Implementation of Machine Learning Algorithm to Exploit Information from Multimodal fMRI/EEG Fused Image Data

AUSTRIAN
MARSHALL PLAN FOUNDATION
VIENNA | AUSTRIA

Research Report

Author:

Dalton H Bermudez Cabrera

Investigation performed in

Technical University of Graz

Department of Biomedical Engineering

Supervised by

Thomas Pock, PhD

Institute of Computer Vision and Graphics

Collaboration with

David Steyrl, M.Sc.

Gernot Müller-Putz, PhD.

Graz University of Technology

Institute of Neural Engineering

TABLE OF CONTENTS

Acknowledgement:

I would like to thank the Austrain Marshall Plan Foundation for offering the opportunity to do research abroad. The experience helped me mature both professionally and personally. It offered me the opportunity to collaborate first hand with professionals in the fields of Computer Vision and Neuro-Engineering.

I would like to give thanks to David Steyrl, Gernot Müller-Putz, and Thomas Pock for aiding me in the technicality of these project. I also thank them for initiating the collaboration between both Computer Vision and Neuro-Engineering institutes, which was fundamental for the conceptual development of the project that is presented in this paper.

Special thanks to David Steryl who facilitated the data used in the research report, as well as, for providing me with knowledge on the limitations of analyzing the provided data set. Also, I am truly grateful for the guidance he gave me during the different stages of this project.

I would like to thank my family for their unconditional support. They have always encouraged me to keep on fighting during troublesome times and for that I am truly grateful. Thanks to my parents for giving me guidance throughout my life. They helped me realize that the significance of life is, simply, what one makes of it. Finally, I could not conclude without mentioning my older sister, who has been the main role model for me. She has been one of the main reason for my decision to potentially pursue a doctorate in the field of Computational Neuroscience/Neuro-Imaging, for which I will be forever grateful.

Abstract:

Over the years, neuroimaging modalities such as Electroencephalogram (EEG) and Function Magnetic Resonance Imaging (FMRI) have been used in the field of neuroscience to study neuronal activity at the macroscale. Each of these modalities have their own limitations when it comes to the information that can be extracted from them. To reduce those limitations, researchers have developed multimodal systems that integrate the information that can be acquired from individual modalities. A current multimodal neuroimaging modality that has been gained popularity is Simultaneous EEG-FMRI. The technique can deliver both the high spatial resolution of the FMRI with high temporal resolution of the EEG. However, it is particularly difficult to acquire a good quality EEG signal within the strong magnetic field of the FMRI since the presence of various artifact, when recording, can obscure the EEG signal. Many technicques are currently being employed to reduce these artifacts. The first objective of this study is to investigate how the FMRI bold signal is correlated with EEG activity. The second objective is to examine if implementing a Machine Learning based artifact removal technique yields tangible results for recovering EEG activity. The third goal is to determine the degree at which the EEG signal quality can influence the classification of different EEG brain waves. Although, there appears to be no significant effects in the classification accuracies ($p < 0.05$), the used method shows to reduce the power of the PA artifact. Therefore, further investigation needs to be conducted on a larger data set to fully validate or invalidate the technique.

Keywords: Support Vector Machine (SVM), Gaussian Mixture Model-Expectation Maximization (GMM-EM), Average Artifact Subtraction (AAS), Gradient Artifact (GA), Pulsatile Artifact (PA), Functional Magnetic Resonance Imaging (FMRI), Electroencephalogram (EEG)

Introduction:

The conjoined field composed of Biomedical Engineering, Computational Neuroscience, and Neuroscience has led to the development of various non-invasive techniques to explore underlining neuronal activity [4]. One such technique, often referred to as Simultaneous EEG-FMRI, involves the simultaneous acquisitions of EEG and FMRI.

