Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

# Study of Electroencephalographic Time Series

Tesis presentada al

**Posgrado en Física Aplicada**

como requisito parcial para la obtención del grado de

**Maestro en Ciencias**
**(Física Aplicada)**

por

Lic. Luke Tecumseh Cleaver Goodman

asesorado por

Dr. Eduardo Moreno Barbosa
y
Dr. Benito de Celis Alonso

Puebla Pue.
Diciembre 2016

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

# Study of Electroencephalographic Time Series

Lic. Luke Tecumseh Cleaver Goodman

asesorado por

Dr. Eduardo Moreno Barbosa
y
Dr. Benito de Celis Alonso

Puebla Pue.

Diciembre 2016

**Título:** Study of Electroencephalographic Time Series
**Estudiante:** Lic. Luke Tecumseh Cleaver Goodman

COMITÉ

Dr. Javier Hernández López
Presidente

Dr. Mario Iván Martínez Hernández
Secretario

Dr. Elías Manjarrez López
Vocal

M.C. Margarita Amaro Aranda
Suplente

Dr. Eduardo Moreno Barbosa    Dr. Benito de Celis Alonso
Asesores

# Acknowledgements

I thank CONACYT for economic support during the creation of this thesis.

I'm also grateful to the BUAP, FCFM, and all of my professors for the training and knowledge that I have received over the past seven years, during my bachelor's and master's degrees in Applied Physics.

I thank Dr. Elias Manjarrez of the Instituto de Fisilogía and Dr. Ignacio Méndez of the Facultad de Psicología for allowing me to use the EEG equipment in their laboratories and providing their assistance and guidance. I also thank Nayeli Huidobro, Mayte Silva and Ana Laura Utrilla for their assistance in the data collection process.

I thank my advisers, thesis committee, fellow graduate students and friends for their valuable comments, questions and feedback during the process of this thesis.

I also thank everyone who contributed to the creation of python for creating a programming language that is about as fun and user-friendly as a programming language can be, and to the creators of the free modules pygame, parallel python, conda, openpyxl and numpy, all of which I used at some point in this thesis.

Finally, I want to thank my two-year old son, Demian, for telling me to have a good day every morning, and both my sons, Demian and my newborn Sebastian, for giving me a reason to keep going forward.

# Introduction

## 1 Summary

Electroencephalography (EEG) is the recording of electric potentials produced by neuronal activity within the cerebral cortex by placing electrodes on specific locations of the scalp. In this thesis, subjects performed real and imaginary movements of their right arm as EEG signals were recorded using a 32-electrode cap at a sampling rate of 1 kHz. Our experiment consisted of twelve 30-second tests, each composed of 5 repetitions of two opposite movements. Indications on timing, speed, and type of movement were provided by a computer program during the experiment. The 12 tests included two base lines with open and closed eyes, four tests covering all degrees of freedom of the shoulder and elbow, and one for hand movement, as well as imagination of all five movements. We used the Wavelet Transform, with the complex Morlet mother wavelet, to identify how characteristic frequencies changed over time. We then developed a method for generating binary classifier functions capable of distinguishing between two sets of movements based on the wavelet coefficients at a variety of frequencies over 30 electrode channels. Posteriorly, we used the binary classifiers to generate a master classifier function which determined the most probable movement at any given time. Furthermore, we identified the key frequencies and regions that distinguish one arm movement from another.

## 2 Motivation

The primary goal of this thesis was to develop the mathematical and computational algorithms necessary to decipher the neuronal signals that are generated in a person's cerebral cortex when they perform specific arm movements. That is to say, to create a brain-computer interface capable of detecting how a person is moving their right arm or how they imagine they are moving their arm, using noninvasive methods. If we could decipher just five actions (up, down, left, right and one extra for click), a person could move a cursor around a computer screen just by thinking about it. If two different states could be detected, a paralyzed person could answer yes or no questions, and if 8 different states could be detected (up, down, left, right, pronation, supination, and two for contraction and extension at the elbow), a person could move a robotic arm by either moving or thinking about moving their own arm in the same fashion.

Brain-computer interface technology has an enormous range of applications, from natural control of prosthetic limbs, communication with locked-in patients, therapy of phantom limbs and rehabilitation of patients with spinal cord injury to virtual reality, meditation training and neurogaming. Furthermore, the technology is so revolutionary, that once popular, thousands of new applications would be developed by the world community, giving rise to technologies that have yet to be imagined. Since the invention of computers, user interfaces have become more and more intuitive, evolving from control panels to mouse and keyboard to touch screen and accelerometers, to body-recognizing cameras that detect from infrared to visible light. Brain-computer interfaces that skip all intermediaries, converting signals directly from thought to command, seem to be a rational next step in this process.

Similar experiments have been carried out, but to our knowledge, no experiment with so many different states has ever yielded acceptable success rates. Zhang et al report 60-90% success rate for an experiment with four very different movement options, described as free state, arm movement, hand crawl and hand open [1]. Bhattacharyya et.al. report 65-70% success rates for 5 movements (forward, backward, left, right and stop) [2].

## 3 Hypothesis

We hypothesize that for each of the arm movements performed during the experiment, certain identifying patterns can be found in the corresponding brainwaves using wavelet analysis, which are consistent among all individuals.

## 4 Aims

1. Identify the characteristic brain regions and frequencies which show increased or decreased activity whenever one movement is performed, but not when other movements are performed.
2. Generate a classifier function capable of distinguishing what movement a subject is performing in real time based on EEG data.

# General Index

# Chapter 1

# Introduction to Concepts

## 1.1 Principles of Electroencephalography

Neuroscientists and other brain enthusiasts sometimes claim that the human brain is among the most complicated systems in the universe. An average adult human brain contains around $8.6 \times 10^{10}$ neurons and $1.5 \times 10^{14}$ synapses, or connections, between them [3]. Whether or not this is more complicated than an ecosystem, galaxy, black whole or proton is a matter of opinion and philosophy, but the fact remains that brains are very complicated systems and even though the anatomy of brains is well mapped out and enormous advances in neuroscience have been made over the past few decades, many aspects of brains are still not very well understood. Furthermore, there are some fundamental aspects, such as the nature of sentience or how the subjective experiences that make up our everyday lives could possibly be encoded by chemical or electrical signals, which remain a complete mystery [4].

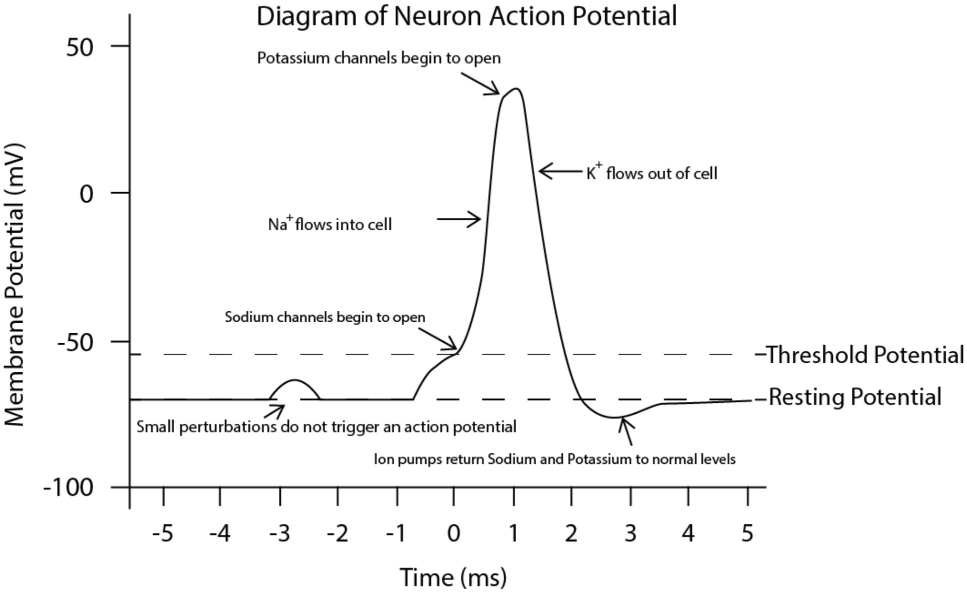


**Figure** 1.1: Diagram of action potential when a neuron fires.

The brain contains both neuronal and non-neuronal cells, but in this thesis, we will discuss only the neuronal cells. Inside each cell, there are certain concentrations of ions such as Potassium (K+) and Sodium (Na+), among others. These ions also exist outside the cell, but in different concentrations. To be more specific, the sodium concentration is about nine times as high outside the cell as inside, whereas the potassium concentration inside the cell is twenty times as much as outside [5, p. 168]. Because ions are charged particles, this difference of concentrations leads to a difference of voltage between the cell's interior and exterior. The difference between interior and exterior voltages is known as the membrane potential, and is normally around -65 mV, but can vary from -80 to -40 mV depending on the type of neuron. The membrane of each cell contains ion channels which, when open, allow ions of a specific type to pass into and out of the cell. If the channels were always open, then the cell would quickly reach equilibrium with respect to the exterior, but they are not; each ion channel has specific gating properties, which determine when they open and close. Potassium and Sodium channels are voltage-gated, meaning that at higher voltages they have a higher chance of opening. Because neurons are constantly being stimulated and ion gates are probabilistic in nature, the voltage of the neuron does not remain constant and fluctuates around the aforementioned resting potential of -65 mV. As long as the fluctuations are small, ion pumps, which actively move ions into and out of the cell by consuming energy from adenosine triphosphate (ATP) molecules, will very quickly return the neuron to its desired ion concentrations and voltage. However, if the fluctuation is large enough to cross the threshold of around -50 mV, then Sodium channels will begin to open, resulting in an influx of positively charged Sodium ions, which will increase the voltage in the cell further, thus causing a chain reaction and opening more Sodium channels. However, when a second threshold is crossed (around +40 mV), Potassium channels will quickly begin to open up, resulting in a flow of positively charged Potassium out of the cell, which eventually cancels then overpowers the effect of the inflowing Sodium, thus reducing the voltage to below its initial value. By this time, both Sodium and Potassium channels will have closed, and the ion pumps will restore the initial concentrations. This increase in a neurons intracellular voltage followed by a rapid decrease is known as an action potential or a spike, and the neuron is said to fire when an action potential occurs [6].

A long offshoot known as an axon leads out of the central cell body of the neuron and then branches out into spines that each end in a presynaptic terminal. When a neuron fires, the action potential does not just appear in a single location; the change of voltage in one part of the neuron causes nearby ion channels to open as well, which increases the voltage there, opening nearby ion channels, and so on. Like a spreading fire, the action potential propagates along the axon until it reaches a presynaptic terminal, where there is a small chance (around 5%) that neurotransmitters will be released and picked up by the post-synaptic terminal of another neuron, producing a voltage increase in the post-synaptic terminal. If the post-synaptic neuron is stimulated in this way by enough neurons within a sufficiently short time, the combined increase in voltage can cross the threshold and cause this second neuron to fire as well. This is how neurons communicate with one another and, at a very basic level, how thoughts are produced.

Due to the movement of charged particles, or ions, across the neuron's membrane during this process, an electrical field is produced with each firing. The electrical field of any given neuron is much too small to be detected outside the skull, but the sum of the electrical activity of the millions of neurons that are firing at any given moment adds up to form a field that can be detected. The

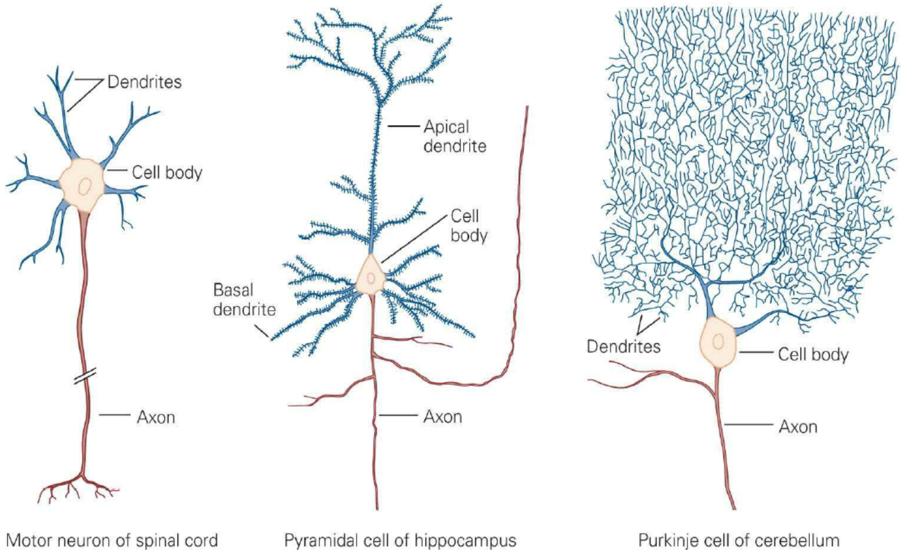
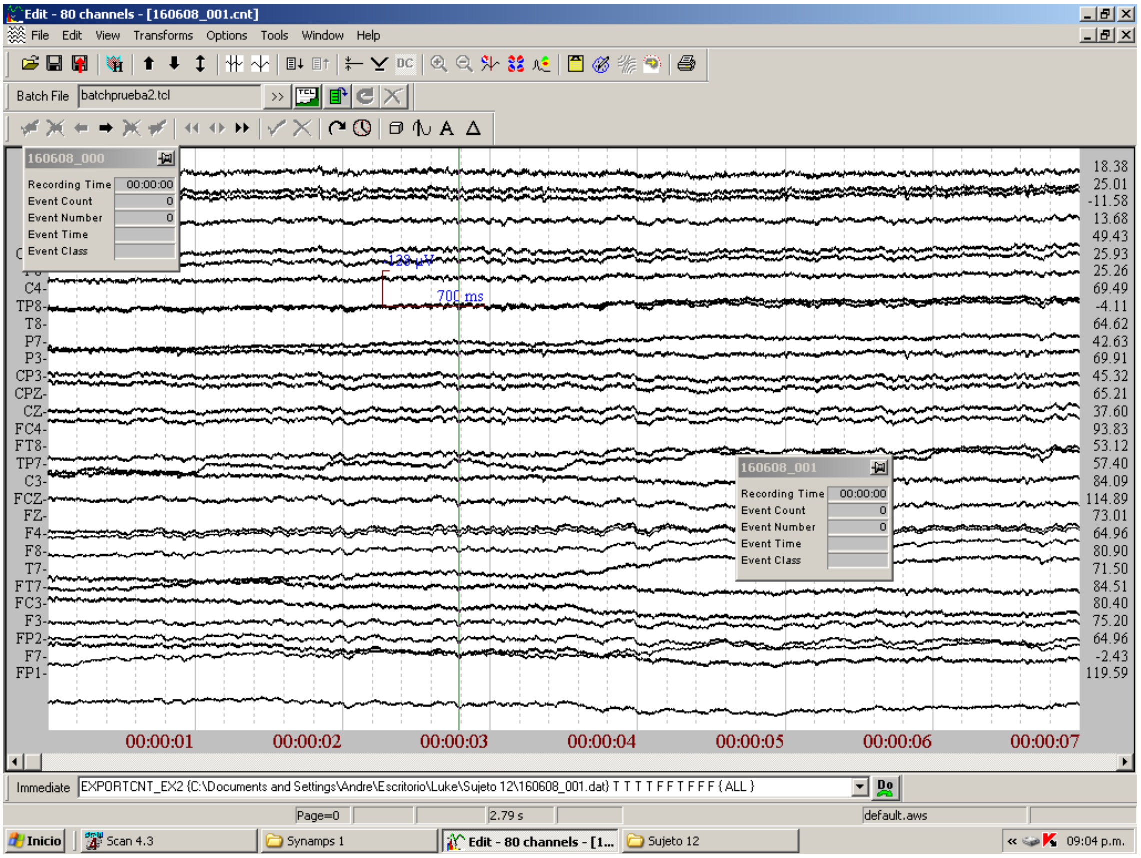


**Figure** 1.2: Diagrams of different types of neurons. Image taken from Principles of Neuroscience [5, p. 67].

fields of neurons firing in random patterns produce destructive interference, but if many neurons in a small region near the brain's surface fire cohesively in more or less the same direction, the component of the accumulated electric field that is perpendicular to the surface of the scalp can be detected by electrodes placed on the scalp. Since neurons tend to be assembled into neural circuits, which perform specific functions such as perception, movement or behavior, it is common for these circuits to fire cohesively, thus producing a measurable signal [6, p. 10].

This is the basic principle underlying the function of electroencephalography, composed of the root words "electro" electricity, "encephalo" brain, and "graph" write, and the suffix "-y" which denotes a system or method. Thus electroencephalography is the method by which electrical signals from the brain are written, and it is commonly abbreviated as EEG. However, the term EEG is also used to refer to "electroencephalograph", the machine used to perform the electroencephalography; "electroencephalogram", the recording obtained from the electroencephalograph; and "electroencephalographical", which is the adjective denoting relationship to electroencephalography. Within this thesis, EEG will be used indiscriminately for all four of these purpose and it will be up to the reader to determine which one is meant from context, although that shouldn't represent much of a problem.

Let us imagine that in one small section of the brain, a chain of neurons fires upward, then activates a second chain which fires downward then activates the initial chain again, and this happens every 0.5 seconds. Then we would expect to find an oscillation of 2 Hz in the EEG signal corresponding to the electrode over that section of the brain. In reality, the pattern would be much more complex, but any time a particular section of the brain is engaged in some periodic mental activity, an oscillation should occur. However, we would also expect all sorts of other oscillations to have been picked up by the electrode at the same time. One way to recover information from the time series is to apply band-pass filters to the EEG data which eliminate oscillations of frequencies that are above or below a certain range. In EEG, there are five commonly used frequency bands,



Batch File batchprueba2.tcl

| | |
|---|---|
| Recording Time | 00:00:00 |
| Event Count | 0 |
| Event Number | 0 |
| Event Time | |
| Event Class | |

Immediate EXPORTCNT_EX2 {C:\Documents and Settings\Andre\Escritorio\Luke\Sujeto 12\160608_001.dat} T T T T F F T F F F { ALL }

Page=0 2.79 s default.aws

**Figure** 1.3: Example of 30 channels of brainwave recordings from an electroencephalograph.

known as $\delta$ (0.2 - 3.5 Hz), $\theta$ (4 - 7.5 Hz), $\alpha$ (8 - 13 Hz), $\beta$ (14 - 30 Hz) and $\gamma$ (30 - 90 Hz), each of which has been found to be associated with specific brain activities. For example, an increase in alpha waves can be seen when eyes are closed, and alpha waves are inhibited during arm movement. Both of these results are in agreement with the findings of this thesis. Frequencies below delta are known as infraslow waves and those above gamma are called high-frequency oscillations (HFO).

| Band | Frequencies | Normally found in |
|---|---|---|
| Delta | 0.2 - 3.5 Hz | deep sleep, babies |
| Theta | 4 - 7.5 Hz | drowsiness, arousal, inhibition, creativity, young children |
| Alpha | 8 - 13 Hz | relaxation, closed eyes, rest state motor neurons |
| Beta | 14 - 30 Hz | alertness, anxiety, analytical thinking, concentration |
| Gamma | 30 - 90 Hz | cross-sensory perception, deep meditation, extreme concentration |

**Table** 1.1: Conventional frequency bands for brainwaves, and the activities with which they are associated.

## 1.2 Wavelet Transform

One of the best mathematical methods for determining the characteristics frequencies of an oscillatory signal that changes over time is the Wavelet Transform, defined mathematically as:

$$W_{\varphi}(s,\tau,x) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} \overline{\varphi\left(\frac{t-\tau}{s}\right)} x(t)dt \tag{1.1}$$

where $\varphi$ is a mother wavelet function, $\bar{\varphi}$ indicates complex conjugate, $s$ is scale, $\tau$ is time displacement and $x$ is the data to be analyzed. The result of this function is a scalar, known as the wavelet coefficient of $x$ at scale $s$ and time $\tau$.

In order to understand the wavelet transform, one should note that it is really not more than a normalized scalar product of two functions. The scalar product, or dot product, of two n-dimensional vectors $\vec{x}$ and $\vec{y}$ is defined as:

$$\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i y_i \tag{1.2}$$

The scalar product denotes the degree of similarity between the vectors $\vec{x}$ and $\vec{y}$. If the scalar product is normalized, by dividing by the product of the norms of $\vec{x}$ and $\vec{y}$: $\frac{\vec{x}\cdot\vec{y}}{xy}$, where $x = |\vec{x}| = \sqrt{(\vec{x}\cdot\vec{x})}$ and similarly for $y$, the number obtained would be between -1 and 1, where 1 indicates that x and y are parallel, -1 indicates that they are antiparallel, and 0 that they are perpendicular. Similarly, the scalar product between two continuous functions $f(x)$ and $g(x)$, defined in the range $x \in (a,b)$ is defined:

$$f \cdot g = \int_{a}^{b} \overline{f(x)} g(x)dx \tag{1.3}$$

and shows the degree of similarity between the two functions. If $\frac{f\cdot g}{\sqrt{f\cdot f}\sqrt{g\cdot g}}$ is near 1, then when one function increases or decreases, so does the other. The peaks and valleys of one function are closely aligned with those of the other. If the normalized scalar product is near -1, then increases in one function correspond to decreases in the other. If the dot product is near zero, then the two functions are completely unrelated.

Readers with a mathematical background may be familiar with the information in the previous paragraph, but I include this reminder because it is essential for understanding the meaning of a wavelet coefficient. In this light, equation (1.1) can be seen as the scalar product between the function we wish to analyze and a mother wavelet, or template. This is to say, it is a measure of how much a function $x$ resembles a template $\varphi$.

There is a large variety of different mother wavelets and selection of one is not an arbitrary or trivial task. A mother wavelet should be chosen depending on the shape of the signal that to be analyzed. For example, the Haar wavelet could be used to identify step patterns, Daubechies to find action potentials, or Morlet to look for sine or cosine oscillations. In some cases, analyzing a signal with two different mother wavelets will provide similar results, but often, the results will be very different.

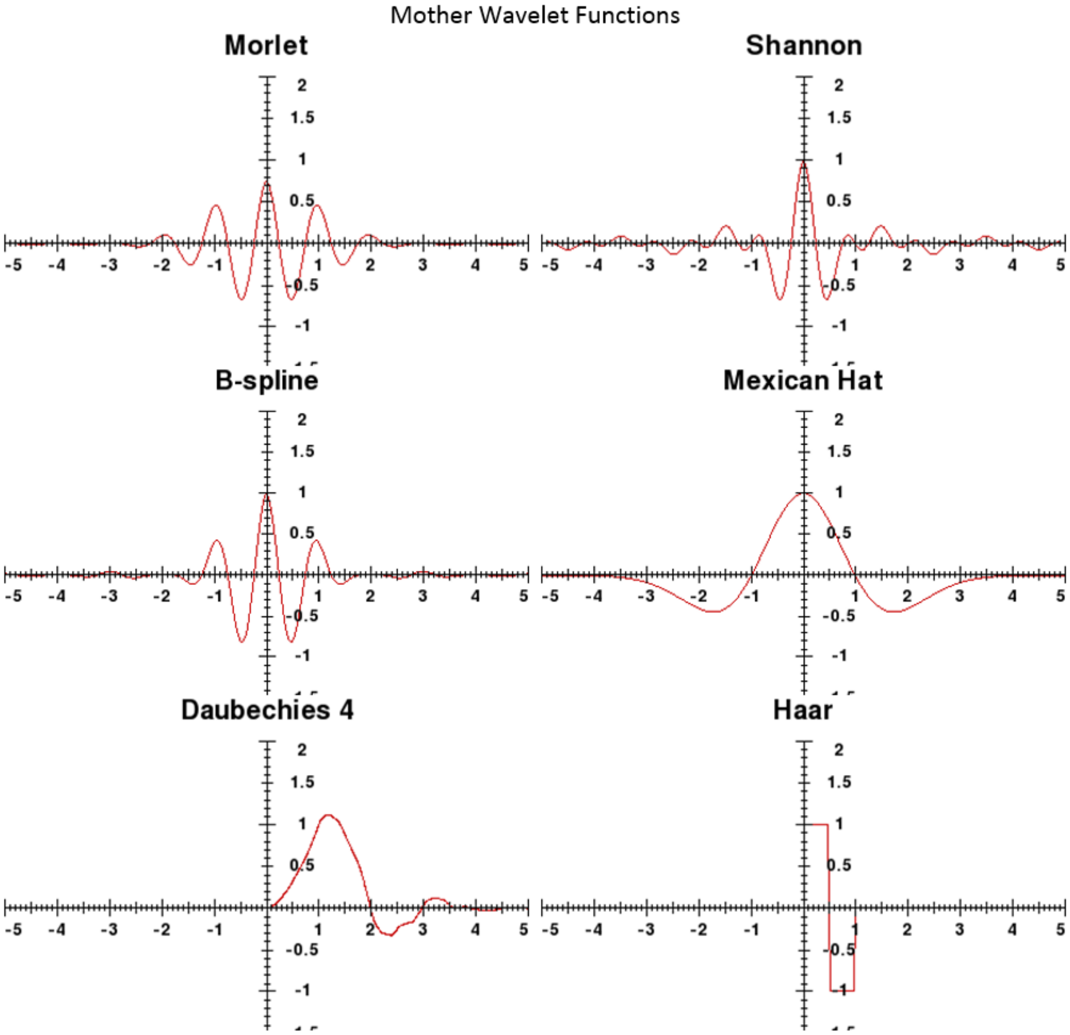


**Figure** 1.4: Examples of six different mother wavelets, where the horizontal axis is time and the vertical axis is the value of the corresponding mother wavelet at that time. The different shapes of the mother wavelets determine what type of signal they can be used to analyze most effectively.

The complex Morlet mother wavelet is defined as:

$$\varphi(t) = \pi^{-\frac{1}{4}} \exp\left(2\pi i t - \frac{t^2}{2}\right) \tag{1.4}$$

This is simply the multiplication of a complex oscillation with a frequency of 1 by a Gaussian curve centered at 0, with standard deviation 1 and height $\pi^{-\frac{1}{4}}$.

The parameters of equation (1.1), scale and time displacement, adjust the size and position of the mother wavelet curve. For example, a higher scale will correspond to a wider but shorter mother wavelet, whereas a lower scale results in a taller and narrower mother wavelet. The time displacement determines the center of the mother wavelet. The results of a continuous wavelet analysis are commonly represented in the form of a scalogram, a 3-dimensional graph in which time displacement is shown on the horizontal axis, scale on the vertical axis, and the wavelet coefficients are represented either on a color scale or by equal-value contour lines.

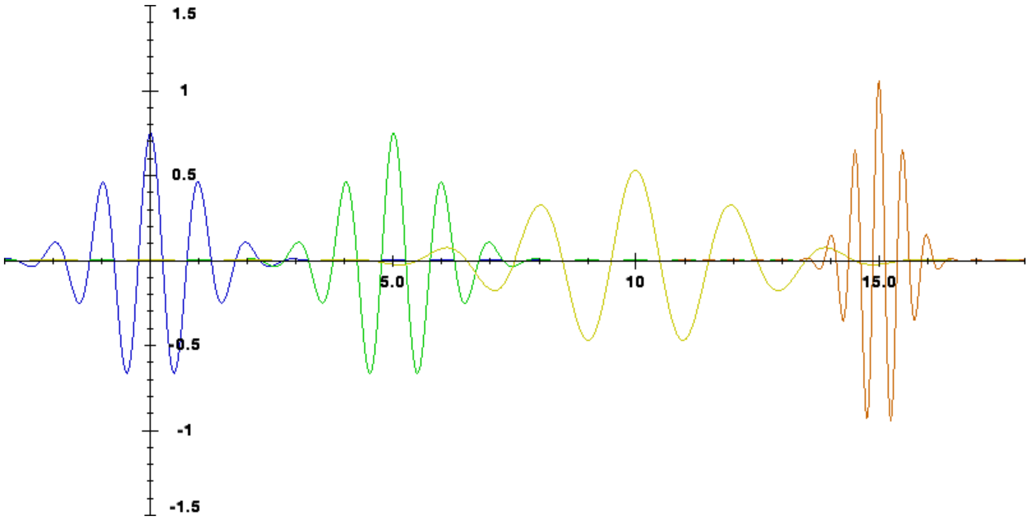


**Figure** 1.5: The real part of a Morlet mother wavelet, at different scales and time displacements. The horizontal axis is time ($t$) and the vertical axis is $\frac{1}{\sqrt{s}}\varphi\left(\frac{t-\tau}{s}\right)$ Blue: $s$=1, $\tau$=0, Green: $s$=1, $\tau$=5, Yellow: $s$=2, $\tau$=10, Orange: $s$=0.5, $\tau$=15

It is important to note that for the Morlet wavelet defined in equation (1.4), the scale is the inverse of the frequency, which means that it is equal to the period of one oscillation. This can be observed directly in Figure 1.5. However, this is not the general case: scale is proportional to period for all mother wavelets, but they are not necessarily identical.

