ANALYSIS AND ELICITATION OF ELECTROENCEPHALOGRAM DATA PERTAINING TO HIGH ALERT AND STRESSFUL SITUATIONS: SOURCE LOCALIZATION THROUGH THE INVERSE PROBLEM

by

ISAAC CHARLES HEIM

DANIEL J. FONSECA, COMMITTEE CHAIR
RICK A. HOUSER
RYAN M. COOK
KEITH A. WILLIAMS
BETH A. TODD

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
In the Department of Mechanical Engineering
In the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2021

ABSTRACT

This dissertation work deals with the design and development of a fuzzy controller to analyze electroencephalogram (EEG) data. The fuzzy controller made use of the multiple functions associated with the different regions of the brain to correlate multiple Brodmann areas to multiple outputs. This controller was designed to adapt to any data imported into it. The current framework implemented supports a math study and a police officer study. The rules for the interactions of the Brodmann areas have been set up for these applications, detailing how well the police subjects' brains exhibited behavior indicative to activation relating to *vision, memory, shape/distance, hearing/sound,* and *theory of mind*. The math subjects' outputs were attuned to their related study which involved transcranial direct current stimulation (tDCS), which is a form of neurostimulation. *Anode affinity, cathode affinity, calculation, memory,* and *decision making* were the outputs focused on for the math study. This task is best suited to a fuzzy controller since interactions between Brodmann areas can be analyzed and the contributions of each area accounted for.

The goal of the controller was to determine long-term behavior of the subjects with repeated sampling. With each addition of data, the controller was able to develop new bounds related to the current condition of the data in the study. Processing this data was accomplished by the creation of an automated filtering script for EEGLAB in MATLAB. The script was designed to rapidly load and filter the files associated with any given dataset. These files were also automatically prepared for analysis with a program called Low Resolution Brain

Electromagnetic Tomography i.e. (LORETA). LORETA was used to solve the inverse problem, which involves identifying where the signals from the surface electrodes originated within the brain through a process called source localization. Once the sources of the EEG signals were located, they were associated with the Brodmann areas. The fuzzy controller then processed this information to automatically generate heat maps which displayed information such as normalized data, z-score, and rankings. Each set of scores displays how the subject's brain was acting, which lined up with the expected results.

DEDICATION

This dissertation is dedicated to everyone who believe in me and helped me during my journey, especially my family and my advisor Daniel Fonseca. Without them, I would not have made it this far. I will always cherish the connections I have made and the lessons I have learned during this adventure, and while this chapter of my life may be coming to an end, I'll just have to get excited about the next step forward, because it is true what they say, it is all about the friends you make along the way.

LIST OF ABBREVIATIONS AND SYMBOLS

A	Anterior
ACC	Anterior cingulate cortex
ADHD	attention-deficit/hyperactivity disorder
AHP	Analytic Hierarchy Process
ATL	Anterior temporal lobe
BOLD	blood oxygenation level-dependent
DLPFC	Dorsolateral prefrontal cortex
EEG	Electroencephalography
ERP	Event-related potential
FAHP	Fuzzy Analytical Hierarchical Process
FDCRT	Fuzzy Distance Correlation Ranking Technique
FDM	Fuzzy Delphi Method
FEM	Finite element method
FFA	Fusiform face area
fMRI	Functional magnetic resonance imagining
FSIRT	Fuzzy Interval-based Ranking Technique
GUI	Graphical user interface
ICA	independent component analysis

L	Left
LEO	Law enforcement officers
LORETA	Low Resolution Brain Electromagnetic Tomography
MATLAB	Matrix laboratory
MDRS	Mean-Downside Risk-Skewness
MEPs	Motor evoked potentials
MN	Minimum Norm
MR	Magnetic resonance
MRI	Magnetic resonance imagining
OFC	Orbital frontal cortex
P	Posterior
R	Right
RINF/RIF	Right inferior frontal cortex
RPAR	Right parietal cortex
sLORETA	Standardized Low Resolution Brain Electromagnetic Tomography
SNR	Signal to noise ratio
tDCS	Transcranial direct current stimulation
TMS	Transcranial magnetic stimulation
ToM	Theory of mind
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
WROP	Weighted Resolution Optimization
~=	Not Equal in rule generation
<	Less Than

=	Equal to
==	Comparison equal to for rule generation
>	Greater than

ACKNOWLEDGEMENTS

This was a difficult journey, but every step along the way, I was blessed with friends and faculty that pressed me forward. Firstly, I'd like to acknowledge Dr. Daniel Fonseca, who helped me choose this project and guided me along the path, tirelessly correcting any mistakes I made to present a complete document. John O'Donnell who really helped me get this project started and remained a trusted advisor thought out the process. Dr. Ryan Cook and Dr. Rick Houser for sharing their expertise on the brain and EEG data. Jumping into a new field of study was a lot simpler with their aid and expertise. Dr. Nader Jalili for all the help and support from the department. Dr. Keith Williams, Dr. Hwan-Sik Yoon, and Dr. Vishesh Vikas for a great experience as a teaching assistant, and always willing to discuss my progress. Dr. Beth Todd for being there at the start of my college experience as an undergrad helping me with the problems I encountered. My brother, C.J. Heim, who got me started in mechanical engineering and advised me to start the master's program which led me into my PhD. Dr. Mahmoodi and Dr. Shen, who provided valuable research opportunities during my master's studies, which aided in my growth towards pursuing my PhD. My family, C.J., Katrina, LouAnn, Chuck Heim and all my relatives who supported and encouraged me throughout the years, never giving up hope that I'd finish. All of my friend's past, present, and future for being my friends and supporting me, especially my adventuring friends who have lived many experiences with me, always exploring something new. Lastly, the MATLAB Discord community who aided in troubleshooting and expanding my scripts saving my countless hours.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Research Background

Electroencephalography (EEG) is a tool used to better understand the inner working of the human brain. By utilizing the bioelectrical potentials generated by the cortex nerve cells within the brain, it is possible to identify which areas of the brain are active [1]. EEG can be either invasive or non-invasive based on the equipment used, but it is more commonly non-invasive with the advancements made since its invention. EEG is an important part of understanding neurological disorders, such as, epilepsy, brain tumours, and locating head damages [2, 3].

Transcranial direct current stimulation (tDCS) is a form of neurostimulation that has been researched, in recent years, as a means of treating cognitive deficiencies. tDCS is a noninvasive stimulation for the brain. By using a pair of electrodes, it is possible to excite the cerebral cortex. The range of applications is vast, ranging from medical issues to learning and training [4].

In the medical field, tDCS has been applied to different neuropsychiatric diseases and disorders including depression, epilepsy, electroanalgesia, stroke, schizophrenia, and Parkinson's disease [4]. tDCS can also be used to enhance performance during cognitive tasks. There have been studies that report tDCS can facilitate training-related performance improvements during simple motor tasks as well as enhancements in the planning abilities of subjects [5, 6]. Cosmo (2015) found that there was an increase in cortical connectivity after stimulation of the left

dorsolateral prefrontal cortex (DLPFC) in sixty attention-deficit/hyperactivity disorder (ADHD) patients. His results suggested that the effects of tDCS are selective based upon the patient. The contributors to variability within individuals include genetics, gender, age, hormone level, and time of day. In tDCS, parameters such as electrode size, current intensity, and current density also affect stimulation, which can lead to an even greater variability between studies [7].

1.2 Research Problem

EEG signals are often regarded as a measurement of neural activity, but they do not give a complete understanding of the underlying neurophysiology. The neural activity of the brain is mediated by the synaptic interactions and the resulting action potential propagation. This means the EEGs signals measurements are the excitation of a neuron traveling along its axon towards the axon terminal which is where the action potentials are transmitted to other neurons, muscle cells or glands [8]. The following are the basics that make up what is commonly known as the inverse problem. The surface electrodes do not exactly show where the electrical signals are coming from. It is possible to predict which electrodes will be activated from each given generator, but much more difficult to do the inverse. When a channel is activated, it is more difficult to determine which generator in the brain caused the excitation. The channels are based on the surface of the scalp, and therefore, cannot convey some of the deeper activity within the brain. The desired outcome is determining these generators, which will determine where the brain started the excitation [3].

By solving the inverse problem, it is possible to determine the area in which the activation originated in the subjects. This is done through a process called source localization. Source localization attempts to gather all the outputs and then inversely relate them back to the generators within the brain. Since the tDCS subjects have areas of the brain stimulated, it is

important to compare them to the control patients to determine the base excitation of the workload provided. By knowing the areas most utilized in control subjects, it is feasible to tailor tDCS parameters in subsequent studies to target these key areas. The main parameters of tDCS that would need to be optimized are current density (mA/cm^2), electrode placement, and duration of stimulation [9]. All of these parameters can be changed, but usually stay within certain bounds. For example, current intensity usually ranges from 0.2 mA to 2 mA, for safety reasons. An example of a tDCS set-up is shown in Figure 1.1.

Figure 1.1 Visualization of tDCS process [5]

It is suggested in Ho et al. (2016) that the following parameter relationships matter: "1) size of the electrode relative to the size of the target area, 2) location of the electric field relative to the target area, and 3) direction of the electric field relative to the neuronal orientation [10]." Ho (2016) suggested that positioning, size, and orientation of tDCS relative to the cortical regions should also be given consideration [10]. Saturnino et al. (2015) researched some additional parameters, such as conductivity of the rubber and sponge (or gel) layers, and connector position. The connector position determined how the current is fed into the electrode

surface. Satumino et al. (2015) also found that the strongest stimulation occurred between the electrodes rather than underneath them, which is consistent with other studies [11].

1.3 The Big Picture

After tDCS stimulation, it is possible to gather data on how the subject's brain was affected to determine changes for the discussed parameters to better target key brain areas. While taking data with an electroencephalogram (EEG), there are several factors to consider as brain waves readings are recorded. Electronic noise can be added to the signal from outside sources, and if left unaccounted for, it can affect how the data recordings are interpreted. Hence, data filtering becomes essential to accurately analyze EEG data.

Filtering out each data set individually can be tedious, and depending on the method used, can take hours for even small data sets. Even with the smaller sets, handling multiple sets of data can lead to the data being filtered erroneously. In order to accommodate data sets, especially the bigger ones, a more automated method is required. Although the method will still take time to process, the ability for the computer to process all the data without user input allows for other tasks to be completed during this time, or to process while the computer would otherwise be idle. The proposed method is devised in such a way that would allow multiple studies to use the same basic framework without the need for recreating the script for every study.

There are several different source localization programs and methods to solve the inverse problem. One is Standardized Low Resolution Brain Electromagnetic Tomography (sLORETA). Due to the nature of sLORETA, it is more accurate if the noise has been addressed. Therefore, filtering the data is necessary to solve this problem. Once the noise of the data has been addressed, source localization can begin. The filtered surface electrode data can then be utilized

with a transformation matrix to identify the generators within the brain. These generators take the form of Brodmann areas, and the inverse problem is solved by identifying which Brodmann areas initiated the excitation on the scalp. Since there are many generators within the brain, the next step is to identify how the generators targeted by tDCS were affected. Finally, this information can be compiled and used to optimize the accuracy and effectiveness of tDCS.

This research work focuses on developing a comparison of the brain under normal conditions and after being stimulated by tDCS. Through learning which areas of the brain are used under normal conditions, individuals stimulated by tDCS can identify what specific brain regions are being stimulated. Solving the inverse problem allows the brain to become a three-dimensional image rather than just surface electrodes. This allows for a deeper comparison between the subjects. By determining which areas are used during normal conditions, tDCS can be utilized to target these areas to increase their activation. This methodology can be used for any task desired. Once the targeted area is determined, then the optimization of tDCS parameters can better target and stimulate the brain to increase focus and proficiency.

1.4 Research Objectives and Scope

There are many paths to take when analyzing EEG data, and many stops along those paths that need to be addressed, from filtering out noise to the final data analysis. Studies can have multiple subjects with multiple sessions. Each additional step brings further steps to the analysis. This research aims to address and improve multiple aspects of analyzing EEG data.

1. Create a fast and streamline way to analyze multiple sessions and sets of data with minimal user input
2. Clean the data of noise and artifacts, as part of the analysis
3. Develop a method to determine the generators within the brain for further analysis

4. Analyze the data, as a whole, with a method to account for the fuzziness of the data
5. Instill modularity into the programming in order to allow multiple types of analysis with easy and minimal effort

The first objective of this research was completed by creating a modular script and system of analyzing EEG data that should allow for a fast, easy, and meaningful analysis. This script also allows for a quick turnaround time between the taking of data and the analysis of the data. This enables data collectors to have a quicker reaction to what each individual subject's data is representing. This would potentially enable investigators to analyze data quick enough to use that gained understanding in the next session of a study. This allows periodic understanding to be gained as soon as possible for studies that take place over the course of weeks or months. In order to accomplish this, it is necessary for the script to be easily understood and formatted by the analyst. For this reason, the script is commented thoroughly explaining what each section does, along with what user inputs are required, and what the user can change to accommodate for the intricacies of the study at hand. The main inputs are gathered at the top of the script for easy access. Each variable is described, along with input parameters, if applicable.

The next step is to address the inverse problem. While the final point of the electrical signals is known with the EEG data, the point of origin is desired. sLORETA is used to process the data and determine the magnitude and location of the activation for the data. This gives a deeper insight into the data provided and allows for the final steps to be accomplished.

The final step varies based on the type of analysis desired. A fuzzy set analysis was conducted in order to categorize and explain the individual data sets, and how they compare to the study as a whole. The constructed fuzzy controller provides insight into the data and suggestions for future tDCS session parameters.

After these steps are completed, the controller was set up to be modular, similar to the scripts. Once the math study was done, a second different study, the police study, was used with the script and controller to test the modularity and results provided. Once the results were satisfactory, the final script and controller were then discussed and presented.

The overall research objectives can be seen in Figure 1.2. This study is broken into three sections. Each section can stand alone but for the full presented analysis to be complete each step must be done for the data provided. Future studies can stop at filtering or after source localization. Data that is not EEG in nature can also skip the first two sections and be implemented right into the controller.

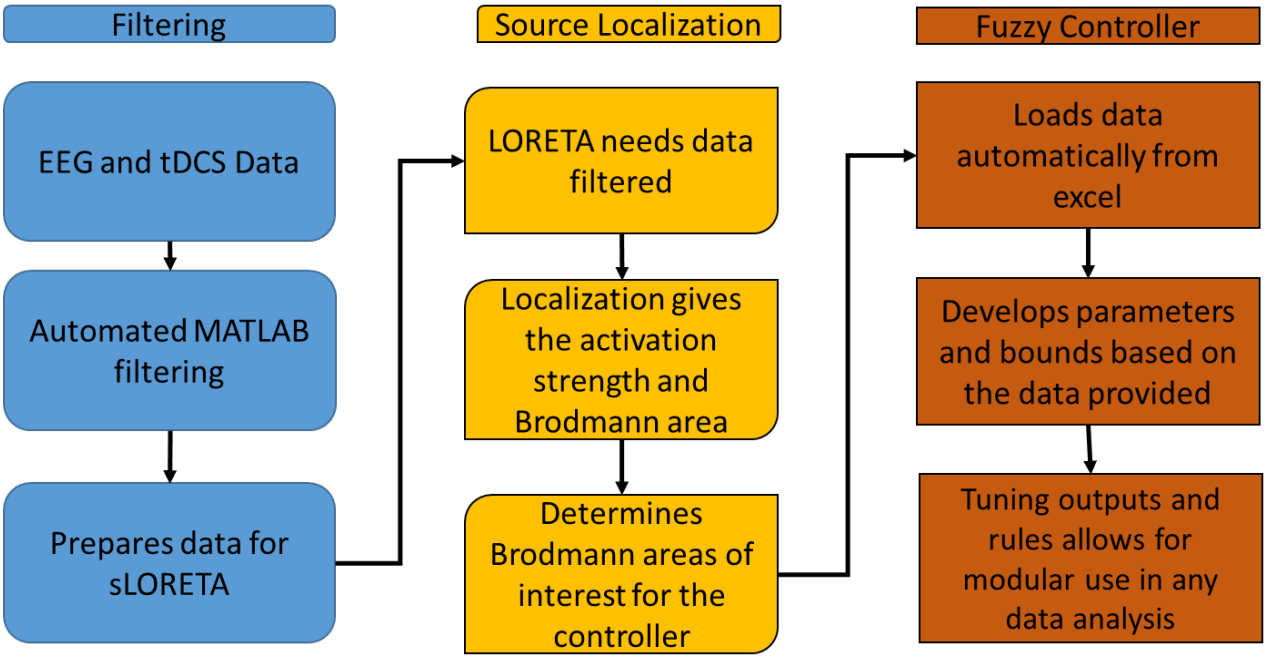

Figure 1.2 Overview of the Research Objectives.

1.5 Background Literature

1.5.1 Introduction

Functional magnetic resonance (MR) imagining (fMRI) and EEG are both types of noninvasive methods used to investigate the inner working of the human brain. Each has its own advantages, drawbacks, and applications. fMRI measures the vascular response, usually used with blood oxygenation level-dependent (BOLD) signals. fMRI offers a whole-brain coverage on the order of cubic millimeters. EEG has low spatial resolution (order of centimeters) and spatial uncertainties due to the volume conductance and inverse problem. EEG only reflects activity on the surface of the brain, and it is therefore not sensitive to some aspects of neuronal activity. EEG is able to read neural activity on a resolution of milliseconds, which allows the mapping of brain changes in response to an event (event-related potential, ERP) [12].

The power of an MRI comes from its strong magnetic field and radio frequencies. This allows the MRI to visualize organs, soft tissue, and bone in great detail. Unlike x-rays, this can be done without any ionizing radiation. The spatial resolution of an MRI is in the millimeters. The use of MRI in research has increased exponentially over the past decade. An image of the electrical circuit of an MRI system is shown in Figure 1.3 [13, 14]. MRI uses its magnetic field in order to capture images of the brain. These images come from various planes from within the brain. A doctor can then take these images to create a three-dimension view from the many different angles [15].

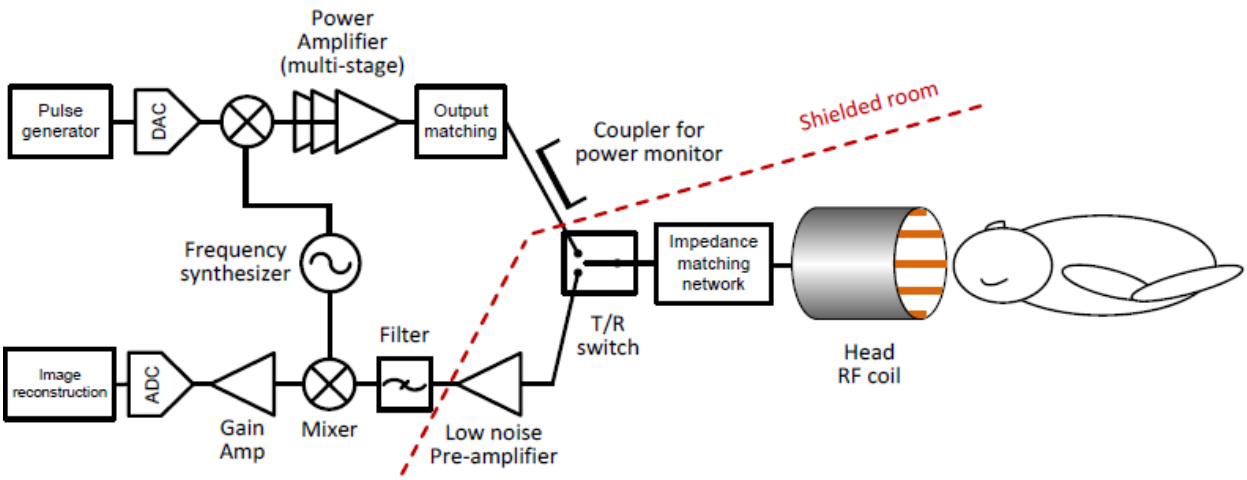

Figure 1.3 The electrical circuit of an MRI system [14].

Common uses of MRI include studying both typical and atypical brain structures in children and adults. MRI has been used the exploration of brain development throughout the first two decades of life. Compared to relatively low cost of EEG equipment and administration, a clinical MRI scan costs approximately $1,500 and from $500 to $600 for a research scan. Despite its high cost, MRI technology has become very popular in observing brain development because it can provide such detail images of the brain with the same procedure through the entire lifespan of brain development [13].

Combining EEG and fMRI data is difficult. While they both have their benefits, the subjects are in different environments and mental states due to how each of the methods are enacted. EEG and fMRI only correlate the brain operations to mental and behavioral processes, while transcranial magnetic stimulation (TMS) is able to manipulate brain activity as an independent variable. TMS has a transient effect that induces “virtual lesions” that can enhance or decrease cortical excitability, stimulate neural populations, or induce local oscillations. Peters

et. al. (2012) showed that concurrent TMS-EEG-fMRI measurements were safe and comfortable while recording good signal quality [12].

1.5.2 History of EEG

EEG was developed by German Physicist Han Berger in 1929 as a neuroimaging technique to measure the potential differences between a pair of electrodes placed on the scalp. Berger tried to locate tumours in the brain by using cerebral localization. The initial trials were performed with chlorinated silver needle electrodes, platinum wires, and zinc-plated steel needles. It was not until 1932 that Kornmuller discovered the significance of using multichannel recordings in order to cover a wider brain region[3, 16]. That same year, J. F. Toennies built the first ink-writing biological amplifier that recorded the brain potentials. With the help of a co-worker, Brian Matthews, he also designed the differential amplifier, which is still an integral part of EEG amplification [16].

In 1937, Oskar Vogt and his wife Cécile developed a concept to compartmentalize sharply separated areas. They discovered boundaries between healthy and diseased areas of the hippocampus. There were thought to be around 200 regions with precise distinctions from field to field. During this time, Toennies (1938) was able to construct the first cathode follower, which recorded high-resistance electrodes. Thus, microelectrode recording was created, and it would prove to be a powerful tool through the 1950's and 1960's [16].

The late 1940's saw two new developments. The first was the rise of invasive techniques that utilized special depth electrodes. This allowed for deep intracerebral region exploration. The second development was automatic frequency analysis, which was further developed and optimized in the 1960's. The 1950's brought about the widespread use of EEG. Every major university had one, and, by the end of the 1950's, there was a large number of them in hospitals

and even in private practices. The 1960's ushered in a high point for EEG with the shift from tracing to automatic data analysis. Computerization flourished during the 1960's and 1970's. In 1965, Cooley and Tukey were credited with the introduction of fast Fourier transforms (FFT) as the basis of power spectral analysis [16].

The 1970's brought great progress to the evoked potential technique. The pattern changer used in the visual evoked potential technique increased the reliability of this method highly. The 1970's and 1980's saw the emergence of structural neuroimaging techniques such as computed tomography and magnetic resonance imaging. This seemed to cause EEG to suffer a blow, but EEG was never a structure-oriented test. Therefore, EEG was still able to find dysfunctional parts of the brain due to a hemispheric brain tumor, a vascular lesion, or a traumatic collision. Topical EEG diagnosis made a comeback with computerized brain mapping (1990) thanks to Frank Duffy [16, 17].

In more recent years, John Ives, Steve Warach, and Franz Schmitt (1992) recorded EEG within the bore of a Siemens 1.5 T magnet. This led to the technical concept that would allow EEG to be recorded during fMRI. On March 2, 2002, Louis Lemieux and Robert Turner hosted the first workshop on EEG-fMRI. EEG-fMRI had to be performed in an 'interleaved' fashion until artifact reduction became available. The readable EEG epochs were taken within the MRI scanner [18].

1.5.3 Applications of EEG

EEG has been utilized for many clinical applications to improve the understanding and treatment of neurological and neurophysiological disorders such as epilepsy, schizophrenia, depression, and Alzheimer's diseases. Epilepsy is characterized by a large and sudden neuronal discharge in the brain, which manifests as a seizure. With EEG, it is possible to develop

computer-based seizure detection in order to better understand and treat an individual with epilepsy or other related neurological disorders. Epilepsy has a very fast propagation, causing several hyper regions to manifest in the images [3]. Therefore, high temporal resolution is needed to fully capture the brainwave data. Due to the EEG's high temporal resolution, on the order of milliseconds, it is regarded as the best non-invasive diagnosis tool used in epilepsy surgery centers [3].

EEG source imaging is utilized in many important applications in cognitive neuroscience and clinical neuroscience. The clinical areas include neurology, psychiatry, and psychopharmacology. Cognitive neuroscience applications mainly focus on investigating the temporal aspects by analyzing event related potentials (ERP). In this sense, neurology focuses more on the study of motor evoked potential, but in clinical applications, the main concern is localizing epileptic foci. In psychiatry and psychopharmacology, the focus tends to be towards localizing the sources in certain frequency bands. While all of these applications aim to localize the sources, the pre-processing of the data is somewhat different for each application [19].

1.5.4 EEG Data Filtering

Nearby electrical equipment can affect the data acquired through EEG. This includes the frequency of power supplies, biological, technological, extrinsic artifacts and other electronics. This influence is commonly referred to as noise. The noise from the power supply is usually removed by a differential circuit as it is a fixed frequency. Biological frequencies are dependent on the subject, such as eye blinks and eye movement. These in particular are called artifacts. They generally have a very high amplitude, and occasionally, greater than the EEG signal. There are three common methods to remove these artifacts [20].

The first is the Artifact Subtraction method. By calculating the attenuation factor for the Electrooculography EOG signal, the signal is then amplified by this factor. The result is then subtracted from the EEG signal, which removes any artifacts. Since this only reads EOG signals sent to the head electrodes, and not EEG signals to the eyes, it might eliminate useful EEG signals during artifact removal [20].

The next method is the Wavelet Transformation method. By using the difference in the statistical characteristics, it is possible to remove artifacts. This method takes into account the signal and the noise. It does not require a reference signal, and it removes artifacts automatically. The issue arises with determining a threshold for the dataset. An unstable threshold can result in the degradation of the EEG data [20].

Lastly, there is the Blind-source Separation (BSS) method. A mixture of principal component analysis and independent component analysis (ICA) are combined in this process. The EEG signal is decomposed into different source components. Setting the source components that relate to artifacts to zero reduces their effects. ICA can produce source components that are a mixture of artifacts and data. Hence, it is possible that some EEG data might be lost during artifact removal. Due to this, many BSS algorithms require human intervention to identify artifact components, which can be subjective and time consuming [20]. ICA can be quite powerful. Fatima et. al. (2013) showed that ICA was effective in isolating and subtracting artifacts from data. In particular, they removed eye movements, cardiac, and muscular artifacts. On average, they removed about 5 out of every 150 components per data set. It was necessary to weigh the removal of artefactual components against the noise that would be introduced via the matrix regularization when the data was transformed back to its original form. Fatima et. al. (2013) was able to significantly improve the detection and localization of their sources [21].

1.5.5 Source Localization and the Inverse Problem

Within EEG data, there are parts of data related to the sources active within the brain. The signals measured from the scalp surface do not directly indicate which neurons, or generators, are active within the brain. This is what is known as the inverse problem. It is called like that since multiple configurations of generators can produce the same surface configuration. It is possible to use source localization to identify these sources, and hence, solve this inverse problem. With reasonable constraints in place, the most probable sources in the brain can be determined at any given time frame [19].

1.5.5.1 Evaluation of Source Localization

Through simulated data, it is possible to evaluate the reasoning and formulas used during source localization. This method involves using a modeled brain to calculate a forward solution. By creating a dipole, the scalp potential distribution can be computed. At this point, solving the inverse problem would lead to an estimate of the source based on the generated surface data. The difference between the actual and estimated areas are used to calculate the error. By determining this error, it is possible to evaluate and compare the accuracy of distributed inverse solutions. This allows new methods to be tested. As part of the evaluation, Michel et. al. (2004) tested different electrode configurations as shown in Figure 1.4. It was concluded that source localization required a uniform sampling of the full head. If the sampling is non-uniform, the results could be drastically wrong [19].

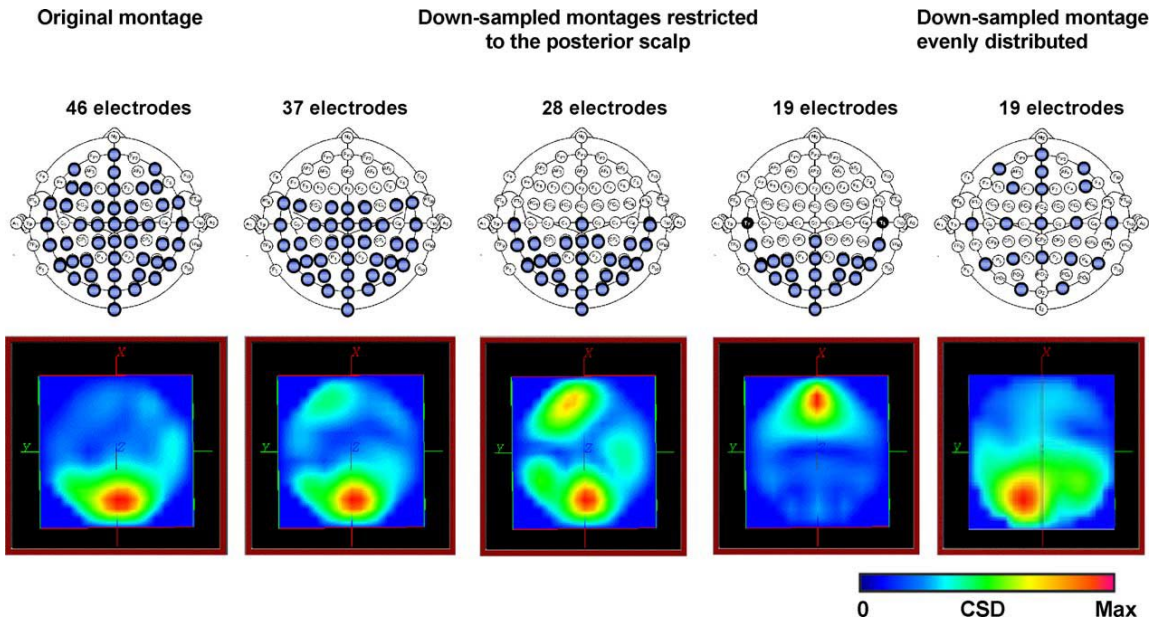

Figure 1.4 Effects of electrode distribution on the source estimation [19]

These simulations usually employ a simple spherical head model. Pascual-Marqui (1999) used dipole localization error to compare different methods of source localization. These methods are Minimum Norm (MN), Column Weighted MN, Low Resolution Brain Electromagnetic Tomography (LORETA), and other general classes of inverse solutions, such as the Backus-Gilbert method and Weighted Resolution Optimization (WROP). From this comparison, LORETA was found to have the smallest localization errors. It was also shown that LORETA has a strong noise-dependency. However, without noise regularization, LORETA is not able to provide any proper localization, even in cases with a high signal-to-noise ratio [19].

1.5.6 Historical Background of Transcranial Direct Current Stimulation

There has been a surge of interest in tDCS over the past decade, seeing an increasing number of applications. tDCS has changed drastically from when it was first researched. For instance, before tDCS, there existed techniques that delivered strong electrical currents to relieve

headaches and epilepsy over the past two centuries. In 1804, Aldini used electric stimulation to treat mental disorders, thus starting electrotherapy. The results for low-level DC current were variable, and in the 1930's, electroconvulsive therapy led to fewer studies into weak DC currents. In 1949, Hebb proposed a mechanism for plasticity within the brain. When the axon of a particular cell is close enough to another cell, and it is part of the firing process, some growth or metabolic change occurs in both cells. In this way, the first and second cells ability to interact will be increased [22].

In 1956, Terzuolo et al. (1956) studied how DC currents affect the neural preparations and the relative orientations of the current to the axon [23]. They found that the frequency of firing could be increased by currents of 3.6×10^{-8} A when injected across the preparation region. These currents were able to increase the frequency without directly initiating an action potential. In 1964, Bindman (1964) used currents of 0.25 $\mu A/mm^2$ on exposed pia using surface electrodes at 3 μA over 12 mm^2 [24]. He found that with just minutes of stimulation in the rat preparations, the neurons exhibited spontaneous activity, and evoked responses for hours thereafter [22].

In 1975, Dymond and colleagues (1975) investigated how the current flows through the scalp, using two pairs of electrodes placed on the frontal poles and over the mastoids. It was found that 45% of the current passed through the brain. In 2006, Miranda et. al. (2006) used a model with modern parameters, defined by Nitsche and Paulus (2000) as 25 cm^2 to 35 cm^2 rubber electrodes and current intensities of 1 mA to 2 mA, applied for 10 to 20 minutes. It was found that only about 10% of a scalp current of 2 mA reaches the cortex [22].

1.5.7 Applications of Transcranial Direct Current Stimulation

Transcranial direct current stimulation (tDCS) is a type of noninvasive stimulation for the brain. By using a pair of electrodes, it is possible to excite the cerebral cortex. The range of

applications is vast, ranging from medical issues to learning and training. tDCS has been applied to different neuropsychiatric diseases and disorders including depression, epilepsy, electroanalgesia, stroke, schizophrenia, and Parkinson's disease [4]. tDCS can also be used to enhance performance during cognitive tasks. There have been many studies that report tDCS can facilitate training-related performance improvements during simple motor tasks [5]. tDCS has been shown to improve planning ability in subjects as well [6]. In individuals with attention-deficit/hyperactivity disorder (ADHD), Cosmo (2015) found that there was an increase in cortical connectivity in the targeted and related areas after stimulation of the left dorsolateral prefrontal cortex (DLPFC) in a sixty-ADHD-patient parallel, randomized, double-blind, sham-controlled trial [7]. The results also suggested that the effects of tDCS are selective.

In Yoon's (2014) study on tDCS and pain relief for spinal cord injury, the results suggested that tDCS can modulate emotional and cognitive components of pain, and also normalize how much attention the brain has on pain related information [25]. This was achieved through anodal stimulation of the motor cortex. Loo (2012) found that in their treatment of depressed patients, there were subjects that received additional benefits. Some subjects reported improved attention and concentration in the weeks of active stimulation. There was one subject with a long standing problem with visual tracking who reported improvements [26].

Based on the work of Clark et al. (2011) and Oullet et al. (2015), it has been shown that "Transcranial direct current stimulation (tDCS) has been found to produce significant changes in behavior, including a large increase of learning and performance for a difficult visual perceptual task" [27]. Moreover, it was determined that tDCS is a safe, inexpensive technology that is easy to use and can improve decision-making towards less risky choices after a 30-minute application on the orbitofrontal cortex (OFC) of healthy patients. The cognitive impulse control of these

patients was also shown to have improved according to the study. This has potential for enhancing learning experience and helping psychological treatment. It presents possibilities for immersive virtual reality as a training method [28].

1.6 Outline of Dissertation

Chapter 2 aims to complete the first research objective of creating a fast and streamline way to analyze multiple sessions and sets of data with minimal user input. Chapter 2 introduces and explains the scripts that were utilized throughout this dissertation. The methods for using the scripts are laid out and thoroughly explained. Using the scripts, De-identified and uniquely named files were automatically loaded from folders, and then filtered and split based on user preferences, a task that would otherwise have to be done manually for each session of each subject in each study. This completes the objective of cleaning the data of noise and artifacts, as part of the analysis. The framework for these scripts is expanded in chapters 3 and 4, building onto the basic model described in Chapter 1, utilizing the scripts in their own unique way. These subsequent chapters demonstrate the modularity and the reach of the script developed by completing analyses on dissimilar data. By utilizing these scripts along with the methods described, it is possible to streamline the filtering and analysis of EEG datasets. The amount of time saved grows with the number of data sets needed to be analyzed, which allows for further examination of the data, and more preparation for future sessions and studies based on the information learned.

Chapters 3 and 4 build on the analysis presented in the Chapter 2 by adding additional layers to the investigation. In that way, it is shown how each script can be modified and used in dissimilar studies by editing the scripts based on the outcomes desired. By utilizing scripts, it is possible to be more consistent on the data analysis between EEG studies, as well as to permit

other researchers to identify the type of analysis preformed and then build on it. As studies utilize these scripts and fine-tune them to their needs within and outside of EEG, the more applications to different sciences will become possible. This will also lead towards a consensus between studies using the scripts, allow for concentric studies to take place between research studies using the scripts put in place.

Chapter 3 discusses how tDCS affects the subjects, and the nature of the EEG data collected from such subjects. Further explanations of the filtering and data processing related to this study is explained. The chapter explains the development of a method to determine the generators within the brain. Using sLORETA, the sources of the EEG signals are determined and then analyzed to better understand how the brain adapts to the task at hand, as well as the overall effects of tDCS stimulation. The final analysis presented in Chapter 3 completes the second to last research objective of analyzing the data, as a whole, with a method to account for the fuzziness of the data. This analysis is then modified to be modular, and prepared for future studies, completing the first half of the fifth and final research objective. Chapter 3 utilizes tDCS data collected during a previous investigation. From 2015 to 2017, Drs. Rick Houser and Daniel Fonseca collaborated on the research study, *The use of Mobile EEG and Stationary EEG Amplifiers in a Pilot Study Focused on the Impact of Low Current Brain Stimulation on Math Understanding and Calculations*. Data sets from patients stimulated by tDCS were collected using a 64-channel mobile EEG amplifier. During the study, undergraduate math students were monitored. They were divided into two groups, one of which was the control group and the other the experimental group. All the participants consisted of freshman and sophomore engineering students at The University of Alabama. The students were enrolling in a pre-calculus algebra course (i.e. Math 112). Phase one included a baseline assessment of the study group, in which

basic EEG data was taken, without tDCS. After the baseline recording were taken, a video discussing intermediate algebra calculations was shown, while still taking EEG data. Two baseline assessments were made with different videos.

Chapter 4 involves a test of the modularity of the scripts constructed in Chapter 3, completing the final objective of instilling modularity into the programming in order to allow multiple types of analyses with easy and minimal effort. The approach used is tested with a new set of data from that used in Chapter 3. The domain data used in Chapter 4 comes from a law enforcement officer study that consisted of law enforcement from a small-sized city located in the Southern U.S. Both expert and novice deputies were tested. Training officers identified police officers who are considered to be experts in shoot or not-to-shoot situations within high threat environments. Novice deputies are those who have recently been hired and are just starting training in law enforcement academy. Each participant completed 12 scenarios. A virtual reality range was utilized to simulate the high threat scenarios that require split second decision making. The police officer study was done under different conditions and requires special allowances for the data to be truly understood. With the motion of the officer and the nature of the study, the analysis differs enough for this to be a true test of the systems setup in Chapter 3. By completing this analysis, it should be possible to see how the officers' brains reacted as the scenarios progressed up until the point of the shot. The conducted research accomplished a fast and streamline way to analyze a multitude of EEG scenarios, and data with minimal changes to the script and approach.

Chapter 5 discusses the conclusions drawn from this research, along with insights gained through the process. The contributions to EEG analysis are also discussed, giving an explanation of ways to enhance or expedite the analysis of similar studies. The final section of the chapter

discusses recommendations for future research. This includes ways to improve on the methods presented and ways to utilize them in new scopes of research.

Chapter 1 References

[1] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, "EEG signal classification for BCI applications by wavelets and interval type-2 fuzzy logic systems," *Expert Systems with Applications,* vol. 42, no. 9, pp. 4370-4380, 2015.

[2] D. Rangaprakash and N. Pradhan, "Study of phase synchronization in multichannel seizure EEG using nonlinear recurrence measure," *Biomedical Signal Processing and Control,* vol. 11, pp. 114-122, 2014.

[3] M. A. Jatoi, N. Kamel, A. S. Malik, I. Faye, and T. Begum, "A survey of methods used for source localization using EEG signals," *Biomedical Signal Processing and Control,* vol. 11, pp. 42-52, 2014.

[4] J. H. Park, S. B. Hong, D. W. Kim, M. Suh, and C. H. Im, "A Novel Array-Type Transcranial Direct Current Stimulation (tDCS) System for Accurate Focusing on Targeted Brain Areas," *IEEE Transactions on Magnetics,* vol. 47, no. 5, pp. 882-885, 2011.

[5] H. L. Filmer, P. E. Dux, and J. B. Mattingley, "Applications of transcranial direct current stimulation for understanding brain function," *Trends in Neurosciences,* vol. 37, no. 12, pp. 742-753, 2014.

[6] C. A. Dockery, R. Hueckel-Weng, N. Birbaumer, and C. Plewnia, "Enhancement of Planning Ability by Transcranial Direct Current Stimulation," *The Journal of Neuroscience,* vol. 29, no. 22, pp. 7271-7277, 2009.

[7] C. Cosmo *et al.*, "Spreading effect of tDCS in individuals with attention-deficit/hyperactivity disorder as shown by functional cortical networks: a randomized, double-blind, sham-controlled trial," *Frontiers in psychiatry,* vol. 6, 2015.

[8] J. Jorge, W. Van Der Zwaag, and P. Figueiredo, "EEG–fMRI integration for the study of human brain function," *NeuroImage,* vol. 102, pp. 24-34, 2014.

[9] J. C. Horvath, O. Carter, and J. D. Forte, "Transcranial direct current stimulation: five important issues we aren't discussing (but probably should be)," *Frontiers in systems neuroscience,* vol. 8, p. 2, 2014.

[10] K.-A. Ho *et al.*, "The Effect of Transcranial Direct Current Stimulation (tDCS) Electrode Size and Current Intensity on Motor Cortical Excitability: Evidence From Single and Repeated Sessions," *Brain Stimulation,* vol. 9, no. 1, pp. 1-7, 2016.

[11] G. B. Saturnino, A. Antunes, and A. Thielscher, "On the importance of electrode parameters for shaping electric field patterns generated by tDCS," *Neuroimage,* vol. 120, pp. 25-35, 2015.

[12] J. C. Peters *et al.*, "On the feasibility of concurrent human TMS-EEG-fMRI measurements," *Journal of neurophysiology,* vol. 109, no. 4, pp. 1214-1227, 2013.

[13] C. A. Nelson, "Incidental findings in magnetic resonance imaging (MRI) brain research," *The Journal of Law, Medicine & Ethics,* vol. 36, no. 2, pp. 315-319, 2008.

[14] Z. Jebri, I. B. Majek, C. Delafosse, C. Pasquet, and Y. Ousten, "A new non-magnetic trimmer for the magnetic resonance imaging system," in *2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2018, pp. 1-5: IEEE.

[15] V. Solanki, M. V. Patel, and M. S. Pati, "Brain MRI Image Classification using Image Mining Algorithms," in *2018 Second International Conference on Computing Methodologies and Communication (ICCMC)*, 2018, pp. 516-519: IEEE.

[16] E. Niedermeyer and F. L. da Silva, *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.

[17] A. R. Riela, "Topographic brain mapping of EEG and evoked potentials: Edited by Dr. K. Maurer. 576 pages. Dm 198.00. Berlin: Springer-Verlag, 1989," ed: Elsevier, 1990.

[18] H. Laufs, "A personalized history of EEG–fMRI integration," *Neuroimage,* vol. 62, no. 2, pp. 1056-1067, 2012.

[19] C. M. Michel, M. M. Murray, G. Lantz, S. Gonzalez, L. Spinelli, and R. G. de Peralta, "EEG source imaging," *Clinical neurophysiology,* vol. 115, no. 10, pp. 2195-2222, 2004.

[20] J. Hu, C.-s. Wang, M. Wu, Y.-x. Du, Y. He, and J. She, "Removal of EOG and EMG artifacts from EEG using combination of functional link neural network and adaptive neural fuzzy inference system," *Neurocomputing,* vol. 151, pp. 278-287, 2015.

[21] Z. Fatima, M. A. Quraan, N. Kovacevic, and A. R. McIntosh, "ICA-based artifact correction improves spatial localization of adaptive spatial filters in MEG," *Neuroimage,* vol. 78, pp. 284-294, 2013.

[22] C. J. Stagg and M. A. Nitsche, "Physiological Basis of Transcranial Direct Current Stimulation," *The Neuroscientist,* vol. 17, no. 1, pp. 37-53, 2011.

[23] C. A. Terzuolo and T. H. Bullock, "Measurement Of Imposed Voltage Gradient Adequate To Modulate Neuronal Firing," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 42, no. 9, pp. 687-694, 1956.

[24] T. Wagner, A. Valero-Cabre, and A. Pascual-Leone, "Noninvasive Human Brain Stimulation," *Annual Review of Biomedical Engineering,* vol. 9, no. 1, pp. 527-565, 2007.

[25] E. J. Yoon, Y. K. Kim, H.-R. Kim, S. E. Kim, Y. Lee, and H. I. Shin, "Transcranial Direct Current Stimulation to Lessen Neuropathic Pain After Spinal Cord Injury: A Mechanistic PET Study," *Neurorehabilitation and Neural Repair,* vol. 28, no. 3, pp. 250-259, March 1, 2014 2014.

[26] C. K. Loo, A. Alonzo, D. Martin, P. B. Mitchell, V. Galvez, and P. Sachdev, "Transcranial direct current stimulation for depression: 3-week, randomised, sham-controlled trial," *The British Journal of Psychiatry,* vol. 200, no. 1, pp. 52-59, 2012.

[27] V. P. Clark, B. A. Coffman, M. C. Trumbo, and C. Gasparovic, "Transcranial direct current stimulation (tDCS) produces localized and specific alterations in neurochemistry: A 1H magnetic resonance spectroscopy study," *Neuroscience Letters,* vol. 500, no. 1, pp. 67-71, 8/1/ 2011.

[28] J. Ouellet, A. McGirr, F. Van den Eynde, F. Jollant, M. Lepage, and M. T. Berlim, "Enhancing decision-making and cognitive impulse control with transcranial direct current stimulation (tDCS) applied over the orbitofrontal cortex (OFC): A randomized and sham-controlled exploratory study," *Journal of Psychiatric Research,* vol. 69, pp. 27-34, 2015.

CHAPTER 2

DEVELOPMENT OF AN AUTOMATED MATLAB-BASED PLATFORM FOR THE ANALYSIS OF MASSIVE EEG DATASETS

EEG studies consist of multiple sessions. Individually loading and processing each data set is repetitive and time consuming. Finding the action markers embedded into the data can be difficult since the times of these markers vary greatly between individuals. These markers indicate sections that need to be analyzed. Based on the particularities of each marker, individual data segments or sections have to be treated differently or independently from each other. Therefore, splitting an EEG data file in order to compare sections or even analyze one section independently can be beneficial. However, de-identified and uniquely named files can be difficult to run through a program since they usually follow ununiformed naming conventions. While file re-naming is an option, this adds additional steps into identifying the nature and uniqueness of the data. Also, the manually re-naming of data files is prone to error in larger studies. All of these challenges add up to considerable hurdles and difficulties when processing and filtering data originated from multiple EEG sessions. A series of MATLAB computer scripts developed by the authors of this paper address these problems.

2.1 Problem Background

Electroencephalography (EEG) is a device that is used to better understand the inner workings of the human brain. The bioelectrical potentials generated by the cortex nerve cells within the brain are utilized to identify which areas of the brain are active [1]. EEG is used in a

non-invasive way, only needing contact with the scalp to record data. EEG's importance is shown in the understanding of neurological disorders, such as, epilepsy, tumors, locating head damages, among others [2, 3]. EEG source imaging is utilized in many important applications in cognitive neuroscience and clinical neuroscience. The clinical areas include neurology, psychiatry, and psychopharmacology. Cognitive neuroscience applications mainly focus on investigating the temporal aspects by analyzing event related potentials (ERP). In this sense, neurology focuses more on the study of motor evoked potential, but in clinical applications, the main concern is localizing epileptic foci. In psychiatry and psychopharmacology, the focus tends to be towards localizing the sources in certain frequency bands. While all of these applications aim to localize the sources, the pre- processing of the data is somewhat different for each application [4].

EEG signals are usually considered a measurement of neural activity, but they do not give a full understanding of the underlying neurophysiology at work. Neural activity in the brain is mediated by the synaptic interactions and the resulting action potential propagation. This means that EEGs signals are the excitation of a neuron traveling from its axon towards the axon terminal. This is where the action potentials are transmitted to other neurons, muscle cells or glands [5]. These surface electrodes do not exactly show where the electrical signals are coming from. By knowing the generators, it is possible to predict which electrodes will be activated from each given generator. When a channel is activated, it is more difficult to determine which generator in the brain caused the excitation. The channels are based on the surface of the scalp and therefore cannot convey some of the deeper activity within the brain.

EEG studies contain numerous files of data. This type of data can have markers to indicate sections that need to be analyzed differently or separately from each other. It can be

difficult to stop recording during sessions, therefore, markers are used to signal beginnings and ends. In addition to these challenges, there is also the fact that studies can contain de-identified and uniquely named files that cannot be easily organized into a list to run through a script. All of these challenges build on each other when it is necessary to process and filter multiple sessions.

2.2 The Need for Makers in EEG Data: A Case Study

The MATLAB data process shown below has applicability with EEG data across disciplines. For example, the data presented in this case study are examples from two multidisciplinary research projects. The first project involved the neurological analysis of brain-wave activation patterns of college students during an immersive, mindfulness experience. The second project examined law enforcement officers (LEOs) decision-making during high threat situations. The project associated with the analysis method will be denoted in text and with the corresponding figures and tables.

For the mindfulness study, EEG data were recorded as participants completed two conditions: (a) a baseline condition and (b) virtual-reality, mindfulness condition. Each condition lasted approximately 10-minutes in length. EEG data was collected using a 64-channel mobile EEG amplifier, EEGO Sports [6]. The unit records data using EEG caps and electrodes attached to a high-performance Windows 8 tablet. The mobile EEG unit has been used to record physiological and neurological data on participants who move frequently (e.g., athletes, LEOs). It was chosen to collect data from 20 channels focused on studying neural processes while engaged in firing a weapon [7]. The reference electrode was CPz and the 19 channels utilized were Fp1, Fp2, F7, F3, Pz, F4, F8, T3, C3, Cz, C4, T4, T5, P3, Pz, P4, T7, T8, T6, TP7, TP8, O1 and O2. Data was collected at a sampling rate of 500 Hz in an ambient temperature room.

Pre-processing of EEG data was conducted in **asa**lab software package [6] and EEGLAB toolbox for MATLAB [8]. Using **asa**lab, a lowpass filter of 30 Hz to filter extraneous noise as first utilized. Data was then converted using EEGLAB toolbox for MATLAB [8] which was used for the remainder of analysis. Continuous data as visually inspected to look for channels affected by superfluous sources of noise. Independent Component Analysis (ICA) was utilized in EEGLAB [9] to re-reference problematic channels and to remove artifacts (e.g., eye blinks, muscle movements). ICA has been shown to be useful in isolating and removing eye and movement artifacts commonly found in data where some participant movement was involved, as was the case in the current study [4].

2.3 Approach

EEG studies consist of multiple sessions. Individually loading and processing each data set is repetitive and time consuming. Finding the action markers embedded into the data can be difficult since the times of these markers vary greatly between individuals. These markers indicate sections that need to be analyzed. Based on the particularities of each marker, individual data segments or sections have to be treated differently or independently from each other. Therefore, splitting the file in order to compare sections or even analyze one section independently can be beneficial.

When a marker designates the beginning of a distinctive segment of data, it becomes possible to extract the aforementioned section. Deidentified and uniquely named files can be difficult to run through a program since they usually follow ununiformed naming conventions. While file re-naming is an option, this adds additional steps into identifying the nature and uniqueness of the data. Also, the manually re-naming of data files is prone to error in larger studies. All of these challenges add up to considerable hurdles and difficulties when processing

and filtering data originated from multiple EEG sessions. A series of computer scripts developed by the authors of this paper address these problems (MATLAB scripts shown in appendix). The first script was programed to quickly look through a main folder, detecting and ultimately reading subsequent subfolders inside the main one, as shown in Figure 2.1. It only sorts through subfolders, so data in the main folder needs to be sectioned into a subfolder. Due to Microsoft's Windows naming convention, if numbers are used while naming the files, it is then necessary to have leading zeros that match the digits of the biggest number, i.e., if the study in interest has subjects in the double digits, files should then be named 01, 02, 03, etc. For triple digits, 001, 002, 003, etc. This is due to how the '.' character interacts with the list. Normal alphabetical naming conventions should work as expected.

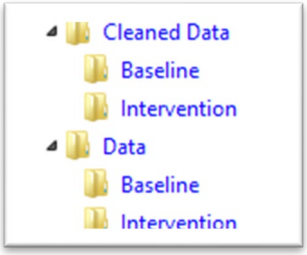

Figure 2.1. Folder structure for data import.

