CAL POLY
SAN LUIS OBISPO

EEG Classification with Discrete Wavelet Transforms and Energy Distribution

A Senior Project presented to the faculty of the

Electrical Engineering Department

California Polytechnic State University, San Luis Obispo

In partial fulfillment of the requirements for the Degree of Bachelors in Science

2013 - 2014

Miguel Contreras and Javier I. Suárez

Table Of Contents

List of Tables & Figures

Acknowledgements

Abstract

The overall aim of this investigation is to allow paralyzed individuals to regain motor movement with thought controlled robotic devices and in doing so provide them with a higher level of independence. The specific goal of this project is to classify EEG movement signals through a combination of discrete wavelet transforms and energy distribution. Using the energy distribution of these signals a neural network can be implemented to more accurately differentiate distinct motor movements.

Introduction

A simple movement of the arm or gesture with a hand can cause the brain to fire hundreds and sometimes thousands of neural signals. The larger goal of this project is to help characterize those signals and compile a library that can be used to recreate motor movements and gestures with an artificial limb. However, this investigation will focus on classifying EEG motor signals more accurately through the use of the discrete wavelet transform, energy distribution of the signal for a specific motor movement, and finally a neural network to determine the movement. In doing so one goal will be to achieve a greater accuracy than previous and existing methods of EEG motor signal characterization. Better classification and characterization of signals will ultimately lead to more accurate representation of the movements. The necessity to better previous methods of classification is to ultimately allow for paraplegics and disabled individuals to regain some motor skills. Established signals from databases from accepted research facilities will be used in order to establish a standard for which our methods can be directly correlated to existing methodologies. Doing this will provide a base and an approach by which one will be able to improve and better these methods in terms of more accurate signal characterization.

Electroencephalography (EEG) is the recording of electrical activity along the scalp. EEG measures voltage fluctuations resulting from ionic current flows within the neurons of the brain. Why make a library? We found that EEG signals' primary use is to determine brain abnormalities, so a normal EEG brain signal will be compared to the patients to determine if anything is wrong. The main diagnostic application of EEG is in the case of epilepsy, as epileptic activity can create clear abnormalities on a standard EEG study. A secondary clinical use of an EEG is in the diagnosis of coma, encephalopathy, and brain death. The third common use of EEG is for studies of sleep and sleep disorders where recordings are typically done for one full night. However, José L. Contreras-Vidal, Alessandro Presacco, Harshavardhan Agashe, and Andrew Paek authors of *Restoration of Whole Body Movement* [2] have been one of the many journals to inspire the pursue of a different use for an EEG signal.

EEG externally measures electrical activity generated by large neural networks in the brain, and research in their laboratory [2] was the first to demonstrate the feasibility of inferring voluntary natural movement from EEG signals, essentially decoding human brain activity used for physical movement.

While similar but invasive neural interface technology under development allows users to think commands that are sent to sophisticated upper- or lower-limb prosthetics or used to control computer cursors, they recently reported the first EEG-based neural interface (needing only a single training session before subjects can operate it) that employs continuous decoding of imagined, continuous hand movements. A noninvasive EEG-based neural interface is easier to repair or replace, if needed, and the technology is very user friendly requiring only a fabric cap and the slight inconvenience of some goo on a person's head where the sensors are attached. Though EEG monitoring is safer than other approaches, many in the scientific community had deemed it unreliable for a brain-computer

Figure 1: Demonstration of a Robotic Exoskeleton [2]

interface, mainly because they believed that the human skull blocks much of the detailed brain activity needed for precision controlled prosthetics.

So by establishing a standard EEG signal library for upper limb movements, we can be one step closer to a safe, reliable, and noninvasive BMI (Brain Machine Interface) to robotic systems that can bring life-changing technology to millions of people who have difficulty generating uninhibited movement.

Background/Literature Review

Networks

"Divide and conquer" is the basis to any network system; basically any complex system can be decomposed into subsections or simpler elements in order to understand the functionality of the system. There are a large number of different types of networks, but they all are characterized by the following components: a set of nodes, and connections between nodes. Networks are used to model a wide range of phenomena in physics, computer science, biochemistry, ethology, mathematics, sociology, economics, telecommunications, and many other areas.

Artificial Neural Networks (ANN)

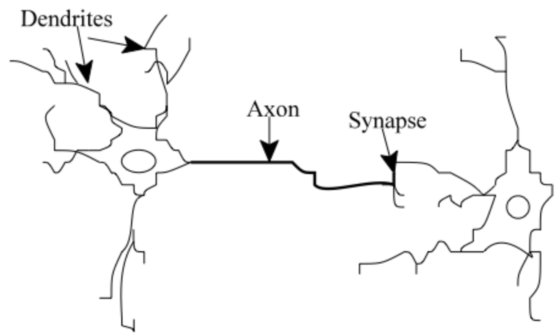


Figure 2: Neural Network

An artificial neuron is a computational model inspired in the natural neurons. Natural neurons receive signals through synapses located on the dendrites or membrane of the neuron. When the signals received are strong enough (surpass a certain threshold), the neuron is activated and emits a signal though the axon. This signal might be sent to another synapse, and might activate other neurons. Basically from Figure 2, ANN consists of inputs (like synapses), which are multiplied by weights (strength of the respective signals), and then computed by a mathematical function, which determines the activation of the neuron. Another function computes the output of the artificial neuron (sometimes in dependence of a certain threshold). The higher a weight of an artificial neuron is, the stronger the input that is multiplied by it will be. Weights can also be negative, so we can say that the signal is inhibited by the negative weight. Given so many inputs, how does one determine the weight of each one? There are algorithms that can adjust the weights of the ANN in order to obtain the desired output from the network. We will be using the back propagation algorithm to determine the appropriate weights for the inputs.

Back Propagation Algorithm

Back propagation algorithm is a common method of training ANN. From a desired output, the network learns from many inputs. In short, it's a supervised learning method, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error is calculated. The idea of the back propagation algorithm is to reduce this error, until the ANN learns the training data. Back propagation algorithm requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks (networks with no feedback). The following is a breakdown of the back propagation algorithm:

Phase 1: Propagation

Each propagation involves the following:

I. Forward propagation of a training patterns input through the neural network in order to generate the propagations output activations
II. Backward propagation of the propagations output activations through the neural network using the training pattern target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight Update

For each weight-synapse (input) follow the following:

I. Multiply its output delta and input activation to get the gradient of the weight
II. Subtract a ratio (percentage) of the gradient from the weight

This ratio influences the speed and quality of learning; it's called the learning rate. The greater the ratio, the faster the neuron trains. The lower the ratio, the more accurate the training is. The sign of the gradient of a weight indicates where the error is increasing; this is why the weight must be updated in the opposite direction. Lastly, phase 1 and 2 must be repeated until the performance of the ANN is satisfactory. For more information on the formulas used throughout the back propagation algorithm refer to *Artificial Neural Networks for Beginners* by Carlos Gershenson [#].

Discrete Wavelet Transformation

Wavelet analysis is to decompose signals into several frequency bands. Selection of appropriate wavelet and the number of decomposition levels are very important for the analysis of signals using Discrete Wavelet Transform. The levels are chosen such that those parts of the signal that correlates well with the frequencies necessary for classification of the signal are retained in the wavelet coefficients.

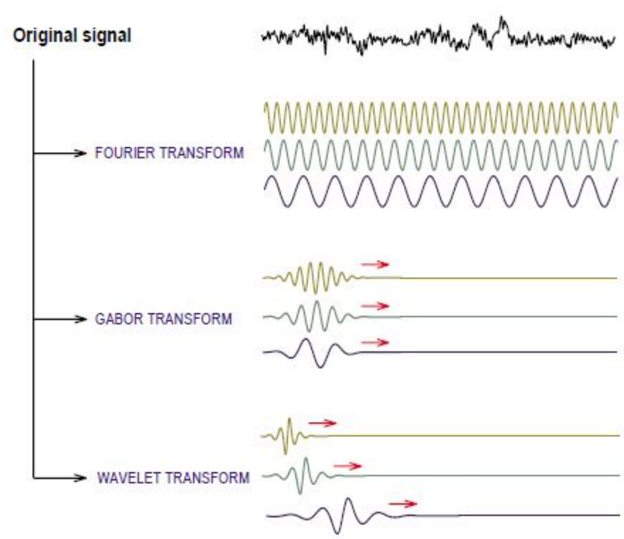


Figure 3: Frequency and Time-Frequency Methods

One of the major advantages to using the DWT over a number of other types of transforms such as the Fourier transform is that one is able to achieve temporal resolution. In essence the DWT is able to capture both the frequency and location of the signal in question.

The Goal

The final goal is to design Neural Network to classify upper limb movement. The bottom image displays the outline for the process of classification of EEG signals used:

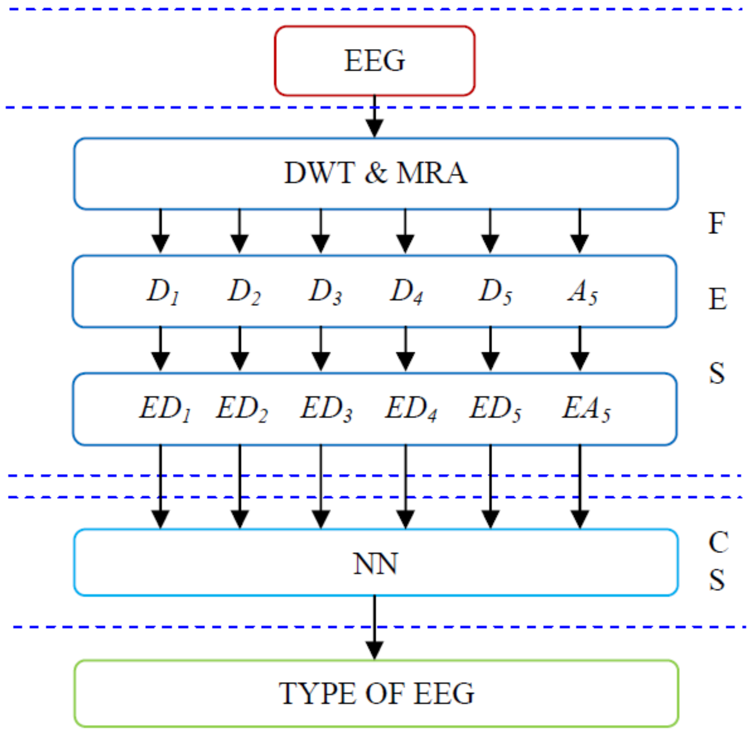


Figure 4: Block Diagram of Classification of EEG Signals [11]

An algorithm block diagram for classification of EEG signals is presented on Figure 4. The algorithm structure is based on two stages: feature extraction stage (FES) and classification stage (CS). The input of the CS is a preprocessed signal using DWT. In this case, EEG signal in the time domain is transformed into the wavelet domain before applying, as input to the CS. Feature extraction is the key for pattern recognition.

Test Plan

Why Use MATLAB?

MATLAB is already is in use in many institutions. It is used in research in academia and industry. Prototype solutions are usually obtained faster in MATLAB than solving a problem from a programming language.

MATLAB is fast, because the core routines in MATLAB are fine tuned for different computer architectures. A study was done by to compare the speed between MATLAB and a program written in C. Since the back propagation algorithm involves matrix manipulations the test chosen was matrix multiplication.

Accuracy

In order to improve the accuracy of how the neural network will differentiate a specific motor movement from another, a sample of patients was used to establish a threshold energy per movement

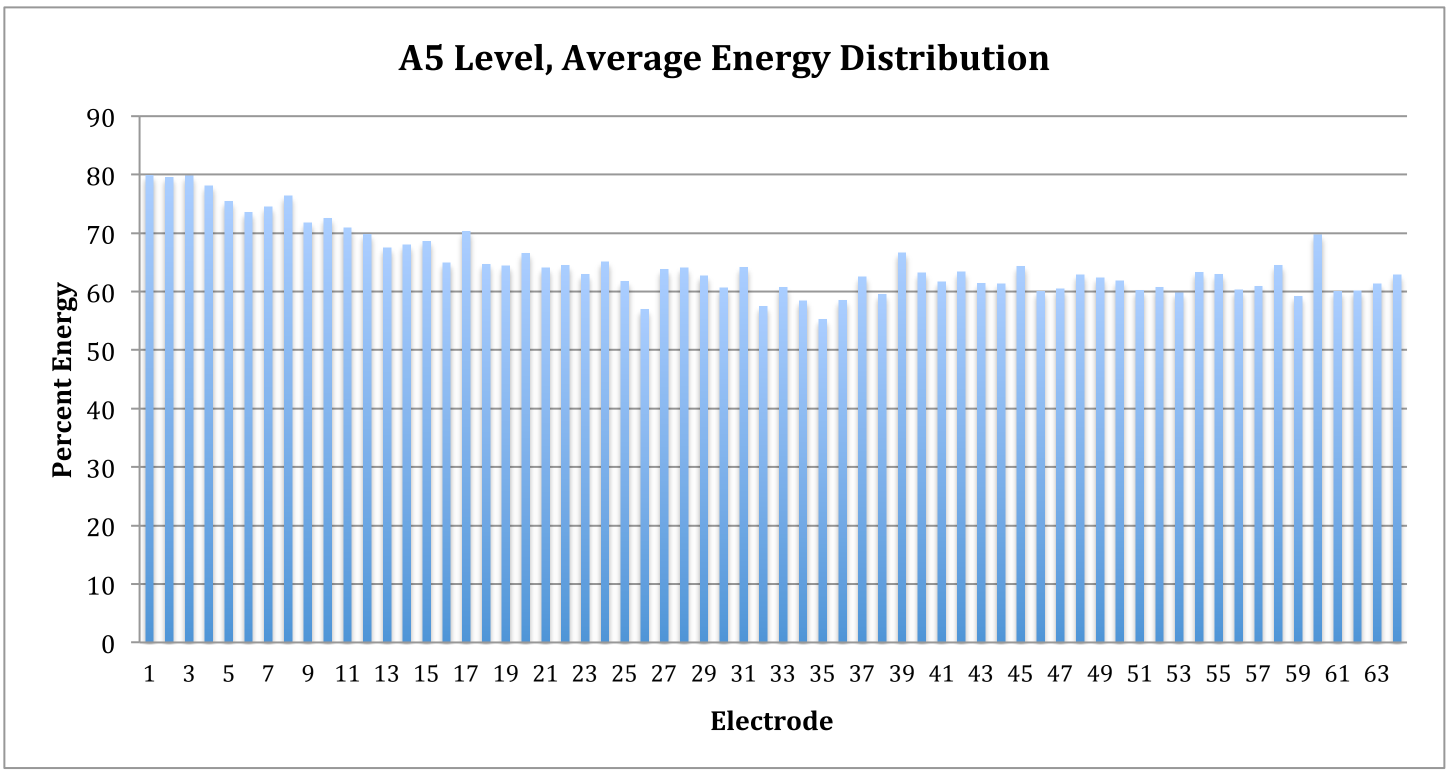


Figure 5: Average Energy Distributions for Level A5 for ten random patients. The movement done was the patient imagining that they were making a fist with either the right or left fist.