When the experimental data to be analyzed is a discrete time series taken within a finite period of time, equation (1.1) can be modified to:

$$W_\varphi(s, \tau, \vec{x}) = \frac{\Delta t}{\sqrt{s}} \sum_k \overline{\varphi\left(\frac{k\Delta t - \tau}{s}\right)} x_k \tag{1.5}$$

where $\Delta t$ is the sampling time of the test and $x_k$ is the $k$-th element of the vector $\vec{x}$ which contains the experimental data.

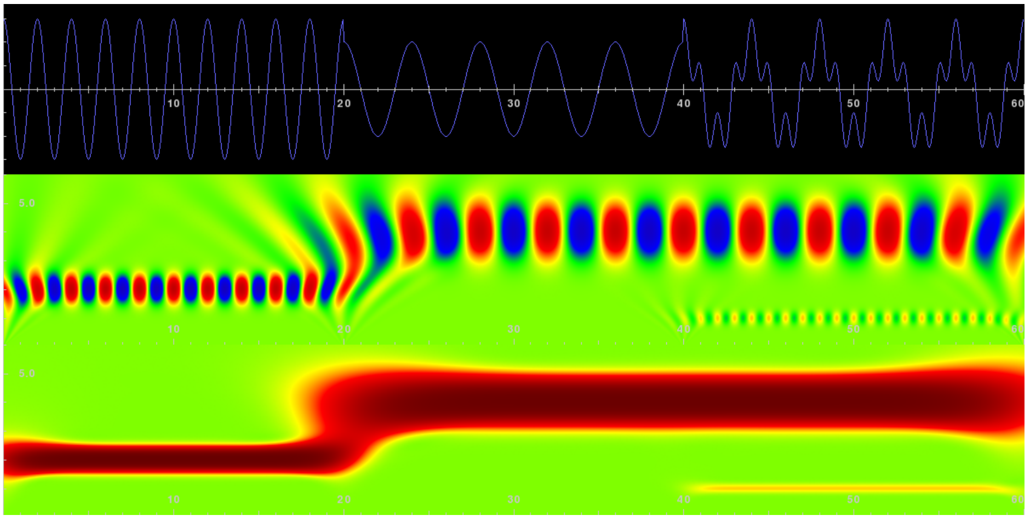


**Figure** 1.6: Top: A synthetic oscillatory function divided in three sections. For $t < 20$, $f(t) = 3\cos(\pi t)$, for $20 \leq t < 40$, $f(t) = 2\cos\left(\frac{\pi t}{2}\right)$, and for $t > 40$, $f(t) = 2\cos\left(\frac{\pi t}{2}\right) + \cos(2\pi t)$. Middle: The real part of a wavelet scalogram for the function $f(t)$, using the Morlet mother wavelet. The horizontal axis represents time displacement and the vertical axis is scale. The color scale shows the wavelet coefficient at each time displacement and scale, where red is high, blue is low, and green is zero. Note that red circles correspond exactly to the peaks in $f$, and blue circles to the low points. In the first 20 seconds, a single low scale oscillation is visible. In the following 20 seconds, there is a higher-scale oscillation, and in the final 20 seconds, the signal is decomposed into two different scales: one high and one low. Bottom: The norm of the wavelet scalogram for the function $f(t)$. Here, all information about the phase of the oscillation is lost, thus focusing on the amplitude.

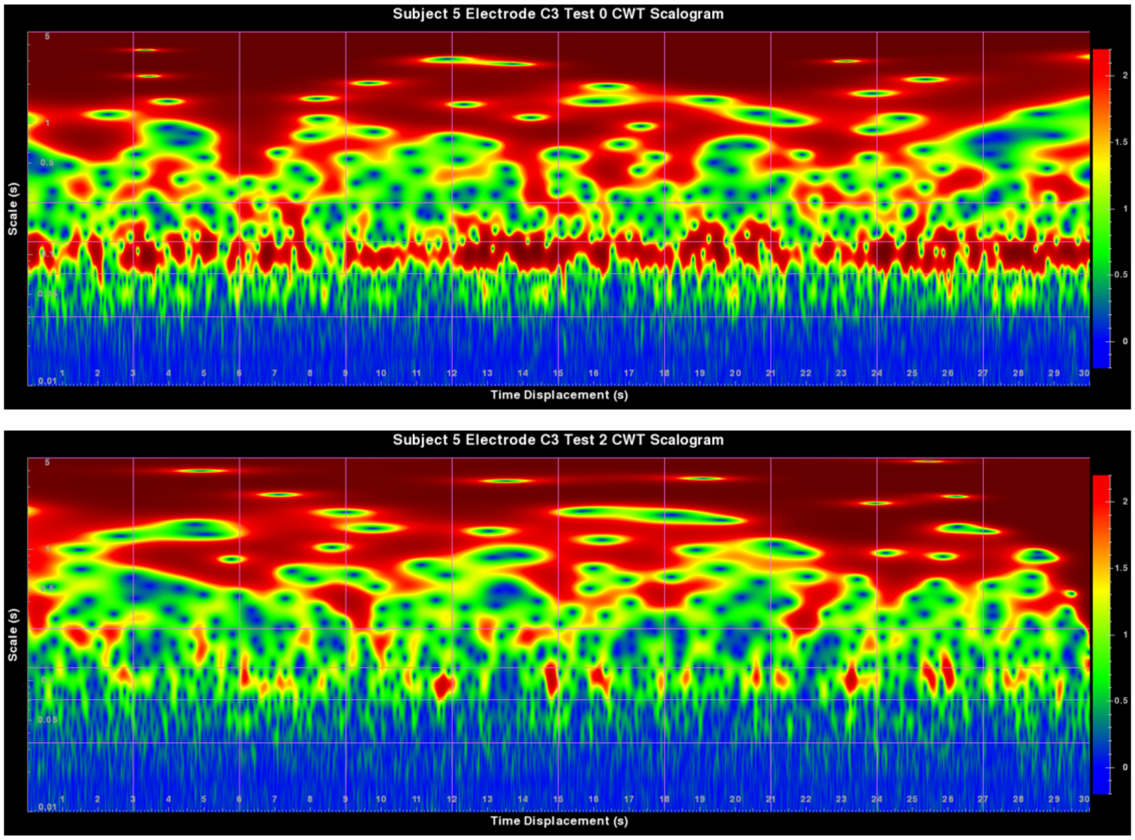


**Figure** 1.7: Examples of the norm of continuous wavelet transform scalograms for actual test data. The vertical axis is logarithmic and is separated into the traditional frequency bands $\delta$, $\theta$, $\alpha$, $\beta$ and $\gamma$ from top to bottom. The vertical lines separate the graph into 3-second intervals. In the first scalogram, the subject's eyes are closed, which can be easily identified by the high alpha component, visible as a red band in the middle of the graph.

## 1.3 BCI

A Brain-Computer Interface (BCI) is a system which receives signals originating in the brain as input and interprets them to provide commands to a computer as output, without using the brain's standard pathways for interacting with the environment, such as muscle activation. BCI has a wide range of applications. For example, if the brainwaves corresponding to hand and arm movements could be correctly decoded, a person who has lost an arm could potentially receive an intuitively-controlled prosthetic that responds to the same mental stimuli as a real arm, allowing it to flex, extend, rotate and grasp like a natural arm without having to relearn how to use it. Similarly, mechanical arms could be used to handle hazardous materials or manipulate dangerous environments, granting technicians all the dexterity of their real hands without placing them in physical danger. In a more extreme example, a quadriplegic or paraplegic patient who has suffered spinal injury, severing the pathways from the brain to muscle could potentially obtain both a BCI to decode mental commands and a neuronal stimulator capable of translating the mental commands back into electrical signals to activate motor neurons, thus returning control to the paralyzed portion of the body. A final application of BCI is in virtual reality, in which an animated character in a computer-generated environment could be controlled in the same way a real body is controlled, and could potentially be trained to perform tasks that a real body cannot, or control appendages that humans lack, like wings, tails, tentacles, or extra arms.

## 1.4 Alternatives to EEG

Aside from EEG, there are a number of other technologies that can be used to gather data for a BCI, each with their own advantages and disadvantages, including spatial and temporal resolutions, as well as size and cost.

Functional Magnetic Resonance Imaging (fMRI) allows the user to view, in 3D, the Blood Oxygen Level Dependent (BOLD) response of the entire brain. When part of the brain is more active than usual, energy from adenosine triphosphate molecules is consumed in order to return the activated neurons to their resting states of -65 mV. These ATP molecules must then be replenished, resulting in an increase of blood flow to the activated brain region. The fMRI detects concentrations of oxygenated and deoxy hemoglobin in the brain and compares the image of activated regions with a reference image. It assumes that differences are due to blood flow and not rapidly deforming brain tissue, and superimposes colorful image of neuronal activity on a grayscale anatomical image of the brain to produce its characteristic images.

Compared to EEG, fMRI provides much higher spatial resolution (as small as 0.33 mm vs 20mm for EEG) and provides information of the entire brain as opposed to only the surface of the cerebral cortex. However, the time resolution is much worse (2-3 seconds compared to .001 s or faster with EEG), and presents a lag with respect to the actual neuronal activity. Furthermore, the machinery required for fMRI is enormous and costs millions of dollars, and data analysis takes hours, making it impractical for casual BCI users.

Another technique used for gathering data on brainwaves is Magnetoencephalography (MEG), which is very similar to EEG, but collects information on the accumulated magnetic field produced by the neurons on the cerebral cortex rather than the electrical field. The time resolution for MEG

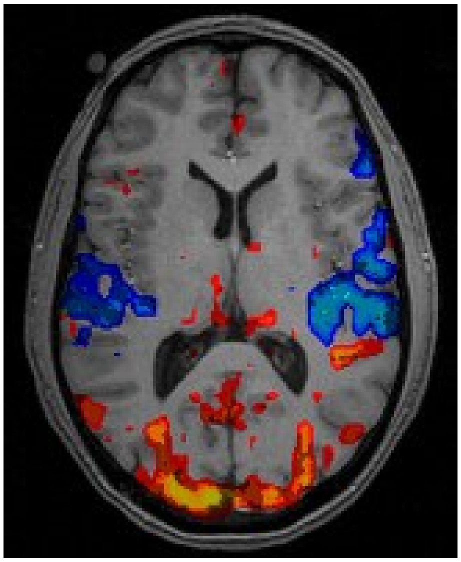


**Figure** 1.8: Example fMRI Scan. Image taken from `http://psychcentral.com/lib/what-is-functional-magnetic-resonance-imaging-fmri/`.

is comparable to that of EEG and the spatial resolution is slightly better (10 mm). Additionally, MEG has the advantage that the brain, skull, scalp and hair have different electrical permeability but roughly the same magnetic permeability, reducing the noise and making it easier to model the magnetic field than the electrical field. However, like the fMRI, the size and cost of MEG machines also makes them prohibitive.

Functional Near Infrared Spectroscopy (fNIRS) is a method by which infrared lasers are used to determine the BOLD response. Skin and bone are mostly transparent to light in the range of 650-950 nm, but hemoglobin is not. Furthermore, oxy- and deoxy-hemoglobin have different absorption spectra for all wavelengths besides 810 nm. Thus, by placing an infrared lasers near the scalp and measuring the spectra of absorbed light, the relative concentrations of oxy- and deoxy-hemoglobin can be determined, and therefore the amount of activity in the sampled part of the brain. Like EEG, fNIRS is non-invasive and harmless to the subject. It is also relatively cheap and portable, compared to other methods. Also like EEG, a signal can only be obtained from the outermost 5-10 mm of brain tissue. The temporal resolution can be above 100 Hz, but there is a delay between actual cerebral activity and the corresponding BOLD response.

The final option for obtaining functional brain data is Electrocorticography (ECoG). The functioning of ECoG is very similar to that of EEG, and this technology could, in fact, be considered a subset or specialization of EEG. The difference is that rather than placing electrodes on the scalp, an array of electrodes is implanted directly on the surface of the cerebral cortex. Whereas EEG records the sum of the activity of millions of neurons, ECoG can record the activity of individual cortical neurons, providing unparalleled spatial and temporal resolution. The drawback of ECoG is that it requires removal of a portion of the skull and implantation of an electrode array on the brains surface, making it an unattractive option except in extreme circumstances.

We decided to use EEG because it is a non-invasive and relatively cheap and portable option with high temporal resolution, which is essential for real-time control. One of the goals of this thesis is to overcome EEG's inherent noisiness and low spatial resolution by way of solid, innovative mathematical methods to allow its use in an effective BCI.

# Chapter 2

# Methods

## 2.1 Equipment

Data was collected using a Neuroscan Synamps EEG with a 32-electrode cap in the standard 10-20 configuration at a sampling rate of 1 kHz.

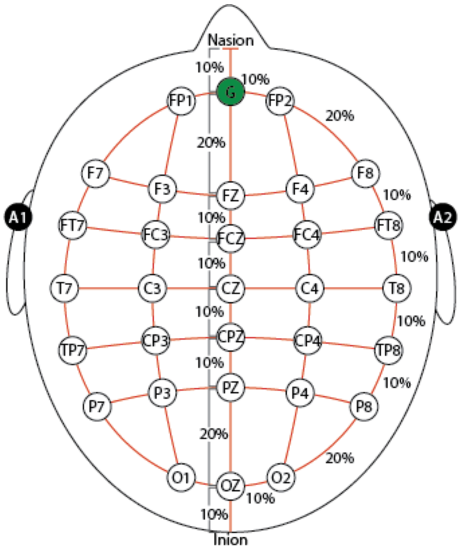


**Figure** 2.1: Diagram of positions of 32 electrodes plus ground in standard 10-20 configuration. G represents the ground and A1 and A2 (placed on the earlobes) are used as reference points.

Aquagel conductive gel was used to improve conductivity between the scalp and electrodes. Neuroscan SCAN 4.3 software was used for data acquisition during experimentation.

## 2.2 Experimental Procedure

### 2.2.1 Preparation

All subjects were right-handed male students from the Facultad de Ciencias Fisico-matemáticas (FCFM) of the Benemérita Universidad Autónoma de Puebla (BUAP). The experiment was carried out in accordance with the Declaration of Helsinki. Subjects were provided with a brief description of the experiment and signed a letter of informed consent before participating. On the day of the experiment, subjects were asked about their recent drug consumption (including alcohol, caffeine and pharmaceuticals) as well as mental wellness and sleep habits in order to control for mind-altering affects. Data collection was carried out in the Laboratorio de Neurofisiología Integrativa in the Instituto de Fisiología of the BUAP and in the Laboratorio de Psicofisiología Experimental of the Facultad de Psicología of the BUAP.

Before the experiment began, the subject removed all metallic objects from his person and sat in a chair with his feet planted firmly on the floor.

The distance from the inion (located at the external occipital protuberance of the skull) to the nasion (at the intersection of the nasal and the frontal bones) was measured over the top of the head. Electrodes OZ and G were placed at 10% of this distance towards the top of the head from the Inion and Nasion, respectively. Electrodes PZ, CPZ, CZ, FCZ and FZ were placed along the center of the head at 20%, 30%, 40%, 50% and 60% of this distance from OZ, respectively. A second distance was measured around the head from OZ to G, and electrodes O2, P8, TP8, T8, FT8, F8 and FP2 were placed at 10%, 20%, 30%, 40%, 50%, 60%, 70% and 90% of this second distance from OZ. The other electrodes were placed using the aforementioned electrodes as guidelines, as shown in Figure 2.1.

Once the electrodes were properly placed, Aquagel conductive gel was injected into each electrode and the impedance was measured using Neuroscan SCAN 4.3 software. In order to ensure accurate electrode readings, additional conductive gel was injected and electrodes were gently rubbed to improve contact with the scalp until all electrodes showed impedances of 5 k$\Omega$ or lower. The signals were filtered at DC-500 Hz bandpass with a notch filter at 60 Hz to remove noise produced by the power line.

### 2.2.2 Experimental Protocol

Nine healthy right-handed male subjects between 20 and 29 years of age performed twelve 30-second tests, each composed of 5 repetitions of two opposite movements. Each movement lasted 2.5 seconds with a half second rest between one movement and the next. Before each test, the subject read instructions on a computer screen indicating what movement would be made. In addition, the subject received oral instructions and a demonstration and was allowed to practice the movement. The subject was asked to perform the movement at a continuous speed during each 2.5 second movement period. During the test, a model hand appeared on the screen and carried out the two opposite movements while the subject followed its indications.

### 2.2.3 Description of Tests

1. **Eyes Closed Resting State:** The subject relaxed with eyes closed for 30 seconds.

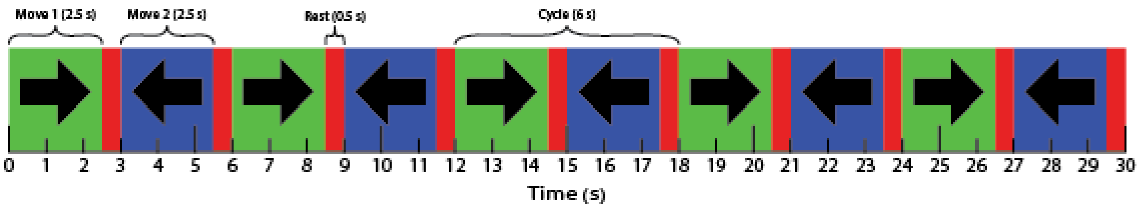


**Figure** 2.2: Diagram of movement timing during a test. Right and Left arrows are indicators of the movement being performed at each time during test 3. For all other tests, timing was the same, but the movements varied, as described in section 2.2.3.

2. **Eyes Open Observation:** The subject relaxed and watched a hand move horizontally across the screen, beginning on the left.

3. **Horizontal Movement:** The subject extended his right arm and pointed it forward parallel to the ground with hand open and thumb up. The subject moved his arm horizontally from the shoulder at constant speed from 45 to 135 with respect to his chest. Movement speed and timing were controlled by indications on a computer screen. The arm began on the left.

4. **Horizontal Imagination:** The subject rested his arm and imagined that he was performing the same action as in the previous test while observing the hand movement on the screen.

5. **Vertical Movement:** The subject extended his right arm and pointed it forward parallel to the ground with hand open and thumb up. The subject moved his arm vertically from the shoulder at constant speed from 135 to 45 with respect to his torso. Movement speed and timing were controlled by indications on a computer screen. The arm began at the top.

6. **Vertical Imagination:** The subject rested his arm and imagined that he was performing the same action as in the previous test while observing the hand movement on the screen.

7. **Rotation:** The subject extended his right arm and pointed it forward parallel to the ground with hand open and thumb up. The subject rotated his arm from the shoulder at constant speed from 0° to 180° with respect to the initial position. Rotation speed and timing were controlled by indications on a computer screen. The arm began with the thumb pointing up.

8. **Rotation Imagination:** The subject rested his arm and imagined that he was performing the same action as in the previous test while observing the hand movement on the screen.

9. **Finger Movement:** The subject extended his right arm and pointed it forward parallel to the ground with hand open and thumb up. The subject closed and opened his four fingers towards the palm at constant speed. Movement speed and timing were controlled by indications on a computer screen. The hand began open.

10. **Finger Imagination:** The subject rested his arm and imagined that he was performing the same action as in the previous test while observing the hand movement on the screen.

11. **Elbow Movement:** The subject extended his right arm and pointed it forward parallel to the ground with hand open and thumb up. The subject flexed and extended his arm at the elbow at constant speed from 90° to 0° with respect to the floor. Movement speed and timing were controlled by indications on a computer screen. The hand began pointing up.

12. **Elbow Imagination:** The subject rested his arm and imagined that he was performing the same action as in the previous test while observing the hand movement on the screen.

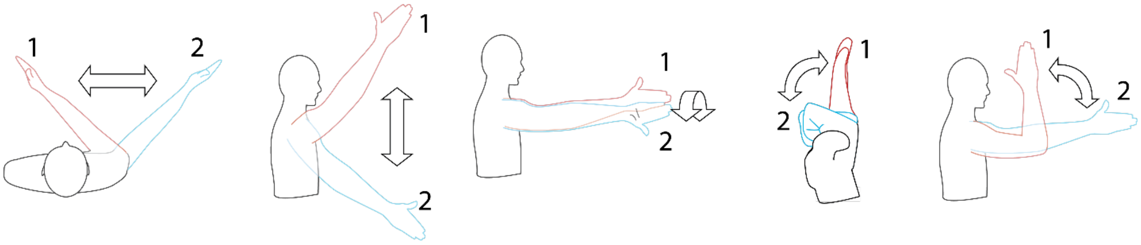


**Figure** 2.3: Diagram of the real and imaginary movements carried out in the experiments. The part in red represents the initial position and the blue the final position.

## 2.3 Control Group

The method for data analysis used in this thesis was designed to find patterns. Ideally, the patterns identified should correspond to the movements performed. However, it may also find patterns that are coincidental and unrelated to the movements. It was therefore essential to include a control group in which there was no difference between the tests. By evaluating both the experimental and the control groups, we could identify how much of the success rate was due to the recognition of patterns caused by movements and how much was due to the coincidental patterns. Therefore, in addition to the experimental group, we included a control group of two right-handed male subjects. For this group, all twelve tests were Eyes Closed Resting State. The control subjects sat with their eyes closed and performed no movements during 30 seconds for all twelve tests.

## 2.4 Data Analysis

### 2.4.1 Problem Statement

EEG experimental data was exported in the form of matrices, the elements of which were voltages acquired from each electrode at each time. One such matrix was obtained for every subject and test. Each column of each matrix contained the readings of a single electrode and each row contained the readings at a single time. Hereafter, we will refer to the column vectors of the matrix for subject $S$, electrode $\varepsilon$, test $T$ as:

$$\vec{X}_{S:\varepsilon:T} \tag{2.1}$$

and the $k$-th element of each column vector as:

$$X_{S:\varepsilon:T,k} \tag{2.2}$$

The data matrices were first processed to obtain feature vectors for each discrete time to provide some insight into what cortical activity was taking place at that time. The goal of our data analysis was to create a master classifier function which would take the feature vector corresponding to any given time as input and identify what movement the subject was carrying out at that time.

## 2.4.2 Feature Vector Creation

### 2.4.2.1 Normalization

The first step taken in preprocessing the experimental data was to normalize it. This step was not necessary in order to create a classifier for the data, but it did allow us to compare data from different subjects. Because the voltage detected by any given electrode on a particular subject depended on factors such as the amount of hair present, conductive gel used, and other external factors, both the baseline voltage and the amplitude of oscillations could vary greatly from subject to subject. For this reason, the experimental data was preprocessed to create normalized data with median 0 and standard deviation 1.

Given a n-dimensional vector $\vec{x}$ with components $x_i$, the median and variance of the components are defined as:

$$\mu = \frac{1}{n}\sum_{i}^{n} x_i \tag{2.3}$$

$$\sigma^2 = \frac{1}{n}\sum_{i}^{n} (x_i - \mu)^2 \tag{2.4}$$

respectively. In order to create a normalized vector $\vec{x}'$ such that $\mu' = 0$ and $\sigma' = 1$, we defined

$$x'_i = \frac{x_i - \mu}{\sigma} \tag{2.5}$$

The proof that $\mu' = 0$ and $\sigma' = 1$ is straightforward and is provided below:
Proof:

$$\mu' = \frac{1}{n}\sum_{i}^{n} x'_i = \frac{1}{n}\sum_{i}^{n} \frac{x_i - \mu}{\sigma} = \frac{1}{n\sigma}\sum_{i}^{n}(x_i) - \frac{n\mu}{n\sigma} = \frac{\mu}{\sigma} - \frac{\mu}{\sigma} = 0 \tag{2.6}$$

$$\sigma'^2 = \frac{1}{n}\sum_{i}^{n}(x'_i - \mu')^2 = \frac{1}{n}\sum_{i}^{n}\left(\frac{x_i - \mu}{\sigma}\right)^2 = \frac{1}{\sigma^2 n}\sum_{i}^{n}(x_i - \mu)^2 = \frac{\sigma^2}{\sigma^2} = 1 \tag{2.7}$$

The data was normalized by calculating the median and standard deviation of the data corresponding to test 1 (Eyes Closed Resting State) of each subject and electrode.

$$\mu_{S:\varepsilon} = \frac{1}{n}\sum_{k}^{n} x_{S:\epsilon:1,k} \tag{2.8}$$

$$\sigma_{S:\varepsilon} = \sqrt{\frac{1}{n}\sum_{k}^{n}\left(x_{S:\varepsilon:1,k} - \mu\right)^2} \tag{2.9}$$

Then, normalized vectors were created for every subject, electrode and test using the following formula:

$$x'_{S:\varepsilon:T,k} = \frac{x_{S:\varepsilon:T,k} - \mu_{S:\varepsilon}}{\sigma_{S:\varepsilon}} \tag{2.10}$$

Because test 1 was used to determine the offset and scale factors for all the tests, only test 1 for each electrode and subject actually ended up with a median of zero and standard deviation of 1. We assume that the impendence for each electrode remained roughly the same for one subject over the course of an experiment.

#### 2.4.2.2 Wavelet Transform

Using equation (1.5) as a basis, we define the wavelet transform for the data set $\vec{x}_{S:\varepsilon:T}$ at time displacement $\tau$ and scale $s$ as:

$$W(s, \tau, S, \varepsilon, T) = \frac{\Delta t}{\sqrt{s}} \sum_k \overline{\varphi\left(\frac{k\Delta t - \tau}{s}\right)} x'_{S:\varepsilon:T,k} \tag{2.11}$$

where $\Delta t = 0.001s$ is the time between samples (the inverse of the sampling rate of 1000 Hz) and

$$\varphi(t) = \pi^{-\frac{1}{4}} \exp\left(2\pi i t - \frac{t^2}{2}\right) \tag{2.12}$$

is the Morlet mother wavelet.

The analysis began by performing the aforementioned wavelet transform on the normalized EEG data for each subject, test and electrode. We used 1500 time displacements distributed evenly in the range of 0 to 30 s and 4 scales distributed logarithmically between .01925 and 0.18257 s, corresponding approximately to the centers of four of the five frequency bands commonly used when analyzing brainwaves, namely, $\delta$ (0.2 - 3.5 Hz), $\theta$ (4 - 7.5 Hz), $\alpha$ (8 - 13 Hz), $\beta$ (14 - 30 Hz) and $\gamma$ (30 - 90 Hz) [11].

| Frequency Band | Frequency Band | Frequency Used | Scale Used |
|---|---|---|---|
| Theta ($\theta$) | 4 - 7.5 Hz | 5.477 Hz | 0.18257 s |
| Alpha ($\alpha$) | 8 - 13 Hz | 11.59 Hz | 0.08625 s |
| Beta ($\beta$) | 14 - 30 Hz | 24.54 Hz | 0.04075 s |
| Gamma ($\gamma$) | 30 - 90 Hz | 51.95 Hz | 0.01925 s |

**Table** 2.1: The four scales chosen as parameters for the wavelet transform, and their corresponding frequencies and frequency bands. Because the scales are distributed logarithmically, each scale is 2.117 times the next one.

No frequency from the delta band was used because its use would be impractical for real-time wavelet analysis. This is discussed further is section 4.2.

### 2.4.3 Vector Concatenation

Definition: Vector Concatenation Operator.

In this thesis, we will use the $\oplus$ operator to indicate vector concatenation.
Let $\vec{x} = (x_1, ..., x_n)$ and $\vec{y} = (y_1, ..., y_m)$ be two vectors.

$$\vec{x} \oplus \vec{y} \equiv (x_1, ..., x_n, y_1, ..., y_m) \tag{2.13}$$

For simplicity, we will also define vector concatenation between two scalars $a$ and $b$ as:

$$a \oplus b \equiv (a, b) \tag{2.14}$$

And between a vector $\vec{x} = (x_1, ..., x_n)$ and a scalar $a$ as:

$$\vec{x} \oplus a \equiv (x_1, ..., x_n, a) \tag{2.15}$$

$$a \oplus \vec{x} \equiv (a, x_1, ..., x_n) \tag{2.16}$$

We will also define the large operator

$$\bigoplus_{(i=1)}^{n} a_i = a_1 \oplus a_2 \oplus ... \oplus a_n \tag{2.17}$$

where $a_i$ could be either vectors or scalars, or the abbreviated form

$$\bigoplus_{i} a_i \tag{2.18}$$

for which the possible values of i are assumed to be known and ordered.

For every subject, test and time displacement, feature vector was created by concatenating the absolute values of the wavelet coefficients for each scale and each electrode. Because 30 electrodes and 4 frequencies were used, this resulted in feature vectors of length 120.