The second script developed is related to the event splitting code, created for the LEO study. An event's start and end terms are specified by the user. The script collects all markers in the file from the table created when the data is loaded, as shown in Figure 2.2. The script then looks for the start and end terms. It is also possible to add a shoot term, which is a third term that ends the section early. This is useful for data that becomes too noisy after a certain trigger but before the end of the data collection. The data will be collected in as shown in Figure 2.3 to be used later in the script. EEG data following the shooting events is called the reflection data. This is the time between the ending and the next beginning of a session. Therefore, the script's output is the scenario from start to end, or start to shoot, if it is marked to end early. Scenario data is

named with an S before the data name in the output folder. There is also an end to start reflection data set saved with an R before the data name, as shown in Figure 2.4. For the final reflection a "trueend" term was added for the last iteration, where the final end will then match up with the true end, giving the last reflection.

Variables - EEG.event

EEG | EEG.event

EEG.event

Fields	latency	type	duration
1	1	'0, Impedance'	0
2	1	'1025, 1 school library start (shoot)'	0
3	5001	'Scenario 1 shot'	0
4	8501	'1026, 1- school libray end'	0
5	11501	'StartS2'	0
6	13501	'shot'	0
7	15001	'end'	0
8	16501	'Start S3'	0
9	19001	'end'	0
10	20001	'start s4'	0
11	23501	'shot'	0
12	26501	'end'	0
13	28501	'start s5'	0
14	31501	'end'	0

Figure 2.2. EEG event variable created when a file is loaded.

Variables - start

start

6x2 cell

	1	2
1	'1025, 1 school library start (shoot)'	1
2	'StartS2'	11501
3	'Start S3'	16501
4	'start s4'	20001
5	'start s5'	28501
6	'trueend'	62165

Variables - endd

endd

5x2 cell

	1	2
1	'1026, 1- school libray end'	8501
2	'end'	15001
3	'end'	19001
4	'end'	26501
5	'end'	31501

Variables - shot

shot

3x2 cell

	1	2
1	'Scenario 1 shot'	5001
2	'shot'	13501
3	'shot'	23501

Figure 2.3. (Left) Start terms including the "trueend" term for retrieving the final reflection. (Right) End terms. (Bottom) The shot terms.

Name	Date modified	Type	Size
R1.fdt	8/23/2018 1:21 AM	FDT File	235 KB
R1.set	8/23/2018 1:21 AM	SET File	33 KB
R2.fdt	8/23/2018 1:21 AM	FDT File	118 KB
R2.set	8/23/2018 1:21 AM	SET File	30 KB
R3.fdt	8/23/2018 1:21 AM	FDT File	79 KB
R3.set	8/23/2018 1:21 AM	SET File	29 KB
R4.fdt	8/23/2018 1:21 AM	FDT File	157 KB
R4.set	8/23/2018 1:21 AM	SET File	31 KB
R5.fdt	8/23/2018 1:21 AM	FDT File	2,383 KB
R5.set	8/23/2018 1:21 AM	SET File	87 KB
S1.fdt	8/23/2018 1:21 AM	FDT File	391 KB
S1.set	8/23/2018 1:21 AM	SET File	37 KB
S2.fdt	8/23/2018 1:21 AM	FDT File	157 KB
S2.set	8/23/2018 1:21 AM	SET File	31 KB
S3.fdt	8/23/2018 1:21 AM	FDT File	196 KB
S3.set	8/23/2018 1:21 AM	SET File	32 KB
S4.fdt	8/23/2018 1:21 AM	FDT File	274 KB
S4.set	8/23/2018 1:21 AM	SET File	34 KB
S5.fdt	8/23/2018 1:21 AM	FDT File	235 KB
S5.set	8/23/2018 1:21 AM	SET File	33 KB

Figure 2.4. Output of the Police Split Script for one file.

The third script consists of filtering and a mean-power analysis created for the mindfulness study. Mean-power analysis involves calculating the average power frequency for a relative time period. Such analyses have been utilized in prior mindfulness-based research [10].

The wave frequencies to be analyzed are set at the beginning of the script, along with other data to be initialized. In order to use the interpolation, channel locations need to be established. The process of interpolation involves replacing the power frequency of an unusable electrode with an estimate of the power frequency from adjacent and useable electrodes. Currently, this is done by a file called 'Fullset.mat' which loads up channel locations specific to the study. This can be made from saving the EEG.chanlocs from a dataset with all desired channels after setting the channel locations in the edit drop down menu. To do so, create a new variable using the following “Fullset=EEG.chanlocs;” or “Fullset=ALLEEG.chanlocs;” based on how the EEG data has been saved in the workspace. Then right click on Fullset and save as, in this case, Fullset.mat. The data should appear as in Figure 2.5.

ChLocs.Fullset

ChLocs.Fullset

Fields	labels	type	theta	radius	X	Y	Z	sph_theta	sph_phi	sph_radius	urchan	ref
1	'Fp1'	[]	-17.9260	0.5150	80.7840	26.1330	-4.0011	17.9260	-2.6980	85	1	[]
2	'Fp2'	[]	17.9260	0.5150	80.7840	-26.1330	-4.0011	-17.9260	-2.6980	85	2	[]
3	'F7'	[]	-53.9130	0.5281	49.8714	68.4233	-7.4895	53.9130	-5.0550	85	3	[]
4	'F3'	[]	-39.9470	0.3446	57.5511	48.2004	39.8697	39.9470	27.9730	85	4	[]
5	'Fz'	[]	0	0.2534	60.7385	0	59.4629	0	44.3920	85	5	[]
6	'F4'	[]	39.8970	0.3445	57.5840	-48.1426	39.8920	-39.8970	27.9900	85	6	[]
7	'F8'	[]	53.8670	0.5281	49.9265	-68.3836	-7.4851	-53.8670	-5.0520	85	7	[]
8	'T7'	[]	-90	0.5332	5.1765e-15	84.5385	-8.8451	90	-5.9730	85	8	[]
9	'C3'	[]	-90	0.2667	3.8681e-15	63.1713	56.8717	90	41.9960	85	9	[]
10	'Cz'	[]	0	0	5.2047e-15	0	85	0	90	85	10	[]
11	'C4'	[]	90	0.2667	3.8679e-15	-63.1673	56.8761	-90	42	85	11	[]
12	'T8'	[]	90	0.5332	5.1765e-15	-84.5385	-8.8451	-90	-5.9730	85	12	[]
13	'P7'	[]	-126.0870	0.5281	-49.8714	68.4233	-7.4895	126.0870	-5.0550	85	13	[]
14	'P3'	[]	-140.0530	0.3446	-57.5511	48.2004	39.8697	140.0530	27.9730	85	14	[]
15	'Pz'	[]	180	0.2534	-60.7385	-7.4383e-15	59.4629	-180	44.3920	85	15	[]
16	'P4'	[]	140.1030	0.3445	-57.5840	-48.1426	39.8920	-140.1030	27.9900	85	16	[]
17	'P8'	[]	126.1330	0.5281	-49.9265	-68.3836	-7.4851	-126.1330	-5.0520	85	17	[]
18	'O1'	[]	-162.0740	0.5150	-80.7840	26.1330	-4.0011	162.0740	-2.6980	85	18	[]
19	'O2'	[]	162.0740	0.5150	-80.7840	-26.1330	-4.0011	-162.0740	-2.6980	85	19	[]
20	'Oz'	[]	180	0.5067	-84.9812	-1.0407e-14	-1.7860	-180	-1.2040	85	20	[]

Figure 2.5. Channel locations from Fullset.mat.

The subsequent portion of this script is the file reader which starts processing all the subfolders found, while going through all the wave frequencies specified and subjects/scenarios found in the subfolders. The first part in this process is pop_chanedit which assigns channel locations necessary for data interpolation. Next is the pop_rejcont which rejects all the data outside of the defined parameters. This may need to be changed based on how the data needs to be cleaned. Next to be triggered are pop_runica which runs independent component analysis

(ICA) on the data followed by pop_rejchan which removes any faulty channels. ICA is a commonly utilized filtering process that removes data artifacts (e.g., eye blinks, muscle movements), commonly found in EEG data — particularly in studies where participants move [11]. The lost channels can be later interpolated back into the analysis with pop_interp by taking data from nearby channels and using them to calculate new data for the removed channel. The final part will be to re-refence the data in order to get a more accurate reference for the data. This will reference the data based on the mean of all the data rather than the current reference, which can vary between studies. This is possible by using the command pop_reref. These commands can be modified to fit most conventional EEG data processing studies. Typing "help (command)" displays how each of these programmed commands can be modified to meet the analyst's particular needs.

The EEG data sets are further processed to find the mean power using spectopotest which is a version of spectopo edited to output the numbers in base 10 rather than in decibels (dB). This was done by editing instances of eegspecdB = 10*log10(eegspecdB) to eegspecdB = eegspecdB. When the script is done running, the command window displays the found bad channels, if interpolation was disabled. The final numbers are then collected in fulleasycopy as shown in Figure 2.6.

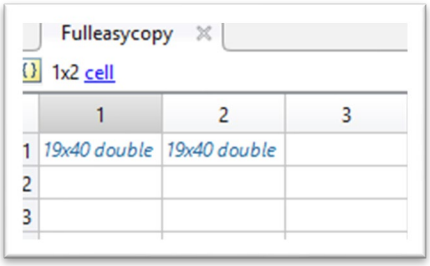

Figure 2.6. Data shown when fulleasycopy variable is double clicked.

Figure 3 shows two folders of 19 subjects. Nineteen baselines and 19 intervention data sets when these cells are double clicked. To see the final data array, the cell can be double

clicked to show data similar to that in Figure 2.7. Two wave analyses were run on the shown data. When more waves are analyzed, the data add onto the right side of the cell based on how the wave analysis was defined. This allows for the data to be easily copied into Microsoft Excel. The top labels should be the channels as stated in the channel_list variable. The left labels can be found in the ColFilesT variable. Alternatively, the FolderMeanpowerarray splits the data into wave analysis sections, as shown in Figure 2.8. These numbers can be accessed by double clicking the blue 19x20 double cells.

Fulleasycopy{1, 1}

Fulleasycopy{1, 1}

	1	2	3	4	5	6	7	8	9	10	11
1	3.9468	4.3709	22.0945	7.4422	5.0522	6.0097	21.8327	6.7397	4.8309	6.6905	2.9024
2	23.8292	7.7023	5.7081	12.0396	3.2276	7.4455	11.9108	5.3011	2.4866	2.5591	1.7623
3	5.7663	4.9444	13.5821	8.7341	8.6073	8.7905	11.3070	5.9854	1.2009	3.9065	2.2629
4	108.8334	55.9710	39.6639	11.0944	7.7033	7.4316	21.3473	9.3594	3.8046	3.0427	4.4247
5	95.6168	76.8766	42.6833	15.2643	8.1345	16.7222	8.3372	10.2986	7.5640	9.3598	11.3360
6	5.7796	4.5844	25.3412	14.0428	10.2522	22.7822	13.6600	14.7553	1.9366	2.1115	12.0985
7	9.4705	9.7959	2.3574	3.8597	3.5800	4.2986	2.8313	3.6322	1.8217	3.5410	1.7354
8	39.5108	41.6839	11.8123	7.5944	6.3926	5.7833	18.0867	3.1011	3.7598	2.5353	6.0586
9	23.4871	35.9805	3.1361	3.2872	3.0227	4.7717	6.3994	7.5886	1.7279	2.0106	11.0419
10	4.6109	4.5266	23.8323	5.9205	5.6291	5.7399	21.2794	9.3339	2.8487	3.8003	3.8459
11	7.2608	6.2844	22.3802	11.9421	10.4463	9.3560	29.5898	7.9433	4.1080	6.7986	6.0237
12	17.2582	90.3068	48.9411	14.8613	6.5558	4.3200	5.4354	3.8745	2.2046	2.4803	2.4162
13	5.8580	5.2097	16.6946	11.7317	6.1435	11.7796	18.4952	7.8740	4.4602	7.1006	4.0900
14	53.3036	47.8963	23.9647	5.1619	4.1572	4.2258	16.5447	4.9388	3.2170	5.0096	3.1187
15	17.2582	90.3068	48.9411	14.8613	6.5558	4.3200	5.4354	3.8745	2.2046	2.4803	2.4162
16	10.2173	44.8408	11.0234	4.4245	7.0612	5.0921	20.9892	5.6575	3.1902	2.9506	1.7021
17	109.1776	54.2163	38.8642	7.5388	7.8960	10.8633	20.9961	8.9045	3.5572	4.5327	4.8665
18	27.3404	6.5087	4.9963	12.1372	3.7555	7.7258	12.3344	5.5451	2.6254	2.8135	1.9430
19	7.0143	7.1512	23.3461	17.7667	13.1693	8.1831	32.7841	28.4559	24.7770	6.3024	8.2640
20											

Figure 2.7. The first 11 of 40 (for this case) columns of Fulleasycopy{1,1} cell array expanded. For 20 channels it will repeat at the first channel starting at column 21.

FolderMeanpowerarray

1x2 cell

	1	2	3
1	3x4 cell	3x4 cell	
2			
3			

FolderMeanpowerarray{1, 1}

FolderMeanpowerarray{1, 1}

	1	2	3	4
1	0.5000	3.1000	8	12.1000
2	3	7.9000	12	24
3	19x20 double	19x20 double	19x20 double	19x20 double

Figure 2.8. Left: FolderMeanpowerarray Structure. Right: data displayed from double clicking the 3x4 cell. Data in the 19x20 double should appear similar to Figure 4.

2.4 Results and Final Remarks

As already mentioned, the EEG data processing demands that ultimately led to the creation of the discussed MATLAB script consisted of the baselines and interventions readings for 20 channels. Interventions and baselines were both ten minutes long on average. The script used an average of 1.2 GB of memory on a 16 GB RAM desktop during the analysis for the mindfulness study. It was used to run the mean power analysis over four different waves. The waves analyzed were Delta, Theta, Alpha, and Beta. The data used for this timed trail was previously put through independent component analysis (ICA) and saved. It is worth noting that ICA can be run on data sets and saved outside of running the full script. This saves time on multiple iterations using the same data sets. The full execution of this script took 27 minutes. This was fully automated, requiring no further user inputs after the start of the script. It was still necessary to extract the data from the workplace, which takes no more than a minute.

Overall, there were four separate wave analyses for each of the 38 data sets consisting of 20 channels in each. This is including the baselines for a total of 3,040 calculations. Therefore, the script accounts for over 100 calculations per minute. This far outclasses what could be done using the normal graphical user interface (GUI) approach. In addition to the calculations, the data is collected in a way that allows for easy export and further analysis. It is also worth mentioning that the scripts created are modular. It is possible to swap out the mean power analysis for

another type of analysis. It would also be possible to add code to automatically save the data from the workspace as Excel documents with channel and subject labels based on the user needs. The scripts are written and commented in a manner that shows a logical progression and facilitates the editing of sections of script for different studies.

Chapter 2 References

[1] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, "EEG signal classification for BCI applications by wavelets and interval type-2 fuzzy logic systems," *Expert Systems with Applications,* vol. 42, no. 9, pp. 4370-4380, 2015.

[2] M. A. Jatoi, N. Kamel, A. S. Malik, I. Faye, and T. Begum, "A survey of methods used for source localization using EEG signals," *Biomedical Signal Processing and Control,* vol. 11, pp. 42-52, 2014.

[3] D. Rangaprakash and N. Pradhan, "Study of phase synchronization in multichannel seizure EEG using nonlinear recurrence measure," *Biomedical Signal Processing and Control,* vol. 11, pp. 114-122, 2014.

[4] C. M. Michel, M. M. Murray, G. Lantz, S. Gonzalez, L. Spinelli, and R. G. de Peralta, "EEG source imaging," *Clinical neurophysiology,* vol. 115, no. 10, pp. 2195-2222, 2004.

[5] J. Jorge, W. Van Der Zwaag, and P. Figueiredo, "EEG–fMRI integration for the study of human brain function," *NeuroImage,* vol. 102, pp. 24-34, 2014.

[6] F. Zanow and T. R. Knösche, "Asa-advanced source analysis of continuous and event-related eeg/meg signals," *Brain topography,* vol. 16, no. 4, pp. 287-290, 2004.

[7] C. A. Domingues *et al.*, "Alpha absolute power: motor learning of practical pistol shooting," *Arquivos de neuro-psiquiatria,* vol. 66, no. 2B, pp. 336-340, 2008.

[8] A. Delorme and S. Makeig, "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *Journal of neuroscience methods,* vol. 134, no. 1, pp. 9-21, 2004.

[9] A. Delorme, T. Sejnowski, and S. Makeig, "Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis," *Neuroimage,* vol. 34, no. 4, pp. 1443-1449, 2007.

[10] J. Lagopoulos *et al.*, "Increased theta and alpha EEG activity during nondirective meditation," *The Journal of Alternative and Complementary Medicine,* vol. 15, no. 11, pp. 1187-1192, 2009.

[11] T. Thompson, T. Steffert, T. Ros, J. Leach, and J. Gruzelier, "EEG applications for sport and performance," *Methods,* vol. 45, no. 4, pp. 279-288, 2008.

CHAPTER 3

FUZZY SET ANALYSIS OF ELECTROENCEPHALOGRAM DATA PERTAINING TO TDCS IN MATH SKILL ENHANCEMENT: SOURCE LOCALIZATION THROUGH THE INVERSE PROBLEM

Electroencephalography (EEG) data is very complex in nature, but there is much that can be learned from it. This learning can be expanded on by utilizing different methods to refine the analysis. By solving the inverse problem, it is possible to localize the sources of the excitation measured at the scalp of the brain. This analysis is applied to transcranial direct current stimulation (tDCS) data. tDCS is a type of noninvasive neural stimulation. With this analysis, the overall data can be broken down into the Brodmann areas and compared between subjects to determine how tDCS subjects' data differed from control subjects. A conventional EEG study does not have the capability for handling multiple interactions between the Brodmann areas and the human factors involved; therefore, a fuzzy controller will be used to account for the fuzziness of the data.

This research project utilizes tDCS data collected during a previous investigation. From 2015 to 2017, Drs. Rick Houser and Daniel Fonseca collaborated on the research study, *The use of Mobile EEG and Stationary EEG Amplifiers in a Pilot Study Focused on the Impact of Low Current Brain Stimulation on Math Understanding and Calculations*. Data sets from patients stimulated by tDCS were collected using a 64-channel mobile EEG amplifier. During the study, undergraduate math students were monitored.

3.1 Problem Background

3.1.1 Problem Overview

EEG data is susceptible to noise from outside sources and only capture the surface electrode data. The end goal is to get this data into a form that can be analyzed and understood. The initial problem starts with determining how to process and view the data. The data can not be directly interpreted as is. The complexity of this problem requires a sequence of steps, starting with filtering and ending with the final step being an artificial intelligence fuzzy set controller to determine information about the control subjects and the transcranial direct current stimulation (tDCS) subjects. The necessary steps to get there include preprocessing the data and exporting it into EEGLAB, an extension of MATLAB. From there, the data is filtered and exported into LORETA (Low Resolution Brain Electromagnetic Tomography). Then an artificial intelligence fuzzy logic controller will interpret this data and aid in the overall understanding of the data given.

3.1.2 Problem Description

Electroencephalography (EEG) measures the bioelectrical signals within the brain. This is done with the aid of an EEG cap that holds the electrodes in place. The cap then measures the bioelectrical potential relative to a determined ground electrode. These potentials are generated by the cortex nerve cells within the brain [1]. This allows for the collection of brain activity from the surface of the scalp. In this way, it is not necessary to use any invasive methods, as EEG data collection can be done by contacting the surface of the scalp.

Unlike magnetic resonance imaging (MRI), EEG can only collect the electrical signals on the surface of the scalp. In order to find the sources of the EEG signals, sLORETA is able to calculate the origin of the signals within the brain. A downside of sLORETA is that it is

susceptible to noisy data which will affect the analysis. Therefore, it is necessary to do further data processing to fully understand the inner workings of the data collected. EEG data is taken under conditions that allow for outside interface, called noise. This noise can take many forms from electrical interference to small biological interference, such as blinking or moving. These noise artifacts can be removed using EEGLAB.

After the filtering is done, LORETA can be utilized to determine where the electrical potentials came from within the brain. The filtering is necessary for LORETA to provide proper localization of the generators. LORETA will provide a 3-D model of the brain with the excitation and area of excitation detailed at any given point and time desired [2]. The data is recorded and sent back into MATLAB to finish the final section of the problem.

The final interpretation of the data is subject to a level of uncertainty. Due to this, it is a perfect application for an artificial intelligence fuzzy logic controller. This will allow for data to be processed with membership values that are not distinct 0 or 1. With this fluidity in the data analysis, it is possible to complete parts of the analysis before explicitly categorizing certain aspects of the data. Then, the output of the controller will be used to better understand the data, the process of tDCS, and how to improve future sessions.

3.2 The Process of Transcranial Direct Current Stimulation

tDCS is accomplished by polarizing the resting membrane potential. If administered for several minutes, the excitability can outlast the time stimulated, allowing tDCS to be used for neurological and psychiatric disorders. Early studies by Nitsche and Paulus (2000) showed that the effects produced by applying tDCS at 1 mA for 10 minutes while using 35 cm^2 electrodes could last up to one hour after stimulation [3]. It was also shown that higher current intensities produced greater changes to the cortical excitability, between 1 mA and 0.2 mA. Since then, 35

cm^2 electrodes have been in common use. This suggests that a higher current produces a greater and longer lasting effect, but it is uncertain if the current density, and not the intensity, is causing these lasting effects [4].

3.2.1 tDCS Administration

For most studies, the main parameters are current intensity (mA), electrode size (cm^2), electrode placement, and duration. Current density (mA/cm^2) is calculated from dividing the current intensity over the electrode area. Most studies follow the findings of Nitsche and Paulus (2000) about polar specific tDCS where the anode excites, and the cathode inhibits. It is also suggested that higher current intensities and longer stimulation produces stronger and more lasting effects. This was later challenged by further studies that showed that there is a nonlinear relationship between current intensity and stimulation, and further by the lack of difference in efficacy between two or three different current intensities of anodal tDCS [5]. Ho and colleagues (2015) researched the impact of current intensity (mA) and electrode size (cm^2) on motor cortical excitability. These are factors that relate to current density (mA/cm^2), which is considered an important factor in the outcome of tDCS. By manipulating these key parameters, the outcome of the stimulation can be altered [4].

3.2.2 Standard Parameters for Administration of tDCS

tDCS administrating parameters can be changed, but usually they are kept within certain bounds. Current intensity usually ranges from 0.2 mA to 2 mA, for safety reasons. It was also suggested by Ho et al. (2016) that the following are the significant parameter relationships: “1) size of the electrode relative to the size of the target, 2) location of the electric field relative to the target area and 3) direction of the electric field relative to the neuronal orientation” [4].

The size and shape of the electrodes compared to the area of stimulation matters. Tailored electrodes are thought to result in a greater increase in excitability when compared to standard non-tailored electrodes [4]. When an electrode is larger than the target area, it may not effectively stimulate the underlying cortex as the current distribution becomes diffused to the point where it is functionally inert. In common practice, electrodes are placed centered over the hotspot, but computational modeling suggests that the peak current densities are actually in-between the electrodes, closer to the anode. The motor hotspot likely still receives some stimulation, but the peak current densities may be peripheral to the hotspot. The electric field also has a direction, as it travels through the brain, which interacts with the neuronal orientation. The standard montage involves the anode on the motor cortex with the cathode on the contralateral supraorbital area to produce excitability, while the opposite orientation produces no excitability. The mechanisms underlying tDCS and its effects are likely multifactorial, and therefore, it is important to consider all the factors involved. If any factor is suboptimal, such as current direction, the excitability could be reduced or negated all together [4].

3.2.3 Long- and Short-Term Effects of tDCS

tDCS has been performed in hundreds of reported studies to date, all of which have not seen any serious side effects, in healthy controls and patient populations [6, 7]. The extent of side effects is limited to slight itching under the electrode, headache, fatigue, and nausea. Side effects are usually limited to a minority of cases of more than 550 subjects [7, 8]. These studies have shown the safety of tDCS. In patients with skin diseases, it is possible that non-intact skin will experience tissue heating. tDCS has shown no evidence of toxic effects to date over the thousands of subjects worldwide. There have been numerous studies explicitly focused on safety [7-9]. Poreisz et al. (2007) reviewed the adverse effects of 77 healthy subjects and 25 patients

over the course of 567 1-mA stimulations [8]. The side effects experienced were broken down by the following percentages: 75% mild tingling sensations, 30% light itching sensation, 35% moderate fatigue, and 11.8% headache. Most of these did not differ from the placebo stimulation. The most severe side effect reported was skin lesions on the site of the electrode [9].

3.2.4 Efficiency of tDCS and Potential Complications

An optimal set of parameters is not defined for tDCS. It is unclear whether alteration of treatment parameters such as session frequency, number of sessions, or current intensity enhances the efficacy of tDCS. Alonzo et al. (2012) found that over a five-day period, subjects who received tDCS daily, as opposed to twice daily, had greater neuronal excitability. It is important to note that the amount of sessions received and not the frequency of sessions may have favored the results of the daily tDCS [10]. To measure the effects of tDCS, a few methods can be used. The excitability can be measured in the motor cortex using transcranial magnetic stimulation (TMS). Electromyography (EMG) can be used to record the motor evoked potentials (MEPs) [4].

3.2.4.1 tDCS Efficiency and Efficacy

Clark et al. (2012) were able to apply tDCS to reduce the time required for subjects to learn a skill [11]. From their fMRI studies, it was determined that the right inferior frontal (RINF/RIF) and right parietal cortex (RPAR), as well as the temporal, cingulate and other brain regions, make up the parts of the brain involved with concealed threat-related objects in the naturalistic environments. Eighty-three healthy subjects participated in varying levels of tDCS at different scalp locations. The paper represented the data in two forms, the first being depicted in Figure 3.1, which shows the progression of accuracy with successive trials. The second form utilizes fMRI to measure the blood oxygen level dependent (BOLD). The BOLD responds were

compared for the scenes with concealed threats and those without. FMRI values were collected as the participants started as novice and proceeded through intermediate then to expert. The data set is shown in Figure 3.2. Readings were also collected an hour after stimulation. These results are relative to the individual's baseline and were taken immediately after training. The data is depicted in Figure 3.3.

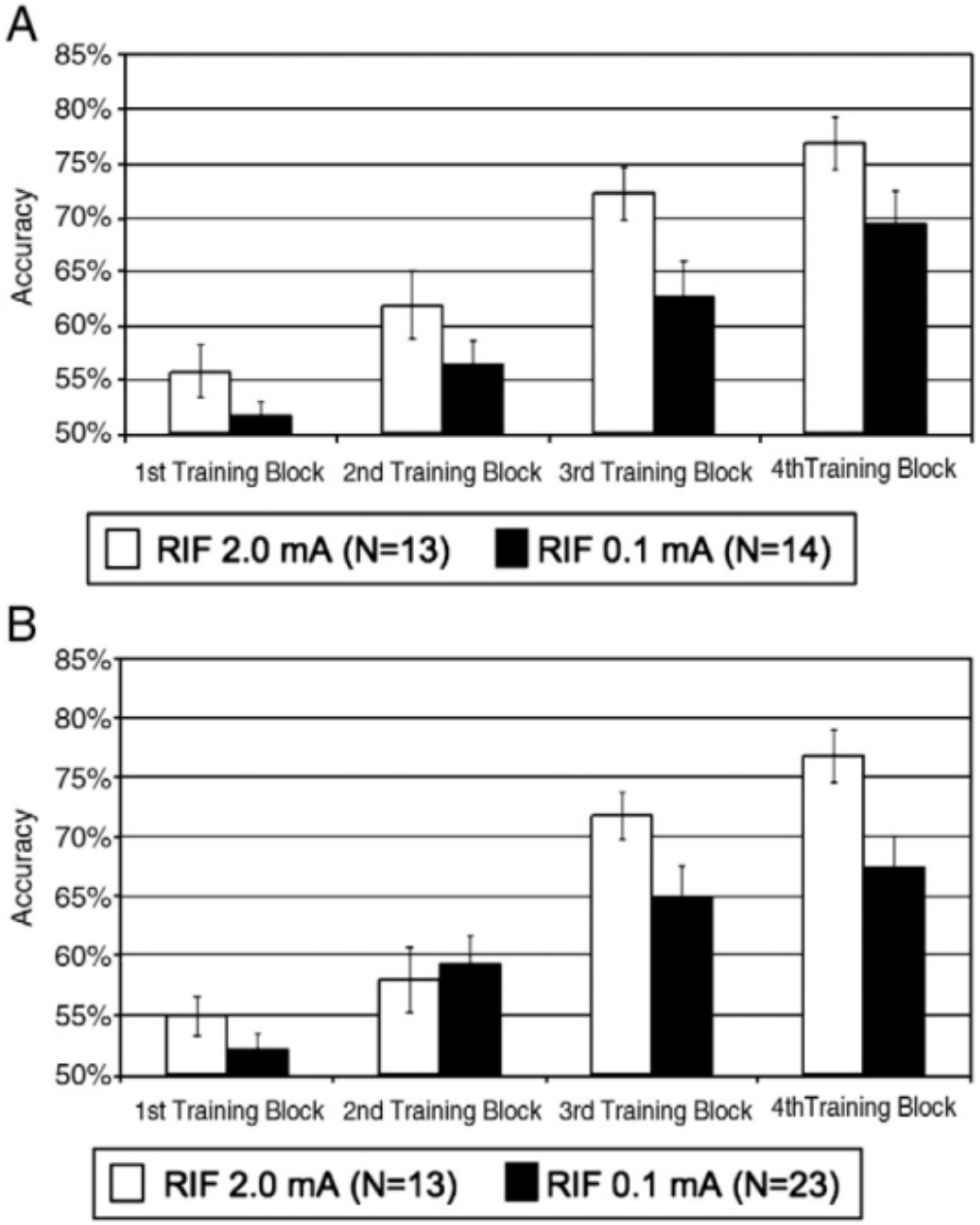

Figure 3.1. Accuracy of successive trials with tDCS [11].

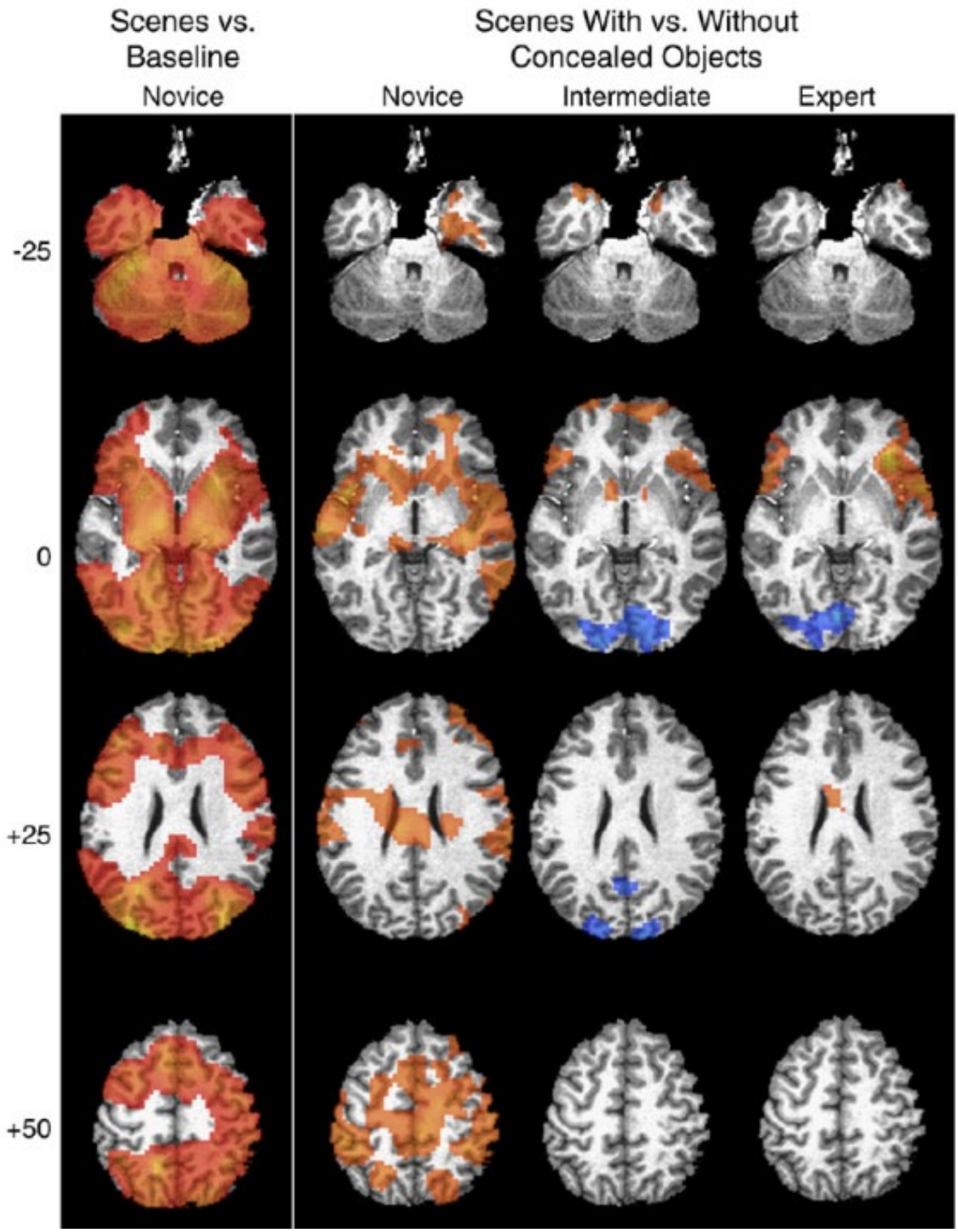

Figure 3.2. fMRI data of the progression of BOLD levels through training [11].

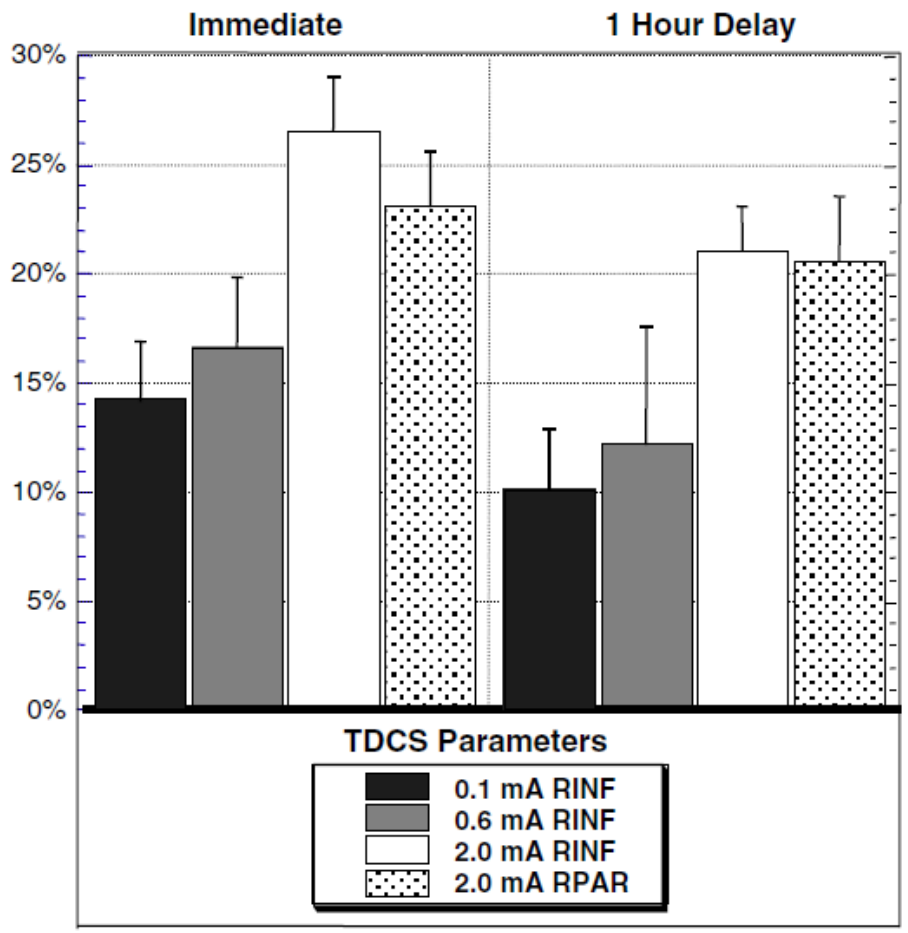

Figure 3.3. Accuracy immediately and an hour after stimulation [11].

As stated earlier, Alonzo et al. (2012) found that over a five day period, subjects who received tDCS daily, as opposed to twice daily, had greater neuronal excitability [10]. By using surface electromyography (EMG), it was possible to measure the motor evoked potentials (MEPs) from the first dorsal interosseus (FDI). The readings were taken with respect to the baseline measured during the trials. The data set is shown in Figure 3.4.

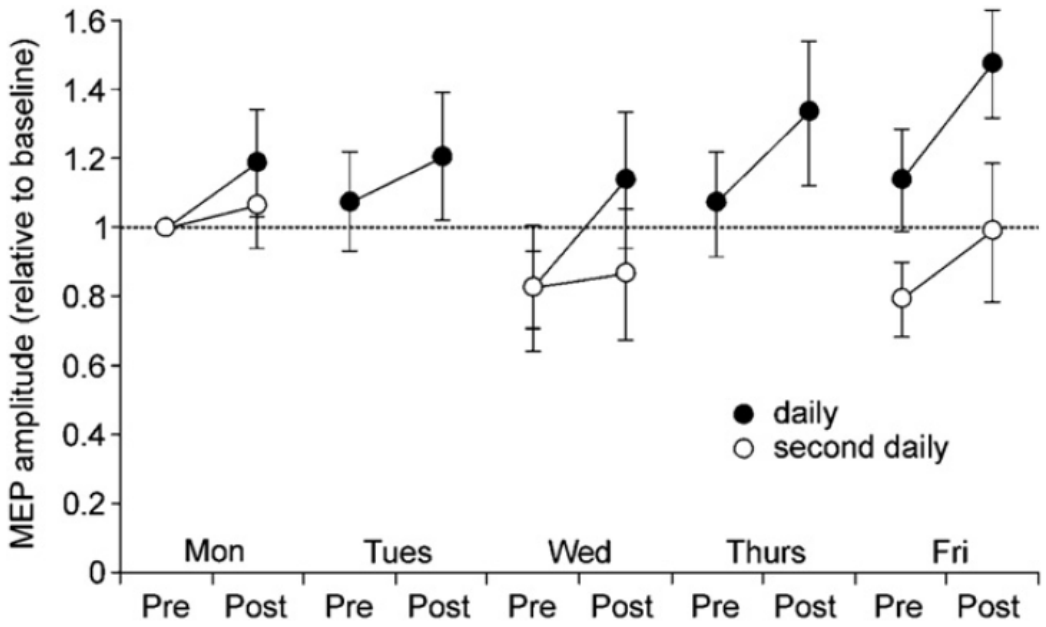

Figure 3.4. Comparison of daily with twice daily [10].

3.2.5 Methods of Software Analysis for tDCS

With tDCS becoming more popular, further in-depth analysis of what happens in the brain during stimulation is needed. Computational simulations allow the independent determination of the most likely patterns of current flow and current density during tDCS. Sadleir et. al. (2010) used a finite element model to depict several individuals. Models of entire heads were created which included data for the conductivity of bone, scalp, blood, CSF, muscle, white matter, grey matter, sclera, fat, and cartilage. Sadleir et. al. (2010) observed current densities beneath the electrodes were large in tissues such as skin, muscle, and CSF. In lower conductivity areas, such as fat and bone, the current densities were lower even near the electrodes [12].

In 2009, Hyun Sang used 3-D high resolution finite element analysis (FEA) to gain a precise analysis of the electromagnetic effect of tDCS [13]. A realistic head model was

developed with anisotropy of the white matter and the skull. Their results show that the skull anisotropy, tissue anisotropy, and white matter affect current flow. The skull anisotropy induces a strong shunting effect, which causes a shift in the stimulated area. The white matter strongly affects the current flow, which changes the current field distribution inside the brain. Tissue anisotropy significantly affects how the brain is stimulated in its deeper areas. It is also important to note that the brain is made up of three sub-regions, each with its own conductivity. These sections include one soft bone layer which is surrounded by two hard bone layers. Only one compartment was used in this study. Figure 3.5 depicts the streamline of the electric field [13].

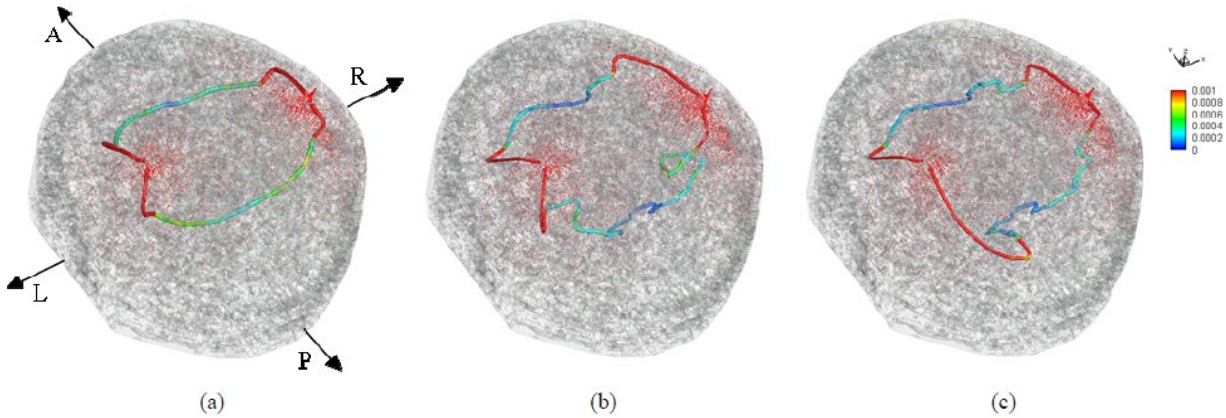

Figure 3.5. Streamline of the electric field from the anode to the cathode (A is Anterior, P is posterior, R is Right, L is left). (a) Represents the isotropic head model, (b) is the anisotropic head model with a fixed anisotropic ratio of 1:10 and (c) is the anisotropic head model with a variable anisotropic ratio [13].

Chang-Hwan Im et. al (2009) also modeled the conductivity and the electric field of a head using 3D finite element method (FEM). By using the structural metrics from MRI data, a finite element model was extracted. The scalp surface was roughly approximated as a sphere, allowing the use of spherical coordinates. By optimizing the angles and locations of the

electrodes based on the data, Chang-Hwan was able to create a model showing optimized electrode locations for a given target area deep in the brain, as shown in Figure 3.6 [14].

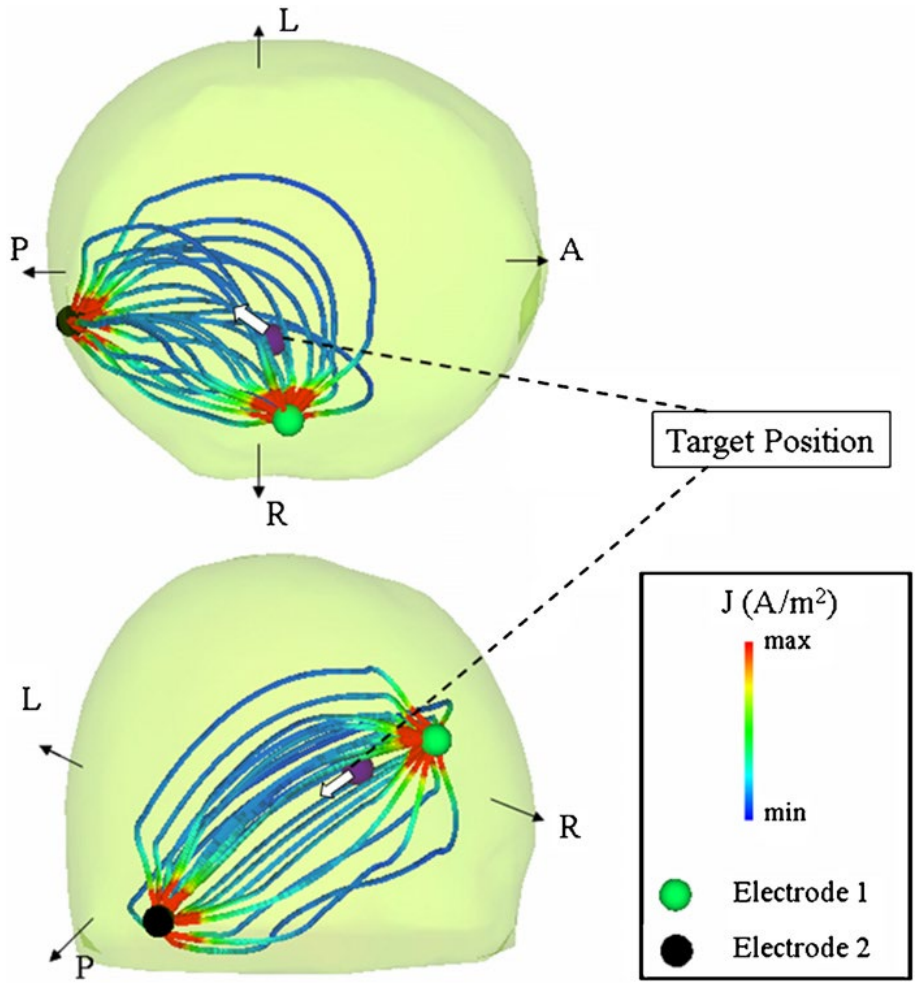

Figure 3.6. Optimized electrode locations based on target area (located deeper in the brain), with multiple currents streamlined from electrode one (A is Anterior, P is posterior, R is Right, L is left) [14].

3.2.6 Fuzzy Set Theory

Fuzzy sets are sets whose elements have a degree of membership between zero and one as opposed to the traditional principle of bivalence with classical set theory [15]. These membership values are described by a continuous membership function defined for all elements in the set, as shown in Figure 3.7. This means that fuzzy sets generalize classical sets given that the "crisp" values of zero and one are a special subset of the membership function.

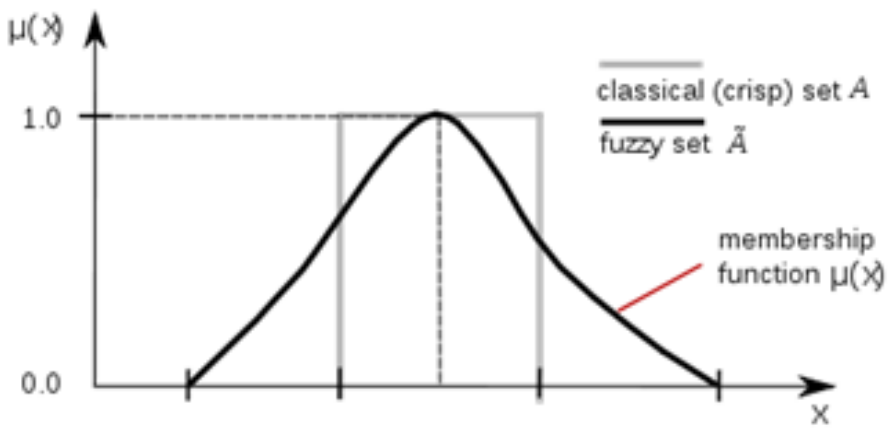

Figure 3.7. Example of a membership function and its classical equivalent [16].

A fuzzy control system is based on this ambiguous logic. It utilizes logical or linguistic variables with continuous values between zero and one to analyze analog inputs [17]. This is advantageous in expressing systemic "partial truths", or degrees of truth, in a set. This allows for the development of experience-based controller design in terms that are more accessible to human understanding [18]. Therefore, every input variable in a fuzzy control system is mapped by a set(s) of membership functions that define how well the element fits within the set. Often times, crisp input values need to be converted into fuzzy values, a process known as "fuzzification." For example, rather than defining a room as "hot" or "cold", the temperature can

be made to be “too-cold”, “cold”, “warm”, “hot”, and “too-hot” as defined in Figure 3.8. This figure represents the “domain of discourse”, which are the truth value curves that define the input range [19].

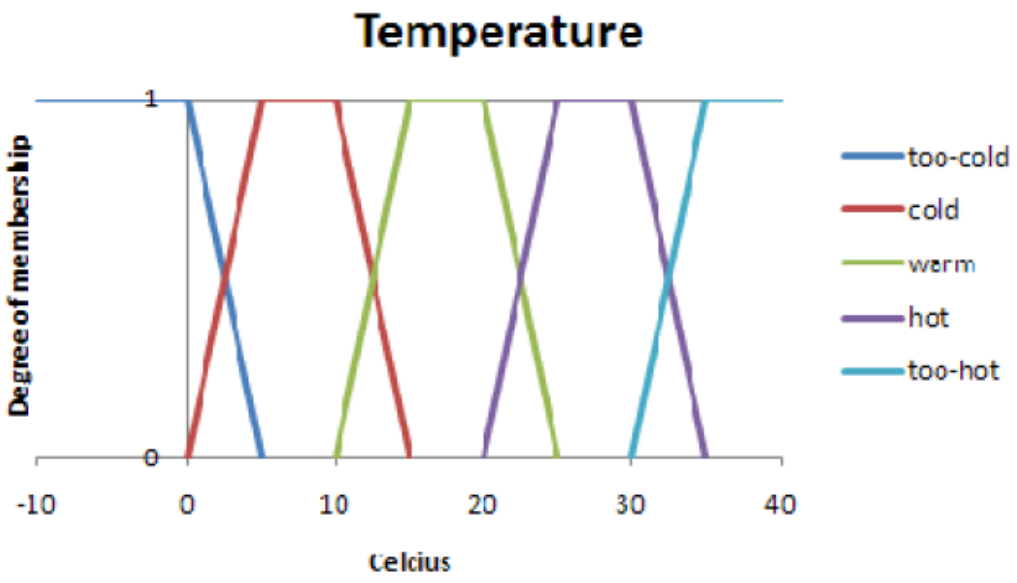

Figure 3.8. Example of temperature represented as a fuzzy set with degrees of membership [20].

Upon mapping the input values variables into membership functions and truth values, the appropriate actions can be determined by a set of rules. An example of a rule for an air conditioning unit with controllable temperature would be “**IF** it is cold inside **AND** too-cold outside, **THEN** heat flow is set to hot”. This approach defines how a controller functions, as many controllers are often made of dozens of **IF-THEN** statements. For combining elements, fuzzy operators such as **AND**, **OR**, and **NOT,** among, others exist to define outputs. There are many implementations of these operators, with the most common methods being to utilize the minimum weighted truth (**AND**), the maximum weighted truth (**OR**), and the compliment of the function (**NOT**) of an element in defining the output [17]. These operators are more formally defined as:

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \max\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\}, x \in X \tag{3.1}$$

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \min\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\}, x \in X \tag{3.2}$$

$$\mu_{\widetilde{A^c}}(x) = 1 - \mu_{\tilde{A}}(x), x \in X \tag{3.3}$$

where $\mu_{\tilde{A}}$ and $\mu_{\tilde{B}}$ are membership functions representing a value x within a range X. As many rules will generate overlapping outputs (the same output with different degrees of truth), the output membership values are generally given the truth generated by the premise based upon a "max-min" inference

$$\tilde{B} = \{(y, \mu_{\tilde{B}}(y)) | y = f(x_1, \dots, x_r), (x_1, \dots, x_r) \in X \tag{3.4}$$

$$\mu_{\tilde{B}}(y) = \begin{cases} \sup \min\limits_{(x_1, \dots, x_r) \in f^{-1}(y)} (\mu_{\widetilde{A_1}}(x_1), \dots, \mu_{\widetilde{A_r}}(x_r)) & if\ f^{-1}(y) \neq 0 \\ 0 & otherwise \end{cases} \tag{3.5}$$

Where $\tilde{B}$ is a fuzzy set in Y, which mapped from a universe X by f. This is referred to as the extension principle [21]. While the method for combing elements is based upon user preference, it must maintain the principle of having the maximum value of all of the minimum values that define an element/combination. The notable aspect here is that not only are the inputs defined by fuzzy sets, the output is also able to be defined as a fuzzy set [22].

For this reason, the output must be defuzzified to generate a crisp output for use in a system [23]. Similar to the fuzzy operators, there are several means to defuzzify the output with advantages and drawbacks to each. The general method applied is a centroid approach, which favors the output value of greatest area [24]. As shown is Figure 3.9, upon finding the maximum truth for each membership function, the "center of mass" is found that combines the resulting areas into one crisp number for actual application.