EEG-FMRI has the capacity to identify electrophysiological activity as well as hemodynamic fluctuations, while providing good spatial-temporal resolution [2][3][4]. The EEG are reflections of the neuronal activity occurring in deep brain structures and on the surface of the brain [30]. This technique provides spatial and temporal resolutions on the order of centimeter and milliseconds, respectively [30]. On the other hand, FMRI measures the fluctuations in blood oxygen concentration levels (BOLD changes) that are being supplied to different brain regions [8][9]. The BOLD signal gives insight on the neuronal activity by relying on the fact that cerebral blood flow and neuronal activation are coupled [8][9]. The FMRI can give spatial and temporal resolution on the order of millimeters and seconds, respectively [1][7]. Simultaneous EEG-FMRI can reduce the disadvantages when using each technology by itself. However, the high magnetic field of the FMRI undermines the quality of the recorded EEG by causing the occurrences of various artifacts in the EEG signal. Two of the main studied artifacts in simultaneous EEG recoding are the Gradient Artifact (GA) and the Pulsatile Artifact (PA) [2][3][29]. The GA, also referred to as the Imaging Artifact, are present due to the rapid changes in magnetic field gradients [2]. While the PA artifact is caused by the motion of the EEG electrodes within the static magnetic field, or B_0 field, due to the pulsatile blood flow coupled with the cardiac activity of the participant [3][29]. On the other hand, artifacts present in the FMRI due to the presence of the EEG can be neglected [18]. The artifacts in the FMRI are mainly present due to the radiofrequency (RF) interaction with the conductive wires of the EEG [18]. However, these artifacts have been found to affect the BOLD signals of the FMRI by less than 1% [18]. Therefore, since similar BOLD affects can be detected, with or without the presence of the EEG system, [18] one can undergo the same processing stages as that of a regular FMRI.

1.1 Purpose of Study:

This paper will introduce the main methodologies that are currently being used to process simultaneous EEG-FMRI and potential techniques that can further improve the processing of EEG-FMRI data. Over the years, a desire to integrate machine learning and deep learning algorithms has upraised. In fact, current research deals with incorporating deep learning algorithms to predict a subject's observation from either their electrophysiological or FMRI bold signal activity [5]. This research will incorporate both unsupervised and supervised machine learning on the simultaneous EEG recording.

The purpose of these report can be consolidated in three different stages. The first stage deals with determining which voxels are most correlated to the time of stimulus performed for the given behavioral paradigm. The second stage involves the exploration of techniques for removing artifacts present in the simultaneous EEG recording. The implemented techniques for the removal of the GA and the PA Artifact were Average Artifact Subtraction (AAS_{GA}) and unsupervised GMM-EM $_{\text{Wiener Filter}}$, respectively. The third stage is to explore how to properly classify distinct brain waves from an EEG recording utilizing machine learning algorithms. This study focuses on the technique known as supervised SVM to classify Motor Imagery from both, EEG acquired outside and inside the scanner.

1.2 Significance of Study:

Simultaneous EEG-FMRI have been used to investigate various electrophysiological activities such as alpha rhythms, Event Related Potentials (ERP), and Evoked Potential (EP) [30]. The removal of the PA artifact from the EEG is essential to unveil these distinct electrophysiological activities. The removal of the PA artifact is particularly important in the detection of alpha rhythms since the PA artifact lies within the same frequency range as that of the alpha waves [30]. The recuperation of the alpha rhythms is essential for the generation of significance maps, for the FMRI images because those maps reflect EEG-FMRI interaction [29]. This interaction enables the to study of how different brainwaves in the EEG are associated with the acquired FMRI Bold signal [29]. A statistical method used to implement voxel-based correlation to examine the EEG-FMRI interactons is the Generalize Linear Model (GLM) [1]. Extracted EEG features are convolved with the Hemodynamic Response Function (HRF) to obtain the regressor for the GLM but prior removal of massive artifacts is necessary to perform this technique using Simultaneous EEG data.

Different signal processing techniques have been employed to reduce massive artifacts. The most commonly used method to reduce GA and PA artifacts is Average Artifact Subtraction (AAS) [29]. The effectiveness of the method relies on the repetitive nature of the waveforms from both the GA and PA artifacts [2][3][12][13]. This makes the technique ideal for the removal of the GA. The AAS_{GA} subtracts a gradient artifact template from the original EEG signal [2][12]. The gradient artifact template is obtained by averaging various intervals (epochs) in the EEG signal were either a set of FMRI slices or a set of FMRI volumes were acquired [2][12][29]. Despite the effectiveness of the AAS to reduce the GA, it is ineffective at completely suppressing the PA artifact [29]. This is mainly due to the time varying nature of both the artifact's period and waveform [3]. Therefore, an interest has risen for the development of new sophisticated algorithm for the removal of the PA artifact. The algorithm for the PA artifact removal employ in this study is a time-varying multichannel Weiner filter with a prior learned utilizing GMM-EM.