This is to say, for subject $S$, test $T$, and time displacement $\tau$, the feature vector $\overrightarrow{A_{S:T,\tau}}$ was constructed in the following way:

$$\overrightarrow{A_{S:T,\tau}} = \bigoplus_{\varepsilon} \bigoplus_{s} |W(s, \tau, S, \varepsilon, T)| \tag{2.19}$$

For $s \in (0.18257, 0.08625, 0.04075, 0.01925)$ and $\epsilon \in (O2, O1, OZ, PZ, P4, CP4, P8, C4, TP8, T8, P7, P3, CP3, CPZ, CZ, FC4, FT8, TP7, C3, FCZ, FZ, F4, F8, T7, FT7, FC3, F3, FP2, F7, FP1)$
Thus:

$$\begin{aligned} \overrightarrow{A_{S:T,\tau}} \quad = \quad & (|W(0.18257, \tau, S, O2, T)|, |W(0.08625, \tau, S, O2, T)|, |W(0.04075, \tau, S, O2, T)|, \\ & |W(0.01925, \tau, S, O2, T)|, |(W(0.18257, \tau, S, O1, T)|, ..., |W(0.01925, \tau, S, FP1, T)|) \end{aligned} \tag{2.20}$$

The complete list of the scales and electrodes for every component of $\overrightarrow{A_{S:T,\tau}}$ can be found in Appendix 3.

#### 2.4.3.1 Feature Vector Grouping

The complete feature vectors were subsequently grouped into sets corresponding to the same movement.

Let $\alpha$ be an ordered set of different states between which we wish to discriminate. Let $\beta$ be an ordered set of sets of feature vectors, such that all the feature vectors in $\beta_i$ correspond to the state $\alpha_i$.

In this thesis, the values of $\alpha_i$ are:
$\alpha_1$= RestClosed, $\alpha_2$= WatchRight, $\alpha_3$= WatchLeft, $\alpha_4$= Right, $\alpha_5$= Left, $\alpha_6$= RightIm, $\alpha_7$= LeftIm, $\alpha_8$= Down, $\alpha_9$= Up, $\alpha_{10}$= DownIm, $\alpha_{11}$= UpIm, $\alpha_{12}$= RotCW, $\alpha_{13}$= RotCCW, $\alpha_{14}$=

RotCWIm, $\alpha_{15}$= RotCCWIm, $\alpha_{16}$= CloseFing, $\alpha_{17}$= OpenFing, $\alpha_{18}$= CloseFingIm, $\alpha_{19}$= OpenFingIm, $\alpha_{20}$= OpenElb, $\alpha_{21}$= CloseElb, $\alpha_{22}$= OpenElbIm, $\alpha_{23}$= CloseElbIm. We used the following algorithm to assign each vector to a set $\beta_i$:

$$
\begin{gathered}
if\ T = 1 \Rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_1 \\
if\ T = 2 \Rightarrow \begin{cases} if\ 0 \leq \tau\%6 \leq 2.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_2 \\ if\ 3 \leq \tau\%6 \leq 5.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_3 \end{cases} \\
if\ \ T = 3 \Rightarrow \begin{cases} if\ 0 \leq \tau\%6 \leq 2.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_4 \\ if\ 3 \leq \tau\%6 \leq 5.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_5 \end{cases} \\
if\ \ T = 4 \Rightarrow \begin{cases} if\ 0 \leq \tau\%6 \leq 2.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_6 \\ if\ 3 \leq \tau\%6 \leq 5.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_7 \end{cases} \\
if\ \ T = 5 \Rightarrow \begin{cases} if\ 0 \leq \tau\%6 \leq 2.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_8 \\ if\ 3 \leq \tau\%6 \leq 5.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_9 \end{cases} \\
if\ \ T = 6 \Rightarrow \begin{cases} if\ 0 \leq \tau\%6 \leq 2.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{10} \\ if\ 3 \leq \tau\%6 \leq 5.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{11} \end{cases} \\
if\ \ T = 7 \Rightarrow \begin{cases} if\ 0 \leq \tau\%6 \leq 2.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{12} \\ if\ 3 \leq \tau\%6 \leq 5.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{13} \end{cases} \\
if\ \ T = 8 \Rightarrow \begin{cases} if\ 0 \leq \tau\%6 \leq 2.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{14} \\ if\ 3 \leq \tau\%6 \leq 5.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{15} \end{cases} \\
if\ \ T = 9 \Rightarrow \begin{cases} if\ 0 \leq \tau\%6 \leq 2.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{16} \\ if\ 3 \leq \tau\%6 \leq 5.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{17} \end{cases} \\
if\ \ T = 10 \Rightarrow \begin{cases} if\ 0 \leq \tau\%6 \leq 2.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{18} \\ if\ 3 \leq \tau\%6 \leq 5.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{19} \end{cases} \\
if\ \ T = 11 \Rightarrow \begin{cases} if\ 0 \leq \tau\%6 \leq 2.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{20} \\ if\ 3 \leq \tau\%6 \leq 5.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{21} \end{cases} \\
if\ \ T = 12 \Rightarrow \begin{cases} if\ 0 \leq \tau\%6 \leq 2.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{22} \\ if\ 3 \leq \tau\%6 \leq 5.5 \rightarrow \overrightarrow{A_{S:T,\tau}} \in \beta_{23} \end{cases}
\end{gathered}
\tag{2.21}
$$

where % indicates the modulo operator, or remainder after division.

### 2.4.4 Feature Vector Reduction

Using large feature vectors has a number of disadvantages. The most obvious is that it requires more computational resources to analyze than a smaller feature vector. The advantage is that the more factors there are available, the easier it should be to find a pattern in them. Although this is true, it is not always an advantage, as it can lead to overfitting, or finding solutions which apply astonishingly well to the training data, but do not work for new data.

We therefore reduced our feature vectors by selecting the components which individually were most efficient at separating vectors from two sets $\beta_i$ and $\beta_j$.

Definition: Separation Index:

$$\chi_{ij} = \frac{|\mu_i - \mu_j|}{\sigma_i + \sigma_j} \tag{2.22}$$

where $\mu_{i,j}$ and $\sigma_{i,j}$ are the mean and standard deviation of a series of real numbers $\gamma_{i,j}$, respectively. In effect, $\chi_{ij}$ is the distance between the averages of two distributions of numbers, as a multiple of the sum of their standard deviations. The separation index remains invariant if a non-zero constant is added to or multiplied by every number in $\gamma_i$ and $\gamma_j$.

In order to create a reduced feature vector, the separation index of each component of the original feature vectors was tested in turn by defining $\gamma_i^l = \{A_l\}_{\vec{A}\in\beta_i}$ and $\gamma_j^l = \{A_l\}_{\vec{A}\in\beta_j}$ where l was the index of the component being tested. After determining the separation index of each component $\chi_{ij}^l$, they were then ordered from highest to lowest and a list $L$ was created, containing the $m$ component indices with the highest separation indices. Then, for every feature vector $\vec{A}$, a reduced feature vector $\vec{Z}$ was created

$$\vec{Z} = \bigoplus_{l\in L} A_l \tag{2.23}$$

### 2.4.5 Binary Classifier Creation

The goal of our analysis was to create a function capable of taking any feature vector as input and determining the movement to which that feature vector corresponds. That is to say, at any given time during any given test, the subject was performing a specific movement. There is also a feature vector associated with that time and test. The classifier function should guess the correct movement for each feature vector as often as possible.

Let $\vec{A} = (A_1, A_2, , A_n)$ be a feature vector.
Our goal was to find a master classifier $\psi : R^n \to \alpha$ which maximized the number of vectors $\vec{A}$ that fulfill the relationship $\psi(\vec{A}) = \alpha_i \Leftrightarrow \vec{A} \in \beta_i$.
In order to define $\psi$, it was first necessary to generate a series of binary classifier functions $f_{ij} : R^n \to \{\alpha_i, \alpha_j\}$ which discriminated between two particular states $\alpha_i$ and $\alpha_j$.
In order to do this, we first created discriminating functions $g_{ij} : R^n \to R$ and then defined

$$f_{ij}(\vec{A}) = \begin{cases} \alpha_i & if\ g_{ij}(\vec{A}) < 0 \\ \alpha_j & otherwise \end{cases} \tag{2.24}$$

Given two sets of state vectors, $\beta_i$ and $\beta_j$, it was relatively easy to generate a linear discriminating function $g_{ij}$ using minimum squares, as follows:

$$g_{ij}(\vec{A}_k) = \sum_{l=1}^{m} C_l Z_{kl} \tag{2.25}$$

where $C_l$ are unknown constants to be determined.

We defined a vector $Y$ containing information about whether each feature vector belonged to

$\beta_i$ or $\beta_j$:

$$Y_k = \begin{cases} -1 & if\ \vec{A}_k \in \beta_i \\ 1 & if\ \vec{A}_k \in \beta_j \end{cases} \tag{2.26}$$

Determining the constants $C$ could then be done relatively easily using minimum squares:

$$C = (Z^T Z)^{-1} Z^T Y \tag{2.27}$$

### 2.4.6 Master Classifier Creation

The method described in the previous section was used for creating binary classifiers. However, the goal was to create a master classifier which would discriminate between all 23 movements. There are two common methods for doing this: One vs All and One vs One. To exemplify these two methods, let us assume we want to discriminate between four states: A, B, C, and D.

The One vs All method would involve creating four binary classifiers: each of which groups all states except one into a single set and compares against the one state which was excluded. Namely, the four classifiers would be: A vs B+C+D, B vs A+C+D, C vs A+B+D and D vs A+B+C. Then, in order to determine the most probable state for any given feature vector, the feature vector is evaluated with the four aforementioned binary classifiers. If in only one evaluation the single state was chosen and in all other evaluations the grouped states are chosen, then that single state would be determined to be the correct answer. In any other case, the result would be indecisive.

The One vs One method for the same four states would require the creation of six binary classifiers: A vs B, A vs C, A vs D, B vs C, B vs D, and C vs D. In order to determine the most probable state for any given feature vector, the feature vector would be evaluated with all six binary classifiers and a record would be kept of which state was selected in each evaluation. Because in this example we are considering 4 states, each state appears in 3 different evaluations, so if any one state were chosen 3 times, that state would be selected as the most probable. Otherwise, no movement would be selected. For our experiment, we used a One vs One system, creating 253 binary classifiers to distinguish between each pair of movements. The master classifier evaluated each feature vector using all 253 binary classifiers and kept track of the selected movement in each case. If any movement was selected 22 times, that movement was returned as the result of the master classifier. Otherwise, no movement was selected and the result “None” was returned.

## 2.5 Evaluation

A binary or master classifier can be evaluated by allowing it to determine the most likely movements for a set of vectors for which the correct movements are already known. The real movement and the guessed movement can then be compared to create a confusion matrix whose elements $M_{ij}$ are equal to the fraction of the time movement $\alpha_j$ was guessed when the real movement was $\alpha_i$.

In medicine, 2x2 confusion matrices are often used to show how successful a particular method is at diagnosing a condition. If a patient has a condition and is diagnosed with the condition, that is known as a true positive. If the patient has a condition but is diagnosed as not having it, that is a false negative. If the patient does not have the condition but is diagnosed with the condition, this is known as a False Positive, and when the patient does not have the condition and is diagnosed as not having it, it is called a True Negative.

In our case, the matrix had not two options, but 23. Additionally, the master classifier had the possibility of returning the diagnosis of "None", meaning that the confusion matrix had a total size of 24x23. In this case, using terms such as True or False Positive or Negative doesn't make much sense, but all results in the diagonal of the matrix can be considered to be "True" or "correct". For simplicity, we synthesize the information in the confusion matrix by averaging the results in the diagonal and call this the Success Rate.

### 2.5.1 Self-validation and Cross-validation

Let us denominate the set of feature vectors used to create the master classifier "training data" and those used for evaluation "test data". The relationship between the training and test data is important and the selection of each can make an enormous impact on the results. When training and test data are the same, this is known as self-validation and can give spectacular results that do not really have any statistical significance or validity.

If a set of feature vectors can be partitioned, with one partition functioning as training data and the other as test data, and the partitions can be changed so that all vectors get a turn as both training and test data, this is called cross-validation.

In this thesis, because each test consisted of five cycles, we chose to partition the data into these five sections. We then used four sections for training and one for testing, and alternated the test section until each section had been used for testing once. Thus, whenever the term "cross-validation" is used hereafter, the results refer to the average of the results of these five evaluations.

Aside from self- and cross-validation, we also performed evaluations for which the training and test groups were completely independent, where the data from one subject or group of subjects was used for training, and the data from another subject or independent group of subjects was used for testing. We call these "cross-subject evaluations".

### 2.5.2 Key Components

Aside from creating and evaluating classifiers, the secondary objective of this thesis was to identify which frequencies and brain regions showed an increase or decrease in activity when certain movements were performed. Since each component of the feature vectors contains information regarding the neural activity in a certain region and at a certain frequency, this objective is analogous to determining which components of the feature vector have the highest effect on the success rate of the evaluation.

For each pair of movements $\alpha_i$, $\alpha_j$, we created a list $L_{ij}$ of the top $m$ components using the separation index, as described in section 2.4.3 and created a reduced feature vector with these $m$ components. We then generated a binary classifier function $f_{ij}$ and evaluated its base success rate, $S_{ij}$.

We subsequently cycled through each component index $k$ of the complete feature vector, and performed the following algorithm:
If $k \in L_{ij}$, that is to say, the $k$-th component was already included in the top $m$ components as determined by separation index, then a new list of components $L_{ij}^{-k}$ was created which included all the elements of $L_{ij}$ except $k$. That is to say, $L_{ij}^{-k} = \{l \in L_{ij} | l \neq k\}$.

We then created the reduced feature vector containing only components in $L_{ij}^{-k}$, generated the binary classifier $f_{ij}^{-k}$ and evaluated its success rate, $S_{ij}^{-k}$.
Finally, we determined the effect of the inclusion of the $k$-th element in the reduced feature vector $\Delta S_{ij}^{k} = S_{ij} - S_{ij}^{-k}$.
In the case that $k \neq L_{ij}$, we created a new list of components $L_{ij}^{+k} = L_{ij} \cup \{k\}$. We then generated the binary classifier $f_{ij}^{+k}$ and evaluated its success rate, $S_{ij}^{+k}$.
In this case, $\Delta S_{ij}^{k} = S_{ij}^{+k} - S_{ij}$.
In this way, we were able to determine how much the success rate of a binary classifier was effected by including or excluding any particular component. Those components with the highest $\Delta S_{ij}^{k}$ were labelled "Key Components" in discriminating between $\alpha_i$ and $\alpha_j$.
Furthermore, by defining

$$\Delta S_i^k = \frac{1}{(p-1)} \sum_{j=1, j \neq i}^{p} \Delta S_{ij}^{k} \tag{2.28}$$

where $p$=23 is the number of different movements, we obtain a number which indicates the importance of the $k$-th component in determining $\alpha_i$, thus providing an indication of a characteristic pattern in the region and frequency associated with the $k$-th component, when performing movement $\alpha_i$.

### 2.5.3 Key Component Evaluation

After determining the key components for each binary classifier, new binary classifiers were created using reduced feature vectors that contained the key components, rather than the top components by separation index. Furthermore, this process could be iterated to continuously find new components that work well together, until there was no change from one iteration to the next. However, it is important to bear in mind that because these classifiers were created to optimize the evaluation of the test data, they needed to be evaluated with an independent data set, which was mutually exclusive from both the training and test data.

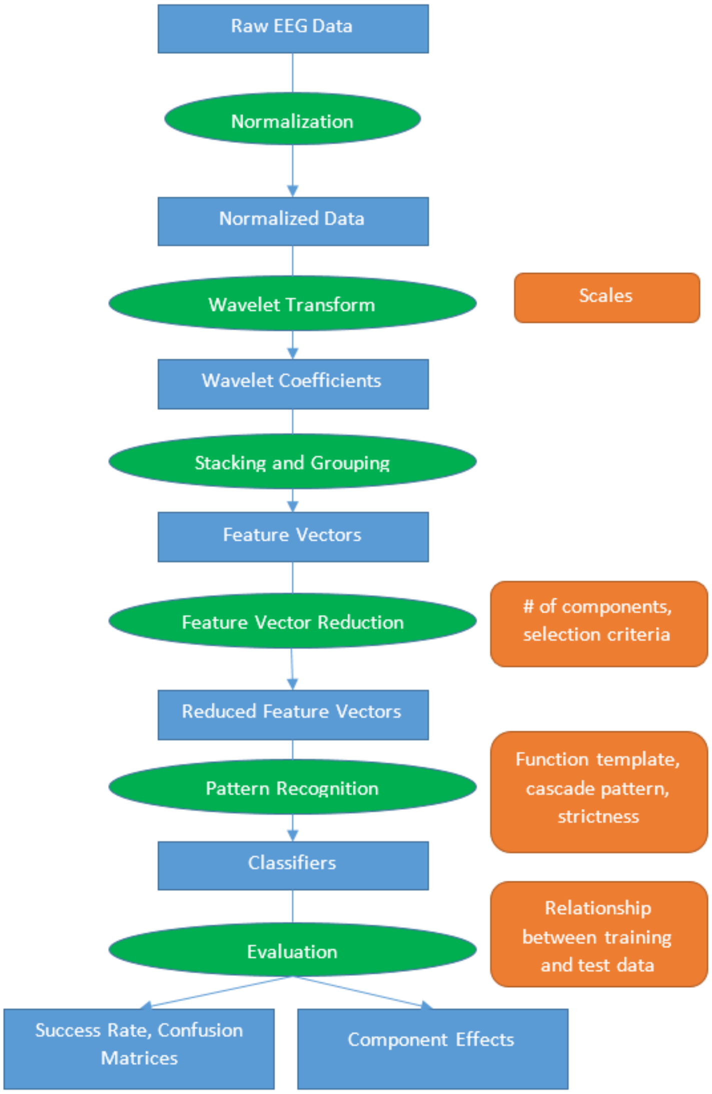
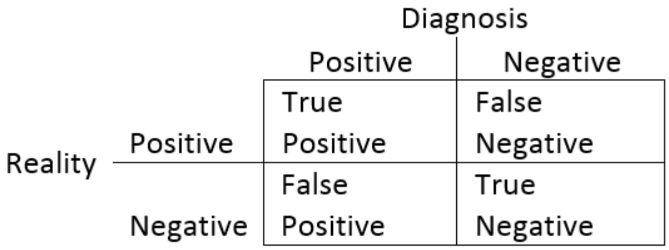


**Figure** 2.4: Data Analysis Flowchart. Blue boxes represent data, green ovals represent processes, and orange boxes represent the variables involved in each process.



| | | Diagnosis | |
|---|---|---|---|
| | | Positive | Negative |
| Reality | Positive | True Positive | False Negative |
| | Negative | False Positive | True Negative |

**Figure** 2.5: Confusion matrix for a binary classifier with only two possible outcomes: Positive or Negative.

# Chapter 3

# Results and Discussion

In this chapter, the results of different analyses are presented, and interpretations of those results are provided.

## 3.1 Parallelization and Algorithm Benchmarking

During the initial phase of this thesis, the calculation of wavelet transforms was a critical computational bottleneck, with analyses taking months to finish. Therefore, many months were dedicated to the development of code that would run in parallel to reduce analysis time from months to weeks and periodic data saving so that the progress of such long analyses would not be lost if the program were interrupted. However, in the end, the implementation of new algorithms made the effects of these efforts negligible in comparison. The development of faster algorithms proved a critical step to making the subsequent steps in the analysis possible within a reasonable time frame, and this is one of the most important contributions of this thesis. Four versions of the wavelet transform algorithm were developed, all of which gave the same results, but in different amounts of time. The first version was very inefficient and will be ignored. Version 2 (v2) was a simple wavelet transform as described in section 2.4.2.2 which returned the wavelet coefficient at a given time displacement and scale. Version 3 (v3) made two critical changes. First, it implemented numpy's vector dot product rather than a purely python-based one I had written. The second critical change was that rather than determining the wavelet coefficient at a single time displacement, it determined all the time displacements at once for a single scale, which eliminated the need to evaluate the mother wavelet more than once per scale. Version 4 (v4) further improved upon Version 3 by approximating the value of the mother wavelet to 0 outside a window of $\tau \pm 4s$, essentially reducing equation (1) to:

$$W_\varphi(s,\tau,x) = \frac{1}{\sqrt{s}} \int_{\tau-4s}^{t+4s} \overline{\varphi\left(\frac{t-\tau}{s}\right)} x(t)dt \tag{3.1}$$

This simplification allowed for much smaller vector dot products to be carried out, since the entire time series did not need to be included, but only a relatively small window. The code for Algorithms v2 and v4 is provided in Appendix 1.

In Figure 3.1 it can be observed that algorithm v2 was quadratically dependent on the number of data. Algorithm v3 was roughly 25 times as fast as v2, but remained quadratically dependent.

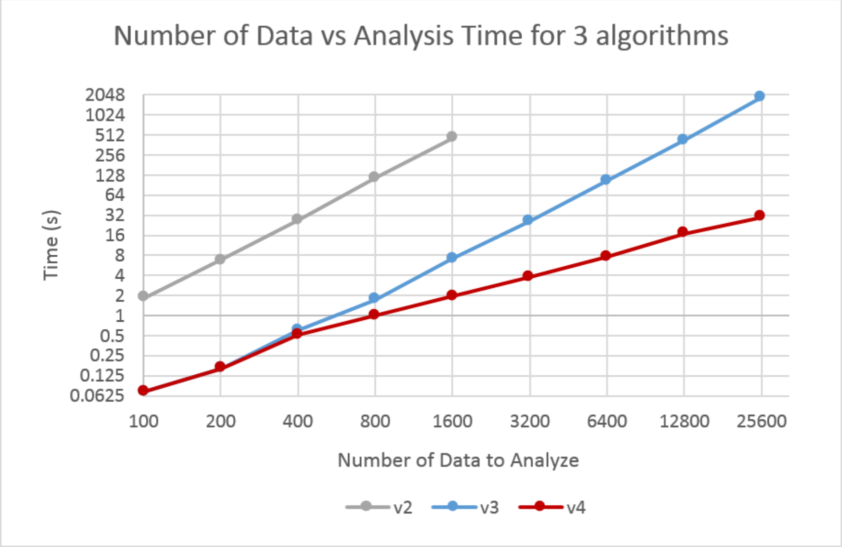


**Figure** 3.1: Comparison in performance of three different wavelet transform algorithms when analyzing synthetic time series of different lengths. Time series were generated by evaluating an arbitrary predefined function at n homogeneously distributed points in time, where n is graphed in the horizontal axis. The wavelet transform was then evaluated at each of the n times, and the total analysis time is graphed on the vertical axis.

Algorithm v4 was as fast as v3 for small data sets, but calculation time was linearly dependent on the number of data.

| Algorithm | Cores Used | Time (s) | Speedup |
|---|---|---|---|
| v2 | 1 | 18837 | 1.0x |
| v3 | 1 | 563.76 | 33.41x |
| v4 | 1 | 19.079 | 987.3x |

**Table** 3.1: Processing times for the same wavelet analysis using three different algorithms on a single core. 31240 data were analyzed to create a scalogram with a resolution of 1200 x 400 pixels, using a 3.4 GHz Intel Core i7 processor. Algorithm v4 performed the task almost a thousand times as fast as v2.

As can be observed in Tables 3.1 and 3.2, algorithm v4 was nearly 3 orders of magnitude faster than v2 when creating scalograms similar to those used in this thesis. Parallelization was useful for reducing the processing time of the slower algorithms, but algorithm v4 was faster when using a single core than when using four, due to the overhead in assigning tasks to different cores when parallel processing.

## 3.2 Effects of Algorithm Parameters on Success Rate

Many stages in the data analysis process outlined in chapter 3 have parameters that could have a large effect on the results, such as the scales to use when performing the wavelet transform, the number of components to include in the reduced feature vector, the criteria for selecting the

| Algorithm | Cores Used | Time (s) | Speedup |
|---|---|---|---|
| v2 | 1 | 2286 | 1.0 x |
| v3 | 1 | 138.3 | 16.5 x |
| v4 | 1 | 2.873 | 795.9 x |
| v2 | 4 | 452.8 | 5.1 x |
| v3 | 4 | 49.99 | 45.7 x |
| v4 | 4 | 15.36 | 148.8 x |

**Table** 3.2: Processing times for the same wavelet analysis using three different algorithms, using single core and parallel. 44800 data were analyzed to create a 100 x 25 pixel scalogram using a 1.00 GHz AMD A6 CPU with 4 cores.

aforementioned components, the type of function to use for the classifier (e.g. linear, quadratic, polynomial), and the selection of data for training and testing. Before analyzing all our data, we first evaluated some of these parameters using a sample of our data, to determine their effects of each on success rate.

### 3.2.1 Number of Components vs Success Rate

One of the most important parameters to determine was the number of components to use in the reduced feature vector. To do this, a sample data set was used to train and test binary classifiers. Classifiers were created using from 2 to 49 components in the reduced feature vectors and the self-validated and cross-validated evaluation results were recorded.

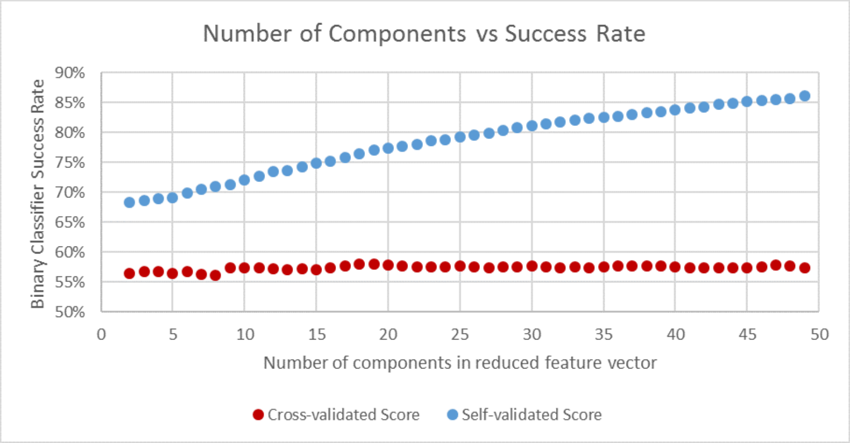


**Figure** 3.2: Effect of the number of components included in the reduced feature vector on the success rate of binary classifiers. Success rate was evaluated using training data (self-validated) and using independent data from the same set as the training data (cross-validated).

It can be observed in Figure 3.2 that the self-validated success rate increases with the number of components, but the cross-validated score remains practically unchanged for 9 components or more. We chose to reduce the feature vectors to 18 components.

### 3.2.2 Test Section vs Success Rate

Each 30 second test was divided into five six-second cycles, and as described in section 3.5.1, when performing cross-validation, four cycles were used as training data and one as test data. We were curious to see if the selection of the one test section had a significant effect on success rate.

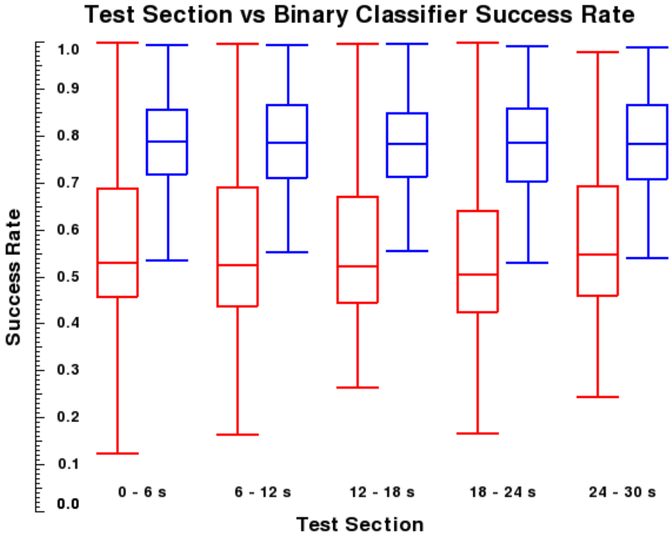


**Figure** 3.3: Box-and whisker plot showing correlation between test section and success rate. Self-validated results are in blue and cross-validated results are in red. When performing cross-validation, four of the five cycles are used as training data while the remaining one is used as test data. This plot shows the range of scores when each of the five cycles is used for testing.

The variations between test section are insignificant with respect to the variations within each test section, for both self- and cross-validated classifiers. This indicates that any test section is equally valid when creating and evaluating a binary classifier with our experimental setup. For increased statistical accuracy, we evaluated all test sections and averaged the results.

## 3.3 Binary Classifier Evaluation Histograms

After evaluating the discriminating functions using test data from the two movements i and j, the results were plotted in a histogram, where vectors corresponding to one movement are in red and those corresponding to another are in blue. The binary classifier considers all negative results to correspond to i and positive results to belong to j. If the two distributions are well separated, the classifier will have a high success rate. Left: Two distributions with a separation index of 2.198. All red results are positive and most blue results are negative, indicating a very high success rate. Right: Two distributions with a separation index of 0.192, meaning that the distributions are on top of each other and this classifier is not reliable.

