcquire,’Callback’,@callbackAcquireButton);
    set(gvars.buttonAcquire,’Visible’,’off’);

    pause(.1);

    % Start Training
    localTrainNN();

    % Restore button
    set(gvars.buttonAcquire,’Visible’,’on’);
    callbackStopAndClear();

%*************************************************************************
% Open File Callback
%*************************************************************************
function callbackFileOpenNN(varargin)
    global gvars;
    [fname,fpath] = uigetfile(’*.mat’,’Open Log File’);
    if˜(fname == 0)
        filename = fullfile(fpath,fname);
        % Recover data
        fdata = load(filename,'myNN');
        gvars.NN = fdata.myNN;
    end
function callbackFileOpenData(varargin)
    global gvars;
    [fname,fpath] = uigetfile('*.mat','Open Log File');
    if~(fname == 0)
        filename = fullfile(fpath,fname);
        % Recover data
        fdata = load(filename,'myRawData','myRecData','myComData',...
        'myIndex','myTime');
        gvars.rawEEG = fdata.myRawData;
        gvars.recordedData = fdata.myRecData;
        gvars.computedData = fdata.myComData;
        gvars.recordedIndex = fdata.myIndex;
        gvars.recordedTime = fdata.myTime;

    end
    localDisplayAll();


%**************************************************************************
% Save File Callback
%**************************************************************************
function callbackFileSaveNN(varargin)
    global gvars;
    try
        fname = ['NNfile_' getDateString()];
        [fname,fpath] = uiputfile(fname,'Save Log File');
        if~(fname == 0)
            filename = fullfile(fpath,fname);
            myNN = gvars.NN;
            save(filename,'myNN');
        end
    catch
        catchErrorFn();
    end

function callbackFileSaveData(varargin)
    global gvars;
    try
        fname = ['DataFile_' getDateString()];
        [fname,fpath] = uiputfile(fname,'Save Log File');
        if~(fname == 0)
            filename = fullfile(fpath,fname);
            myRawData = gvars.rawEEG;
            myRecData = gvars.recordedData;
            myComData = gvars.computedData;
            myIndex = gvars.recordedIndex;
            myTime = gvars.recordedTime;
            save(filename,'myRawData','myRecData','myComData',...
             'myIndex','myTime');
        end
    catch
        catchErrorFn();
    end

%*************************************************************************
% Callback: Zoom In
%*************************************************************************
function callbackZoomIn(varargin)
    global gvars;
    gvars.secondsToShow = gvars.secondsToShow *.7;

%*************************************************************************
% Callback: Zoom Out
%*************************************************************************
function callbackZoomOut(varargin)
    global gvars;
    gvars.secondsToShow = gvars.secondsToShow /.7;

%*************************************************************************
% Callback: Shift Right
%*************************************************************************
function callbackShiftRight(varargin)
    global gvars;
    gvars.tshift = gvars.tshift + gvars.secondsToShow *.3;

%*************************************************************************
% Callback: Shift Left
%*************************************************************************
function callbackShiftLeft(varargin)
    global gvars;
    gvars.tshift = gvars.tshift - gvars.secondsToShow *.3;

%*************************************************************************
% Callback: Error Reporting
%*************************************************************************
function catchErrorFn()
    e = lasterror;
    msg = e.message;
    line = [e.stack.line];
    myerror = ['Internal Error on line ' num2str(line) ': ' msg];
    disp(myerror);

%*************************************************************************
% Local Function: Get Time Axis
%*************************************************************************
function rv = localGetTimeAxis(varargin)
```

```
    global gvars;
    tspan = gvars.secondsToShow;
    tshift = gvars.tshift;
    tcurrent = gvars.recordedTime(gvars.recordedIndex);
    tmax = max(tcurrent+tshift,tspan);
    tmin = max(0,tmax-tspan);

    rv = [tmax tmin];