Figure 5 illustrates an example of level A4 average energies for 10 people (3 runs per person). From the image we can observe what the average energy per electrode is starting from Fp1 through Iz for a total of 64 electrodes (Refer to EEEG Electrode Map).

Overall, the averages seem to fluctuate dramatically with each run and even more with a different test subject. Thus, in order to insure a more accurate FNN, we must expand the number of subjects we will get the threshold energy level in order to classify the signal. The total number of subjects from the PhysioNet database is 109, however the graph above only shows 10 subjects that did 3 runs each for a total of 30 unique energies per electrode. This number will be expanded as the project proceeds.

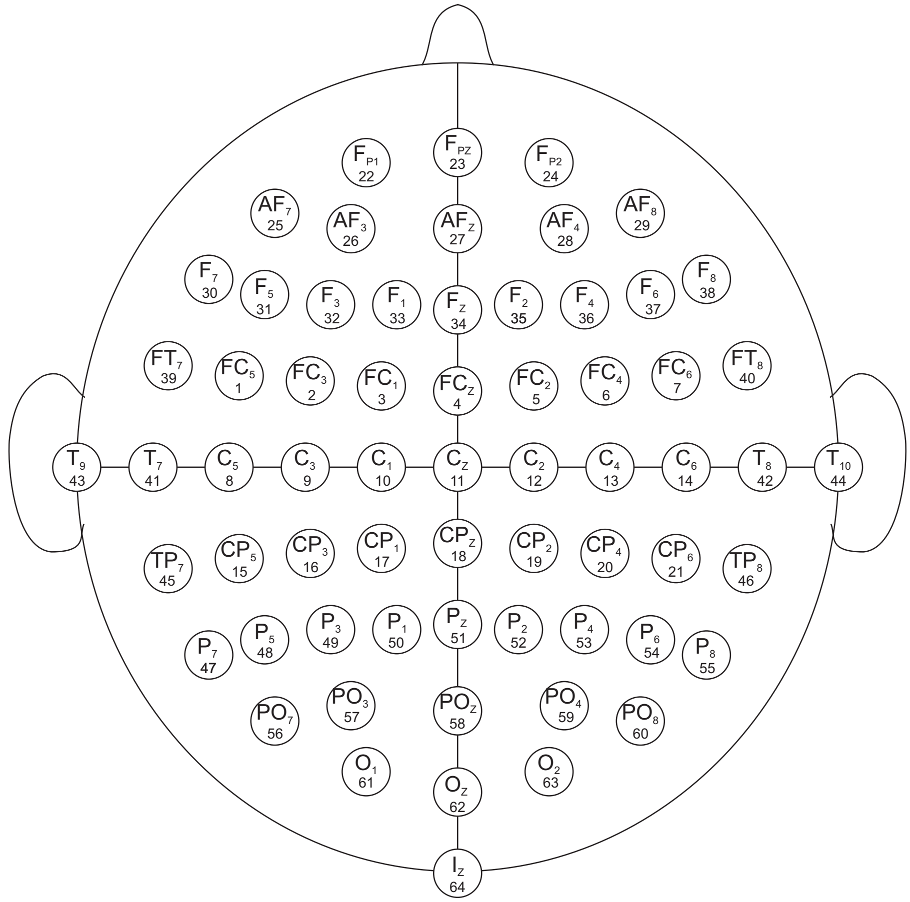


Figure 6: EEG Electrode Map; 64 standard electrodes.

The next step for a further accurate network, we must also classify the decomposition levels. These would be a series of four further levels with 64 average energy distributions. With this we can gather a total of 320 averages that must be met in order to properly consider the signal to be a specific movement. This step will be done for all the remaining levels, which include D4, D3, D2, and D1. With all this information this would give us a total of 1,635 threshold voltages to average in order to correctly classify our 320 averages that will establish the classification of the EEG signal that will be inputted into the program.

Functional Decomposition

Level 0

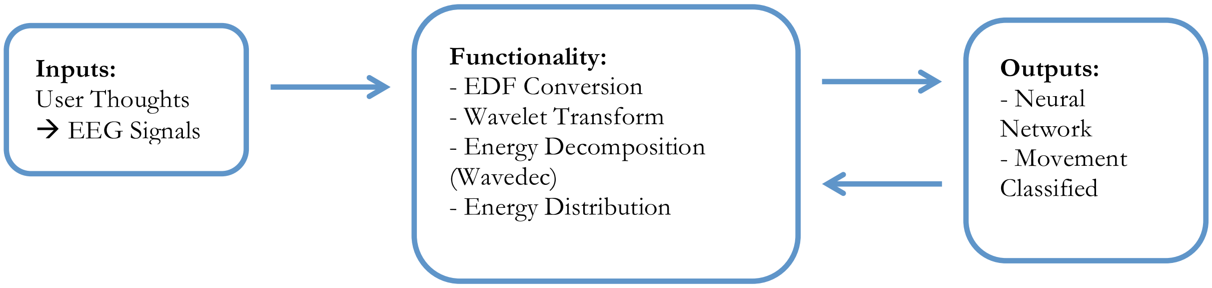


FIGURE 7. Upper Limb Restoration; EEG Classification Library - LEVEL ZERO BLOCK DIAGRAM

TABLE 1
Upper Limb Restoration; EEG Classification Library – FUNCTIONAL REQUIREMENTS FOR LEVEL 0

EEG Classification with Discrete Wavelet Transforms and Energy Distribution	
Input	User thoughts → EEG Signals: A key component to this tool is the easy interface between the customer and device, thus the only required input would be the EEG signals from motor movement activity in the brain.
Output	The output of would be a Neural Network which will be self trained to accepts multiple energy levels depending on the movement.
Functionality	Within the core of the functionality module will a European Data Format (EDF) conversion function found from the MATLAB website. The following will consist of Discrete Wavelet Transforms (DWT) and Energy Decomposition Function (Wavedec). This will give us the Energy Distribution of that specific movement. From the energy distribution the NN will be completed.

The input to will be the user EEG signals that we plan on comparing to the classified EEG library that will be developed. The purpose to developing and comparing the input EEG signals is to not mistake the input EEG signal command for another with similar characteristics. This will further allow a more accurate developed solution. In the main block (functionality) the MATLAB code will then process the incoming EEG signals and further send commands to the NN.

Level 1

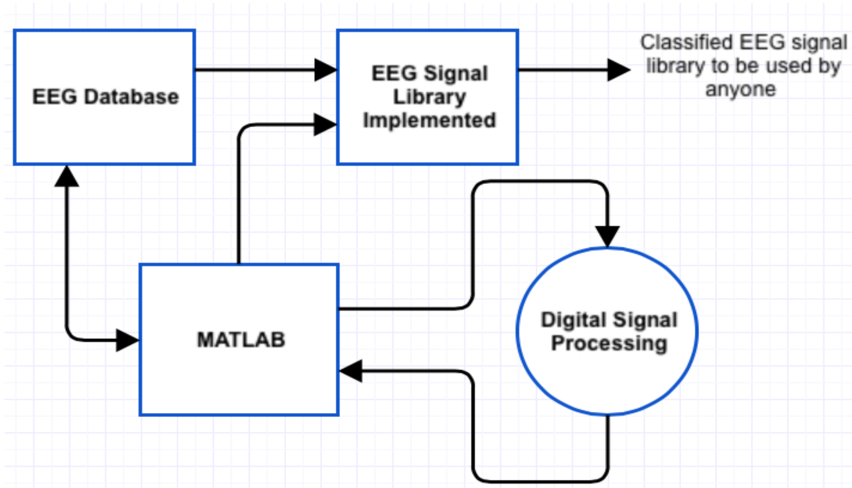


FIGURE 8. Upper Limb Restoration; EEG Classification Library - LEVEL ONE BLOCK DIAGRAM (FOR DSP)

TABLE 2
Upper Limb Restoration; EEG Classification Library – FUNCTIONAL REQUIREMENTS FOR LEVEL 1

EEG Classification with Discrete Wavelet Transforms and Energy Distribution	
Input	The input to our level one, which is strictly the digital signal processing (DSP) of the EEG signals, is external EEG signals gathered from a database. The reason we have chosen to use a database instead of signals gathered from an EEG machine, is simply to avoid error possibilities that can arise when using our own EEG machine. Using field standard EEG signals will further increase our accuracy and the possibility of creating a system to filter and amplify the EEG signals.
Signal Manipulation	After the signals have been uploaded to MATLAB via the EEG database, certain signal manipulations will be implemented in order to look at specific characteristics for the upper limb movements. The system that will be used to manipulate the EEG signals is yet to be determined, but from current research, two approaches are now a possibility that we can use and perhaps further perfect the specific system.
Output	The output to this system would be the key take away from the beginning stages of our project. Once we have set up characteristics for the upper limb EEG signals, we will set up a library with the implemented EEG signals. This library will be a database that anyone can use if they need the signals. From here the next stage of our project will be to interface with a microcontroller.

This portion of the project will have most of our work time due to the importance of classifying upper-limb EEG signals. The first step to this diagram is gathering the already established EEG signals from a database. The secondary block is the MATLAB block in which we will use the back propagation algorithm approach to find the specific characteristics for the upper-limb movement EEG signals. DSP can be considered a sub-block of the MATLAB block due to the signal processing work done by the software. After an EEG signal has been classified and characterized, it will then be placed into a library that will be readily available to anyone wanting to use the signal as a reference. This block diagram is also a sub-block of the level zero diagram. The upper-limb classified EEG signal block will be after the user input EEG signals.

Development and Construction

Experimental Protocol for Data Set

The dataset used was attained from an online resource, PhysioNet [#]. The data consist of one to two minute EEG recordings, obtained from 109 volunteers. EEG recordings where gathered using a 64-channel EEG recorder (BCI200) system. Each subject performed 14 experimental runs. In summary, the experimental runs where as follows:

1. Baseline, eyes open
2. Baseline, eyes closed
3. Task 1: open and close left or right fist
4. Task 2: imagine opening and closing left and right fist
5. Task 3: open and close both fists and feet
6. Task 4: imagine opening and closing both fists and feet
7. Task 1: open and close left or right fist
8. Task 2: imagine opening and closing left and right fist
9. Task 3: open and close both fists and feet
10. Task 4: imagine opening and closing both fists and feet
11. Task 1: open and close left or right fist
12. Task 2: imagine opening and closing left and right fist
13. Task 3: open and close both fists and feet
14. Task 4: imagine opening and closing both fists and feet

The data provided by PhysioNet is in EDF+ format (European Data Format). In order to properly read the EEG test results we had to use an edfRead script provided by the MathWorks website.

Interpreting European Data Format (EDF)

The data provided by PhysioNet database is in EDF+ format (European Data Format). Running the script that one obtained from the Mathworks website the EEG signals were broken down by electrode. The EEG signal readings found in figure 5 are from electrodes placed on the primary motor cortex of the brain of an individual who imagined they were making a fist. The magnitude of the signals ranges from approximately 176uV to 245uV. This can be attributed to the placement of the probes, and the origin of the signal with respect to the brain.

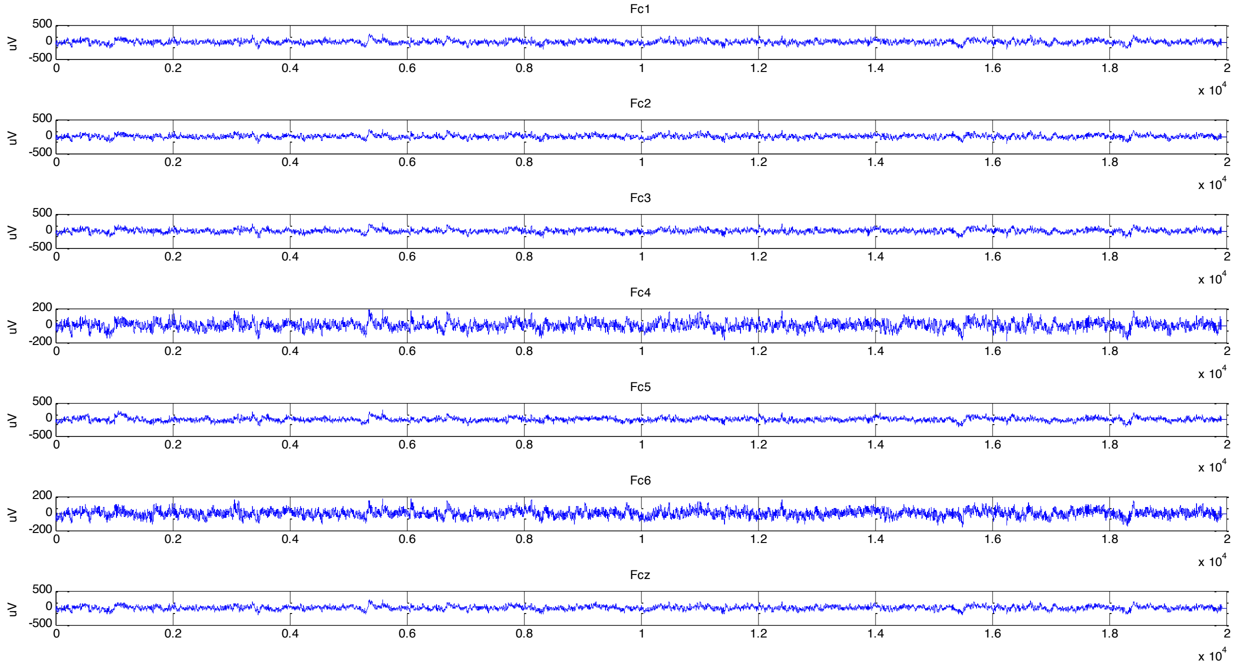


Figure 9: EEG signals taken from subject thinking about making a fist with either the right or left hand. Measurements taken from the electrodes wired to the primary motor cortex(electrodes: Fc1, Fc2, Fc3, Fc4, Fc5, Fc6, Fcz).

The EEG signal readings found in figure 6 are from electrodes also placed on the primary motor cortex of the brain of an individual physically made a fist with their hand. The magnitude of the signals ranges from approximately 278uV to 314uV. This can be attributed to the placement of the probes, and the origin of the signal with respect to the brain. It is interesting to note that the magnitudes of these signals increases as a result of physically performing an action as opposed to merely imagining it.