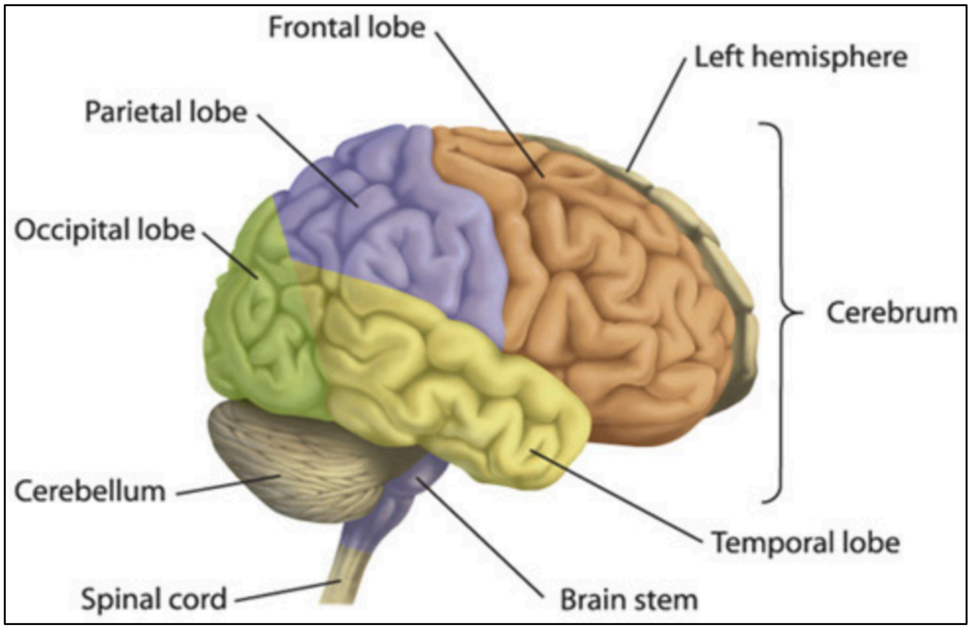
Neuro-Imaging Background:

This section describes the underlining physics of each of the individual modalities within the multimodal simultaneous EEG-FMRI. The first subsection introduces the basic biology of brain anatomy and functionality. The second subsection deals with the basic principles of EEG recoding, the different brain signals that are generated from it and their interpretation. The third subsection describes the functionality of the FMRI, as well as, the biological implications of FMRI. The final subsection focuses on the principles of simultaneous EEG-FMRI, as well as, the two major studied artifacts encounter in the simultaneous EEG recording.

2.1 Brain Anatomy:

The brain is an organ with the complex function of giving rise to consciousness and processing sensory input. These two main functions are processed by different brain areas which can be divided into three functional groups sensory maps (retina, hair cells in the ear, olfactory bulb), primary sensory cortex (such as primary visual cortex) and higher order neocortex (such as the parietal area, prefrontal cortex) [30]. Overall, the brain is composed of different regions such as the cerebral cortex, brain stem, the cerebellum, basal ganglia, thalamus and the hypothalamus as depicted in figure 2.1.

The Cerebral Cortex consists of the left and right hemispheres, which are interlinked by the corpus callosum [29][30]. The two hemispheres are composed of the Frontal lobe, Temporal lobe, Parietal lobe, and the Occipital lobe [29][30]. Each of these regions are responsible for the processing of different sensory information. The Frontal lobe focuses on carrying out higher order thinking, decision making, and planning [29][30]. The Temporal lobe functions in the processing of auditory information from the ears [29][30]. The Occipital lobe carries out processing of visual information obtained from the eyes [29][30]. The Parietal lobe processes information from the limbs [29][30]. This report aims at examining, non-invasively, the activity from the Parietal lobe.