%************************************************************************
% Local Function: Get Time Axis
%************************************************************************
function y = localCreateDisplayData(trange)
    global gvars;

        tmin = trange(1);
        tmax = trange(2);
    % Compute corresponding index range
        itop = gvars.recordedIndex-gvars.fs*1;
        imax = max(1,min(round(tmax*gvars.fs),itop));
        imin = max(1,min(round(tmin*gvars.fs),itop));

    % Undersample for display (speed reasons)
        indices = round(linspace(imin,imax,gvars.secondsToShow*gvars.fs));

    %Grab Desired Data
        y = zeros(8,length(indices));
        yscales = [ 10ˆ0 10ˆ1 10ˆ3 10ˆ1 10ˆ3 .5 .5]*.2;
        yshifts = [ 4 3 3 2 2 1 1]*.2;
        y(1,:) = yscales(1)*gvars.rawEEG(indices,1)’+yshifts(1);  %EEG
        %Alpha
        y(2,:) = yscales(2)*gvars.computedData(indices,1)’+yshifts(2);
        %P(Alpha)
        y(3,:) = yscales(3)*gvars.computedData(indices,2)’+yshifts(3);
        %Beta
        y(4,:) = yscales(4)*gvars.computedData(indices,3)’+yshifts(4);
        %P(Beta)
        y(5,:) = yscales(5)*gvars.computedData(indices,4)’+yshifts(5);
        %ARM
        y(6,:) = yscales(6)*gvars.recordedData(indices,2)’+yshifts(6);
        %Prediction
        y(7,:) = yscales(7)*gvars.simulatedData(indices,1)’+yshifts(7);
        y(8,:) = gvars.recordedTime(indices,1);


     %Return


%************************************************************************
% Local Function: Computes the spectrum
```

```
%***************************************************************************
function y = localComputeSpectrum(trange)
    global gvars;

    % Compute Time bounds
        tmin = trange(1);
        tmax = trange(2);
    % Compute corresponding index range
        imax = max(1,min(round(tmax*gvars.fs),gvars.recordedIndex-1));
        imin = max(1,min(round(tmin*gvars.fs),gvars.recordedIndex-1));

    % Undersample for display (speed reasons)
        indices = round(linspace(imin,imax,imax-imin));

    y = zeros(2,129);
    x = gvars.recordedData(indices,1)’;
    mM = minmax(x);
    if(mM(1)~=mM(2))
        % Compute Spectrum

        [s f] = pburg(x,10,[],300);

        y(1,:) = s;
        y(2,:) = f;
    end

function varout = localBandPower(data,freqs)
    global gvars;

    kspan = length(data);
    pwr = ones(size(freqs,1),kspan);

    if(std(data)<.000001)
        % This is a safety check
        varout = pwr*0;
        return
    end

    [spectrum f] = pburg(data,10,[],gvars.fs);

    df = f(2)-f(1);

    for j=1:size(freqs,1),
        istart = min(max(round(freqs(j,1)/df+1),1),kspan);
        iend = min(max(round(freqs(j,2)/df+1),1),kspan);
        pwr(j,:) = pwr(j,:)*sum(spectrum(istart:iend));
    end


varout = pwr;
```

```
%*************************************************************************
% Local Function: Display All the data
%*************************************************************************
function localDisplayAll(varargin)
    global gvars;

    if gvars.recordedIndex < 5
        return
    end

        % Compute Time range
            tmax = gvars.minutesToRecord;
            tmin = 0;

        % Set the Data
        displayData = localCreateDisplayData([tmin tmax]);

        xdata = displayData(8,:);
        ydata = displayData(1:7,:);
       % Draw the lines
        set(gvars.myplot,’XLim’,[tmin tmax]);
        for i=1:length(gvars.legend),
            if(get(gvars.legend(i),’Value’)==1)
                set(gvars.lines(i),’XData’,xdata);
                set(gvars.lines(i),’YData’,ydata(i,:));
                set(gvars.lines(i),’Visible’,’on’);
            else
                set(gvars.lines(i),’Visible’,’off’);
            end
        end