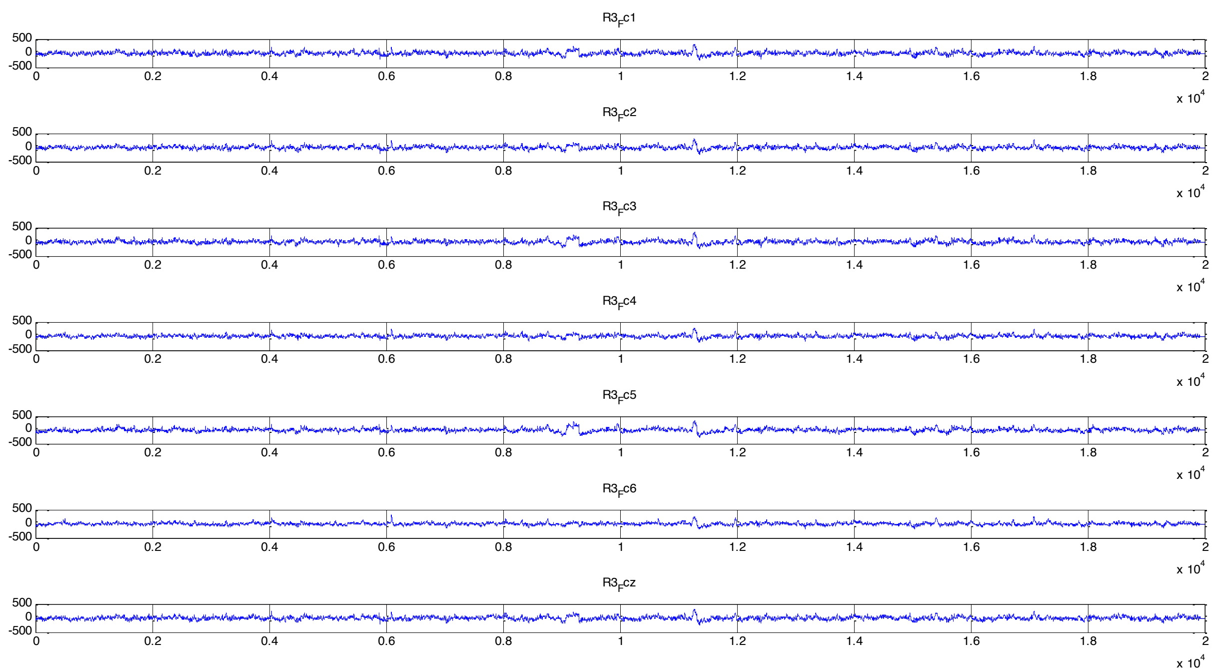


Figure 10: EEG signals taken from subject physically making a fist with either the right or left hand. Measurements taken from the electrodes wired to the primary motor cortex (electrodes: Fc1, Fc2, Fc3, Fc4, Fc5, Fc6, Fcz).

Decomposition of the signals via Discrete Wavelet Transform: Daubechies DB4

Using the Wavelet Toolbox found in Matlab, a DWT was applied to one signal. More specifically the signal that was read from an electrode placed closest to the region of origin of that signal. The electrode placed in the right hemisphere of the brain on the ridge of the primary motor cortex is where motion from hand is most easily detected. Applying the DWT to the signal and using the DB4 for its smoothing characteristics that is ideal for detecting EEG signals, the original signal was decomposed and demonstrated in figure 7.

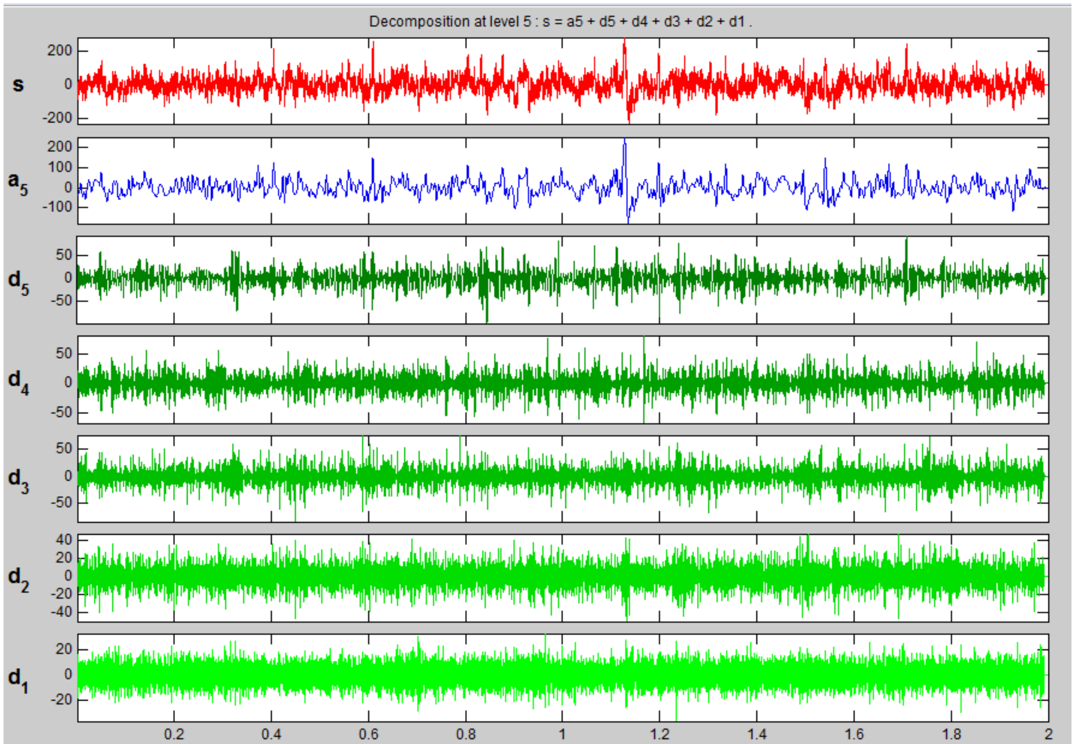


Figure 11: DWT using Db4 wavelet to decompose a signal of an individual making a fist (Electrode: Fcz).

Automated Implementation of EEG classification

Given that the wavelet toolbox would not allow for the best/fastest method of classifying our EEG signal, we have decided to fully automate our process with in the following format. Here is a brief overview of the code:

1. The code will interpret the European Data Format given a function:
    a. [header, data] = edfread (In here goes EEG signal to be classified)

2. The program will assign each of the 64 electrodes its proper dataset from the EDF file.

3. Program goes into a loop with a count of 64 (for each electrode)
    a. Each electrode goes through the Wave Decomposition function
        i. [Coef,Length]=wavedec(electrode(x).num, 4, 'db4')
            1. electrode(x).num → calls upon the specific electrode in a structure
            2. 4 → going down "4 levels"
                a. The "4 levels" was considered after experimenting with a Maximum Wavelet Decomposition (will touch more on it later)
            3. 'db4' → daubechies DB4

b. After coefficients are exported (refer to Wavedec Function section), program assigns coefficients to its proper approximation level and decomposition levels.
    i. Given that its "4 levels," the following levels are generated:
        1. A4
        2. D4
        3. D3
        4. D2
        5. D1

c. Program then calculates energy for each level by doing the following math:
    i. $\mathrm{Coef}(1)^2 + \mathrm{Coef}(2)^2 \ldots + \mathrm{Coef}(\mathrm{n})^2 = \text{Energy of Level}$

d. Program sums up the total energies for each level to gather Total Energy of signal.
    i. $\mathrm{Energy}(\mathrm{A4}) + \mathrm{Energy}(\mathrm{D4}) \ldots + \mathrm{Energy}(\mathrm{D1}) = \text{Total Energy}$

e. Code then calculates the percent that each energy level attributes to the total energy
    i. $\frac{\text{Energy (A4)}}{\text{Total Energy}} \times 100 = \mathrm{PercentEnergy}(\mathrm{A4})$
    ii. Same calculation for level D4, D3, D2, D1.

f. After percent energy is calculated for each level, they are each stored in an array which will contain the percent energy for each level, for example:
    i. a4LevelTotals(count) = PEcA4
        1. This array will contain the PercentEnergy(A4) for all the 64 electrodes in an established order so we know the percent energy for any specific electrode.
    ii. d4LevelTotals(count) = PEcD4
        1. Contains energy for Decomposition Level 4
    iii. d3LevelTotals(count) = PEcD3
        1. Contains energy for Decomposition Level 3
    iv. d2LevelTotals(count) = PEcD2
        1. Contains energy for Decomposition Level 2
    v. d1LevelTotals(count) = PEcD1
        1. Contains energy for Decomposition Level 1

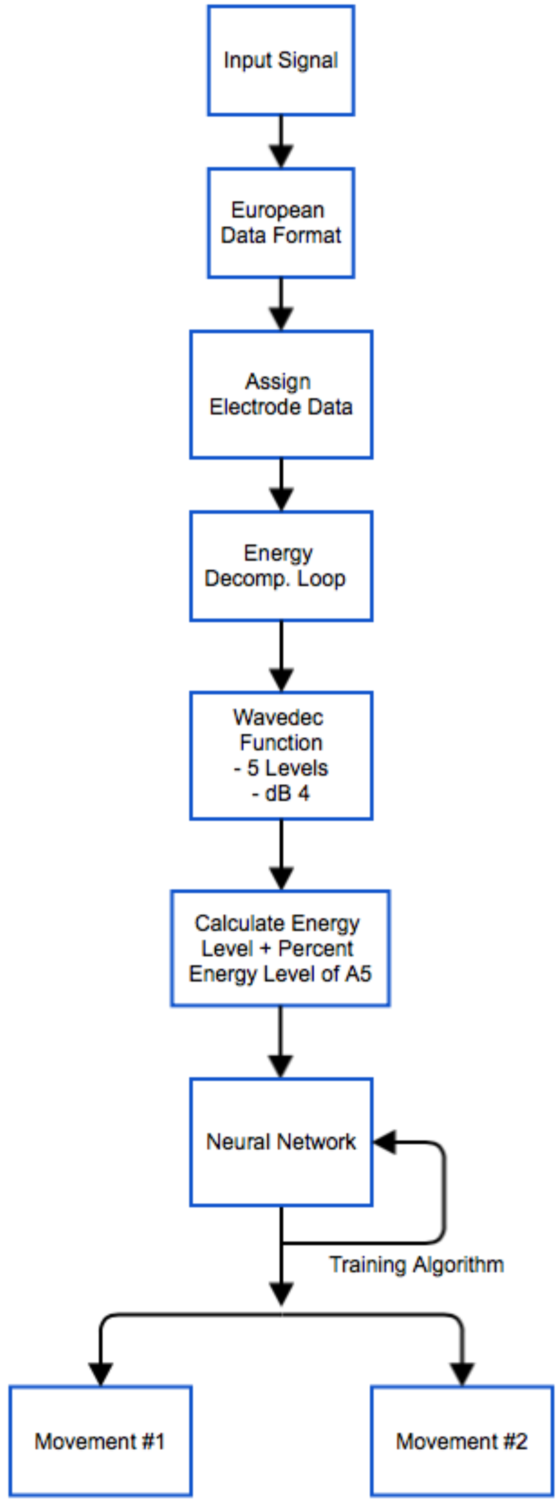


Figure 12: Program Flow Diagram

Functions

- wavedec

The wavedec function performs a multilevel one-dimensional wavelet analysis using either a specific wavelet (*'wname'*). [C,L] = wavedec(X,N,*'wname'*) returns the wavelet decomposition of the signal X at level N, using *'wname'*. N must be a strictly positive integer. The output decomposition structure contains the wavelet decomposition vector C and the book keeping vector L.

The structure is organized as in this level-3 decomposition example.

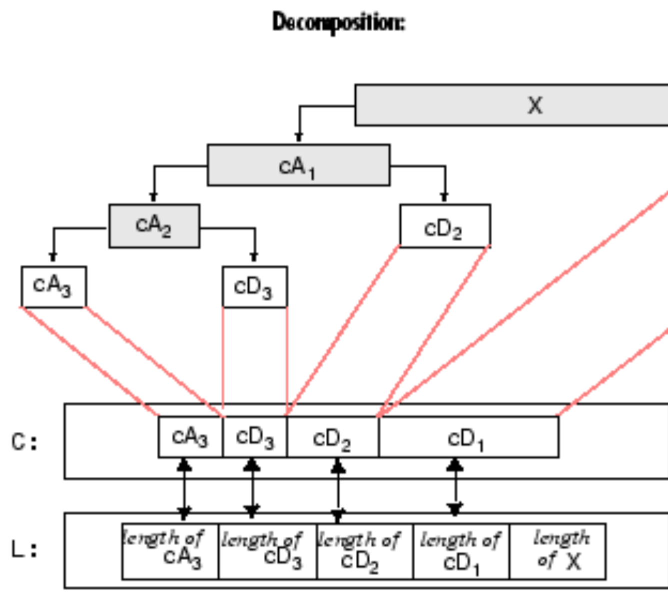


Figure 13: wavedec Level-3 Decomposition Example

Algorithm:

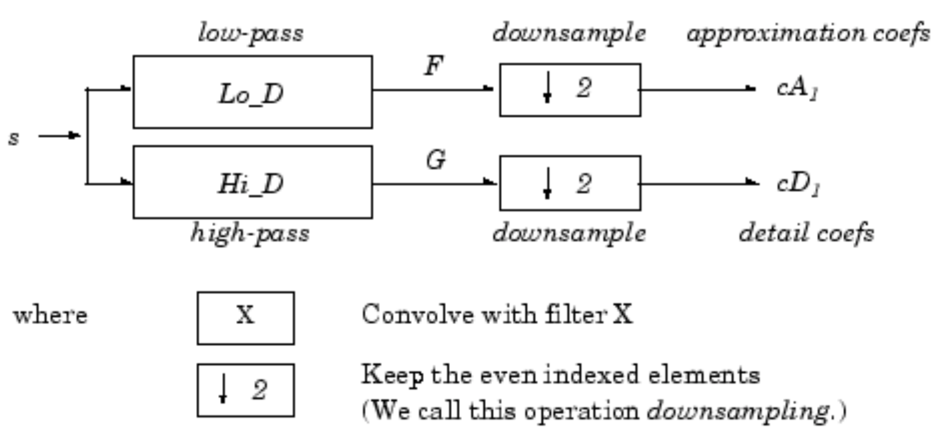


Figure 14: wavedec First Step (Low-pass Filter, Downsample, Coefficients)

Energy Distribution of Signals: A5 decomposition Level

Table 3: Energy distribution Level A4 decomposition with 9 electrodes.

Level A4 Decomposition - 9 Electrodes (Motor Cortex Ft7-Ft8)				
Patient	% Energy - Movement 1	% Energy - Movement 2	% Difference in Energy between movements	% Average Difference
1	75.8766	80.5588	4.6822	3.43288
2	48.8825	56.6646	7.7821	
3	77.2263	87.5926	10.3663	
4	89.6263	88.6606	.9657	
5	35.9608	34.7745	1.1863	
6	69.1438	73.0392	3.8954	
7	23.2852	32.2493	8.9641	
8	81.5847	82.2730	.6883	
9	58.3894	53.0947	5.2947	
10	94.3427	95.7352	1.3925	
11	86.5316	87.1087	.57713	
12	85.8393	88.6520	2.8126	
13	51.8002	58.5064	6.7062	
14	41.5892	42.8446	1.2553	
15	46.4858	48.9776	2.4917	
16	83.0943	79.2505	3.8438	
17	63.9006	62.9451	.95548	
18	87.9317	86.5966	1.3350	
19	57.8718	65.8113	7.9395	
20	87.3019	89.1077	1.8058	

The above table demonstrates that decomposing a signal at A4 will yield an average of approximately 3.43% in energy difference between two different movements performed by the same individual. Patient 4 and 8 show a subtle difference in energy between both movements, at approximately less than a percent. Patients 5 and 10 also demonstrate a slightly larger difference than a percent in energy located at the level 4 decomposition, between the movements.