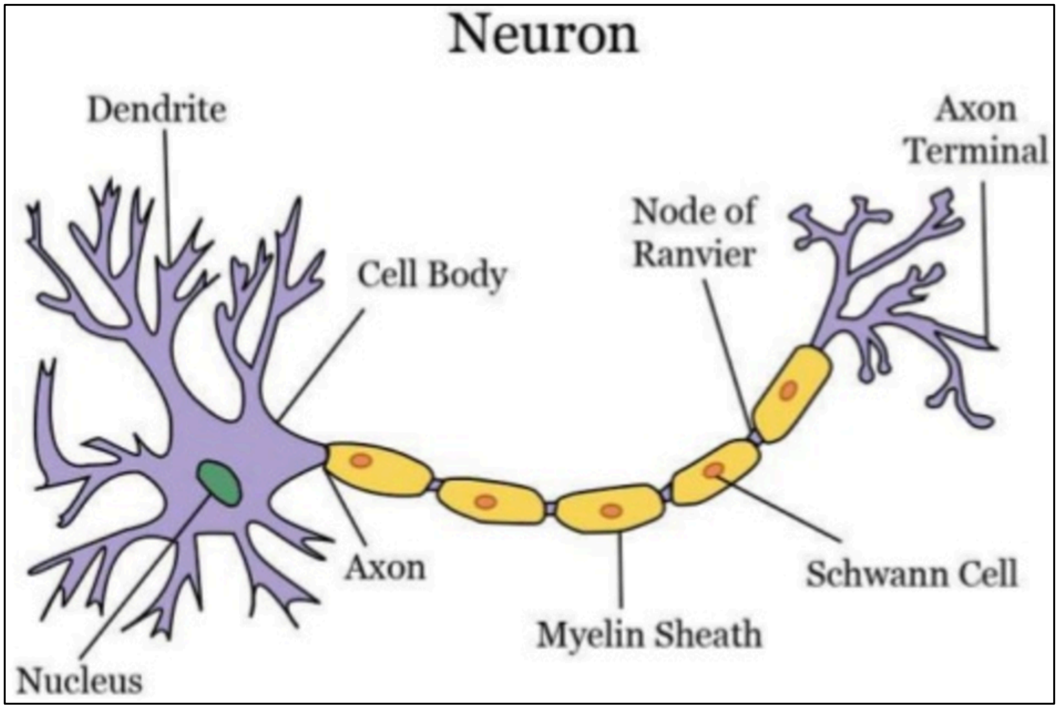
Sources: https://inside-the-brain.com/2013/03/07/what-is-attention-and-where-is-it-in-the-brain/

Figure 2.1: Brain Parts

2.2 Neuronal Network Structure and Function:

The brain is composed of various cell called neurons. The brain is often described by its gray and white matter components [6]. The cerebral cortex is considered to be gray matter due the strong presence of a number of neuronal cell bodies [6]. On the other hand, white matter consists of all the interconnected myelinated axons of a neuron that often travel in bundles across the brain to reach different brain areas [6].

Nerve impulses transmit information to and from distinct brain regions to activate and/or deactivate these brain regions [30]. A neuron consists of cell body known as the soma, dendrites, axon, and axon terminals as depicted in Figure 2.2. The function of an axon is to propagate the voltage changes all the way to the terminal where it can release neurotransmitters onto neighboring neurons, muscles and other body organs [30]. To enable a faster signal transmission in the form of voltage changes, axons are covered by a myelin sheath which serve as insulators [30]. Axons act as conduction pathway for electrical activity. The axon of a single neuron may form attachments (or synapses) with as many as 1000 distinct neurons [29][30]. Between neurons, the activity is transmitted from the presynaptic neuron to the postsynaptic neuron through the site known as the synapse [29][30]. The electrical activity is driven by ionic gradients established by differences of anion and cation concentrations inside and outside the neuron. The occurrence of a neuron activation depends on action potentials and neurotransmitters.



Sources:http://zoologicalzone.blogspot.com/2013/12/general-structure-of-neuron-axons-vs.html