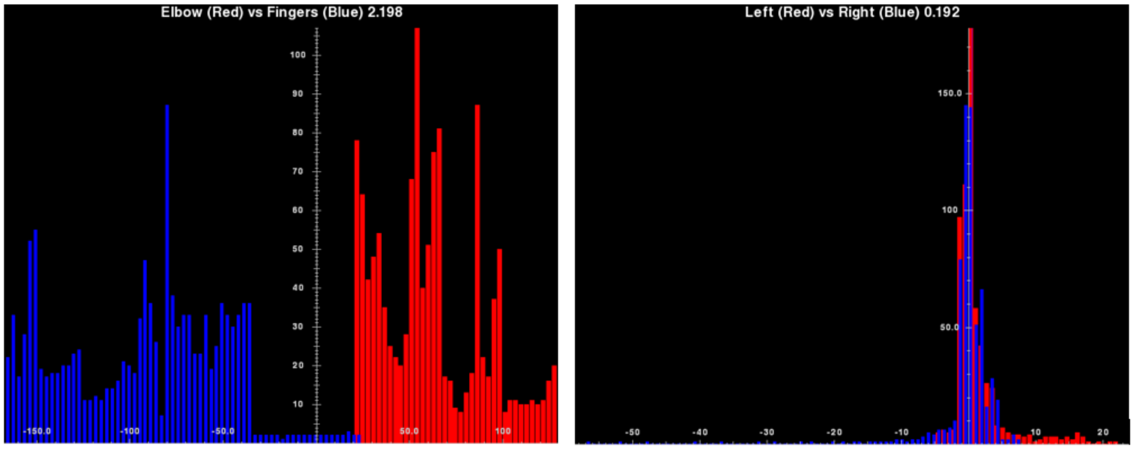


**Figure** 3.4: Examples of histograms of the linear discriminating functions gij:Rn R described in section 2.4.4.

## 3.4 Separation Index vs Success Rate

For both self- and cross-validated data, there is a clear correlation between separation index and success rate. As would be expected, when the separation index is near zero, the success rate is near 50%, since it is basically impossible to distinguish between the two data sets. As the separation index increases, the success rate does as well. In the cross-validation case, the lower part of the curve below 50% occurs because the separation index is defined with an absolute value. This means that the distributions of linear discriminating function evaluations for the two data sets are crossed, so that although the two distributions are separate, the one that should be on the left is on the right, and vice versa.

A classifier is generated using training data and the separation index of the classifier can be determined using the same training data. However, the classifier is intended to be used to classify unknown test data. It is therefore important to determine the correlation between the separation index using training data and the success rate for independent test data. This shows us how well the separation index predicts the success rate for new data. In the figure, we see two different tendencies. On one hand, separation indices between 0.2 and 1.3 seem to show no correlation to success rate, and on the other, separation indices above 1.4 are strongly related to success rate. This indicates that when a binary classifier is generated, its separation index can be evaluated. With the experiment and data presented in this thesis, it appears that separation indices above 1.4 indicate a reliable binary classifier, whereas those below 1.4 do not mean much at all. It would be very interesting to determine if this tendency appears for other experimental data as well. Upon further inspection, the high separation index data all corresponded to binary classifiers between movements from different tests. Binary classifiers discriminating between opposite movements all had separation indices below 1.4

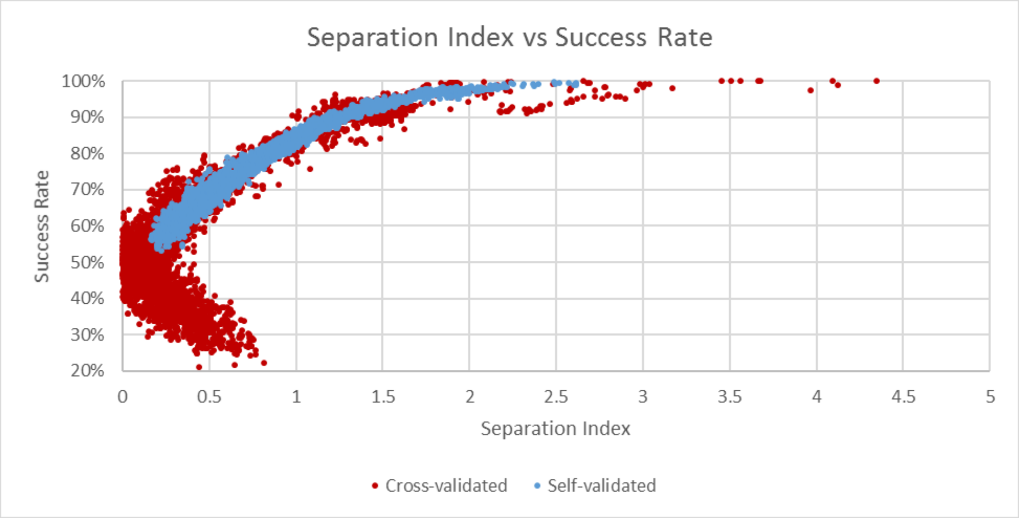


**Figure** 3.5: Scatter plot comparing separation index to binary classifier success rates for both self-validated and cross-validated classifiers. 5704 binary classifiers were generated using different subjects, movement pairs, and numbers of components. The separation index and success rate of each binary classifier was determined with both self-validated and cross-validated evaluations.

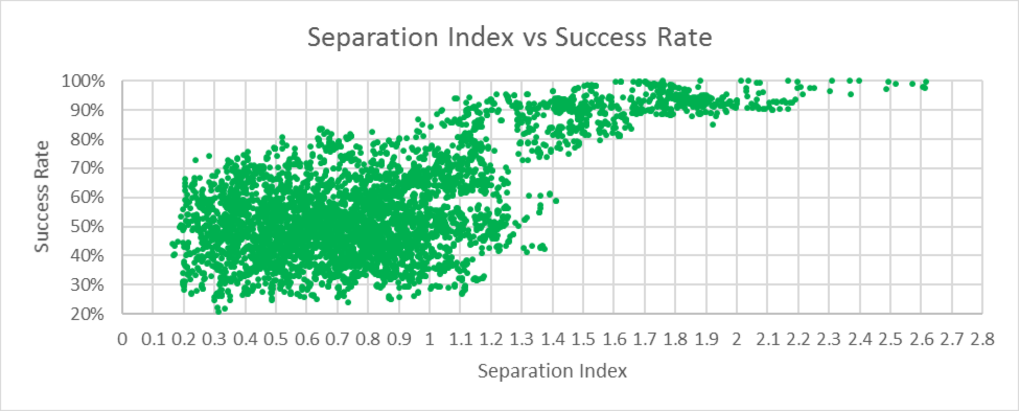


**Figure** 3.6: Scatter plot showing correlation between training data separation index and test data success rate.

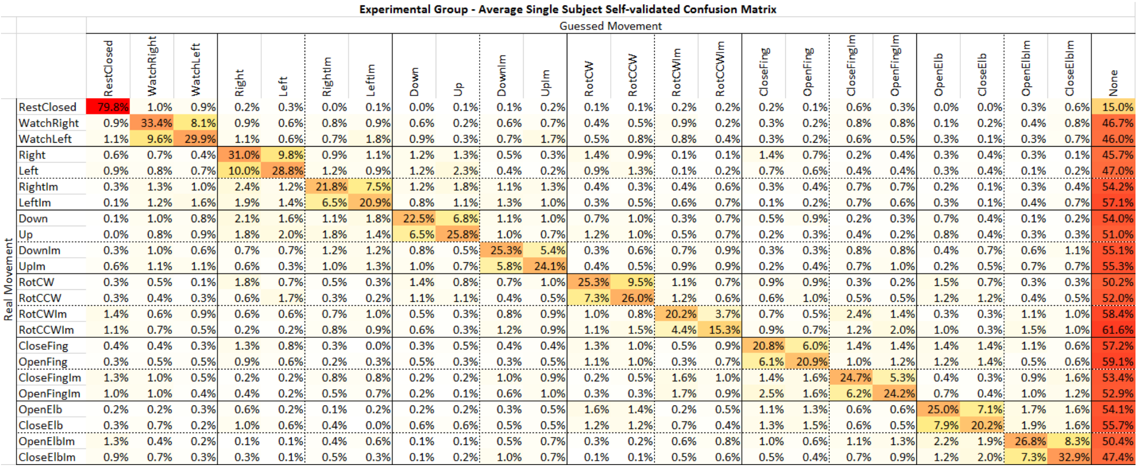
## 3.5 Confusion Matrices

### 3.5.1 Single Subject self-validated

#### 3.5.1.1 Experimental Group Results

The final column in Table 3.3, labelled "None", indicates the percentage of master classifier evaluations which yielded indecisive results. Remember that the master classifier evaluates a total of 276 binary classifiers (all iterations of two different movements out of 23 options), each of which determines which of two possible states the feature vector is more likely to belong. If any one state, or movement, is chosen 22 times, that movement is the result of the master classifier. Otherwise, the result is None. For all movements except RestClosed, around 50% of master classifier evaluations return None.

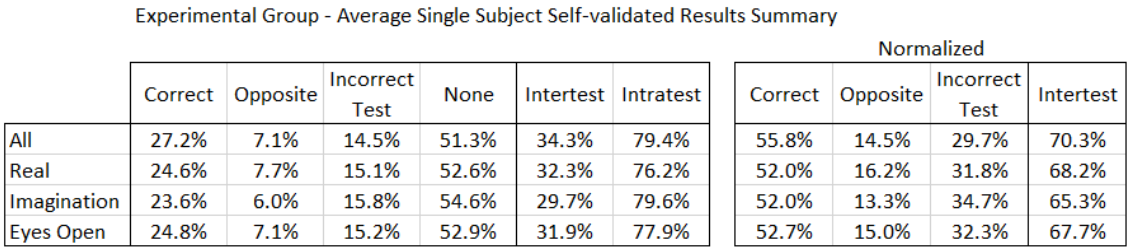
The diagonal orange-red line in Table 3.3 shows that the most common response of the master classifier evaluations were correct. The yellow boxes to the right or below the diagonal indicate that the next most common responses were the opposite movement. That is to say, there is much higher confusion between movements belonging to the same test (ie: Right and Left) than between any two movements belonging to different tests.



**Experimental Group - Average Single Subject Self-validated Confusion Matrix**

| | Guessed Movement | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Real Movement | RestClosed | WatchRight | WatchLeft | Right | Left | RightIm | LeftIm | Down | Up | DownIm | UpIm | RotCW | RotCCW | RotCWIm | RotCCWIm | CloseFing | OpenFing | CloseFingIm | OpenFingIm | OpenElb | CloseElb | OpenElbIm | CloseElbIm | None |
| RestClosed | 79.8% | 1.0% | 0.9% | 0.2% | 0.3% | 0.0% | 0.1% | 0.0% | 0.1% | 0.1% | 0.2% | 0.1% | 0.1% | 0.2% | 0.2% | 0.2% | 0.1% | 0.6% | 0.3% | 0.0% | 0.0% | 0.3% | 0.6% | 15.0% |
| WatchRight | 0.9% | 33.4% | 8.1% | 0.9% | 0.6% | 0.8% | 0.9% | 0.6% | 0.2% | 0.6% | 0.7% | 0.4% | 0.5% | 0.9% | 0.2% | 0.3% | 0.2% | 0.8% | 0.8% | 0.1% | 0.2% | 0.4% | 0.8% | 46.7% |
| WatchLeft | 1.1% | 9.6% | 29.9% | 1.1% | 0.6% | 0.7% | 1.8% | 0.9% | 0.3% | 0.7% | 1.7% | 0.5% | 0.8% | 0.8% | 0.4% | 0.3% | 0.2% | 0.6% | 0.5% | 0.3% | 0.1% | 0.3% | 0.7% | 46.0% |
| Right | 0.6% | 0.7% | 0.4% | 31.0% | 9.8% | 0.9% | 1.1% | 1.2% | 1.3% | 0.5% | 0.3% | 1.4% | 0.9% | 0.1% | 0.1% | 1.4% | 0.7% | 0.2% | 0.4% | 0.3% | 0.4% | 0.3% | 0.1% | 45.7% |
| Left | 0.9% | 0.8% | 0.7% | 10.0% | 28.8% | 1.2% | 0.9% | 1.2% | 2.3% | 0.4% | 0.2% | 0.9% | 1.3% | 0.1% | 0.2% | 0.7% | 0.6% | 0.4% | 0.4% | 0.3% | 0.4% | 0.1% | 0.2% | 47.0% |
| RightIm | 0.3% | 1.3% | 1.0% | 2.4% | 1.2% | 21.8% | 7.5% | 1.2% | 1.8% | 1.1% | 1.3% | 0.4% | 0.3% | 0.4% | 0.6% | 0.3% | 0.4% | 0.7% | 0.7% | 0.2% | 0.1% | 0.3% | 0.4% | 54.2% |
| LeftIm | 0.1% | 1.2% | 1.6% | 1.9% | 1.4% | 6.5% | 20.9% | 0.8% | 1.1% | 1.3% | 1.0% | 0.3% | 0.5% | 0.6% | 0.7% | 0.1% | 0.2% | 0.7% | 0.6% | 0.3% | 0.1% | 0.4% | 0.7% | 57.1% |
| Down | 0.1% | 1.0% | 0.8% | 2.1% | 1.6% | 1.1% | 1.8% | 22.5% | 6.8% | 1.1% | 1.0% | 0.7% | 1.0% | 0.3% | 0.7% | 0.5% | 0.9% | 0.2% | 0.3% | 0.7% | 0.4% | 0.1% | 0.2% | 54.0% |
| Up | 0.0% | 0.8% | 0.9% | 1.8% | 2.0% | 1.8% | 1.4% | 6.5% | 25.8% | 1.0% | 0.7% | 1.2% | 1.0% | 0.5% | 0.7% | 0.2% | 0.3% | 0.4% | 0.2% | 0.8% | 0.4% | 0.3% | 0.3% | 51.0% |
| DownIm | 0.3% | 1.0% | 0.6% | 0.7% | 0.7% | 1.2% | 1.2% | 0.8% | 0.5% | 25.3% | 5.4% | 0.3% | 0.6% | 0.7% | 0.9% | 0.3% | 0.3% | 0.8% | 0.8% | 0.4% | 0.7% | 0.6% | 1.1% | 55.1% |
| UpIm | 0.6% | 1.1% | 1.1% | 0.6% | 0.3% | 1.0% | 1.3% | 1.0% | 0.7% | 5.8% | 24.1% | 0.4% | 0.5% | 0.9% | 0.9% | 0.2% | 0.4% | 0.7% | 1.0% | 0.2% | 0.5% | 0.7% | 0.7% | 55.3% |
| RotCW | 0.3% | 0.5% | 0.1% | 1.8% | 0.7% | 0.5% | 0.3% | 1.4% | 0.8% | 0.7% | 1.0% | 25.3% | 9.5% | 1.1% | 0.7% | 0.9% | 0.9% | 0.3% | 0.2% | 1.5% | 0.7% | 0.3% | 0.3% | 50.2% |
| RotCCW | 0.3% | 0.4% | 0.3% | 0.6% | 1.7% | 0.3% | 0.2% | 1.1% | 1.1% | 0.4% | 0.5% | 7.3% | 26.0% | 1.2% | 0.6% | 0.6% | 1.0% | 0.5% | 0.5% | 1.2% | 1.2% | 0.4% | 0.5% | 52.0% |
| RotCWIm | 1.4% | 0.6% | 0.9% | 0.6% | 0.6% | 0.7% | 1.0% | 0.5% | 0.3% | 0.8% | 0.9% | 1.0% | 0.8% | 20.2% | 3.7% | 0.7% | 0.5% | 2.4% | 1.4% | 0.3% | 0.3% | 1.1% | 1.0% | 58.4% |
| RotCCWIm | 1.1% | 0.7% | 0.5% | 0.2% | 0.2% | 0.8% | 0.9% | 0.6% | 0.3% | 1.2% | 0.9% | 1.1% | 1.5% | 4.4% | 15.3% | 0.9% | 0.7% | 1.2% | 2.0% | 1.0% | 0.3% | 1.5% | 1.0% | 61.6% |
| CloseFing | 0.4% | 0.4% | 0.3% | 1.3% | 0.8% | 0.3% | 0.0% | 0.3% | 0.5% | 0.4% | 0.4% | 1.3% | 1.0% | 0.5% | 0.9% | 20.8% | 6.0% | 1.4% | 1.4% | 1.4% | 1.4% | 1.1% | 0.6% | 57.2% |
| OpenFing | 0.3% | 0.5% | 0.5% | 0.9% | 0.6% | 0.2% | 0.3% | 0.5% | 0.3% | 0.3% | 0.5% | 1.1% | 1.0% | 0.3% | 0.7% | 6.1% | 20.9% | 1.0% | 1.2% | 1.2% | 1.4% | 0.5% | 0.6% | 59.1% |
| CloseFingIm | 1.3% | 1.0% | 0.5% | 0.2% | 0.2% | 0.8% | 0.8% | 0.2% | 0.2% | 1.0% | 0.9% | 0.2% | 0.5% | 1.6% | 1.0% | 1.4% | 1.6% | 24.7% | 5.3% | 0.4% | 0.3% | 0.9% | 1.6% | 53.4% |
| OpenFingIm | 1.0% | 1.0% | 0.4% | 0.4% | 0.2% | 0.5% | 0.7% | 0.2% | 0.1% | 0.6% | 1.0% | 0.3% | 0.3% | 1.7% | 0.9% | 2.5% | 1.6% | 6.2% | 24.2% | 0.7% | 0.4% | 1.0% | 1.2% | 52.9% |
| OpenElb | 0.2% | 0.2% | 0.3% | 0.6% | 0.2% | 0.1% | 0.2% | 0.2% | 0.2% | 0.3% | 0.5% | 1.6% | 1.4% | 0.2% | 0.5% | 1.1% | 1.3% | 0.6% | 0.6% | 25.0% | 7.1% | 1.7% | 1.6% | 54.1% |
| CloseElb | 0.3% | 0.7% | 0.2% | 1.0% | 0.6% | 0.4% | 0.0% | 0.6% | 0.6% | 0.5% | 0.5% | 1.2% | 1.2% | 0.7% | 0.4% | 1.3% | 1.5% | 0.6% | 0.5% | 7.9% | 20.2% | 1.9% | 1.6% | 55.7% |
| OpenElbIm | 1.3% | 0.4% | 0.2% | 0.1% | 0.1% | 0.4% | 0.6% | 0.1% | 0.1% | 0.5% | 0.7% | 0.3% | 0.2% | 0.6% | 0.8% | 1.0% | 0.6% | 1.1% | 1.3% | 2.2% | 1.9% | 26.8% | 8.3% | 50.4% |
| CloseElbIm | 0.9% | 0.7% | 0.3% | 0.3% | 0.1% | 0.5% | 0.3% | 0.1% | 0.2% | 1.0% | 0.7% | 0.1% | 0.1% | 0.5% | 0.6% | 0.5% | 0.4% | 0.7% | 0.9% | 1.2% | 2.0% | 7.3% | 32.9% | 47.4% |

**Table** 3.3: Confusion matrix showing the average of the confusion matrices of 9 experimental subjects. Each row of the confusion matrix shows which move was guessed by the master classifier for vectors corresponding to the movement shown in the first column of the row. This confusion matrix shows the results when the training and test data are the same.



Experimental Group - Average Single Subject Self-validated Results Summary

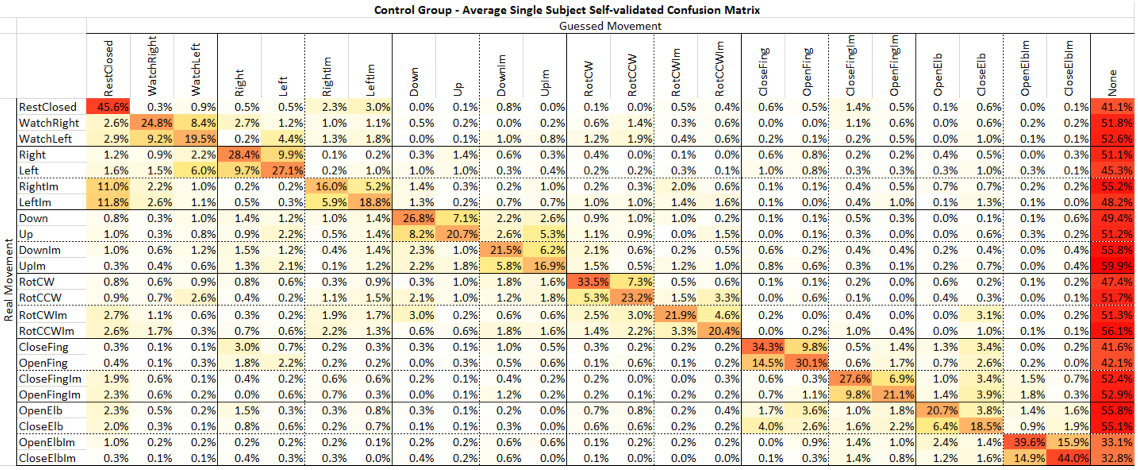
| | | | | | | | Normalized | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Correct | Opposite | Incorrect Test | None | Intertest | Intratest | Correct | Opposite | Incorrect Test | Intertest |
| All | 27.2% | 7.1% | 14.5% | 51.3% | 34.3% | 79.4% | 55.8% | 14.5% | 29.7% | 70.3% |
| Real | 24.6% | 7.7% | 15.1% | 52.6% | 32.3% | 76.2% | 52.0% | 16.2% | 31.8% | 68.2% |
| Imagination | 23.6% | 6.0% | 15.8% | 54.6% | 29.7% | 79.6% | 52.0% | 13.3% | 34.7% | 65.3% |
| Eyes Open | 24.8% | 7.1% | 15.2% | 52.9% | 31.9% | 77.9% | 52.7% | 15.0% | 32.3% | 67.7% |

**Table** 3.4: Summary of the results in the confusion matrix of Table 3.3. The column labelled “Correct” shows the overall success rate, which is the average of the matrixs diagonal. “Opposite” shows the rate at which the opposite movement is chosen. “Incorrect Test” shows the total of all cells that are not correct, opposite or None, and “None” shows the average of the None column. “Intertest” shows the rate at which the correct test is chosen; it is the sum of the Correct and Opposite columns. Intratest shows the rate at which the correct movement is chosen between the two movements of the same test; it is the Correct column divided by the sum of Correct and Opposite. The right box, labelled “Normalized” shows the same numbers, if None results are ignored. That is to say, if all the elements of each row of the matrix are multiplied by (1  N) where N is the None value for that row. The “All” row contains data averaged from all rows of the confusion matrix. The “Real” row contains data only corresponding to real physical movements (Right, Left, Up, Down, RotCW, RotCCW, CloseFing, OpenFing, OpenElb and CloseElb). The “Imagination” row contains data corresponding to imaginary movements, and the “Eyes Open” row contains information about all rows except the first one.

It can be seen in Table 3.4 that the master classifier success rate is 27.2%, which is outstanding, especially considering that not only are there 23 options to choose from, but in order to be chosen by the master classifier, a move must be chosen 22 times by 22 different binary classifiers. This means that the success rate of random chance is $2^{-22} = 2.38 \times 10^{-7}$, meaning that our success rate is over a million times the success rate of an algorithm that just randomly chooses between two options for each binary classifier. The normalized success rate is 55.8%, which is over 10 times the normalized random chance success rate of 123=4.35%. Of the responses that are neither correct nor None, 1/3 correspond to the opposite movement, and the other 2/3 are divided among all other 21 movements, indicating a much higher confusion among the two movements of the same test.

#### 3.5.1.2 Control Group Results

By any measure, results shown in Tables 3.3 and 3.4 look very promising. However, when compared to those in Tables 3.5 and 3.6 they are a bit less spectacular. The control group performed no movements at all, meaning there was no real difference between the state vectors corresponding to one movement and those corresponding to another. Although the experimental group did achieve higher success rates and None rates and lower opposite and incorrect test rates than the control group, the difference was minimal and statistically insignificant, and corresponded almost entirely to the higher success rate of RestClosed in the experimental group.