In order to increase the overall average energy difference in movements among all the patients two distinct approaches were taken into account: increasing the electrode count and decomposing the EEG signal further. The table below is similar to that of the data recorded in the table above with respect to level A4 decomposition but with the exception that it took into account all 64 of the standard EEG electrodes(see figure).

Table 4: Energy distribution Level A4 decomposition with 64 electrodes.

Level A4 Decomposition - 64 Electrodes				
Patient	% Energy - Movement 1	% Energy - Movement 2	% Difference in Energy between movements	% Average Difference
1	75.7620	79.0408	3.2788	3.007825
2	49.0944	54.4899	5.3955	
3	76.5248	83.2141	6.6893	
4	90.2131	89.0814	1.1317	
5	36.3416	36.4388	.0972	
6	64.6201	71.0707	6.4506	
7	28.4932	34.8048	6.3116	
8	84.2791	84.9705	.6914	
9	67.8716	61.0518	6.8198	
10	92.3272	93.4886	1.1614	
11	86.1876	87.0317	.8441	
12	85.8303	88.4216	2.5913	
13	55.2560	59.5821	4.3261	
14	45.5479	47.2256	1.6777	
15	47.6417	49.7969	2.1552	
16	81.5702	80.4170	1.1532	
17	68.8141	68.1921	.622	
18	86.0311	85.4799	.5512	
19	53.8167	58.8668	5.0501	
20	86.2993	89.4576	3.1583	

Using the 64 standard electrodes and decomposing the energy in each of those electrodes to level A4 does increase the average energy difference in the two distinct physical movements. In fact the average difference drops from 3.43% to 3.00% yielding an overall decrease of 0.43%, and consequently increase the difficulty that the neural network would have in distinguishing the two physical movements.

As previously mentioned, another approach that could possibly increase the overall average energy difference between movements would be to further decompose the signal to a higher level. The Matlab function, *wmaxlev*, actually returns one less than the maximum expected level of decomposition for a signal. The function had originally returned 4 as the maximum level of decomposition, but given the information of the Matlab function, decomposing the signals to level A5 can be observed in the table below.

Table 5:Energy distribution Level A5 decomposition with 64 electrodes.

Level A5 Decomposition - 64 Electrodes				
Patient	% Energy - Movement 1	% Energy - Movement 2	% Difference in Energy between movements	% Average Difference
1	63.5524	68.6011	5.0487	3.29983
2	38.9468	46.2054	7.2586	
3	64.5063	70.4318	5.9255	
4	81.2137	81.8807	.667	
5	30.6663	31.2195	.5532	
6	56.8446	60.8651	4.0205	
7	22.6453	28.1720	5.5267	
8	76.0442	77.8036	1.7594	
9	64.6161	58.0287	6.5874	
10	88.8388	90.2861	1.4473	
11	79.4242	81.6918	2.2676	
12	71.8652	74.3515	2.4863	
13	42.1059	47.1228	5.0169	
14	36.2146	37.4564	1.2418	
15	35.6724	39.6670	3.9946	
16	73.2898	72.7145	.5753	
17	65.1848	63.5361	1.6487	
18	79.0195	78.1454	.9641	
19	41.8677	45.0784	3.2107	
20	78.7067	84.5030	5.7963	

Table 1 and table 3 show a minimal difference in average energy differences between movements. Although decomposing the signal to level A4 and using only the 9 electrodes located in the motor cortex yields a higher average in energy between the movements, these results can be deceiving in the sense that one is trying to characterize the energy in a signal with only 9 electrodes. Using 9 electrodes yields a very sharp difference from patient to patient. So in order to minimize the large jumps in energy one opted to decompose the signal further to level A5 and employ all 64 electrodes in order to more accurately characterize each movement for each patient. Furthermore, using more electrodes in essence acts a low pass filter in order to filter out any noise that may compromise the signal.

Establishing the Neural Network

The next steps consist of establishing the Neural Network (NN); the following are the results of our attempts to create a Neural Network to handle the task that we desire. This is a high-level block diagram view of our logic behind the NN:

Output Black Box:

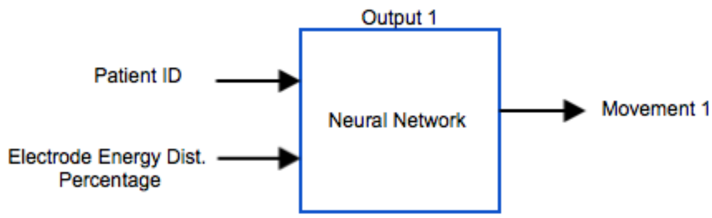


Figure 15: Block Diagram of NN

We decided to test 30 patients with three sets of 10 patients, the patients were selected randomly out of 109 patients in order to test the validity of the NN, the following are three tables and graphs with our input data to the NN. Where the inputs are a patient ID which was employed as a means to aid the neural networks ability to better distinguish movements. The patient ID was chosen to be a fraction of unity in order to avoid saturating the neural network.

Table 6: Data Set 1

Input	Output
0.1, 0.635	1
0.1, 0.686	-1
0.2, 0.389	1
0.2, 0.462	-1
0.3, 0.645	1
0.3, 0.704	-1
0.4, 0.812	1
0.4, 0.818	-1
0.5, 0.306	1
0.5, 0.312	-1
0.6, 0.568	1
0.6, 0.608	-1
0.7, 0.226	1
0.7, 0.281	-1
0.8, 0.760	1
0.8, 0.778	-1
0.9, 0.646	1
0.9, 0.580	-1
1, 0.888	1
1, 0.902	-1

Note: Input = (Patient ID, 64-Electrode A5 Percent Energy Average), Output = Designated Number for Movement

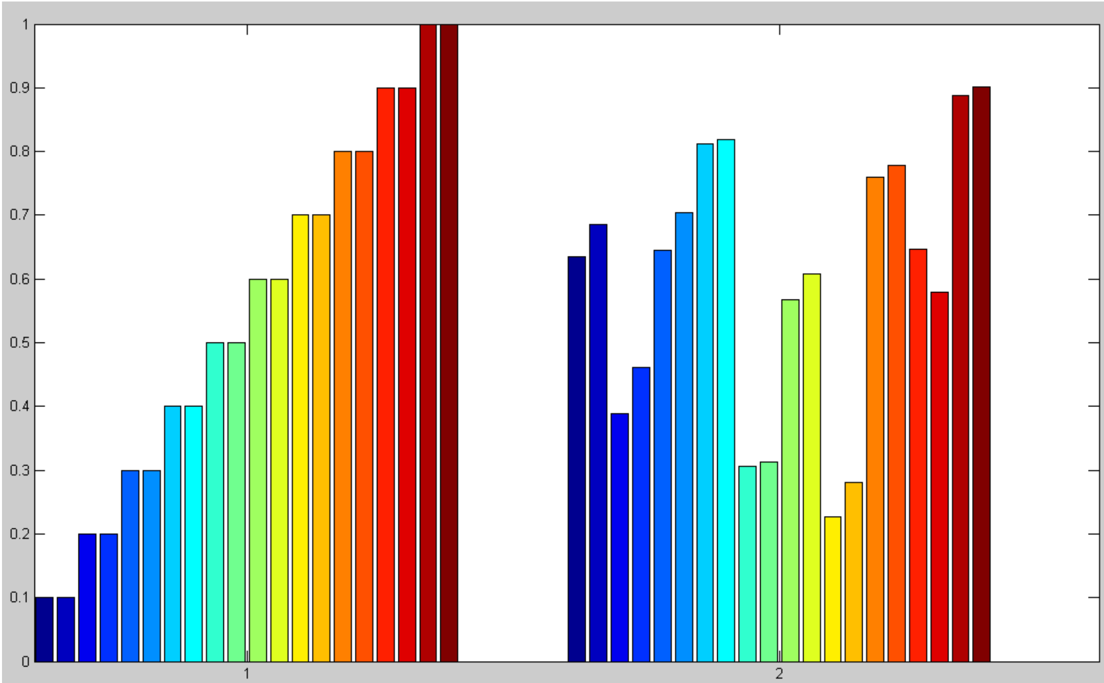


Figure 16: Energy Distribution of Set.1

Table 7: Data Set 2

Input	Output
0.1, 0.794	1
0.1, 0.817	-1
0.2, 0.719	1
0.2, 0.744	-1
0.3, 0.421	1
0.3, 0.471	-1
0.4, 0.362	1
0.4, 0.375	-1
0.5, 0.357	1
0.5, 0.397	-1
0.6, 0.733	1
0.6, 0.727	-1
0.7, 0.652	1
0.7, 0.635	-1
0.8, 0.790	1
0.8, 0.7814	-1
0.9, 0.419	1
0.9, 0.451	-1
1, 0.794	1
1, 0.817	-1

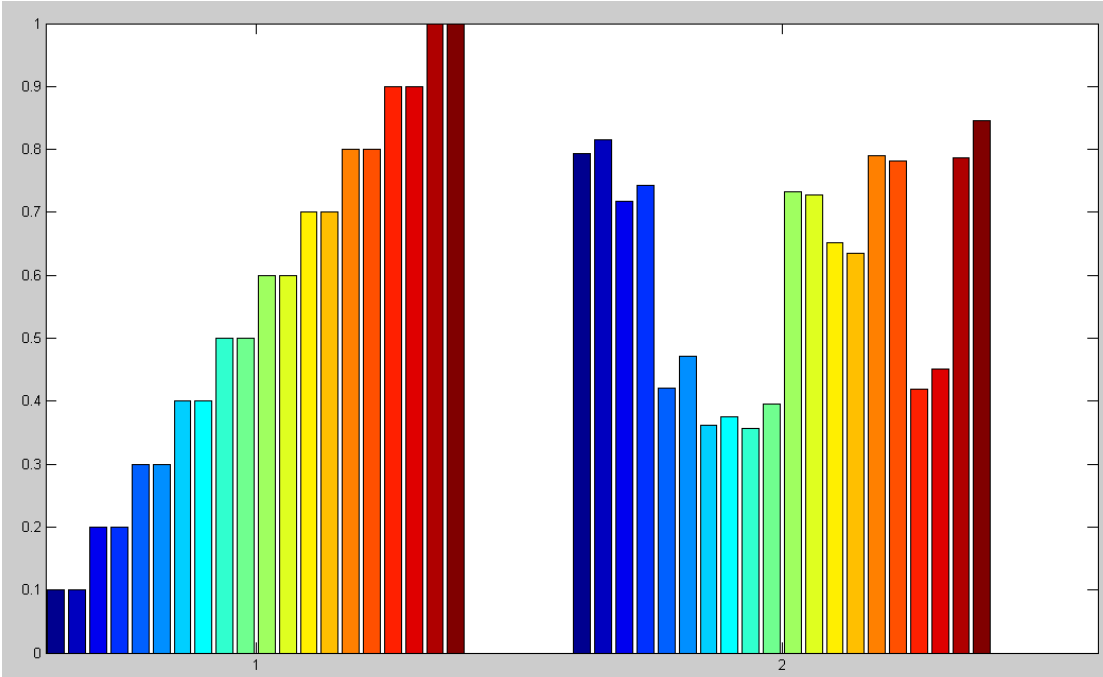


Figure 17: Energy Distribution of Set.2

Table 8: Data Set 3

Input	Output
0.1, 0.529	1
0.1, 0.473	-1
0.2, 0.694	1
0.2, 0.704	-1
0.3, 0.605	1
0.3, 0.585	-1
0.4, 0.155	1
0.4, 0.179	-1
0.5, 0.594	1
0.5, 0.659	-1
0.6, 0.740	1
0.6, 0.719	-1
0.7, 0.799	1
0.7, 0.837	-1
0.8, 0.823	1
0.8, 0.793	-1
0.9, 0.811	1
0.9, 0.845	-1
1, 0.809	1
1, 0.886	-1

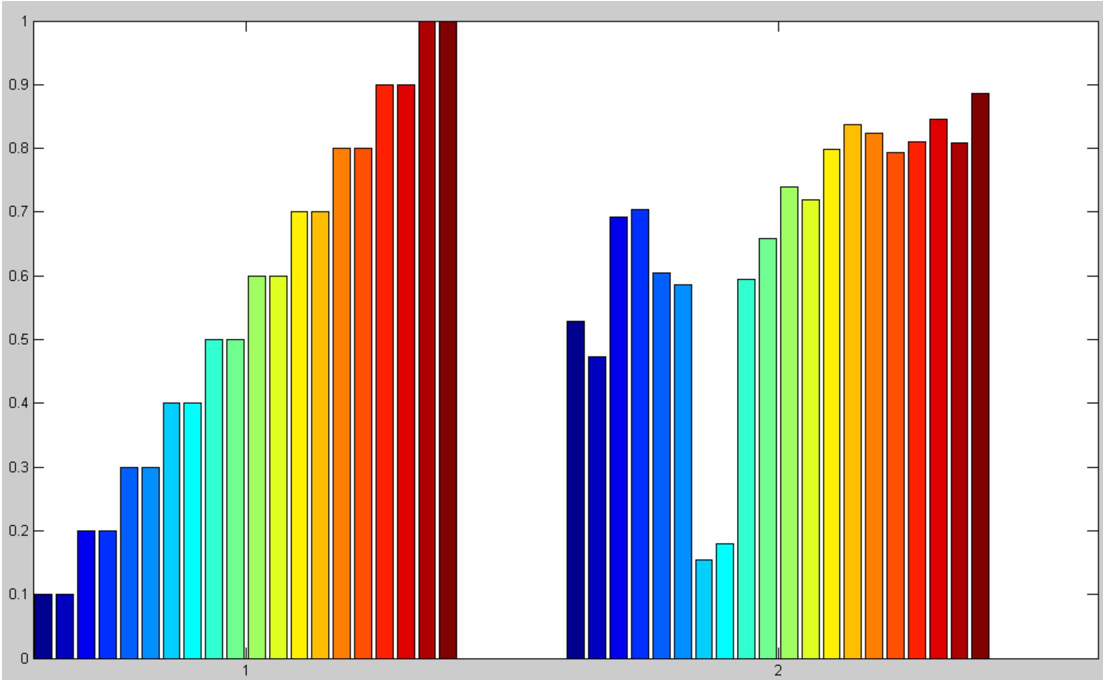


Figure 18: Energy Distribution of Set.3

One decided to employ three data sets of 10 patients per run for the NN. The first set of the data represents ten patients doing two different sets of movements. The following are the movements:

Movement 1 (1) = Imagine Opening and Closing Left or Right Fist
Movement 2 (-1) = Imagine Opening and Closing Both Fists or Both Feet

From these movements, we established a percent energy average for all 64-electrodes (refer to earlier section for more information) which is the second input to the NN, with the first being the Patient ID. From Figure ## & Figure##, we can observe the variation between the percent energy distribution of both sets of patients. This allows us to further test the NN due to the variation of each patient. The next step was to implement prototype NN to test our block diagram logic.