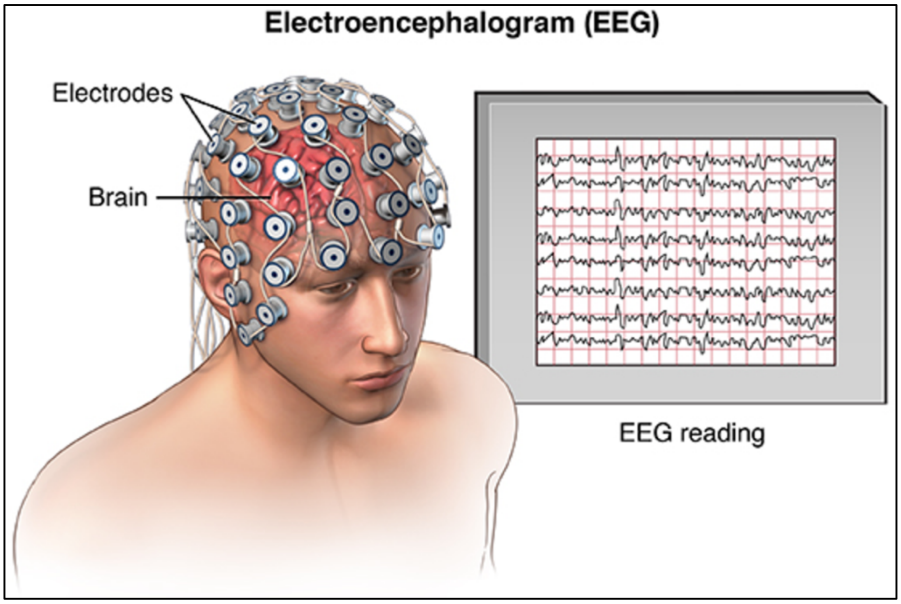
Figure 2.2: Nerve Cell Structure

2.3 Fundamentals of an Action Potential:

Action potentials are regarded as all or non-responses, meaning a neuron can only fire if a specific ionic gradient (Voltage difference) is achieved [30]. The steady state potential (resting state) of a nerve cell is approximately -65 mV [30]. The potential need for a neuron to achieve an action potential is approximately -55 mV [30]. Once a neuron depolarization reaches the threshold value of -55 mV then the neuron fires an action potential and it is said to be active.

2.4 Electroencephalogram (EEG):

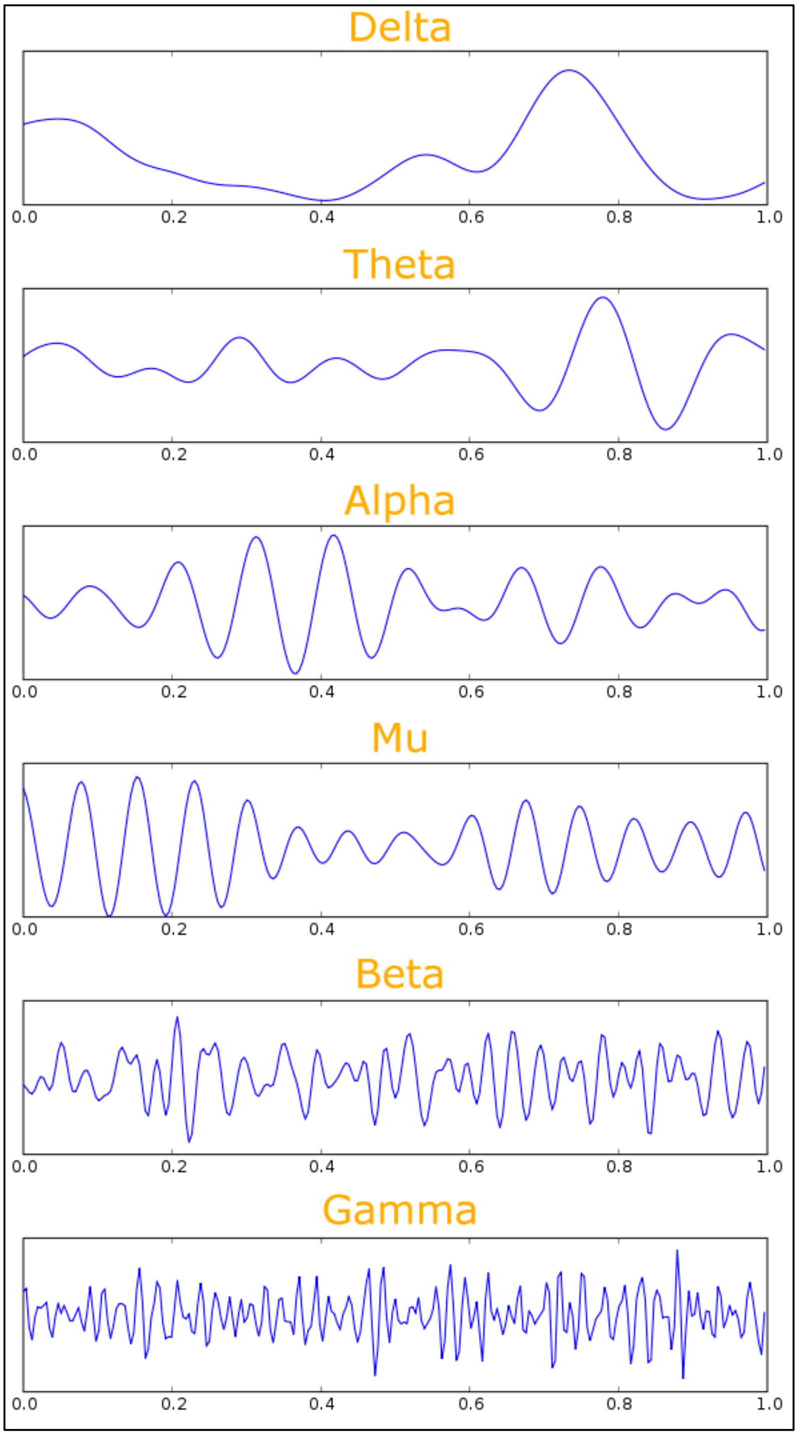
The electroencephalogram is a non-invasive modality that measures brain activity using electrodes that are placed in different areas of the scalp (Figure 2.3). The captured EEG signal is a collection of electric voltage fields generated from various clusters of neurons [30]. Electric voltage fields (or local field potentials) are the sum of overlapping currents that flows from the intracellular to extracellular space and vice versa wheather neurons fire synchronously [30]. The EEG activity is the conduction from the sum projections of the various local field potentials. Therefore, it can be said that scalp EEG is simply described by a radon transform of the various local field potentials.