        % Compute the Power Spectrum
        tmax = (gvars.recordedIndex-1)/gvars.fs;
        if(tmax > 1)
            spectrum = localComputeSpectrum([tmax-.1 tmax]);
            scale = max(max(spectrum(1,:)),.000001);
            spectrum(1,:) = spectrum(1,:)/scale;
            set(gvars.spectrumLine,’XData’,spectrum(2,:));
            set(gvars.spectrumLine,’YData’,spectrum(1,:));
            set(gvars.spectrumLine,’Visible’,’on’);
        end

    assertTimer(gvars.monitorTimer,’on’)

%*************************************************************************
% Local Function: Train the neural network
%*************************************************************************
function localTrainNN(varargin)
    global gvars;
    % Format the data for learning
    indices = 1:gvars.recordedIndex;
    INPUT = gvars.computedData(indices,2)’; %P(Alpha)
    TARGET = gvars.recordedData(indices,2)’; %ARM Position

    % Cull the ugly parts of the data
    good_indices = 5*gvars.fs:1:length(INPUT)-5*gvars.fs;
    INPUT = INPUT(good_indices)/max(INPUT(good_indices));
    TARGET = TARGET(good_indices)/max(TARGET(good_indices));

    n_inputs = gvars.nInputs;
    input_spacing = round((gvars.simWindow*gvars.fs)/n_inputs);
    trial_spacing = get(gvars.ai,’TimerPeriod’)*gvars.fs;
    target_indices = round((input_spacing*(n_inputs-1))+1:length(TARGET));

    INPUT = createInputVector(INPUT,input_spacing,n_inputs,trial_spacing);
    TARGET = createInputVector(TARGET(target_indices),1,1,trial_spacing);
    size(INPUT)
    size(TARGET)
    % Train NN
    disp(’training begun’)
    tic
    [gvars.NN, tr] = train(gvars.NN,INPUT,TARGET);
    toc
    disp(’training complete’)

    load gong;
    wavplay(y,Fs)

    % Backfill predicted data to show results
    disp(’back-filling simulating data’)
    rv = localSimNN(1,gvars.recordedIndex);
    gvars.simulatedData = zeros(size(gvars.simulatedData));
    for k = 0:trial_spacing-1,
        indices = gvars.recordedIndex-trial_spacing*length(rv)+...
         1-k:trial_spacing:gvars.recordedIndex-k;
        gvars.simulatedData(indices,1) = rv;
    end
    localFilterSim(1,gvars.recordedIndex);
    disp(’back-filling complete’)
    disp(’done’)

%*************************************************************************
% Local Function: Simulate the NN
%*************************************************************************
function rv = localSimNN(istart,iend)
    global gvars;

    %Compute how many samples to use as inputs for 1 iteration
    %   (this will be undersampled later)
    ispan = gvars.simWindow*gvars.fs;

    %If the desired simulation window is too small resize it to perform 1
    %   iteration
    %If the window is larger than 1 iteration, perform multiple iterations
    if(ispan > (iend-istart))
        istart = iend-ispan;
        if(istart<1)
           rv = 0;
           return;
        end
    end

    % Acquire Input Values
    indices = istart:iend;
    INPUT = gvars.computedData(indices,2)’; %P(Alpha)

    % Normalize Inputs
    INPUT = INPUT / max(gvars.computedData(:,2));

    % Create Input Matrix
    n_inputs = gvars.nInputs;
    input_spacing = round((gvars.simWindow*gvars.fs)/n_inputs);
    trial_spacing = get(gvars.ai,’TimerPeriod’)*gvars.fs;

    INPUT = createInputVector(INPUT,input_spacing,n_inputs,...
     trial_spacing);

    % Execute Simulation
    rv = sim(gvars.NN,INPUT);

    % De-Normalize Output
    rv = rv*max(gvars.recordedData(:,2));