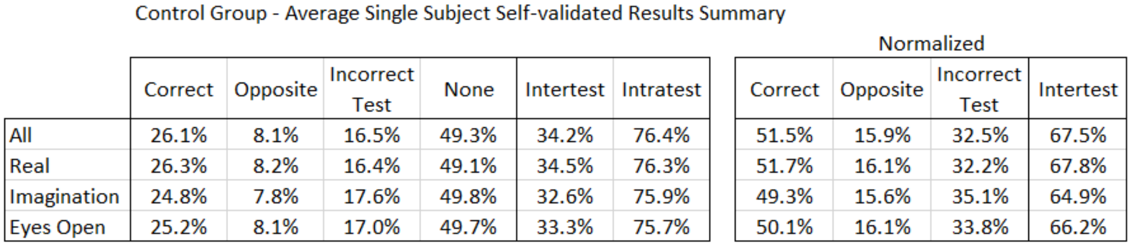


Control Group - Average Single Subject Self-validated Confusion Matrix

Guessed Movement

| | | RestClosed | WatchRight | WatchLeft | Right | Left | RightIm | LeftIm | Down | Up | DownIm | UpIm | RotCW | RotCCW | RotCWIm | RotCCWIm | CloseFing | OpenFing | CloseFingIm | OpenFingIm | OpenElb | CloseElb | OpenElbIm | CloseElbIm | None |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Real Movement | RestClosed | 45.6% | 0.3% | 0.9% | 0.5% | 0.5% | 2.3% | 3.0% | 0.0% | 0.1% | 0.8% | 0.0% | 0.1% | 0.0% | 0.5% | 0.4% | 0.6% | 0.5% | 1.4% | 0.5% | 0.1% | 0.6% | 0.0% | 0.1% | 41.1% |
| | WatchRight | 2.6% | 24.8% | 8.4% | 2.7% | 1.2% | 1.0% | 1.1% | 0.5% | 0.2% | 0.0% | 0.2% | 0.6% | 1.4% | 0.3% | 0.6% | 0.0% | 0.0% | 1.1% | 0.6% | 0.0% | 0.6% | 0.2% | 0.2% | 51.8% |
| | WatchLeft | 2.9% | 9.2% | 19.5% | 0.2% | 4.4% | 1.3% | 1.8% | 0.0% | 0.1% | 1.0% | 0.8% | 1.2% | 1.9% | 0.4% | 0.6% | 0.2% | 0.1% | 0.2% | 0.5% | 0.0% | 1.0% | 0.1% | 0.1% | 52.6% |
| | Right | 1.2% | 0.9% | 2.2% | 28.4% | 9.9% | 0.1% | 0.2% | 0.3% | 1.4% | 0.6% | 0.3% | 0.4% | 0.0% | 0.1% | 0.0% | 0.6% | 0.8% | 0.2% | 0.2% | 0.4% | 0.5% | 0.0% | 0.3% | 51.1% |
| | Left | 1.6% | 1.5% | 6.0% | 9.7% | 27.1% | 0.2% | 1.0% | 1.0% | 1.0% | 0.3% | 0.4% | 0.2% | 0.2% | 0.3% | 0.1% | 1.0% | 0.8% | 0.3% | 0.3% | 0.3% | 1.0% | 0.3% | 0.1% | 45.3% |
| | RightIm | 11.0% | 2.2% | 1.0% | 0.2% | 0.2% | 16.0% | 5.2% | 1.4% | 0.3% | 0.2% | 1.0% | 0.2% | 0.3% | 2.0% | 0.6% | 0.1% | 0.1% | 0.4% | 0.5% | 0.7% | 0.7% | 0.2% | 0.2% | 55.2% |
| | LeftIm | 11.8% | 2.6% | 1.1% | 0.5% | 0.3% | 5.9% | 18.8% | 1.3% | 0.2% | 0.7% | 0.7% | 1.0% | 1.0% | 1.4% | 1.6% | 0.1% | 0.0% | 0.4% | 1.0% | 0.1% | 1.3% | 0.1% | 0.0% | 48.2% |
| | Down | 0.8% | 0.3% | 1.0% | 1.4% | 1.2% | 1.0% | 1.4% | 26.8% | 7.1% | 2.2% | 2.6% | 0.9% | 1.0% | 1.0% | 0.2% | 0.1% | 0.1% | 0.5% | 0.3% | 0.0% | 0.1% | 0.1% | 0.6% | 49.4% |
| | Up | 1.0% | 0.3% | 0.8% | 0.9% | 2.2% | 0.5% | 1.4% | 8.2% | 20.7% | 2.6% | 5.3% | 1.1% | 0.9% | 0.0% | 1.5% | 0.0% | 0.1% | 0.3% | 0.0% | 0.0% | 0.6% | 0.3% | 0.2% | 51.2% |
| | DownIm | 1.0% | 0.6% | 1.2% | 1.5% | 1.2% | 0.4% | 1.4% | 2.3% | 1.0% | 21.5% | 6.2% | 2.1% | 0.6% | 0.2% | 0.5% | 0.6% | 0.2% | 0.4% | 0.4% | 0.2% | 0.4% | 0.0% | 0.4% | 55.8% |
| | UpIm | 0.3% | 0.4% | 0.6% | 1.3% | 2.1% | 0.1% | 1.2% | 2.2% | 1.8% | 5.8% | 16.9% | 1.5% | 0.5% | 1.2% | 1.0% | 0.8% | 0.6% | 0.3% | 0.1% | 0.2% | 0.7% | 0.0% | 0.4% | 59.9% |
| | RotCW | 0.8% | 0.6% | 0.9% | 0.8% | 0.6% | 0.3% | 0.9% | 0.3% | 1.0% | 1.8% | 1.6% | 33.5% | 7.3% | 0.5% | 0.6% | 0.1% | 0.1% | 0.2% | 0.0% | 0.6% | 0.2% | 0.1% | 0.2% | 47.4% |
| | RotCCW | 0.9% | 0.7% | 2.6% | 0.4% | 0.2% | 1.1% | 1.5% | 2.1% | 1.0% | 1.2% | 1.8% | 5.3% | 23.2% | 1.5% | 3.3% | 0.0% | 0.6% | 0.1% | 0.0% | 0.4% | 0.3% | 0.0% | 0.1% | 51.7% |
| | RotCWIm | 2.7% | 1.1% | 0.6% | 0.3% | 0.2% | 1.9% | 1.7% | 3.0% | 0.2% | 0.6% | 0.6% | 2.5% | 3.0% | 21.9% | 4.6% | 0.2% | 0.0% | 0.4% | 0.0% | 0.0% | 3.1% | 0.0% | 0.2% | 51.3% |
| | RotCCWIm | 2.6% | 1.7% | 0.3% | 0.7% | 0.6% | 2.2% | 1.3% | 0.6% | 0.6% | 1.8% | 1.6% | 1.4% | 2.2% | 3.3% | 20.4% | 0.0% | 0.2% | 1.0% | 0.4% | 0.0% | 1.0% | 0.1% | 0.1% | 56.1% |
| | CloseFing | 0.3% | 0.1% | 0.1% | 3.0% | 0.7% | 0.2% | 0.3% | 0.3% | 0.1% | 1.0% | 0.5% | 0.3% | 0.2% | 0.2% | 0.2% | 34.3% | 9.8% | 0.5% | 1.4% | 1.3% | 3.4% | 0.0% | 0.2% | 41.6% |
| | OpenFing | 0.4% | 0.1% | 0.3% | 1.8% | 2.2% | 0.2% | 0.2% | 0.0% | 0.3% | 0.5% | 0.6% | 0.1% | 0.6% | 0.1% | 0.2% | 14.5% | 30.1% | 0.6% | 1.7% | 0.7% | 2.6% | 0.2% | 0.0% | 42.1% |
| | CloseFingIm | 1.9% | 0.6% | 0.1% | 0.4% | 0.2% | 0.6% | 0.6% | 0.2% | 0.1% | 0.4% | 0.2% | 0.2% | 0.0% | 0.0% | 0.3% | 0.6% | 0.3% | 27.6% | 6.9% | 1.0% | 3.4% | 1.5% | 0.7% | 52.4% |
| | OpenFingIm | 2.3% | 0.6% | 0.2% | 0.0% | 0.6% | 0.7% | 0.3% | 0.0% | 0.1% | 1.2% | 0.2% | 0.2% | 0.1% | 0.2% | 0.2% | 0.7% | 1.1% | 9.8% | 21.1% | 1.4% | 3.9% | 1.8% | 0.3% | 52.9% |
| | OpenElb | 2.3% | 0.5% | 0.2% | 1.5% | 0.3% | 0.3% | 0.8% | 0.3% | 0.1% | 0.2% | 0.0% | 0.7% | 0.8% | 0.2% | 0.4% | 1.7% | 3.6% | 1.0% | 1.8% | 20.7% | 3.8% | 1.4% | 1.6% | 55.8% |
| | CloseElb | 2.0% | 0.3% | 0.1% | 0.8% | 0.6% | 0.2% | 0.7% | 0.1% | 0.1% | 0.3% | 0.0% | 0.6% | 0.6% | 0.2% | 0.2% | 4.0% | 2.6% | 1.6% | 2.2% | 6.4% | 18.5% | 0.9% | 1.9% | 55.1% |
| | OpenElbIm | 1.0% | 0.2% | 0.2% | 0.2% | 0.2% | 0.2% | 0.1% | 0.2% | 0.2% | 0.6% | 0.6% | 0.1% | 0.2% | 0.2% | 0.2% | 0.0% | 0.9% | 1.4% | 1.0% | 2.4% | 1.4% | 39.6% | 15.9% | 33.1% |
| | CloseElbIm | 0.3% | 0.1% | 0.1% | 0.4% | 0.3% | 0.3% | 0.0% | 0.4% | 0.2% | 0.6% | 0.0% | 0.1% | 0.0% | 0.0% | 0.0% | 0.1% | 0.3% | 1.4% | 0.8% | 1.2% | 1.6% | 14.9% | 44.0% | 32.8% |

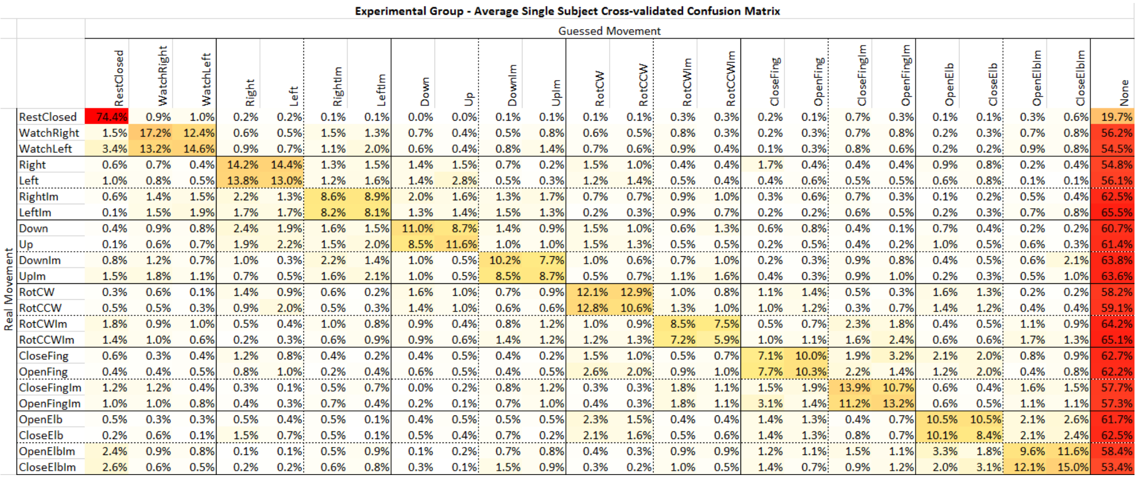
**Table** 3.5: Confusion matrix showing the average of the self-validated confusion matrices of 2 control subjects.



Control Group - Average Single Subject Self-validated Results Summary

| | | | | | | | Normalized | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Correct | Opposite | Incorrect Test | None | Intertest | Intratest | Correct | Opposite | Incorrect Test | Intertest |
| All | 26.1% | 8.1% | 16.5% | 49.3% | 34.2% | 76.4% | 51.5% | 15.9% | 32.5% | 67.5% |
| Real | 26.3% | 8.2% | 16.4% | 49.1% | 34.5% | 76.3% | 51.7% | 16.1% | 32.2% | 67.8% |
| Imagination | 24.8% | 7.8% | 17.6% | 49.8% | 32.6% | 75.9% | 49.3% | 15.6% | 35.1% | 64.9% |
| Eyes Open | 25.2% | 8.1% | 17.0% | 49.7% | 33.3% | 75.7% | 50.1% | 16.1% | 33.8% | 66.2% |

**Table** 3.6: Summary of the results in Table 3.5.

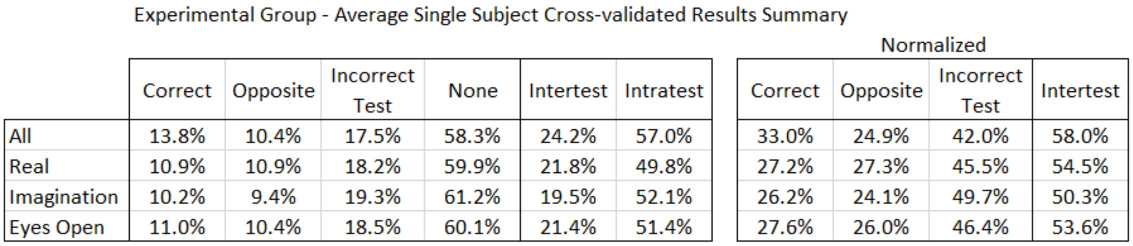


Experimental Group - Average Single Subject Cross-validated Confusion Matrix

Guessed Movement

| | | RestClosed | WatchRight | WatchLeft | Right | Left | RightIm | LeftIm | Down | Up | DownIm | UpIm | RotCW | RotCCW | RotCWIm | RotCCWIm | CloseFing | OpenFing | CloseFingIm | OpenFingIm | OpenElb | CloseElb | OpenElbIm | CloseElbIm | None |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Real Movement | RestClosed | 74.4% | 0.9% | 1.0% | 0.2% | 0.2% | 0.1% | 0.1% | 0.0% | 0.0% | 0.1% | 0.1% | 0.1% | 0.1% | 0.3% | 0.3% | 0.2% | 0.1% | 0.7% | 0.3% | 0.1% | 0.1% | 0.3% | 0.6% | 19.7% |
| | WatchRight | 1.5% | 17.2% | 12.4% | 0.6% | 0.5% | 1.5% | 1.3% | 0.7% | 0.4% | 0.5% | 0.8% | 0.6% | 0.5% | 0.8% | 0.3% | 0.2% | 0.3% | 0.7% | 0.8% | 0.2% | 0.3% | 0.7% | 0.8% | 56.2% |
| | WatchLeft | 3.4% | 13.2% | 14.6% | 0.9% | 0.7% | 1.1% | 2.0% | 0.6% | 0.4% | 0.8% | 1.4% | 0.7% | 0.6% | 0.9% | 0.4% | 0.1% | 0.3% | 0.8% | 0.6% | 0.2% | 0.2% | 0.9% | 0.8% | 54.5% |
| | Right | 0.6% | 0.7% | 0.4% | 14.2% | 14.4% | 1.3% | 1.5% | 1.4% | 1.5% | 0.7% | 0.2% | 1.5% | 1.0% | 0.4% | 0.4% | 1.7% | 0.4% | 0.4% | 0.4% | 0.9% | 0.8% | 0.2% | 0.4% | 54.8% |
| | Left | 1.0% | 0.8% | 0.5% | 13.8% | 13.0% | 1.2% | 1.6% | 1.4% | 2.8% | 0.5% | 0.3% | 1.2% | 1.4% | 0.5% | 0.4% | 0.4% | 0.6% | 0.5% | 0.5% | 0.6% | 0.8% | 0.1% | 0.1% | 56.1% |
| | RightIm | 0.6% | 1.4% | 1.5% | 2.2% | 1.3% | 8.6% | 8.9% | 2.0% | 1.6% | 1.3% | 1.7% | 0.7% | 0.7% | 0.9% | 1.0% | 0.3% | 0.6% | 0.7% | 0.3% | 0.1% | 0.2% | 0.5% | 0.4% | 62.5% |
| | LeftIm | 0.1% | 1.5% | 1.9% | 1.7% | 1.7% | 8.2% | 8.1% | 1.3% | 1.4% | 1.5% | 1.3% | 0.2% | 0.3% | 0.9% | 0.7% | 0.2% | 0.2% | 0.6% | 0.5% | 0.2% | 0.3% | 0.7% | 0.8% | 65.5% |
| | Down | 0.4% | 0.9% | 0.8% | 2.4% | 1.9% | 1.6% | 1.5% | 11.0% | 8.7% | 1.4% | 0.9% | 1.5% | 1.0% | 0.6% | 1.3% | 0.6% | 0.8% | 0.4% | 0.1% | 0.7% | 0.4% | 0.2% | 0.2% | 60.7% |
| | Up | 0.1% | 0.6% | 0.7% | 1.9% | 2.2% | 1.5% | 2.0% | 8.5% | 11.6% | 1.0% | 1.0% | 1.5% | 1.3% | 0.5% | 0.5% | 0.2% | 0.5% | 0.4% | 0.2% | 1.0% | 0.5% | 0.6% | 0.3% | 61.4% |
| | DownIm | 0.8% | 1.2% | 0.7% | 1.0% | 0.3% | 2.2% | 1.4% | 1.0% | 0.5% | 10.2% | 7.7% | 1.0% | 0.6% | 0.7% | 1.0% | 0.2% | 0.3% | 0.9% | 0.8% | 0.4% | 0.5% | 0.6% | 2.1% | 63.8% |
| | UpIm | 1.5% | 1.8% | 1.1% | 0.7% | 0.5% | 1.6% | 2.1% | 1.0% | 0.5% | 8.5% | 8.7% | 0.5% | 0.7% | 1.1% | 1.6% | 0.4% | 0.3% | 0.9% | 1.0% | 0.2% | 0.3% | 0.5% | 1.0% | 63.6% |
| | RotCW | 0.3% | 0.6% | 0.1% | 1.4% | 0.9% | 0.6% | 0.2% | 1.6% | 1.0% | 0.7% | 0.9% | 12.1% | 12.9% | 1.0% | 0.8% | 1.1% | 1.4% | 0.5% | 0.3% | 1.6% | 1.3% | 0.2% | 0.2% | 58.2% |
| | RotCCW | 0.5% | 0.5% | 0.3% | 0.9% | 2.0% | 0.5% | 0.3% | 1.4% | 1.0% | 0.6% | 0.6% | 12.8% | 10.6% | 1.3% | 1.0% | 1.0% | 1.2% | 0.3% | 0.7% | 1.4% | 1.2% | 0.4% | 0.4% | 59.1% |
| | RotCWIm | 1.8% | 0.9% | 1.0% | 0.5% | 0.4% | 1.0% | 0.8% | 0.9% | 0.4% | 0.8% | 1.2% | 1.0% | 0.9% | 8.5% | 7.5% | 0.5% | 0.7% | 2.3% | 1.8% | 0.4% | 0.5% | 1.1% | 0.9% | 64.2% |
| | RotCCWIm | 1.4% | 1.0% | 0.6% | 0.2% | 0.3% | 0.6% | 0.9% | 0.9% | 0.6% | 1.4% | 1.2% | 1.2% | 1.3% | 7.2% | 5.9% | 1.0% | 1.1% | 1.6% | 2.4% | 0.6% | 0.6% | 1.7% | 1.3% | 65.1% |
| | CloseFing | 0.6% | 0.3% | 0.4% | 1.2% | 0.8% | 0.4% | 0.2% | 0.4% | 0.5% | 0.4% | 0.2% | 1.5% | 1.0% | 0.5% | 0.7% | 7.1% | 10.0% | 1.9% | 3.2% | 2.1% | 2.0% | 0.8% | 0.9% | 62.7% |
| | OpenFing | 0.4% | 0.4% | 0.5% | 0.8% | 1.0% | 0.2% | 0.4% | 0.6% | 0.5% | 0.4% | 0.5% | 2.6% | 2.0% | 0.9% | 1.0% | 7.7% | 10.3% | 2.2% | 1.4% | 1.2% | 2.0% | 0.4% | 0.8% | 62.2% |
| | CloseFingIm | 1.2% | 1.2% | 0.4% | 0.3% | 0.1% | 0.5% | 0.7% | 0.0% | 0.2% | 0.8% | 1.2% | 0.3% | 0.3% | 1.8% | 1.1% | 1.5% | 1.9% | 13.9% | 10.7% | 0.6% | 0.4% | 1.6% | 1.5% | 57.7% |
| | OpenFingIm | 1.0% | 1.0% | 0.8% | 0.4% | 0.3% | 0.7% | 0.4% | 0.2% | 0.1% | 0.7% | 1.0% | 0.4% | 0.3% | 1.8% | 1.1% | 3.1% | 1.4% | 11.2% | 13.2% | 0.6% | 0.5% | 1.1% | 1.1% | 57.3% |
| | OpenElb | 0.5% | 0.3% | 0.3% | 0.5% | 0.4% | 0.5% | 0.1% | 0.4% | 0.5% | 0.5% | 0.5% | 2.3% | 1.5% | 0.4% | 0.4% | 1.4% | 1.3% | 0.4% | 0.7% | 10.5% | 10.5% | 2.1% | 2.6% | 61.7% |
| | CloseElb | 0.2% | 0.6% | 0.1% | 1.5% | 0.7% | 0.5% | 0.1% | 0.5% | 0.4% | 0.7% | 0.2% | 2.1% | 1.6% | 0.5% | 0.6% | 1.4% | 1.3% | 0.8% | 0.7% | 10.1% | 8.4% | 2.1% | 2.4% | 62.5% |
| | OpenElbIm | 2.4% | 0.9% | 0.8% | 0.1% | 0.1% | 0.5% | 0.9% | 0.1% | 0.2% | 0.7% | 0.8% | 0.4% | 0.3% | 0.9% | 0.9% | 1.2% | 1.1% | 1.5% | 1.1% | 3.3% | 1.8% | 9.6% | 11.6% | 58.4% |
| | CloseElbIm | 2.6% | 0.6% | 0.5% | 0.2% | 0.2% | 0.6% | 0.8% | 0.3% | 0.1% | 1.5% | 0.9% | 0.3% | 0.2% | 1.0% | 0.5% | 1.4% | 0.7% | 0.9% | 1.2% | 2.0% | 3.1% | 12.1% | 15.0% | 53.4% |

**Table** 3.7: Confusion matrix showing the average of the cross-validated confusion matrices of 9 experimental subjects.



Experimental Group - Average Single Subject Cross-validated Results Summary

| | Correct | Opposite | Incorrect Test | None | Intertest | Intratest |
|---|---|---|---|---|---|---|
| All | 13.8% | 10.4% | 17.5% | 58.3% | 24.2% | 57.0% |
| Real | 10.9% | 10.9% | 18.2% | 59.9% | 21.8% | 49.8% |
| Imagination | 10.2% | 9.4% | 19.3% | 61.2% | 19.5% | 52.1% |
| Eyes Open | 11.0% | 10.4% | 18.5% | 60.1% | 21.4% | 51.4% |

Normalized

| | Correct | Opposite | Incorrect Test | Intertest |
|---|---|---|---|---|
| All | 33.0% | 24.9% | 42.0% | 58.0% |
| Real | 27.2% | 27.3% | 45.5% | 54.5% |
| Imagination | 26.2% | 24.1% | 49.7% | 50.3% |
| Eyes Open | 27.6% | 26.0% | 46.4% | 53.6% |

**Table** 3.8: Summary of the results in Table 3.7.

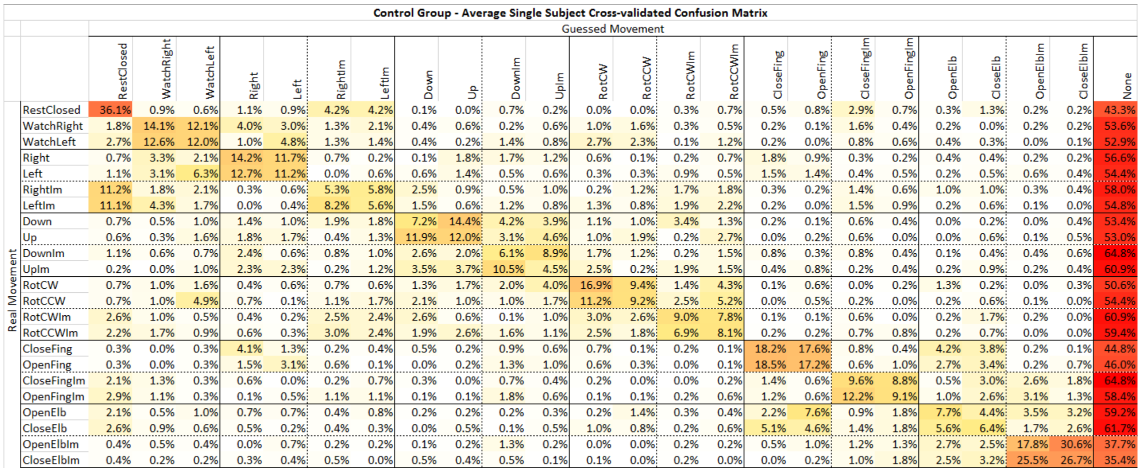
## 3.5.2 Single Subject cross-validated

### 3.5.2.1 Experimental Group Results

Cross-validated results are significantly more reliable than self-validated ones. Although the training and test data are from the same subject, and are in fact subsets of the same larger data set, they are still mutually exclusive.

In tables 3.7 and 3.8, we can observe a number of tendencies. As is to be expected, None is the most common result at around 60%. The exception is in the case of RestClosed, which has a 74.4% success rate and only 19.7% None rate. We also observe a thick yellow diagonal line, showing that although the classifier can choose the correct test, it is basically just guessing between the two movements within each test.

### 3.5.2.2 Control Group Results



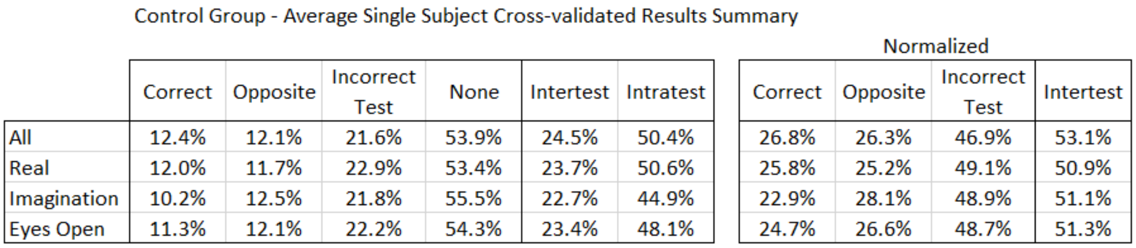
Control Group - Average Single Subject Cross-validated Confusion Matrix

Guessed Movement

| Real Movement | RestClosed | WatchRight | WatchLeft | Right | Left | RightIm | LeftIm | Down | Up | DownIm | UpIm | RotCW | RotCCW | RotCWIm | RotCCWIm | CloseFing | OpenFing | CloseFingIm | OpenFingIm | OpenElb | CloseElb | OpenElbIm | CloseElbIm | None |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RestClosed | 36.1% | 0.9% | 0.6% | 1.1% | 0.9% | 4.2% | 4.2% | 0.1% | 0.0% | 0.7% | 0.2% | 0.0% | 0.0% | 0.3% | 0.7% | 0.5% | 0.8% | 2.9% | 0.7% | 0.3% | 1.3% | 0.2% | 0.2% | 43.3% |
| WatchRight | 1.8% | 14.1% | 12.1% | 4.0% | 3.0% | 1.3% | 2.1% | 0.4% | 0.6% | 0.2% | 0.6% | 1.0% | 1.6% | 0.3% | 0.5% | 0.2% | 0.1% | 1.6% | 0.4% | 0.2% | 0.0% | 0.2% | 0.2% | 53.6% |
| WatchLeft | 2.7% | 12.6% | 12.0% | 1.0% | 4.8% | 1.3% | 1.4% | 0.4% | 0.2% | 1.4% | 0.8% | 2.7% | 2.3% | 0.1% | 1.2% | 0.2% | 0.0% | 0.8% | 0.6% | 0.4% | 0.3% | 0.0% | 0.1% | 52.9% |
| Right | 0.7% | 3.3% | 2.1% | 14.2% | 11.7% | 0.7% | 0.2% | 0.1% | 1.8% | 1.7% | 1.2% | 0.6% | 0.1% | 0.2% | 0.7% | 1.8% | 0.9% | 0.3% | 0.2% | 0.4% | 0.4% | 0.2% | 0.2% | 56.6% |
| Left | 1.1% | 3.1% | 6.3% | 12.7% | 11.2% | 0.0% | 0.6% | 0.6% | 1.4% | 0.5% | 0.6% | 0.3% | 0.3% | 0.9% | 0.5% | 1.5% | 1.4% | 0.4% | 0.5% | 0.2% | 0.5% | 0.6% | 0.4% | 54.4% |
| RightIm | 11.2% | 1.8% | 2.1% | 0.3% | 0.6% | 5.3% | 5.8% | 2.5% | 0.9% | 0.5% | 1.0% | 0.2% | 1.2% | 1.7% | 1.8% | 0.3% | 0.2% | 1.4% | 0.6% | 1.0% | 1.0% | 0.3% | 0.4% | 58.0% |
| LeftIm | 11.1% | 4.3% | 1.7% | 0.0% | 0.4% | 8.2% | 5.6% | 1.5% | 0.6% | 1.2% | 0.8% | 1.3% | 0.8% | 1.9% | 2.2% | 0.2% | 0.0% | 1.5% | 0.9% | 0.2% | 0.6% | 0.1% | 0.0% | 54.8% |
| Down | 0.7% | 0.5% | 1.0% | 1.4% | 1.0% | 1.9% | 1.8% | 7.2% | 14.4% | 4.2% | 3.9% | 1.1% | 1.0% | 3.4% | 1.3% | 0.2% | 0.1% | 0.6% | 0.4% | 0.0% | 0.2% | 0.0% | 0.4% | 53.4% |
| Up | 0.6% | 0.3% | 1.6% | 1.8% | 1.7% | 0.4% | 1.3% | 11.9% | 12.0% | 3.1% | 4.6% | 1.0% | 1.9% | 0.2% | 2.7% | 0.0% | 0.2% | 0.6% | 0.0% | 0.0% | 0.6% | 0.1% | 0.5% | 53.0% |
| DownIm | 1.1% | 0.6% | 0.7% | 2.4% | 0.6% | 0.8% | 1.0% | 2.6% | 2.0% | 6.1% | 8.9% | 1.7% | 1.2% | 0.2% | 1.5% | 0.8% | 0.3% | 0.8% | 0.4% | 0.1% | 0.4% | 0.4% | 0.6% | 64.8% |
| UpIm | 0.2% | 0.0% | 1.0% | 2.3% | 2.3% | 0.2% | 1.2% | 3.5% | 3.7% | 10.5% | 4.5% | 2.5% | 0.2% | 1.9% | 1.5% | 0.4% | 0.8% | 0.2% | 0.4% | 0.2% | 0.9% | 0.2% | 0.4% | 60.9% |
| RotCW | 0.7% | 1.0% | 1.6% | 0.4% | 0.6% | 0.7% | 0.6% | 1.3% | 1.7% | 2.0% | 4.0% | 16.9% | 9.4% | 1.4% | 4.3% | 0.1% | 0.6% | 0.0% | 0.2% | 1.3% | 0.2% | 0.0% | 0.3% | 50.6% |
| RotCCW | 0.7% | 1.0% | 4.9% | 0.7% | 0.1% | 1.1% | 1.7% | 2.1% | 1.0% | 1.0% | 1.7% | 11.2% | 9.2% | 2.5% | 5.2% | 0.0% | 0.5% | 0.2% | 0.0% | 0.2% | 0.6% | 0.1% | 0.0% | 54.4% |
| RotCWIm | 2.6% | 1.0% | 0.5% | 0.4% | 0.2% | 2.5% | 2.4% | 2.6% | 0.6% | 0.1% | 1.0% | 3.0% | 2.6% | 9.0% | 7.8% | 0.1% | 0.1% | 0.6% | 0.0% | 0.2% | 1.7% | 0.2% | 0.0% | 60.9% |
| RotCCWIm | 2.2% | 1.7% | 0.9% | 0.6% | 0.3% | 3.0% | 2.4% | 1.9% | 2.6% | 1.6% | 1.1% | 2.5% | 1.8% | 6.9% | 8.1% | 0.2% | 0.2% | 0.7% | 0.8% | 0.2% | 0.7% | 0.0% | 0.0% | 59.4% |
| CloseFing | 0.3% | 0.0% | 0.3% | 4.1% | 1.3% | 0.2% | 0.4% | 0.5% | 0.2% | 0.9% | 0.6% | 0.7% | 0.1% | 0.2% | 0.1% | 18.2% | 17.6% | 0.8% | 0.4% | 4.2% | 3.8% | 0.2% | 0.1% | 44.8% |
| OpenFing | 0.3% | 0.0% | 0.3% | 1.5% | 3.1% | 0.6% | 0.1% | 0.0% | 0.2% | 1.3% | 1.0% | 0.6% | 0.3% | 0.0% | 0.1% | 18.5% | 17.2% | 0.6% | 1.0% | 2.7% | 3.4% | 0.2% | 0.7% | 46.0% |
| CloseFingIm | 2.1% | 1.3% | 0.3% | 0.6% | 0.0% | 0.2% | 0.7% | 0.3% | 0.0% | 0.7% | 0.4% | 0.2% | 0.0% | 0.0% | 0.2% | 1.4% | 0.6% | 9.6% | 8.8% | 0.5% | 3.0% | 2.6% | 1.8% | 64.8% |
| OpenFingIm | 2.9% | 1.1% | 0.3% | 0.1% | 0.5% | 1.1% | 1.1% | 0.1% | 0.1% | 1.8% | 0.6% | 0.1% | 0.1% | 0.2% | 0.6% | 1.2% | 0.6% | 12.2% | 9.1% | 1.0% | 2.6% | 3.1% | 1.3% | 58.4% |
| OpenElb | 2.1% | 0.5% | 1.0% | 0.7% | 0.7% | 0.4% | 0.8% | 0.2% | 0.2% | 0.2% | 0.3% | 0.2% | 1.4% | 0.3% | 0.4% | 2.2% | 7.6% | 0.9% | 1.8% | 7.7% | 4.4% | 3.5% | 3.2% | 59.2% |
| CloseElb | 2.6% | 0.9% | 0.6% | 0.5% | 0.2% | 0.4% | 0.3% | 0.0% | 0.5% | 0.1% | 0.5% | 1.0% | 0.8% | 0.2% | 0.6% | 5.1% | 4.6% | 1.4% | 1.8% | 5.6% | 6.4% | 1.7% | 2.6% | 61.7% |
| OpenElbIm | 0.4% | 0.5% | 0.4% | 0.0% | 0.7% | 0.2% | 0.2% | 0.1% | 0.2% | 1.3% | 0.2% | 0.0% | 0.0% | 0.2% | 0.2% | 0.5% | 1.0% | 1.2% | 1.3% | 2.7% | 2.5% | 17.8% | 30.6% | 37.7% |
| CloseElbIm | 0.4% | 0.2% | 0.2% | 0.3% | 0.4% | 0.5% | 0.0% | 0.5% | 0.4% | 0.5% | 0.1% | 0.1% | 0.0% | 0.2% | 0.0% | 0.0% | 0.2% | 1.0% | 1.8% | 2.5% | 3.2% | 25.5% | 26.7% | 35.4% |