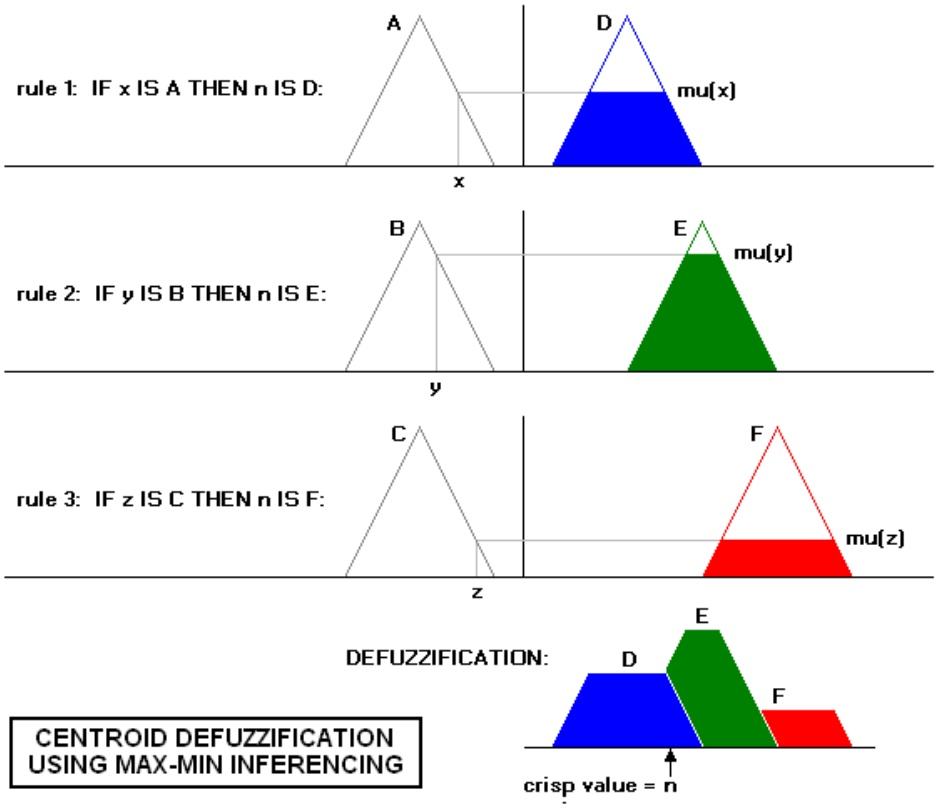

Figure 3.9. Centroid defuzzification using max-min inferencing [25].

The flexibility of input-output relations afforded by fuzzy sets allows for inference based on a set of rules designed on expert knowledge in practice. This permits the development of systems with built-in intelligence and can be utilized as an addition to existing controllers to account for factors outside of the control of the user. The most useful aspect is however, its ability to work in place where mathematical models either do not exist or are computationally too expensive to utilize. This allows for an empirical rule base design and is the basis for many effective control algorithms [22, 26].

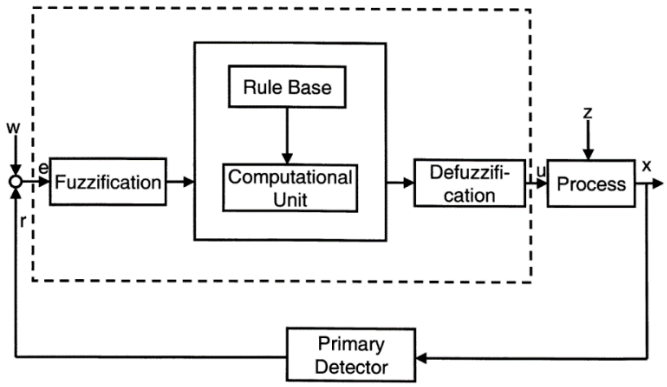

Figure 3.10. Basic fuzzy feedback controller [17].

One of those control algorithms is the Mamdani controller; a basic controller is shown in Figure 3.10. A type of feedback controller, it represents one of the first implementations of fuzzy set theory to control theory [17]. It works much in the way previously discussed, i.e, fuzzification, rule base application, and defuzzification for use [22]. The unique part of a Mamdani controller is its rule base. It works by determining the input's degree of membership in the rule-antecedent, computing the rule consequences, and then totaling the rule consequences to determine the fuzzy set "control action". The degree of membership is based upon the chosen operator; for example, an **AND** operator would result in values based upon

$$\alpha_r = min_{i=1,\dots,n}\{\mu_i{}^{ji}\left(x_i{}^{input}\right)\} \tag{3.6}$$

where α_r is the degree of rule r for all linguistic variables i and the terms of the linguistic variables j. Following this, the rule consequences can be determined by clipping the consequence to the height of the degree of membership determined in equation (6) via

$$\mu_r{}^{conseq}(u) = min\{\alpha_r, \mu^j(u)\} \tag{3.7}$$

where $\mu_r{}^{conseq}(u)$ is the rule consequence based on a given membership function $\mu^j(u)$ which represents the term of the control variable. The resulting fuzzy set of a control action is then represented by:

$$\mu^{conseq}(u) = max_r\{\mu_r{}^{conseq}(u)\} \tag{3.8}$$

before it is set to be defuzzified by one of the aforementioned methods to be utilized in a feedback loop.

3.2.6.1 Fuzzy Set as Classifiers and Optimizers to tackle the Inverse Problem

The ambiguity of where the excitation is manifesting within the brain is a major issue in the inverse problem. Since the excitation could be created by various neurons, it can be difficult to determine exactly which ones are responsible for the active electrodes. The ambiguity involved in the inverse problem needs to be addressed. One method that has been used to account for ambiguity is fuzzy set theory. In recent years, fuzzy logic applications and methods have grown vastly.

Fuzzy techniques allow the classification of imprecise data. Nava (1998) was able to classify the imprecise inputs to a speech recognition system [27]. Almulla et. al. (2015) devised a way to use a fuzzy hybrid technique to rank real world web services, by combining a Fuzzy Distance Correlation Ranking Technique (FDCRT) and a subjective Fuzzy Interval-based Ranking Technique (FSIRT). The objective data came directly from available data, while the subjective data came from domain experts. By using the linear combination of the two techniques, the system was able to overcome any shortcomings of objective and subjective techniques [28].

Fuzzy set theory has been widely used in decision making processes. Variables can be expressed qualitatively and quantitatively within fuzzy set theory. Fuzzy set theory allows the fuzziness contained in human language, judgement, evaluation and decisions to still be used as data. Rather than use a crisp yes (1) or no (0), membership values can be expressed over the interval which [0,1] reflects a degree of membership. Zyoud (2016) developed a system to

manage water loss in developing countries by integrating fuzzy set theory with Analytic Hierarchy Process (AHP) and Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). The weights and evaluation criteria were created using fuzzy AHP. These were then combined with Fuzzy TOPSIS, which then ranks the options based on how well they meet the overall objective decided by the evaluations and decision makers [29].

Parameshwaran et. al. (2015) used fuzzy set theory to determine the optimal selection of robots based on objective and subjective criteria. This was achieved by utilizing the Fuzzy Delphi Method (FDM), the Fuzzy Analytical Hierarchical Process (FAHP), the Fuzzy modified TOPSIS, and the Fuzzy VIKOR and Brown-Gibson model for the selection of robots [30]. Marsala et. al. (2015) used fuzzy data mining to develop virtual emotions and recognize natural emotions. Data mining involves pre-processing, learning, analyses, selection, etc., and is often thought of alongside machine learning. The general behavior associated with the emotions can be extracted from digital images, videos or sounds and even man-machine interactions [31]. Karsak (2015) used a multi-criteria group decision making (MCDM) approach to evaluate suppliers for businesses. With this method, Karsak was able to manage non-homogeneous information from multiple information sources and proceed to rank them based on supplier selection criteria and the ratings of suppliers [32].

Kaur and Rachana (2016) optimized vendors of materials and components based on multiple criteria. The objectives were to minimize the net price, maximize the quality, and maximize the on-time deliveries. By utilizing intuitionistic fuzzy optimization (IFO), the uncertainty and the additional degrees of freedom were converted into a crisp linear form that could be solved with an optimizing software, Tora [33].

Pourrahmani et. al. (2015) designed an algorithm to assist in evacuation routes for public vehicles. The algorithm utilizes fuzzy numbers to determine the demand at each pick up point, based on the number of evacuees. Vehicles have to travel further if they lack capacity to hold all of the evacuees. The algorithm detects failures by comparing the number of evacuees at each point and compares it to vehicle capacity. With this information, travel times can be determined for each case, and the optimal parameters are then selected [34].

Wang and Chen (2017) were able to bring multiple sources of data together using a multi-view fuzzy data clustering. Each source of data can be given different weights. By taking the information from multiple sources, multi-view data clustering is able to categorize a data set. With the availability of multiple sources, multi-view fuzzy data clustering would expedite the process of accounting for multiple sources at once. In addition, MinimaxFCM uses a minimax optimization allowing the maximum disagreements of different weighted views to be minimized. In this way, the weights of each can be learned automatically during the clustering process [35].

In the stock market, there is high uncertainty when it comes to investments. The securities and returns can rarely be described statistically. This becomes an issue when investment portfolios need to be optimized to the investor. Nikulin et. al. (2016) developed fuzzy portfolio optimization models based on Markowitz mean-variance approach. Using fuzzy numbers, Nikulin was able to extend the range of the portfolio effectiveness function [36].

Saborido et. al. (2015) also used a fuzzy technique that allowed investors to input their requirements into their own algorithms to set up a personal portfolio selection. The model is called Mean-Downside Risk-Skewness (MDRS). MDRS takes into account the multidimensional nature of portfolio selection, while accommodating the requirements of each unique investor, based on his or her requirements. The model optimizes expected return, downside-risk, and the

skewness of a given portfolio, all while considering account budget, bound and cardinality constraints [37].

In regard to the inverses problem, fuzzy sets' ability to handle ambiguity allows a more streamline approach to analyze data gathered from sLORETA. During the analysis of the generators in the brain, there is a spread of multiple regions used throughout the session. Fuzzy set theory can use this spread to analyze the frequency and the likelihood of each region being related to the task at hand. This can identify the most used region, as well as other related area of the brain related to the task in question.

With these regions mapped out, tDCS can be used to better target the generators employed for the task. Fuzzy logic can then be useful to determine which areas to target and which configuration best fits each individual going through tDCS stimulation. A fuzzy set controller can thus be designed to take in multiple conditions to be considered, including seemingly abstract details, which could account for initially unseen connections between variables and outputs.

3.2.7 Literature Review Summary

There has been a recent increase in tDCS research. Its vast range of potential uses allows tDCS to be used for medical applications as well as task-oriented situations. While many studies have shown the benefits of tDCS, there are very few reported studies that explore how the initial generating neurons are affected. By utilizing EEG data, and solving the inverse problem, it is possible to determine where the surface data originated from. This would show if the areas targeted by tDCS stimulation were indeed stimulated, as intended. A postulate here is that through fuzzy sets, it is conceivable to analyze multiple subjects and discern if tDCS was applied, as well as where the maximum stimulation occurred for each subject. Moreover, other

key factors can be determined as well, such as how long the stimulation lasted and how much it changed over time. In this way, a better understanding of control and tDCS subjects can be developed. This research work focuses on using that understanding to aid in tDCS administration. By doing so, tDCS can better enhance the brain regions determined to be most utilized during the task at hand. Since the methodology is set up specific rules to analyze the data and determine conclusions, it can be applied to any task.

3.3 Methodology Overview

3.3.1 Research Scope and Objectives

With tDCS gaining more popularity in recent years, it is important to understand how exactly it affects the subject's brain. Tools like EEG, TMS, and MRI have been used through the years. Each has its advantages and disadvantages, as discussed earlier. By using EEG, one can observe the activity on the surface of the scalp, but not where that activity comes from within the brain. With MRI, it is possible to observe the inner workings of the brain, but the technology limits movement and most machines require one to be stationary inside a tube. The solution to these problems involves developing a series of models that show where the generators in the brain activate on the scalp. This is the forward solution, i.e. taking into account how the activity flows normally. Through data from multiple studies, Standardized Low Resolution Brain Electromagnetic Tomography (sLORETA) can be used to solve the inverse problem. With the source localization built into sLORETA, the activity on the surface of the scalp allows sLORETA to determine where the activity within the brain originated [38].

tDCS has been applied to different neuropsychiatric diseases and disorders [39]. tDCS has also been used to improve focus and performance in cognitive activities. By fully understanding how tDCS affects deep into the brain, it can be better utilized to target the

required regions to treat or enhance them. This research project focuses on localizing the sources of brain activity in tDCS and control group. This will allow assessment of tDCS on a subject's brain during a traditional tDCS session. It also aims at determining the duration of the lasting effects after tDCS stimulation is over, as well as the areas of the brain that received the most stimulation. This information can then be used to better understand how tDCS is affecting the brain, and then optimize future tDCS session utilizing that understanding.

Thus, the specific objectives of this section of the research were, in order of execution:

- To develop a filtering scheme to remove noise from EEG/tDCS data
- To replace any bad channels with interpolated data from nearby channels
- To devise a model to tackle the issue of representing the data in sLORETA format
- To solve the inverse problem through source localization with sLORETA to determine which Brodmann area initiated the excitation on the surface of the scalp, along with the lasting duration after and magnitude over time
- To develop a fuzzy controller to process and analyze data as it pertains to tDCS
- To determine information such as if tDCS was applied, where it was strongest, the trends in activity over time, and the duration of the stimulation
- To instill modularity in the methodology to allow for other studies to utilize the model with different EEG data

3.3.2 Research Methods

3.3.2.1 Field Data

The conducted study used tDCS data collected during a previous research investigation. From 2015 to 2017, Drs. Rick Houser and Daniel Fonseca collaborated on the research study, *The use of Mobile EEG and Stationary EEG Amplifiers in a Pilot Study Focused on the Impact*

of Low Current Brain Stimulation on Math Understanding and Calculations. The funding was acquired through a University of Alabama's RGC3 grant. Data was collected using a 64-channel mobile EEG amplifier from tDCS subjects. During the study, undergraduate math students were monitored. They were divided into two groups of individuals, one of which was the control group and the other the experimental group. All the participants consisted of freshman and sophomore engineering students at The University of Alabama. The students were enrolling in a pre-calculus algebra course (i.e. Math 112). Phase one included a baseline assessment of the study group, in which basic EEG data was taken, without tDCS. After the baseline recording were taken, a video discussing intermediate algebra calculations was shown, while still taking EEG data. Two baseline assessments were made with different videos.

Low current brain stimulation was administered to the experimental group, with the anode (positive electrode) placed on P3, based on the 10-20 international system of EEG electrode placement shown in Figure 3.11. This is due to P3 being associated with calculations and reasoning. The cathode (negative electrode) was place at T4 based on the 10-20 international system. The experimental group was administered 2 mA over 20 minutes for each instance. The control group was given a sham condition in which the administration provided was 1.0 mA, lasting for 30 seconds. To replicate the same experience, the control subjects were treated as though they were given tDCs for the full 20 minutes. After watching the algebra video, subjects completed calculations based on instructions received from the video. Data collection was gathered from 64 electrode sites e.g. Fp1, Fp2, F7, F3, Pz, F4, F8, T3, C3, Cz, C4, T4, T5, P3, Pz, P4, T6 O1 & O2. The maximum sampling rate was 2048 Hz.

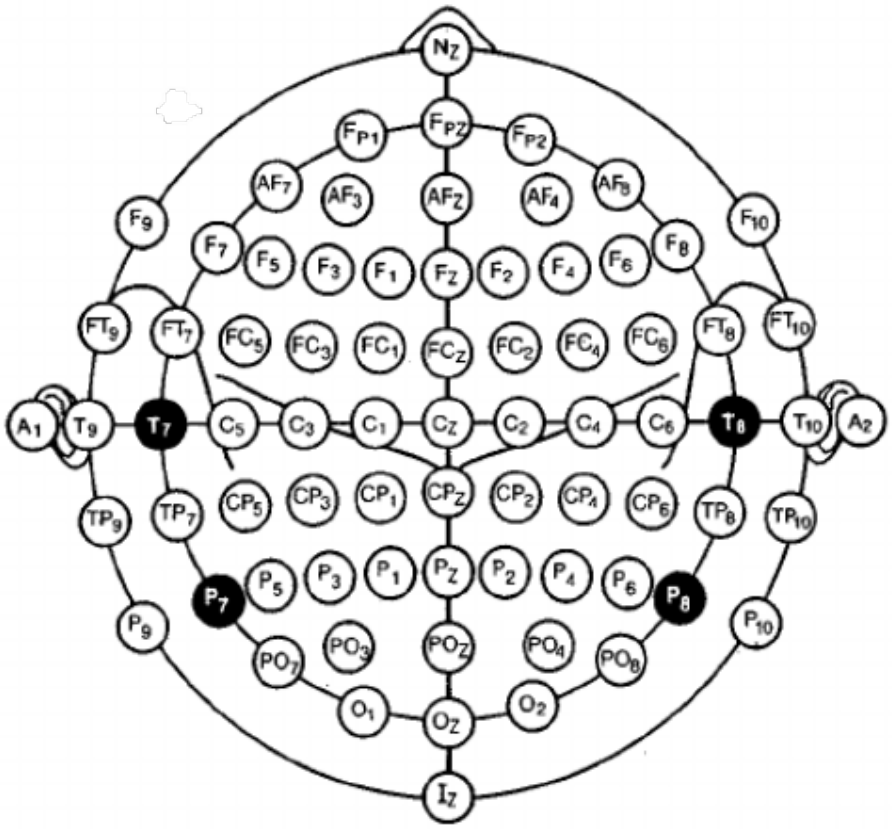

Figure 3.11. 10-20 international electrode placement [40].

3.3.2.2 Data Processing

Test data was processed through a data acquisition program called **asa**lab. The built-in tools of **asa**lab allowed for an effective filtering of the acquired data. After the filtering was done, the data needed to be converted into a format that LORETA can read in order to calculate the source localization to determine where the initial neuron that generated the surface excitation was located. By using the EEGLAB interactive MATLAB toolbox developed by Swartz Center for Computational Neuroscience, it was possible to convert the data taken with **asa**lab to LORETA. Figure 3.12 shows unfiltered data represented in EEGLAB. These high spikes, which are noise in the data, will be accounted for during the filtering process.

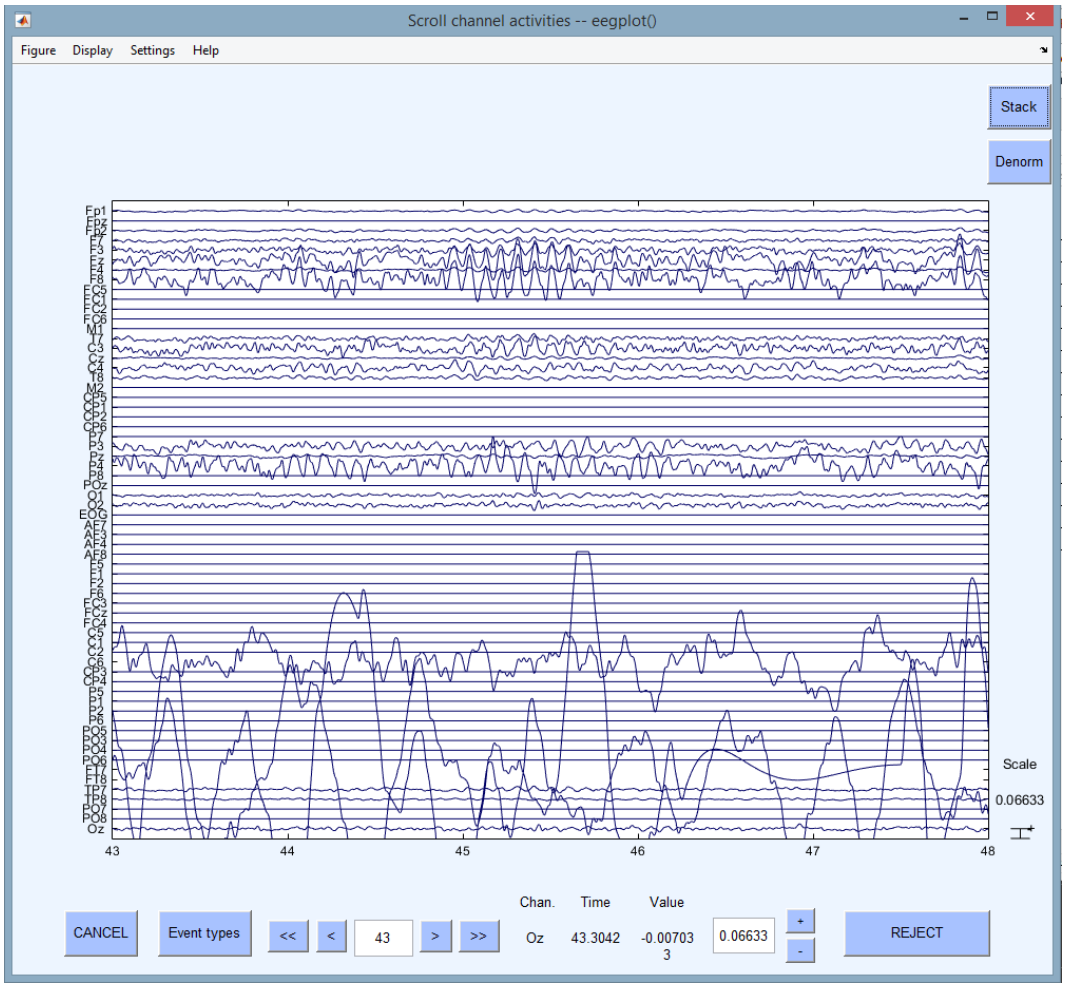

Figure 3.12. Unfiltered data represented in EEGLAB.

In order to streamline the filtering process, a MATLAB script was written. The first section of the script allows for multiple subjects and sessions to be loaded at once from a file location on the computer. The script first looks into the folder detailed in the script. It then makes a list of every type of file of the specified type. In this case, *.set* files were taken. The script proceeds to iterate through each file until every file has been processed.

The processing of the data starts with loading the data from the list of files. Once loaded, the unused channels are removed and only the specified channels requested in the script will remain. At this point, the script loads the standard locations of the remaining channels to be later used for interpolation purposes. The script runs ICA, followed by a purge of any bad channel

components. This is when the channel locations from earlier allows for interpolating back any rejected data using the location of the channels nearby. The script then sections the data off into segments based on length or parameters defined in the script. These segments are subsequently filtered before they are standardized in length to be loaded into LORETA.

3.3.3 Data Modeling

Before the collected data could be used, certain parameters, such as the electrode configuration, needed to be established within LORETA. When input into LORETA, the electrodes were still in the 10-20 international system, but sLORETA uses Talairach coordinates, which corresponded to where the electrodes are on the scalp spatially. It is possible to measure these spatial coordinates per individual, but LORETA has built in models that were averaged over many participants to compute mean coordinates relative to each channel location. LORETA then took these coordinates and calculated a transformation matrix for sLORETA or eLORETA with a chosen regularization method. The EEG files could then be processed with sLORETA using the transformation matrix. This produced viewing files that can be observed over any given time frame.

Time frames were chosen based on the checkpoints determined to better understand what occurred during the study. Based on its overall time, the session was broken up into five sections and analyzed with sLORETA. The start window showed how the brain is adapted to tDCS and the task conducted. The next section of the data made it possible to examine how the brain adapted to the task. During the third section of the data, the brain was well enthralled in the task at hand. The fourth section was used to observe how well the brain maintained focus. Finally, data from the last section was used to determine how the brain reacted as the activity is finishing, and how well the stimulation lasted. Example data from sLORETA is shown in Figure 3.13.

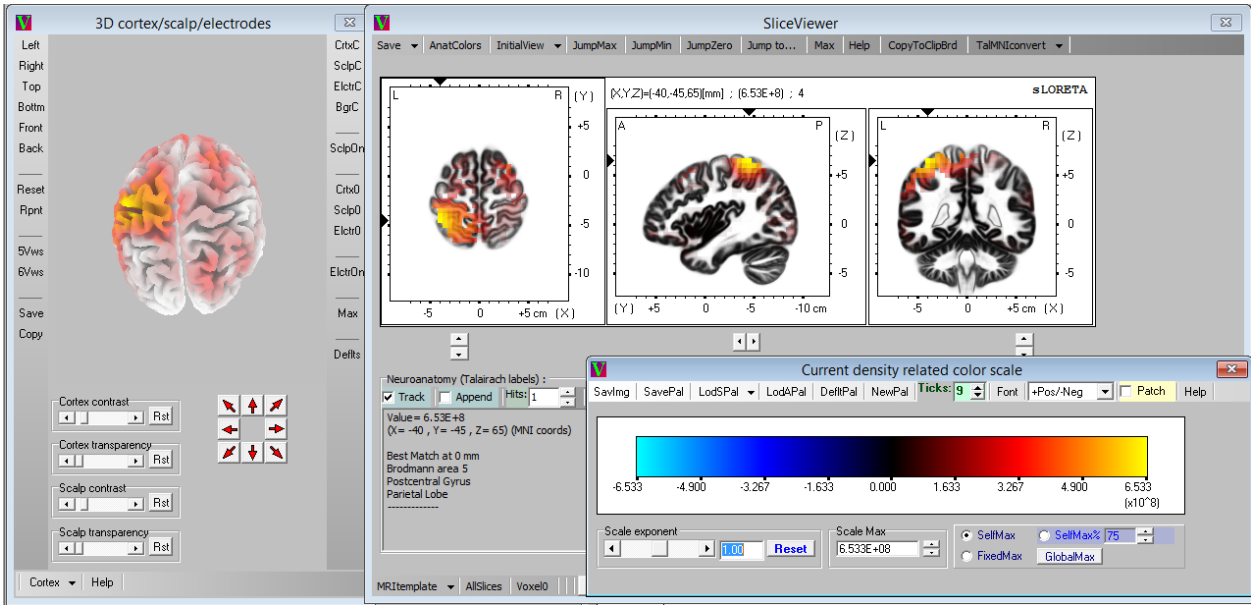

Figure 3.13. Data from sLORETA data showing cross sections of brain activation.

3.3.3.1 Experiment Design and Implementation

This study utilized EEG data along with the consultation of the tDCS administrators at The University of Alabama in order to complement the data and provide an extra understanding for the controller developed. The research study consisted of objective information, while expert observations would be considered subjective. A fuzzy hybrid system can be used to combine subjective and objective data [28, 30]. This combination allows observations that otherwise would be unseen from data alone.

Based upon the collected data, as well as correlations from the available literature, it was possible to develop a set of fuzzy inputs and outputs. Fuzzifying the results was based on the region of the brain being active, and level of activation. The rules that defined the controller were developed based on the data and programmed with MATLAB's Fuzzy Toolbox through a

Mamdani controller scheme, a centroid based method. The resulting controller was then tested at known states to compare desired stimulation and the resulting output parameters.

The fuzzy controller was used to analyze the data by correlating the levels of activation of every participant at each given time. This fuzzy controller determined trends in the data. Once trained, the controller was able to establish whether or not tDCS had an impact on the brain. It also assessed the level of activation to expect in future studies, as well as which regions were most activated during the task, along with any secondary regions that showed to share/hinder the activation of the main region of the brain. The controller was constructed in an input/output user friendly fashion. There is also the potential to incorporate additional rules, inputs, and outputs to the controller, as needed. A flowchart depicting the in-depth methodology followed in this research study is shown in Figure 3.14.

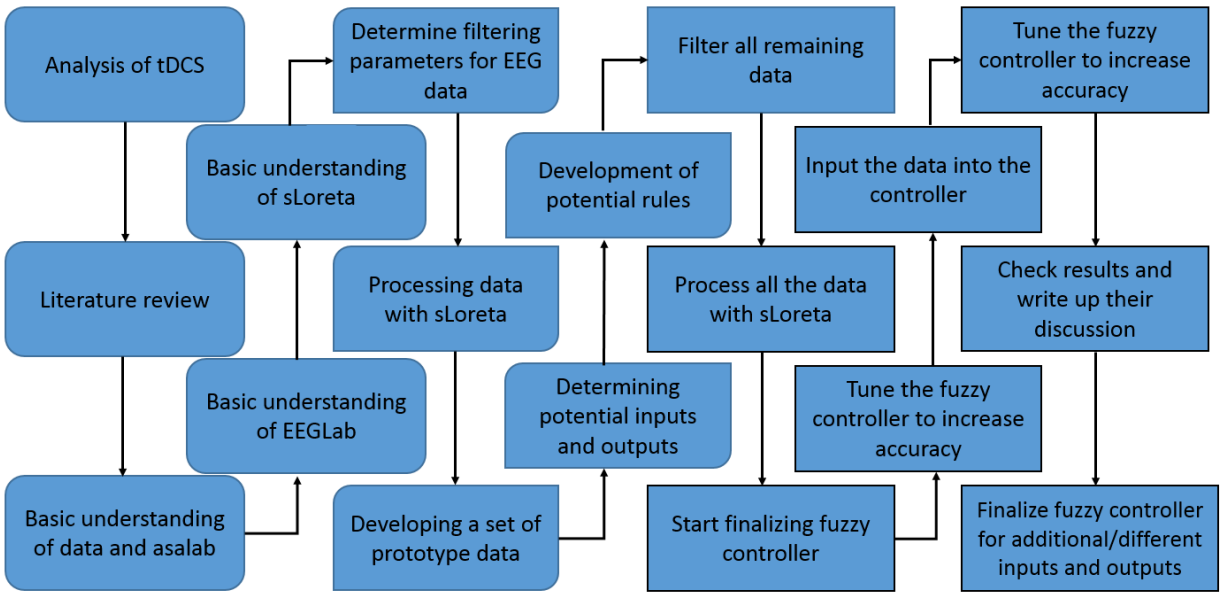

Figure 3.14. Research Project Flowchart.

3.3.4 Research Plan

The first main activity involved analyzing tDCS to understand the parameters and how it stimulates the brain. This knowledge was further enhanced by a literature review, giving a further understanding of how each of the application parameters affected tDCS. EEG data was recorded using **asa**lab, and simple noise filters were applied. The data was then exported for use in EEGLAB. At this point, a better understanding of EEGLAB was necessary to filter the data and prepare it for sLORETA. The data was then exported in a form that sLORETA could read and use.

Initially, one set of data was filtered and analyzed to determine key parameters to analyze the data e.g. filtering thresholds, segment lengths, Brodmann areas and activation. These parameters were later used to process and analyze the remaining data sets. Data from this analysis were used to generate inputs and outputs for the fuzzy controller. The rules for the input and outputs were setup to establish the functionality of the controller as it takes and processes data. Developing a fuzzy controller prototype furthered the understanding of how the EEG data should be filtered and analyzed. Afterwards, the remaining data points were filtered and processed with sLORETA.

The fuzzy controller was developed to identify trends in the activity over time, location, and duration. The controller was tuned to increase its accuracy in determining these trends. It was then enhanced to assess how best to improve stimulation on the grounds of how well each subject matched up to the rest of the study group. In the case of single individual trials, there is no comparison group, so the controller was based on the individual's previous stimulation trials. The accuracy of the controller was critiqued further with discussion from domain experts in

tDCS administration at the University of Alabama. Lastly, the controller's heuristics were fine-tuned to account for additional variables.

The next section of this document provides a more in-depth explanation of the collected data and followed approach. It details how the collected data was filtered with MATLAB. Subsequently, the analysis of the data gained from sLORETA is also explained. This will be followed by a discussion of how the controller was setup and finalized, and how the data was imported into it.

3.4 Project Methodology

3.4.1 Nature of Data Collected for this Study

From 2015 to 2017, Drs. Rick Houser and Daniel Fonseca collaborated on the research study *The use of Mobile EEG and Stationary EEG Amplifiers in a Pilot Study Focused on the Impact of Low Current Brain Stimulation on Math Understanding and Calculations*. Undergraduate students were studied using a 64-channel mobile EEG amplifier. The two groups formed consisted of one group of control subjects and the other group was the experimental tDCS subjects. The subjects were freshman and sophomore engineering students at The University of Alabama. The students were enrolled in a pre-calculus algebra course (i.e. Math 112). The first study included a baseline assessment of the study group, in which tDCS was not administered.

3.4.2 EEGLAB

3.4.2.1 ICA Filtering and **asa**lab

Data sets from the control group and the experimental group were processed using **asa**lab. The data was run through a basic band-pass filter to eliminate any power noise and high

and low amplitudes according to a pre-established threshold. The files were then loaded into MATLAB using the EEGLAB toolbox. The **asa**lab plugin was downloaded to read the **asa**lab files as they were saved originally. A summary of the dataset loaded is presented in Figure 3.15. Once the files were loaded into EEGLAB, the channel locations were mapped based on the 10-20 international system names. The International 10-20 system is used to correlate the external skull locations to the underlying cortical areas [41]. The coordinates received were based on the Brain Electrical Source Analysis (BESA) file for 4-shell dipfit spherical model. An example of a mapped channel is shown in Figure 3.16. This assigned coordinates to be used within EEGLAB for future calculations. The channels used during this study were Fp1, Fp2, F7, F3, Fz, F4, F8, T7, C3, Cz, C4, T8, P3, Pz, P4, O1, O2, TP7, TP8, and Oz. During this process, any extra channels without data were discarded.

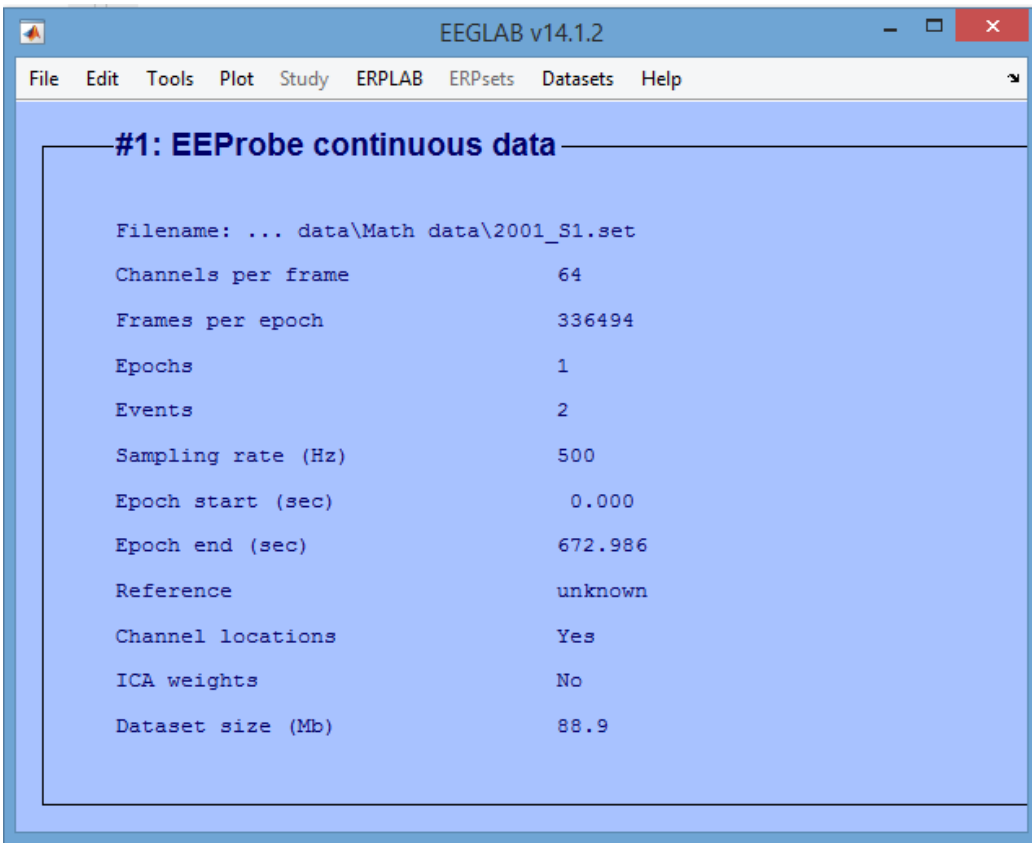

Figure 3.15. EEGLAB data summary of loaded dataset.

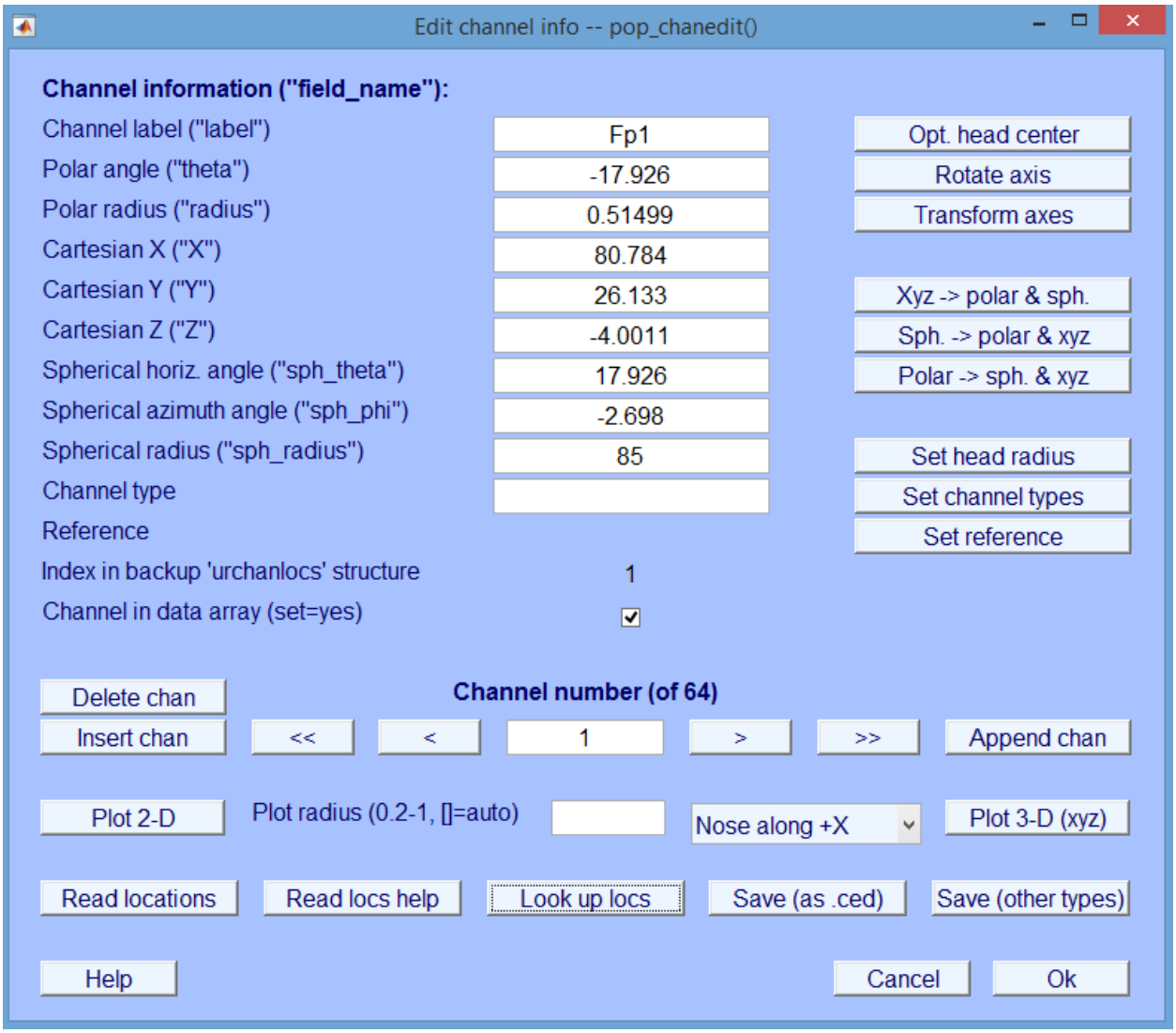

Figure 3.16. Adding in channel locations to data.

This would be followed by an independent component analysis (ICA) on the data using the runica algorithm. This allowed the data to be split into components. Each component represented a different portion of the data. Figure 3.17 shows an example of raw data while Figure 3.18 depicts an example of how that data is split into individual components to be analyzed. A 2-D and 3-D visualization of how each component is represented as shown in Figure 3.19 and Figure 3.20, respectively. In these figures, each head model is mapping out the activation area of the component assigned to it. At this point, any channel could be rejected

manually using either model for reference, but due to the number of data sets, the script was utilized. The script could then identify the components and determine which ones contained data, and which noise. The components that were mostly made up of noise could be removed. Using the mapped channels, the removed noisy channels could be replaced by interpolated data from the other channels in the data set. This is done by cross checking which channels were extracted during ICA and using the surrounding channels to piece back together a clean channel. The data is then prepared for segmenting.

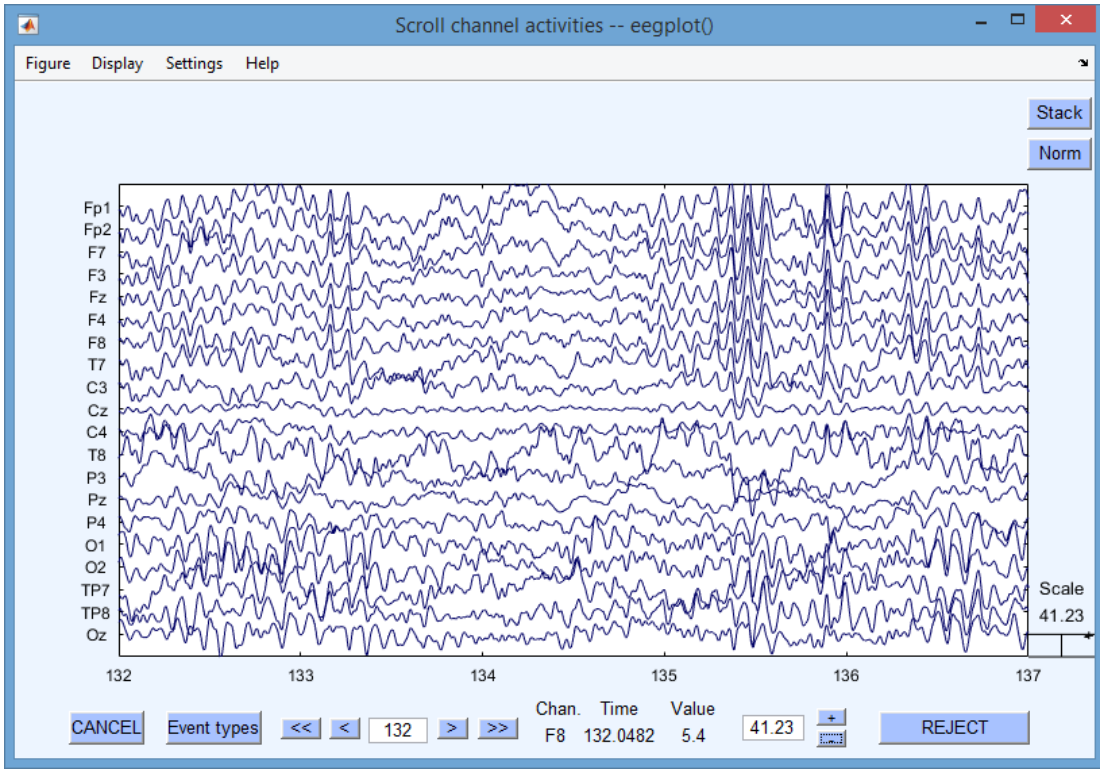

Figure 3.17. Raw channel data.

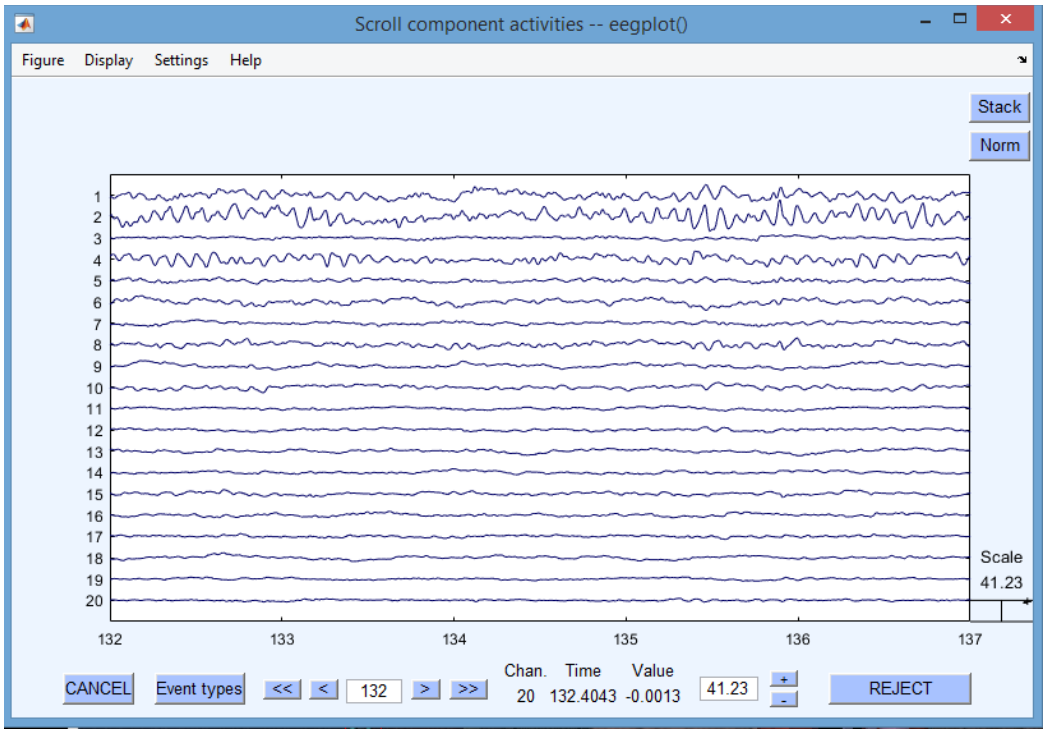

Figure 3.18. Independent component analysis of the Raw channel data.

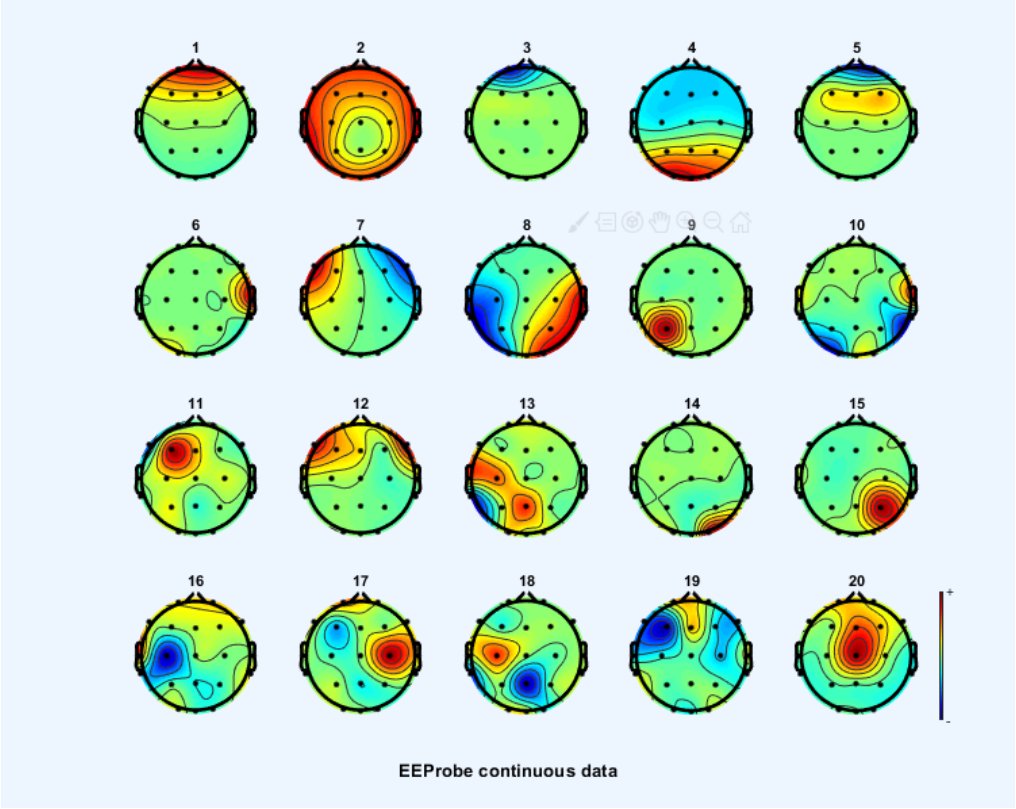

Figure 3.19. 2-D Head models showing activation of ICA channels.

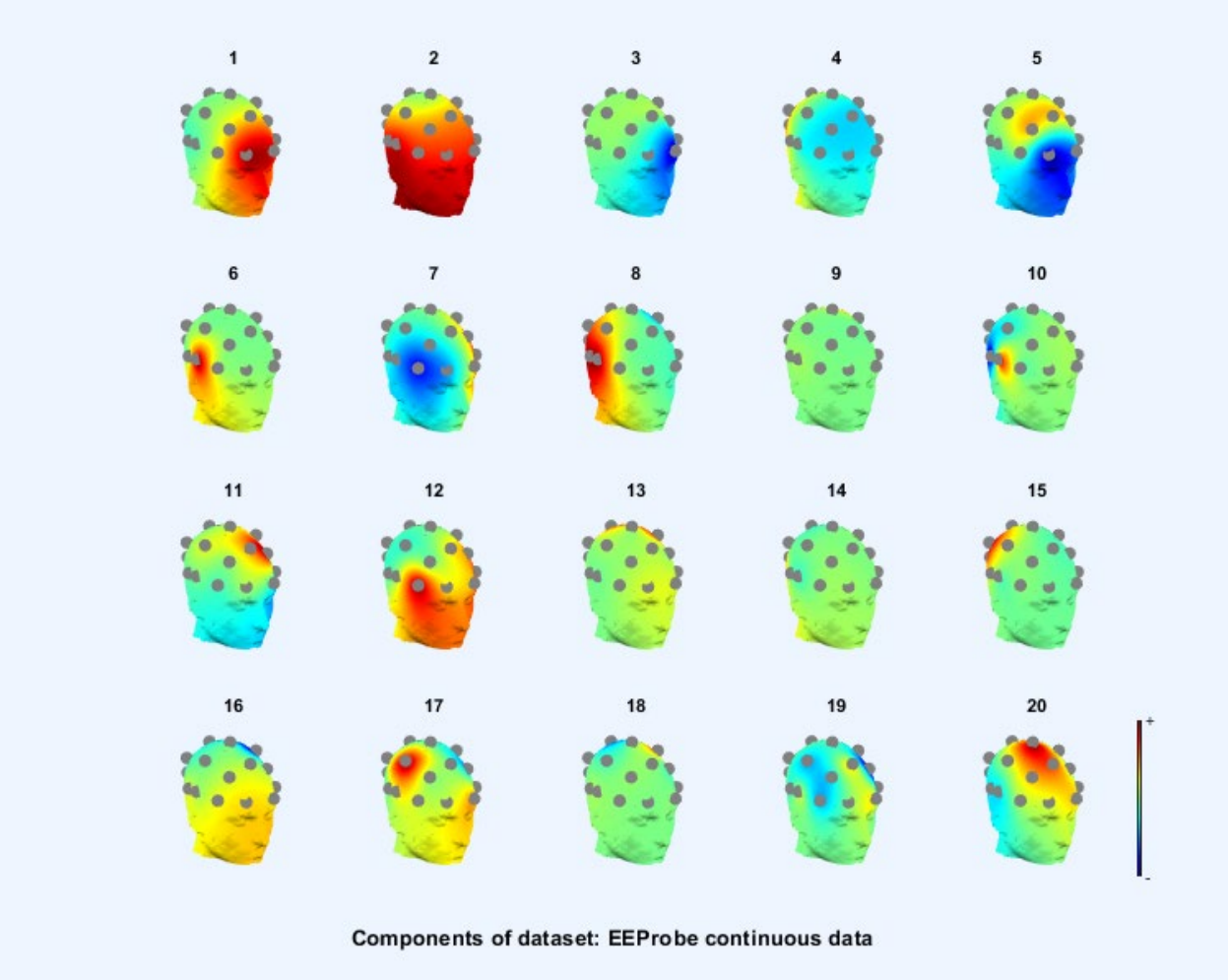

Figure 3.20. 3-D Head models showing activation of ICA channels.