    % Output Smoothing done in DAQ Action

%*************************************************************************
% Local Function: Filter the signal on a given region
%*************************************************************************
function localFilterSim(istart,iend)
    global gvars

    indices = istart:iend;
    trial_spacing = get(gvars.ai,’TimerPeriod’)*gvars.fs;
    gvars.simulatedData(indices) = ...
     myThresholdFilter(gvars.simulatedData(indices),[-inf .1],0);
    gvars.simulatedData(indices) = ...
     myAverageFilter(gvars.simulatedData(indices),2*gvars.fs);
    gvars.simulatedData(indices) = ...
     myThresholdFilter(gvars.simulatedData(indices),[-inf .1],0);
```

Appendix C: IRB Application

North Carolina State University
Institutional Review Board for the Use of Human Subjects in Research
SUBMISSION FOR NEW STUDIES

Title of Project:

Principal Investigator *D. Brooks Adcock Jr.* Department Biomedical Engineering

Source of Funding (**required** information): This is a low budget project that is funded personally by the investigator.
(if externally funded include sponsor name and university account number)

Campus Address 890 Oval Dr (EB2) room 2034. Raleigh NC 27607-7911

Email: dbadcock@ncsu.edu Phone: (919)-606-7003 Fax:

RANK: ☐ Faculty
☒ Student: ☐Undergraduate; ☒ Masters; or ☐PhD
☐ Other (specify): ______

As the principal investigator, my signature testifies that I have read and understood the University Policy and Procedures for the Use of Human Subjects in Research. I assure the Committee that all procedures performed under this project will be conducted exactly as outlined in the Proposal Narrative and that any modification to this protocol will be submitted to the Committee in the form of an amendment for its approval prior to implementation.

Principal Investigator:

D. Brooks Adcock Jr	______________________________	10/2/2006
(typed/printed name)	(signature)	(date)

As the faculty sponsor, my signature testifies that I have reviewed this application thoroughly and will oversee the research in its entirety. I hereby acknowledge my role as the ***principal investigator of record****.*

Faculty Sponsor:

Dr. Edward Grant (PhD)	______________________________	10/2/2006
(typed/printed name)	(signature)	(date)

PLEASE COMPLETE IN DUPLICATE AND DELIVER, ALONG WITH A PROPOSAL NARRATIVE, TO:
Institutional Review Board, Box 7514, or email as an attachment to debra_paxton@ncsu.edu

****

For SPARCS office use only
Reviewer Decision (Expedited or Exempt Review)
☐ Exempt ☐ Approved ☐ Approved pending modifications ☐ Table

Expedited Review Category: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☒ 7 ☐ 8a ☐ 8b ☐ 8c ☐ 9

______________________	______________________	______________
Reviewer Name	Signature	Date

North Carolina State University
Institutional Review Board for the Use of Human Subjects in Research
GUIDELINES FOR A PROPOSAL NARRATIVE

In your narrative, address each of the topics outlined below. Every application for IRB review must contain a proposal narrative, and failure to follow these directions will result in delays in reviewing/processing the protocol.

A. INTRODUCTION

1. Briefly describe in lay language the purpose of the proposed research and why it is important.

   This project aims to test a Biosignal Computer Interface (an electrical signal emitted by the body that can be read non-invasively on the skin) that has been developed by the researcher for predicting body movement. This interface has been previously created at other universities, and this is a duplication of other research. The final result of this project will hopefully be used by other researchers at NCSU to further biosignal interface research to ultimately produce a mechanism that could permit a paralyzed person to control a wheel chair without making any physical movements.

   Electrical activity will be read via EEG (electroencephalogram) located on the scalp. Arm position will be measured via an arm brace located on the subject's left arm. Subjects will be asked to move their arm for a period of ten minutes during which time data will be recorded. Data from arm and EEG sensors will be analyzed in software to unveil predictive patterns. Predictive patterns will be tested for accuracy.

2. If student research, indicate whether for a course, thesis, dissertation, or independent research.
   a. **Masters Thesis**

B. SUBJECT POPULATION

1. How many subjects will be involved in the research?

   The initial sample size for this study will be 20 subjects. If more data is needed, the researcher will file for a continuation.

2. Describe how subjects will be recruited. Please provide the IRB with any recruitment materials that will be used.

   Subjects will be fellow students or lab partners of the researcher. They will be asked verbally if they desire to participate. Some "talking points" of verbal information provided to potential subjects are attached.