**Table** 3.9: Confusion matrix showing the average of the cross-validated confusion matrices of 2 control subjects.

Again, the control group's results cast a shadow on those of the experimental group. Here, we see very similar results. Although there is a bit more confusion in the control group, the thick yellow line tendency is definitely present, suggesting that there are some short-term patterns exist within the brain, which remain more or less constant during a 30-second test, but not from one test



Control Group - Average Single Subject Cross-validated Results Summary

|  | Correct | Opposite | Incorrect Test | None | Intertest | Intratest | Normalized Correct | Normalized Opposite | Normalized Incorrect Test | Normalized Intertest |
|---|---|---|---|---|---|---|---|---|---|---|
| All | 12.4% | 12.1% | 21.6% | 53.9% | 24.5% | 50.4% | 26.8% | 26.3% | 46.9% | 53.1% |
| Real | 12.0% | 11.7% | 22.9% | 53.4% | 23.7% | 50.6% | 25.8% | 25.2% | 49.1% | 50.9% |
| Imagination | 10.2% | 12.5% | 21.8% | 55.5% | 22.7% | 44.9% | 22.9% | 28.1% | 48.9% | 51.1% |
| Eyes Open | 11.3% | 12.1% | 22.2% | 54.3% | 23.4% | 48.1% | 24.7% | 26.6% | 48.7% | 51.3% |

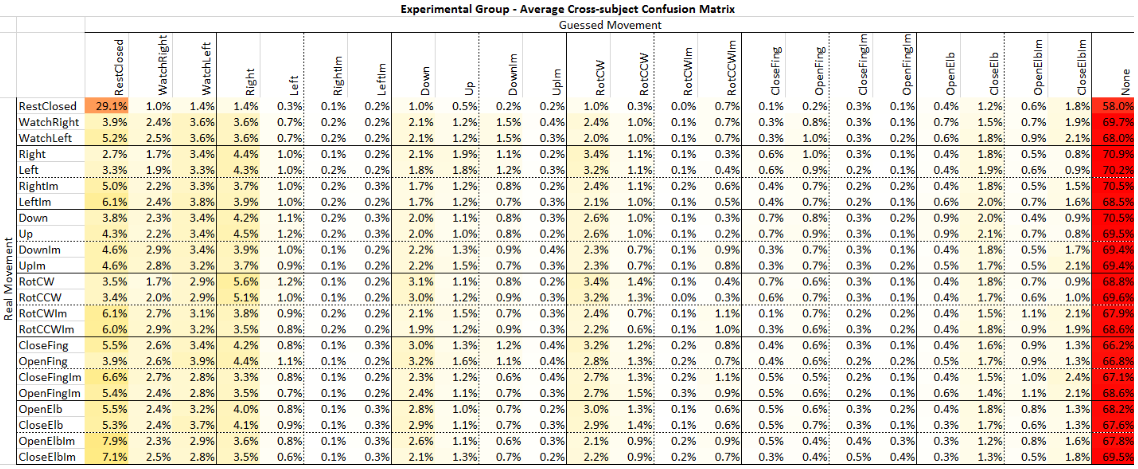
**Table** 3.10: Summary of the results in Table 3.9.

to another, and these patterns are in no way related to the movements being carried out, hence their appearance in both experimental and control groups.

The one significant difference between experimental and control groups is the correct identification of Eyes Closed Resting State. This means that at the very least, the master classifiers can effectively detect when a subject's eyes are closed.

## 3.5.3 Cross-subject

### 3.5.3.1 Experimental Group Results



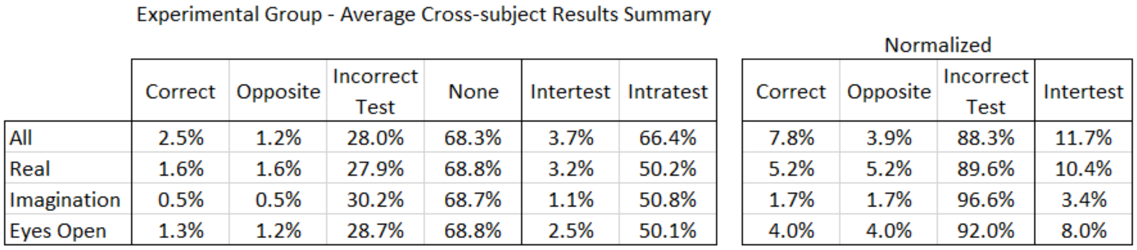
Experimental Group - Average Cross-subject Confusion Matrix

Guessed Movement

| Real Movement | RestClosed | WatchRight | WatchLeft | Right | Left | RightIm | LeftIm | Down | Up | DownIm | UpIm | RotCW | RotCCW | RotCWIm | RotCCWIm | CloseFing | OpenFing | CloseFingIm | OpenFingIm | OpenElb | CloseElb | OpenElbIm | CloseElbIm | None |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RestClosed | 29.1% | 1.0% | 1.4% | 1.4% | 0.3% | 0.1% | 0.2% | 1.0% | 0.5% | 0.2% | 0.2% | 1.0% | 0.3% | 0.0% | 0.7% | 0.1% | 0.2% | 0.3% | 0.1% | 0.4% | 1.2% | 0.6% | 1.8% | 58.0% |
| WatchRight | 3.9% | 2.4% | 3.6% | 3.6% | 0.7% | 0.2% | 0.2% | 2.1% | 1.2% | 1.5% | 0.4% | 2.4% | 1.0% | 0.1% | 0.7% | 0.3% | 0.8% | 0.3% | 0.1% | 0.7% | 1.5% | 0.7% | 1.9% | 69.7% |
| WatchLeft | 5.2% | 2.5% | 3.6% | 3.6% | 0.7% | 0.2% | 0.2% | 2.1% | 1.2% | 1.5% | 0.3% | 2.0% | 1.0% | 0.1% | 0.7% | 0.3% | 1.0% | 0.3% | 0.2% | 0.6% | 1.8% | 0.9% | 2.1% | 68.0% |
| Right | 2.7% | 1.7% | 3.4% | 4.4% | 1.0% | 0.1% | 0.2% | 2.1% | 1.9% | 1.1% | 0.2% | 3.4% | 1.1% | 0.1% | 0.3% | 0.6% | 1.0% | 0.3% | 0.1% | 0.4% | 1.8% | 0.5% | 0.8% | 70.9% |
| Left | 3.3% | 1.9% | 3.3% | 4.3% | 1.0% | 0.2% | 0.2% | 1.8% | 1.8% | 1.2% | 0.3% | 3.2% | 1.1% | 0.1% | 0.4% | 0.6% | 0.9% | 0.2% | 0.1% | 0.4% | 1.9% | 0.6% | 0.9% | 70.2% |
| RightIm | 5.0% | 2.2% | 3.3% | 3.7% | 1.0% | 0.2% | 0.3% | 1.7% | 1.2% | 0.8% | 0.2% | 2.4% | 1.1% | 0.2% | 0.6% | 0.4% | 0.7% | 0.2% | 0.2% | 0.4% | 1.8% | 0.5% | 1.5% | 70.5% |
| LeftIm | 6.1% | 2.4% | 3.8% | 3.9% | 1.0% | 0.2% | 0.2% | 1.7% | 1.2% | 0.7% | 0.3% | 2.1% | 1.0% | 0.1% | 0.5% | 0.4% | 0.7% | 0.2% | 0.1% | 0.6% | 2.0% | 0.7% | 1.6% | 68.5% |
| Down | 3.8% | 2.3% | 3.4% | 4.2% | 1.1% | 0.2% | 0.3% | 2.0% | 1.1% | 0.8% | 0.3% | 2.6% | 1.0% | 0.1% | 0.3% | 0.7% | 0.8% | 0.3% | 0.2% | 0.9% | 2.0% | 0.4% | 0.9% | 70.5% |
| Up | 4.3% | 2.2% | 3.4% | 4.5% | 1.2% | 0.2% | 0.3% | 2.0% | 1.0% | 0.8% | 0.2% | 2.6% | 1.0% | 0.1% | 0.2% | 0.7% | 0.9% | 0.3% | 0.1% | 0.9% | 2.1% | 0.7% | 0.8% | 69.5% |
| DownIm | 4.6% | 2.9% | 3.4% | 3.9% | 1.0% | 0.1% | 0.2% | 2.2% | 1.3% | 0.9% | 0.4% | 2.3% | 0.7% | 0.1% | 0.9% | 0.3% | 0.7% | 0.3% | 0.1% | 0.4% | 1.8% | 0.5% | 1.7% | 69.4% |
| UpIm | 4.6% | 2.8% | 3.2% | 3.7% | 0.9% | 0.1% | 0.2% | 2.2% | 1.5% | 0.7% | 0.3% | 2.3% | 0.7% | 0.1% | 0.8% | 0.3% | 0.7% | 0.3% | 0.2% | 0.5% | 1.7% | 0.5% | 2.1% | 69.4% |
| RotCW | 3.5% | 1.7% | 2.9% | 5.6% | 1.2% | 0.1% | 0.2% | 3.1% | 1.1% | 0.8% | 0.2% | 3.4% | 1.4% | 0.1% | 0.4% | 0.7% | 0.6% | 0.3% | 0.1% | 0.4% | 1.8% | 0.7% | 0.9% | 68.8% |
| RotCCW | 3.4% | 2.0% | 2.9% | 5.1% | 1.0% | 0.1% | 0.2% | 3.0% | 1.2% | 0.9% | 0.3% | 3.2% | 1.3% | 0.0% | 0.3% | 0.6% | 0.7% | 0.3% | 0.1% | 0.4% | 1.7% | 0.6% | 1.0% | 69.6% |
| RotCWIm | 6.1% | 2.7% | 3.1% | 3.8% | 0.9% | 0.2% | 0.2% | 2.1% | 1.5% | 0.7% | 0.3% | 2.4% | 0.7% | 0.1% | 1.1% | 0.1% | 0.7% | 0.2% | 0.2% | 0.4% | 1.5% | 1.1% | 2.1% | 67.9% |
| RotCCWIm | 6.0% | 2.9% | 3.2% | 3.5% | 0.8% | 0.2% | 0.2% | 1.9% | 1.2% | 0.9% | 0.3% | 2.2% | 0.6% | 0.1% | 1.0% | 0.3% | 0.6% | 0.3% | 0.2% | 0.4% | 1.8% | 0.9% | 1.9% | 68.6% |
| CloseFing | 5.5% | 2.6% | 3.4% | 4.2% | 0.8% | 0.1% | 0.3% | 3.0% | 1.3% | 1.2% | 0.4% | 3.2% | 1.2% | 0.2% | 0.8% | 0.4% | 0.6% | 0.3% | 0.1% | 0.4% | 1.6% | 0.9% | 1.3% | 66.2% |
| OpenFing | 3.9% | 2.6% | 3.9% | 4.4% | 1.1% | 0.1% | 0.2% | 3.2% | 1.6% | 1.1% | 0.4% | 2.8% | 1.3% | 0.2% | 0.7% | 0.4% | 0.6% | 0.2% | 0.2% | 0.5% | 1.7% | 0.9% | 1.3% | 66.8% |
| CloseFingIm | 6.6% | 2.7% | 2.8% | 3.3% | 0.8% | 0.1% | 0.2% | 2.3% | 1.2% | 0.6% | 0.4% | 2.7% | 1.3% | 0.2% | 1.1% | 0.5% | 0.5% | 0.2% | 0.1% | 0.4% | 1.5% | 1.0% | 2.4% | 67.1% |
| OpenFingIm | 5.4% | 2.4% | 2.8% | 3.5% | 0.7% | 0.1% | 0.2% | 2.4% | 1.1% | 0.7% | 0.3% | 2.7% | 1.5% | 0.3% | 0.9% | 0.5% | 0.6% | 0.2% | 0.1% | 0.6% | 1.4% | 1.1% | 2.1% | 68.6% |
| OpenElb | 5.5% | 2.4% | 3.2% | 4.0% | 0.8% | 0.1% | 0.3% | 2.8% | 1.0% | 0.7% | 0.2% | 3.0% | 1.3% | 0.1% | 0.6% | 0.5% | 0.6% | 0.3% | 0.2% | 0.4% | 1.8% | 0.8% | 1.3% | 68.2% |
| CloseElb | 5.3% | 2.4% | 3.7% | 4.1% | 0.9% | 0.1% | 0.3% | 2.9% | 1.1% | 0.7% | 0.3% | 2.9% | 1.4% | 0.1% | 0.6% | 0.5% | 0.7% | 0.3% | 0.1% | 0.3% | 1.7% | 0.6% | 1.3% | 67.6% |
| OpenElbIm | 7.9% | 2.3% | 2.9% | 3.6% | 0.8% | 0.1% | 0.3% | 2.6% | 1.1% | 0.6% | 0.3% | 2.1% | 0.9% | 0.2% | 0.9% | 0.5% | 0.4% | 0.4% | 0.3% | 0.3% | 1.2% | 0.8% | 1.6% | 67.8% |
| CloseElbIm | 7.1% | 2.5% | 2.8% | 3.5% | 0.6% | 0.1% | 0.3% | 2.1% | 1.3% | 0.7% | 0.2% | 2.2% | 0.9% | 0.2% | 0.7% | 0.3% | 0.4% | 0.5% | 0.4% | 0.3% | 1.3% | 0.5% | 1.8% | 69.5% |

**Table** 3.11: Average of 107 confusion matrices using training data from individual subjects and groups of subjects and testing data from all subjects not included in the training data.

When one subject or group of subjects is used for training and another is used for testing, the results are rather dismal. In Table 3.11 it can be seen that any resemblance of a diagonal line has disappeared. In fact, the results for the experimental group are worse than those for the control group, with the exception of Eyes Closed Resting State, which can be determined with 29.1% success rate. When normalized to exclude the None rate, this increases to 69.2%, which is excellent. Excluding RestClosed, the normalized Eyes Open success rate is 4.0%, which is lower than the 4.3% random chance.



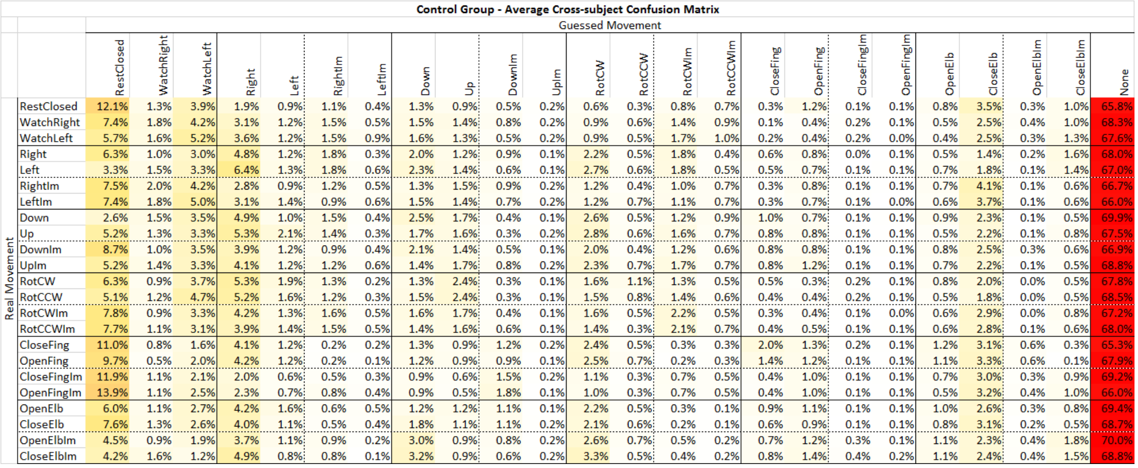
Experimental Group - Average Cross-subject Results Summary

| | Correct | Opposite | Incorrect Test | None | Intertest | Intratest | Normalized Correct | Normalized Opposite | Normalized Incorrect Test | Normalized Intertest |
|---|---|---|---|---|---|---|---|---|---|---|
| All | 2.5% | 1.2% | 28.0% | 68.3% | 3.7% | 66.4% | 7.8% | 3.9% | 88.3% | 11.7% |
| Real | 1.6% | 1.6% | 27.9% | 68.8% | 3.2% | 50.2% | 5.2% | 5.2% | 89.6% | 10.4% |
| Imagination | 0.5% | 0.5% | 30.2% | 68.7% | 1.1% | 50.8% | 1.7% | 1.7% | 96.6% | 3.4% |
| Eyes Open | 1.3% | 1.2% | 28.7% | 68.8% | 2.5% | 50.1% | 4.0% | 4.0% | 92.0% | 8.0% |

**Table** 3.12: Summary of the results in Table 3.11.

In conclusion, a master classifier created using data for one group of subjects can effectively determine if a feature vector from a subject outside the training group corresponds to Eyes Closed Resting State with a high degree of accuracy. However, if it does not correspond to this state, the classifier cannot determine better than chance which move was made.

### 3.5.3.2 Control Group Results



Control Group - Average Cross-subject Confusion Matrix

Guessed Movement

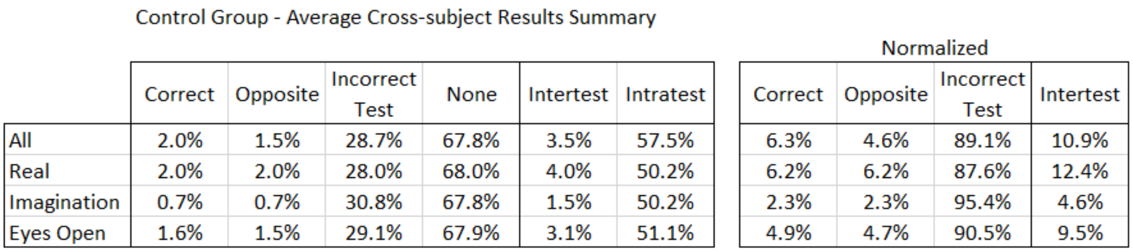
| | | RestClosed | WatchRight | WatchLeft | Right | Left | RightIm | LeftIm | Down | Up | DownIm | UpIm | RotCW | RotCCW | RotCWIm | RotCCWIm | CloseFing | OpenFing | CloseFingIm | OpenFingIm | OpenElb | CloseElb | OpenElbIm | CloseElbIm | None |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Real Movement | RestClosed | 12.1% | 1.3% | 3.9% | 1.9% | 0.9% | 1.1% | 0.4% | 1.3% | 0.9% | 0.5% | 0.2% | 0.6% | 0.3% | 0.8% | 0.7% | 0.3% | 1.2% | 0.1% | 0.1% | 0.8% | 3.5% | 0.3% | 1.0% | 65.8% |
| | WatchRight | 7.4% | 1.8% | 4.2% | 3.1% | 1.2% | 1.5% | 0.5% | 1.5% | 1.4% | 0.8% | 0.2% | 0.9% | 0.6% | 1.4% | 0.9% | 0.1% | 0.4% | 0.2% | 0.1% | 0.5% | 2.5% | 0.4% | 1.0% | 68.3% |
| | WatchLeft | 5.7% | 1.6% | 5.2% | 3.6% | 1.2% | 1.5% | 0.9% | 1.6% | 1.3% | 0.5% | 0.2% | 0.9% | 0.5% | 1.7% | 1.0% | 0.2% | 0.4% | 0.2% | 0.0% | 0.4% | 2.5% | 0.3% | 1.3% | 67.6% |
| | Right | 6.3% | 1.0% | 3.0% | 4.8% | 1.2% | 1.8% | 0.3% | 2.0% | 1.2% | 0.9% | 0.1% | 2.2% | 0.5% | 1.8% | 0.4% | 0.6% | 0.8% | 0.0% | 0.1% | 0.5% | 1.4% | 0.2% | 1.6% | 68.0% |
| | Left | 3.3% | 1.5% | 3.3% | 6.4% | 1.3% | 1.8% | 0.6% | 2.3% | 1.4% | 0.6% | 0.1% | 2.7% | 0.6% | 1.8% | 0.5% | 0.5% | 0.7% | 0.1% | 0.1% | 0.7% | 1.8% | 0.1% | 1.4% | 67.0% |
| | RightIm | 7.5% | 2.0% | 4.2% | 2.8% | 0.9% | 1.2% | 0.5% | 1.3% | 1.5% | 0.9% | 0.2% | 1.2% | 0.4% | 1.0% | 0.7% | 0.3% | 0.8% | 0.1% | 0.1% | 0.7% | 4.1% | 0.1% | 0.6% | 66.7% |
| | LeftIm | 7.4% | 1.8% | 5.0% | 3.1% | 1.4% | 0.9% | 0.6% | 1.5% | 1.4% | 0.7% | 0.2% | 1.2% | 0.7% | 1.1% | 0.7% | 0.3% | 0.7% | 0.1% | 0.0% | 0.6% | 3.7% | 0.1% | 0.6% | 66.0% |
| | Down | 2.6% | 1.5% | 3.5% | 4.9% | 1.0% | 1.5% | 0.4% | 2.5% | 1.7% | 0.4% | 0.1% | 2.6% | 0.5% | 1.2% | 0.9% | 1.0% | 0.7% | 0.1% | 0.1% | 0.9% | 2.3% | 0.1% | 0.5% | 69.9% |
| | Up | 5.2% | 1.3% | 3.3% | 5.3% | 2.1% | 1.4% | 0.3% | 1.7% | 1.6% | 0.3% | 0.2% | 2.8% | 0.6% | 1.6% | 0.7% | 0.8% | 0.8% | 0.1% | 0.1% | 0.5% | 2.2% | 0.1% | 0.8% | 67.5% |
| | DownIm | 8.7% | 1.0% | 3.5% | 3.9% | 1.2% | 0.9% | 0.4% | 2.1% | 1.4% | 0.5% | 0.1% | 2.0% | 0.4% | 1.2% | 0.6% | 0.8% | 0.8% | 0.1% | 0.1% | 0.8% | 2.5% | 0.3% | 0.6% | 66.9% |
| | UpIm | 5.2% | 1.4% | 3.3% | 4.1% | 1.2% | 1.2% | 0.6% | 1.4% | 1.7% | 0.8% | 0.2% | 2.3% | 0.7% | 1.7% | 0.7% | 0.8% | 1.2% | 0.1% | 0.1% | 0.7% | 2.2% | 0.1% | 0.5% | 68.8% |
| | RotCW | 6.3% | 0.9% | 3.7% | 5.3% | 1.9% | 1.3% | 0.2% | 1.3% | 2.4% | 0.3% | 0.1% | 1.6% | 1.1% | 1.3% | 0.5% | 0.5% | 0.3% | 0.2% | 0.1% | 0.8% | 2.0% | 0.0% | 0.5% | 67.8% |
| | RotCCW | 5.1% | 1.2% | 4.7% | 5.2% | 1.6% | 1.2% | 0.3% | 1.5% | 2.4% | 0.3% | 0.1% | 1.5% | 0.8% | 1.4% | 0.6% | 0.4% | 0.4% | 0.2% | 0.1% | 0.5% | 1.8% | 0.0% | 0.5% | 68.5% |
| | RotCWIm | 7.8% | 0.9% | 3.3% | 4.2% | 1.3% | 1.6% | 0.5% | 1.6% | 1.7% | 0.4% | 0.1% | 1.6% | 0.5% | 2.2% | 0.5% | 0.3% | 0.4% | 0.1% | 0.0% | 0.6% | 2.9% | 0.0% | 0.8% | 67.2% |
| | RotCCWIm | 7.7% | 1.1% | 3.1% | 3.9% | 1.4% | 1.5% | 0.5% | 1.4% | 1.6% | 0.6% | 0.1% | 1.4% | 0.3% | 2.1% | 0.7% | 0.4% | 0.5% | 0.1% | 0.1% | 0.6% | 2.8% | 0.1% | 0.6% | 68.0% |
| | CloseFing | 11.0% | 0.8% | 1.6% | 4.1% | 1.2% | 0.2% | 0.2% | 1.3% | 0.9% | 1.2% | 0.2% | 2.4% | 0.5% | 0.3% | 0.3% | 2.0% | 1.3% | 0.2% | 0.1% | 1.2% | 3.1% | 0.6% | 0.3% | 65.3% |
| | OpenFing | 9.7% | 0.5% | 2.0% | 4.2% | 1.2% | 0.2% | 0.1% | 1.2% | 0.9% | 0.9% | 0.1% | 2.5% | 0.7% | 0.2% | 0.3% | 1.4% | 1.2% | 0.1% | 0.1% | 1.1% | 3.3% | 0.6% | 0.1% | 67.9% |
| | CloseFingIm | 11.9% | 1.1% | 2.1% | 2.0% | 0.6% | 0.5% | 0.3% | 0.9% | 0.6% | 1.5% | 0.2% | 1.1% | 0.3% | 0.7% | 0.5% | 0.4% | 1.0% | 0.1% | 0.1% | 0.7% | 3.0% | 0.3% | 0.9% | 69.2% |
| | OpenFingIm | 13.9% | 1.1% | 2.5% | 2.3% | 0.7% | 0.8% | 0.4% | 0.9% | 0.5% | 1.8% | 0.1% | 1.0% | 0.3% | 0.7% | 0.5% | 0.4% | 1.0% | 0.1% | 0.1% | 0.5% | 3.2% | 0.4% | 1.0% | 66.0% |
| | OpenElb | 6.0% | 1.1% | 2.7% | 4.2% | 1.6% | 0.6% | 0.5% | 1.2% | 1.2% | 1.1% | 0.1% | 2.2% | 0.5% | 0.3% | 0.1% | 0.9% | 1.1% | 0.1% | 0.1% | 1.0% | 2.6% | 0.3% | 0.8% | 69.4% |
| | CloseElb | 7.6% | 1.3% | 2.6% | 4.0% | 1.1% | 0.5% | 0.4% | 1.8% | 1.1% | 1.1% | 0.2% | 2.1% | 0.6% | 0.2% | 0.1% | 0.6% | 0.9% | 0.1% | 0.1% | 0.8% | 3.1% | 0.2% | 0.5% | 68.7% |
| | OpenElbIm | 4.5% | 0.9% | 1.9% | 3.7% | 1.1% | 0.9% | 0.2% | 3.0% | 0.9% | 0.8% | 0.2% | 2.6% | 0.7% | 0.5% | 0.2% | 0.7% | 1.2% | 0.3% | 0.1% | 1.1% | 2.3% | 0.4% | 1.8% | 70.0% |
| | CloseElbIm | 4.2% | 1.6% | 1.2% | 4.9% | 0.8% | 0.8% | 0.1% | 3.2% | 0.9% | 0.6% | 0.2% | 3.3% | 0.5% | 0.4% | 0.2% | 0.8% | 1.4% | 0.4% | 0.2% | 1.1% | 2.4% | 0.4% | 1.5% | 68.8% |

**Table** 3.13: Average of 32 confusion matrices with the same training subjects as used in Table X.9, but using 2 control subjects for testing data.

The confusion matrix in which the test data was taken from the control group looks very similar to the one for the experimental group, except that RestClosed is more often selected as the correct movement for the control group. Since all tests for the control group were really Eyes Closed Resting State, RestClosed is in fact the correct answer.