3.4.2.2 Segmenting Data

Since the EEG data is to be analyzed in sections over the course of the session, it is necessary to break the sessions down into segments for ease of analyze. This begins with each session being split into segmented sections based on the overall session length. A visualization of this process is shown in Figure 3.21. For sessions longer than 30 minutes, the variable *DesiredSeg* can be used to change the size of the segmented data. A variable named *lengthcheck* can be set to 0 in order to set a static segment length for datasets of variable sizes. For data that is of variable sizes, the *MaxSegs* variable limits the number of created segments. Data sets with

segments less than the *MaxSegs* will be treated normally only taking what is available. Lastly, as part of this segmentation process, it is possible to set the size of the output segments, discussed further in the next section. This can be defined with the *outputL* variable.

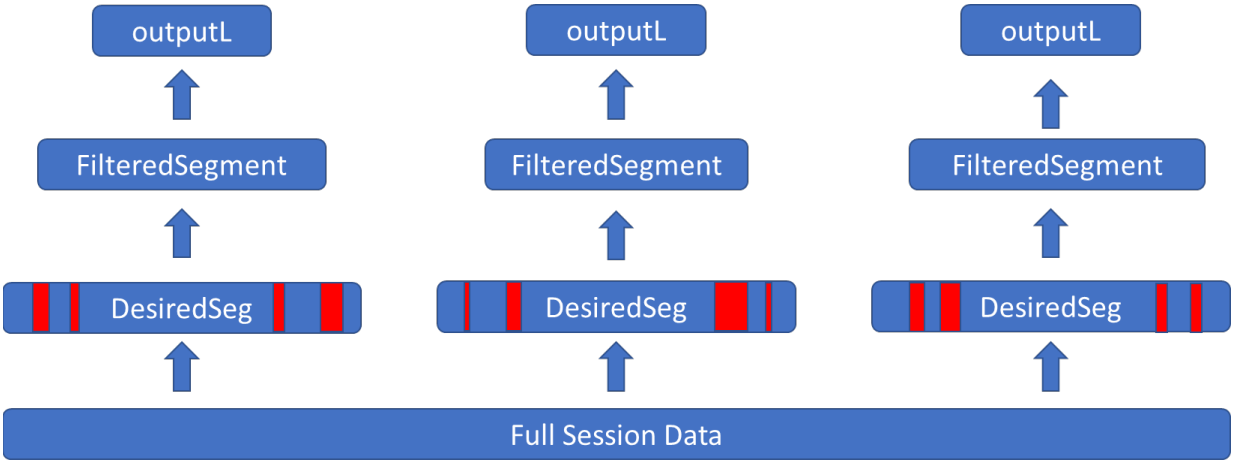

Figure 3.21. Visual representation of variables and data.

3.4.2.3 sLORETA Preparation and Exporting

Once the data is split into these sections, the following step consists of preparing it for sLORETA. It is now possible to filter the data by running automatic continuous rejection. This would cut out sections of noise that were not rejected during ICA. This filtering would account for any artifacts such as blinks, movement, and other artifacts outside normal ranges for EEG data, but doing so removes sections of data. This artifact rejection is done through the use of 'statistical' thresholding. An example of data being removed is shown in Figure 3.22. This causes each segment to be variable in lengths. Thus, in order to keep the data lenght consistent, an interval based on *outputL* is taken from the remaining data in all the sections. The baseline data is analyzed the same way. The baseline analysis will be done between 60 to 240 seconds to account for the subject getting settled into the session [42].

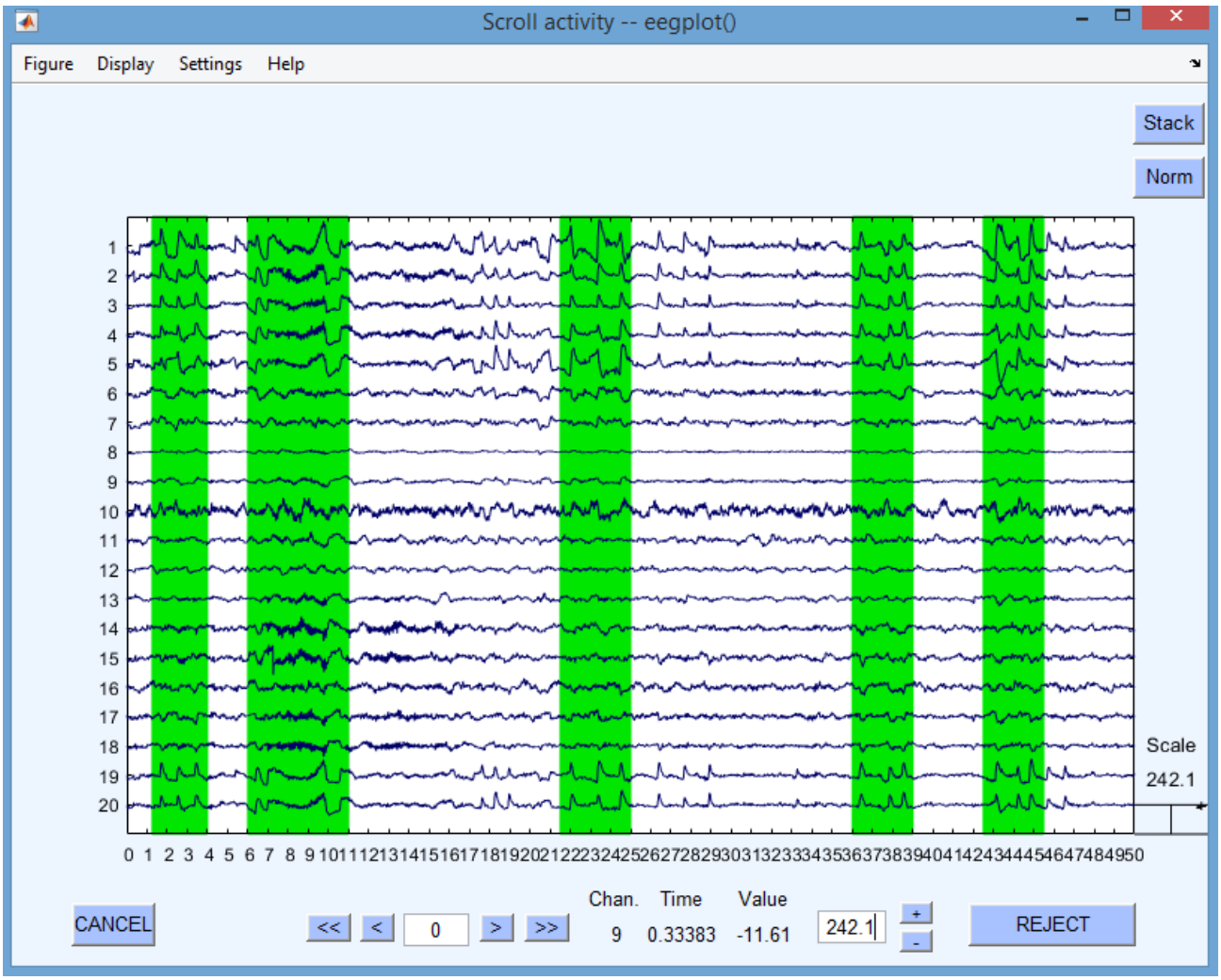

Figure 3.22. Automatic continuous rejection highlighting noisy data to be removed.

3.4.3 sLORETA

3.4.3.1 Overview of Process

The data needed to be converted into a file format that could be read by sLORETA. This was done using another EEGLAB function developed by Brock University Cognitive and Affective Neuroscience Laboratory. The specific steps following the conversion process were:

- Determining the electrode Talairach coordinates

- Calculating the electrode transformation matrix
- Running the sLORETA with a signal to noise ratio of 10 (SNR10)
- Analyzing the created viewer files to retrieve data
- Organizing the data into excel
- Uploading the data into Simulink for the Fuzzy Controller
- Labeling and identifying parameters within the data to utilize in the analysis
- Running the controller to analyze and determine which parameters were lacking.

3.4.3.2 Setting up and Running sLORETA

Once these files were ready, the sLORETA part of the data processing needed to be prepared. The first item required is the electrode Talairach coordinates, which can be processed by the "electrode names to coordinates" section in LORETA, shown in Figure 3.23. Figure 3.24 depicts the electrode maker, which is just a graphical representation of the electrodes chosen to be made into Talairach coordinates. Talairach is the best-known brain atlas; it provides a set of saggital, coronal, and axial sections labeled by anatomical structure and Brodmann's areas. Normalizing the data according to Talairach's procedure allows for a seemingly simple way to determine the anatomical location for any particular location [43]. With these coordinates, it is then possible to set up the transformation matrix with the corresponding electrode coordinates (see Figure 3.25). The output matrices are based on different regularization magnitudes. By choosing automatic, a few sets of files will be created. Of these files, the Signal to Noise (SNR10) file was chosen, which processes the data with a signal to noise ratio of 10. This ratio should have been accomplished with the aforementioned filtering.

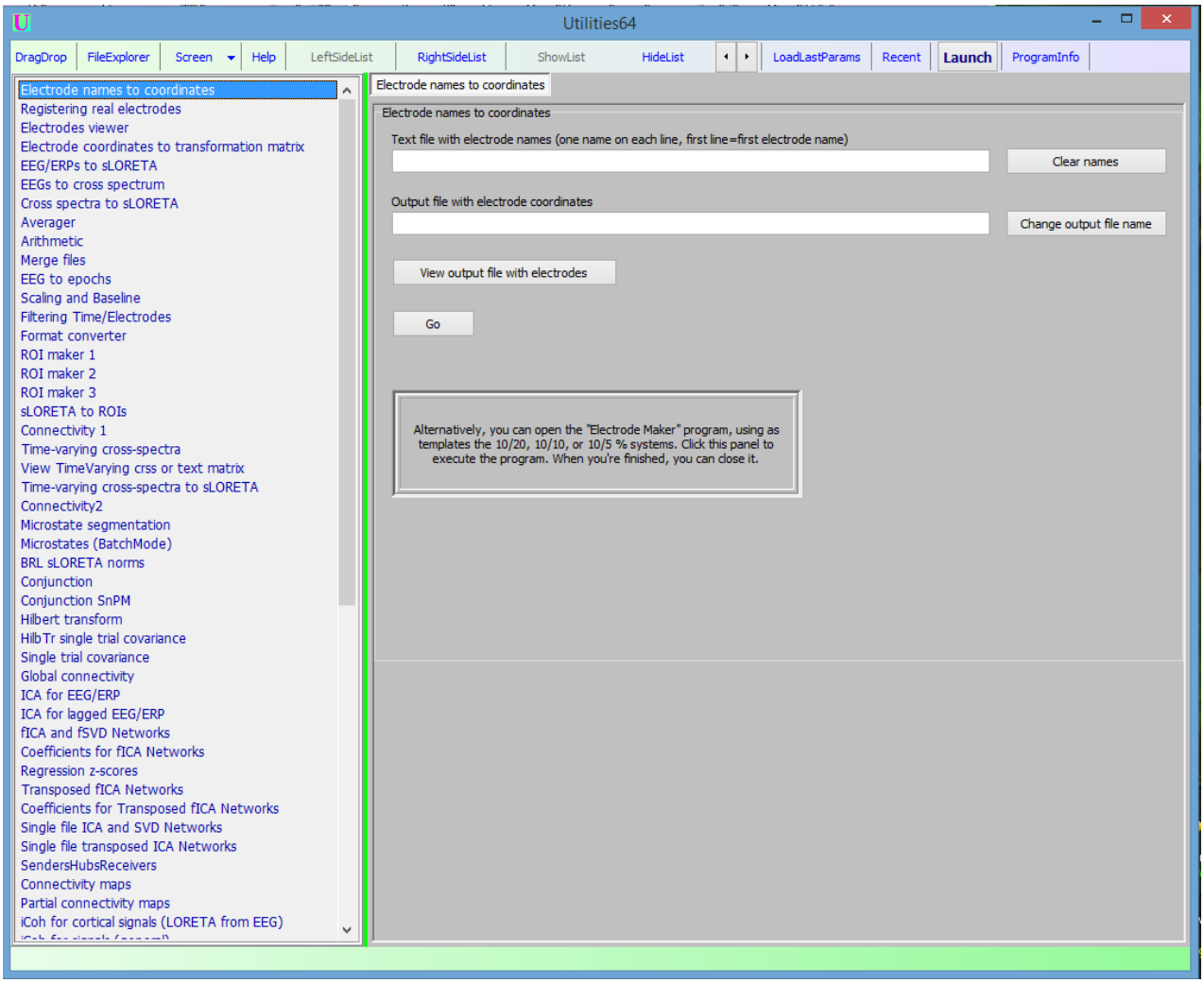

Figure 3.23. Electrode names to coordinates.

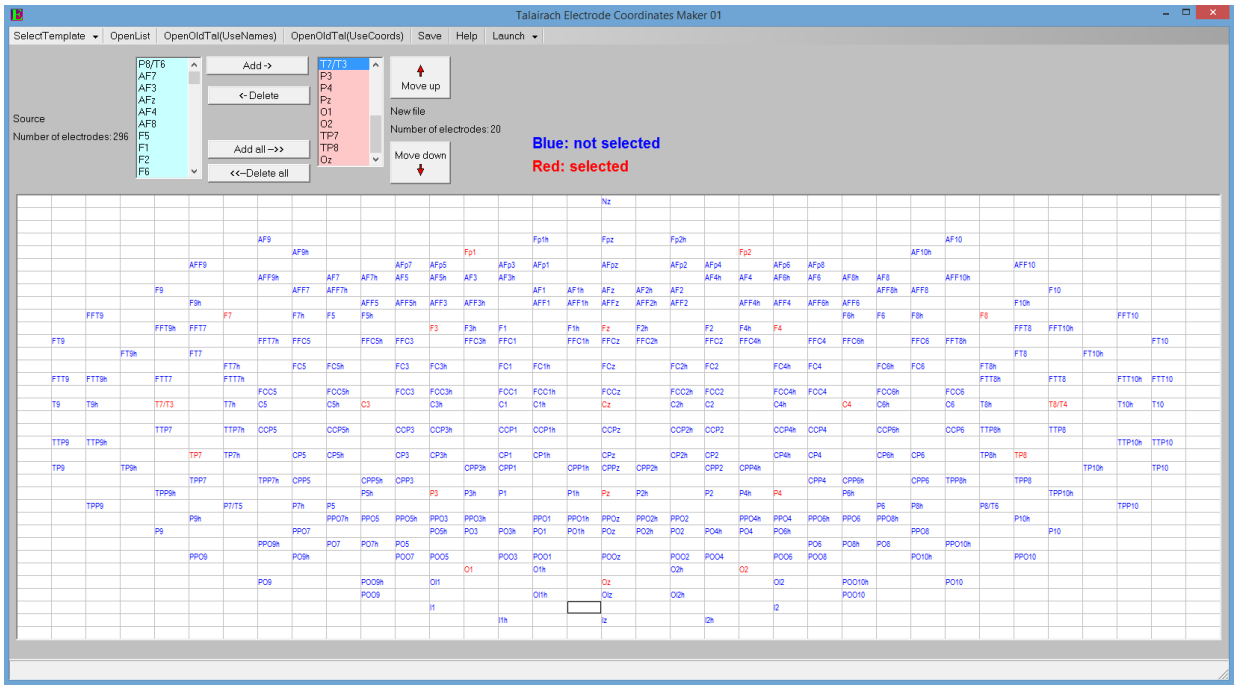

Figure 3.24. Talairach electrode coordinate maker.

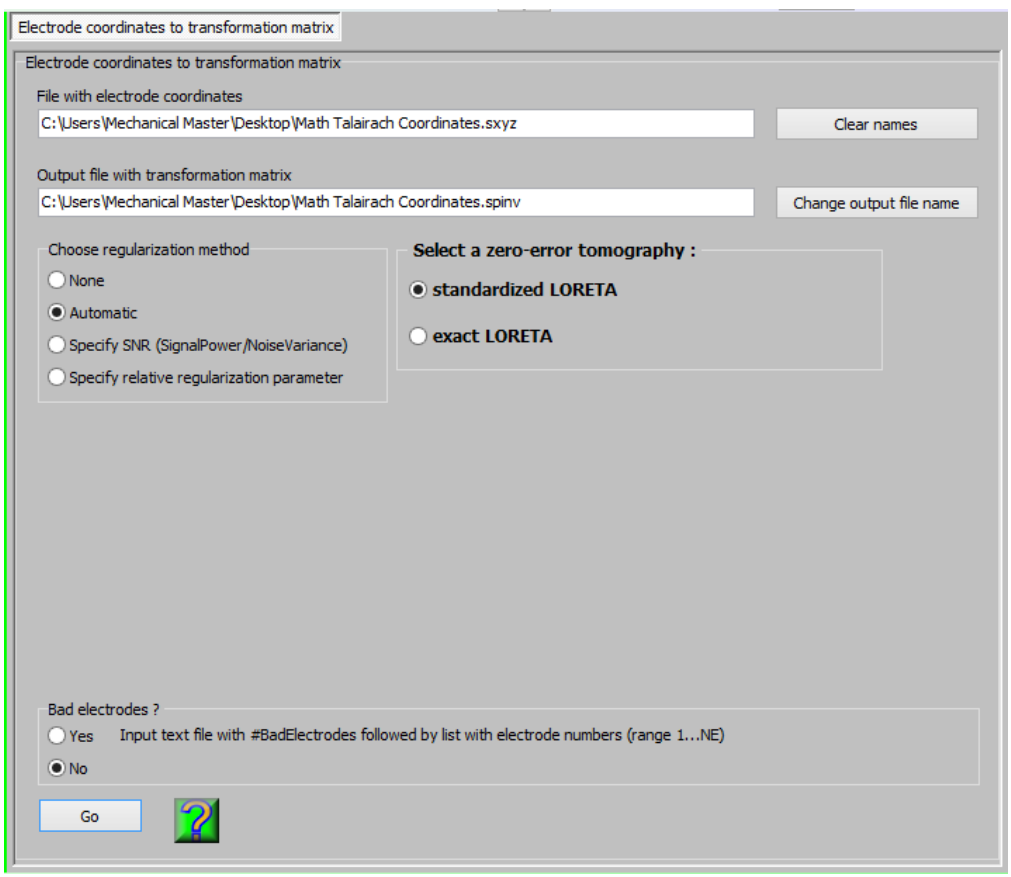

Figure 3.25. Electrode coordinates to transformation matrix.

The Viewer files were then created from the EEG/ERPs to sLORETA using the EEG/ERPs to sLORETA section shown in Figure 3.26. The SNR10 file was used as the transformation matrix for all the time frames while computing sLORETA. The output of this process was the viewer files. These files were then opened in the viewer section of LORETA. In order to see all of the data at once, the time frames per page were increased to encompass the whole part of the session. The average of the section was then computed which gave the maximum voxels, along with the Brodmann area at which it occurred. The Brodmann area will be used to determine if the tDCS subjects were using the same Brodmann areas as the control subjects. Figure 3.27 shows 2-D cross sections of the brain at the maximum activation location. Figure 3.28 shows Cross sections of the head from the bottom slowly progressing to the top. Lastly, Figure 3.29 depicts a 3-D representation of the activation in the brain.

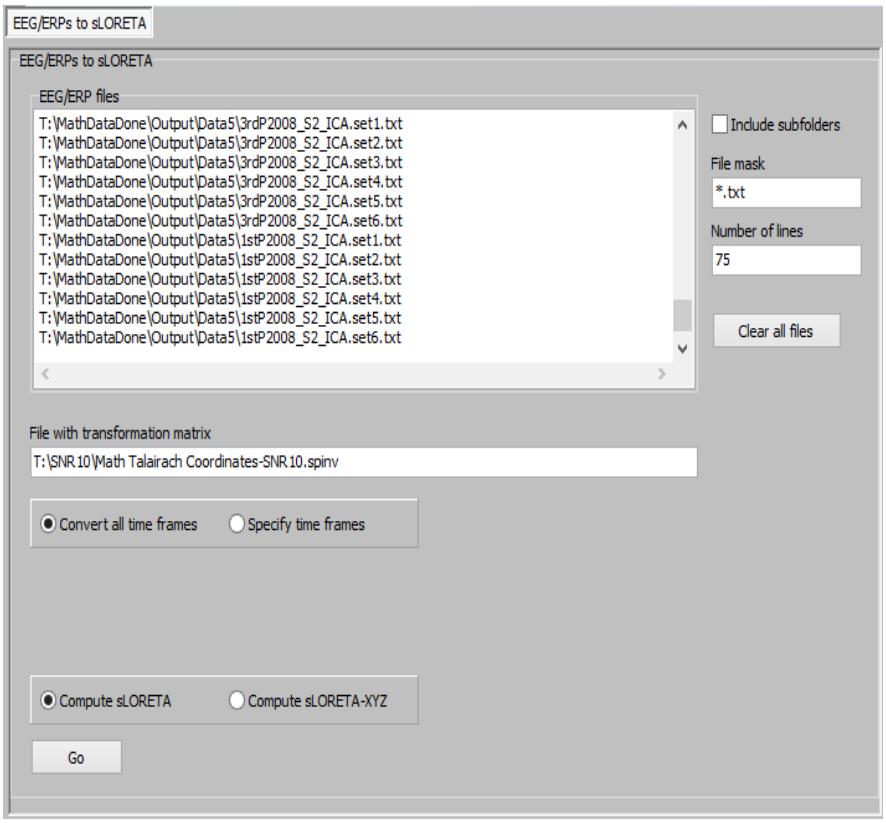

Figure 3.26. EEG/ERPs to sLORETA.

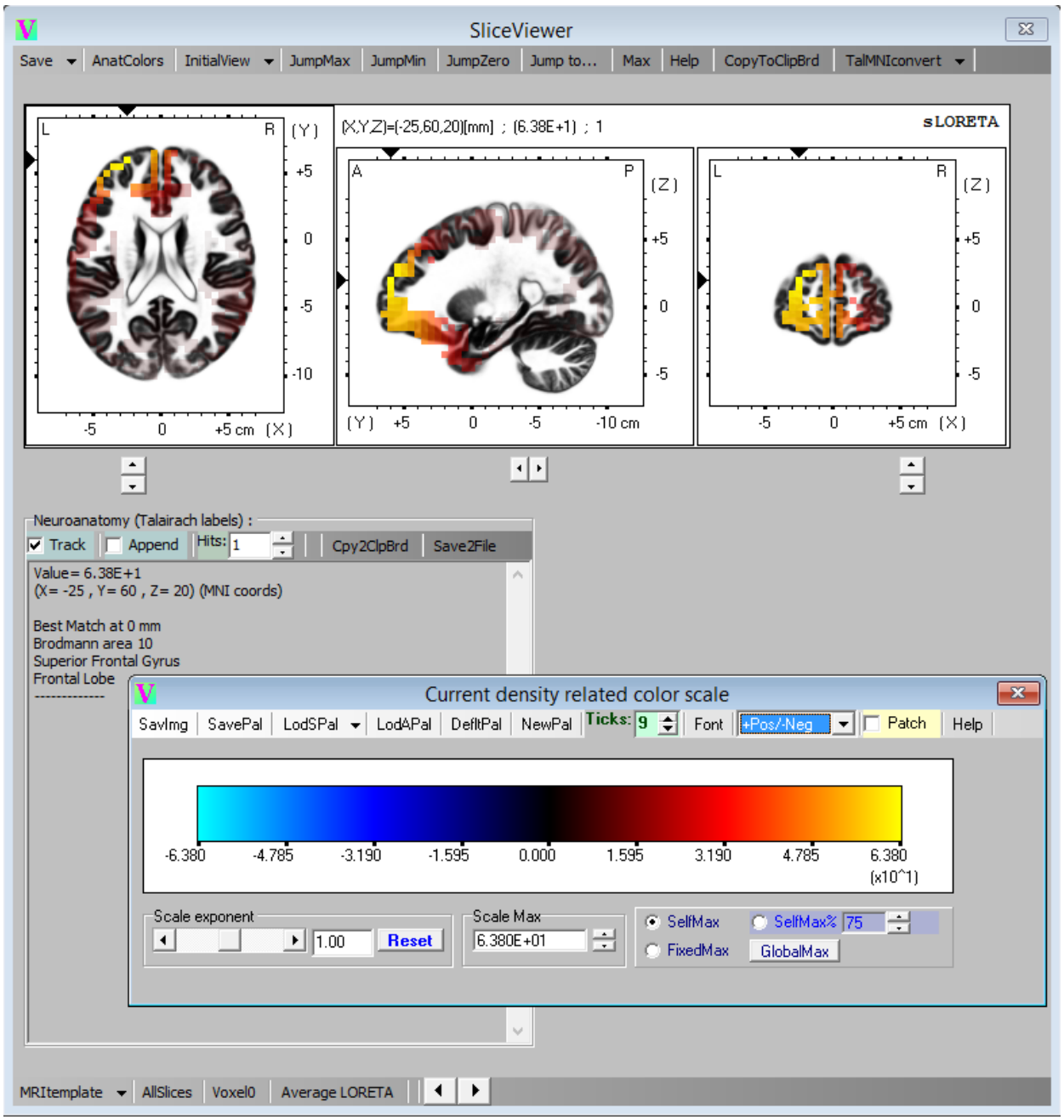

Figure 3.27. sLORETA SliceViewer. Left: Cross section of the head from the top. Middle: Cross section of the head from the left side. Right: Cross section of the head from the front.

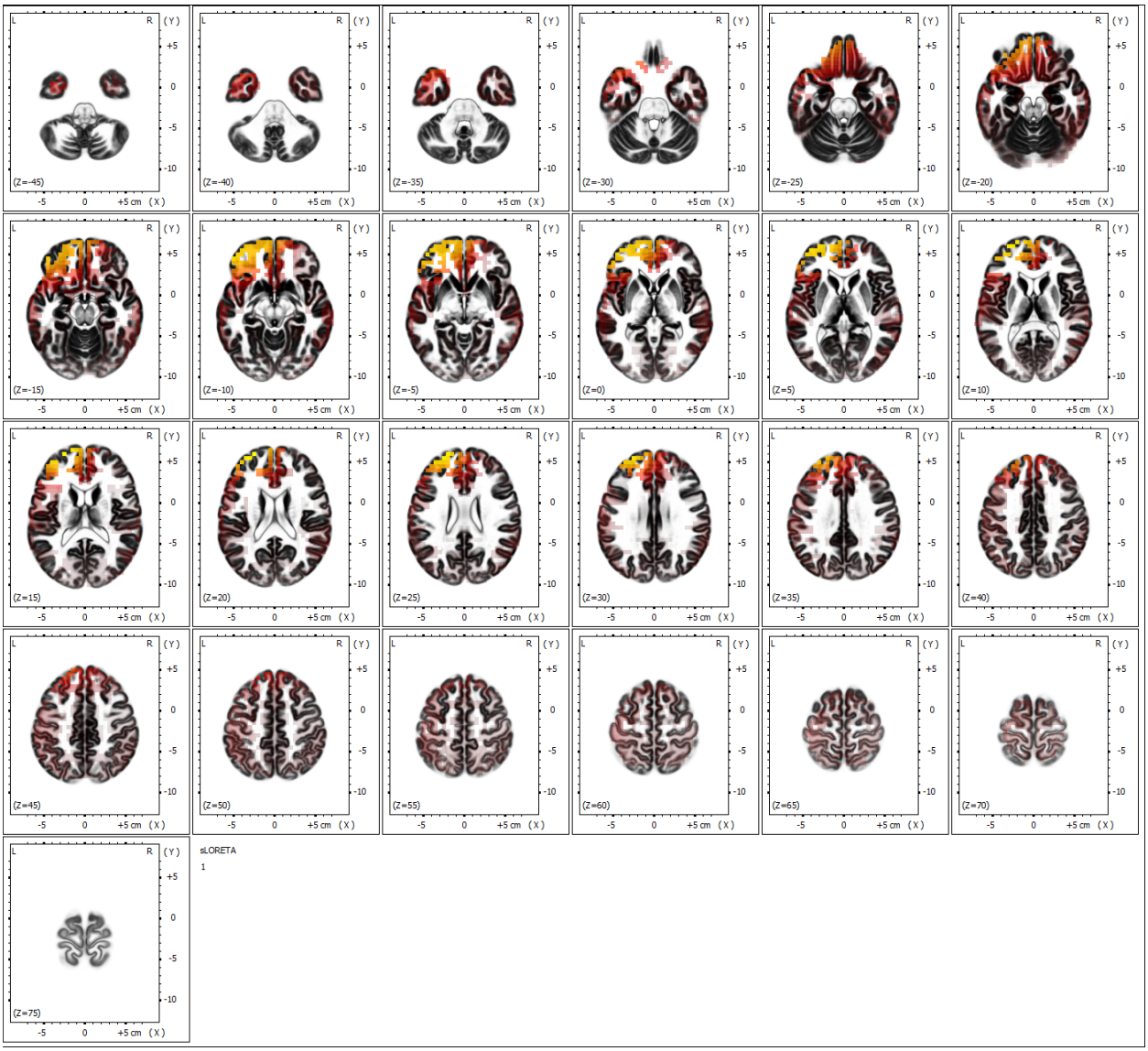

Figure 3.28. Cross sections of the head, starting from the bottom slowly progressing to the top.

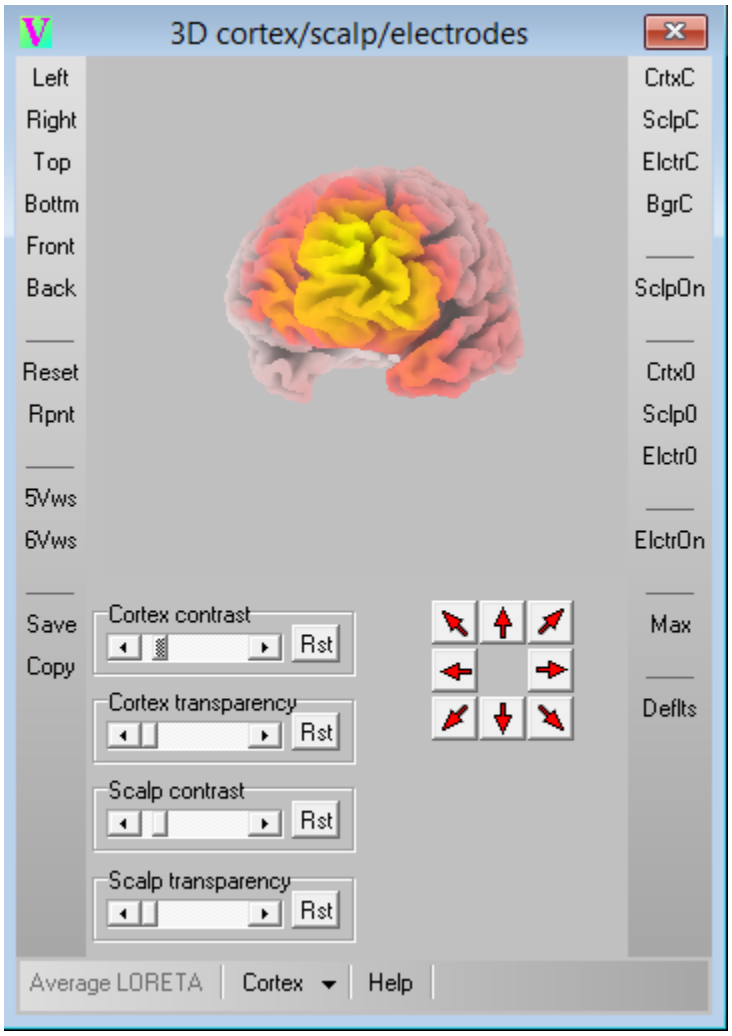

Figure 3.29. 3-D representation of the activation in the brain.

3.5 Determining Inputs

An initial analysis of the data was done to see which Brodmann areas had the highest excitation among all of the candidates during the sessions. The areas of interest that showed the highest voxels were Brodmann areas 7, 9, 10, 11, 20, 21, 39, and 47. Their locations can be seen in Figure 3.30. Due to the nature of Brodmann areas, there are multiple stimuli that excite any given area. For this reason, the following descriptions of the Brodmann areas focus on the generalized role along with any relevant notes. A broader explanation of each area can be found

in the Cortical Functions Reference Manual (2012). Areas will be presented in numerical order starting with Brodmann area 7 [44].

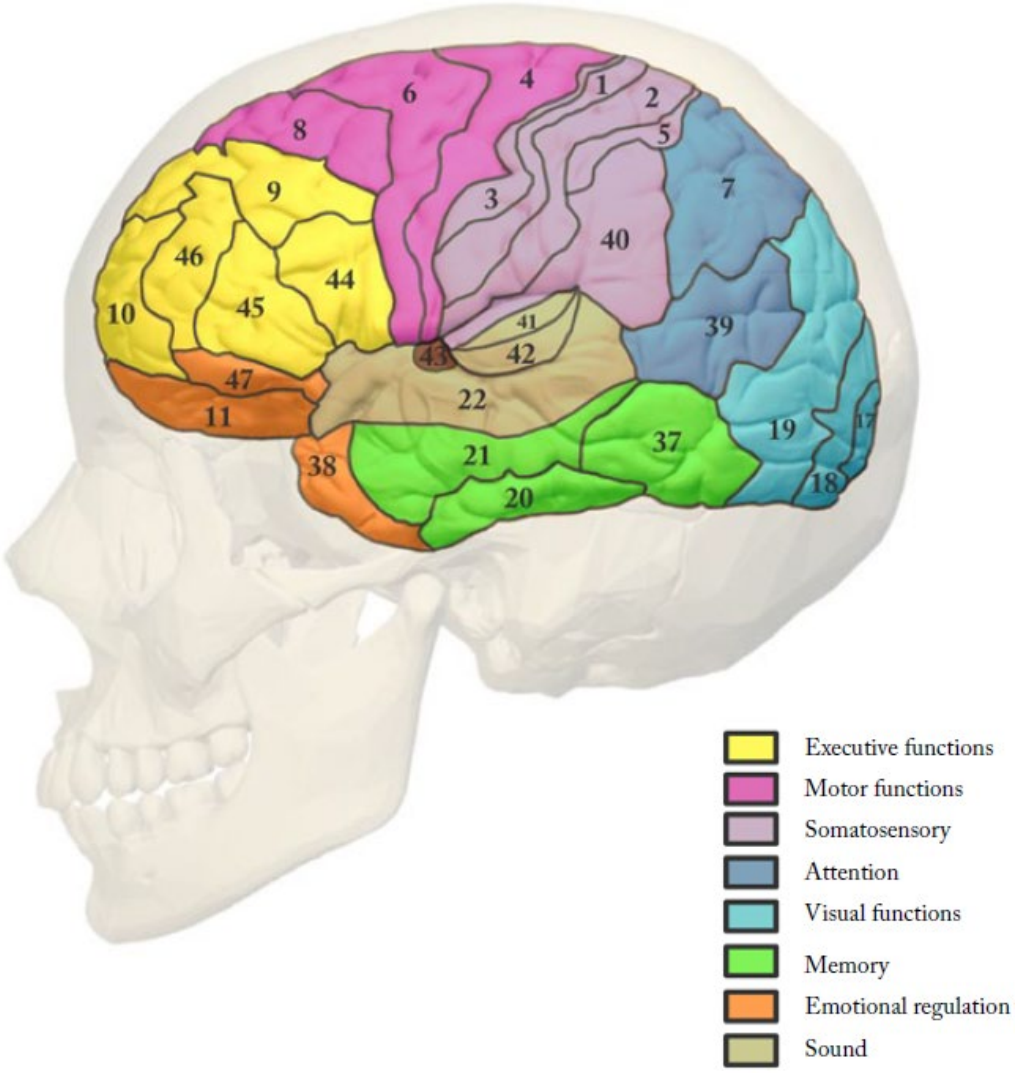

Figure 3.30. Brodmann Cortical Areas [44].

Brodmann area 7 is referred to as the somatosensory association cortex. It is part of the parietal cortex, which is believed to play a role in visuo-motor coordination, such as reaching to grasp an object. The region that Brodmann area 7 (the superior parietal lobe) is associated with rhyme detection and semantic categorization tasks, along with temporal context recognition.

Some additional processes worth noting are association with working memory (motor, visual, auditory, emotional, verbal) [44].

The dorsolated prefrontal cortex includes Brodmann areas 9 and 46, by a more restricted definition, but in a broader definition, it also includes areas 9 through 12 along with 45 through 47. Areas 9 and 10 are significate in brain operations involving memory. More specifically, in memory encoding, memory retrieval, and working memory. Brodmann areas 9 and 10 show certain executive functions as well. These include “executive control of behavior”, “inferential reasoning”, and “decision making”. There is a long list of additional processes, but the ones of note to this study include: Error processing/detection, attention to human voices, and calculation/numerical processes [44].

Area 11, the orbitofrontal area, is associated with general olfaction, nonspeech processing, decision making involving reward, and face name association. This area covers the medial part of the ventral surface in the frontal lobe, meaning that it is near areas 9, 10, and 47 [44].

The inferior temporal gyrus is area 20, which is part of the temporal cortex. This area is believed to be a major part of high-level visual processing and recognition memory. Also, worth noting is visual fixation and dual working memory task processing. Area 21 is the middle temporal Gyrus. Its exact function is unclear, but holds connections to processing distance, recognition of known faces, deductive reasoning and accessing word meaning while reading [44].

Area 39 is the angular gyrus. This area is associated with a number of processes including language, mathematics and cognition, including theory of mind. The left side in particular involves calculations, arithmetic learning, and abstract coding of numerical magnitude [44].

Lastly, the inferior prefrontal gyrus is area 47. Area 47 is involved in many functions related to language and emotion. Other processes include working memory, nonspatial auditory processing, decision making, and deductive reasoning [44].

3.6 Fuzzy Controller

The MATLAB controller script, included in Appendix D, allowed for a quick loading of the all the data processed with sLORETA. The data was organized into columns based on Brodmann Area and the rows were based on time. In the controller creator script, there is clear distinction between the required inputs that need to be changed and then the rest of the script that can be changed, as desired, to add additional functionality. The script is commented to allow for easier visualization and aids the user in finding areas to add unique nuances for their data set.

There are a few variables that need to be setup. The *MainFolder* variable leads to the folder with all of the data files. The *xlsread* function is designed to read Microsoft Excel documents and extract the data from them. The data can have empty cells and be of mismatched size, and the script will accommodate this. If baselines are available, they can be placed in any column and designated by *Baselineloc* variable. The column location must be the same for all sheets. If a single baseline applies to multiple sheets, this can be copy-and-pasted into all relevant sheets in the identified column.

Since the script allows for mismatched sizes of data, it is possible that a data set could be weighted more based on having more data points when calculating average and standard

deviation. The *EqualWeight* variable produces pre-calculations which provide each dataset an equal weight. The next three variables, *PerLow, PerMed, PerHigh,* are percentiles used later in the script to develop the bounds for the membership functions. The *InpOverlap* variable determines how wide the input variables are based on standard deviations. When set to 1, one standard deviation is the width of the membership functions. At a value of 0.5, the width would be half of that. The difference can be seen in Figure 3.31. These variables can be tunned based on the data and the rules developed.

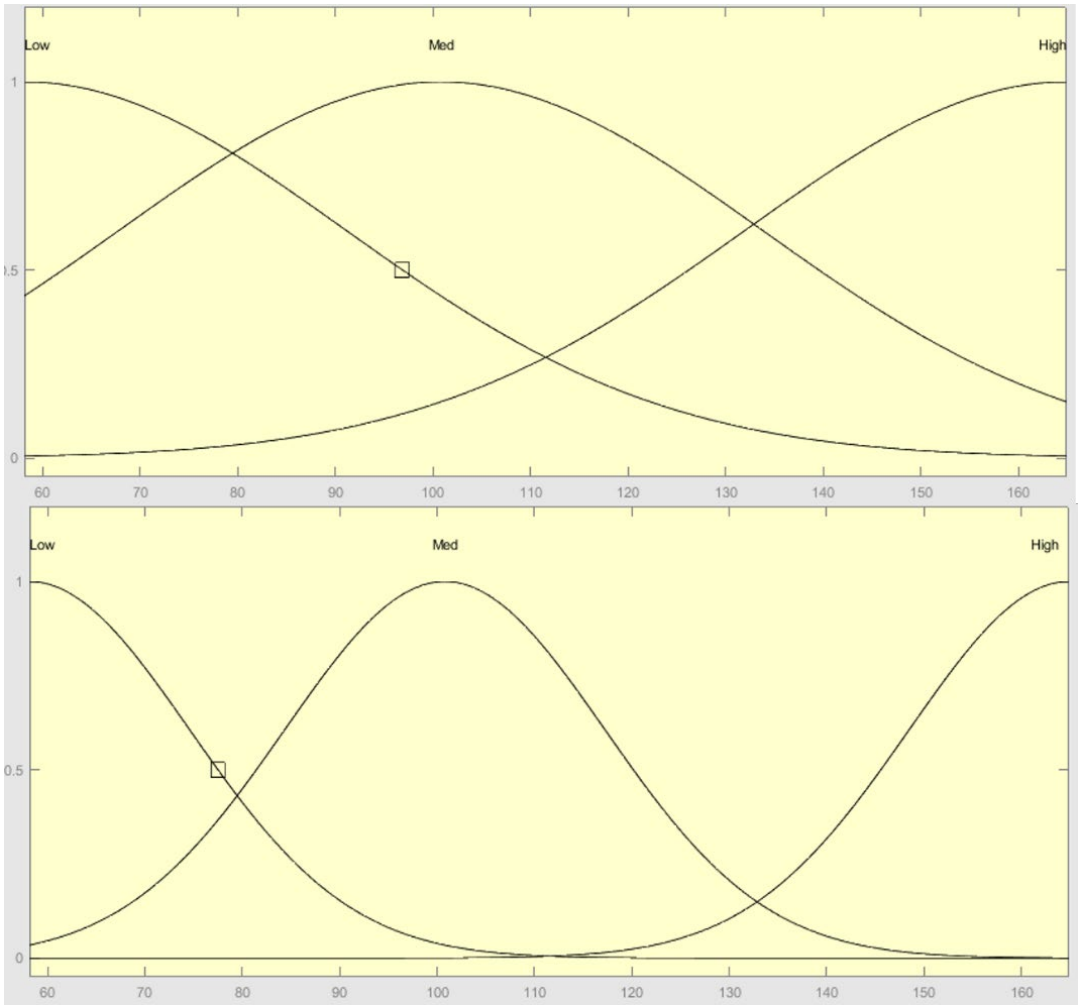

Figure 3.31. Membership functions based on *InpOverlap* being 1 standard deviation (top) and being 0.5 (bottom).

The *Controller* variable establishes the controller and names it. The next block of code is used to set up the inputs. The *nameInput* variable names the columns going from left to right. The *nameMF* and *input* variables are linked in the sense that the first of *nameMF* is the name of the membership function, while the first of *input* is the type of membership function, such as *gaussmf* for a Gaussian membership function. In the script, a list of other type of membership functions that can be used.

The next section of the script sets up the outputs of the controller. Using the *nameOutput* variable, as many outputs as desired can be created. The membership functions can be changed with the same list of membership functions provided. The defuzzification method can also be set under the outputs. Other methods can be found in the fuzzy controller help section or the drop-down areas in the controller itself.

The most impactful section for the output of the controller is the rule generation. This is where the most thought and time will be invested. Figure 3.32 shows the process of rule generation for the math study. The Brodmann areas are first associated with the Brodmann cortical regions shown in the blue column. The rule column provides a more detailed explanation of each input. These descriptions are then mapped to the chosen outputs based on their perceived effect on each output. The green, blue, and yellow lines represent high, medium, and low effect, respectively [44].

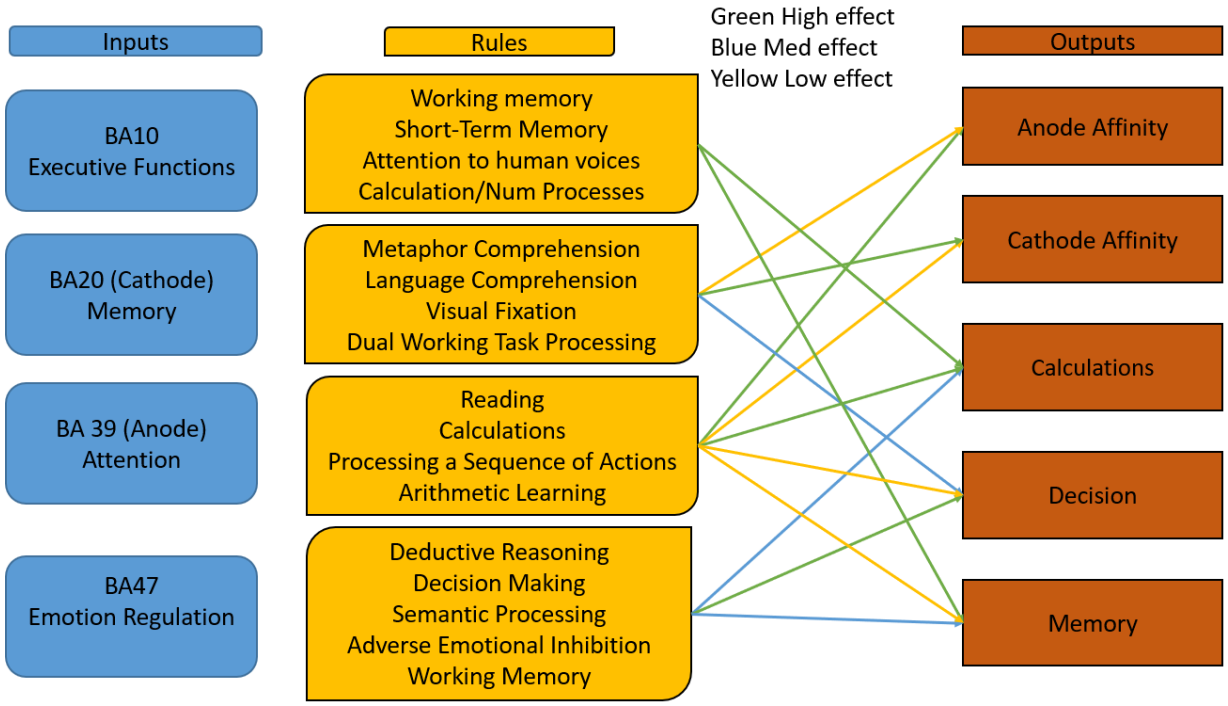

Figure 3.32. Rule generation for the math study.

The order of rules determines how the rules appear in the rule viewer. Figure 3.33 illustrates the appearance of the rule viewer. It is a visual representation of each individual rule. The inputs can be changed to observe how the outputs react. This can be used to understand how each rule affects the output. The input at the bottom left of the rule viewer can be changed to subject data to see the breakdown for each subject. The *Acopy* variable at the end of the script can be changed to the session of interest, and then, the output copied into the rule viewer to see the how the rules are affecting this particular session.

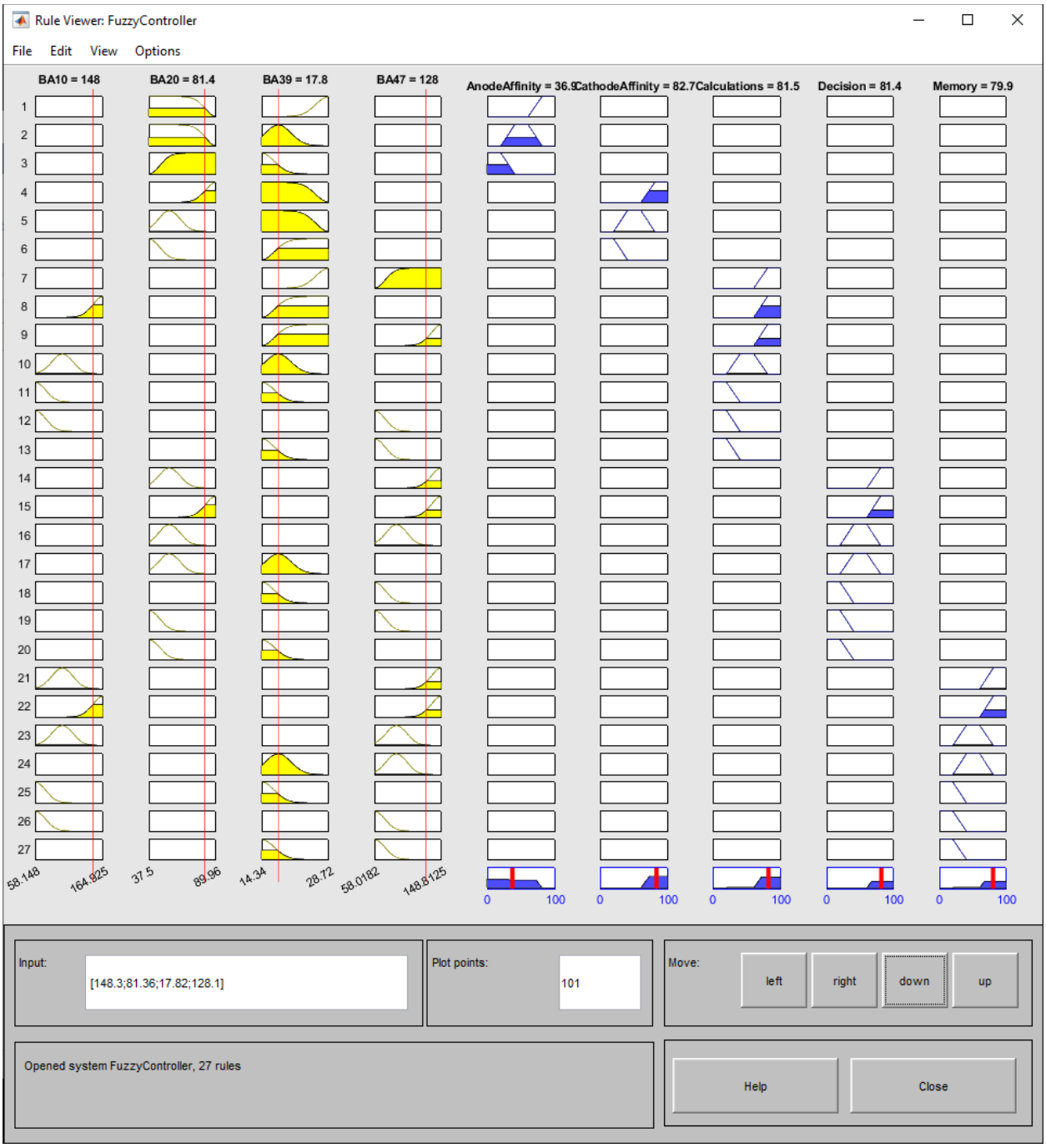

Figure 3.33. MATLAB ruleviewer showing the breakdown of outputs based on rules.

The process of creating rules in the script is shown in Figure 3.32. The script requires a specific syntax for the rules. The "==" sign is used for variables to show that particular variable

is equal to high, med, or low. “BA39==High => Calculations=High” means that when Brodmann area 39 is high then this output calculations is high. There is also coding for “or” and “and” shown by | for or and & for and. In addition, the “~=” sign can be used to show that the output is true when the input is not equal to low. “BA39~=High & BA20==High => CathodeAffinity=High” means that when Brodmann 39 is not low and Brodmann area 20 is high then there is a high CathodeAffinity. The rule variable is setup with the first rule, being the *rule(1)=*. The rest of the rules should follow the same format of *rule(end+1)=* which adds the current rule to the end of the rule matrix, which ultimately is merged into the controller. At the bottom of the rule viewer, the culmination of the rules and the defuzzification method are indicated by the red lines on the right hand side (Figure 3.33).

3.7 Study Results

This study focused on tDCS data; thus, the outputs include anode affinity, cathode affinity, calculations, decision, and memory. The full outputs are shown in a heat map displayed in Figure 3.34. To offset the fact that defuzzification with the centroid method never gives a score of 100, the data has been normalized. The script was ran and data analyzed incrementally which allowed the rules to be evaluated and tuned each iteration for more accurate results. The script also displays z-scores based on local and global data as shown in Figure 3.35 and Figure 3.36, respectively. Each subject has a C or an E after their subject number, such as 1C and 3E. The C stands for “control” and E for “experiment”. The E subjects are the tDCS stimulated individuals.