3. List specific eligibility requirements for subjects (or describe screening procedures), including those criteria that would exclude otherwise acceptable subjects.
   a. **Inclusion criteria**
      i. **Availability**
   b. **Exclusion criteria**
      i) A potential subject will be excluded if the subject…
         Is pregnant
         Has a history of seizures, allergic reactions to topical ointments, stroke, or cranial surgery
         Requires oxygen
         Has a pacemaker or other implanted electrical device
         Has a history of medical problems with the limbs including arthritis, surgery, fractures, etc
         Has problems sitting or standing
         Has strabismus
         Has spastic activity of the face, eyes or mouth

ii) A potential subject will be excluded if the subject has excessive hair. Excessive hair makes electrode placement and removal uncomfortable for the subject and difficult for the researcher.

4. Explain any sampling procedure that might exclude specific populations.
   As there are no strenuous inclusion criteria, subjects will be selected based on convenience. Thus subjects will primarily be students of the CRIM lab and others working regularly in EB2.

5. Disclose any relationship between researcher and subjects - such as, teacher/student; employer/employee.
   Anticipated relationships may be lab mates or acquaintances.

6. Check any vulnerable populations included in study:

minors (under age 18) - if so, have you included a line on the consent form for the parent/guardian signature

fetuses

pregnant women

persons with mental, psychiatric or emotional disabilities

persons with physical disabilities

economically or educationally disadvantaged

prisoners

elderly

students from a class taught by principal investigator

other vulnerable population.

If any of the above are used, state the necessity for doing so. Please indicate the approximate age range of the minors to be involved.

Vulnerable populations will not be recruited for this study. Recruited subjects will be between the ages of 18 and 60 years of age.

C. PROCEDURES TO BE FOLLOWED

1. In lay language, describe completely all procedures to be followed during the course of the experimentation. Provide sufficient detail so that the Committee is able to assess potential risks to human subjects.
   1) **Subjects will be provided with consent form and any questions will be answered**
   2) **Subject is asked to make themselves comfortable in a bean bag chair.**
   3) **Subject is assisted in putting on an arm brace on the left arm**
   4) **Skin electrode site areas are prepped with a slightly abrasive cleaning gel.**
   5) **Skin electrodes are filled with gel, placed on sites and taped down**
   6) **Skin electrode contact is tested by looking at signal noise and adjusted if necessary.**
   7) **Subject is asked to close their eyes and relax**
   8) **Subject is instructed to relax with eyes closed and to flex their left arm at whim for the next 10min.**
   9) **Data is recorded for 10min**
   10) **Subject is allowed open eyes and relax while Data is analyzed (80 seconds)**
       a. **Computer analysis performed on simulated data has thus far taken no more than 74 seconds. If a fault occurs or the computer freezes, steps 11, 12 will be immediately skipped.**
       b. **Disconnecting the subject during analysis would destroy the data as electrode placements are not precisely reproducible.**

11) **Subject is again asked to relax, close eyes and move arm at whim.**
12) **Data is recorded for another 10 minutes.**
13) **Skin electrodes are disconnected. Arm brace is removed.**
14) **Excess Gel is cleaned off**
15) **Subject is thanked for participation**

2. How much time will be required of each subject?

Task	**Duration**	**Variability**
Setup	5min	+/- 1min (estimated)
Data Recording I	10min	precisely computer controlled
Data Analysis	74seconds	+/- 2seconds
Data Recording II	10min	precisely computer controlled
Take Down	2min	+/- 30 sec (estimated)
Total	**28 min 14seconds**	**+/- 1min 32seconds**
Factor of safety		**+5minutes**
Grand Total	**33 min 14seconds**	

Beginning to End trial times will be no more than *33 minutes and 14 seconds*. If any trial reaches 30 minutes it will be prematurely terminated so as not to exceed 33 minutes.

D. POTENTIAL RISKS

1. State the potential risks (physical, psychological, financial, social, legal or other) connected with the proposed procedures and explain the steps taken to minimize these risks.

The only significant risks from this experiment are shock hazard from handling electronic equipment and possible allergic reaction from the electrode gel.