Neural Network Code

First Attempt:
The following is an image of the first implemented code along with the results:

```
p = [[0.1;0.635] [0.1;0.686] [0.2;0.389] [0.2;0.462] [0.3;0.
t = [1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0];
net = newp([0 1;0 1],1,'tansig','learnp'); %Setting up NN

net.iw{1,1} = [0.015 0.768]; %Input weights
net.b{1,1} = 0.971; %Input bias
net.trainParam.epochs = 100;
net = train(net,p,t);


test_output = sim(net,p);
weight_final = net.iw{1,1}; % show the value of weights
bias_final = net.b; % show the value of bias
plotpv(p,t) % only works if t = 0 or 1
plotpc(net.iw{1,1},net.b{1,1})
```

Figure 19: First NN Attempt

As we can see from the code, variable "p" is the input to the NN. The first number is our Patient ID number and the second is our established threshold percent energy for that specific person and whichever movement he/she is thinking about. The second variable is "t" which is the output required if the first two initial statements are true. For example, if patient "0.1" reaches the percent energy threshold of 0.686, then the second movement is triggered which will output, "0" (Note: the second output is later changed to "-1"). After establishing the rules, the function "newp" was used to implement the NN; the following are the results of the testing:

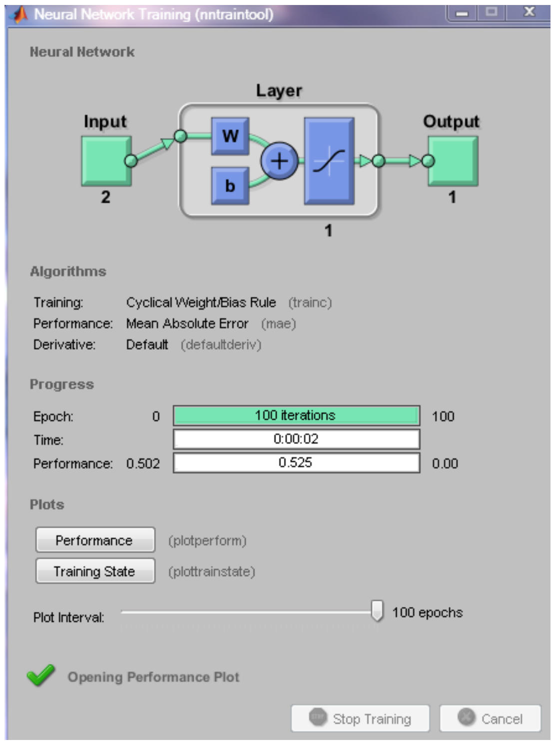


Figure 20: NN Tool, Attempt 1

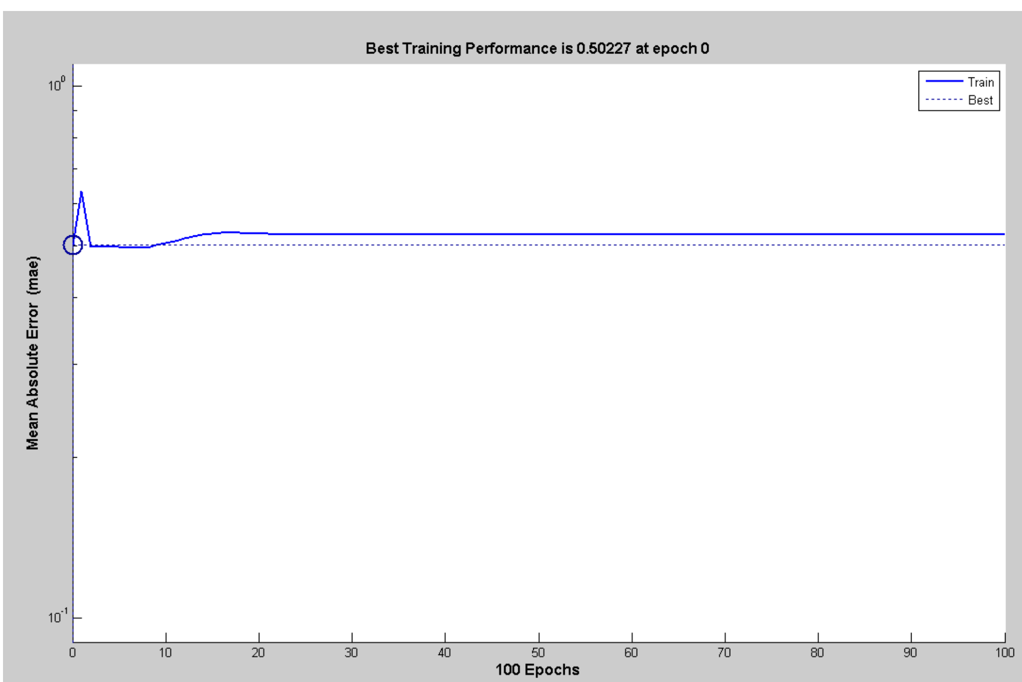


Figure 21: NN Performance Plot, Attempt 1

From the previous images we can observe that the NN did not work to our satisfactory performance rate, which is $1x10^{-7}$ or approximately 0. The average performance turned out to be 0.502277 in 100 iterations. With these results we clearly needed a different approach to the NN. The second attempt was implemented with a bigger NN that would be able to perform with more neurons and hidden layers.

Second Attempt:
The following is an image of the second attempt with a bigger NN:

```
p = [[0.1;0.635] [0.1;0.686] [0.2;0.389] [0.2;0.462] [0.3;0.645] [0.3;0.704] [0.4;0.812]

% Movement#1: 1
% Movement#2: -1

t = [1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1];

net = newff([0 1;-1 1], [15 1],{'tansig' 'purelin'}, 'trainlm', 'learngdm','mse');

net.layers{1}.initFcn='initwb';
net.layers{2}.initFcn='initwb';
net.inputWeights{1,1}.initFcn='rands';
net.layerWeights{2,1}.initFcn='rands';
net.biases{1,1}.initFcn='rands';
net.biases{2,1}.initFcn='rands';
net=init(net);

net.trainParam.epochs=200;
net.trainParam.goal=1e-07;
net=train(net,p,t);
test_output=sim(net,p);
```

Figure 22: Second NN Attempt

This figure represents the second and successful attempt at the NN. The same variables apply to the preceding code, however a new function (newff) was used. This function has the capability of establishing hidden layers and more neurons. The following is a brief overview of the function:

```
net = newff(PR,[S1 S2...SNl],{TF1 TF2...TFNl},BTF,BLF,PF)

Description
NEWFF(PR,[S1 S2...SNl],{TF1 TF2...TFNl},BTF,BLF,PF) takes,
          PR  - Rx2 matrix of min and max values for R input elements.
          Si  - Size of ith layer, for Nl layers.
          TFi - Transfer function of ith layer, default = 'tansig'.
          BTF - Backprop network training function, default = 'trainlm'.
          BLF - Backprop weight/bias learning function, default = 'learngdm'.
          PF  - Performance function, default = 'mse'.
and returns an N layer feed-forward backprop network.
```

In the new code we established a total of 15 neurons for 1 layer and 1 for the next. The previous code is also the first data set of people. The weights of each input are established randomly by the function “rands” however this only does a random weight from 0 to 1. This is done so that the NN does not exhaust itself this higher order number. This is also the reason why we decided to use, for example, 0.732 instead of 73.2%. If we used higher order numbers like so, the NN will take too long to establish its feed forward learning patterns. The secondary inputs to the NN function are training patterns and learning algorithms. These functions where chosen by the default suggestions. Except, “tansig” which was recommend by our advisor. Being that we are using tangent function, an easier more distinguishable output would be 1 and -1 which are the limits to the function. This is the reason for why the movements are classified like so. Lastly, we trained the NN a total of 200 times although it did not need all 200 to reach the trained stage. In the next section, we show the results for both trials using this newly established NN.

Simulations

Trial 1:

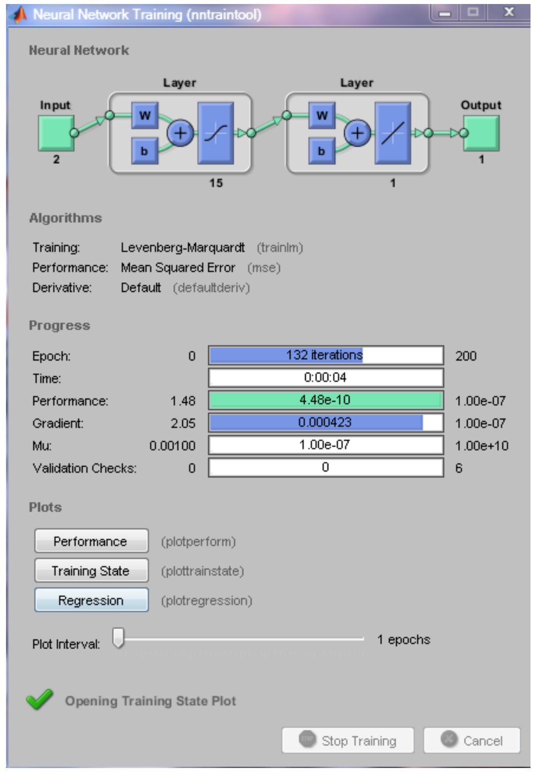


Figure 23: Train Tool, Trial 1

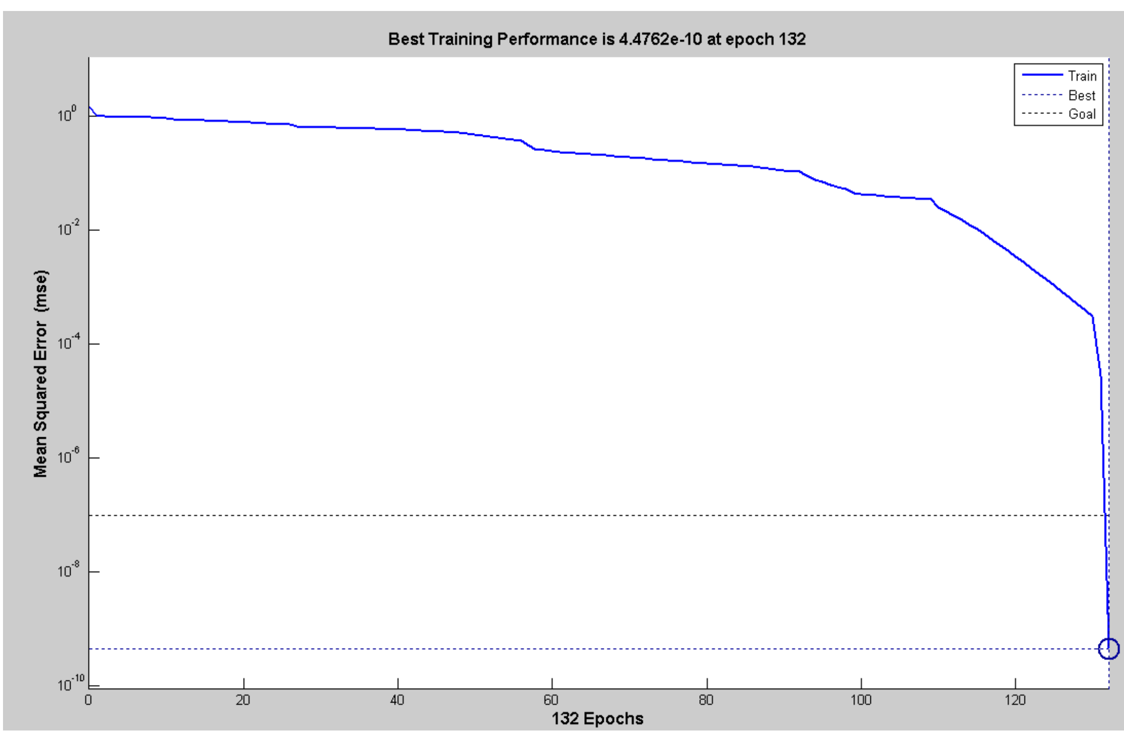


Figure 24: Training Performance, Trial 1

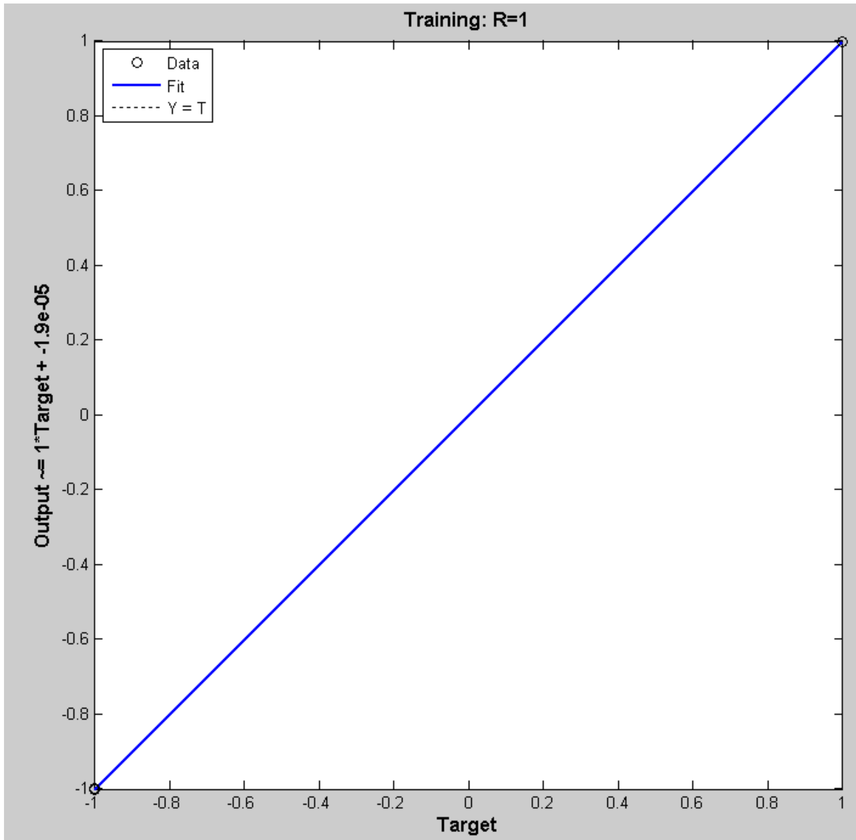


Figure 25: Training Regression, Trial 1

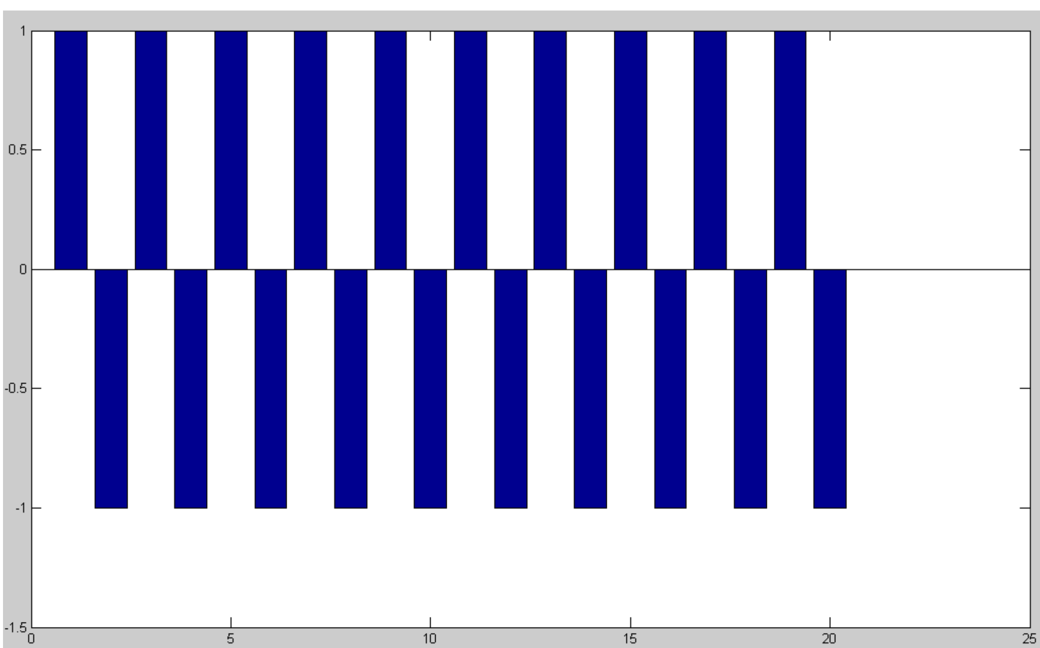


Figure 26: Test Results, Trial 1

From the previous simulations of trial 1, we can observe that 132 iterations where required in order to train the NN to detect the proper movement. Figure # shows the training tool to the NN, this tool shows the general outline of the NN which in this case is 15 neurons and 1 hidden layer.