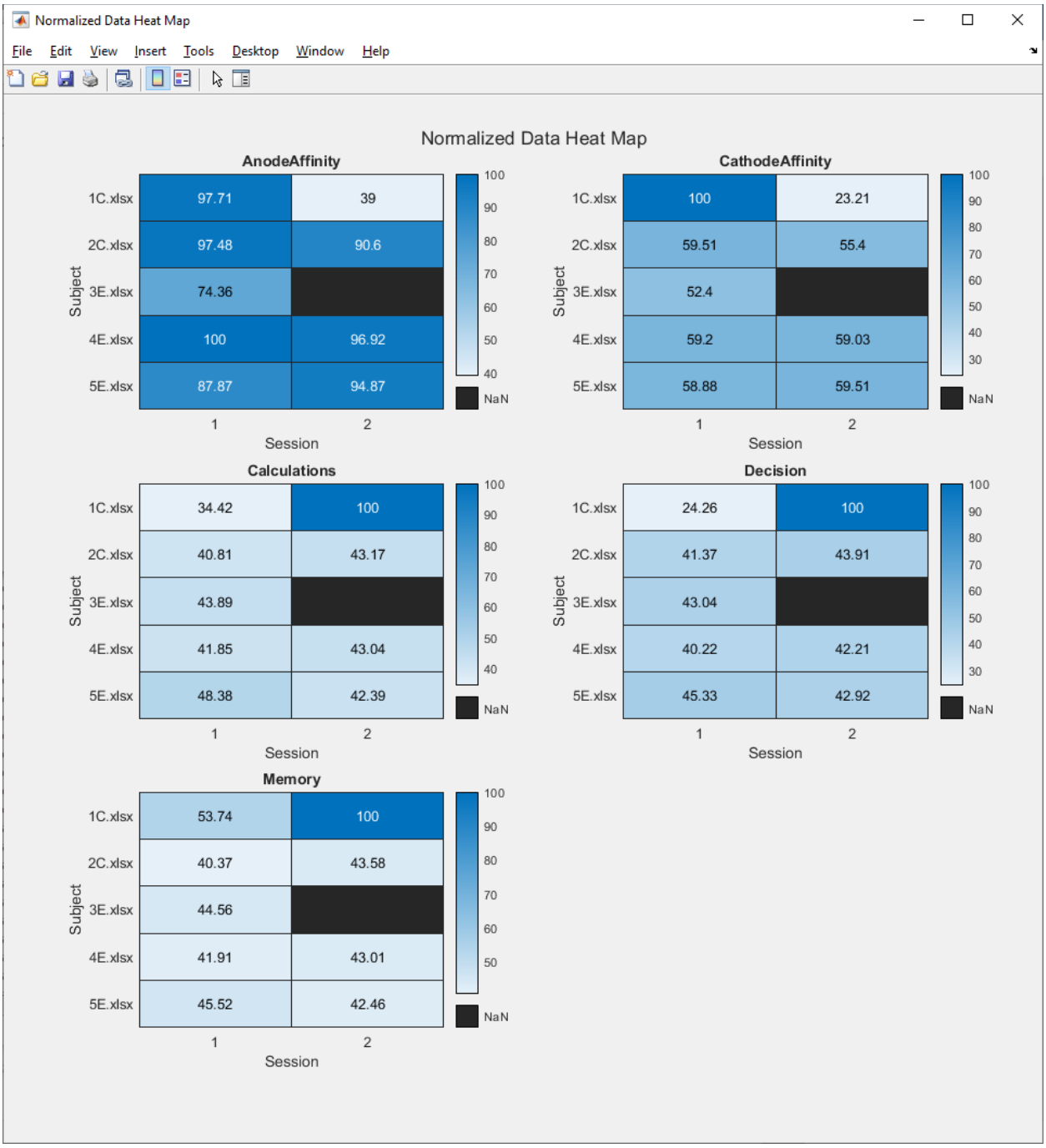

Figure 3.34. Math study normalized data.

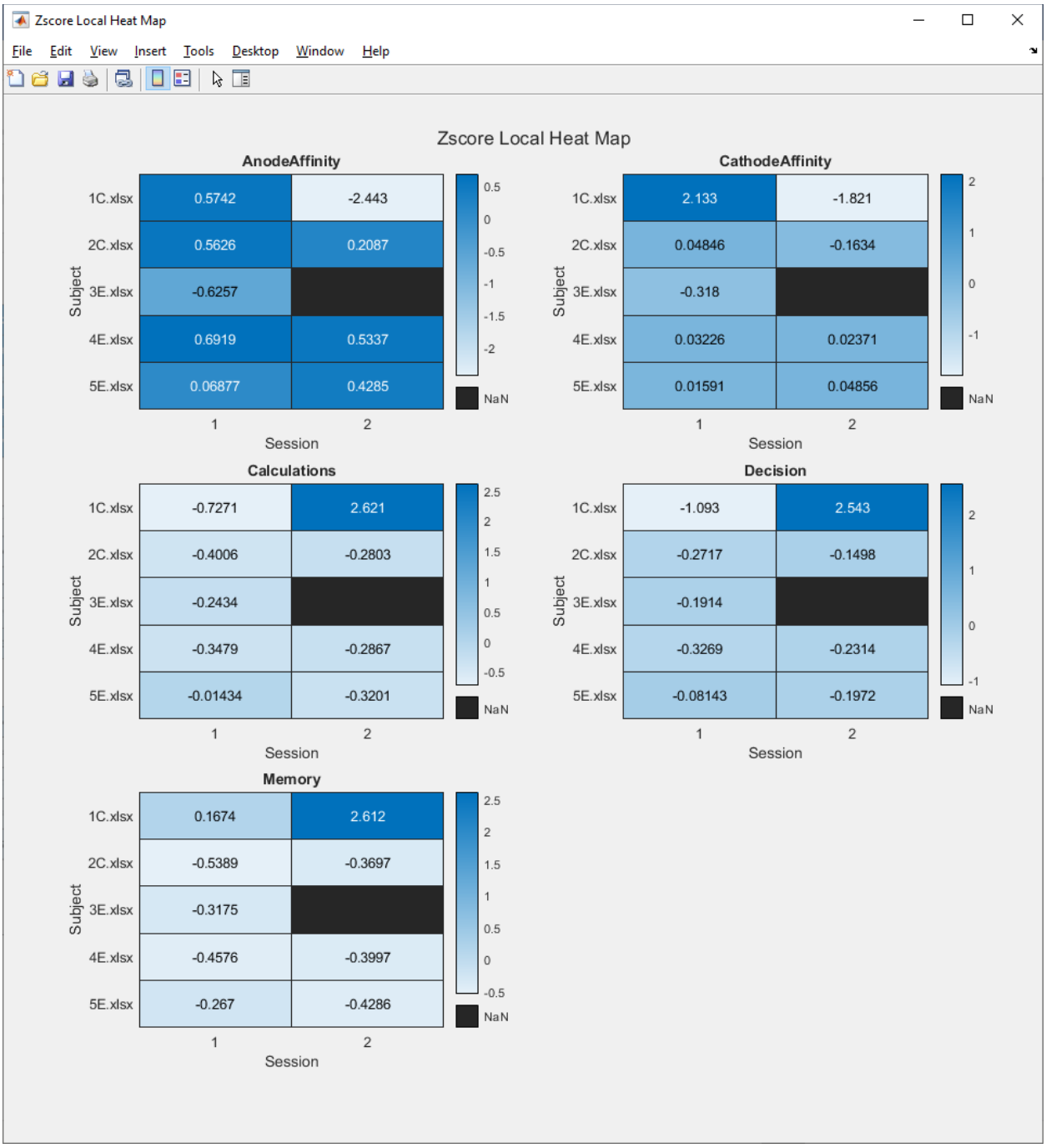

Figure 3.35. Math local z-score heat map

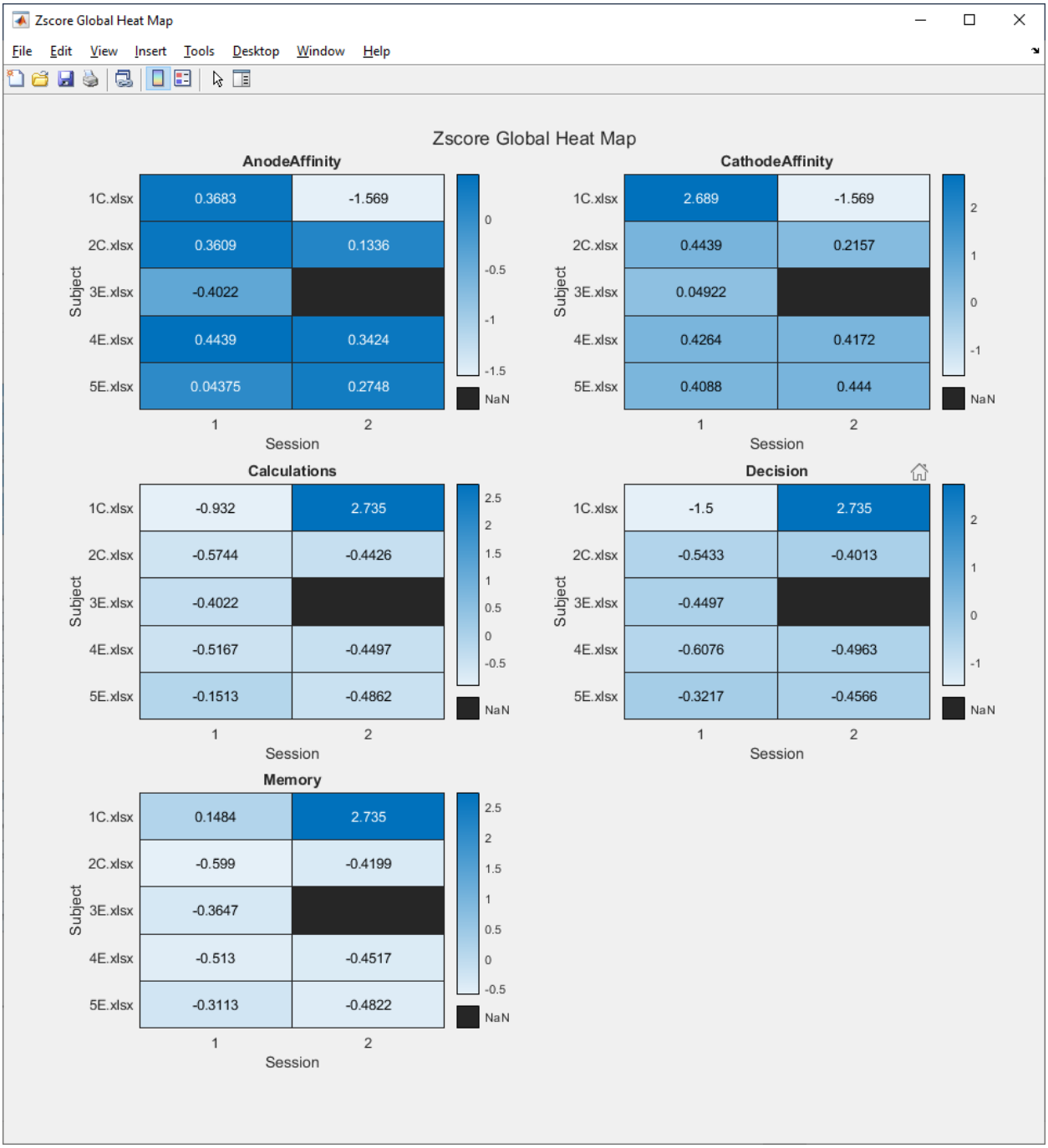

Figure 3.36. Math global z-score heat map.

An additional analysis (Figure 3.37) interprets the outputs as data points in a 5th dimensional space. This allows for the distance formula to be utilized to determine how different (far away) or similar (close) the datapoints are. The heatmap is used to show the distance from one session to the next. The darker colors represent data that is further apart, and the lighter colors represent data that shares a similar area in 5th dimensional space, representing sameness in the datapoints.

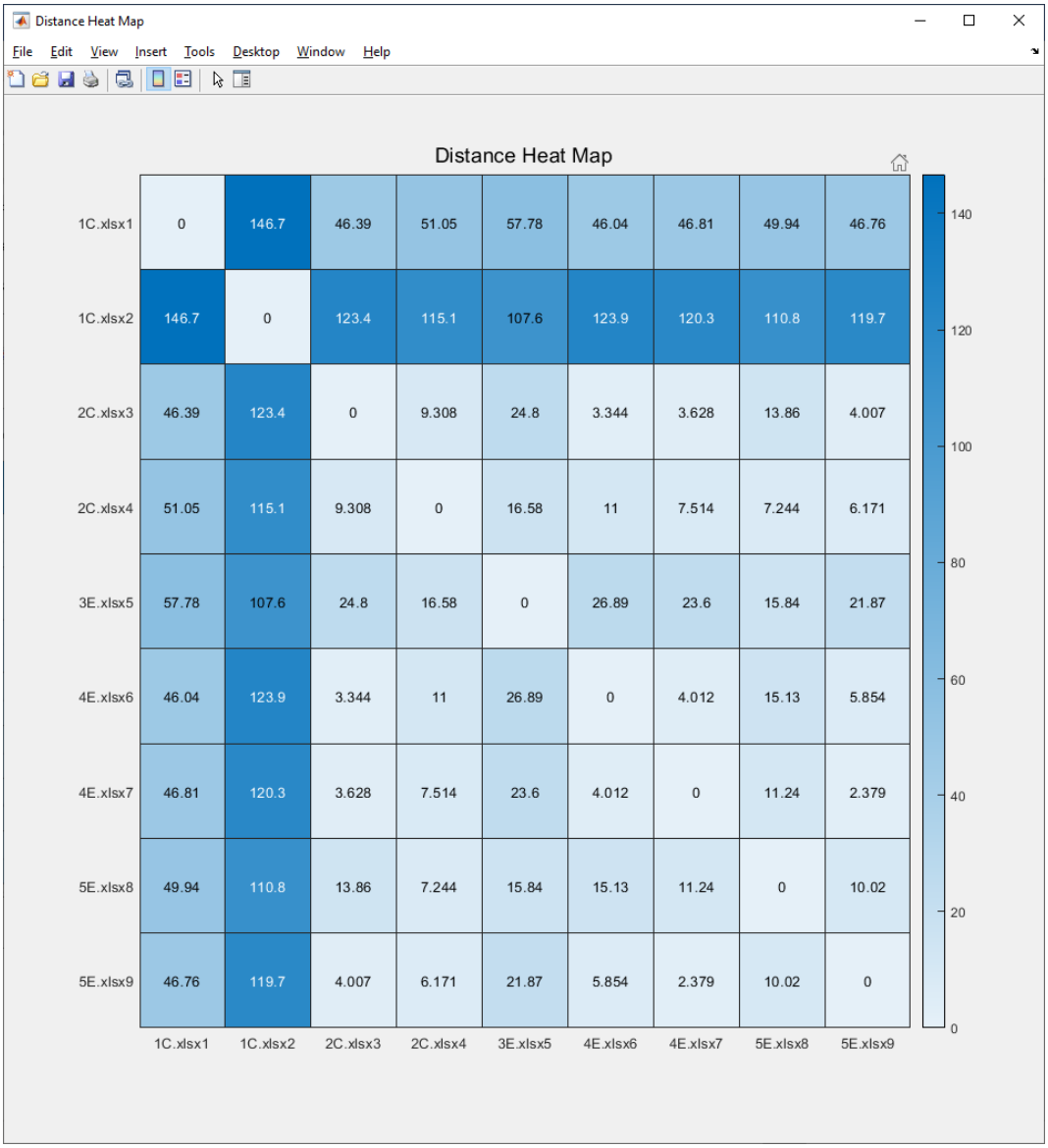

Figure 3.37. Math distance heat map.

This study contained nine sessions of interest. Therefore, the data was more prone to being skewed by outliers. Subject 1C had z-scores greater than 2 standard deviations from the other sessions. By removing Subject 1C, the remaining data is shown clearer by shifting the average and allowing datapoints to be represented better. The updated normalized data heat map, local z-scores and global z-scores are depicted in Figure 3.38, Figure 3.39, and Figure 3.40. Lastly, the updated distance heat map shows a less extreme difference in the data in Figure 3.41.

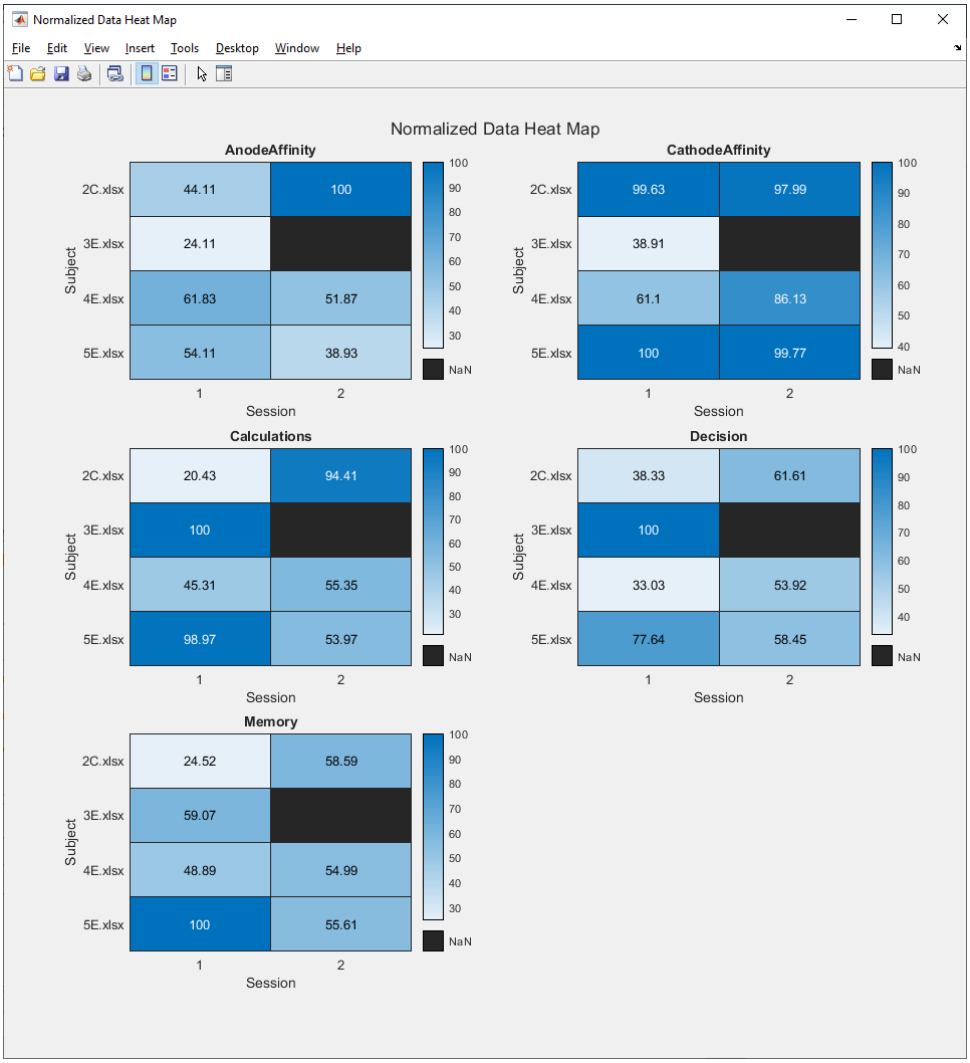

Figure 3.38. Updated math normalized data heat map.

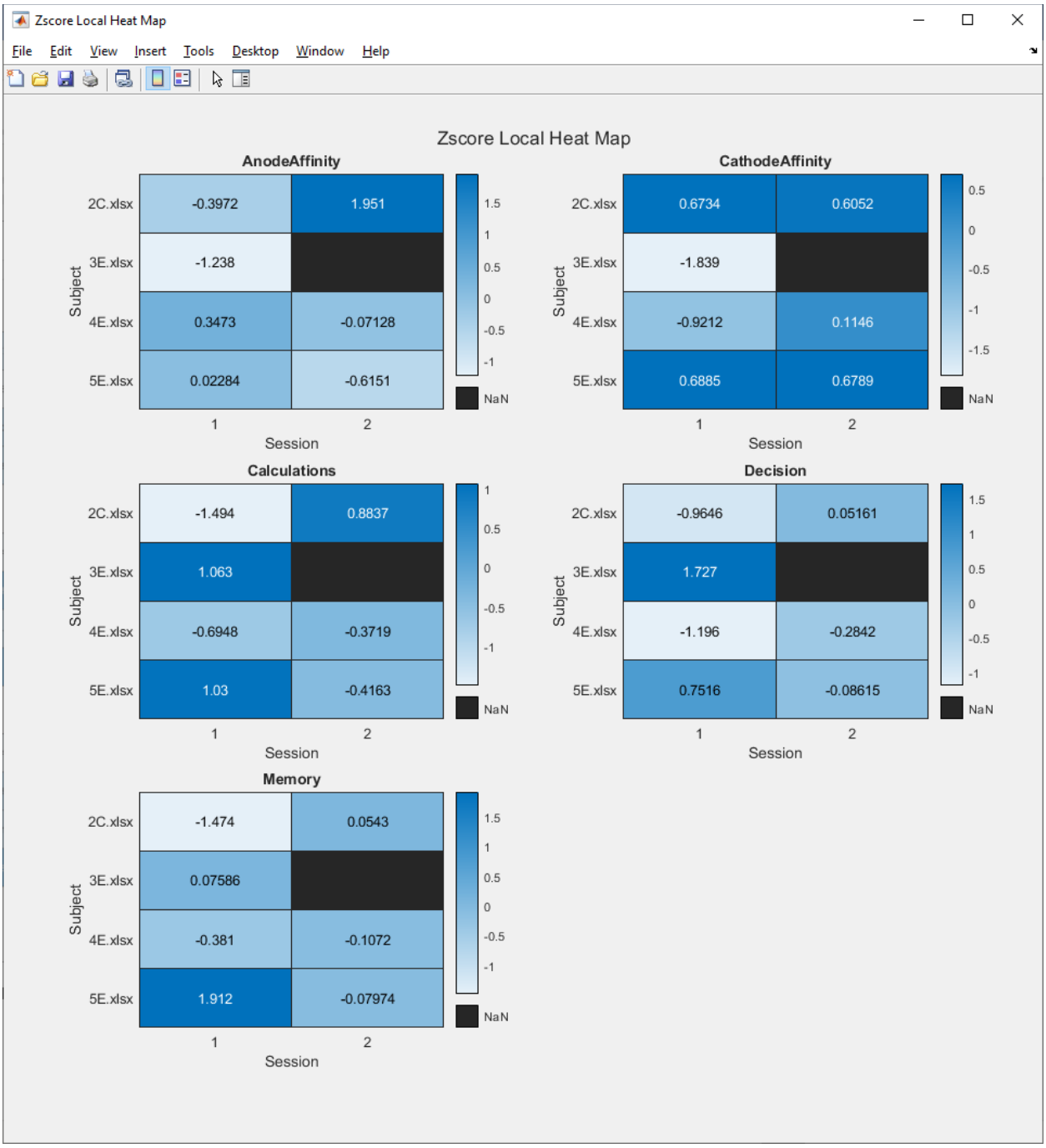

Figure 3.39. Updated math Local z-score data heat map.

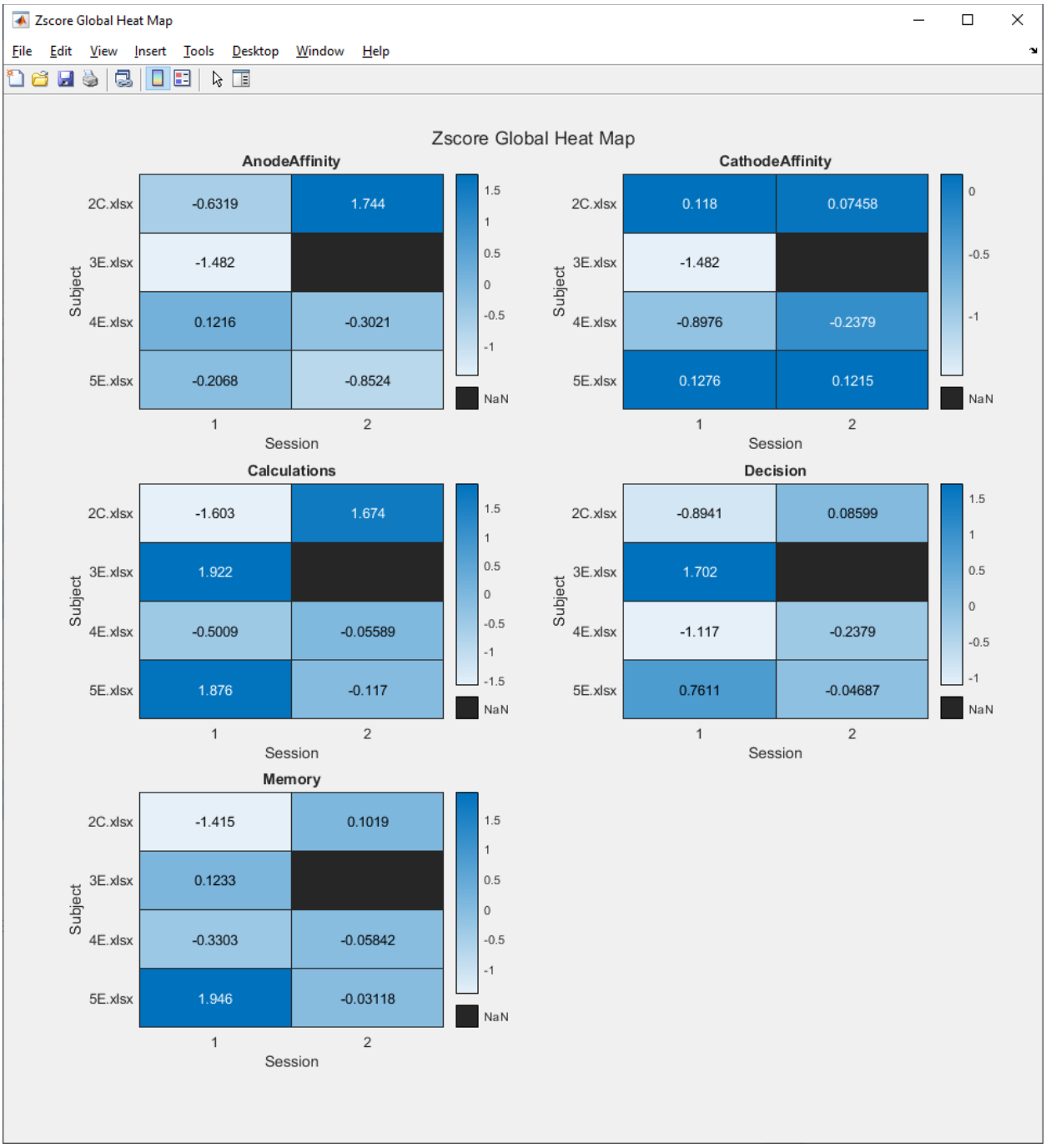

Figure 3.40. Updated math global z-score data heat map.

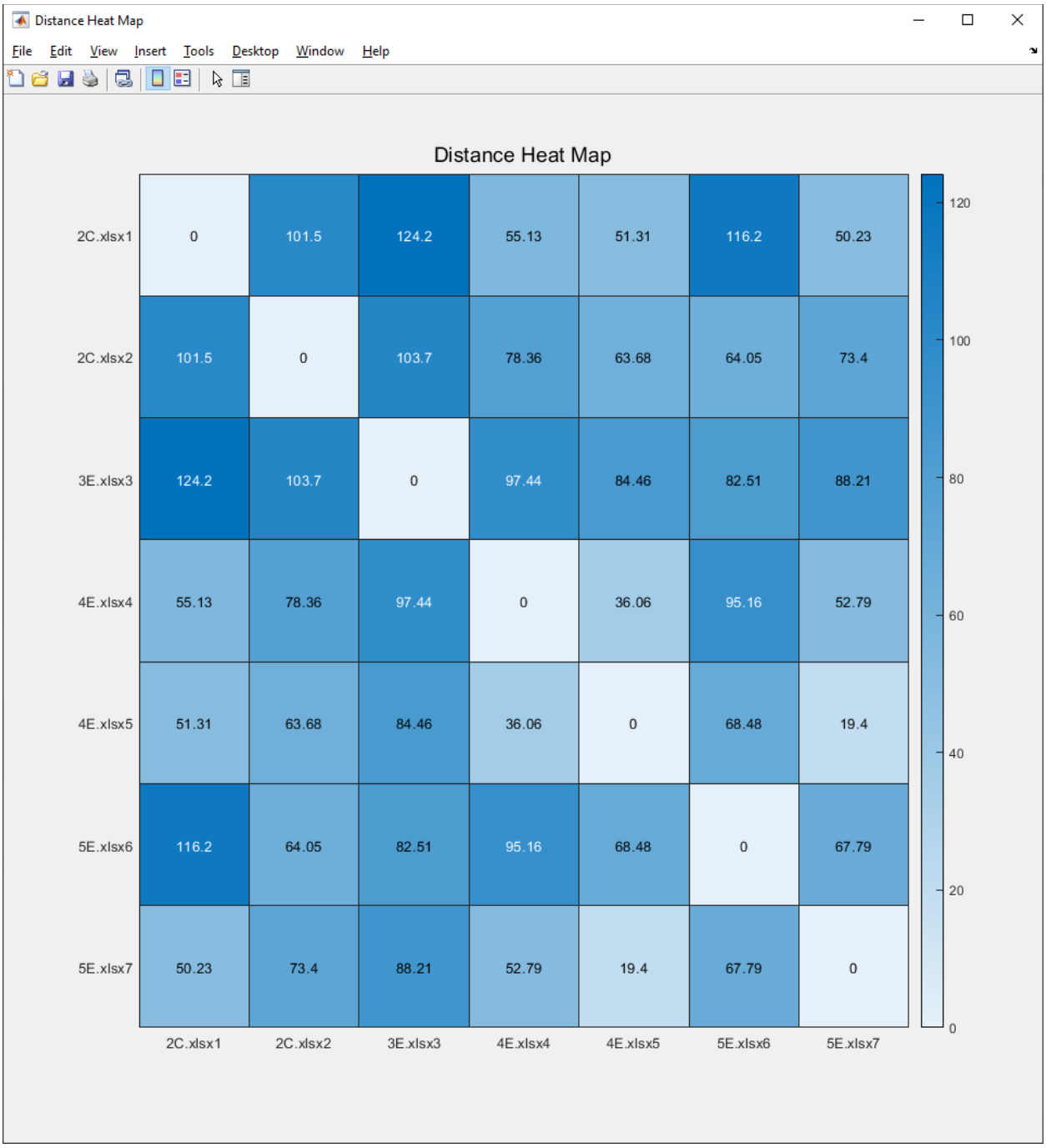

Figure 3.41. Updated math data distance heat map.

3.7.1 Discussion of Attained Results

The math data provided an insight into the limitations of the controller. The initial run with all of the data showed a higher affinity towards the Anode for most of the data, but this was potentially due to the skew of the data due to Subject 1C. When Subject 1C data was removed, the remaining data shifted to show higher affinity in the cathode region, but again, the data set is smaller than desired for this type of controller.

One option that arises, even with small data sets, is to contrast how well the activation affected each particular subject's sessions. This still falls "victim" to the overall dataset, but it is possible to determine things such as Subject 5E seemed to favor the cathode over the anode, while Subject 4E favored the cathode more during his 2nd session. The control group also has relevant data to unpack, as the activation shown by the group is unaffected by stimulation. This means that Subject 2C consistently used the cathode region about the same but favored the region the anode was placed in during their 2nd trail.

The tasks designated changed between the sessions, which could explain this behavior. The tasks dictated to the subjects activate different areas of the brain, allowing for a better understanding of brain development overtime. The end goal with this controller is to add more data as time progresses. If the anode starts to show higher affinity, then the positions of the cathode and anode can be swapped to account for such a change, and the new scores can then be compared in the calculations, memory, and decision outputs to see how modifying the tDCS stimulation affected the brain of the subject.

On an individual level, the math data also draws some interesting conclusions. It is a well asserted fact that people react differently to tDCS. While comparing between stimulated individuals allows for an analysis of how the subjects reacted in the grand scheme of things, a

purely personal analysis would allow a subject to measure up their own mind against itself. This would show the growth or change in activation and affinity overtime. Each subject is able to determine their areas that could benefit from being stimulated and thus work to improve that skill set overtime with the aid of tDCS. Subject 3E seems to favor decision making, followed by calculations, and lastly memory. The presented material might have been more familiar and allowed for more decisions to be made along the way. In future sessions, the individual could try to stimulate the calculation region more if they are already familiar with the topic. These results will be discussed further in Chapter 5.

Chapter 3 References

[1] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, "EEG signal classification for BCI applications by wavelets and interval type-2 fuzzy logic systems," *Expert Systems with Applications,* vol. 42, no. 9, pp. 4370-4380, 2015.

[2] C. M. Michel, M. M. Murray, G. Lantz, S. Gonzalez, L. Spinelli, and R. G. de Peralta, "EEG source imaging," *Clinical neurophysiology,* vol. 115, no. 10, pp. 2195-2222, 2004.

[3] M. A. Nitsche and W. Paulus, "Excitability changes induced in the human motor cortex by weak transcranial direct current stimulation," *The Journal of Physiology,* vol. 527, no. 3, pp. 633-639, 2000.

[4] K.-A. Ho *et al.*, "The Effect of Transcranial Direct Current Stimulation (tDCS) Electrode Size and Current Intensity on Motor Cortical Excitability: Evidence From Single and Repeated Sessions," *Brain Stimulation,* vol. 9, no. 1, pp. 1-7, 2016.

[5] T. Chew, K.-A. Ho, and C. K. Loo, "Inter- and Intra-individual Variability in Response to Transcranial Direct Current Stimulation (tDCS) at Varying Current Intensities," *Brain Stimulation,* vol. 8, no. 6, pp. 1130-1137, 2015.

[6] M. A. Nitsche *et al.*, "Transcranial direct current stimulation: state of the art 2008," *Brain stimulation,* vol. 1, no. 3, pp. 206-223, 2008.

[7] C. J. Stagg and M. A. Nitsche, "Physiological Basis of Transcranial Direct Current Stimulation," *The Neuroscientist,* vol. 17, no. 1, pp. 37-53, 2011.

[8] C. Poreisz, K. Boros, A. Antal, and W. Paulus, "Safety aspects of transcranial direct current stimulation concerning healthy subjects and patients," *Brain Research Bulletin,* vol. 72, no. 4–6, pp. 208-214, 5/30/ 2007.

[9] A. R. Brunoni *et al.*, "Clinical Research with Transcranial Direct Current Stimulation (tDCS): Challenges and Future Directions," *Brain Stimulation,* vol. 5, no. 3, pp. 175-195, 2012.

[10] A. Alonzo, J. Brassil, J. L. Taylor, D. Martin, and C. K. Loo, "Daily transcranial direct current stimulation (tDCS) leads to greater increases in cortical excitability than second daily transcranial direct current stimulation," *Brain Stimulation,* vol. 5, no. 3, pp. 208-213, 2012.

[11] V. P. Clark *et al.*, "TDCS guided using fMRI significantly accelerates learning to identify concealed objects," *NeuroImage,* vol. 59, no. 1, pp. 117-128, 1/2/ 2012.

[12] R. J. Sadleir, T. D. Vannorsdall, D. J. Schretlen, and B. Gordon, "Transcranial direct current stimulation (tDCS) in a realistic head model," *Neuroimage,* vol. 51, no. 4, pp. 1310-1318, 2010.

[13] H. S. Suh, S. H. Kim, W. H. Lee, and T. S. Kim, "Realistic simulation of transcranial direct current stimulation via 3-d high-resolution finite element analysis: Effect of tissue anisotropy," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 638-641.

[14] I. Chang-Hwan, J. Hui-Hun, C. Jung-Do, L. Soo Yeol, and J. Ki-Young, "Determination of optimal electrode positions for transcranial direct current stimulation (tDCS)," *Physics in Medicine and Biology,* vol. 53, no. 11, p. N219, 2008.

[15] L. A. Zadeh, "Fuzzy sets," *Information and Control,* vol. 8, no. 3, pp. 338-353, 1965/06/01 1965.

[16] B. Jankuloski and Maksim, "A comparison between a fuzzy set and crisp set.," F. crisp.gif, Ed., ed. Wikipedia.

[17] H.-J. Zimmermann, Fuzzy Set Theory—and Its Applications, 4 ed.: Springer Netherlands, 2001. [Online]. Available.

[18] B. E. Postlethwaite, "Fuzzy control and fuzzy systems, W. Pedrycz, Research Studies Press, 1989, ISBN 0471 9231 17, xiv + 225pp. £37.50," *International Journal of Adaptive Control and Signal Processing,* vol. 5, no. 2, pp. 152-152, 1991.

[19] R. Zhang, Y. Phillis, and V. Kouikoglou, 1, Ed. *Fuzzy Control of Queuing Systems*. Springer-Verlag London, 2005.

[20] P. Singhala, D. N. Shah, and B. Patel, "Temperature Control using Fuzzy Logic," *International Journal of Instrumentation and Control Systems,* vol. 4, no. 1, 2014.

[21] G. Gerla and L. Scarpati, "Extension principles for fuzzy set theory," *Information Sciences,* vol. 106, no. 1–2, pp. 49-69, 4// 1998.

[22] K. M. Passino and S. Yurkovich, *Fuzzy Control*, 1 ed. Addison-Wesley, 1998, p. 475.

[23] W. V. Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," *Fuzzy Sets and Systems,* vol. 108, no. 2, pp. 159-178, 12/1/ 1999.

[24] I. Iancu, *A Mamdani Type Fuzzy Logic Controller* (Fuzzy Logic - Controls, Concepts, Theories and Applications). InTech, 2012.

[25] B. Boffy, "Centroid defuzzification using max-min inferencing in a fuzzy control system.," Fuzzy_control_-_centroid_defuzzification_using_max-min_inferencing, Ed., ed. Wikipedia, 2003.

[26] D. DuBois, *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, Inc., 1997, p. 393.

[27] P. A. Nava, "A Neuro-Fuzzy System for Speech Recognition," in *Proceedings of the 1998 International Congress on Electronics and Electrical Engineering (ELECTRO 98)*, 1998, pp. 503-507.

[28] M. Almulla, H. Yahyaoui, and K. Al-Matori, "A new fuzzy hybrid technique for ranking real world Web services," *Knowledge-based Systems,* vol. 77, pp. 1-15, 2015.

[29] S. H. Zyoud, L. G. Kaufmann, H. Shaheen, S. Samhan, and D. Fuchs-Hanusch, "A framework for water loss management in developing countries under fuzzy environment: Integration of Fuzzy AHP with Fuzzy TOPSIS," *Expert Systems with Applications,* vol. 61, pp. 86-105, 2016.

[30] R. Parameshwaran, S. P. Kumar, and K. Saravanakumar, "An integrated fuzzy MCDM based approach for robot selection considering objective and subjective criteria," *Applied Soft Computing,* vol. 26, pp. 31-41, 2015.

[31] C. Marsala and B. Bouchon-Meunier, "Fuzzy data mining and management of interpretable and subjective information," *Fuzzy Sets and Systems,* vol. 281, pp. 252-259, 2015.

[32] E. E. Karsak and M. Dursun, "An integrated fuzzy MCDM approach for supplier evaluation and selection," *Computers & Industrial Engineering,* vol. 82, pp. 82-93, 2015.

[33] P. Kaur and K. Rachana, "An intutionistic fuzzy optimization approach to vendor selection problem," *Perspectives in Science,* vol. 8, pp. 348-350, 2016.

[34] E. Pourrahmani, M. R. Delavar, and M. A. Mostafavi, "Optimization of an evacuation plan with uncertain demands using fuzzy credibility theory and genetic algorithm," *International Journal of Disaster Risk Reduction,* vol. 14, pp. 357-372, 2015.

[35] Y. Wang and L. Chen, "Multi-view fuzzy clustering with minimax optimization for effective clustering of data from multiple sources," *Expert Systems with Applications,* vol. 72, pp. 457-466, 2017.

[36] E. E. Nikulin, D. G. Perepelitsa, O. A. Zhdanova, S. A. Smetanin, and E. V. Nazarova, "Fuzzy Portfolio Optimization Model With Estimation Of Results," *The Journal of Internet Banking and Commerce,* 2016.

[37] R. Saborido, A. B. Ruiz, J. D. Bermúdez, E. Vercher, and M. Luque, "Evolutionary multi-objective optimization algorithms for fuzzy portfolio selection," *Applied Soft Computing,* vol. 39, pp. 48-63, 2016.

[38] R. D. Pascual-Marqui, "Standardized low-resolution brain electromagnetic tomography (sLORETA): technical details," *Methods Find Exp Clin Pharmacol,* vol. 24, no. Suppl D, pp. 5-12, 2002.

[39] J. H. Park, S. B. Hong, D. W. Kim, M. Suh, and C. H. Im, "A Novel Array-Type Transcranial Direct Current Stimulation (tDCS) System for Accurate Focusing on Targeted Brain Areas," *IEEE Transactions on Magnetics,* vol. 47, no. 5, pp. 882-885, 2011.

[40] S. E. P. Nomenclature, "American Electroencephalographic Society Guidelines for," *Journal of clinical Neurophysiology,* vol. 8, no. 2, pp. 200-202, 1991.

[41] U. Herwig, P. Satrapi, and C. Schönfeldt-Lecuona, "Using the international 10-20 EEG system for positioning of transcranial magnetic stimulation," *Brain topography,* vol. 16, no. 2, pp. 95-99, 2003.

[42] S. C. f. C. Neuroscience. (2009, 5/24). *Chapter 01: Rejecting Artifacts*. Available: https://sccn.ucsd.edu/wiki/Chapter_01:_Rejecting_Artifacts

[43] R. A. Poldrack, J. A. Mumford, and T. E. Nichols, *Handbook of functional MRI data analysis*. Cambridge University Press, 2011.

[44] Trans Cranial Technologies (2012). Cortical Functions. Hong Kong: Trans Cranial Technologies.

CHAPTER 4

SHOOT /NO SHOOT DECISION MAKING UNDER DURESS: A POLICE OFFICER TRAINING STUDY

This work involves a test of the modularity of the scripts constructed in Chapter 3 and aims to prove that multiple types of analyses can be performed easily with minimal effort editing the scripts. The domain data used in this chapter comes from a law enforcement officer (LEO) study that consisted of volunteered officers from a small-sized city located in the Southern U.S. During the study, both expert and novice deputies were tested. Certified Trainers identified police officer's skill level in shoot or not-to-shoot situations within high threat environments. A virtual reality range was utilized to simulate high threat scenarios requiring split second decision making. Due to the motion of the officers and the nature of the study, the analysis required a considerable amount of noise filtering for the data to be processed further. Through this study, it is possible to observe how the officers' brains reacted as the scenarios progressed. The conducted research accomplished a fast and streamline way to analyze a multitude of EEG scenarios, with minimal changes to the script and approach previously developed.

4.1 Problem Background

The job of a law enforcement officer (LEO) in the United States is notably dangerous, as police-citizen encounters have the potential to turn to deadly violence. In 2016, there were 66 LEOs feloniously killed in the line of duty and 491 LEOs killed between the 2006 to 2015 [1]. Encounters with the public are extremely dynamic, and a situation may shift from compliance of

the citizen into an assaultive nature in a matter of seconds. Thus, it is imperative that officers receive enough training (both in the academy and in-service) in deciding when and how to use the proper amount of force in every police-citizen interaction. A key goal of research on LEO decision-making is to explore factors that can de-escalate or prevent situations ending in death for everyone involved.

4.2 Introduction

Law enforcement aims to maintain safety for the community and the officers that serve that community. Oliva, Morgan and Compton (2010) states that one of the primary objectives of law enforcement is to restore and preserve the peace as well as to ensure the safety of the community [2]. Yet, death and injury are still a real threat to law enforcement officers and community members, and events are well documented. Forty-one law enforcement officers were killed in the line of duty during 2015, and 491 killed between the years of 2006 to 2015 [1]. Moreover, Vila, James, James, and Wagoner (2012) stated that law enforcement officers kill approximately 400 people a year. In all of these situations, a court of law found the actions of the law enforcement officers to be justified [3]. Arrest related deaths from June 2015 to March 2016 was over 1300 individuals [4]. No statistics were collected on deaths of unarmed community members killed by law enforcement. Brittner (1970) concluded that any law enforcement and community member encounters potentially involve the use of force, despite it being the last resort. Law enforcement is best achieved when de-escalation efforts are utilized, so the encounter does not reach the use of deadly force. This is when the situation allows for de-escalation, as all scenarios do not allow for this [5].

A law enforcement officer's decision to shoot or not to shoot in high threat environments is complex. There are many factors to consider, such as physical skills (act of firing a weapon

with repetitive practice), physiological responses (highly stressed versus in control of emotions), individual perceptions that influence decisions (e.g. race), cognitive decision-making (anticipating the outcomes), and ethical decision-making (recognizing ethical issues). The underlying neuro processes in a decision to shoot or not to shoot in a high threat environment is not well understood. Given the significant impact of these decisions in high threat environment it is critical to begin investigating [6]. Another factor in the decision to shoot or not shoot is theory of mind (ToM). Theory of mind is attributed to the capacity to understand the metal states of others. This includes intentions, motivations, and beliefs. This realization drives the behavior of the individual interpreting the situation [7].

Research has been conducted into the neurological processes of law enforcement officers' decisions to shoot or not to shoot in high threat environments. In one such study, experts have shown differences in power, brain wave power, and power in different brain regions. Brain waves have been clearly differentiated by frequency. There are five different waves that have been identified as delta, theta, alpha, beta and gamma. Delta waves range from 1 to 4 Hz; theta waves contain the range of 4 to 7 Hz; alpha waves include 8 to 12 Hz; and beta waves are 13-30 Hz. Haufler, Spalding, Santa Maria, and Hatfield (2000) found that experts had an increased alpha power (10-11 Hz) in the left prefrontal cortex while aiming when compared with novices. Also, experienced officers had higher frequency in the theta bands (6-7 Hz) in both left and right prefrontal cortex during aiming [6].

Law enforcement officers may have little time to assess high threat situations in complex environments, and in that time, they must make a decision to shoot or not to shoot. An important, albeit often-neglected, aspect of this decision-making process is what happens after the threat has been neutralized. Following high threat situations that are ambiguous and challenging,

individuals engage in a period of cognitive reflection [8]. Since there is no obvious solution, this reflection period can further the knowledge and understanding of the law enforcement officer's brains process [9]. While the reflection period of high threat situations of law enforcement officers has yet to be investigated, there is empirical evidence from the medical profession that engaging in reflection is connected to improving decision-making. Medical doctors are much like law enforcement officers. They rely on previous training and expertise to make quick decisions, and decode difficult situations using automatic processes [10]. These automatic processes are effective in commonly occurring cases. Although, in complex cases that do not follow this mold, these processes are relatively ineffective as the doctors' collection of existing knowledge may not be useful to the current situation [10, 11]. Berner and Garber (2008) reviewed the accuracy of doctors' medical diagnoses and concluded that the doctors' inability to reflect on situations was connected to missed or wrong diagnosis [10]. Conversely, their ability to reflect on challenging cases is associated with improved diagnostic accuracy and better clinical reasoning [11, 12].

Studies on expert marksmen have utilized the reflection period in order to learn more about the decision-making process that occurs prior to firing a weapon [13]. The heartrate and skin conductance of novice and expert marksmen before, during, and after discharging pistols is comp1ared as well. These researchers focused on comparing the best and worst shots of both groups. In doing so, the researchers found evidence of two physiological processes of which are arousal and vigilance. Arousal is defined as anxiety that occurs during pressure situations, while vigilance is defined as attention concentrated on a particular stimulus. The study found that experts' arousal level had little to no variation before, during, and after the shot, while the novice marksmen's demonstrated arousal throughout the experiment. The more vigilance displayed by

the experts prior to the shot correlated with accuracy. Additionally, while the experts were more accurate, their vigilance decreased at a slower rate than when they missed. Tremayne and Barry (2001) interpreted this as the expert marksmen were more "locked in" prior to the shot are more reflective over a longer period of time after shooting [13].

Law enforcement officers' reflection and insight analysis in high threat decisions is still not prevalent. Similar findings from the medical profession and recreational shooting, provide evidence that reflection periods are critical for improved decision-making. There have been a few studies using physiological assessments such as EEG in an attempt to understand the reflection and insight [14-17]. Researchers have found that alpha, theta and gamma brain waves activity correlated with certain brain regions during periods of reflection. Kounious, et al., (2006) observed an increase in anterior cingulate cortex (midline) activity when engaged in insight orientated tasks and specifically prior to engaging in such tasks. They concluded that the increase activity may be associated with managing relevant thoughts and suppressing irrelevant thoughts to the task [15].

There are more factors that impact law enforcement officers' decision-making. These include the context of the situation and racial attitudes [18]. During emotional circumstances, the amygdala has been noted to be active and in issues of race the amygdala response is heightened further [18]. Halliburton noted that the amygdala serves as a signal for detecting threats and, more importantly, the identification of those that can be trusted. Consequently, the biases and perceptions of race that officers have affect the decisions based on them which in turn impact their responses. This leads to officers potentially being hypervigilant when race is involved. Race influences significant amygdala activity which is location in the central part of the primitive limbic system. Determining trust is essential to survival which is why it is subject to this

influence [18]. Understanding how to mitigate this "natural" responses, as well as, decisions making in high threat environments is important.

Racial bias in shooting decisions may be explained by the effects of race on perception-based decision making [19-21]. Payne's (2001) study involved participants primed with either a white or black face before they were asked to decide if a visual object was a tool or gun. They discovered that during trials primed with a black face, participants were quicker at responding correctly to a gun and were also more likely to label a tool as a gun. Another part of the study presented pictures of black or white men holding tools or guns and participants were asked to decide whether to press a 'shoot' button or a 'no shoot' button based on if there was a gun or not [22]. Correll et al. (2007) observed that community members and police officers both were faster to shoot black men with guns than white men with guns. The participants were also slower in scenarios which required to a 'not shoot' for black men without guns than white men without guns [19].

Based on a diffusion model, Correll et al. (2015) concluded that information on race made people more likely to misperceive a tool as a gun due to bias affecting ambiguous visual information. In the same shoot/no-shoot approach, it was found that the amygdala had increased functional connectivity with regions in the ventral visual processing stream which is known to be involved in visual object identification [20, 21]. Race information may influence the classification of tools and weapons because of the effect of amygdala activity on visual processing. These results establish a model in which it may be possible to expand the understanding of neural representation of race and how the information is incorporated into decisions within the brain. Research in recent years propose that the neural underpinnings of race, stereotyping, and prejudice and a number of brain regions have been consistently found to

be activated during tasks involving race [23, 24]. These activated regions all include the amygdala. They are the anterior cingulate cortex (ACC), the dorsolateral prefrontal cortex (DLPFC), the fusiform face area (FFA), insula, orbital frontal cortex (OFC), and the anterior temporal lobe (ATL).

Computer simulations aid in understanding law enforcement officers' decisions in high threat situations. They have also been used to investigate the impact of race of the potential offender in making decisions to shoot or not to shoot. There is however a more natural method in virtual scenarios that can aid in assessing decisions to respond in high threat situations. High threat scenarios not only provide opportunities to understand the neural processes during the decision to shoot or not to shoot but also provide the opportunity to study the reflection period after the scenario is completed. These high threat simulations in the virtual firing ranges help to understand the neural and behavioral processes of law enforcement officers.

4.3 Description of the Study

The conducted research aimed to gain insights pertaining to the neural activity in officers while participating in simulated high threat situations. The study involved the use of Standardized Low-Resolution Brain Electromagnetic Tomography (sLORETA) to identify the neural generators for the high threat scenarios. These generators correspond to different functions of the brain and could then be used to understand the capabilities and focus of the officers.

4.3.1 Participants and Procedure

The officers that volunteered for this research included four male local-level LEOs from a small sized city (20,000 to 100,000) located in the Southern U.S. The participants averaged 5.75 years of experience with a standard deviation of 6.75 years. All of the LEOs identified their

race as white and had an average age of 33.5 years (SD = 11.96 years). In order to maintain participant confidentiality, other demographic information was excluded from the study. Participants completed 18 scenarios over three sessions (six scenarios per session), simulating high threat situations using a firearms training system. Each session lasted approximately an hour, and included scenarios which had the intended outcome of the use of deadly force (i.e., the participants firing their service weapon) and ones that did not have deadly force as the intended outcome (i.e., participants should not fire their weapon during the scenario). In the situations that called for deadly force, the suspect directed a weapon at the officer or a bystander, thereby enabling the legal use of deadly force by the officer [25]. Scenarios were presented in random order. EEG data was collected for a 3-minute baseline prior to starting each session (participants sat in a chair in a comfortable position with their eyes closed). Participants were given a 2-minute break in between each scenario. This study focused on analyzing the shoot scenarios that were completed by the participants. The no-shoot scenarios were excluded for each participant for this analysis because, unlike the scenarios that required the participants to discharge their weapon, the LEOs successfully deescalated the situation (i.e., suspect dropped his or her weapon, suspect was not holding a weapon). Therefore, it was not possible to neurologically determine when (or if) the participants made the decision to not shoot. All participants completed nine high stress scenarios; all participants completed the five of the same intended shoot scenarios. The remaining four scenarios were different between the participants. It is possible for a scenario to elicit a different response from participants as compared to another scenario. As such, introducing this confounding variable was not desired. Therefore, the focused of the analyses was on the same scenarios that were completed by all participants.