Shock Hazard is reduced via redundant safety components within the sensor circuitry. The circuit will fail before dangerous voltages are reached, acting as a fail safe. Overall circuit is less dangerous than listening to headphones. Literature regarding the safety of this device in comparison with other, tested devices is attached. A new technique, completely isolating the subject from electrical ground is implemented so there's no direct connection between the subject and any power source. This feature reduces the likelihood of shock. Literature about this technique is also attached.

Risk of anaphylactic or other allergic reaction is low considering the gels are made for skin use. Any subject who states a history of reaction to topical substances will be rejected from the study.

Should any reaction or any other accident occur, the PI will attend to the subject and activate the 911 system as necessary.

2. Will there be a request for information which subjects might consider to be personal or sensitive (e.g. private behavior, economic status, sexual issues, religious beliefs, or other matters that if made public might impair their self-esteem or reputation or could reasonably place the subjects at risk of criminal or civil liability)?

YES

a. If yes, please describe and explain the steps taken to minimize these risks.

Participants won't be allowed to participate if they have any of the medical conditions described in B.3. These conditions will be read verbally, in private and no written record of the subject's conditions will be kept. Nor will the subjects need to specify which condition, of many, applies. While this may cause some embarrassment for subjects, specific conditions will not be identified, and all responses will be kept confidential.

3. Could any of the study procedures produce stress or anxiety, or be considered offensive, threatening, or degrading? If yes, please describe why they are important and what arrangements have been made for handling an emotional reaction from the subject.
   NO

4. How will data be recorded and stored?
   Biosignal data will be stored in Matlab .mat files. Patient name is not relevant, thus data will be numerically indexed by participant with no personally identifying information attached.
   a. How will identifiers be used in study notes and other materials?
   Only numeric identifiers will be used.
   b. How will reports will be written, in aggregate terms, or will individual responses be described?
   Only numeric identifier will be used.

5. If audio or videotaping is done how will the tapes be stored and how/when will the tapes be destroyed at the conclusion of the study.
   N/A

6. Is there any deception of the human subjects involved in this study? If yes, please describe why it is necessary and describe the debriefing procedures that have been arranged.
   NO

E. POTENTIAL BENEFITS

This does not include any form of compensation for participation.

1. What, if any, direct benefit is to be gained by the subject? If no direct benefit is expected, but indirect benefit may be expected (knowledge may be gained that could help others), please explain.
   There will be no benefit for the subject. This knowledge will be used in future research designed to help those with neurological problems regain mobility.

F. COMPENSATION

1. Explain compensation provisions if the subject withdraws prior to completion of the study.
   No compensation is given for this study
2. If class credit will be given, list the amount and alternative ways to earn the same amount of credit.
   N/A

G COLLABORATORS

1. If you anticipate that additional investigators (other than those named on **Cover Page**) may be involved in this research, list them here indicating their institution, department and phone number.
   N/A
2. Will anyone besides the PI or the research team have access to the data (including completed surveys) from the moment they are collected until they are destroyed
   NO

H. ADDITIONAL INFORMATION

1. If a questionnaire, survey or interview instrument is to be used, attach a copy to this proposal.

2. Attach a copy of the informed consent form to this proposal.

3. Please provide any additional materials that may aid the IRB in making its decision.

North Carolina State University is a land-grant university and a constituent institution of The University of North Carolina

Office of Research
and Graduate Studies

NC STATE UNIVERSITY

Sponsored Programs and
Regulatory Compliance
Campus Box 7514
1 Leazar Hall
Raleigh, NC 27695-7514

919.515.7200
919.515.7721 (fax)

From: Debra A. Paxton, IRB Administrator
North Carolina State University
Institutional Review Board

Date: October 26, 2006

Project Title: Intelligent Control Interfaces for Biosensors

IRB#: 382-06-10

Dear Mr. Adcock;

The project listed above has been reviewed in accordance with expedited review procedures under Addendum 46 FR8392 of 45 CFR 46 and is approved for one year from its date of review. **This protocol expires on October 25, 2007, and will need continuing review before that date.**