This number of neurons was chosen from trial and error. From the previous unsuccessful runs, 15 was the limit at which we would gather enough training iterations to make a successful NN before hitting our established 200 limit iteration. This tool also shows us links to see outcomes such as the performance, training state, and regression of the NN.

Figure #, shows the “test_output=sim(net,p)” function. This function serves to do the final test on the training of the NN. As we can observe from the previous image, the users have been determined to do either movement 1 or -1. This test was given to 10 people for this trial, thus providing us with a total of 20 data points. Figure #, shows the finished outcome of this test run, fortunately this test run yielded us with 20 out of 20 accurate movement classifications. The following trial did not initially give us this percentage.

Trial 2:

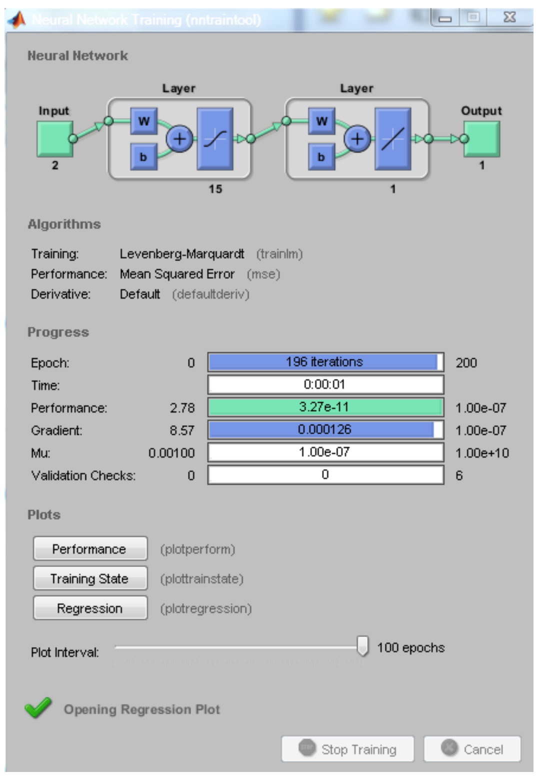


Figure 27: Training Tool, Trial 2

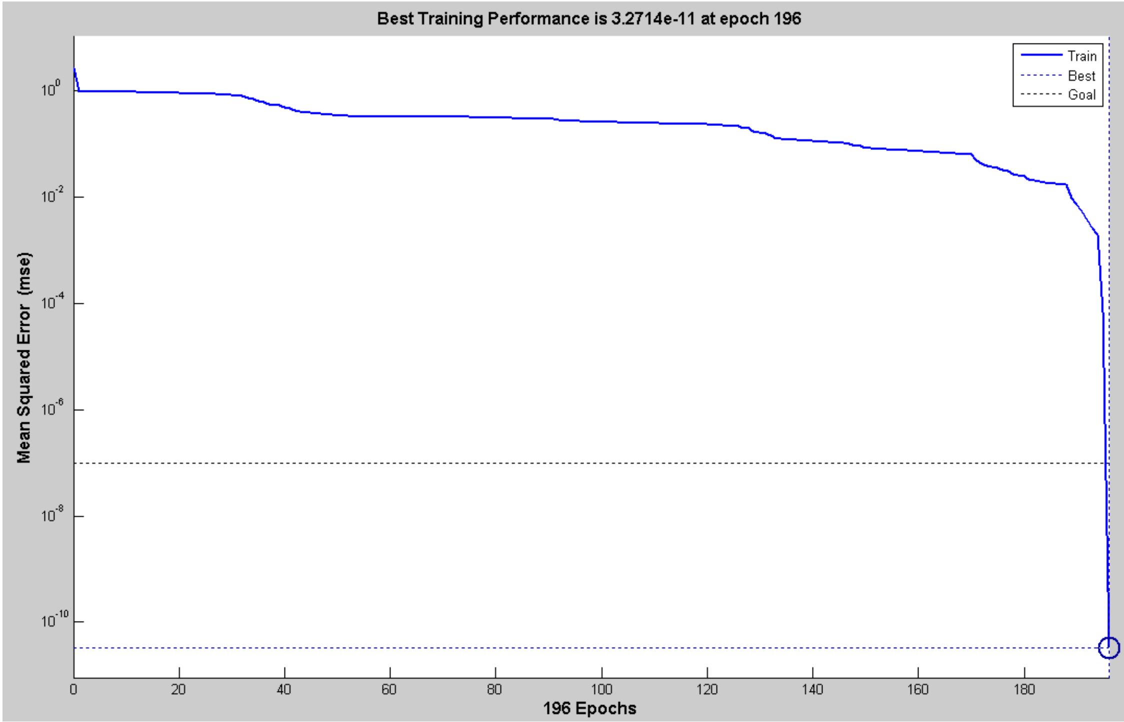


Figure 28: Training Performance, Trial 2

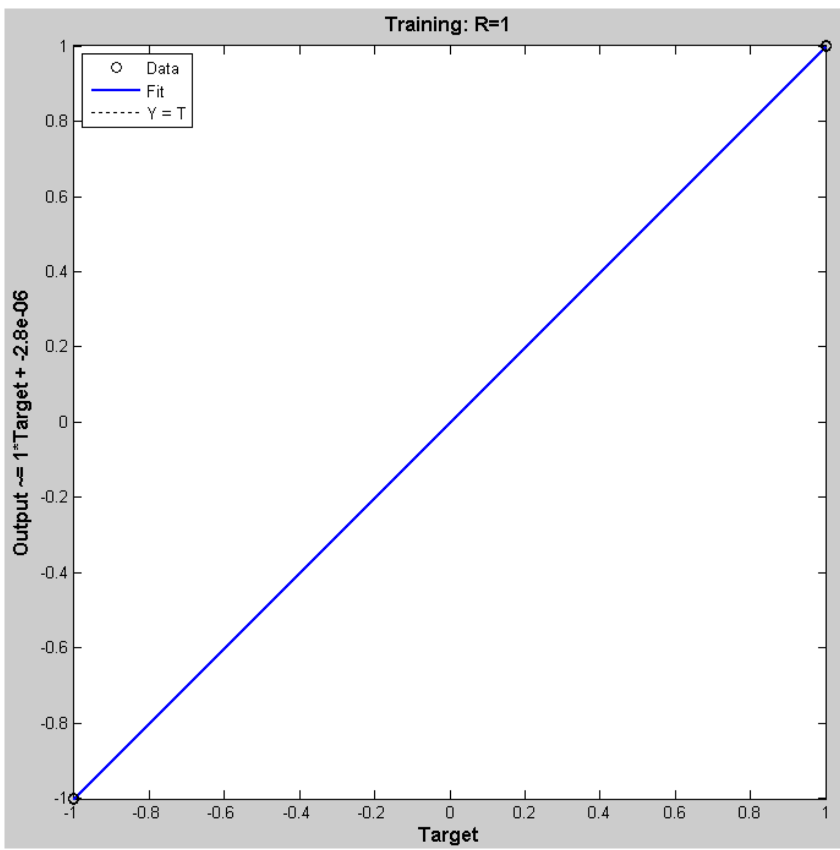


Figure 29: Training Regression, Trial 2

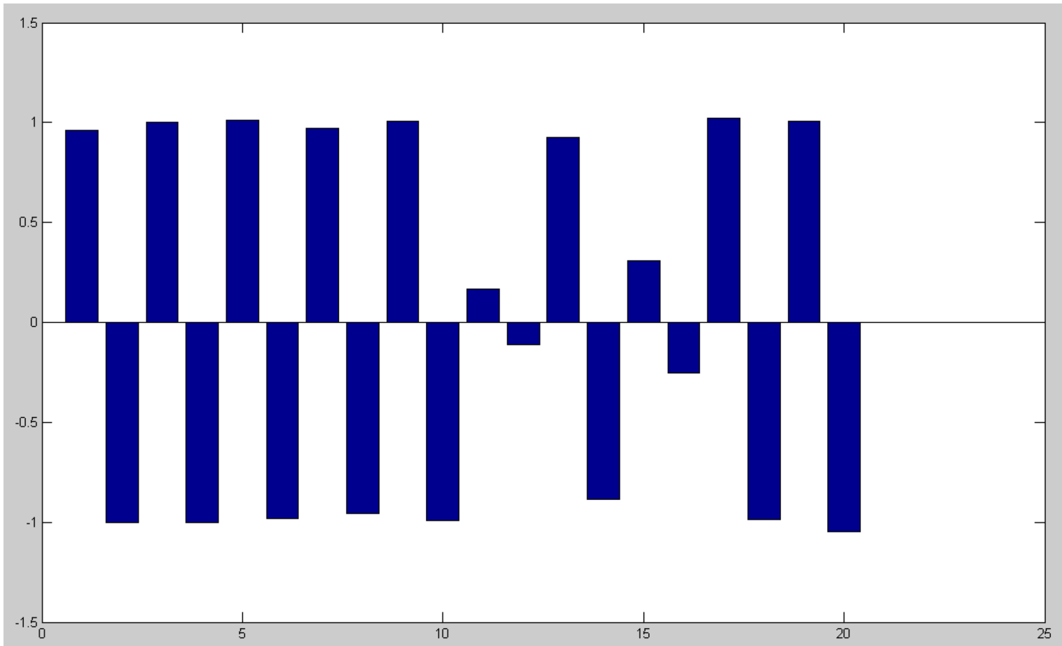


Figure 30: Test Results, Trial 2

Trial 2, from Figure #, we can observe was not as successful as Trial 1. The test results show that approximately two patients (Patient 0.6 & 0.8) did not have a successful EEG classification from the NN. This is not the same percentage accuracy as the first trial but we had expected to see some error from our NN. However, this trial provided us with crucial information about the reliability of the NN.

From Figure #, indicates that's this training took approximately 196 iterations, which is more than the first trial. In addition, the gradient was greater for this second trial with a final value of 8.57; this higher value indicates more fluctuation within the training. With more fluctuation comes more likely hood of failure in the NN. Also, due to the instability of this NN, the Training State graphs also seem to fluctuated significantly.

Trial 3:

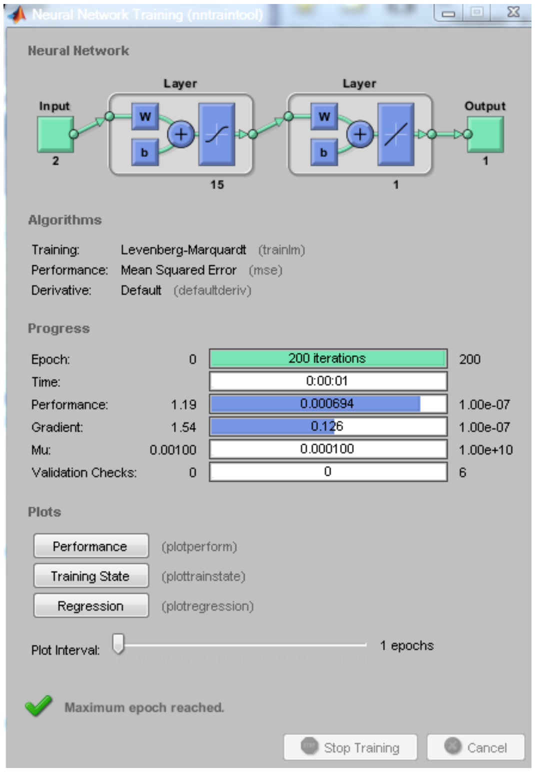


Figure 31: Training Tool, Trial 3

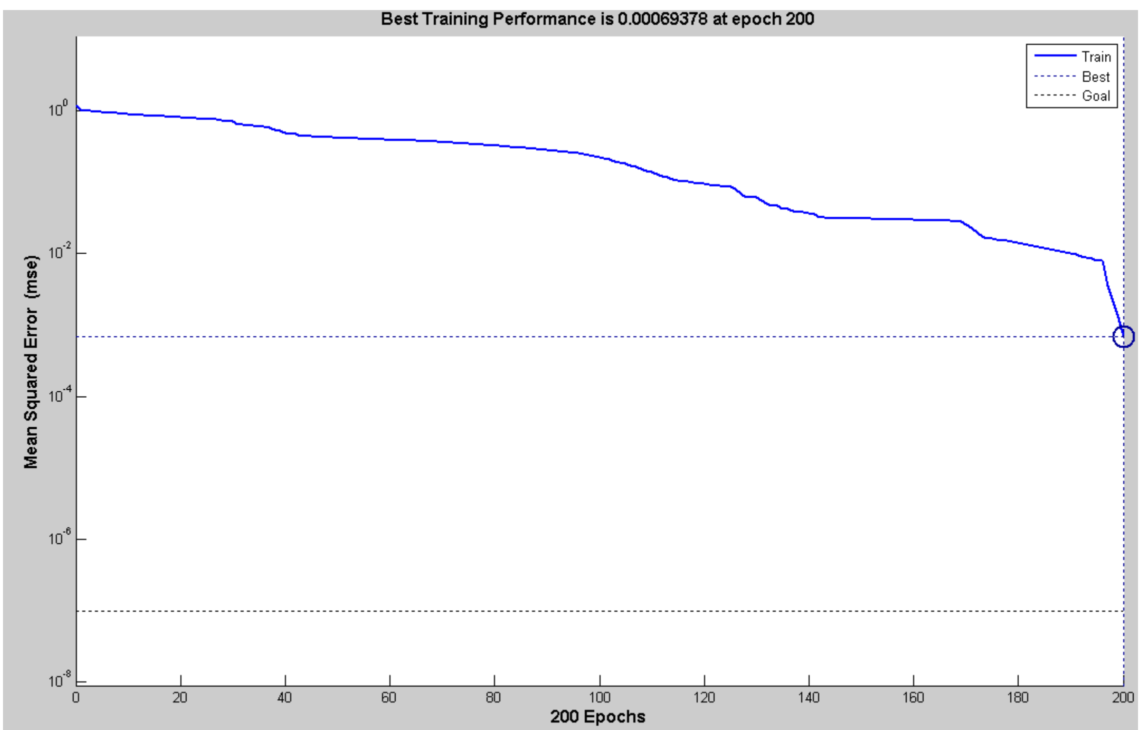


Figure 32: Training Performance, Trial 3

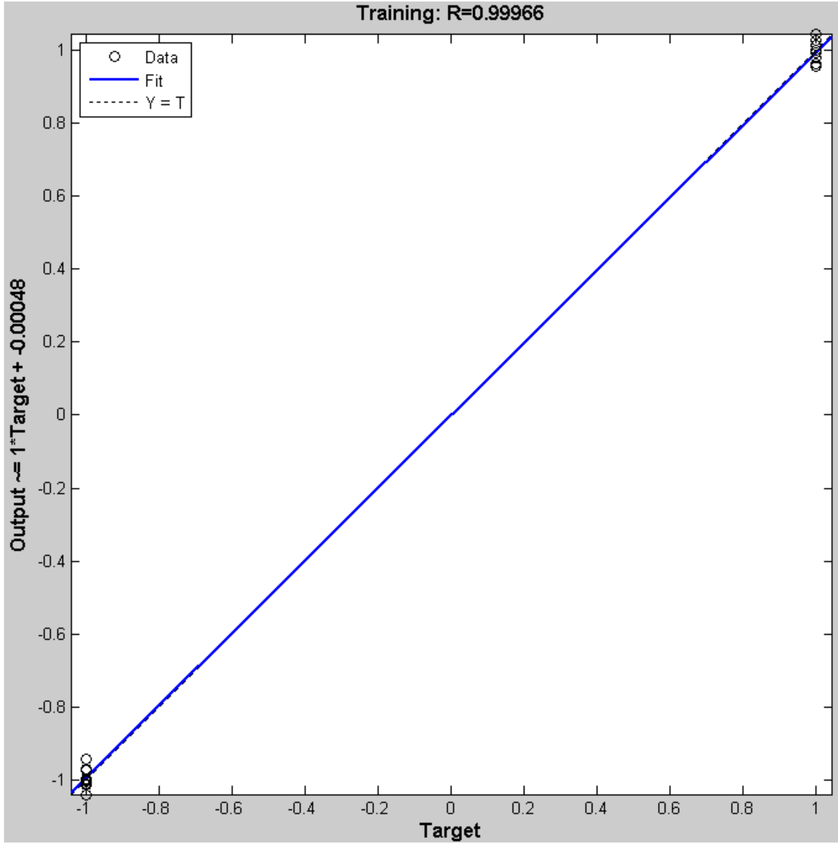


Figure 33: Training Regression, Trial 3

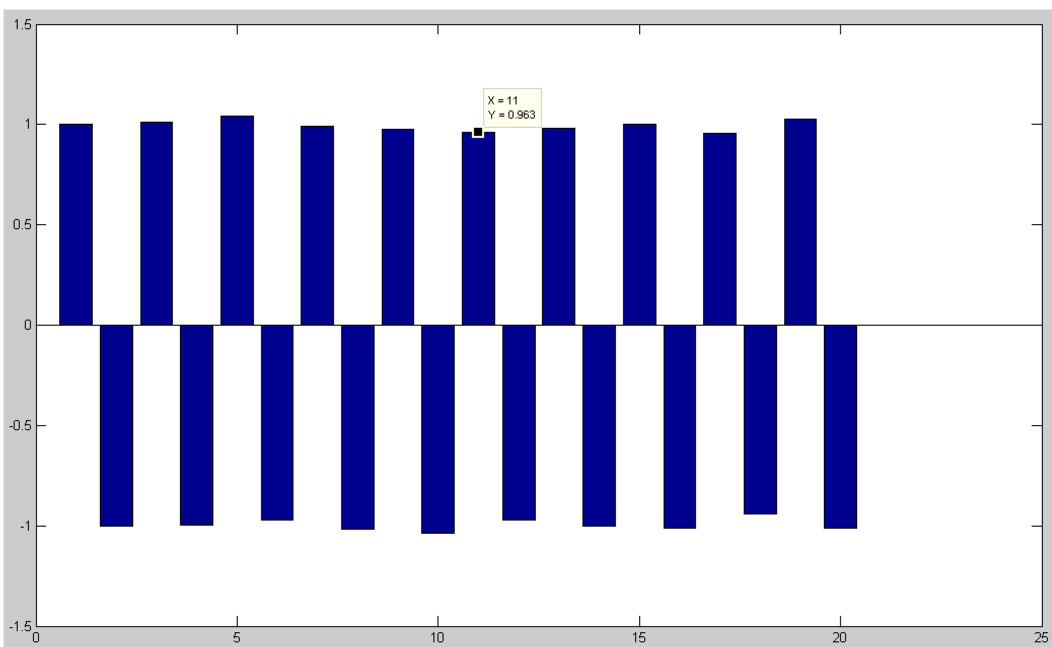


Figure 34: Test Results, Trial 3

Conclusion / Recommendations

With the completion of the final design of our optimally efficient neural network, three runs in sets of 10 were executed and the results can be observed in the table 7 below.

Table 9: Results from Neural Network. 30 runs, with an approximate accuracy of 94%*.

Patient	Movement 1		Movement 2	
	Result	Expected	Result	Expected
1	1	1	-1	-1
2	1	1	-1	-1
3	1	1	-1	-1
4	1	1	-1	-1
5	1	1	-1	-1
6	1	1	-1	-1
7	1	1	-1	-1
8	1	1	-1	-1
9	1	1	-1	-1
10	1	1	-1	-1
11	0.98	1	-1	-1
12	1.01	1	-0.99	-1
13	1.03	1	-0.97	-1
14	0.98	1	-0.96	-1
15	0.99	1	-1	-1
16	0.23	1	-0.22	-1
17	0.87	1	-0.84	-1
18	0.27	1	-0.25	-1
19	0.98	1	-0.95	-1
20	0.96	1	-1.02	-1
21	1	1	-0.99	-1
22	1.02	1	-0.98	-1
23	1.04	1	-0.97	-1
24	0.98	1	-1.01	-1
25	0.97	1	-1.03	-1
26	0.96	1	-0.96	-1
27	0.97	1	-0.97	-1
28	0.98	1	-0.98	-1
29	0.96	1	-0.97	-1
30	1	1	-0.98	-1
Key:	**Classified Signal**	**Questionable Signal**	**Unclassified Signal**	

It can be seen that the neural network yielded an approximate accuracy of 94%. That being said, the pool of data that was readily available in order to train the neural network was much larger than the one used to train it. The available data came from 109 different patients who all performed the same two motions. For the training of our network, we only incorporated 30 of the 109 individuals.

It is important to note that out of the 30 patients that were used, two of their responses could not be classified due particularly close energy distributions for the two movements in question. Furthermore one of the patients energy distributions were slightly below what we considered classifiable(approximately 90% of unity). However, it is important to realize that all signals are not going to be able to be classified 100%, the brain is a complex system and simple gestures require a wide array of action potentials firing from hundreds sometimes thousands of neurons. However, one suggestion to further move this investigation forward would be to incorporate fuzzy logic to make a final decision from the data at the output of the neural network.

Using fuzzy logic would help establish a more rigid set of standards to classify the signal. In essence the fuzzy logic will function as a type of mux to help distinguish the movements. In the case of patient 17 from table 9, the fuzzy logic would help determine whether or not it is a particular type of signal. For example, assume that that the fuzzy logic had a set of system rules which one implements in order to determine the type of EEG signal is in question. These rules could be implemented in the form that would allow questionable signals like that of patient 17's in terms of their energy distribution to be more readily classifiable.

Overall the primary goal of the project was achieved in terms of being able to classify particular EEG signals that correspond to a particular movement. The addition of fuzzy logic to our neural network design will greatly increase the accuracy in terms of dealing with questionable outputs of our neural network.

Appendix A – Analysis of Sr. Project

Project Title: Upper Limb Restoration; EEG Classification Library
Student's Name: Miguel Contreras & Javier Suárez
Student's Signature:
Advisor's Name: Xiao-Hua (Helen) Yu
Advisor's Initials:
Date:

• Summary of Functional Requirements
Primary Goal: The goal of this project is to develop new methods for the classification of EEG motor signals that can be compiled into a library for future projects. The intended classification of signals will range from being able to distinguish movements in all of the major axes to a combination of movement in any of the axes. The compilation of the classified signals into a library will later be implemented into a design that will employ the use of a robotic arm to translate EEG signals to real world movements.
Secondary Goal: The goal of this project is to develop both methods and technologies for brain controlled robotic devices like a robotic arm. Examples of the intended features will range from being able to grasp the left most object from a table of objects. The robotic arm will be autonomous to a degree that will allow for it to make intelligent decisions based on the surrounding environment.

• Primary Constraints
Describe significant challenges or difficulties associated with your project or implementation. For example, what were limiting factors, or other issues that impacted your approach?
Non-Invasive approach: The challenging portion of having a non-invasive approach is the limitations on signal accuracy that an EEG cap provides.
Digital Signal Processing (DSP): Further filtering and amplification (DSP) of the EEG signals will be the most challenging portion of the project. This is why we are choosing to significantly focus on the early stages of the project.
EEG Cap: Due to the fact that EEG machines are fairly costly, we are sharing an EEG with three other students who are currently working on projects themselves.

• Economic
• What economic impacts result? Consider:
The economic impact of Neuro Tools is focused more on the natural resources and availability of semiconductor material to produce the tools for extraction of EEG signals. The basis of Neuro Tools is a semiconductor that sits at the heart of the EEG machine that will ultimately be the core of the project. However, that is not to say that while the product itself is a machine it eliminates all human capital and costs with it. Maintenance will be a necessity and thus human capital will be required and as a result this will ultimately warrant a higher product costs. Fortunately the product will be manufactured rather than handmade. Manufacturing provides automation and ultimately reducing the overall costs of human capital and necessity for human interaction with the production of the product. As discussed earlier, the major obstacle this product will face is the prices incurred by the purchasing of semiconductor material. While this project will not necessarily design the semiconductor that will be the core of the project, this will provide the basis for future projects involving implanted devices needing the development of an ASIC.
• When and where do costs and benefits accrue throughout the project's lifecycle?
For any given projects, costs accrue on the early stages (prototypes) and first fab outs of the product. Further design revisions will allow us to take into account more cost efficient ways of implementing the same product. For example, component cost and PCB layout/penalization design optimization for cost reductions. After the first fab outs the product should start accumulating cost benefits.
• What inputs does the experiment require? How much does the project cost? Who pays?
The project will have user EEG signals as inputs, which will require an EEG cap to capture the signals. After some filtering and amplification of the EEG signals, a microprocessor, with usage of various sensors (distance detection, inductive sensors) will send commands to an external robotic arm. We're estimating a total cost for this product of $550.00 (without EEG machine) and a total cost of $3840.00 for labor.
Original estimated cost of component parts (as of the start of your project).

We're setting aside $300 in miscellaneous cost adders and estimating that our pre-fabricated robotic arm will be approx. $250 giving us a total of max cost of components: $550.00

Additional equipment costs (any equipment needed for development?)

EEG Machine – Approx. $1500.00

Microprocessor/Sensors – Approx. $50

• How much does the project earn? Who profits?

According to our calculations we're expecting a profit of 45% long term. Initially not much profit will be made because we want to make sure customers are happy with products so first shipments will be at fabrication cost. The profits will benefit customers in that it will be placed for further research and solutions to improving product for future developments.

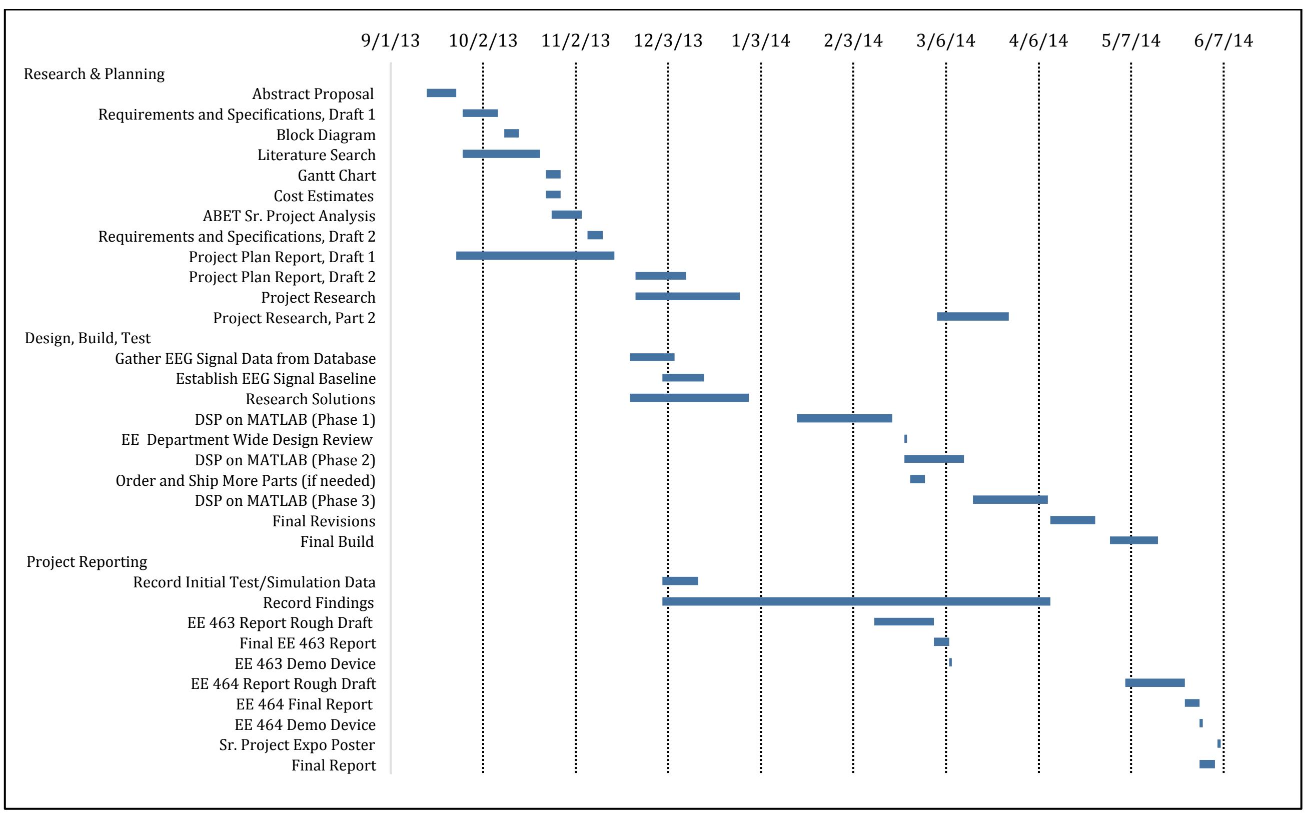
• Timing

When do products emerge? How long do products exist? What maintenance or operation costs exist?