4.3.2 Facilities and Materials

4.3.2.1 High-fidelity Training System

This experiment was performed at a police training facility equipped with a high-fidelity firearms training system virtual simulator developed by Meggitt Training System. The system is setup in a virtual shooting range, wherein the scenarios are projected onto a wall-size screen located on the far wall. The gun used by the participants was a real handgun (Glock 19) that had been modified to shoot infrared light when the trigger was pulled. In addition, the handgun was affixed with a carbon-dioxide cartridge to give the feel of a real gunshot with recoil when the gun was fired.

4.3.2.2 Video scenario

Each scenario was approximately one to two minutes long. These scenarios presented situations or activities that LEOs could experience during their everyday activities (e.g., DUI traffic stop, hostage situation, etc.). Scenarios analyzed in this study were ones that required the LEOs to utilize deadly force. Before each scenario began, the participants were provided with a brief description of each scenario, but the LEOs were not informed of the intended outcome. The participants did not receive any demographic information or mental health indicators of the suspects prior to the start of the scenario. To better illustrate the content of a scenario, five selected scenarios out of the ones used in the study are described as follows:

Scenario Sample #1, participants were instructed to respond to an active shooter (White male, approximately 30 to 40 years old) at a school. Participants were led to a room in the scenario, where a suspect had taken a teacher hostage.

Scenario Sample #2 involved participants making a traffic stop for a car that ran a stop sign. The passenger (White female, approximately 30 to 40 years old) reaches

in the driver-side console for the registration and pulls out a firearm which she then directs at the participant.

Scenario Sample #3, two LEOs (one LEO is the participant) stop a suspect with a warrant out for their arrest. While the second LEO is talking to the suspect, a child (White female, approximately 8-11 years old) exits the vehicle with a shotgun and points it at the second LEO and then at the participant.

Scenario Sample #4 requires participants to respond to a business establishment where the suspect (White male, approximately 30 to 40 years old) has his former boss taken hostage. The participant is led to a small office where the suspect has a knife that he is waving around while yelling at a bystander.

Scenario Sample #5, participants respond to an aggressive student (White male, teenager) in the school library. The participants arrive at the scene to find the suspect with a knife after he has cornered three other students in the library.

These scenarios were presented in random order for each participant. An example of the setup is shown in Figure 4.1.

Figure 4.1. Example of officer responding through simulation to high threat scenario.

4.3.3 EEG Recording

EEG data was collected using a 64-channel mobile EEG amplifier called EEGO Sports [26]. This unit records data using EEG caps and electrodes connected to a mobile amplifier with

the data saved to a high-performance Windows 8 tablet. This mobile EEG unit is manufactured for use on participants who move frequently (e.g., athletes) to record physiological and neurological data. Nineteen channels were focused on in this study. These channels aimed to target the neural processes of the subject while engaged in firing a weapon [27]. The reference electrode was CPz and the 19 channels utilized were Fp1, Fp2, F7, F3, Pz, F4, F8, T3, C3, Cz, C4, T4, T5, P3, Pz, P4, T7, T8, T6, TP7, TP8, O1 and O2. Data collection took place at a sampling rate of 500 Hz in an ambient temperature room. The measured impedances were all maintained below 20kΩ.

4.3.4 Pre-processing of EEG Data

EEG data was pre-processed in **asa**lab software package [26] and EEGLAB toolbox for MATLAB [28]. The noise was processed with a 30 Hz lowpass filter. Data was then converted to a format compatible with EEGLAB for the remainder of analysis. As in prior studies, there was some participant movement involved [29]. Independent Component Analysis (ICA) was utilized [30] to remove artifacts (e.g., eye blinks, muscle movements). This allowed EEGLAB to identify and reject noisy channels before interpolating them back. This was done to preserve all data leading up to the shot. Data was taken from 30 seconds prior to each shot.

4.3.5 Determining Inputs

Data from the police officer study was initially observed as a whole to determine which areas showed the most activation over the time leading to the shot. For the first set of sessions, the most stimulated area was Brodmann area 18. Other high activation Brodmann areas that were seen through the course of the study were areas 10, 20, and 21. After identifying these areas, a more in-depth analysis was done breaking the 30 second period into sections of 5 seconds in which the activation of each was recorded. To better understand the results, the common

functionality of these Brodmann areas are described in the next few paragraphs. Figure 4.2 can be used as a reference to see the locations of each Brodmann area and the associated cortical function.

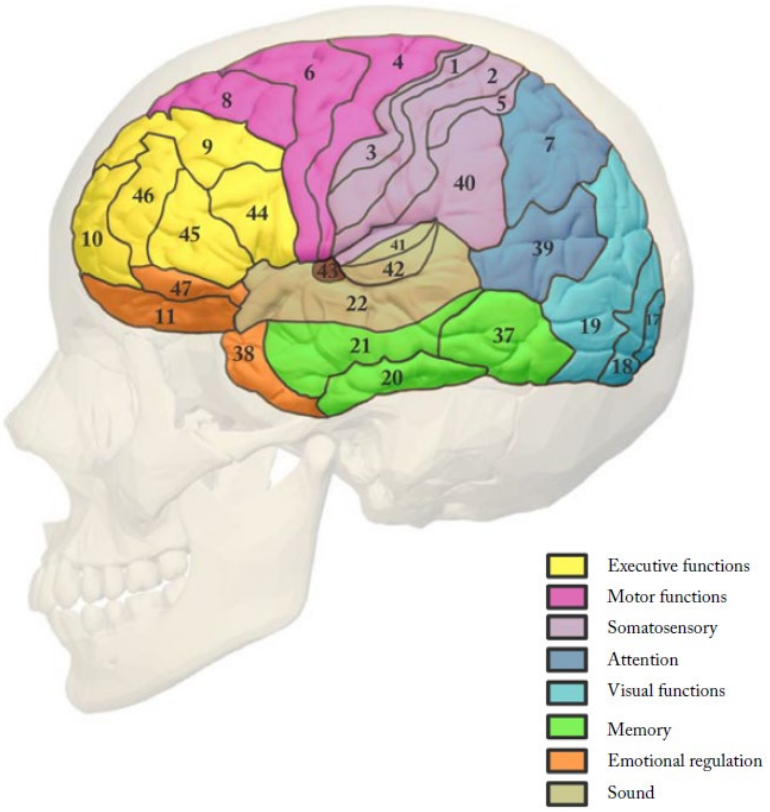

Figure 4.2. Brodmann Cortical Areas [31].

Brodmann areas 9 and 10 are significate in brain operations involving memory. These areas are part of the dorsolated prefrontal cortex. While memory is associated with this Brodmann area including memory encoding, memory retrieval, and working memory, it also includes certain executive functions as well. These include “executive control of behavior”, “inferential reasoning”, and “decision making”. There is a long list of additional processes, but the ones of note to this study include: Error processing/detection, attention to human voices, metaphor comprehension, word-stem completion, and verb generation [31].

The prestriate cortex, Brodmann area 18, is the second major area in the visual cortex, and the first region in the visual association area. The neurons in this area consist of simple visual characteristics such as, orientation, spatial frequency, size, color, and shape. In addition, the V2 cells also respond to various complex shape characteristics, such as the orientation of illusory contours and whether the stimulus is part of the figure or the ground. Brodmann area 18 is also important in object recognition memory. Some more specific associated functions include detection of light intensity, tracking visual motion patterns, discrimination of finger gestures, word and face encoding, and horizonal saccadic eye movements [31].

The next Brodmann area of interest is area 20, the inferior temporal gyrus, which is part of the temporal cortex. This area is associated with high-level visual processing, language understanding, and recognition memory. Other functions worth noting is visual fixation and dual working memory task processing. Lastly, Brodmann area 20 is attributed to intentions to others, also known as theory of mind [31].

Area 21 is located in the middle temporal Gyrus. While its exact function is unclear, it is related to different functions such as processing distance, recognition of known faces, deductive reasoning, observation of motion, processing of complex sounds, and sentence generation. It is also worth noting again that attribution of intentions to others is listed for this Brodmann area as well. Examples of sLORETA data obtained from the officers are shown in Figure 4.3 and Figure 4.4. The data is then organized into excel sheets to prepared to be uploaded into the fuzzy controller [31].

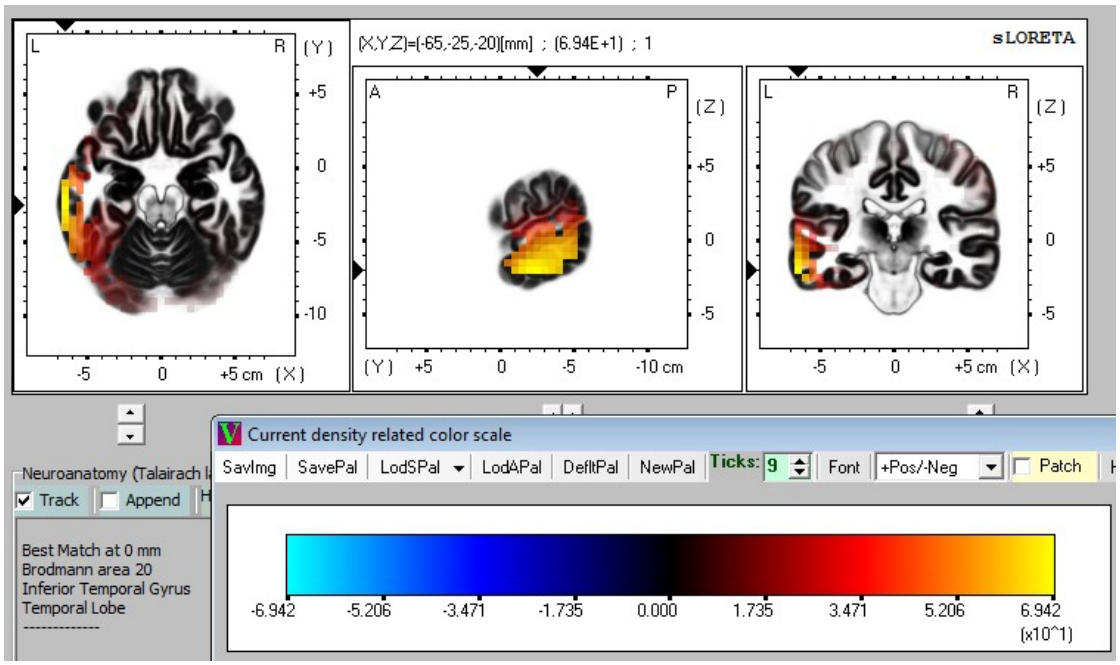

Figure 4.3. Baseline sLORETA (Low Resolution Brain Electromagnetic Tomography) example

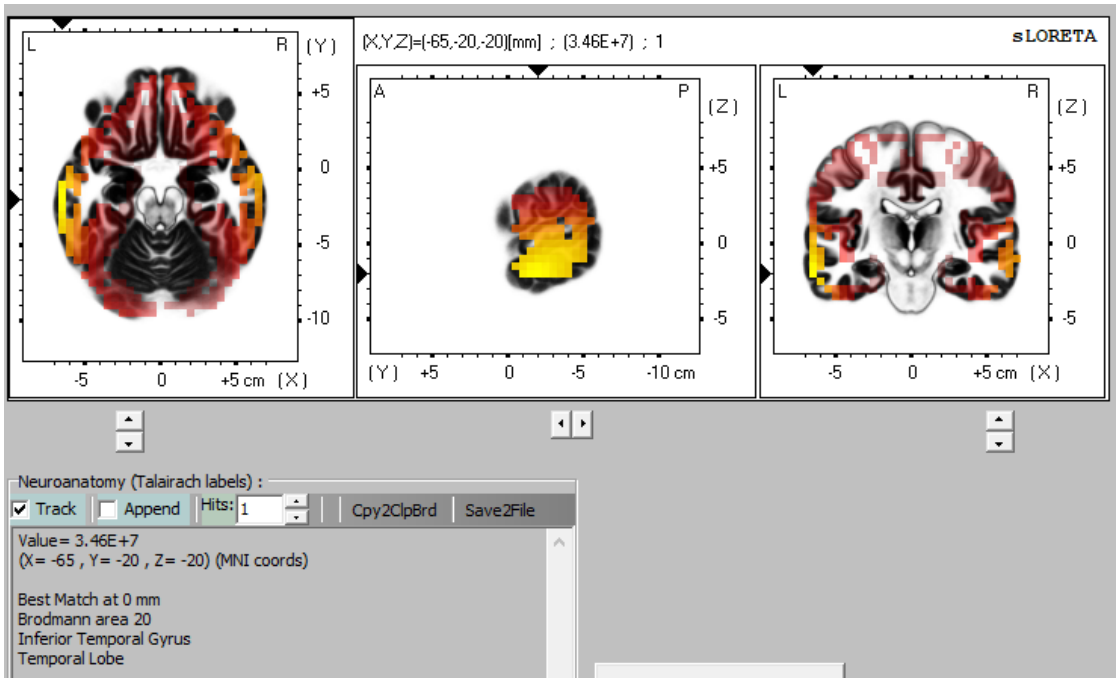

Figure 4.4. sLORETA (Low Resolution Brain Electromagnetic Tomography) example from one officer just prior to decision to shoot (5 second prior to shooting) in high threat situation.

4.4 Fuzzy Controller

The script attached the Appendix E was used to quickly formulate a working fuzzy logic controller. This code loads up all the data in a specified location by the user, and develops parameters based on the data. Some basic rules and input and outputs are already included but additional user inputs can tailor this controller to better suit the given data set. The code begins with a scan of the files in the specified folder. This is followed by code to load all of the data into MATLAB. From there, it is possible to declare which, if any, data sets should be excluded from the full data set analysis. As this controller is able to fit and change based on the data input, it is possible to incrementally update the rule interactions based on new data's average, ranges, and standard deviations. It is also feasible to change the data of interest by removing files from the folder.

Data is organized based on the input parameters that will be put into the controller. The subjects in question are then compared within each other to setup the constraints of the controller. The controller then determines how well each subject preformed within the study compared to other subjects. A membership value will determine how strong the brain activation of the subject was. These values are then processed through the controller with the help of the rules set up. The process utilized in this study for determining these rules came from an understanding of each Brodmann area, as these functions were then mapped to certain outputs shown in Figure 4.5 [31]. Some basic rules for these are coded in, but it is suggested that tunning should be done based on the data to achieve the desired results. It is not necessary to lay out every conceivable rule combination as the controller is designed to utilize efferently what is given.

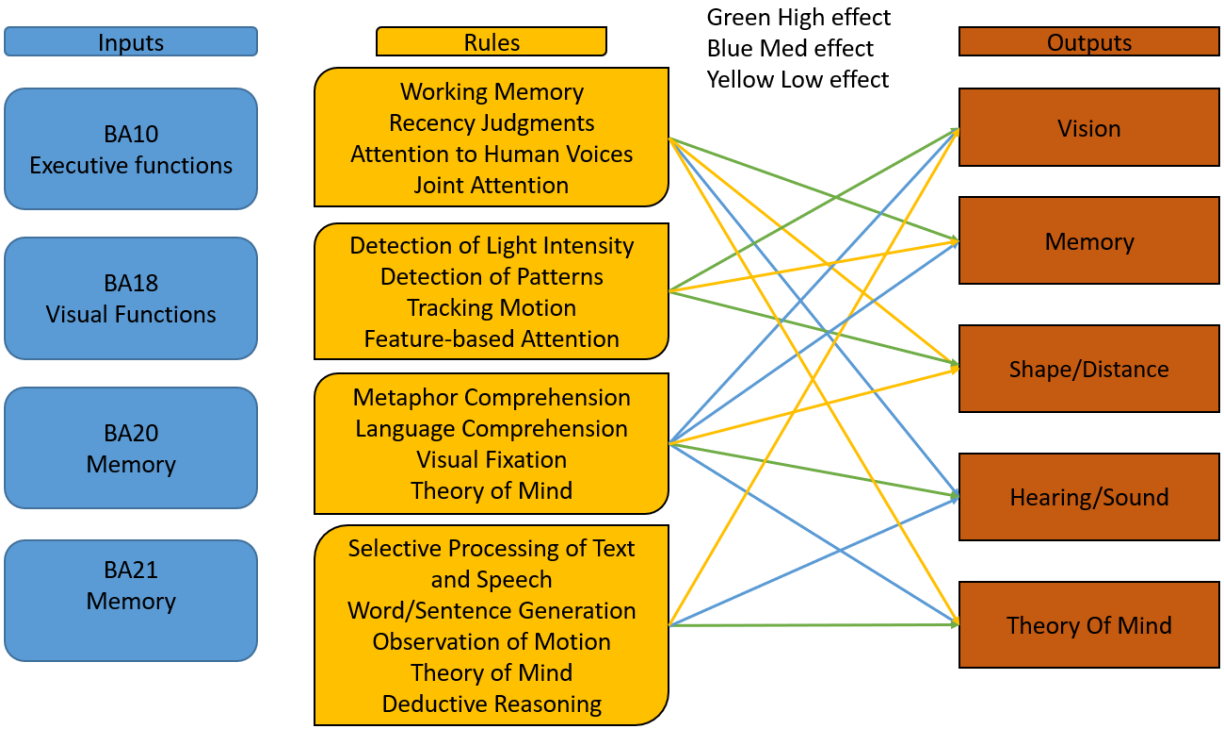

Figure 4.5. Rule generation for the police study.

When the program runs, a window will pop up displaying the current state of the controller, including the inputs and outputs and a box in the middle that contains the rules for the interactions. It is then possible to plot inputs in a 3-D mesh using the surface command under view or to visualize any given input with the rules command in the same area. The 3-D mesh is shown in Figure 4.6.

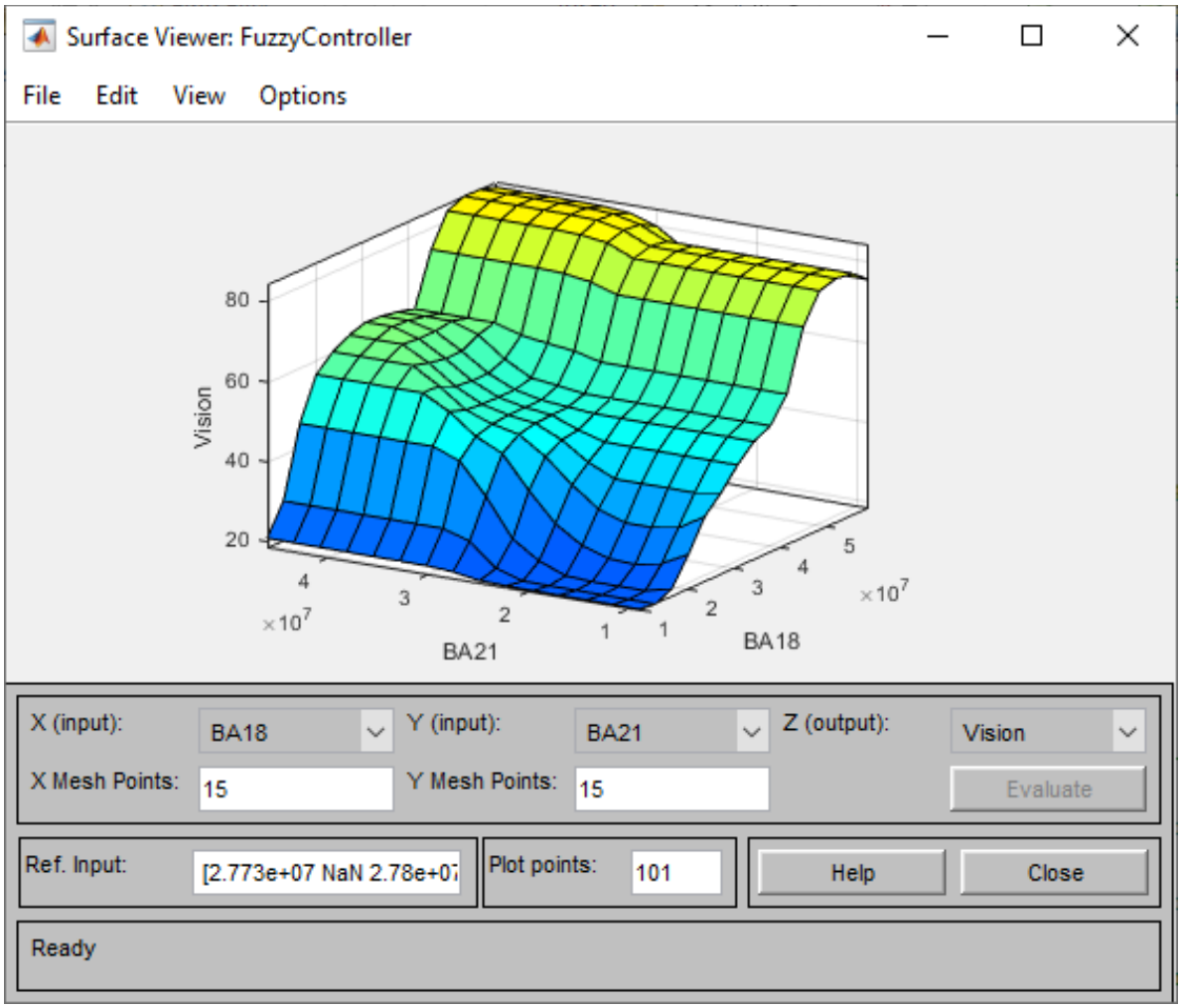

Figure 4.6. Surface Viewer 3-D Mesh for the interaction between Brodmann areas 18 and 21 on vision.

The process of creating rules in the script is shown in Figure 4.5. The script requires a certain syntax for the rules to be added. The "==" sign is used for inputs to show that a particular variable is equal to high, med, or low. "BA18==High => Vision=High" means that when Brodmann area 18 is high then this output vision output is high. There is also coding for "or" and "and" shown by | for or and & for and. In addition, the "~=" sign can be used to show that the output is true when the input is not equal to low. "BA20~=Low & BA18==High => Vision=High" means that when Brodmann 20 is not low and Brodmann area 18 is high then

vision is high. The rule variable is setup with the first rule being *rule(1)*=. The rest of the rules should follow the same format of *rule(end+1)*= which adds the current rule to the end of the rule matrix, which ultimately gets merged into the controller. The interactions are shown in ruler viewer (Figure 4.7).

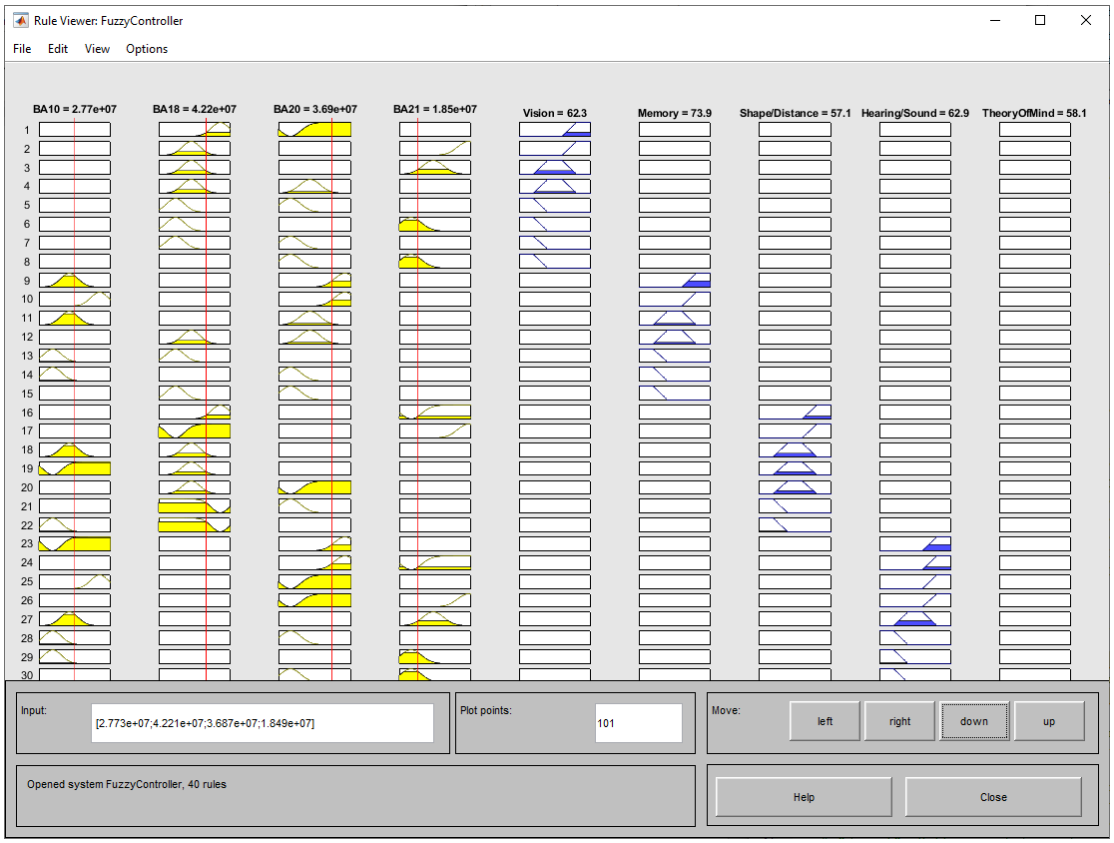

Figure 4.7. MATLAB ruleviewer showing the breakdown of outputs based on rules.

4.5 Study Results

The script automatically processed the rules and calculations for each subject and session. A heat map was created which displayed the full output (Figure 4.8). To offset the fact that defuzzification with the centroid method never gives a score of 100, the data was normalized from 1 to 100. The script was ran and data analyzed incrementally which allowed the rules to be

evaluated and tuned each iteration for more accurate results. The script also displays z-score data based on local and global data as shown in Figure 4.9 and Figure 4.10.

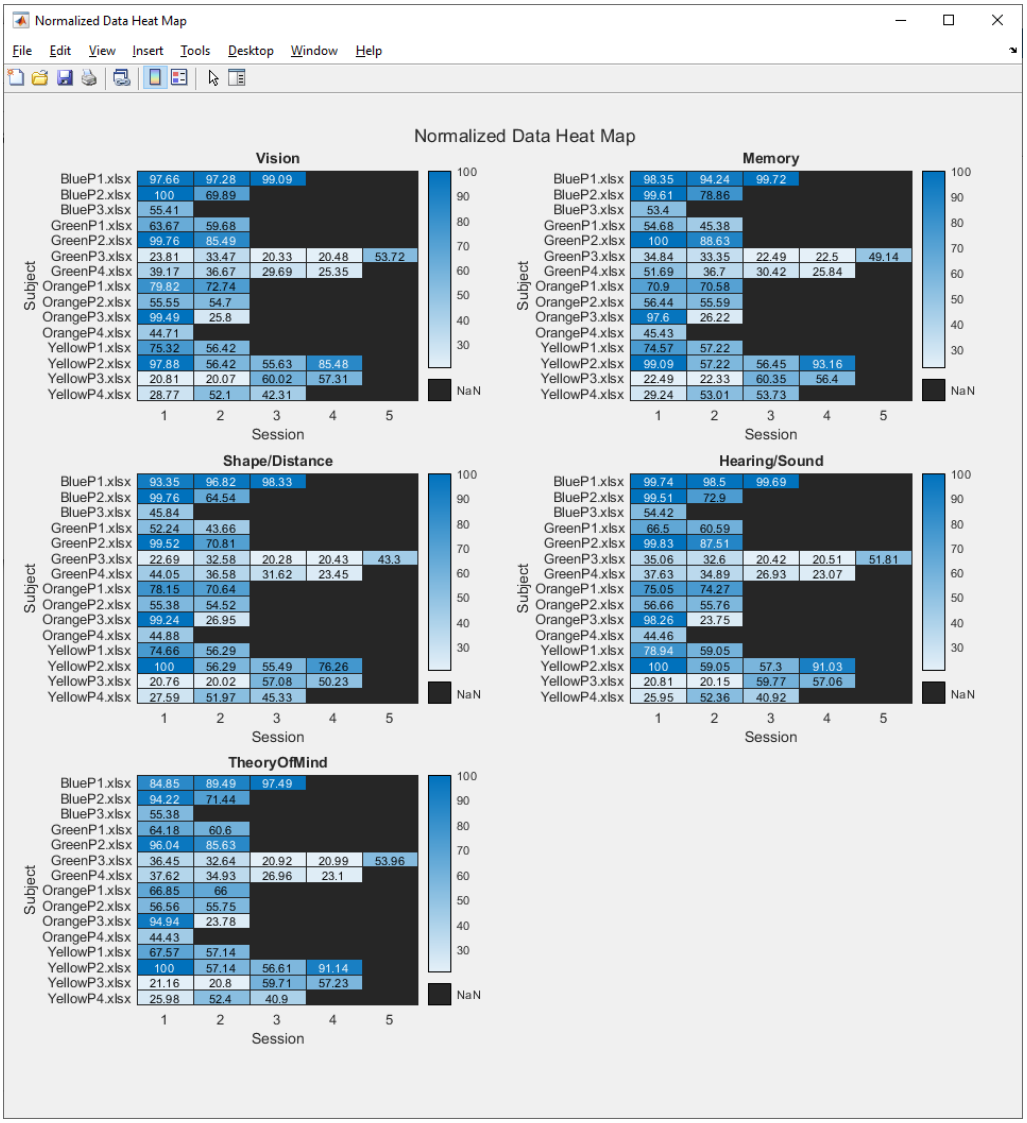

Figure 4.8. Normalized Output Data Heatmap.

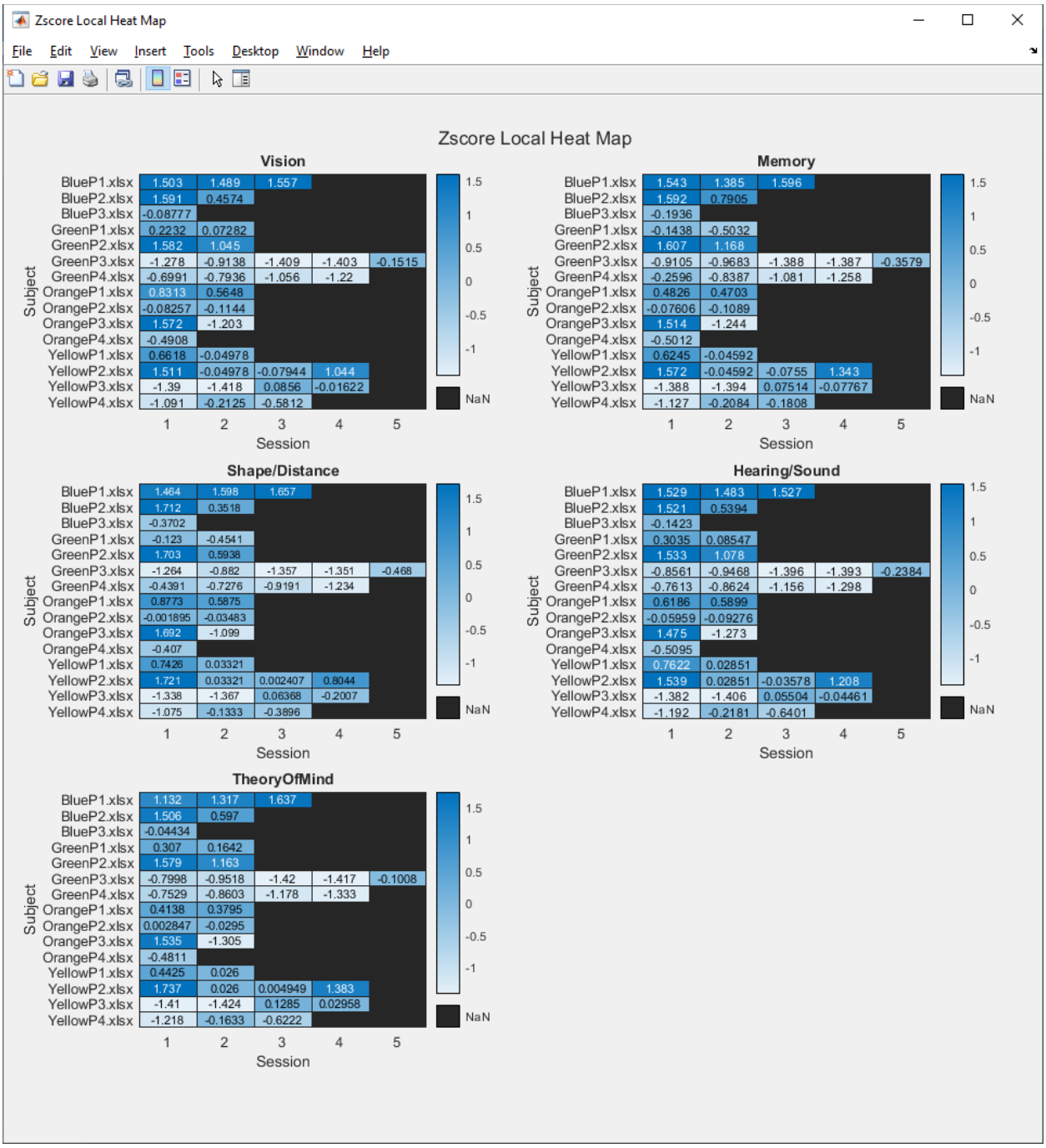

Figure 4.9. Z-score Local Output Data Heatmap.

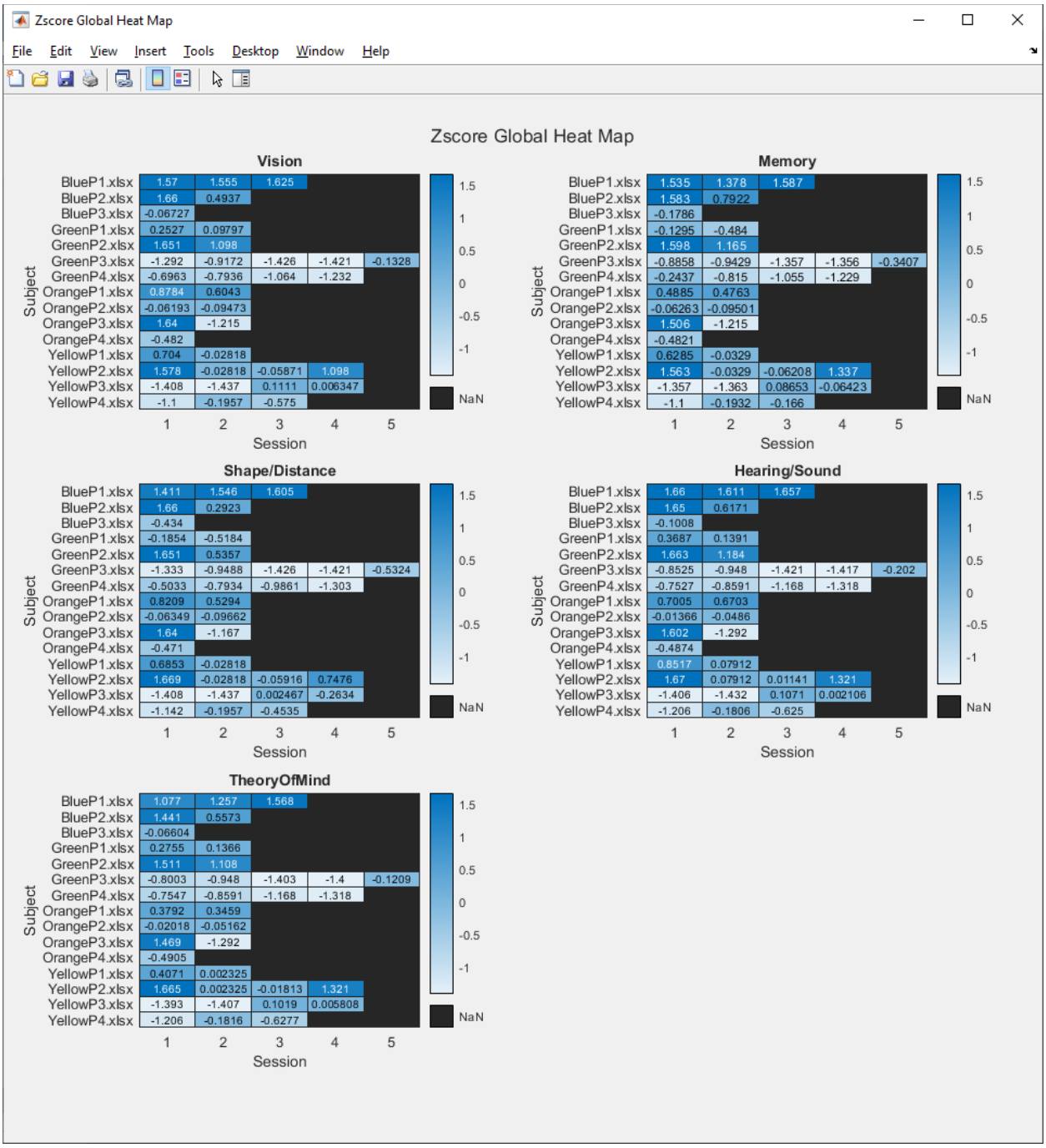

Figure 4.10. Z-score Global Output Data Heatmap.

An additional analysis utilized the outputs as data points in a 5th dimensional space. This allows for the distance formula to calculate how different (far away) or similar (close) the datapoints are. The results of this analysis are presented in a heatmap which is used to show the distance between each session (Figure 4.11). The darker colors indicate data that is further apart, and the lighter colors represent data that shares a similar area in 5th dimensional space, representing sameness in the datapoints.

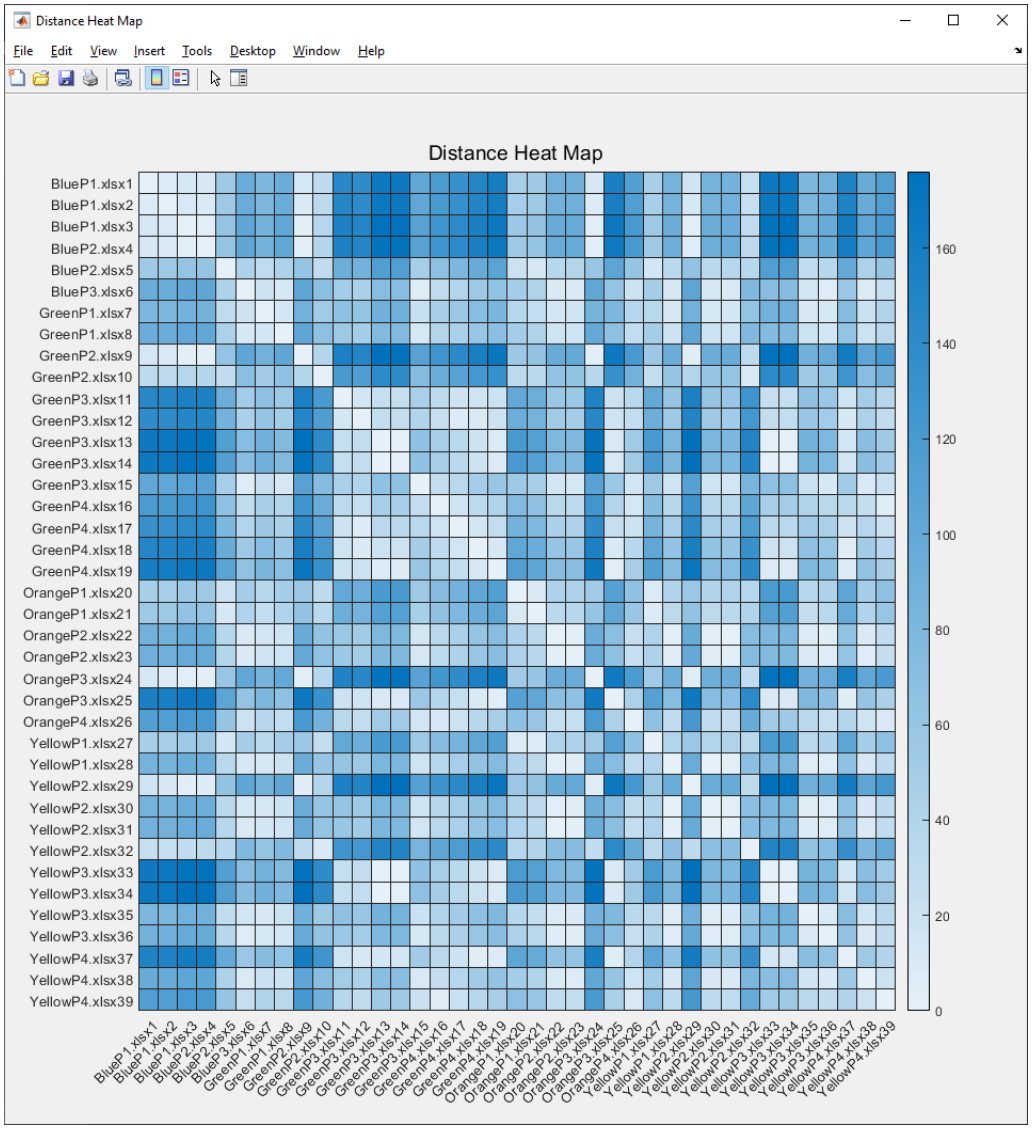

Figure 4.11. Distance Heat Map Data.

The data presented thus far has been a subset of the collected data in the study. Due to the nature of the way the inputs are setup, data with a standard deviation greater than two would skew the data making it more difficult to analyze the rest of the data. While the data variations can still be seen in the smaller data, the vastly larger data skews the results by shifting the average and standard deviation much higher than manageable. Figure 4.12 shows the z-score local data for the complete set of data. All datapoints that were over two standard deviations were ultimately removed.

Figure 4.12. Complete Local Z-scores for Police Data.

The complete normalized data heat map can be seen in Figure 4.13. For all of the Police data, the scenarios were organized by participant, and then by color. Each color represents a certain scenario type. Blue scenarios are road stops. Orange scenarios are combat operations involving SWAT or hostages. Green relates to routine and welfare checks. The last is Yellow, representing threat responses to a location such as a public library or a place of work.

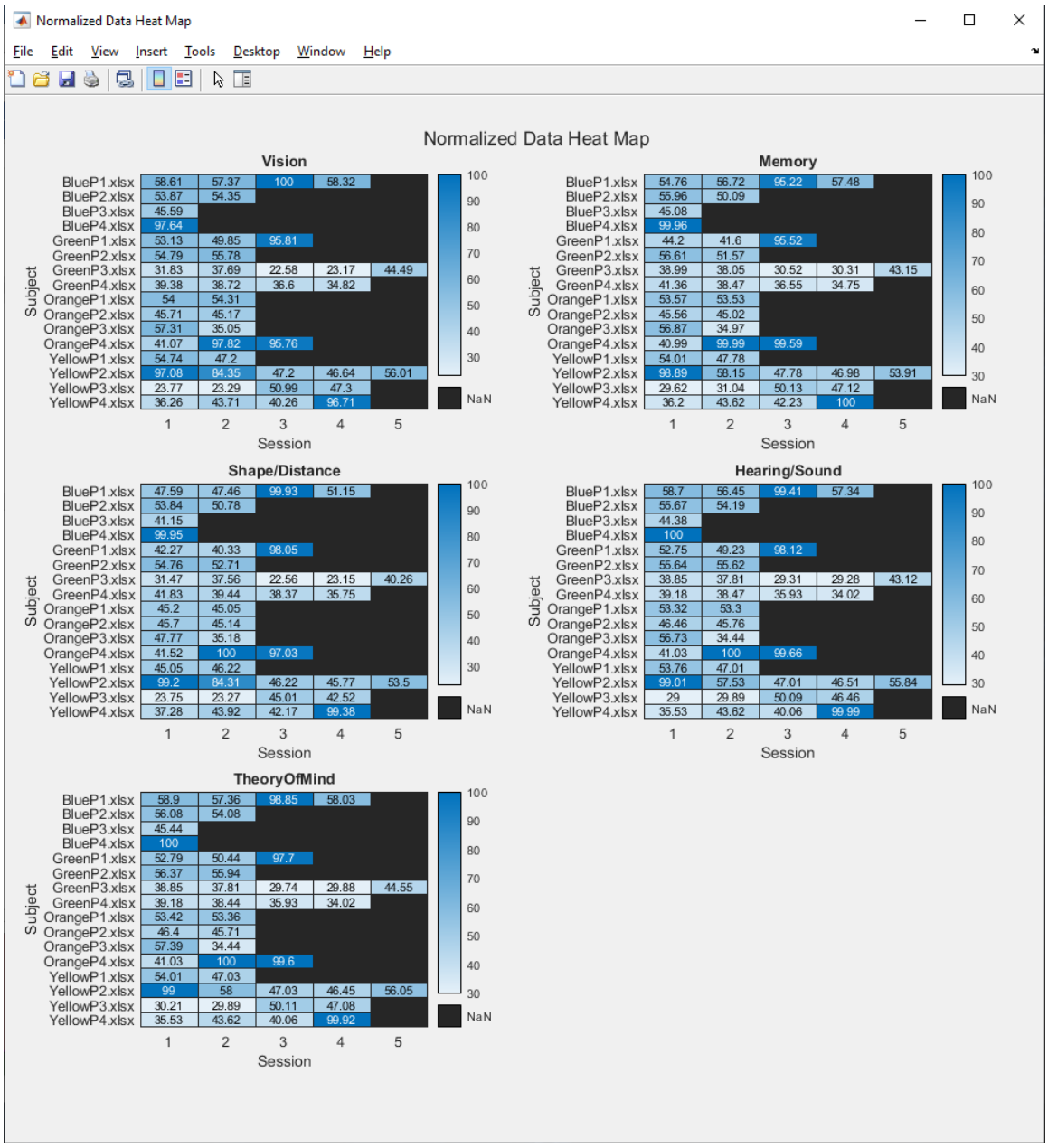

Figure 4.13. Complete Normalized Police Heat Map Data.

4.5.1 Discussion of Attained Results

The conducted analysis yielded some interesting insights about the EEG data. Officers P1 and P2 had less than five years of experience, P3 and P4 had more than five. The less experienced officers, in most cases, showed more activation in their Broadman areas. While they still in training, they are actively growing and developing as officers, requiring more thought and brainpower. Participant 2 had lower Shape/Distance scores compared to the rest of the analyzed outputs. This observation is indicative of the first use for this controller, which is dedicated object-based training would aid in the growth of this officer.

Participant 2 had a greater activation in in their first blue scenario. This scenario could become a good scenario for this officer to practice maintaining a heighted state of activation. This logic can also be used to find other scenarios that stimulate brain activity outside of the ones completed by Participant 2. Certain scenarios elicit greater excitation in certain areas of the brain. In this way, it would be possible to rank scenarios based how well they target a certain area of the brain or how well they target a certain skill set. For example, yellow and blue scenarios seemed to favor Hearing/Sound more so than the orange and green scenarios. Officers can then analyze themselves and determine which scenarios would best aid their growth.

To summarize, the collected police data can be utilized for personal evaluation, in order to see how each officer measures up to others. Identifying which types of scenarios target which parts of the brain would aid in further skill set training. This would enable officers to determine scenarios that would be worth reviewing or replaying to train areas of their brain. Lastly, the ability to compare and pair partners for training or mentoring. As more data is added over training sessions, the numbers should become more discrete, and trends easier to see and interpret. These results will be discussed further in Chapter 5.

Chapter 4 References

[1] FBI. (2016). *Federal Bureau of Investigation (2016)*. Available: https://ucr.fbi.gov/crime-in-the-u.s/2016/crime-in-the-u.s.-2016

[2] J. R. Oliva, R. Morgan, and M. T. Compton, "A practical overview of de-escalation skills in law enforcement: Helping individuals in crisis while reducing police liability and injury," *Journal of Police Crisis Negotiations,* vol. 10, no. 1-2, pp. 15-29, 2010.

[3] B. Vila, L. James, S. M. James, and L. B. Waggoner, "Developing a common metric for evaluating police performance in deadly force situations," WASHINGTON STATE UNIV PULLMAN2012.

[4] D. Banks, P. Ruddle, E. Kennedy, and M. G. Planty, *Arrest-related deaths program redesign study, 2015-16: Preliminary findings*. Department of Justice, Office of Justice Programs, Bureau of Justice Statistics, 2016.

[5] E. Bittner, *The functions of the police in modern society: A review of background factors, current practices, and possible role models* (no. 2059). National Institute of Mental Health, Center for Studies of Crime and Delinquency, 1970.

[6] A. J. Haufler, T. W. Spalding, D. Santa Maria, and B. D. Hatfield, "Neuro-cognitive activity during a self-paced visuospatial task: comparative EEG profiles in marksmen and novice shooters," *Biological psychology,* vol. 53, no. 2-3, pp. 131-160, 2000.

[7] S. L. Stewart and J. A. Kirkham, "Predictors of individual differences in emerging adult theory of mind," *Emerging Adulthood,* p. 2167696820926300, 2020.

[8] H. Wald, J. Borkan, J. Taylor, D. Anthony, and P. Shmuel, "Fostering and evaluating reflective capacity in medical education: development of the REFLECT (Reflection Evaluation for Learners' Enhanced Competencies Tool) rubric for evaluating students' reflective writing," *Acad Med,* vol. 87, no. 1, pp. 41-50, 2012.

[9] M. A. Jennifer, "Handbook of reflective and experiential learning," *Theory and Practice, Londres y Nueva York: Routledge Falmer,* 2004.

[10] E. S. Berner and M. L. Graber, "Overconfidence as a cause of diagnostic error in medicine," *The American journal of medicine,* vol. 121, no. 5, pp. S2-S23, 2008.

[11] S. Mamede, H. G. Schmidt, and J. C. Penaforte, "Effects of reflective practice on the accuracy of medical diagnoses," *Medical education,* vol. 42, no. 5, pp. 468-475, 2008.

[12] S. Mamede and H. G. Schmidt, "The structure of reflective practice in medicine," *Medical education,* vol. 38, no. 12, pp. 1302-1308, 2004.

[13] P. Tremayne and R. J. Barry, "Elite pistol shooters: physiological patterning of best vs. worst shots," *International journal of psychophysiology,* vol. 41, no. 1, pp. 19-29, 2001.

[14] E. M. Bowden, M. Jung-Beeman, J. Fleck, and J. Kounios, "New approaches to demystifying insight," *Trends in cognitive sciences,* vol. 9, no. 7, pp. 322-328, 2005.

[15] J. Kounios *et al.*, "The prepared mind: Neural activity prior to problem presentation predicts subsequent solution by sudden insight," *Psychological science,* vol. 17, no. 10, pp. 882-890, 2006.

[16] A. Dietrich and R. Kanso, "A review of EEG, ERP, and neuroimaging studies of creativity and insight," *Psychological bulletin,* vol. 136, no. 5, p. 822, 2010.

[17] S. Sandkühler and J. Bhattacharya, "Deconstructing insight: EEG correlates of insightful problem solving," *PLoS one,* vol. 3, no. 1, 2008.

[18] C. M. Halliburton, "Race, Brain Science, and Critical Decision-Making in the Context of Constitutional Criminal Procedure," *Gonz. L. Rev.,* vol. 47, p. 319, 2011.

[19] J. Correll, B. Park, C. M. Judd, B. Wittenbrink, M. S. Sadler, and T. Keesee, "Across the thin blue line: police officers and racial bias in the decision to shoot," *Journal of personality and social psychology,* vol. 92, no. 6, p. 1006, 2007.

[20] J. Correll, B. Wittenbrink, M. T. Crawford, and M. S. Sadler, "Stereotypic vision: How stereotypes disambiguate visual stimuli," *Journal of personality and social psychology,* vol. 108, no. 2, p. 219, 2015.

[21] K. B. Senholzi, B. E. Depue, J. Correll, M. T. Banich, and T. A. Ito, "Brain activation underlying threat detection to targets of different races," *Social neuroscience,* vol. 10, no. 6, pp. 651-662, 2015.

[22] B. K. Payne, "Prejudice and perception: the role of automatic and controlled processes in misperceiving a weapon," *Journal of personality and social psychology,* vol. 81, no. 2, p. 181, 2001.

[23] D. M. Amodio, "The neuroscience of prejudice and stereotyping," *Nature Reviews Neuroscience,* vol. 15, no. 10, pp. 670-682, 2014.