## 3.5.4 Cross-subject training group size

For cross-subject evaluations, we created many master classifiers using different groups of subjects as training data, and were curious to see if the number of subjects used had any effect on success rate. Figure 3.7 shows the normalized success rates for training groups between 1 and 6 subjects. In each case, the size of the test group was 9-n, where n is the number of subjects in the training



Control Group - Average Cross-subject Results Summary

| | Correct | Opposite | Incorrect Test | None | Intertest | Intratest | Normalized Correct | Normalized Opposite | Normalized Incorrect Test | Normalized Intertest |
|---|---|---|---|---|---|---|---|---|---|---|
| All | 2.0% | 1.5% | 28.7% | 67.8% | 3.5% | 57.5% | 6.3% | 4.6% | 89.1% | 10.9% |
| Real | 2.0% | 2.0% | 28.0% | 68.0% | 4.0% | 50.2% | 6.2% | 6.2% | 87.6% | 12.4% |
| Imagination | 0.7% | 0.7% | 30.8% | 67.8% | 1.5% | 50.2% | 2.3% | 2.3% | 95.4% | 4.6% |
| Eyes Open | 1.6% | 1.5% | 29.1% | 67.9% | 3.1% | 51.1% | 4.9% | 4.7% | 90.5% | 9.5% |

**Table** 3.14: Summary of the results in Table 3.13.

group. For the control group, shown in red, the test size was always 2. There is no clear relationship between training group size and success rate. The results for the experimental group are better than those for the control group in all cases, but as we saw in the previous section, this is due to the correct identification of Eyes Closed Resting State.

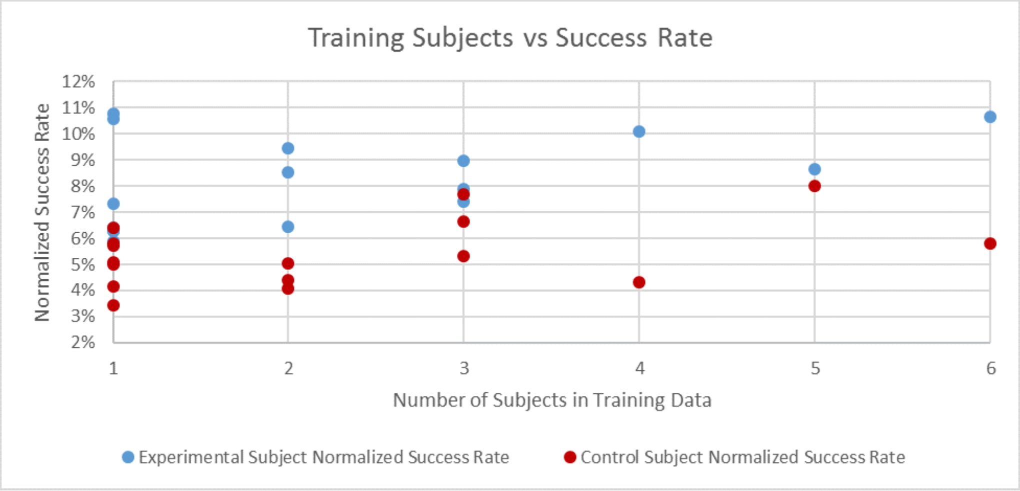
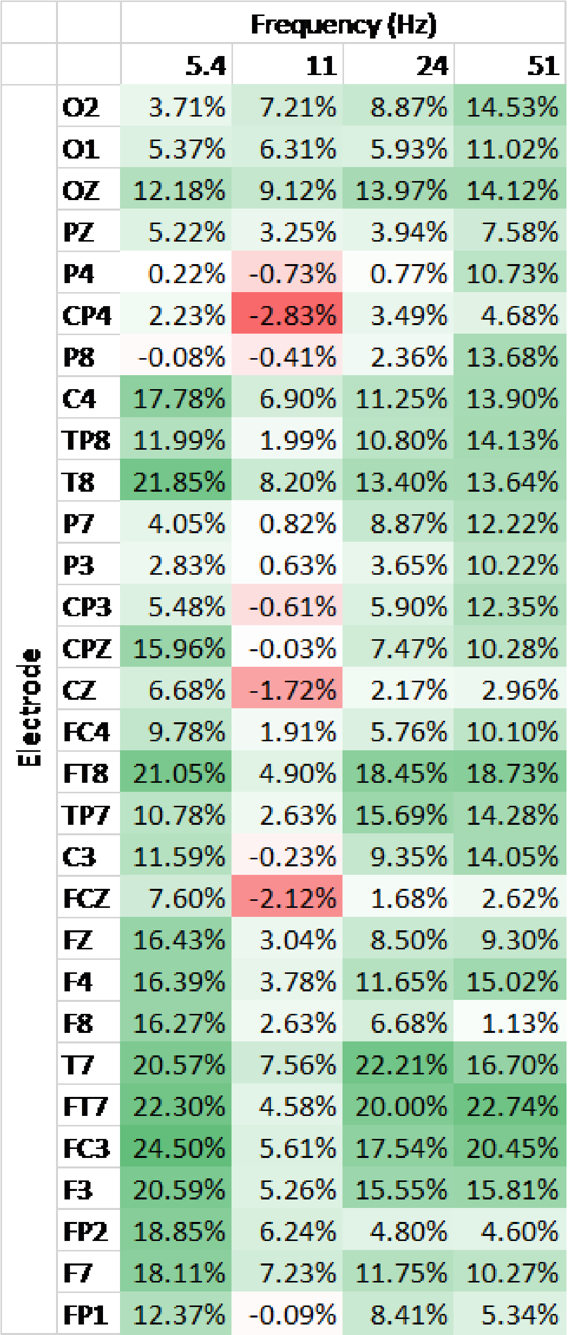


**Figure** 3.7: Scatter plot showing the relationship between the number of subjects used for training data and the normalized master classifier success rate, for both experimental and control groups.

## 3.6 Key Components and Key Component Classifiers

One of the aims of this thesis was to identify the key brain regions and frequencies which show an increase or decrease in activity when one movement is performed, but not when others are performed. Since each electrode reads activity from a specific brain region, this problem is analogous to identifying the components of the feature vector that most improve the success rate of all binary classifiers which include any given movement. We therefore identified the key components for each binary classifier, then determined the average effect of all binary classifiers containing each movement.



| | | Frequency (Hz) | | | |
|---|---|---|---|---|---|
| | | 5.4 | 11 | 24 | 51 |
| Electrode | O2 | 3.71% | 7.21% | 8.87% | 14.53% |
| | O1 | 5.37% | 6.31% | 5.93% | 11.02% |
| | OZ | 12.18% | 9.12% | 13.97% | 14.12% |
| | PZ | 5.22% | 3.25% | 3.94% | 7.58% |
| | P4 | 0.22% | -0.73% | 0.77% | 10.73% |
| | CP4 | 2.23% | -2.83% | 3.49% | 4.68% |
| | P8 | -0.08% | -0.41% | 2.36% | 13.68% |
| | C4 | 17.78% | 6.90% | 11.25% | 13.90% |
| | TP8 | 11.99% | 1.99% | 10.80% | 14.13% |
| | T8 | 21.85% | 8.20% | 13.40% | 13.64% |
| | P7 | 4.05% | 0.82% | 8.87% | 12.22% |
| | P3 | 2.83% | 0.63% | 3.65% | 10.22% |
| | CP3 | 5.48% | -0.61% | 5.90% | 12.35% |
| | CPZ | 15.96% | -0.03% | 7.47% | 10.28% |
| | CZ | 6.68% | -1.72% | 2.17% | 2.96% |
| | FC4 | 9.78% | 1.91% | 5.76% | 10.10% |
| | FT8 | 21.05% | 4.90% | 18.45% | 18.73% |
| | TP7 | 10.78% | 2.63% | 15.69% | 14.28% |
| | C3 | 11.59% | -0.23% | 9.35% | 14.05% |
| | FCZ | 7.60% | -2.12% | 1.68% | 2.62% |
| | FZ | 16.43% | 3.04% | 8.50% | 9.30% |
| | F4 | 16.39% | 3.78% | 11.65% | 15.02% |
| | F8 | 16.27% | 2.63% | 6.68% | 1.13% |
| | T7 | 20.57% | 7.56% | 22.21% | 16.70% |
| | FT7 | 22.30% | 4.58% | 20.00% | 22.74% |
| | FC3 | 24.50% | 5.61% | 17.54% | 20.45% |
| | F3 | 20.59% | 5.26% | 15.55% | 15.81% |
| | FP2 | 18.85% | 6.24% | 4.80% | 4.60% |
| | F7 | 18.11% | 7.23% | 11.75% | 10.27% |
| | FP1 | 12.37% | -0.09% | 8.41% | 5.34% |

**Table** 3.15: Example of component effects for the Down vs RestClosed binary classifier for one subject. Each column shows the effects of one frequency, and each row corresponds to one electrode. The numbers in the matrix show the percentage of increase in the binary classifier success rate when the wavelet coefficient for the corresponding frequency and electrode is included in the reduced feature vector.

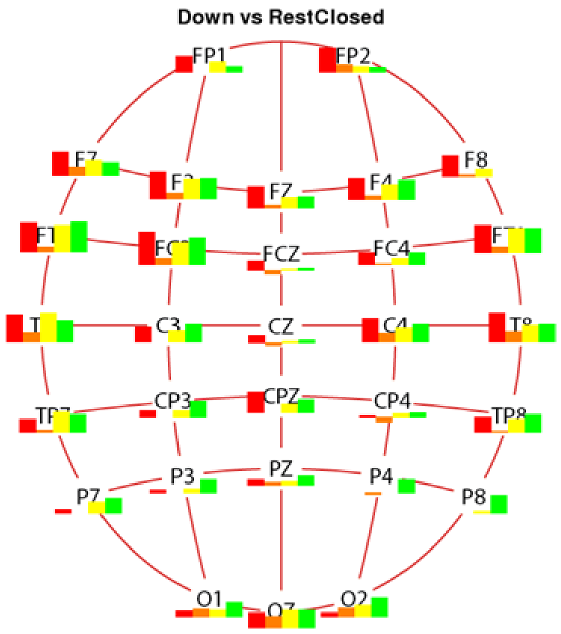


**Figure** 3.8: Left shows a geometrical representation of the information provided in Table Y, where each electrode is placed in its relative location on the scalp. The top of the graph represents the front of the head. The height of the Red bar represents 5.4 Hz (Theta), Orange: 11 Hz (Alpha), Yellow: 24 (Beta) Hz, and Green: 51 Hz (Gamma). The Right graph highlights the 10 electrode-frequency combinations with the highest effect.

## 3.7 4-movement Key Components

A series of simpler analyses were run in which only four movements were used: Right, Left, Down and Up. The idea was that if a classifier could be created that successfully determined between these four movements, it could be applied to allow a user to navigate a 2-D environment. Furthermore, it much easier to visualize the data in these tests than for the 23-movement tests.

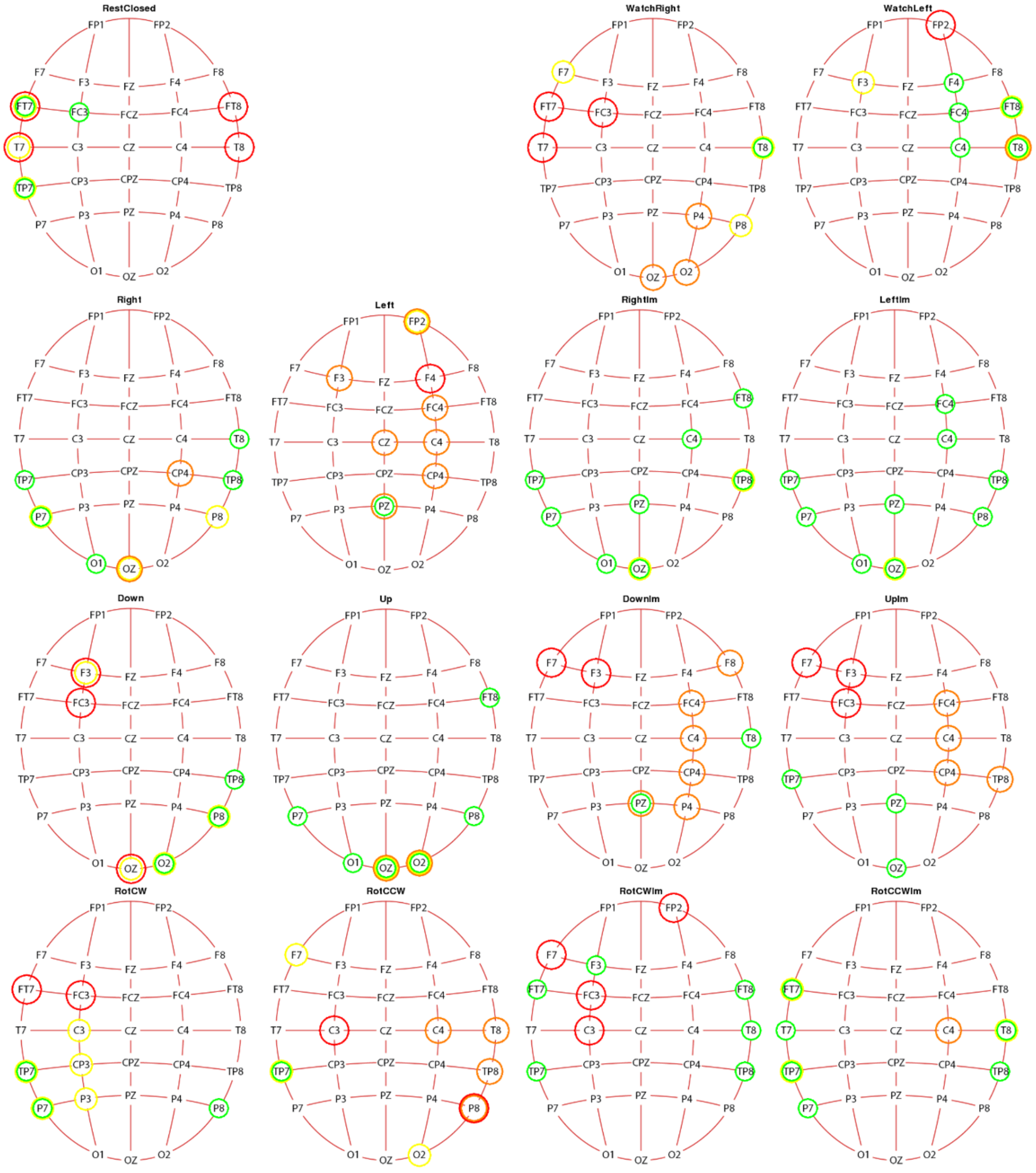
For all tests, the same four subjects were used for training, and five pairs of subjects were used for testing, including two subsets of the training group (essentially self-validation), an optimization group, an independent test group, and the control group.

Initially, reduced feature vectors were created using the top ten components as determined by separation index. Subsequently, the top ten key components were determined, as described in the previous section, but optimized for their effect on the success rate when using data from the optimization group.

### 3.7.1 Key Component Evolution

After determining the key components, a new generation of classifier functions was created which used the top 10 key components in their reduced feature vectors, rather than the top 10 components as determined by separation index.

We would expect this process to result in an increased success rate for the optimization group, and if the key components selected really contained information pertaining to the movement, then


RestClosed
WatchRight
WatchLeft
Right
Left
RightIm
LeftIm
Down
Up
DownIm
UpIm
RotCW
RotCCW
RotCWIm
RotCCWIm
FP1
FP2
F7
F3
FZ
F4
F8
FT7
FC3
FCZ
FC4
FT8
T7
C3
CZ
C4
T8
TP7
CP3
CPZ
CP4
TP8
P7
P3
PZ
P4
P8
O1
OZ
O2

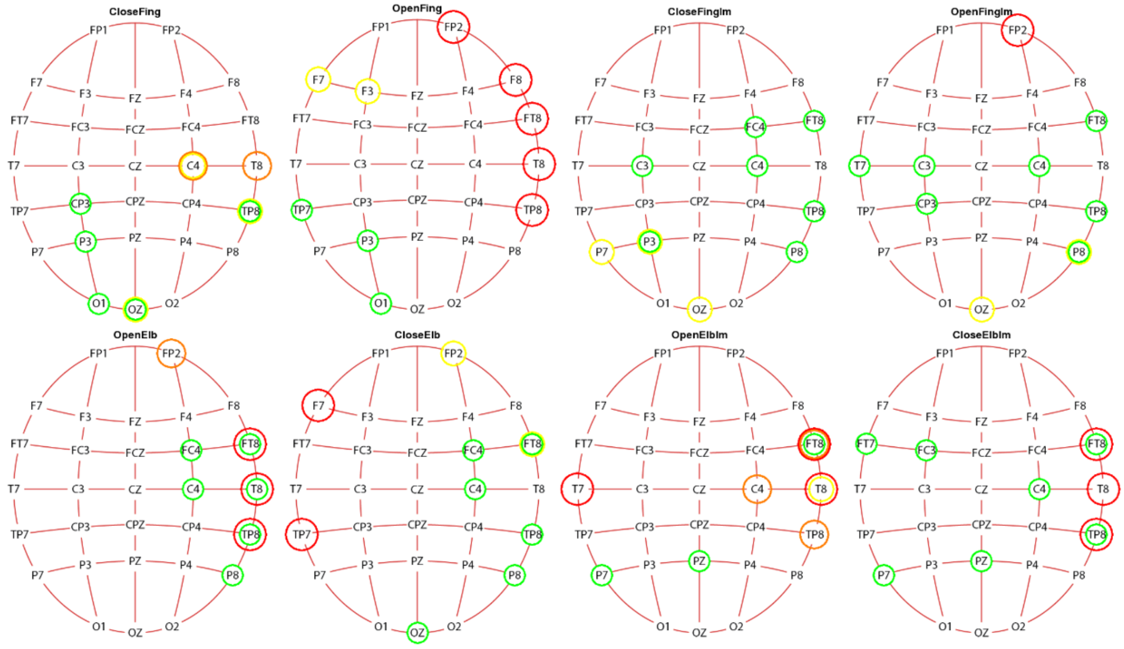


**Figure** 3.9: Graphs showing the 10 electrodes and frequencies which have the highest positive effect on success rate during cross-subject trials for each movement. The colors of the circles around each electrode indicate frequency, as follows: Red: 5.4 Hz (Theta), Orange: 11 Hz (Alpha), Yellow: 24 (Beta) Hz, Green: 51 Hz (Gamma).

the independent group should also show an improvement which is greater than that of the control group.

This process was iterated four times, with each generation using the components that were determined to have the greatest effect by the previous generation.

It can be seen in Figure 3.12 that the first generation classifier did in fact improve the success rate of the optimization group by 8.2% (9.3% normalized). However, the initial success rate for the optimization group was astonishingly low (far below that of random chance), and the first generation only improved this rate up to that of random chance. Increases were also seen in the success rates of the independent and control groups, but both remain below random chance. Repeated iterations did not have any clear effect on the success rates beyond generation 2, and the independent and control groups had success rates very near that of random chance (25%) independently of the generation.

We should bear in mind that the method for determining key components involved first determining the top components based on separation index, and then determining the change in success rate when new components were added or elements of that list were removed. Therefore, the key components found were those that best complement the top separation index components. We discovered that they did not necessarily complement each other, however. In further generations, this difficulty remained.

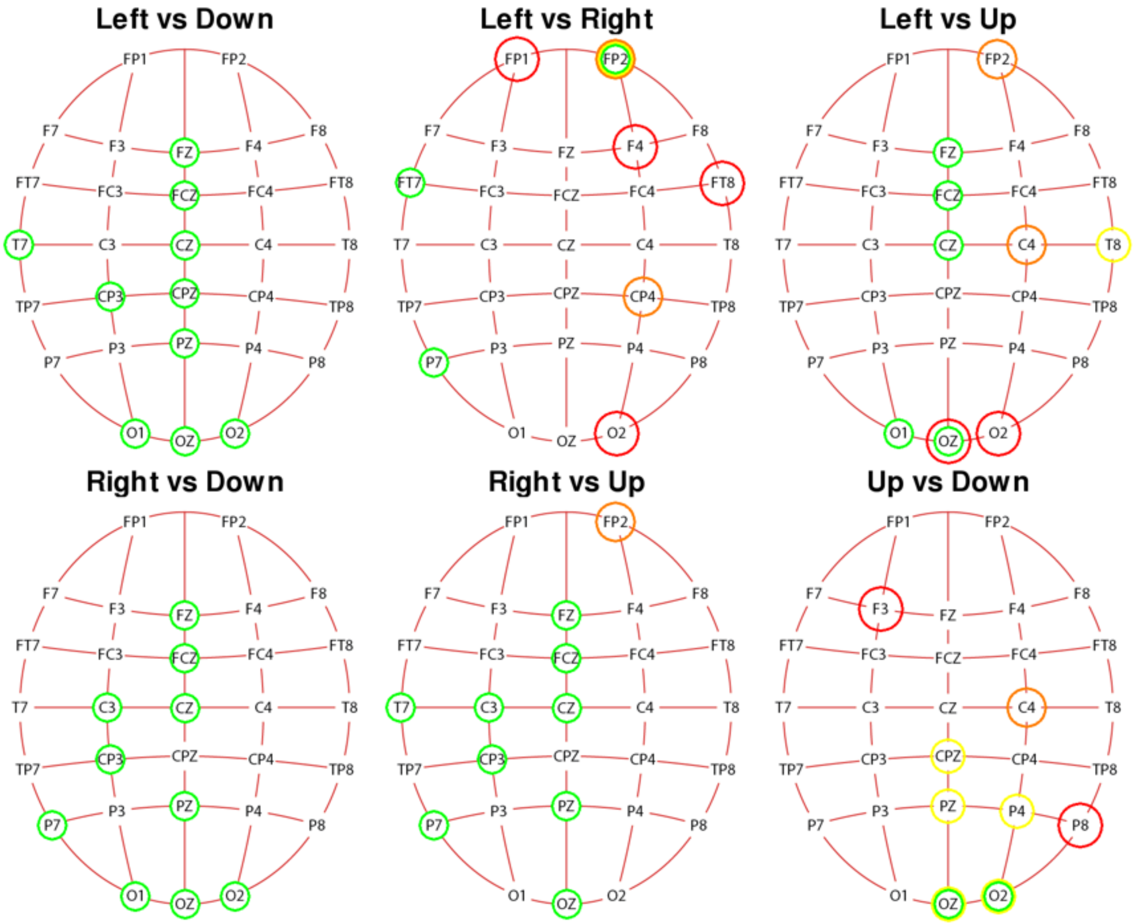


**Figure** 3.10: Top 10 key component diagrams for each of the six binary classifiers necessary to create a 4-movement master classifier. The colors of the circles around each electrode indicate frequency, as follows: Red: 5.4 Hz (Theta), Orange: 11 Hz (Alpha), Yellow: 24 (Beta) Hz, Green: 51 Hz (Gamma).

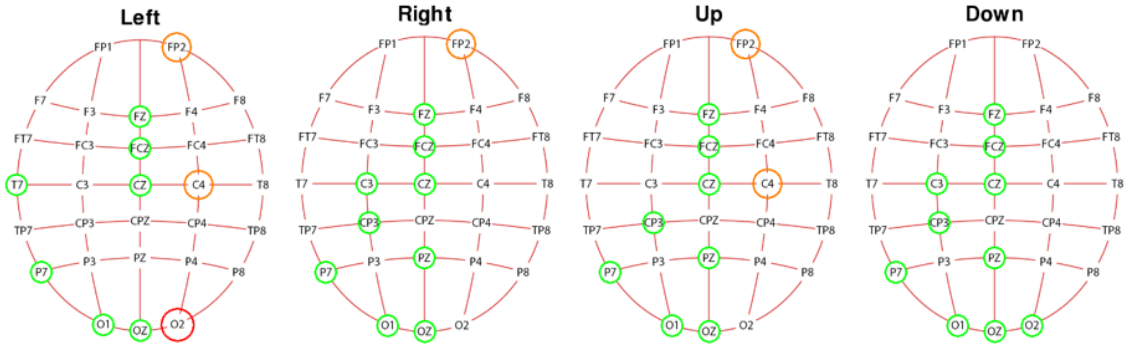


**Figure** 3.11: Top 10 key component diagrams for each of the four movements, creating by averaging the key component results for each of the three binary classifiers that include each movement.

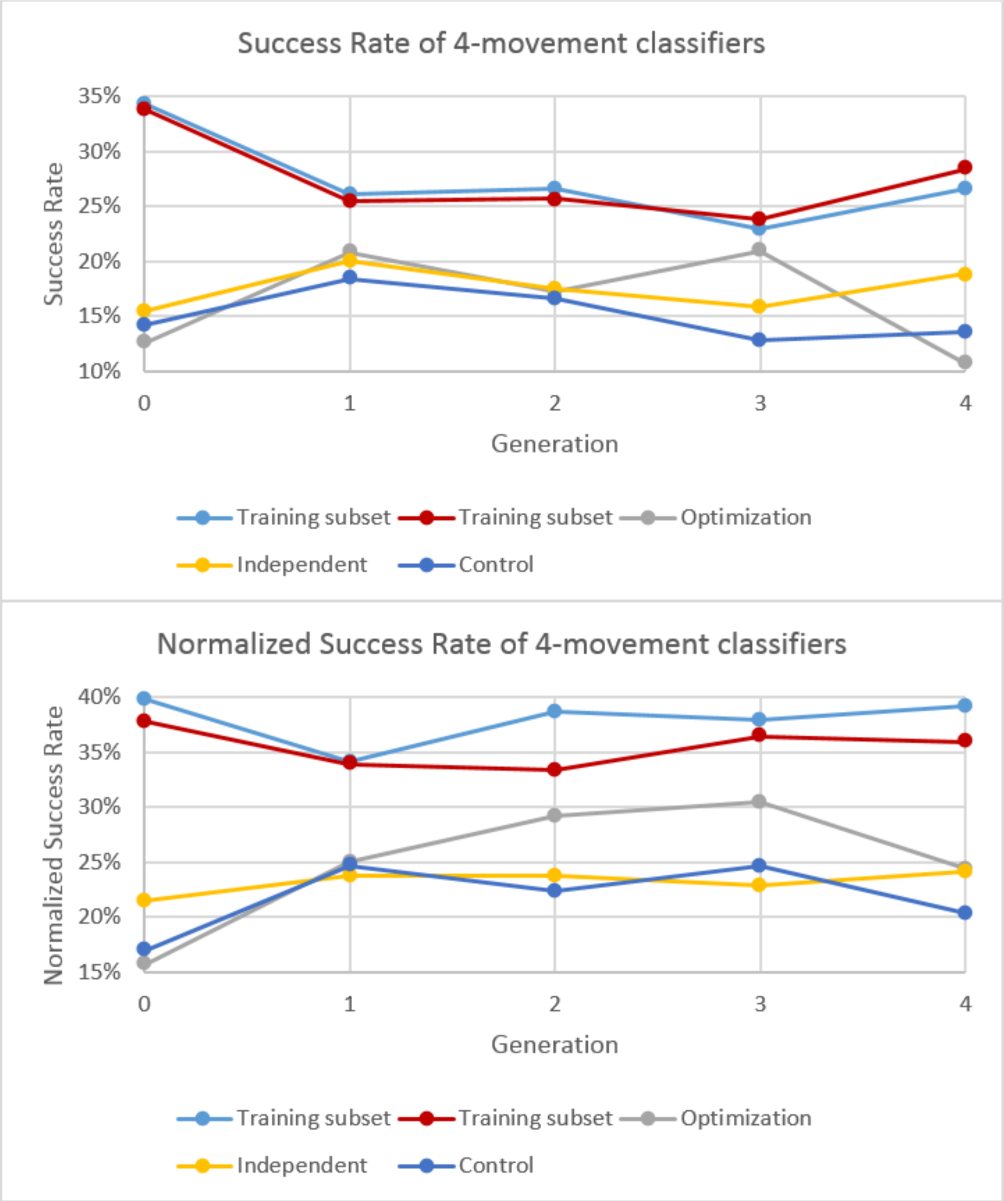


**Figure** 3.12: The effect of the classifier generation on success rate. Subjects 14,15,16,17 were used for training and 18,19 for optimization. The top graph shows master classifier success rate, and the bottom graph shows the normalized master classifier success rate, where None results are ignored.

# Chapter 4

# Conclusions

## 4.1 Key Points

1. A pattern recognition algorithm was successfully developed which can assign a feature vector to one of any number of states.

   23-movement master classifier functions can be evaluated in an average of 4.8 milliseconds on a single CPU at 2.8 GHz, allowing it to easily be used in real time. 4-movement master classifiers can be evaluated in as little as 0.07 ms.

   Unfortunately, the success rate of the master classifiers for determining the correct movement using EEG experimental data was not significantly higher than for control data. This indicates that although the classifiers did identify patterns, those patterns were not related to the movement signal we had hoped to identify, with the exception of Eyes Closed Resting State. The master classifiers show a high degree of accuracy when detecting which test was performed, but much lower accuracy at determining the correct movement within each test.

   In part this is due to EEG's low spatial resolution. Since the motor cortex, which controls conscious movements, is covered by only a couple of electrodes, it is difficult to detect a difference between very similar movements.

   Nevertheless, the algorithm created could be applied to other problems or to cleaner BCI data such as those obtained from ECoG recordings.