Products are expected to emerge about two months from its early stages of production. The reason why we are assuming the product will take this long to produce is due to the testing and validation prior to the first customer shipment. We are to make sure that all the products have been thoroughly tested for customer satisfaction. When the product is in the field, we are estimating a life span on minimum 10 years. The reasoning behind this prediction is simply the increase in reliability for electronic components in the medical field. The product should not require much maintenance, however there are recommended steps in order to keep the product in good conditions for a longer duration. Recommendations can include: battery replacements (still researching the expected life for each battery charge), environment temperature, not exceeding physical limitations of the robotic arm, etc.

Original estimated development time (as of the start of your project), as Gantt or Pert chart

The original timeline deadline for the project is on April 30, 2014. Which gives the project from design to build process a total of 229 days.



	Start Date	Duration (Days)	End Date
Research & Planning			
Abstract Proposal	September 13, 2013	10	September 23, 2013
Requirements and Specifications, Draft 1	September 25, 2013	12	October 7, 2013
Block Diagram	October 9, 2013	5	October 14, 2013
Literature Search	September 25, 2013	26	October 21, 2013
Gantt Chart	October 23, 2013	5	October 28, 2013
Cost Estimates	October 23, 2013	5	October 28, 2013
ABET Sr. Project Analysis	October 25, 2013	10	November 4, 2013
Requirements and Specifications, Draft 2	November 6, 2013	5	November 11, 2013
Project Plan Report, Draft 1	September 23, 2013	53	November 15, 2013
Project Plan Report, Draft 2	November 22, 2013	17	December 9, 2013
Project Research	November 22, 2013	35	December 27, 2013
Project Research, Part 2	March 3, 2014	24	March 27, 2014
Design, Build, Test			
Gather EEG Signal Data from Database	November 20, 2013	15	December 5, 2013
Establish EEG Signal Baseline	December 1, 2013	14	December 15, 2013
Research Solutions	November 20, 2013	40	December 30, 2013
DSP on MATLAB (Phase 1)	January 15, 2014	32	February 16, 2014
EE Department Wide Design Review	February 20, 2014	1	February 21, 2014
DSP on MATLAB (Phase 2)	February 20, 2014	20	March 12, 2014
Order and Ship More Parts (if needed)	February 22, 2014	5	February 27, 2014
DSP on MATLAB (Phase 3)	March 15, 2014	25	April 9, 2014
Final Revisions	April 10, 2014	15	April 25, 2014
Final Build	April 30, 2014	16	May 16, 2014
Project Reporting			
Record Initial Test/Simulation Data	December 1, 2013	12	December 13, 2013
Record Findings	December 1, 2013	130	April 10, 2014
EE 463 Report Rough Draft	February 10, 2014	20	March 2, 2014
Final EE 463 Report	March 2, 2014	5	March 7, 2014
EE 463 Demo Device	March 7, 2014	1	March 8, 2014
EE 464 Report Rough Draft	May 5, 2014	20	May 25, 2014
EE 464 Final Report	May 25, 2014	5	May 30, 2014
EE 464 Demo Device	May 30, 2014	1	May 31, 2014
Sr. Project Expo Poster	June 5, 2014	1	June 6, 2014
Final Report	May 30, 2014	5	June 4, 2014

What happens after the project ends?

If the project is not completed by April 30, 2014; we will continue in our efforts to make the project complete. Given that the “meat” of the project is in the DSP of the EEG signals, the primary goal as of right now is to successfully distinguish between each motor skill and its specific EEG signal characteristic.

• **If manufactured on a commercial basis:**
• Estimated number of devices sold per year
Approx. 50 Devices
• Estimated-manufacturing cost for each device
Approx. $1,600.00
• Estimated purchase price for each device
Approx. $2,320.00
• Estimated profit per year
Approx. $36,000.00
• Estimated cost for user to operate device, per unit time (specify time interval)
Approx. $0.10 per hour (note: this estimate came from the life expectancy for the batteries used in device)

• **Environmental**
• Describe any environmental impacts associated with manufacturing or use, explain where they occur and

quantify.
Any given material used in the factory to create, robotic arm, microprocessor, and EEG machine. Materials may include, but not limited too: Steel, PCB materials (epoxy resin, copper, energy (fuel) produced by PCB assembly line), EEG machine chassis (plastic).
• Which natural resources and ecosystem services does the project use directly and indirectly?
Directly: Non-recyclable materials – batteries and components of the PCB into the immediate ecosystem.
Indirectly: Fuel (manufactures factories and transportation vehicles). Other consequences from fabrications can include, but not limited too:

- Contaminated rinse water (may be contaminated with heavy metals and/or solvents)
- Waste chemicals
-Effluents, which may contain metals such as copper, lead, chromium, antimony, nickel, and gold, organic solvents, acids and cyanides
- Waste boards
- Acidic air emissions
- VOC emmissions

• Which natural resources and ecosystem services does the project improve or harm?
Improve: No petroleum of gas fumes will be dispersed during the usage of the machine so no ecosystems in the immediate surroundings will be effected by the product.
Harm: Natural resources and ecosystems that will be harmed from the product would be the locations where the product will be fabricated. The fab center would be located around an ecosystem that will be harmed by fumes, industrialization/expansion, and perhaps dumping by the fabrication.
• How does the project impact other species?
Project does not directly impact other species, however indirectly from manufactures and transportation; certain species in the vicinity will get impacted from petroleum/gas.

• **Manufacturability**
• Describe any issues or challenges associated with manufacturing.
Challenges: the main challenges with this device would be the number of devices predicted to manufacture per year. Given that it's a customized and thoroughly tested product, the production numbers will not be to high. With this, we feel it will be hard to find fabricators willing to support such products that will not be massed produced. Also given that it will not be massed produced, profits will be harder to achieve.
Issues: the product will have one common fab house that will assemble all the parts together, however everything will be outsourced from various fabricators. EEG cap and robotic arm will be sourced from different suppliers. Also the outsourced parts must have various suppliers in case of natural disasters, the fab line will not come to a halt.

• **Sustainability**
• Describe any issues or challenges associated with maintaining the completed device, or system.
Sustainability of the product should be fairly simple. The primary maintenance required would be the replacements of batteries in device and further recommendations given by us to the user. Recommendations for maintenance may include proper usage of robotic arm (ex. Not exceeding lifting limitations, etc.), usage of product under specified environments (not underwater, not under extreme heat/coldness, no exposure to physical damage).
• Describe how the project impacts the sustainable use of resources.
Given that the product runs on batteries, the only impact that the project will have on resources will be the recycling and manufacturing of the batteries.
• Describe any upgrades that would improve the design of the project.
Perhaps later create a self-powered/chargeable device to disregard the battery maintenance and environmental issues. This will allow the users to save money in the long run by not purchasing batteries.
• Describe any issues or challenges associated with upgrading the design.
Challenges will include the portability of the product, as it will need some sort of power generating supply that will expand the component list for the product and thus add more weight, larger physical dimensions, and increase the production cost. Further research and development will be required which will push back the final design deadline.

• **Ethical**

• Describe ethical implications relating to the design, manufacture, use or misuse of the project.
The development of this system presents some serious potential for misuse. This system will provide the classification of brain signals extracted via an EEG machine and place them in a library that can be employed later by other brain computer interfaces. For the purpose of this study, the signals will be used as the input to a tool, robotic arm, that will provide the necessary motor help a paralyzed or disabled individual needs.
However, the potential for misuse is in the creation of the library.
This library could be the stepping-stone that others might use to generate their own type of library that could be used for ethically wrong reasons. One aim of this project is to send the filtered, amplified, and classified signals wirelessly to a micro-controller to translate signals into hardware functions. It would be naive to believe that no parties would benefit from the ability to transmit EEG signals wirelessly to a remote location to perform acts of violence. The ability to control robotic devices, vehicles, or drones wirelessly via a brain machine interface is a recipe for misuse. Unwilling to provide tools that can aid in the potential harm of life, the classification of signals will be limited to a finite number of commands, and will not be open sourced to the public. This is not to say that one cannot determine a similar algorithm, but more so as a preventative measure to avoid allowing the user direct exposure with the code.
In the development of medical devices in terms of bioMEMS, biosensors, and bioelectronics as a whole one has to take into consideration that while efforts may be focused around the idea of providing medical solutions to existing problems, at the same time these same solutions can be turned into devices that the designer did not design for. However, if one was to focus on the notion that anything could be used for wrong, then innovation would die and as a result progress would halt. The only thing designers and engineers as a whole can do is incorporate certain measures and standards to limit the availability of a solutions inner workings to the public and hope that their efforts to help is greater than that of another parties effort to use technology as a means to act morally and ethically criminal.

• Health and Safety
• Describe any health and safety concerns associated with design, manufacture or use of the project.
Given that the project is for paralyzed individuals, we must create a machine that will not in any way physically harm the user during their experience with the machine. The non-invasive approach will be taken under stern supervision to make sure the electrodes, microprocessor, and robotic arm are not going to cause any harm. The first safety concern is the direct placements of the electrodes in the users head and the second is the full control of the external robotic arm by the user (no glitches in the signals being detected).

• Social and Political
• Describe social and political issues associated with design, manufacture, and use.
Robots can work round the clock, are easier to repair, don't get sick and don't require staff amenities. Replacing people with robots was seen as a way of reducing labor costs. The replacement of people by automated systems contributes to unemployment in society, especially for the most disadvantaged group — unskilled workers — which can result in long-term unemployment. Although this product will not have this effect in the masses, it might replace perhaps in house nurses or other individuals that help the paralyzed individual.
• Who does the project impact? Who are the direct and indirect stakeholders?
This project will allow paralyzed individuals to regain motor movement with thought controlled robotic devices, thus provide them with a higher level of independence. So with this the direct stakeholders will be the paralyzed individuals that will be using the thought controlled robotic arm as well as the owners of the product, and suppliers. The indirect stakeholders can be the immediate family if the paralyzed individuals and research in the BMED field.
• How does the project benefit or harm various stakeholders?
The project will benefit the direct stakeholders in that for the paralyzed individual, the project efforts will help the individual regain a higher level of independence. With the higher gain of independence, the immediate families of the individual will also gain more confidence in the paralyzed individual in their day-to-day routines/activities. The owners of the product and suppliers will also benefit, if the product has successful results. Furthermore, with observations and test analysis on the product running parallel with the production, much information will be gathered which can be directly applied to the medical field efforts in thought controlled robotic devices [2]. As of right now the only stakeholders that may be harmed can be the owners and the suppliers of the device.
• To what extent do stakeholders benefit equally? Pay equally? Does the project create any inequities?
• Consider various stakeholders' locations, communities, access to resources, economic power, knowledge, skills, and political power.

The stakeholders of this project will ultimately all benefit equally in terms of knowing that they have all contributed to a project that is aimed at helping disabled individuals. However, that is not to say they will be compensated the same, rather every contributing party will likely be compensated differently according to their original contribution and participation. As a result of the product being introduced as a good in the capital market, smaller countries and individuals from those markets will likely play a smaller role in the economic success of the product, however, that is not to say that they will not benefit from it in terms of receiving subsidies for the product should they have a large need for it.

• **Development**

Describe any new tools or techniques, used for either development or analysis that you learned independently during the course of your project.
Throughout the course of this plan many new techniques, software, and development tools skills will be acquired. These include, but not limited too: MATLAB (DSP), LTSpice/PSpice, embedded language (programming microprocessor), EEG machine interface, etc.

References

[1] Chua-Chin Wang; Chi-Chun Huang; Tzung-Je Lee; Cheng-Mu Wu; Gang-Neng Sung; Kuan-Wen Fang; Sheng-Lun Tseng; Jia-Jin Chen, "An Implantable SOC Chip for Micro-stimulating and Neural Signal Recording," *Circuits and Systems,* pp.682-685, 4-7 Dec. 2006. [Online]. Available: IEEE Xplore, http://www.ieee.org. [Accessed Oct. 16, 2013].

[2] Contreras-Vidal, J.L.; Presacco, A.; Agashe, H.; Paek, A., "Restoration of Whole Body Movement: Toward a Noninvasive Brain-Machine Interface System," *Pulse, IEEE*, vol.3, no.1, pp.34-37, Jan. 2012. [Online]. Available: IEEE Xplore, http://www.ieee.org. [Accessed Oct. 16, 2013].

[3] Erfani, A.; Erfanian, A., "The effects of mental practice and concentration skills on EEG brain dynamics during motor imagery using independent component analysis," *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, vol.1, pp.239-242, 1-5 Sept. 2004. [Online]. Available: IEEE Xplore, http://www.ieee.org. [Accessed Oct. 16, 2013].

[4] Jinghao Feng; Na Yan; Hao Min, "A low-power low-noise amplifier for EEG/ECG signal recording applications," *ASIC (ASICON), 2011 IEEE 9th International Conference,* pp.145-148, 25-28 Oct. 2011. [Online]. Available: IEEE Xplore, http://www.ieee.org. [Accessed Oct. 21, 2013].

[5] Lovelace, Joseph A.; Witt, Tyler S.; Beyette, Fred R., "Modular, bluetooth enabled, wireless electroencephalograph (EEG) platform," *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pp.6361-6364, 3-7 July 2013. [Online]. Available: IEEE Xplore, http://www.ieee.org. [Accessed Oct. 16, 2013].

[6] Mladenov, T.; Kim, K.; Nooshabadi, S., "Accurate motor imagery based dry electrode brain-computer interface system for consumer applications," *Consumer Electronics (ISCE), 2012 IEEE 16th International Symposium*, pp.1-4, 4-6 June 2012. [Online]. Available: IEEE Xplore, http://www.ieee.org. [Accessed Oct. 16, 2013].

[7] NCC Medical Co., "Type-C Routine EEG System," Nation7128W datasheet, 2003.

[8] Nirenberg, Lloyd M.; Hanley, John; Stear, Edwin B., "A New Approach to Prosthetic Control: EEG Motor Signal Tracking With an Adaptively Designed Phase-Locked Loop," *Biomedical Engineering, IEEE Transactions on*, vol. BME-18, no.6, pp.389-398, Nov. 1971. [Online]. Available: IEEE Xplore, http://www.ieee.org. [Accessed Oct. 16, 2013]

[9] Sanei, Saeid and J.A. Chambers, "EEG Signal Processing," Chichester, England: John Wiley & Sons Ltd, 2007.

[10] Wilson, Scott B.; Scheuer, Mark; Johnson, Dale; Clear, Scott, "Quick press on EEG electrode with bendable legs," U.S. Patent 8 428 681, September 8, 2012.

[11] Avdakovic S , Dizdarevic K , Nuhanovic A, Omerhodzic I. Energy Distribution of EEG Signals: EEG Signal Wavelet-Neural Network Classifier, in *Proceedings of the Annual Technical Conference -Society of Plastics Engineers*, May 1-5, 1994, San Francisco, 2865-2867. Brookfield, CT: Society of Plastics Engineers.