[24] J. T. Kubota, M. R. Banaji, and E. A. Phelps, "The neuroscience of race," *Nature neuroscience,* vol. 15, no. 7, p. 940, 2012.

[25] R. E. Broomé, "An empathetic psychological perspective of police deadly force training," *Journal of Phenomenological Psychology,* vol. 42, no. 2, pp. 137-156, 2011.

[26] F. Zanow and T. R. Knösche, "Asa-advanced source analysis of continuous and event-related eeg/meg signals," *Brain topography,* vol. 16, no. 4, pp. 287-290, 2004.

[27] C. A. Domingues *et al.*, "Alpha absolute power: motor learning of practical pistol shooting," *Arquivos de neuro-psiquiatria,* vol. 66, no. 2B, pp. 336-340, 2008.

[28] A. Delorme and S. Makeig, "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *Journal of neuroscience methods,* vol. 134, no. 1, pp. 9-21, 2004.

[29] T. Thompson, T. Steffert, T. Ros, J. Leach, and J. Gruzelier, "EEG applications for sport and performance," *Methods,* vol. 45, no. 4, pp. 279-288, 2008.

[30] A. Delorme, T. Sejnowski, and S. Makeig, "Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis," *Neuroimage,* vol. 34, no. 4, pp. 1443-1449, 2007.

[31] Trans Cranial Technologies (2012). Cortical Functions. Hong Kong: Trans Cranial Technologies.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

5.1 Discussion of Research Results

Chapter 5 discusses the conclusions drawn from this research, along with insights gained through the process. The contributions to Electroencephalography (EEG) data analysis are also discussed, giving an explanation of ways to enhance or expedite the analysis of similar studies. The final section of the Chapter discusses recommendations for future research.

The understanding gained from this dissertation work will allow for future studies to target and stimulate regions of the brain for mental acuity in a wide array of tasks. This, along with the law enforcement study, leads to a better understanding of the neural processes in the cognitive and ethical decision-making process, and allows for development while training on patrol and in the classroom. Understanding transcranial direct current stimulation (tDCS) is the beginning of future applications and advancements in mental focus, which can be applied to many other real-life applications.

5.1.1 Discussion of Math Study Results

The math study involved the analysis of tDCS data, thus, the points of interest in this discussion will focus on how that stimulation affected the activation of the brain and how the subjects reacted during the session. For a streamlined discussion, rankings of the outputs will be utilized (shown in Figure 5.1). The data that led to these rankings is present in Chapter 3. The numbers in the chart indicate the rank of that particular output. A score of five in *calculations*

means that among all of the five outputs, *calculations* was the lowest. Conversely, a one indicates that this output was the highest, and therefore, the most activated area during the session.

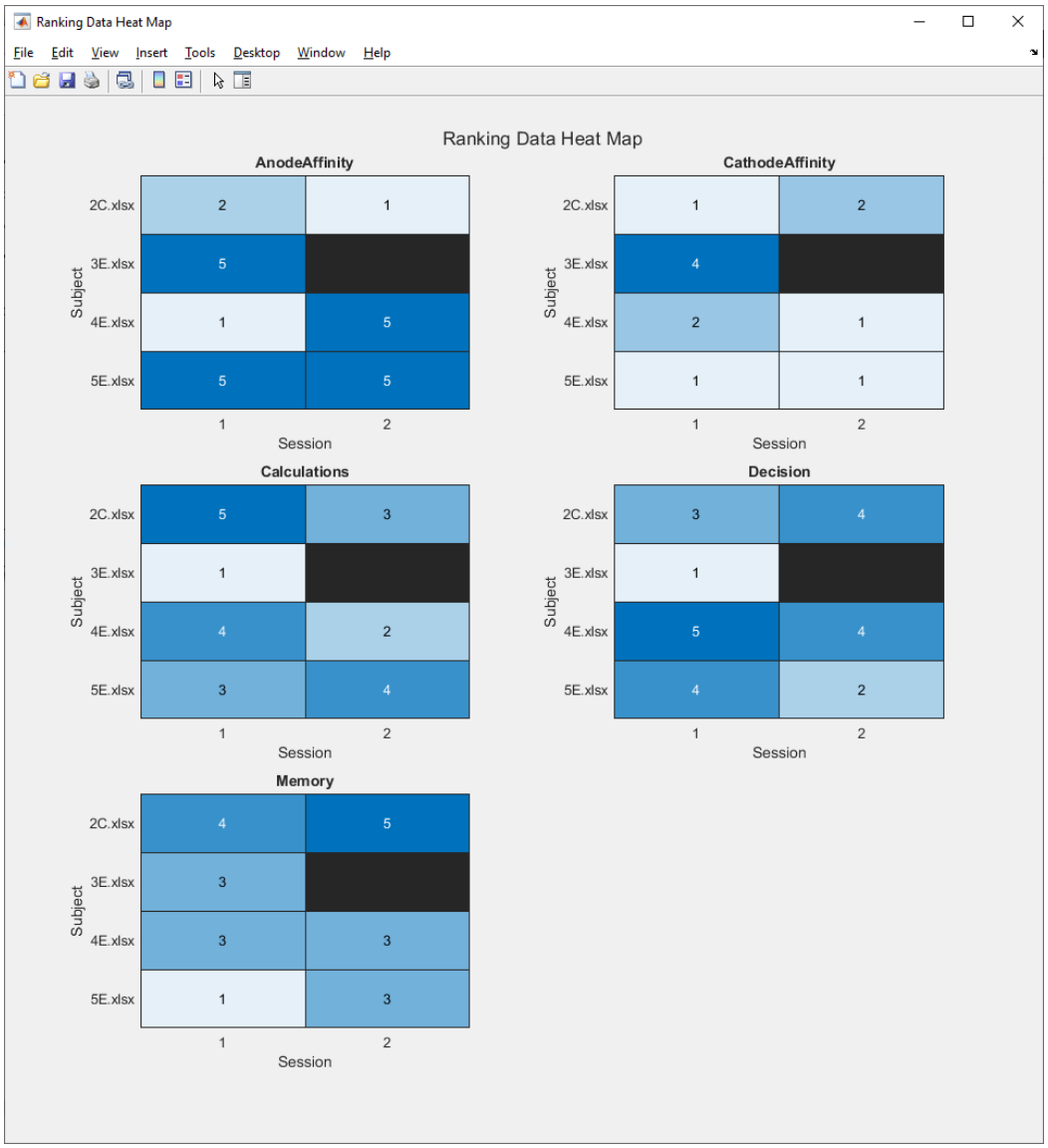

Figure 5.1 Ranking of Math Study Data.

In Figure 5.1, Subject 1C was excluded in the heatmap made automatically with MATLAB as this subject was skewing the data. Subject 1C's rankings are displayed in Table 5.1, along with the other subjects. Subject 1C was a control subject and therefore did not receive

an application of tDCS. In session one, Subject 1C showed the most activation in the cathode region of the brain. While the subject did not receive any stimulation, this still indicates it was a region of interest in this particular session. As for the rest of the subject's outputs, *memory* and *calculations* over decision making were favored. This indicates that Subject 1C focused on remembering the skills taught and applying them. The need for *decision making* was lower due to the level of consideration in determining the final answer of the given math problem or the approach along the way.

Table 5.1. Session 1 Ranking

Session 1						
Subject	Type	AnodeAffinity	CathodeAffinity	Calculations	Decision	Memory
1C	Control	2	1	4	5	3
2C	Control	2	1	5	3	4
3E	Experiment	5	4	1	1	3
4E	Experiment	1	2	4	5	3
5E	Experiment	5	1	3	4	1

Subject 2C was also a control subject. This subject's *cathode* and *anode affinity* mirrored that of the other control subject. This indicates that without outside intervention, these areas of the brain experienced the most activation. The subject acted differently than the other control subject though. Subject 2C favored *decision making* and *memory* over *calculations*. This indicates that the subject focused more on the accuracy of their approach over the actual calculations. This could signal that the subject was discerning how to best guess the answer.

Subject 3E was the first experiment subject. The benefits of tDCS will be analyzed for the experimental subjects. The activation in the subject's brain showed favor to the cathode over the anode. Since the rankings of these nodes were lower than the rest, this means the activation was stimulated elsewhere in the brain. The *calculations* and *decision making* were tied for

highest with *memory* coming in third. This indicates that less activation was needed for *memory* as *decision making* and *calculations* were already underway.

Subject 4E was also an experimental subject. In the subject's application of tDCS, the subject favored the anode over the cathode. The rest of the subject's outputs mirrored Subject 1C, favoring *memory* and *calculations* over *decision making*. Again, focusing on remembering the skills taught and applying them, rather than the decision-making process, allowed for more consideration in calculating the final answer of the problem in question.

Subject 5E belonged to the experimental group as well. This was the first to strongly favor the cathode over the anode, ranking first in *cathode affinity* and last in *anode affinity*. Favoring the anode is an important discovery. Subject 5E can use that knowledge to aid in further tasks where stimulation is available. Subject 5E also ranked first with *memory,* meaning that the stimulation aided in remembering how to do the calculations. *Decision making* was the lowest meaning that the subject felt confident in the approach chosen. Session two involved a different math problem, but the experimental and control subjects remained the same. Subject 3E did not return for his second session. The data analyzed is presented in Table 5.2, along with the changes between the sessions in Table 5.3. The changes are represented with a + or a – if there was a change of at least 5%. If the change was less than 5%, then ~= is indicated in the table.

Table 5.2. Session 2 Ranking

Session 2						
Subject	Type	AnodeAffinity	CathodeAffinity	Calculations	Decision	Memory
1C	Control	4	5	1	1	1
2C	Control	1	2	3	4	5
4E	Experiment	5	1	2	4	3
5E	Experiment	5	1	4	2	3

Table 5.3. Change between session (+ for 5%, - for -5%, ~= for <5%)

Change						
Subject	Type	AnodeAffinity	CathodeAffinity	Calculations	Decision	Memory
1C	Control	-	-	+	+	+
2C	Control	-	-	+	+	+
4E	Experiment	~=	~=	~=	+	~=
5E	Experiment	+	~=	-	-	-

Subject 1C saw some big changes in the second session, maxing out *calculations*, *decision making*, and *memory*. The *cathode affinity* and *anode affinity* activations both went down. The placebo effect from the first session might have worn off, since the control subjects were made to believe they received tDCS as well. It is also worth noting that the rankings of the *anode affinity* and *cathode affinity* swapped for this subject. This data was ultimately removed from the overall analysis since the activation was over two standard deviations above the average. This could be due to too much body movement or other types of "noise" that could not be accounted for with the filtering in place.

The next participant was Subject 2C which saw similar changes to Subject 1C. This again could be the brain becoming "wise" to the sham tDCS setting the subject was given, thus removing the placebo effect. All of the other outputs went up from the first session with *calculations* and *decision* taking the leading roles, indicating that *memory* was not needed for this session. Like Subject 1C, the *anode affinity* and *cathode affinity* swapped ranking order for Subject 2C.

Subject 4E seemed to attune to the stimulation more for this session, showing similar ranking to Subject 5E in session one. The *cathode affinity* ranking went up to first while the anode went down to last, but the actual change within the brain was less than 5%. Subject 4E remained within 5% for most of their outputs besides *decision,* which saw an increase.

Calculations and *memory* still retained the higher ranks showing focus on working the problem and not decisions related to the problem.

The last analyzed participant, Subject 5E, maintained affinity for the anode, and also experienced an increased activation in that region. The *cathode affinity* saw less than 5% change. The other outputs saw an overall decrease in activation with *decision* becoming the top output followed by *memory* and *calculations*. More thought went into the accuracy of the approach rather than the calculations themselves. This could indicate that the subject had to make a logical guess.

Overall, this study shows some interesting insights on the changes between the sessions and how interactions within the brain can lead to insights about the subjects themselves. The affinity for cathode and anode seemed to attune to the subjects between the sessions, which aligns with the knowledge gained in the literature review.

5.1.2 Discussion of Police Study Results

The police study was broken into four different data categories. These included road stops (blue); routine and welfare checks (green); combat operations (orange); and threat response (yellow) (Table 5.4 through Table 5.7) The presented tables all follow the same format, where a score of one indicates a high brain activity, and a score of five represents the lowest observed brain activity. Subjects P1 and P2 were the lesser experienced officer while Subjects P3 and P4 were the veteran officers.

Table 5.4 Road Stop Scenario Rankings

Road Stops (Blue)						
Subject	Type	Vision	Memory	Shape/Distance	Hearing/Sound	TheoryOfMind
P1	< Five Years	2.67	2.33	3.67	1.33	5.00
P2	< Five Years	2.50	2.00	3.50	3.00	4.00
P3	> Five Years	1.00	4.00	5.00	3.00	2.00

Table 5.5 Routine and Welfare Checks Scenario Rankings

Routine and Welfare Checks (Green)						
Subject	Type	Vision	Memory	Shape/Distance	Hearing/Sound	TheoryOfMind
P1	< Five Years	3.00	4.00	5.00	1.50	1.50
P2	< Five Years	3.50	1.00	4.50	2.00	4.00
P3	> Five Years	3.00	2.20	5.00	3.00	1.80
P4	> Five Years	2.50	1.25	2.25	4.75	4.25

Table 5.6 Combat Operations Scenario Rankings

Combate Opertaions (Orange)						
Subject	Type	Vision	Memory	Shape/Distance	Hearing/Sound	TheoryOfMind
P1	< Five Years	1.50	4.00	2.50	2.00	5.00
P2	< Five Years	4.00	3.00	5.00	1.00	2.00
P3	> Five Years	2.00	3.00	1.50	4.00	4.50
P4	> Five Years	3.00	1.00	2.00	4.00	5.00

Table 5.7 Threat Response Scenario Rankings

Threat Response (Yellow)						
Subject	Type	Vision	Memory	Shape/Distance	Hearing/Sound	TheoryOfMind
P1	< Five Years	3.00	3.00	4.00	1.00	4.00
P2	< Five Years	4.25	2.50	4.00	1.50	2.00
P3	> Five Years	2.50	1.75	5.00	3.25	2.50
P4	> Five Years	3.00	1.00	3.33	4.00	3.67

Each scenario had five calculated outputs. These included *vision*, *memory*, *shape/distance*, *hearing/sound* and *theory of mind*. These values were all calculated individually for each scenario (Figure 5.2). Officer P4 did not participate in any scenario that would be considered a road stop, so BlueP4 has been excluded from the data. The officers competed a considerably high number of studies to be *analyzed* individually, thus, the average ranking for

each output and each category were computed in (Figure 5.3). These values were then organized into Table 5.4 through Table 5.7. The results of the road stops are discussed first starting with the lesser experienced officers. Then, the results obtained from the more experience officers are presented.

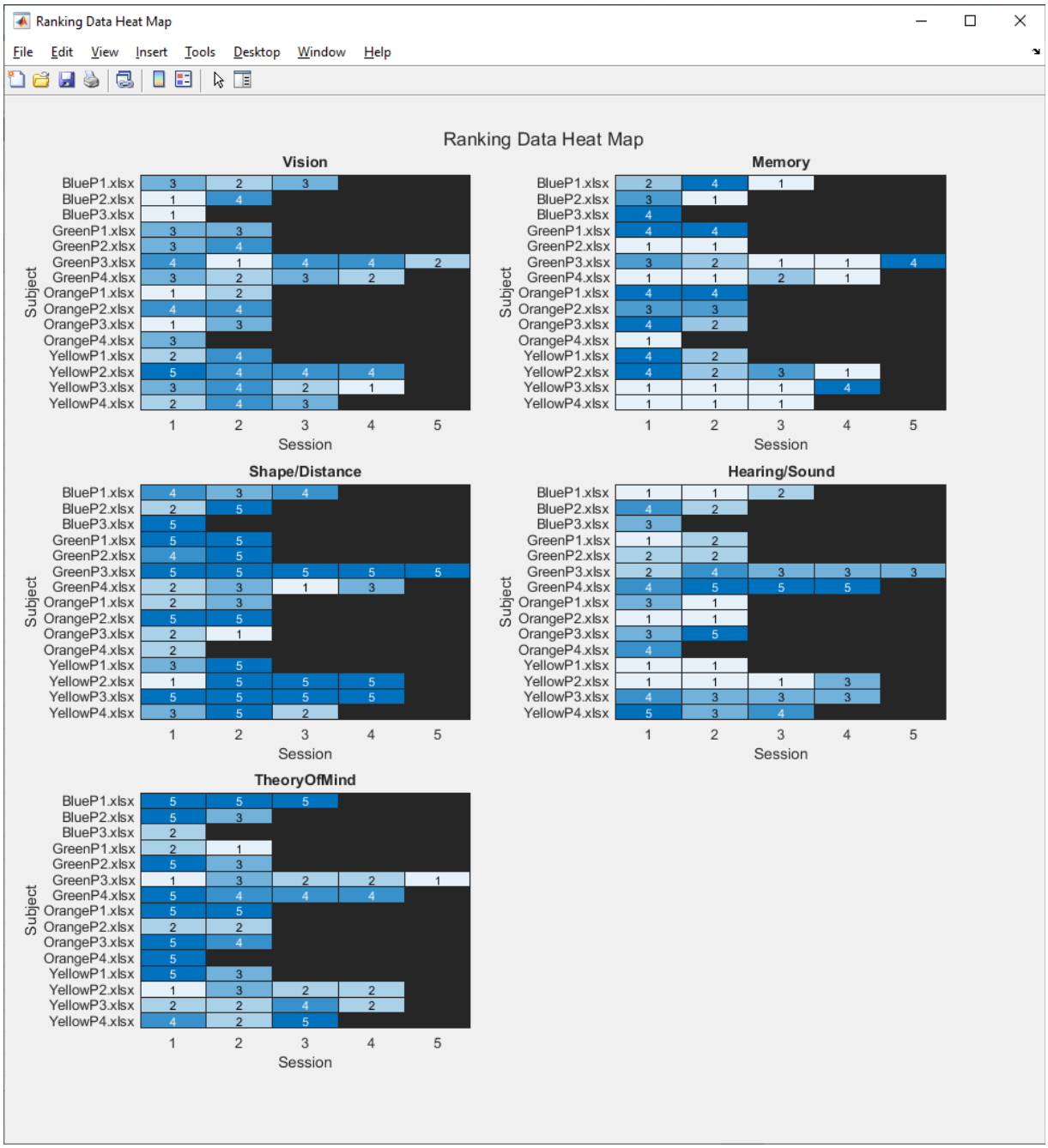

Figure 5.2 Ranking Data Heat Map

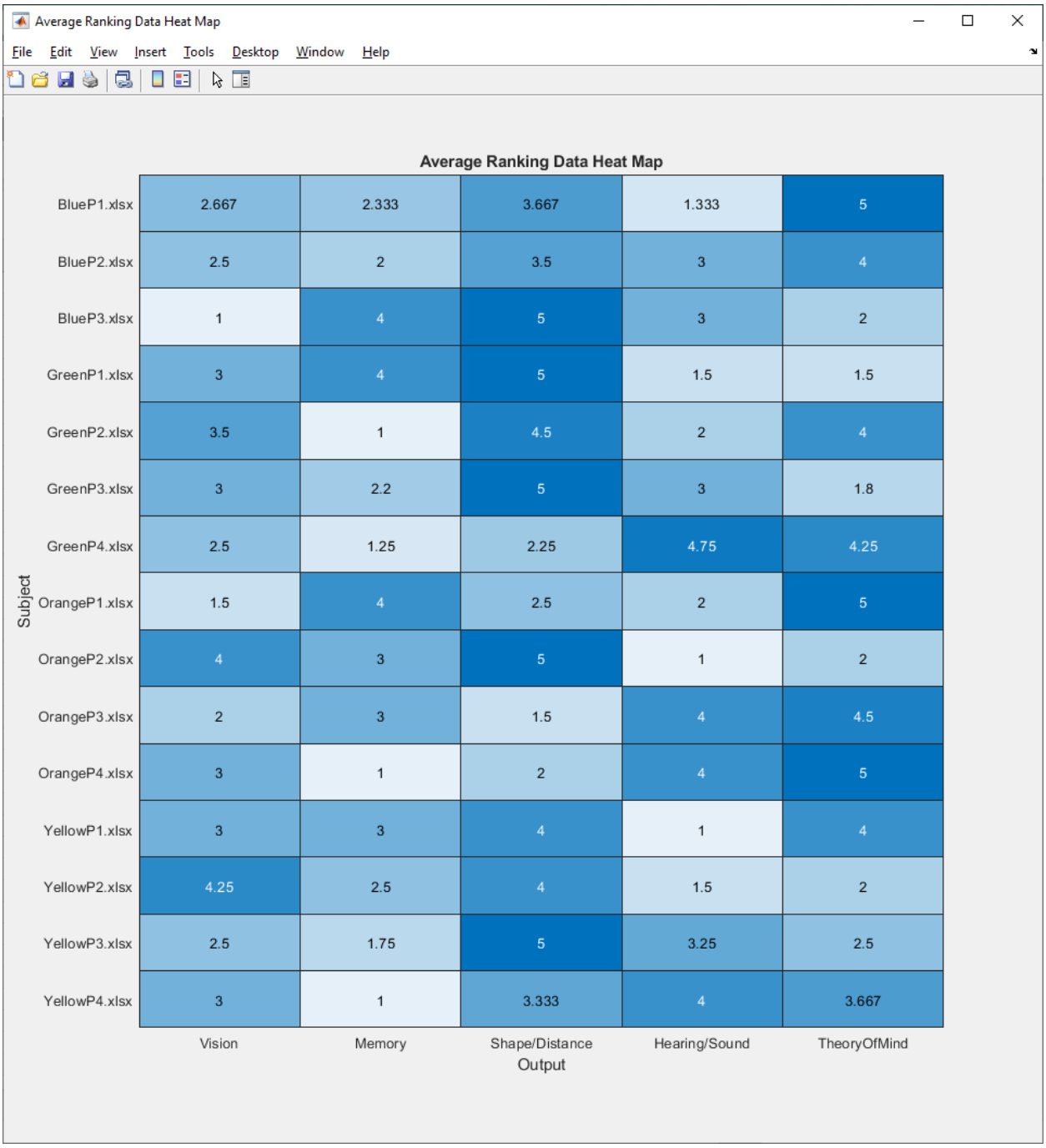

Figure 5.3 Average Ranking Data Heat Map

Officer P1 was a lesser experienced officer. In this study, he completed three road stop scenarios. During these scenarios, the most activated areas of Officer P1's brain was related to *hearing/sound*. This was followed by *memory*, which indicates that the officer chose to listen and correlate sounds during the road stops. The next outputs were *vision* and *shape/distance*. These outputs convey that the officer was still aware of their surroundings, but his *theory of mind* score shows that they failed to consider what the civilian was thinking or planning.

Officer P2 was also a lesser experienced officer. For the two road stop scenarios completed, *memory* was the highest. *Vision* was the second highest meaning that the officer was trying to remember what he was seeing or what his training taught him to do in this situation. This was followed by the perception-based outputs, *shape/distance* and *hearing/sound*, as more means of assessing the situation. Lastly, theory of mind scored a four.

The experienced Officer P3 utilized *vision* and *theory of mind* in his one road stop scenario. Based on what he could see and thought the civilian was going to do, he did not put much activation into *memory* or *shape/distance*. *Shape/distance*, in this study, is interpreted as determining if an object is a weapon, and this officer did not think further or did not need to think further about any shapes. *Hearing/sound* scored a three, which means he were listening for additional information.

The next category is the routine and welfare checks. Officer P1 participated in two of these scenarios. The highly ranked outputs for these scenarios was *theory of mind* and *hearing/sound*. Listening and trying to understand the civilians corresponds well to welfare checks. *Vision* ranked next as a method of examining the situation with some training memories following after. *Shape/distance* was ranked last meaning it was either not considered or the weapon was already identified.

Officer P2 used a different approach than that of Officer P1, who relied heavily on *theory of mind*. Officer P2 partook in two scenarios and decided to believe in his training, giving *memory* his highest score. This was followed by *hearing/sound* related to talking and the environment. *Vision* ranked third with *theory of mind* fourth as additional ways to assess the situation. *Shape/distance* was the lowest for this Officer as well leading to a similar conclusion that the weapon was either visible or not considered.

Officer P3 took a similar approach to that of Officer P1, utilizing *theory of mind* as his highest scored output. Officer P3 was involved in five routine and welfare check scenarios. As an experienced officer, using *theory of mind* in tandem with *memory* allowed this officer to assess the situation. *Hearing/sound* and *vision* received scores of three as perception-based skills. *Shape/distance* fell in last again leading to the same conclusion of a visible weapon, or that the weapon was not considered.

Officer P4 was the other experienced officer. Officer P4 logged four routine and welfare check scenarios. Officer P4 had a similar approach to that of Officer P2, who favored *memory* over *theory of mind*. Officer P4 was aided by his experience and kept keenly aware of weapons with a highly ranked *shape/distance* and *vision*. Lastly, *hearing/sound* and *theory of mind* were both ranked last. This Officer was potentially able to size up the situation without the need of these skills.

The third category is combat operations. Officer P1 was evaluated on two scenarios. During these scenarios, Officer P1 favored his perception-based skills highly such as *vision*, *shape/distance*, and *hearing/sound*. *Memory* was ranked fourth meaning that combat operations were already ingrained into his body. *Theory of mind* ranked last meaning that it had little or no perceived use in these scenarios.

Officer P2 had a hostage situation as one of his two scenarios, with the other being a Special Weapons and Tactics (SWAT) operation. *Hearing/sound* was his highest, but *theory of mind* came in second. In operations like these, communicating and understanding the intentions of the squad mates is important. Also, for hostage situations, listening and trying to diffuse the situation is respectable. *Memory* was used to maintain proper protocol. *Vision* and *shape/distance* came in last, likely due to the straightforwardness of the scenarios.

Officer P3 partook in two combat operation scenarios. In his scenarios, he relied heavily on assessing the situation visually. He fell back onto his *memory*, as a more experienced officer, to handle the scenarios. Lastly, *hearing/sound* and *theory of mind* were not utilized or had no perceived value in these scenarios.

Officer P4 only recorded one scenario. His *memory* ranked the highest meaning that similar operations or tactics were being considered. Outside of that, Officer P4 followed similar suit to Officer P3 relying on his *visual* skills to assess the situation, leaving *hearing/sound* and *theory of mind* to be considered unnecessary for this scenario.

The last category considered is threat response. Officer P1 conducted two scenarios in this category. *Hearing/sound* ranked his highest, with everything else falling to the lower half of the scoring. *Vison* and *memory* both had a score of three as ways to help assess and visualize the scenario. One of the scenarios was a robbery suspect running, so listening and trying to understand where the robber would go, also falls into these skills.

Officer P2's threat response scenarios consisted of five scenarios, of which four were shootings. The highest scores were *hearing/sound* and *theory of mind*. These were attributed to locating the shooters with sound and determining their potential actions. *Memory* was also high,

relating to his training or memorizing his surroundings. The lowest scores were *vision* and *shape/distance* meaning that line of sight might have been obscured by walls or objects.

Officer P3 handled four close combat encounters, one of which left his partner disarmed. The activation spread was high for *vision, memory*, and *theory of mind*. Being close to the encounters already, *shape/distance* and *hearing/sound* were not deemed as important as being able to see remember and predict the actions of the suspect in question.

For Officer P4, there were three scenarios to be completed. Officer P4 had used his *memory* consistently and this category is no different. His experience aids him in assessing situations, but in this case, the rest of his outputs were low. *Vision*, *shape/distance* were both scored around three as the officer's means of visually interpreting the scenario. Officer P4 also partook in the partner disarmed scenario, but Officer P4's *theory of mind* suggests that he was not worried about how his partner would react and rather focused on the task at hand.

Overall, this study represents an interesting look into the minds of the officers. Comparing and contrasting the lesser experienced officers with the veteran officers allowed for a greater understanding of the methods taken to complete each category of scenarios. These insights allow for a better understanding of each officer's skill set and approach to the given problems. Each set of scores painted a picture of how the subject's brain was acting during his task, which lined up with the expected results given the scenarios.

5.2 Intellectual Merit of the Conducted Research

This study has three major parts, automatic filtering, source localization, and the application of a fuzzy controller for data analysis. Each section can stand alone, if a different filtering is used, then the procedure for LORETA would still work. If filtering is the only thing desired, the automatic script can be used. The fuzzy controller is designed to process any sort of

data. For this research, all three sections were required to process and prepare the data for the Fuzzy controller to complete the analysis. The breakdown of the sections and the process is shown in Figure 5.4. The fact that each of these sections can stand alone makes each single research effort substantially more useful in the grand scheme.

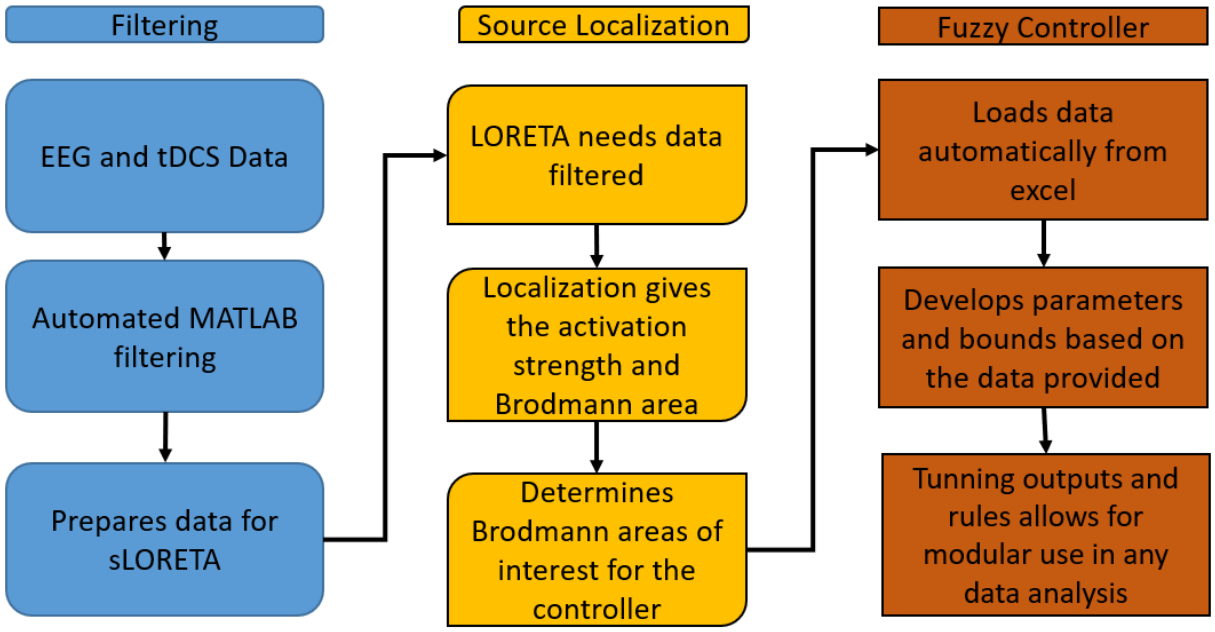

Figure 5.4. Breakdown of study sections.

The automatic filtering (see script in Appendix B) contributes towards handling larger data sets in a timely and accurate manner. Manually filtering large studies of over 50 subjects, such as [1, 2], involves individually loading and filtering each session which is time consuming and prone to error. This automated filtering script aims at alleviating this burden. EEGLAB is able to process event-related EEG, magnetoencephalography (MEG) and other electrophysiological data, which means this script has many applications in biomedical research. These uses can branch out even further if individual sections of the script are considered such as the data grabber included in Appendix A. With the help of the data grabber, the script is able to incrementally analyze each file and split the data based on the variables set. Since there are many

systems that are designed to output excel tables, this data grabber section can be used to load all of the files from a system and break up those that contain multiple occurrences.

The source localization process within LORETA works with any data already compatible with LORETA. LORETA's source localizations calculations transform the data into viewer files which give a three-dimensional representation of the brain and the active areas along with the level of activation. It is possible to utilize this method for any EEG study to determine the areas of interest in the brain, for the given data. The process for LORETA described in this research focuses on preparing the data for one of the many applications of the program. This preparation can branch out to other applications of the software. LORETA is a specialized tool for EEG data, making it a powerful tool to analyzing brain data.

The fuzzy controller can be used with any study, given that the data is in the correct format described in the comments laid out in its MATLAB script. The controller adapts itself as more data is added by adjusted the bounds of what is considered low, medium, or high. Therefore, the more data fed into the controller the more accurate it will become. This is where the full process comes into play, in regard to EEG data. The optimization of the process to get EEG data filtered and the sources localized leads to a quicker turn around on new data. The reason why the fuzzy controller is not a true controller is because this process takes time, but with the streamlined process, it is possible to quickly get the new data filtered, localized, and then added into the controller in a timely manner to allow for the new analysis to be ready for the next session.

This rapid turnaround time enables the controller to be utilized to its greatest potential. Researchers are able to quickly upload and compare data to previous sessions, allowing them to adjust their next session based on any insight gained. This insight could include any number of

observations such as determining areas of the brain to analyze further or finding a potentially harmful trend in the data. In that regard, researchers can gather data for each scenario and group them based on the areas of the brain excited during the subjects' performance. These areas of the brain would then be interpreted into the outputs of interest. This process can be utilized for any study that can be mapped out with rules. The way that the outputs are set up, the researcher is able to assess not only the subjects but also the content of the sessions. This should remain true for any study analyzed with this controller.

The fuzzy controller excels in handling data sets where low, med, and high are not exact and multiple variables exist. When these conditions are met, the controller is able to handle the ambiguity of the data, by creating its own bounds based on what is present. The potential for growth means that this controller could evolve even further bringing in a whole new range of data analyses.

5.3 Recommendations for Future Research

5.3.1 Scripts and filtering Recommendations

EEG readings become very complex when filtering comes into play. Currently, the script is set up to require the user to determine the parameters for the filtering desired. By adding additional code, it could be possible to load the data preemptively to determine the bounds of the filtering required. This can be taken a step further and rather than automatically determining the filtering for the whole study, individual sessions would have their parameters calculated as part of the process leading to a more extensive filtering.

Another recommendation would be to account for removed channels with more than interpolation. This could be accomplished by moving from artifact removal to artifact correction. It is difficult to restore noisy segments of data after they have been affected, so an independent

component analysis (ICA) is run to remove the channels perceived as noise. To recover these channels, interpolating the surrounding channels allows for the data to be complete again, but this process could be improved greatly.

The next potential expansion for this study would be to add-in functionality in the script to allow for cross data analysis. What this means is that the data would be analyzed in a way that makes the study normalized, and then be directly compared to another similar study. The issue with this is that EEG protocol and devices differ enough such that, for instance, if study "A" uses machine "A", conductive jelly "A", and protocol "A" for data collection, the output may vary greatly based on the number of changed parameters when compared to another study. This could be accomplished by loading both of the separate data sets in and then tagging both studies to be processed relative to the studies themselves. Then the two datasets would be processed together after normalization. In the long term, this would allow for data bases to exist and be consistently added-to in order to better understand any given field of research.

5.3.2 LORETA and Source Localization Recommendations

A major bottleneck of the whole process lies in the source localization. LORETA was chosen based on its accuracy. That being said, there appears no easy process of automation for what was desired. The EEGLAB program utilized for the filtering phase could potentially be used for source localization as well to streamline the full EEG data from within MATLAB. Certain checkpoints where data is exported would be suggested since the filtering process should only need to be done once, and the exported data would just be used in the second part.

LORETA is set up to use the default head model built into their application. For these models, simulations were performed on a three-shell spherical head model using the Talairach human brain atlas. The Brain Imaging Centre at Montreal Neurological Institute retains the

available digitized MRI files [3]. While these models are useful, it is possible to measure and collect the head models of subjects using fMRI. These models would allow for a more accurate source localization within LORETA. This too would benefit from automation. Part of the process of LORETA was determining the areas of interest for each given study. The automation would analyze each Brodmann area and determine a ranking for the whole study based on the most active regions of the brain. This process could be in either MATLAB or LORETA.

5.3.3 Fuzzy Controller Recommendations

The Fuzzy controller is the driving force of this research, but there are still ways to enhance it. The first potential improvement was quickly realized as too complicated for the scope of this research and that was having an automated rule generation engine, this would be a difficult process to accomplish even if the problem domain remained within EEG applications. Each Brodmann area would need to be mapped to multiple outputs, and even then, the user would need to determine which inputs and outputs to use. There is an option to map every Broadman area's outputs to the inputs as a starting point and then expand over time.

The current iteration of the controller presents multiple collections of data, but there are not any explicitly stated conclusions build into the controller. While it would be better to manually analyze the data, a nice starting point could be automatic strings that present data in descriptive form such as 'the activation of vision was consistently high for Subject 1', 'the theory of mind calculations were all high for this scenario category' and 'Subject 2 favors the anode stimulation more consistently'.

Finally, the controller could be moved into Simulink and turned into a true active controller for applications outside of EEG, or if the whole process is automated within MATLAB, it could be possible to send data intermittently in real time to be mapped to tDCS

parameters for a semi-active controller. The data would have to build up for at least a short time, depending on how well the filtering has evolved, and then additional readings could be added to the controller piece by piece.

Chapter 5 References

[1] C. Cosmo *et al.*, "Spreading effect of tDCS in individuals with attention-deficit/hyperactivity disorder as shown by functional cortical networks: a randomized, double-blind, sham-controlled trial," *Frontiers in psychiatry,* vol. 6, 2015.

[2] K.-A. Ho *et al.*, "The Effect of Transcranial Direct Current Stimulation (tDCS) Electrode Size and Current Intensity on Motor Cortical Excitability: Evidence From Single and Repeated Sessions," *Brain Stimulation,* vol. 9, no. 1, pp. 1-7, 2016.

[3] R. D. Pascual-Marqui, "Standardized low-resolution brain electromagnetic tomography (sLORETA): technical details," *Methods Find Exp Clin Pharmacol,* vol. 24, no. Suppl D, pp. 5-12, 2002.

APPENDICES

Appendix A

Data Grabber

```
%This file is used to quickly load all of a specified type of data from a
%folder. The type of file will be specified and then this will load all of
%the files in the given folder. This list can then be used to load all the
%data files into another script.

clc
clear all

MainFolder='C:\\Data'; %This is the main folder with subject folders
TypeFile='.set'; %looking for specified files
Output='C:\\Output'; %Location to Output data
Subject_List={};
Times={};
folder={};


fol=1; %sets up file matrix
files={}; % setting up file matrix
Main= dir(MainFolder); % Lists files in main folder
NMainp=size(Main); % determines number of folders
NMain=NMainp(1); %saves number of folders
for nfolder=3:NMain %Creates directories of each folder in the Main Folder
    prep=Main(nfolder);
    pfolder=prep.name;
    folder=sprintf('%s/%s',MainFolder,pfolder);
    subfoldername{nfolder-2}=folder;
    Files{nfolder-2}= dir(folder); % Lists files in main folder.
    fprep=size(Files{1,nfolder-2}); %returns number of files in folder
    ffile=fprep(1);
    for check=3:ffile
        nprep=Files{1,nfolder-2}(check);
        nfile=nprep.name;
        compare=strfind(nfile,TypeFile); % Checks to find the type of file
specified
        if (compare>1)
            files{fol}=nfile;
            fol=fol+1;
            compare=0;
        end


    end

    ColFiles{1,nfolder-2}=files; %Collects All folder data
```

```
    ColFilesT{1,nfolder-2}=transpose(files); %Collects All folder data
    fol=1; %resets i
    files={}; %resets files
end
```

Appendix B

Filtering Script

```
%This code is designed to grab all of a specified data type from within a
%specified folder. Then each data set is time stamped with a marker based
%on how the data is to be split up. The script then runs ICA on the data.
%The data is then filtered with ICA rejecting bad channels. Bad channels
%are then interpolated back. The data is then split up into sections based
%on the markers. These sections are then filtered with automatic continuous
%rejection. These sections are then saved in a file format to be used with
%sLORETA. They are named based on the file the data is from and the section
%of data represented.


%Make sure to run eeglab.m before running this code
%Create Fullset.mat from loading a set and running edit->channel locations
%then save the data in ALLEEG.channellocs

%Set subjects equal to your number of subjects
%Edit Channel list with channel names (Pz, F4, Oz, FP7, etc.)
%Edit the data path to where your data is currently.

%Change parameters below as desired

%Make sure that the output folder has identical folders to the input
%folders


clc %Clears Screen
clear all %Clears Variables


sampRate = 500; %Sampling rate of your data

ChLocs=load('MathDataFullset.mat'); %'Fullset.mat' loads up channel
locations. You can make this from saving the ALLEEG.channellocs from a
dataset with all desired channels
lengthcheck=0; %Set equal to 1 to segment data differently based on data
length, otherwise set to 0
DesiredSeg=300; %For data longer than 30 minutes set DesiredSeg to desired
segment length in seconds
StaticSeg=20; %Set lengthcheck to 0 if the segments are desired to be
consistent and not based on data length
outputL=10; %Desired length of data exported in seconds
offsettime=0; %Set to 0 for start or desired start time in seconds
MaxSegs=5; %Limits the maximum number of segments the data is split into.
RejL=0; %Sets Frequency low limit for Automatic rejection
```

```
RejH=40; %Sets Frequency high limit for Automatic rejection


channel_list={'Fp1' 'Fp2' 'F7' 'F3' 'Fz' 'F4' 'F8' 'T7' 'C3' 'Cz' 'C4' 'T8'
'P3' 'Pz' 'P4' 'O1' 'O2' 'TP7' 'TP8' 'Oz'}; %Change based on channels that
need analysis
nchanp=size (channel_list);
nchan=nchanp(2); %Sets nchan equal to number of channels

ColEvent={}; %Setting up Event Collecting
baddata={};  %Setting up data checking
badnumber=1; %Setting up variables

MainFolder='C:\\MathData\\Data\\'; %This is the main folder with data
TypeFile='.set'; %Looking for this type of file
Output='C:\\MathData\\Output2'; %Location to Output data

fol=1; %Sets up file matrix
files={}; %Setting up file matrix
Main= dir(MainFolder); %Lists files in main folder
NMainp=size(Main); %Determines number of folders
NMain=NMainp(1); %Saves number of folders
for nfolder=3:NMain %Creates directories of each folder in the Main Folder
    prep=Main(nfolder);
    pfolder=prep.name;
    folder=sprintf('%s/%s/',MainFolder,pfolder); %Creates folder location
    ofolder=sprintf('%s/%s/',Output,pfolder); %Creates output location
    subfoldername{nfolder-2}=folder; %Collects folder locations
    osubfoldername{nfolder-2}=ofolder; %Collects output locations
    Files{nfolder-2}= dir(folder); % Lists files in main folder.
    fprep=size(Files{1,nfolder-2}); %Returns number of files in folder
    ffile=fprep(1);
    for check=3:ffile
        nprep=Files{1,nfolder-2}(check);
        nfile=nprep.name;
        compare=strfind(nfile,TypeFile); %Checks to find the type of file
specified
        if (compare>1)
            files{fol}=nfile;
            fol=fol+1;
            compare=0;
        end

    end

    ColFiles{1,nfolder-2}=files; %Collects All folder data
    ColFilesT{1,nfolder-2}=transpose(files); %Collects All folder data
    fol=1; %Resets i
    files={}; %Resets files

end


%Used to setup the data
for subfolder=3:NMain %Cycles through the subfolder data
```

```
subject_list=ColFilesT{1,subfolder-2}; %Makes subject list based on subfolder
Data_path= subfoldername(subfolder-2); %Selects the current folder for the
data path
Output_path= osubfoldername(subfolder-2);%Selects the current folder for the
output path
nsubjp=size (subject_list); %Sets up the number of subjects/files
nsubj=nsubjp(1); %Sets the number of subjects/files


for s=1:nsubj
EEG = pop_loadset ('filename',[subject_list(s)], 'filepath' , Data_path{1});
%Load data, if error run EEGLAB.m

oldevent=EEG.event; %Saves current old events here if needed

new.type='start';%Setting up to overwrites event matrix
new.latency=1;
new.duration=0;

EEG.event=new; %Starts a new event list

EEG = pop_select(EEG,'channel',channel_list);%Data will be retained from
these channels (used to remove empty channels)

EEG=pop_chanedit(EEG, 'lookup','standard-10-5-cap385.elp'); %Give channels
locations. need to change home path

DataL=EEG.xmax; %Returns the data length in seconds

if (lengthcheck==1) %Will segment data based on length

if (DataL<=900) %Segments in sets of 3 minutes
    Type=180; %Short data type
end
if(900<DataL && DataL<1800) %Segments in sets of 5 minutes
    Type=300; %Medium data type
end
if(DataL>=1800)
    Type=DesiredSeg; %Long data type (Set based on desired segment lengths)
end
end

if (lengthcheck==0)
    Type=StaticSeg; %Defined up top if the segments are desired to be
consistent
end


% EEG = pop_runica(EEG, 'extended',1,'interupt','on'); %Runs ICA for channel
rejection; can be commented out if ICA has been done already

EEG = pop_rejchan(EEG,'threshold',5,'norm','on','measure','kurt'); %Rejects
noisy channels
```

```
EEG = pop_interp(EEG, ChLocs.MathdataFullset, 'spherical'); %Interpolates
back rejected channels

EEG = pop_reref(EEG, []); %Re-references the electrodes based on the mean


extra=mod(EEG.xmax,Type); %Determines size of extra data if it cannot make
the full segment
cycles=floor(EEG.xmax/Type); %Determines the number of full segments in the
data
Tcycles=min(cycles,MaxSegs); %Takes the minimum of MaxSegs and the actual
segments available

lat=0; %Initializes lat for marking
for marking=1:(Tcycles+1) %The first cycle is for zero-time marker

Mark=sprintf('Marker%d', marking); %Adds markers based on type
EEG = pop_editeventvals(EEG,'add',{1 [] [] []},'changefield',{1 'latency'
lat},'changefield',{1 'type' Mark},'changefield',{1 'duration' 0}); %Inserts
event for eachtime frame
%Number after latency is in seconds, name after type is what event is called.
Number after duration is duration
lat=lat+Type;

end

if (extra>(Type/2)) %Adds the final marker to the last section of data if it
is longer than half the other data length
EEG = pop_editeventvals(EEG,'add',{1 [] [] []},'changefield',{1 'latency'
DataL},'changefield',{1 'type' 'EndMarker'},'changefield',{1 'duration' 0});
%Inserts event for each time frame
end


%Code to split up the file based on markers
time=transpose({EEG.event.latency}); %Collects times
event=transpose({EEG.event.type}); %Collects events
full=[event,time];
pevent=size(event);
nevent=pevent(1); %Returns the number of events
i=1; %Initializing i
MarkerMat={}; %Setting up start matrix
compare=0;
Markerterm='Marker'; %Input the name of the start marker (not case sensitive)
samplingfreq=EEG.srate; %Sets sampling frequency


for c=1:nchan %If interpolation is not done, this will return the name of any
removed channels

    chanName=channel_list(c);

Chfind={EEG.chanlocs.labels}; %Shows Current EEG channel names
Chs=size(Chfind); %Gives number of channels
```

```
i=1;%Resets i
find=0;

while (i<(Chs(2)+1)&& find==0)
      if (strcmp(Chfind(i),chanName)==1); %Looks for missing channels
           chanNr=i
           find=1
      else
          i=i+1;
      end

end

if(find==0)
      fprintf('Channel not found\n') %Prints out list of missing channels if
they are not interpolated back
      badchan=(sprintf('%s_%s',subject_list{s},chanName{1}));
      baddata{badnumber}=badchan;
      badnumber=badnumber+1;
end

i=1;%resets i
for e=1:nevent
compare=strfind(lower(full{e,1}),lower(Markerterm)); % Checks to find the
marker term is specified
    if (compare>=1)
            MarkerMat(i,:)=full(e,:); %Saving to file list
            i=i+1;
            compare=0;
    end

end

ptotalSessions=size(MarkerMat); %Sets up number of sessions
ntotalSessions=ptotalSessions(1)-1;
i=1; %resets i
atotalSessions=min(ntotalSessions,MaxSegs); %Takes the minimum of MaxSegs and
the actual session available

for (session=1:atotalSessions)
   first=cell2mat(MarkerMat(session,2)); %Sets to start time
   nfirst=floor(first/samplingfreq); %Converts to seconds
   last=cell2mat(MarkerMat(session+1,2)); %Sets the end time
   nlast=floor(last/samplingfreq); %Sets end time
   EEGsessionFull{session} = pop_select( EEG,'time',[nfirst+offsettime
nlast+offsettime] ); %Grabs section of data for continuous rejection
   EEGsessionFull{session} = pop_rejcont(EEGsessionFull{session}
,'freqlimit',[RejL RejH]
,'threshold',10,'epochlength',0.5,'contiguous',4,'addlength',0.25,'taper','ha
mming'); %Continuous rejection

   % If the code stops here because Hamming window is not available use the
next line instead of the previous
   % EEGsessionFull{session} = pop_rejcont(EEGsessionFull{session}
,'freqlimit',[RejL RejH]
```

```
,'threshold',10,'epochlength',0.5,'contiguous',4,'addlength',0.25,'taper','no
ne'); %If MATLAB doesn't have Hamming use this

   SectionL=EEGsessionFull{session}.xmax; %Returns the data length in seconds

   nfirst=0; %Sets to start time (secs)
   nlast=SectionL; %Sets the end time (secs)
   mid=(nlast-nfirst)/2; %Sets mid time

   EEGsession1{session} = pop_select( EEGsessionFull{session},'time',[nfirst
nfirst+outputL] ); %Grabs beginning section data
   EEGsession2{session} = pop_select( EEGsessionFull{session},'time',[mid-
(outputL/2) mid+(outputL/2)] ); %Grabs mid section data
   EEGsession3{session} = pop_select( EEGsessionFull{session},'time',[nlast-
outputL nlast] ); %Grabs end section data

   Filename1=(sprintf('1stP%s%d.txt',subject_list{s},session)); %Used to Set
up the saved file Name
   Filename2=(sprintf('2ndP%s%d.txt',subject_list{s},session)); %Used to Set
up the saved file Name
   Filename3=(sprintf('3rdP%s%d.txt',subject_list{s},session)); %Used to Set
up the saved file Name

   writesLORdat(EEGsession1{session},Filename1, Output_path{1}); %Saves Data
in sLORETA format in the Output folder specified at the top
   writesLORdat(EEGsession2{session},Filename2, Output_path{1}); %Saves Data
in sLORETA format in the Output folder specified at the top
   writesLORdat(EEGsession3{session},Filename3, Output_path{1}); %Saves Data
in sLORETA format in the Output folder specified at the top

end
end
end

ColFiles{1,nfolder-2}=files; %Collects All folder data
i=1; %Resets i
files={}; %Resets files

end

fprintf('Channels not found on Data sets\n') %Prints the comment
baddata %Prints out bad channels
```