NOTE:

1. This board complies with requirements found in Title 45 part 46 of The Code of Federal Regulations. For NCSU the Assurance Number is: M1263; the IRB Number is: 01XM.

2. The IRB must be notified of any changes that are made to this study.

3. Your approval for this study lasts for one year from the review date. If your study extends beyond that time, including data analysis, you must obtain continuing review from the IRB.

Please provide a copy of this letter to your faculty sponsor. Thank you.

Sincerely,

Debra Paxton
NCSU IRB

North Carolina State University
INFORMED CONSENT FORM for RESEARCH

Intelligent Control Interfaces for Biosensors

D. Brook Adcock Jr. **Dr. Edward Grant (PhD)**

We are asking you to participate in a research study. The purpose of this study is to test a Biosignal Computer Interface that will use your body's electrical signals to predict body movements. If this test is successful, future research may allow paralyzed people to move, allow amputees to control bionic limbs, or even allow anyone to control robots with their mind.

INFORMATION

If you agree to participate in this study, you will be asked to be attached to a Noninvasive Biosignal Computer Interface. If you consent, sensors will be temporarily placed on your skin with gel and tape; you will wear an arm brace with a sensor. During the experiment you will be asked to relax and move your braced arm at random for a period of 10 minutes while a computer records information from the sensors. At the end of the experiment, the sensors and brace will be removed.

RISKS

The risks of this experiment are very low. The risks of this experiment are slightly less than that those of listening to music on a computer with earphones. Foreseen risks are listed below.

1. **As when handling any electronic equipment there is chance of being shocked. Multiple redundant safety measures have been taken to electrically isolate you from any potential hazards.**
2. **As when placing any material in contact with the skin, there is a chance of allergic reaction. Should you have a reaction, you will be attended to immediately.**
3. **Some subjects describe the gel used to attach the electrodes as being "slimy and cold". Sometimes the gel leaves behind a residue that can be easily washed off. Some people find this unpleasant. It is unknown what effect this gel has on cloths, hair dye, etc.**

The risks of this study are increased if you have any of the following medical conditions read to you by the investigator.

If none of the medical conditions apply to you, please initial here:_______

I the researcher have read the document "Prohibited Medical Conditions" to the potential subject. Initial here:_________

BENEFITS

There are no direct benefits to the subjects. This knowledge will be used in future research designed to help those with neurological problems regain mobility.

CONFIDENTIALITY

The information in the study will be kept confidential. Data will be stored in Matlab .mat files referenced to you only by a number that is not linked to your identity. No reference will be made in oral or written reports which could link you to the study.

COMPENSATION

No compensation will be given to you for your participation.

EMERGENCY MEDICAL TREATMENT

Should you be injured during this study the experimenter will attend to you immediately. Should you need any type of medical treatment, the 911 system will be activated and you will be transported to the most appropriate Emergency Room Receiving Facility.

CONTACT

If you have questions at any time about the study or the procedures, you may contact the researcher, D. Brooks Adcock Jr., at 2404 Still Forest Pl, Apt E, Raleigh NC, 27607, or (919)-606-7003. If you feel you have not been treated according to the descriptions in this form, or your rights as a participant in research have been violated during the course of this project, you may contact Mr. Matthew Ronning, Assistant Vice Chancellor, Research Administration, Box 7514, NCSU Campus (919/513-2148)

PARTICIPATION

Your participation in this study is voluntary; you may decline to participate without penalty. If you decide to participate, you may withdraw from the study at any time without penalty and without loss of benefits to which you are otherwise entitled. If you withdraw from the study before data collection is completed your data will be returned to you or destroyed at your request.

CONSENT

"I have read and understand the above information. I have received a copy of this form. I agree to participate in this study with the understanding that I may withdraw at any time."

Subject's signature____ Date _________________

Investigator's signature___________________________________ Date _________________