2. A fast Wavelet Transform algorithm was developed which reduces the time taken to perform by three orders of magnitude, thus overcoming a tremendous computational obstacle. No comparison has been made with commercial software to determine if this algorithm is an improvement on existing knowledge.

3. A method was developed for determining the effect of the inclusion of each component of a feature vector in a reduced feature vector. Furthermore, the knowledge of key components can be implemented to create improved second generation classifiers. Unfortunately, due to the ineffectiveness of the master classifiers, it is difficult to determine how important the key components identified really are.

4. A graphing module was developed for python which creates scatter plots, line graphs, histograms and box-and-whiskers plots, as well as 3-d graphs. The only prerequisite for the module is pygame.

5. Using a sample set of binary classifiers, it was determined that a separation index of above 1.4 indicates that the classifier can effectively determine between two states. Below 1.4, there is no guarantee of the effectiveness of the binary classifier. It would be interesting to determine if this tendency exists with other types of data as well.

6. The confusion matrix results show a clear example of the importance of the relationship between training and test data in pattern classification problems. It is possible to obtain very high success rates if the training and test data are the same. However, this has absolutely no bearing on the actual effectiveness of the classifier. Self-validation is like asking a mother if her son is handsome.

## 4.2 Considerations

The following assumptions and considerations were made during the design of this thesis which play an important role in the experimental design and implementation:

1. We assume that a "move" signal exists in the cortex, which is constant during the entire process of arm movement. This is not necessarily the case, however. There could be instead "start moving" and "stop moving" signals that are sent from the cortex, but the control to continue moving between those two signals could reside in a lower brain region or in the spinal column.

2. It is known that each hemisphere of the brain controls the contralateral side of the body. Since we study only right arm movements, we would expect the activity to be primarily in the left hemisphere. However, it has been seen that split-brain subjects can in fact move both their contralateral and their ipsilateral arm with either hemisphere but only their contralateral hand. It is also known that motor activity is primarily controlled by the motor cortex (denoted by C in the electrode configuration) and that a reduction in alpha waves is associated with muscular movement. This indicates that we would expect alpha frequencies in electrode C3, and possible C4, to be the primary indicators of physical movement. However, one or two electrodes and a single frequency are not enough to effectively discriminate between 23 possible movements, 10 of which are actually involve physical action. We therefore apply no bias on what electrodes and frequencies will be taken into account based on previous knowledge.

3. The determination of a wavelet coefficient requires the evaluation of an integral from negative infinite to positive infinite time. For practical purposes, this is impossible and the integral must be evaluated in some finite time period. In this thesis, we chose to integrate in a time frame between $\tau - 4s$ and $\tau + 4s$, where $\tau$ and $s$ are the parameters for the wavelet function (time displacement and scale, respectively). Because the morlet mother wavelet tends to zero as time tends to plus or minus infinity, over 99.99% of the wave information is contained

within these bounds. (Note: if the bounds are set to $\tau \pm 3s$, 99.73% of the information is contained, for $\tau \pm 2s$, it is roughly 95.45% and for $\tau \pm s$, it is around 68.27%). This implies two inherent difficulties:

1. The beginning and end of any time series to be studied with wavelet transform do not provide reliable wavelet coefficients, because part of the mother wavelet is "cut off" by the lack of time series data. How much of the beginning and end depends on the scale. The delta frequency range varies from 0.2 to 3.5 Hz, which corresponds to scales of .286-5 s.

   This means, if we use a $\tau \pm 4s$ time period, the first and last 20 seconds of data do not provide accurate information. Considering that our data is only 30 seconds in length, this means that none of it is reliable at a scale of 5. For the second lowest frequency range, theta, the maximum scale is 0.25 s, meaning that only the first and last second of the data set provide inaccurate information.

2. The primary application of this research is for the brainwaves detected to be able to control something, be it a prosthetic arm or a computer. This means that the time between the moment a command is sent and the moment the response is made is crucial. After a signal is sent, it must be read by the EEG, then wavelet transforms must be performed, a feature vector created, and a classifier evaluated to determine what command was given. Then that command must be carried out by the arm or computer. Even if the hardware and algorithms were perfect, carrying these operations out in almost no time, it would still be necessary to wait until 4 times the scale seconds have passed before the wavelet analysis can even be performed. Again, for low frequency beta waves, this amounts to 20 seconds. If a friend 20 meters away threw a Frisbee to you at 14 m/s, without going into the dynamics of Frisbee flight, you would have to respond about 18 seconds before your friend even began to throw. With practice, you may be able to catch a Frisbee, throw it and catch it again in 20 seconds, allowing you to always be responding to the previous throw, but it would be difficult to set up your entire life in 20 second cycles. Anyone who has played online computer games can tell you that a time lag of over 100 ms between command time and response time is noticeable, above 250 ms is annoying and with a lag over 1 s, it is practically impossible to do anything.

Due to this inconvenience of real-time wavelet transform evaluation, we decided against including beta wave oscillations.

## 4.3 Author's Note

From the beginning, I wanted to have a complete knowledge and control of all the mathematics and programming involved in my thesis. Many libraries already exist which allow a programmer to use the math without really having any idea of what it means, but my idea was that if I understood all the math well enough to program it myself, then I would be able to improve it. So, I went ahead and wrote every single part of the code on my own in python 2.7, from the program that gives instructions during the experiment and the graphing library to the mathematical methods and complex analysis processes. For the sake of fair attribution, I want to clarify that my graphing library does use pygame for opening windows and manipulating pixel colors, and I eventually included numpy for its speedy vector and matrix operations, as well as for loading large amounts of data. Openpyxl is the final third-party module I used, which optionally allows data to be saved in excel format rather than text.

Initially, analysis time was a serious obstacle. Each wavelet scalogram took hours to complete. The first method of data analysis attempted was wavelet coherence, which was abandoned because each graph would have taken months to generate. In order to overcome this obstacle, a number of months was dedicated to developing software that would analyze data in parallel and periodically save progress so that in the event of an error, the entire analysis would not be lost. A scheduler program was also developed so that computers could be left running for weeks, performing one analysis after another without human involvement. The parallelization phase ended when new algorithms were developed (described in section 4.1) which reduced processing time by almost three orders of magnitude.

The next method attempted was to perform a continuous wavelet transform to create a scalogram with different scales on the vertical axis and time displacements on the horizontal axis, then average all the column vectors corresponding to different time displacements to obtain a single spectrum for each movement. These spectra were then used as templates to compare against the individual column vectors at any given time and determine which template spectrum was most similar using convolutions.

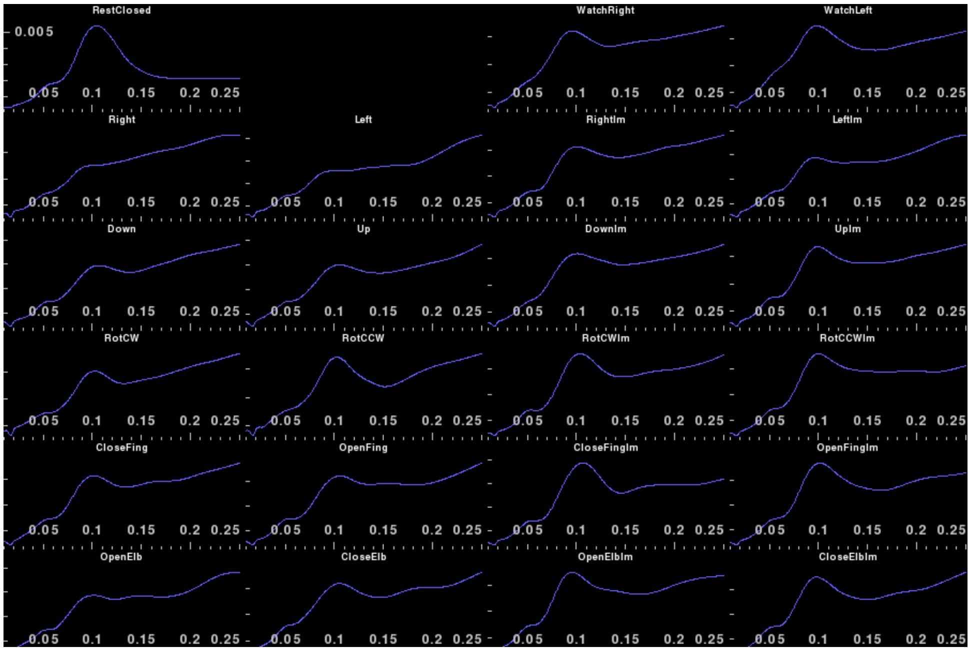


**Figure** 4.1: Example spectra of all 23 movements for electrode C3 of a single subject.

I tried to improve on this method by adjusting my definition of convolution between spectra. Rather than being a simple dot-product between two vectors iAiBi , where A is the spectra at time t and B is one of the comparison template spectra, I made it a weighted dot-product iAiBiCi where C was a vector of the importance of each frequency. The C vector should have peaks at the frequencies that provide the most information about the movement being performed and drop to zero at frequencies that are entirely random or have no effect at all. Since I didn't know what the key frequencies were, I created a number of different genetic algorithms to generate C iteratively over a number of generations. Initially this provided promising results, but they turned out to be anomalous.

Afterwards, Primary Component Analysis and Neural Networks were both attempted but gave unsatisfactory results. The next method attempted was an iterative method for creating classifier functions to distinguish between two sets of data. This method also gave very promising initial results but had the tendency to overfit to the training data, making it ineffective when analyzing new data.

In total, over 1 MB of python code was written, corresponding to around 22,000 lines, in addition to the 29 MB of auto-generated code (around 632,000 lines) written by my code. I made at least 2500 graphs, analyzed the order of 108 data points and also evaluated around 108 wavelet transforms.

The moral of the story is that it's easy to come up with promising results, but there is no guarantee that the interpretation given to the results corresponds to reality.

# Appendix 1

# Code Snippets

The following code snippets, written in python 2.7, show explicitly how the wavelet analysis processing time can be decreased by multiple orders of magnitude. The CWT() function is a basic wavelet transform for a given time series, scale, time displacement and mother wavelet, referred to as Version 2 elsewhere in this thesis. fWT is the improved "Fast Wavelet Transform" which returns all the wavelet coefficients at different times with the same scale, and is referred to as Version 4. The complete_fWT() function returns the wavelet coefficients for all time displacements and scales required to create a complete scalogram. The dimensions of the scalogram are given by a Canvas instance. The Canvas class is not included in this appendix because it is too large, but a surrogate Canvas class is provided at the end, which contains only the necessary properties for complete_fWT() to work.

```
from math import pi, e

def morlet (t):
    '''
    Morlet mother wavelet'''
    return pi**-.25*e**(2*pi*complex(0,1)*t-t**2/2.)

def integrate (X,dt=1,initial=0):
    '''
    Optimized Trapezoidal Integration with constant dt
    X = time series
    dt = sampling time
    initial = value of integral at beginning of time series'''
    return dt*((X[0]+X[len(X)-1])*.5+sum(X[1:len(X)-2]))+initial

def CWT(X,s,tau,psi,dt=1):
    '''
    Returns the Continuous Wavelet Transform of the time series X at time tau and
scale s
```

```
    X = time series
    s = scale = 1/omega = 1/(2*pi*f)
    tau = time offset
    psi = mother wavelet function
    dt = sampling time'''
    return s**-.5*integrate([psi((n*dt-tau)/s).conjugate()*X[n] for n in
range(len(X))],dt)

def fWT(X,psi,s,dt,mrange=None,handle_complex=''abs''):
    '''fast Wavelet Transform:  returns the wavelet coefficients for all the time
displacements at a specific scale
    X = time series data to be analyzed.  Should be a numpy array
    psi = mother wavelet function
    s = scale
    dt = sampling time of time series
    mrange = a list of data indices to be used.  The time displacements are [m*dt
for m in mrange].  The data returned is as long as mrange.
    handle_complex = what sort of data to return
    handle_complex=''abs''returns the norm of the wavelet coefficient,
    handle_complex=''real''returns the real part of the wavelet coefficient
    any other value will return the complex value of the wavelet coefficient'''
    N=len(X)
    if not mrange:
        mrange=range(N)
    A=float(dt)/s
    B=float(dt)*s**-.5
    p=min(N,int(4*s/dt)+1)
    phi=[psi(A*k) for k in range(-p,p)]
    WX=[]
    for m in mrange:
        if m-p>=0:
            x_low=m-p
            phi_low=0
        else:
            x_low=0
            phi_low=p-m
        if m+p<=N:
            x_high=m+p
            phi_high=2*p
        else:
            x_high=N
            phi_high=p-m+N
        try:
            ## This conditional allows the norm, real part or complex value to be
returned, but it will slow down the algorithm a little.
            ## For actual use, two options should just be commented out, with no
"handle_complex" variable in the function declaration.
            if handle_complex==''abs'':
                WX.append((m*dt,s,abs(B*X[x_low:x_high].dot(phi[phi_low:phi_high]))))
## norm of wavelet coefficient
            elif handle_complex==''real'':
                WX.append((m*dt,s,B*X[x_low:x_high].dot(phi[phi_low:phi_high]).real))
## real part of wavelet coefficient
            else:
                WX.append((m*dt,s,B*X[x_low:x_high].dot(phi[phi_low:phi_high])))
## complex value of wavelet coefficient
        except ValueError, e:
            print ''X.shape:'',X.shape
            print ''len(phi):'',len(phi)
            print ''x_low,x_high:'',x_low,x_high
            print ''phi_low,phi_high:'',phi_low,phi_high
            print e
            raise Exception
    return WX

def complete_fWT(X,canv,mother,dt,handle_complex=''abs''):
    '''returns the wavelet coefficients for every pixel on the canvas canv
    X = time series data to be analyzed
    canv = Canvas object
    mother = mother wavelet function
    dt = sampling rate of time series
    handle_complex = what sort of data to return
        handle_complex=''abs''returns the norm of the wavelet coefficient,
        handle_complex=''real''returns the real part of the wavelet coefficient
        any other value will return the complex value of the wavelet
coefficient'''
    mrange=[int(.5+t/dt) for t in canv.x_values()]
    return sum([fWT(X,mother,s,dt,mrange,handle_complex) for s in
canv.y_values()],[])

class Canvas:
    def __init__(self, xdomain=(0,1), ydomain=(0,1), width=100, height=100,
xlog=False, ylog=False):
        self.xdomain=xdomain
        self.ydomain=ydomain
        self.width=width
        self.height=height
        self.xlog=xlog
        self.ylog=ylog

    def get_x(self,pixel):        ## returns the x value of the X pixel
        if self.xlog:
            return self.xdomain[0]*(float(self.xdomain[1])/self.xdomain[0])**
(float(pixel)/(self.width-1))
        else:
            return self.xdomain[0]+float(pixel)*(self.xdomain[1]-self.xdomain[0])
/(self.width-1)

    def get_y(self,pixel):        ## returns the y value of the Y pixel
        if self.ylog:
            return self.ydomain[1]*(float(self.ydomain[0])/self.ydomain[1])**
(float(pixel)/(self.height-1))
        else:
            return self.ydomain[1]+float(pixel)*(self.ydomain[0]-self.ydomain[1])
/(self.height-1)

    def y_values(self):
        for Y in range(self.height):
            yield self.get_y(Y)

    def x_values(self):
        for X in range(self.width):
            yield self.get_x(X)
```

# Appendix 2

# Movement Names

In the code written for data analysis, abbreviations are used to refer to each of the 23 movements. These abbreviations also appear in some graphs and charts.

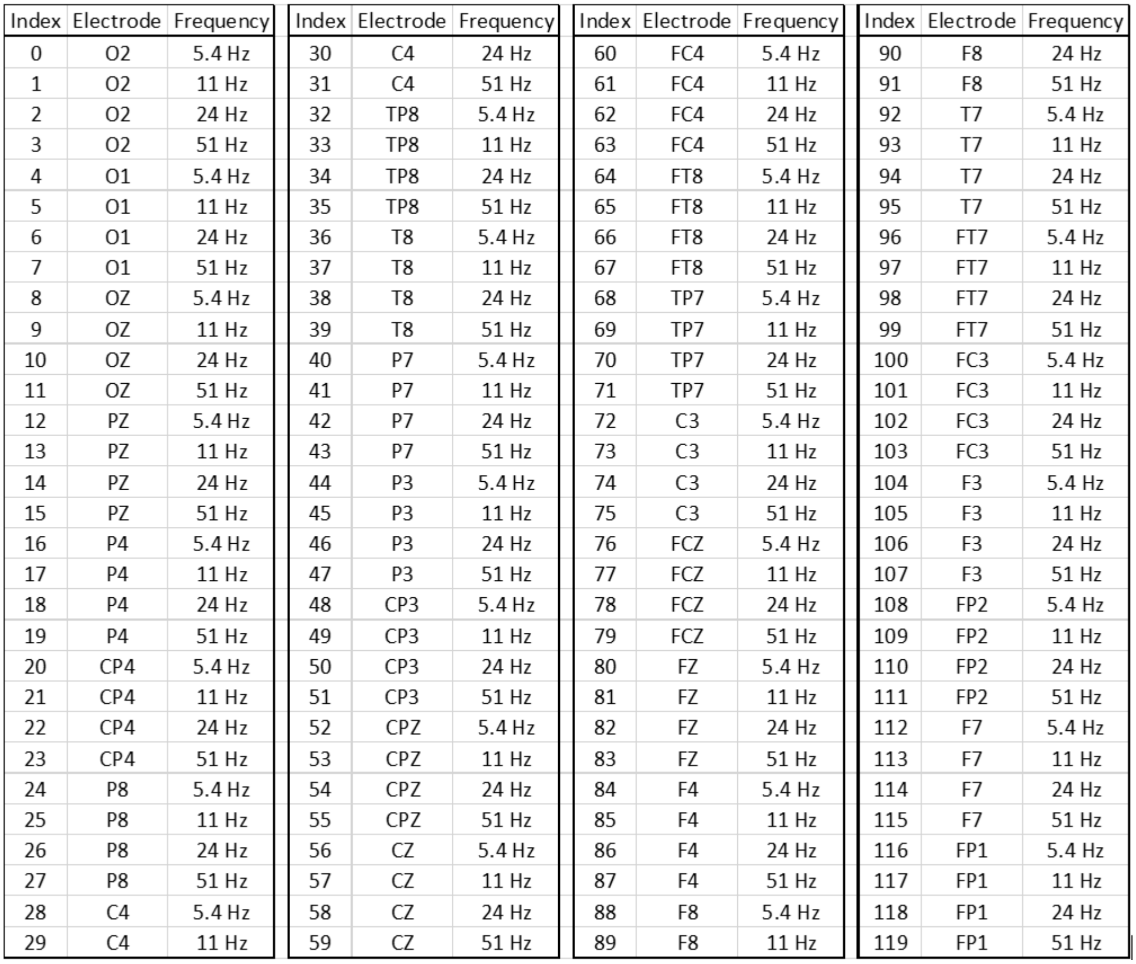
| Test | Test Name | Movement 1 | Movement 2 |
|---|---|---|---|
| 1 | Eyes Closed Resting State | RestClosed | RestClosed |
| 2 | Eyes Open Observation | WatchRight | WatchLeft |
| 3 | Horizontal Movement | Right | left |
| 4 | Horizontal Imagination | RightIm | LeftIm |
| 5 | Vertical Movement | Down | Up |
| 6 | Vertical Imagination | DownIm | UpIm |
| 7 | Rotation | RotCW | RotCCW |
| 8 | Rotation Imagination | RotCWIm | RotCCWIm |
| 9 | Finger Movement | CloseFing | OpenFing |
| 10 | Finger Imagination | CloseFingIm | OpenFingIm |
| 11 | Elbow Movement | OpenElb | CloseElb |
| 12 | Elbow Imagination | OpenElbIm | CloseElbIm |

RotCW means rotate clockwise  this is also known as supination. RotCCW is rotate counterclockwise and is also known as pronation.

# Appendix 3

# Feature Vector Components

The 120 component feature vectors contain wavelet transforms for different frequencies using the information from different electrodes. The following table shows which electrode and frequency is associated with each index of the feature vector.



| Index | Electrode | Frequency |
|---|---|---|
| 0 | O2 | 5.4 Hz |
| 1 | O2 | 11 Hz |
| 2 | O2 | 24 Hz |
| 3 | O2 | 51 Hz |
| 4 | O1 | 5.4 Hz |
| 5 | O1 | 11 Hz |
| 6 | O1 | 24 Hz |
| 7 | O1 | 51 Hz |
| 8 | OZ | 5.4 Hz |
| 9 | OZ | 11 Hz |
| 10 | OZ | 24 Hz |
| 11 | OZ | 51 Hz |
| 12 | PZ | 5.4 Hz |
| 13 | PZ | 11 Hz |
| 14 | PZ | 24 Hz |
| 15 | PZ | 51 Hz |
| 16 | P4 | 5.4 Hz |
| 17 | P4 | 11 Hz |
| 18 | P4 | 24 Hz |
| 19 | P4 | 51 Hz |
| 20 | CP4 | 5.4 Hz |
| 21 | CP4 | 11 Hz |
| 22 | CP4 | 24 Hz |
| 23 | CP4 | 51 Hz |
| 24 | P8 | 5.4 Hz |
| 25 | P8 | 11 Hz |
| 26 | P8 | 24 Hz |
| 27 | P8 | 51 Hz |
| 28 | C4 | 5.4 Hz |
| 29 | C4 | 11 Hz |
| 30 | C4 | 24 Hz |
| 31 | C4 | 51 Hz |
| 32 | TP8 | 5.4 Hz |
| 33 | TP8 | 11 Hz |
| 34 | TP8 | 24 Hz |
| 35 | TP8 | 51 Hz |
| 36 | T8 | 5.4 Hz |
| 37 | T8 | 11 Hz |
| 38 | T8 | 24 Hz |
| 39 | T8 | 51 Hz |
| 40 | P7 | 5.4 Hz |
| 41 | P7 | 11 Hz |
| 42 | P7 | 24 Hz |
| 43 | P7 | 51 Hz |
| 44 | P3 | 5.4 Hz |
| 45 | P3 | 11 Hz |
| 46 | P3 | 24 Hz |
| 47 | P3 | 51 Hz |
| 48 | CP3 | 5.4 Hz |
| 49 | CP3 | 11 Hz |
| 50 | CP3 | 24 Hz |
| 51 | CP3 | 51 Hz |
| 52 | CPZ | 5.4 Hz |
| 53 | CPZ | 11 Hz |
| 54 | CPZ | 24 Hz |
| 55 | CPZ | 51 Hz |
| 56 | CZ | 5.4 Hz |
| 57 | CZ | 11 Hz |
| 58 | CZ | 24 Hz |
| 59 | CZ | 51 Hz |
| 60 | FC4 | 5.4 Hz |
| 61 | FC4 | 11 Hz |
| 62 | FC4 | 24 Hz |
| 63 | FC4 | 51 Hz |
| 64 | FT8 | 5.4 Hz |
| 65 | FT8 | 11 Hz |
| 66 | FT8 | 24 Hz |
| 67 | FT8 | 51 Hz |
| 68 | TP7 | 5.4 Hz |
| 69 | TP7 | 11 Hz |
| 70 | TP7 | 24 Hz |
| 71 | TP7 | 51 Hz |
| 72 | C3 | 5.4 Hz |
| 73 | C3 | 11 Hz |
| 74 | C3 | 24 Hz |
| 75 | C3 | 51 Hz |
| 76 | FCZ | 5.4 Hz |
| 77 | FCZ | 11 Hz |
| 78 | FCZ | 24 Hz |
| 79 | FCZ | 51 Hz |
| 80 | FZ | 5.4 Hz |
| 81 | FZ | 11 Hz |
| 82 | FZ | 24 Hz |
| 83 | FZ | 51 Hz |
| 84 | F4 | 5.4 Hz |
| 85 | F4 | 11 Hz |
| 86 | F4 | 24 Hz |
| 87 | F4 | 51 Hz |
| 88 | F8 | 5.4 Hz |
| 89 | F8 | 11 Hz |
| 90 | F8 | 24 Hz |
| 91 | F8 | 51 Hz |
| 92 | T7 | 5.4 Hz |
| 93 | T7 | 11 Hz |
| 94 | T7 | 24 Hz |
| 95 | T7 | 51 Hz |
| 96 | FT7 | 5.4 Hz |
| 97 | FT7 | 11 Hz |
| 98 | FT7 | 24 Hz |
| 99 | FT7 | 51 Hz |
| 100 | FC3 | 5.4 Hz |
| 101 | FC3 | 11 Hz |
| 102 | FC3 | 24 Hz |
| 103 | FC3 | 51 Hz |
| 104 | F3 | 5.4 Hz |
| 105 | F3 | 11 Hz |
| 106 | F3 | 24 Hz |
| 107 | F3 | 51 Hz |
| 108 | FP2 | 5.4 Hz |
| 109 | FP2 | 11 Hz |
| 110 | FP2 | 24 Hz |
| 111 | FP2 | 51 Hz |
| 112 | F7 | 5.4 Hz |
| 113 | F7 | 11 Hz |
| 114 | F7 | 24 Hz |
| 115 | F7 | 51 Hz |
| 116 | FP1 | 5.4 Hz |
| 117 | FP1 | 11 Hz |
| 118 | FP1 | 24 Hz |
| 119 | FP1 | 51 Hz |

# Appendix 4

# Presentations

Three posters and one presentation derived from this thesis have been presented in the following four scientific conventions:

VI Congreso Nacional de Tecnología Aplicada a Ciencias de la Salud
Puebla, Pue, Mexico: June 4-6, 2015
Poster: "Diseño de Software para Análisis de Señales Neuronales"

Society for Neuroscience Annual Meeting 2015
Chicago, IL, USA: October 17-21, 2015
Poster: "Wavelet Analysis on Electroencephalographic Time Series to Identify Key Patterns Corresponding to Arm Movements"

LVIII Congreso Nacional de Física
Merida, Yucatan, Mexico: October 5-9, 2015
Poster: "Identificación de Patrones Característicos en Señales Electroencefalográficas para Interfaz Cerebro-Computadora"

XIV Mexican Symposium on Medical Physics
Mexico City, Mexico: March 16-17, 2016
Presentation: "Wavelet Analysis on Electroencephalographic Time Series to Identify Key Patterns Corresponding to Arm Movements for Brain-Computer Interface"

# Bibliography

[1] Li Yao-Nan, Wang Yun-Xia, Zhang Jin-Jin, Zhang Xiao-Dong, *"An Approach for Pattern Recognition of EEG Applied in Prosthetic Hand Drive "*. Systemics, Cybernetics and Informatics. Vol 9. pp 51-56. 2011.

[2] Basu Debabrota, Bhattacharyya Saugat. *"Interval type-2 fuzzy logic based multiclass ANFIS algorithm for real-time EEG based movement control of a robot arm "*. Robotics and Autonomous Systems. Vol 68. pp 104-115. 2015. `http://dx.doi.org/10.1016/j.robot.2015.01.007`

[3] Azevedo FAC, Carvalho LRB, Grinberg LT, Farfel JM, Ferretti REL, Leite REP, Filho WJ, Lent R and Herculano-Houzel S. *"Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain"*. The Journal of Comparative Neurology. Vol 513(5). pp 532-541. 2009.

[4] Pinker Steven. *"How the Mind Works"*. W.W. Norton & Company, Inc. 2009.

[5] Hudspeth A.J., Jessel Thomas, Kandel Erik, Schwartz James, Siegelbaum Steven. *"Principles of Neural Science"*. McGraw-Hill Companies, Inc. $5^{th}$ Edition. 2013.

[6] Augustine George, Fitzpatrick David, Hall William LaMantia Anthony-Samuel, Purves Dale, White Leonard, *"Neuroscience"*. Sinauer Associates, Inc. 5th Edition. 2012.

[7] Buzsáki György. *"Rhythms of the Brain"*. Oxford University Press. 2006.

[8] Bolner Federico, Rohu Victor. *"Decoding Movement Direction for Brain-Computer Interfaces using Depth and Surface EEG Recordings"*. Aalborg University, Aalborg. 2012.

[9] Duda Richard, Hart Peter, Stork David. *"Pattern Classification"*. John Wiley & Sons. $2^{nd}$ Edition. 2001.

[10] Gazzaniga Michael. *"Tales from Both Sides of the Brain: A Life in Neuroscience"*. Ecco. $1^{st}$ Edition. 2015.

[11] Lopes da Silva Fernando. *"EEG and MEG: Relevance to Neuroscience"*. Neuron. Vol 80 Issue 5, pp 1112-1128. 2013. `http://dx.doi.org/10.1016/j.neuron.2013.10.017`

[12] Chambers J.A., Sanei Saeid. *"EEG Signal Processing"*. John Wiley & Sons. 2007.

[13] Compo Gilbert, Torrence Christopher. *"A Practical Guide to Wavelet Analysis"*. Bulletin of the American Meteorological Society. Vol 79. pp 61-78. 1998.

[14] Torrence Christopher, Webster Peter. *"Interdecadal Changes in the ENSO-Monsoon System"*. Journal of Climate. Vol 12. pp 2679-2690. 1999.