Appendix C

Police Split Script

```
% This file is used to recognize the given start term and the given end
% term for EEG events. The script will then break the data file into
% smaller sections. In this file there is also a shot term which will end
% the section early. This will also save the reflection time after the
% scenario has ended and before the next one starts.



EEG = pop_loadset('filename','Test data 5
scenarios.set','filepath','C:\\Mainfolder\\');
Output='C:\\Mainfolder\\file\\'; %Location to Output data
NOutput='C:\\Mainfolder\\file\\p1\\'; %Location to Output data


time=transpose({EEG.event.latency}); %collects times
event=transpose({EEG.event.type}); %collects events
full=[event,time];
pevent=size(event);
nevent=pevent(1); %returns the number of events
i=1; %initializing i
start={}; % setting up start matrix
shot={}; % setting up shot matrix
endd={}; % setting up end matrix
compare=0;
startterm='start'; %input the name of the start marker (not case sensitive)
shotterm='shot'; %input shot term
endterm='end'; %input end term
samplingfreq=EEG.srate; %sets sampling frequency



for (e=1:nevent)
compare=strfind(lower(full{e,1}),lower(startterm)); % Checks to find the
start term specified
    if (compare>=1)
            start(i,:)=full(e,:); %Saving to file list
            i=i+1;
            compare=0;
    end

end
i=1;

for (e=1:nevent)
```

```
compare=strfind(lower(full{e,1}),lower(shotterm)); % Checks to find the shot
term specified
    if (compare>=1)
            shot(i,:)=full(e,:); %Saving to file list
            i=i+1;
            compare=0;
    end

end

i=1;
for (e=1:nevent)
compare=strfind(lower(full{e,1}),lower(endterm)); % Checks to find the end
term specified
    if (compare>=1)
        endd(i,:)=full(e,:); %Saving to file list
            i=i+1;
            compare=0;
    end

end

ptotalSessions=size(start); %sets up number of sessions
ntotalSessions=ptotalSessions(1);
pshot=size(shot); %sets up number of shots
nshot=pshot(1);
shotfind=0; %resets shot find
Trueendtime={EEG.xmax*samplingfreq}; %returns the data length in seconds
Trueend={'trueend'};%used to retrieve the final reflection from the data
start=[start;Trueend,Trueendtime];
for (session=1:ntotalSessions)
   first=cell2mat(start(session,2)); %sets to start time
   nfirst=floor(first/samplingfreq); %Converts to seconds
   last=cell2mat(endd(session,2)); %sets the end time
   nlast=floor(last/samplingfreq); %sets end to end time if no shot is found
   tlast=floor(last/samplingfreq);
   refend=cell2mat(start(session+1,2)); %sets reflection end
   nrefend=floor(refend/samplingfreq);
   for (cshot=1:nshot)
       tshot=cell2mat(shot(cshot,2)); %testing shot time
       if (shotfind<1)
           if (tshot>first && tshot<last)
                nlast=floor(tshot/samplingfreq); %sets last to shot
                shotfind=1; %returns that the shot is found
           end

        end

   end
EEGsession{session} = pop_select( EEG,'time',[nfirst nlast] ); %Grabs Session
data
EEGreflection{session} = pop_select( EEG,'time',[tlast nrefend] ); %Grabs
Refelction data
FilenameS=(sprintf('S%d.set',session)); %Used to Set up the saved file Name
FilenameR=(sprintf('R%d.set',session)); %Used to Set up the saved file Name
```

```
Ssession = pop_saveset(EEGsession{session},
'filename',FilenameS,'filepath',NOutput); %Saves data as .set dataend
Sreflection= pop_saveset( EEGreflection{session},
'filename',FilenameR,'filepath',NOutput); %Saves data as .set dataend
shotfind=0;
end
```

Appendix D

Math MATLAB controller Script

```
%documentation https://www.mathworks.com/help/fuzzy/referencelist.html

%Each subject visit should be a file. In that file, each sheet represents a
%session during the visit. Baselines are the first sheet and each sheet
%after will use those baselines. Within in the sheet, channels should be
%unique to columns, with each column representing a Brodmann area or
%channel as desired. A unique set of data is also acceptable

%remove files from folder as needed to adjust what data is compared against

close all %Closes all figures
clear all %Clears all variables
clc       %Clears the screen

MainFolder="T:\\MathMatlabData392\\"; %The folder that contains the files
Baselineloc=1; %set equal to column of baseline
RemoveBase=0; %set to 1 to remove base from average and max
calculations(recommended)
BaseCalcs=0;%set to 1 to do calculations based on base
Exclude=0; %set equal to 1 to exclude the following dataset from the overall
analysis
Edata=5; %Excludes the file in this position from the data
EqualWeight=1; %Set equal to 1 for each session to have the same weight in
overall calculations
PerLow=5;%Sets the percentile for Data to be considered Low
PerMed=50;%Sets the percentile for Data to be considered Med
PerHigh=95;%Sets the percentile for Data to be considered High
InpOverlap=.5;%Sets the overlap of the Input variables

Controllername="FuzzyController";
Controller = mamfis('Name',Controllername); %initializes a mamdani fuzzy
controller variable

nameInput={"BA10","BA20","BA39","BA47"}; %names inputs
nameMF={"Low","Med","High"}; %names membership functions
input={"gaussmf","gaussmf","gaussmf"}; %Sets the type of membership funtion

% "gbellmf" Generalized bell-shaped membership function gbellmf
% "gaussmf" Gaussian membership function    gaussmf
% "gauss2mf"    Gaussian combination membership function    gauss2mf
% "trimf"   Triangular membership function  trimf
% "trapmf"  Trapezoidal membership function trapmf
% "sigmf"   Sigmoidal membership function   sigmf
% "dsigmf"  Difference between two sigmoidal membership functions   dsigmf
% "psigmf"  Product of two sigmoidal membership functions   psigmf
```

```
% "zmf" Z-shaped membership function    zmf
% "pimf"    Pi-shaped membership function   pimf
% "smf" S-shaped membership function    smf

%%%Outputs defined below
nameOutput={"AnodeAffinity","CathodeAffinity","Calculations","Decision","Memo
ry"}; %names inputs

Controller = addOutput(Controller,[0 100],'Name',nameOutput{1}); %bracket
indicates range;section after name is used to name input
Controller = addMF(Controller,nameOutput{1},"trapmf",[-20 0 20
40],'Name',"Low");
Controller = addMF(Controller,nameOutput{1},"trapmf",[20 40 60
80],'Name',"Med");
Controller = addMF(Controller,nameOutput{1},"trapmf",[60 80 100
120],'Name',"High");

Controller = addOutput(Controller,[0 100],'Name',nameOutput{2}); %bracket
indicates range;section after name is used to name input
Controller = addMF(Controller,nameOutput{2},"trapmf",[-20 0 20
40],'Name',"Low");
Controller = addMF(Controller,nameOutput{2},"trapmf",[20 40 60
80],'Name',"Med");
Controller = addMF(Controller,nameOutput{2},"trapmf",[60 80 100
120],'Name',"High");

Controller = addOutput(Controller,[0 100],'Name',nameOutput{3}); %bracket
indicates range;section after name is used to name input
Controller = addMF(Controller,nameOutput{3},"trapmf",[-20 0 20
40],'Name',"Low");
Controller = addMF(Controller,nameOutput{3},"trapmf",[20 40 60
80],'Name',"Med");
Controller = addMF(Controller,nameOutput{3},"trapmf",[60 80 100
120],'Name',"High");

Controller = addOutput(Controller,[0 100],'Name',nameOutput{4}); %bracket
indicates range;section after name is used to name input
Controller = addMF(Controller,nameOutput{4},"trapmf",[-20 0 20
40],'Name',"Low");
Controller = addMF(Controller,nameOutput{4},"trapmf",[20 40 60
80],'Name',"Med");
Controller = addMF(Controller,nameOutput{4},"trapmf",[60 80 100
120],'Name',"High");

Controller = addOutput(Controller,[0 100],'Name',nameOutput{5}); %bracket
indicates range;section after name is used to name input
Controller = addMF(Controller,nameOutput{5},"trapmf",[-20 0 20
40],'Name',"Low");
Controller = addMF(Controller,nameOutput{5},"trapmf",[20 40 60
80],'Name',"Med");
Controller = addMF(Controller,nameOutput{5},"trapmf",[60 80 100
120],'Name',"High");

Controller.DefuzzificationMethod = "centroid"; %used to change
Defuzzification Method
```

```
% myFIS.ImplicationMethod = "customimp";
% myFIS.AggregationMethod = "customagg";
% myFIS.DTypeReductionMethod = "customtr";


% ==    IS (in rule antecedent)
% ~=    IS NOT
% & AND
% | OR
% =>    THEN
% = IS (in rule consequent)

%First rule needs to be 1 and then the rest can be end+1 in any order
rule(1)= "BA20~=High & BA39==High => AnodeAffinity=High";
rule(end+1)= "BA20~=High & BA39==Med => AnodeAffinity=Med";
rule(end+1)= "BA20~=Low & BA39==Low => AnodeAffinity=Low";

rule(end+1)= "BA39~=High & BA20==High => CathodeAffinity=High";
rule(end+1)= "BA39~=High & BA20==Med => CathodeAffinity=Med";
rule(end+1)= "BA39~=Low & BA20==Low => CathodeAffinity=Low";

rule(end+1)= "BA47~=Low & BA39==High => Calculations=High";
rule(end+1)= "BA10==High & BA39~=Low => Calculations=High";
rule(end+1)= "BA47==High & BA39~=Low => Calculations=High";
rule(end+1)= "BA10==Med & BA39==Med => Calculations=Med";
rule(end+1)= "BA10==Low & BA39==Low => Calculations=Low";
rule(end+1)= "BA10==Low & BA47==Low => Calculations=Low";
rule(end+1)= "BA39==Low & BA47==Low => Calculations=Low";

rule(end+1)= "BA20==Med & BA47==High => Decision=High";
rule(end+1)= "BA47==High & BA20==High => Decision=High";
rule(end+1)= "BA47==Med & BA20==Med => Decision=Med";
rule(end+1)= "BA20==Med & BA39==Med => Decision=Med";
rule(end+1)= "BA47==Low & BA39==Low => Decision=Low";
rule(end+1)= "BA20==Low & BA47==Low => Decision=Low";
rule(end+1)= "BA20==Low & BA39==Low => Decision=Low";

rule(end+1)= "BA10==Med & BA47==High => Memory=High";
rule(end+1)= "BA10==High & BA47==High => Memory=High";
rule(end+1)= "BA10==Med & BA47==Med => Memory=Med";
rule(end+1)= "BA47==Med & BA39==Med => Memory=Med";
rule(end+1)= "BA10==Low & BA39==Low => Memory=Low";
rule(end+1)= "BA47==Low & BA10==Low => Memory=Low";
rule(end+1)= "BA47==Low & BA39==Low => Memory=Low";


%
ruletest=(sprintf('"%s==%s%s',nameInput{1,inputs},nameMF{1,i},nameMF{1,i}));
%test for auto rule making

%The coding for inputs can be modified in the code below as desired.
%Everything else should run as is.

%%Start of code
Sessionlist=dir(MainFolder); %Makes a list of all files and data
```

```
Session=transpose({Sessionlist.name}); %Takes just the names of the files
SizeSession=size(Session); %Prepares the number of files
SessionT=SizeSession(1); %Sets equal to number of files

for i=3:SessionT %Starts to load files and skips "." and ".."
Filename=sprintf('%s\\%s',MainFolder,Session{i}); %Sets up file name
[status,sheets] = xlsfinfo(Filename);
sheetsTp=size(sheets); %Prepares number of sheets
sheetsT=sheetsTp(2); %Sets equal to number of sheets

for sheet=1:sheetsT
Data{i-2,sheet}=xlsread(Filename,sheet); %Loads excel data
end

end

ExData=Data;%sets up for excluding data
FullData=Data;%Saves full data in a variable
if Exclude==1
ExData(Edata,:)=[]; %exclues the data set defined above
Data=ExData; %Makes the excluded data the main data set
end

DataI=size(Data); %Gets information of Data size
RowT=DataI(1);  %Equal to the total rows
ColumnOT=DataI(2); %Overall total columns

BaselineOffsetData{RowT,ColumnOT}=[];%Sets up variable and prealoocated size
BaselineRealativeData{RowT,ColumnOT}=[];%Sets up variable and prealoocated
size
ColumnMassData=[]; %Sets up variable
ColumnMassEData=[]; %Sets up variable
NormalizedEData=[]; %Sets up variable

%Setting up to process individual data
for Row=1:RowT

ColumnTp=(~cellfun(@isempty,Data(Row,:))); %Determines total Columns in the
row
ColumnT=sum(ColumnTp); %Gathers the amount of columns used

for Column=1:ColumnT

columnDTp=size(Data{Row,Column}); %Determines total Columns in the cell
columnDT=columnDTp(2); %Gathers the amount of columns used

for columnData=1:columnDT

if BaseCalcs==1
Base=Data{Row,Baselineloc}; %Baselineloc column defined above
BaselineOffsetData{Row,Column}(:,columnData)=Data{Row,Column}(:,columnData)-
Base(:,columnData);%subtracts baseline
BaselineRealativeData{Row,Column}(:,columnData)=Data{Row,Column}(:,columnData
)/Base(:,columnData);%divides baseline
```

```
end
SubjectMaxData{Row,Column}(:,columnData)=max(Data{Row,Column}(:,columnData));
%Finds Max of Data
SubjectMinData{Row,Column}(:,columnData)=min(Data{Row,Column}(:,columnData));
%Finds Min of Data
SubjectMeanData{Row,Column}(:,columnData)=mean(Data{Row,Column}(:,columnData)
); %Finds Mean of Data
SubjectStdevData{Row,Column}(:,columnData)=std(Data{Row,Column}(:,columnData)
); %Finds Standard Deviation of Data
%More calculations can be added here as desired

end
end
end

if RemoveBase==1
NoBaseData=Data; %sets equal to data
NoBaseData(:,Baselineloc)=[]; %removes the defined column of baseline data
CalcData=NoBaseData; %prepares for calculations
SubjectMaxData(:,Baselineloc)=[]; %removes the defined column of baseline
data
SubjectMinData(:,Baselineloc)=[]; %removes the defined column of baseline
data
SubjectMeanData(:,Baselineloc)=[]; %removes the defined column of baseline
data
SubjectStdevData(:,Baselineloc)=[]; %removes the defined column of baseline
data
else
CalcData=Data; %prepares for calculations
end

if EqualWeight==1
CalcData=SubjectMeanData; %Sets the data to be calculated to be equal per
session
end

DataI=size(CalcData); %gets information of Data size
RowT=DataI(1);
%Setting up to get overall data from columns
for Row=1:RowT
ColumnTp=(~cellfun(@isempty,CalcData(Row,:))); %preps for data sets
ColumnT=sum(ColumnTp); %Gathers the amount of cells used
for Column=1:ColumnT
ColumnMassData=[ColumnMassData;CalcData{Row,Column}]; % amasses data into one
matrix
end
end

ColumnMassTp=size(ColumnMassData); %preps size
ColumnMassT=ColumnMassTp(2); %takes total columns

for ColumnM=1:ColumnMassT
DataMax(ColumnM)=max(ColumnMassData(:,ColumnM)); %Calculates Max of all data
DataMin(ColumnM)=min(ColumnMassData(:,ColumnM)); %Calculates Min of all data
DataMean(ColumnM)=mean(ColumnMassData(:,ColumnM)); %Calculates Mean of all
data
```

```
DataStdev(ColumnM)=std(ColumnMassData(:,ColumnM)); %Calculates Standard
deviation of all data
DataMedL(ColumnM)=prctile(ColumnMassData(:,ColumnM),PerLow); %Calculates
Standard deviation of all data
DataMedM(ColumnM)=prctile(ColumnMassData(:,ColumnM),PerMed); %Calculates
Standard deviation of all data
DataMedH(ColumnM)=prctile(ColumnMassData(:,ColumnM),PerHigh); %Calculates
Standard deviation of all data
%More calculations can be added here as desired
end

LimitHigh=DataMax; %Sets the high limit to the max
LimitLow=DataMin; %Sets the low limit to the min

% Variable size based on defined and avg and sdx2 or sdx3 etc.
%%%% This can be changed to better reflect the data as desired
for inputs=1:size(nameInput,2)
Controller = addInput(Controller,[LimitLow(inputs)
LimitHigh(inputs)],'Name',nameInput{1,inputs}); %bracket indicates
range;section after name is used to name input
InputSize={[DataStdev(inputs)*InpOverlap
DataMedL(inputs)],[DataStdev(inputs)*InpOverlap
DataMedM(inputs)],[DataStdev(inputs)*InpOverlap DataMedH(inputs)]}; % make
these based off of data Avg overlap etc.
% InputSize={[DataStdev(inputs) DataMedL(inputs)-
(1*DataStdev(inputs))],[DataStdev(inputs)
DataMedM(inputs)],[DataStdev(inputs)
DataMedH(inputs)+(1*DataStdev(inputs))]}; %Adds standard dev to above for
bigger overlap
for i=1:size(input,2) %Adds the MF to each rule
Controller =
addMF(Controller,nameInput{1,inputs},input{1,i},InputSize{1,i},'Name',nameMF{
1,i});
end
end

%If the following line gives an error check spelling and syntax of rules and
inputs and outputs
Controller = addRule(Controller,rule); %adds the rules from above now that
inputs are defined


for Row=1:RowT

ColumnTp=(~cellfun(@isempty,Data(Row,:))); %Determines total Columns in the
row
ColumnT=sum(ColumnTp); %Gathers the amount of columns used

for Column=1:ColumnT

columnDTp=size(Data{Row,Column});
columnDT=columnDTp(2);


for columnData=1:columnDT
```

```
SubjectEvalData{Row,Column}=evalfis(Controller,(SubjectMeanData{Row,Column}))
; %evaluates the data based on the data inputs

end
end
end

SessionNames=Session(3:SessionT);

if Exclude==1
SessionNames(Edata,:)=[]; %excludes the data set defined above
end

FullOutput=[SessionNames,SubjectEvalData];

%Setting up to get overall data from Eval columns
DataE=size(SubjectEvalData); %gets information of Data size
RowET=DataE(1);
for RowE=1:RowET
ColumnETp=(~cellfun(@isempty,SubjectEvalData(RowE,:))); %preps for data sets
ColumnET=sum(ColumnETp); %Gathers the amount of cells used
for ColumnE=1:ColumnET
ColumnMassEData=[ColumnMassEData;SubjectEvalData{RowE,ColumnE}]; % amasses
data into one matrix
end
end

ColumnMassETp=size(ColumnMassEData); %preps size
ColumnMassET=ColumnMassETp(2); %takes total columns


% Additional calculations can be added here, if desired
for ColumnEM=1:ColumnMassET
DataEMax(ColumnEM)=max(ColumnMassEData(:,ColumnEM)); %Calculates Max of
Evaluations
DataEMin(ColumnEM)=min(ColumnMassEData(:,ColumnEM)); %Calculates Min of
Evaluations
DataEMean(ColumnEM)=mean(ColumnMassEData(:,ColumnEM)); %Calculates Mean of
Evaluations
DataEStdev(ColumnEM)=std(ColumnMassEData(:,ColumnEM)); %Calculates Standard
deviation of Evaluations

end

DataEFullMean=mean(ColumnMassEData,'all'); %Calculates Mean of the full
Matrix
DataEFullStdev=std(ColumnMassEData,0,'all'); %Calculates Standard deviation
of the full Matrix

for RowE=1:RowET %Prepares and normalizes the data

ColumnTp=(~cellfun(@isempty,SubjectEvalData(RowE,:))); %Determines total
Columns in the row
```

```
ColumnT=sum(ColumnTp); %Gathers the amount of columns used

for Column=1:ColumnT

columnDTp=size(SubjectEvalData{RowE,Column});
columnDT=columnDTp(2);

for columnData=1:columnDT

NormalizedEData{RowE,Column}(:,columnData)=(SubjectEvalData{RowE,Column}(:,co
lumnData)/DataEMax(:,columnData)*100);%Normalizes Data
ZscoreEData{RowE,Column}(:,columnData)=((SubjectEvalData{RowE,Column}(:,colum
nData)-DataEMean(:,columnData))/DataEStdev(:,columnData));%Finds z-score of
Data
ZscoreEFullData{RowE,Column}(:,columnData)=((SubjectEvalData{RowE,Column}(:,c
olumnData)-DataEFullMean)/DataEFullStdev);%Finds z-score of Data

end
end
end

FullOutputNorm=[SessionNames,NormalizedEData];
MatrixNames=[];
MatrixNumbers=[];
DataE=size(FullOutputNorm); %gets information of Data size
RowET=DataE(1);

for RowE=1:RowET %Prepares and takes the distance between the data

ColumnTp=(~cellfun(@isempty,FullOutputNorm(RowE,:))); %Determines total
Columns in the row
ColumnT=sum(ColumnTp); %Gathers the amount of columns used

for Column=2:ColumnT
MatrixNames=[MatrixNames,FullOutputNorm(RowE,1)];
MatrixNumbers=[MatrixNumbers,FullOutputNorm(RowE,Column)];

end
end

MatrixNamesU= string(MatrixNames) + (1:numel(MatrixNames)); %makes names
unique for heatmap
% cellstr(MatrixNamesU) % Can be used to change it back if needed

DataE=size(MatrixNumbers); %gets information of Data size
RowET=DataE(1);

for RowE=1:RowET %Prepares and takes the distance between the data

ColumnTp=(~cellfun(@isempty,MatrixNumbers(RowE,:))); %Determines total
Columns in the row
ColumnT=sum(ColumnTp); %Gathers the amount of columns used

for Column=1:ColumnT
```

```
columnDTp=size(MatrixNumbers{RowE,Column});
columnDT=columnDTp(2);
Dist1=MatrixNumbers{RowE,Column}; %Changes Dist1 for big loops
for Column2=1:ColumnT
Dist2=MatrixNumbers{RowE,Column2}; %Changes Dist2 for smaller loops
Square=[]; %empties square before loop
for columnData=1:columnDT
x2=Dist2(:,columnData); %Grabs numbers from cell based on iteration
x1=Dist1(:,columnData); %Grabs numbers from cell based on iteration
Square(columnData)=(x2-x1)^2; %takes the square of the difference
end
Sum=sum(Square); %takes the sum of the parts
Sqrt=sqrt(Sum); %takes the square root of the parts
% example sqrt( (x2-x1)^2+(y2-y1)^2+(z2-z1)^2 +(a2-a1)^2 +(b2-b1)^2
DistMatrixP(Column,Column2)=Sqrt;
end
end
end
DistMatrix=num2cell(DistMatrixP);
FullDistMatrixP=[MatrixNamesU;DistMatrix];
LeftsideP=transpose(MatrixNamesU); %prepares leftside
Leftside=[0;LeftsideP]; %prepares leftside
FullDistMatrix=[Leftside,FullDistMatrixP];
figure('Name','Distance Heat Map','NumberTitle','off');
heatmap(MatrixNamesU,LeftsideP,DistMatrixP)
sgtitle('Distance Heat Map');

%Makes the heatmaps for the figures named
len=length(ZscoreEData{1,1});
ZscoreEData(cellfun(@isempty,ZscoreEData)) = {nan(1,len)};
ZscoreEDataHeat = cell2mat(ZscoreEData); % Convert to array
figure('Name','Zscore Local Heat Map','NumberTitle','off');

for Split = 1:len
ZscoreEDataHeatF(:,:,Split) = ZscoreEDataHeat(:,Split:len:end); % split up
columns by page
end

for Map = 1:size(ZscoreEDataHeatF,3)
nexttile %makes the next map show in the same window
heatmap(ZscoreEDataHeatF(:,:,Map),'Title',Controller.Outputs(1,Map).Name,'YLa
bel','Subject','Xlabel','Session','YData',SessionNames) %Subject and Session
can be changed as desired
end
sgtitle('Zscore Local Heat Map');

%Makes the heatmaps for the figures named
len=length(ZscoreEFullData{1,1});
ZscoreEFullData(cellfun(@isempty,ZscoreEFullData)) = {nan(1,len)};
ZscoreEFullDataHeat = cell2mat(ZscoreEFullData); % Convert to array
figure('Name','Zscore Global Heat Map','NumberTitle','off');

for Split = 1:len
ZscoreEFullDataHeatF(:,:,Split) = ZscoreEFullDataHeat(:,Split:len:end); %
split up columns by page
end
```

```
for Map = 1:size(ZscoreEFullDataHeatF,3)
nexttile %makes the next map show in the same window
heatmap(ZscoreEFullDataHeatF(:,:,Map),'Title',Controller.Outputs(1,Map).Name,
'YLabel','Subject','Xlabel','Session','YData',SessionNames) %Subject and
Session can be changed as desired
end
sgtitle('Zscore Global Heat Map');

%Makes the heatmaps for the figures named
len=length(NormalizedEData{1,1});
NormalizedEData(cellfun(@isempty,NormalizedEData)) = {nan(1,len)};
NormalizedEDataHeat = cell2mat(NormalizedEData); % Convert to array
figure('Name','Normalized Data Heat Map','NumberTitle','off');


for Split = 1:len
NormalizedEFullDataHeatF(:,:,Split) = NormalizedEDataHeat(:,Split:len:end); %
split up columns by page
end

for Map = 1:size(NormalizedEFullDataHeatF,3)
nexttile %makes the next map show in the same window
heatmap(NormalizedEFullDataHeatF(:,:,Map),'Title',Controller.Outputs(1,Map).N
ame,'YLabel','Subject','Xlabel','Session','YData',SessionNames) %Subject and
Session can be changed as desired
end
sgtitle('Normalized Data Heat Map');



%Makes the heatmaps for the figures named
Ranking = NormalizedEData;
idx = cellfun(@(x) any(isnan(x)),Ranking);
Ranking(~idx) = cellfun(@RankingFunction, Ranking(~idx), "UniformOutput",
false);
len=length(Ranking{1,1});
Ranking(cellfun(@isempty,Ranking)) = {nan(1,len)};
RankingDataHeat = cell2mat(Ranking); % Convert to array
figure('Name','Ranking Data Heat Map','NumberTitle','off');


for Split = 1:len
RankingFullDataHeatF(:,:,Split) = RankingDataHeat(:,Split:len:end); % split
up columns by page
end

for Map = 1:size(RankingFullDataHeatF,3)
nexttile %makes the next map show in the same window
heatmap(RankingFullDataHeatF(:,:,Map),'Title',Controller.Outputs(1,Map).Name,
'YLabel','Subject','Xlabel','Session','YData',SessionNames,'ColorbarVisible',
'off','ColorMethod','min') %Subject and Session can be changed as desired

end
```

```
sgtitle('Ranking Data Heat Map');


for Set=1:size(RankingFullDataHeatF,3)
for Column=1:size(RankingFullDataHeatF,1)

AverageRanking(Column,Set)=mean(RankingFullDataHeatF(Column,:,Set),'omitnan')
%calculates the mean omitting NaN data points

end
end

figure('Name','Average Ranking Data Heat Map','NumberTitle','off');
heatmap(AverageRanking,'Title','Average Ranking Data Heat
Map','YLabel','Subject','Xlabel','Output','YData',SessionNames,'XData',nameOu
tput,'ColorbarVisible','off','ColorMethod','min') %Subject and Session can be
changed as desired

% The following can be uncommented to have their functions pop up
% automatically
% plotmf(Controller,'input',2) %Plots the membership functions of indicated
input
fuzzy(Controller) %opens controller
% ruleview(Controller) %opens ruleviewer to visualize rules
Acopy=SubjectMeanData{1,1}; %copy and paste data of interest into ruleview


function y = RankingFunction(A)
    [~, y] = ismember(A, sort(A,'descend'));
end
```

Appendix E

Police MATLAB controller Script

```
%documentation https://www.mathworks.com/help/fuzzy/referencelist.html

%Each subject visit should be a file. In that file, each sheet represents a
%session during the visit. Baselines are the first sheet and each sheet
%after will use those baselines. Within in the sheet, channels should be
%unique to columns, with each column representing a Brodmann area or
%channel as desired. A unique set of data is also acceptable

%remove files from folder as needed to adjust what data is compared against

close all %Closes all figures
clear all %Clears all variables
clc       %Clears the screen

%set to 0 or 1 with 1 being yes and 0 being no
MainFolder="T:\\PoliceMatlabData3\\"; %The folder that contains the files
Baselineloc=1; %set equal to column of baseline
RemoveBase=0; %set to 1 to remove base from average and max
calculations(reccomended)
BaseCalcs=0;%set to 1 to do calculations based on base
Exclude=0; %set equal to 1 to exclude the following dataset from the overall
analysis
Edata=5; %Excludes the file in this position from the data
EqualWeight=1; %Set equal to 1 for each session to have the same weight in
overall calculations
PerLow=15;%Sets the percentile for Data to be considered Low
PerMed=50;%Sets the percentile for Data to be considered Med
PerHigh=95;%Sets the percentile for Data to be considered High
InpOverlap=.5;%Sets the overlap of the Input variables

Controllername="FuzzyController";
Controller = mamfis('Name',Controllername); %initializes a mamdani fuzzy
controller variable

nameInput={"BA10","BA18","BA20","BA21"}; %names inputs
nameMF={"Low","Med","High"}; %names membership functions
input={"gaussmf","gaussmf","gaussmf"}; %Sets the type of membership function

% "gbellmf" Generalized bell-shaped membership function gbellmf
% "gaussmf" Gaussian membership function    gaussmf
% "gauss2mf"    Gaussian combination membership function    gauss2mf
% "trimf"   Triangular membership function  trimf
% "trapmf"  Trapezoidal membership function trapmf
% "sigmf"   Sigmoidal membership function   sigmf
% "dsigmf"  Difference between two sigmoidal membership functions   dsigmf
```

```
% "psigmf"  Product of two sigmoidal membership functions   psigmf
% "zmf" Z-shaped membership function    zmf
% "pimf"    Pi-shaped membership function   pimf
% "smf" S-shaped membership function    smf

%%%Outputs defined below
nameOutput={"Vision","Memory","Shape/Distance","Hearing/Sound","TheoryOfMind"
}; %names inputs

Controller = addOutput(Controller,[0 100],'Name',nameOutput{1}); %bracket
indicates range;section after name is used to name input
Controller = addMF(Controller,nameOutput{1},"trapmf",[-20 0 20
40],'Name',"Low");
Controller = addMF(Controller,nameOutput{1},"trapmf",[20 40 60
80],'Name',"Med");
Controller = addMF(Controller,nameOutput{1},"trapmf",[60 80 100
120],'Name',"High");

Controller = addOutput(Controller,[0 100],'Name',nameOutput{2}); %bracket
indicates range;section after name is used to name input
Controller = addMF(Controller,nameOutput{2},"trapmf",[-20 0 20
40],'Name',"Low");
Controller = addMF(Controller,nameOutput{2},"trapmf",[20 40 60
80],'Name',"Med");
Controller = addMF(Controller,nameOutput{2},"trapmf",[60 80 100
120],'Name',"High");

Controller = addOutput(Controller,[0 100],'Name',nameOutput{3}); %bracket
indicates range;section after name is used to name input
Controller = addMF(Controller,nameOutput{3},"trapmf",[-20 0 20
40],'Name',"Low");
Controller = addMF(Controller,nameOutput{3},"trapmf",[20 40 60
80],'Name',"Med");
Controller = addMF(Controller,nameOutput{3},"trapmf",[60 80 100
120],'Name',"High");

Controller = addOutput(Controller,[0 100],'Name',nameOutput{4}); %bracket
indicates range;section after name is used to name input
Controller = addMF(Controller,nameOutput{4},"trapmf",[-20 0 20
40],'Name',"Low");
Controller = addMF(Controller,nameOutput{4},"trapmf",[20 40 60
80],'Name',"Med");
Controller = addMF(Controller,nameOutput{4},"trapmf",[60 80 100
120],'Name',"High");

Controller = addOutput(Controller,[0 100],'Name',nameOutput{5}); %bracket
indicates range;section after name is used to name input
Controller = addMF(Controller,nameOutput{5},"trapmf",[-20 0 20
40],'Name',"Low");
Controller = addMF(Controller,nameOutput{5},"trapmf",[20 40 60
80],'Name',"Med");
Controller = addMF(Controller,nameOutput{5},"trapmf",[60 80 100
120],'Name',"High");
```

```
Controller.DefuzzificationMethod = "centroid"; %used to change
Defuzzification Method
% myFIS.ImplicationMethod = "customimp";
% myFIS.AggregationMethod = "customagg";
% myFIS.DTypeReductionMethod = "customtr";


% ==    IS (in rule antecedent)
% ~=    IS NOT
% & AND
% | OR
% =>    THEN
% = IS (in rule consequent)

%First rule needs to be 1 and then the rest can be end+1 in any order
rule(1)= "BA20~=Low & BA18==High => Vision=High";
rule(end+1)= "BA21==High & BA18==Med => Vision=High";
rule(end+1)= "BA21==Med & BA18==Med => Vision=Med";
rule(end+1)= "BA20==Med & BA18==Med => Vision=Med";
rule(end+1)= "BA20==Low & BA18==Low => Vision=Low";
rule(end+1)= "BA21==Low & BA18==Low => Vision=Low";
rule(end+1)= "BA20==Low & BA18==Low => Vision=Low";
rule(end+1)= "BA21==Low & BA20==Low => Vision=Low";



rule(end+1)= "BA10==Med & BA20==High => Memory=High";
rule(end+1)= "BA10==High & BA20==High => Memory=High";
rule(end+1)= "BA10==Med & BA20==Med => Memory=Med";
rule(end+1)= "BA20==Med & BA18==Med => Memory=Med";
rule(end+1)= "BA10==Low & BA18==Low => Memory=Low";
rule(end+1)= "BA20==Low & BA10==Low => Memory=Low";
rule(end+1)= "BA20==Low & BA18==Low => Memory=Low";


rule(end+1)= "BA18==High & BA21~=Low => Shape/Distance=High";
rule(end+1)= "BA18~=Low & BA21==High=> Shape/Distance=High";
rule(end+1)= "BA10==Med & BA18==Med => Shape/Distance=Med";
rule(end+1)= "BA10~=Low & BA18==Med => Shape/Distance=Med";
rule(end+1)= "BA20~=Low & BA18==Med => Shape/Distance=Med";
rule(end+1)= "BA20==Low & BA18~=High => Shape/Distance=Low";
rule(end+1)= "BA10==Low & BA18~=High => Shape/Distance=Low";


rule(end+1)= "BA10~=Low & BA20==High => Hearing/Sound=High";
rule(end+1)= "BA21~=Low & BA20==High => Hearing/Sound=High";
rule(end+1)= "BA10==High & BA20~=Low => Hearing/Sound=High";
rule(end+1)= "BA21==High & BA20~=Low => Hearing/Sound=High";
rule(end+1)= "BA10==Med & BA21==Med => Hearing/Sound=Med";
rule(end+1)= "BA10==Low & BA20==Low => Hearing/Sound=Low";
rule(end+1)= "BA10==Low & BA21==Low => Hearing/Sound=Low";
rule(end+1)= "BA20==Low & BA21==Low => Hearing/Sound=Low";
```

```
rule(end+1)= "BA20~=Low & BA21==High => TheoryOfMind=High";
rule(end+1)= "BA21~=Low & BA20==High => TheoryOfMind=High";
rule(end+1)= "BA10~=Low & BA20==High => TheoryOfMind=High";
rule(end+1)= "BA10~=Low & BA21==High => TheoryOfMind=High";
rule(end+1)= "BA20==Med & BA21==Med => TheoryOfMind=Med";
rule(end+1)= "BA20~=Low & BA10==Med => TheoryOfMind=Med";
rule(end+1)= "BA21~=Low & BA10==Med => TheoryOfMind=Med";
rule(end+1)= "BA21==Low & BA20==Low => TheoryOfMind=Low";
rule(end+1)= "BA21==Low & BA10==Low => TheoryOfMind=Low";
rule(end+1)= "BA20==Low & BA10==Low => TheoryOfMind=Low";

%
ruletest=(sprintf('"%s==%s%s',nameInput{1,inputs},nameMF{1,i},nameMF{1,i}));
%test for auto rule making

%The coding for inputs can be modified in the code below as desired.
%Everything else should run as is.

%%Start of code
Sessionlist=dir(MainFolder); %Makes a list of all files and data
Session=transpose({Sessionlist.name}); %Takes just the names of the files
SizeSession=size(Session); %Prepares the number of files
SessionT=SizeSession(1); %Sets equal to number of files

for i=3:SessionT %Starts to load files and skips "." and ".."
Filename=sprintf('%s\\%s',MainFolder,Session{i}); %Sets up file name
[status,sheets] = xlsfinfo(Filename);
sheetsTp=size(sheets); %Prepares number of sheets
sheetsT=sheetsTp(2); %Sets equal to number of sheets

for sheet=1:sheetsT
Data{i-2,sheet}=xlsread(Filename,sheet); %Loads excel data
end

end

ExData=Data;%sets up for excluding data
FullData=Data;%Saves full data in a variable
if Exclude==1
ExData(Edata,:)=[]; %exclues the data set defined above
Data=ExData; %Makes the excluded data the main data set
end

DataI=size(Data); %Gets information of Data size
RowT=DataI(1);  %Equal to the total rows
ColumnOT=DataI(2); %Overall total columns

BaselineOffsetData{RowT,ColumnOT}=[];%Sets up variable and prealoocated size
BaselineRealativeData{RowT,ColumnOT}=[];%Sets up variable and prealoocated
size
ColumnMassData=[]; %Sets up variable
ColumnMassEData=[]; %Sets up variable
NormalizedEData=[]; %Sets up variable

%Setting up to process individual data
```

```
for Row=1:RowT

ColumnTp=(~cellfun(@isempty,Data(Row,:))); %Determines total Columns in the
row
ColumnT=sum(ColumnTp); %Gathers the amount of columns used

for Column=1:ColumnT

columnDTp=size(Data{Row,Column}); %Determines total Columns in the cell
columnDT=columnDTp(2); %Gathers the amount of columns used

for columnData=1:columnDT

if BaseCalcs==1
Base=Data{Row,Baselineloc}; %Baselineloc column defined above
BaselineOffsetData{Row,Column}(:,columnData)=Data{Row,Column}(:,columnData)-
Base(:,columnData);%subtracts baseline
BaselineRealativeData{Row,Column}(:,columnData)=Data{Row,Column}(:,columnData
)/Base(:,columnData);%divides baseline
end
SubjectMaxData{Row,Column}(:,columnData)=max(Data{Row,Column}(:,columnData));
%Finds Max of Data
SubjectMinData{Row,Column}(:,columnData)=min(Data{Row,Column}(:,columnData));
%Finds Min of Data
SubjectMeanData{Row,Column}(:,columnData)=mean(Data{Row,Column}(:,columnData)
); %Finds Mean of Data
SubjectStdevData{Row,Column}(:,columnData)=std(Data{Row,Column}(:,columnData)
); %Finds Standard Deviation of Data
%More calculations can be added here as desired

end
end
end

if RemoveBase==1
NoBaseData=Data; %sets equal to data
NoBaseData(:,Baselineloc)=[]; %removes the defined column of baseline data
CalcData=NoBaseData; %prepares for calculations
SubjectMaxData(:,Baselineloc)=[]; %removes the defined column of baseline
data
SubjectMinData(:,Baselineloc)=[]; %removes the defined column of baseline
data
SubjectMeanData(:,Baselineloc)=[]; %removes the defined column of baseline
data
SubjectStdevData(:,Baselineloc)=[]; %removes the defined column of baseline
data
else
CalcData=Data; %prepares for calculations
end

if EqualWeight==1
CalcData=SubjectMeanData; %Sets the data to be calculated to be equal per
session
end
```

```
DataI=size(CalcData); %gets information of Data size
RowT=DataI(1);
%Setting up to get overall data from columns
for Row=1:RowT
ColumnTp=(~cellfun(@isempty,CalcData(Row,:))); %preps for data sets
ColumnT=sum(ColumnTp); %Gathers the amount of cells used
for Column=1:ColumnT
ColumnMassData=[ColumnMassData;CalcData{Row,Column}]; % amasses data into one
matrix
end
end

ColumnMassTp=size(ColumnMassData); %preps size
ColumnMassT=ColumnMassTp(2); %takes total columns

for ColumnM=1:ColumnMassT
DataMax(ColumnM)=max(ColumnMassData(:,ColumnM)); %Calculates Max of all data
DataMin(ColumnM)=min(ColumnMassData(:,ColumnM)); %Calculates Min of all data
DataMean(ColumnM)=mean(ColumnMassData(:,ColumnM)); %Calculates Mean of all
data
DataStdev(ColumnM)=std(ColumnMassData(:,ColumnM)); %Calculates Standard
deviation of all data
DataMedL(ColumnM)=prctile(ColumnMassData(:,ColumnM),PerLow); %Calculates
Standard deviation of all data
DataMedM(ColumnM)=prctile(ColumnMassData(:,ColumnM),PerMed); %Calculates
Standard deviation of all data
DataMedH(ColumnM)=prctile(ColumnMassData(:,ColumnM),PerHigh); %Calculates
Standard deviation of all data
%More calculations can be added here as desired
end

LimitHigh=DataMax; %Sets the high limit to the max
LimitLow=DataMin; %Sets the low limit to the min

% Variable size based on defined and avg and sdx2 or sdx3 etc.
%%%% This can be changed to better reflect the data as desired
for inputs=1:size(nameInput,2)
Controller = addInput(Controller,[LimitLow(inputs)
LimitHigh(inputs)],'Name',nameInput{1,inputs}); %bracket indicates
range;section after name is used to name input
InputSize={[DataStdev(inputs)*InpOverlap
DataMedL(inputs)],[DataStdev(inputs)*InpOverlap
DataMedM(inputs)],[DataStdev(inputs)*InpOverlap DataMedH(inputs)]}; % make
these based off of data Avg overlap etc.
% InputSize={[DataStdev(inputs) DataMedL(inputs)-
(1*DataStdev(inputs))],[DataStdev(inputs)
DataMedM(inputs)],[DataStdev(inputs)
DataMedH(inputs)+(1*DataStdev(inputs))]}; %Adds standard dev to above for
bigger overlap
for i=1:size(input,2) %Adds the MF to each rule
Controller =
addMF(Controller,nameInput{1,inputs},input{1,i},InputSize{1,i},'Name',nameMF{
1,i});
end
end
```

```
%If the following line gives an error check spelling and syntax of rules and
inputs and outputs
Controller = addRule(Controller,rule); %adds the rules from above now that
inputs are defined


for Row=1:RowT

ColumnTp=(~cellfun(@isempty,Data(Row,:))); %Determines total Columns in the
row
ColumnT=sum(ColumnTp); %Gathers the amount of columns used

for Column=1:ColumnT

columnDTp=size(Data{Row,Column});
columnDT=columnDTp(2);


for columnData=1:columnDT

SubjectEvalData{Row,Column}=evalfis(Controller,(SubjectMeanData{Row,Column}))
; %evaluates the data based on the data inputs

end
end
end

SessionNames=Session(3:SessionT);

if Exclude==1
SessionNames(Edata,:)=[]; %excludes the data set defined above
end

FullOutput=[SessionNames,SubjectEvalData];

%Setting up to get overall data from Eval columns
DataE=size(SubjectEvalData); %gets information of Data size
RowET=DataE(1);
for RowE=1:RowET
ColumnETp=(~cellfun(@isempty,SubjectEvalData(RowE,:))); %preps for data sets
ColumnET=sum(ColumnETp); %Gathers the amount of cells used
for ColumnE=1:ColumnET
ColumnMassEData=[ColumnMassEData;SubjectEvalData{RowE,ColumnE}]; % amasses
data into one matrix
end
end

ColumnMassETp=size(ColumnMassEData); %preps size
ColumnMassET=ColumnMassETp(2); %takes total columns


% Additional calculations can be added here, if desired
for ColumnEM=1:ColumnMassET
```

```
DataEMax(ColumnEM)=max(ColumnMassEData(:,ColumnEM)); %Calculates Max of
Evaluations
DataEMin(ColumnEM)=min(ColumnMassEData(:,ColumnEM)); %Calculates Min of
Evaluations
DataEMean(ColumnEM)=mean(ColumnMassEData(:,ColumnEM)); %Calculates Mean of
Evaluations
DataEStdev(ColumnEM)=std(ColumnMassEData(:,ColumnEM)); %Calculates Standard
deviation of Evaluations

end

DataEFullMean=mean(ColumnMassEData,'all'); %Calculates Mean of the full
Matrix
DataEFullStdev=std(ColumnMassEData,0,'all'); %Calculates Standard deviation
of the full Matrix

for RowE=1:RowET %Prepares and normalizes the data

ColumnTp=(~cellfun(@isempty,SubjectEvalData(RowE,:))); %Determines total
Columns in the row
ColumnT=sum(ColumnTp); %Gathers the amount of columns used

for Column=1:ColumnT

columnDTp=size(SubjectEvalData{RowE,Column});
columnDT=columnDTp(2);

for columnData=1:columnDT

NormalizedEData{RowE,Column}(:,columnData)=(SubjectEvalData{RowE,Column}(:,co
lumnData)/DataEMax(:,columnData)*100);%Normalizes Data
ZscoreEData{RowE,Column}(:,columnData)=((SubjectEvalData{RowE,Column}(:,colum
nData)-DataEMean(:,columnData))/DataEStdev(:,columnData));%Finds z-score of
Data
ZscoreEFullData{RowE,Column}(:,columnData)=((SubjectEvalData{RowE,Column}(:,c
olumnData)-DataEFullMean)/DataEFullStdev);%Finds z-score of Data

end
end
end

FullOutputNorm=[SessionNames,NormalizedEData];
MatrixNames=[];
MatrixNumbers=[];
DataE=size(FullOutputNorm); %gets information of Data size
RowET=DataE(1);

for RowE=1:RowET %Prepares and takes the distance between the data

ColumnTp=(~cellfun(@isempty,FullOutputNorm(RowE,:))); %Determines total
Columns in the row
ColumnT=sum(ColumnTp); %Gathers the amount of columns used

for Column=2:ColumnT
```

```
MatrixNames=[MatrixNames,FullOutputNorm(RowE,1)];
MatrixNumbers=[MatrixNumbers,FullOutputNorm(RowE,Column)];

end
end

MatrixNamesU= string(MatrixNames) + (1:numel(MatrixNames)); %makes names
unique for heatmap
% cellstr(MatrixNamesU) % Can be used to change it back if needed

DataE=size(MatrixNumbers); %gets information of Data size
RowET=DataE(1);

for RowE=1:RowET %Prepares and takes the distance between the data

ColumnTp=(~cellfun(@isempty,MatrixNumbers(RowE,:))); %Determines total
Columns in the row
ColumnT=sum(ColumnTp); %Gathers the amount of columns used

for Column=1:ColumnT

columnDTp=size(MatrixNumbers{RowE,Column});
columnDT=columnDTp(2);
Dist1=MatrixNumbers{RowE,Column}; %Changes Dist1 for big loops

for Column2=1:ColumnT

Dist2=MatrixNumbers{RowE,Column2}; %Changes Dist2 for smaller loops
Square=[]; %empties square before loop

for columnData=1:columnDT

x2=Dist2(:,columnData); %Grabs numbers from cell based on iteration
x1=Dist1(:,columnData); %Grabs numbers from cell based on iteration
Square(columnData)=(x2-x1)^2; %takes the square of the difference

end
Sum=sum(Square); %takes the sum of the parts
Sqrt=sqrt(Sum); %takes the square root of the parts
% example sqrt( (x2-x1)^2+(y2-y1)^2+(z2-z1)^2 +(a2-a1)^2 +(b2-b1)^2
DistMatrixP(Column,Column2)=Sqrt;
end
end
end
DistMatrix=num2cell(DistMatrixP);
FullDistMatrixP=[MatrixNamesU;DistMatrix];
LeftsideP=transpose(MatrixNamesU); %prepares leftside
Leftside=[0;LeftsideP]; %prepares leftside
FullDistMatrix=[Leftside,FullDistMatrixP];
figure('Name','Distance Heat Map','NumberTitle','off');
heatmap(MatrixNamesU,LeftsideP,DistMatrixP)
sgtitle('Distance Heat Map');

%Makes the heatmaps for the figures named
len=length(ZscoreEData{1,1});
```

```
ZscoreEData(cellfun(@isempty,ZscoreEData)) = {nan(1,len)};
ZscoreEDataHeat = cell2mat(ZscoreEData); % Convert to array
figure('Name','Zscore Local Heat Map','NumberTitle','off');

for Split = 1:len
ZscoreEDataHeatF(:,:,Split) = ZscoreEDataHeat(:,Split:len:end); % split up
columns by page
end

for Map = 1:size(ZscoreEDataHeatF,3)

nexttile %makes the next map show in the same window
heatmap(ZscoreEDataHeatF(:,:,Map),'Title',Controller.Outputs(1,Map).Name,'YLa
bel','Subject','Xlabel','Session','YData',SessionNames) %Subject and Session
can be changed as desired

end
sgtitle('Zscore Local Heat Map');

%Makes the heatmaps for the figures named
len=length(ZscoreEFullData{1,1});
ZscoreEFullData(cellfun(@isempty,ZscoreEFullData)) = {nan(1,len)};
ZscoreEFullDataHeat = cell2mat(ZscoreEFullData); % Convert to array
figure('Name','Zscore Global Heat Map','NumberTitle','off');

for Split = 1:len

ZscoreEFullDataHeatF(:,:,Split) = ZscoreEFullDataHeat(:,Split:len:end); %
split up columns by page

end

for Map = 1:size(ZscoreEFullDataHeatF,3)
nexttile %makes the next map show in the same window
heatmap(ZscoreEFullDataHeatF(:,:,Map),'Title',Controller.Outputs(1,Map).Name,
'YLabel','Subject','Xlabel','Session','YData',SessionNames) %Subject and
Session can be changed as desired
end
sgtitle('Zscore Global Heat Map');

%Makes the heatmaps for the figures named
len=length(NormalizedEData{1,1});
NormalizedEData(cellfun(@isempty,NormalizedEData)) = {nan(1,len)};
NormalizedEDataHeat = cell2mat(NormalizedEData); % Convert to array
figure('Name','Normalized Data Heat Map','NumberTitle','off');


for Split = 1:len

NormalizedEFullDataHeatF(:,:,Split) = NormalizedEDataHeat(:,Split:len:end); %
split up columns by page

end

for Map = 1:size(NormalizedEFullDataHeatF,3)
```

```
nexttile %makes the next map show in the same window
heatmap(NormalizedEFullDataHeatF(:,:,Map),'Title',Controller.Outputs(1,Map).N
ame,'YLabel','Subject','Xlabel','Session','YData',SessionNames) %Subject and
Session can be changed as desired

end
sgtitle('Normalized Data Heat Map');


%Makes the heatmaps for the figures named
Ranking = NormalizedEData;
idx = cellfun(@(x) any(isnan(x)),Ranking);
Ranking(~idx) = cellfun(@RankingFunction, Ranking(~idx), "UniformOutput",
false);
len=length(Ranking{1,1});
Ranking(cellfun(@isempty,Ranking)) = {nan(1,len)};
RankingDataHeat = cell2mat(Ranking); % Convert to array
figure('Name','Ranking Data Heat Map','NumberTitle','off');


for Split = 1:len

RankingFullDataHeatF(:,:,Split) = RankingDataHeat(:,Split:len:end); % split
up columns by page

end

for Map = 1:size(RankingFullDataHeatF,3)
nexttile %makes the next map show in the same window
heatmap(RankingFullDataHeatF(:,:,Map),'Title',Controller.Outputs(1,Map).Name,
'YLabel','Subject','Xlabel','Session','YData',SessionNames,'ColorbarVisible',
'off','ColorMethod','min') %Subject and Session can be changed as desired

end
sgtitle('Ranking Data Heat Map');


for Set=1:size(RankingFullDataHeatF,3)

for Column=1:size(RankingFullDataHeatF,1)

AverageRanking(Column,Set)=mean(RankingFullDataHeatF(Column,:,Set),'omitnan')
%calculates the mean omitting NaN data points

end

end

figure('Name','Average Ranking Data Heat Map','NumberTitle','off');
heatmap(AverageRanking,'Title','Average Ranking Data Heat
Map','YLabel','Subject','Xlabel','Output','YData',SessionNames,'XData',nameOu
tput,'ColorbarVisible','off','ColorMethod','min') %Subject and Session can be
changed as desired
```

```
% The following can be uncommented to have their functions pop up
% automatically
% plotmf(Controller,'input',2) %Plots the membership functions of indicated
input
fuzzy(Controller) %opens controller
% ruleview(Controller) %opens ruleviewer to visualize rules
Acopy=SubjectMeanData{1,1}; %copy and paste data of interest into ruleview


function y = RankingFunction(A)
    [~, y] = ismember(A, sort(A,'descend'));
end
```

Appendix F

IRB Certification

THE UNIVERSITY OF ALABAMA | Office of the Vice President for Research & Economic Development
Office for Research Compliance

May 31, 2017

Isaac C. Heim, MS
Department of Mechanical Engineering
College of Engineering
University of Alabama
Box 870276

Re: IRB # EX-17-CM-037 "Statistical Analysis of Archived Data Pertaining to tDCS in Math Performance"

Dear Mr. Heim:

The University of Alabama Institutional Review Board has granted approval for your proposed research. Your protocol has been given exempt approval according to 45 CFR part 46.101(b)(4) as outlined below:

(4) Research involving the collection or study of existing data, documents, records, pathological specimens, or diagnostic specimens, if these sources are publicly available or if the information is recorded by the investigator in such a manner that subjects cannot be identified, directly or through identifiers linked to the subjects.

Your application will expire on May 30, 2018. If your research will continue beyond this date, complete the relevant portions of Continuing Review and Closure From. If you wish to modify the application, complete the Modification of an Approved Protocol Form. When the study closes, complete the appropriate portions of FORM: Continuing Review and Closure.

Should you need to submit any further correspondence regarding this proposal, please include the assigned IRB application number.

Good luck with your research.

Sincerely,

Carpanato T. Myles, , IM, CIP
Director & Researc ompli nce Officer
Office for Research Compliance

358 Rose Administration Building | Box 870127 | Tuscaloosa, AL 35487-0127
205-348-8461 | Fax 205-348-7189 | Toll Free 1-877-820-3066