Sources: https://speakingofresearch.com/tag/eeg/

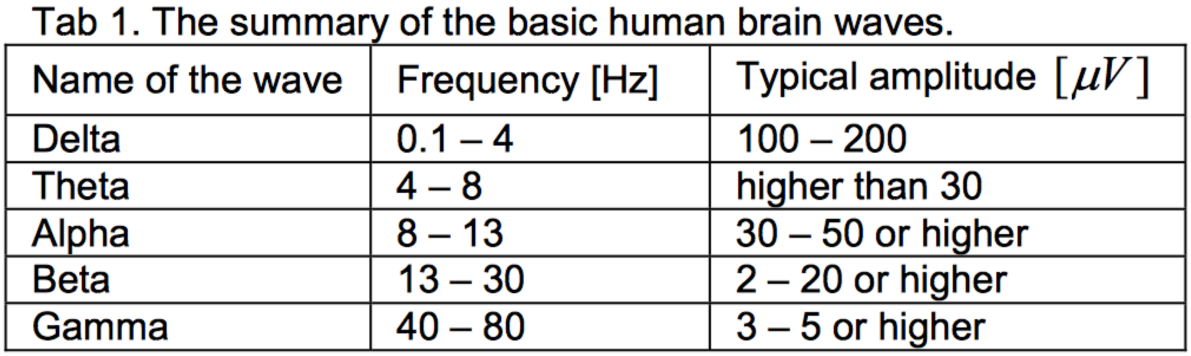
Figure 2.3: Electroencephalogram

The amplitude of the brain waves in the EEG is determined by the quantity of neurons and fibers that fire synchronously to one another [30]. The temporal resolutions of EEG on the order of milliseconds makes it a perfect tool to measure short duration signals. However, since EEG is only the sum projections of various local field potentials (LFP) sources, it has a poor spatial resolution. The ability to read LFPs and correlate it with different brain states lies in the cyclical nature of voltage changes. EEG LFP signals ranges between frequencies of 0.1 – 100 cycles per second (Hz) [4][30]. Researchers have been able to show that specific frequency ranges that relate to either motor, sleep, cognitive behaviors depend on which brain areas are being recruited and working together to drive such behaviors. The brain waves are usually classified in six main categories, the Delta (δ), Theta (θ), Alpha (α), Beta (β), and Gamma (γ) as shown in Figure 2.4 [30]. Each category can be differentiated by both amplitude and frequency ranges and they might be pronounced during specific behaviors (Table 2.5) [30]. The Delta wave is strong during deep sleep [30]. The Theta wave is associated with early stages of sleep [30]. Alpha wave is present during a relaxed state of mind. Beta wave is an indication of awareness and focus attention [30]. The Gamma wave is related to being in a state of high perception [30]. Although, this are the main brain waves there are others such as the mu (μ) brain wave. The frequency range of the mu wave is the same as the alpha wave. The mu (μ) brain wave reflects synchronous firing of pyramidal neurons of the motor cortex [4][30]. The mu waves are present during sensorimotor cortex activity and this signal is often used to decode movements for brain machine interface prosthetic devices [4][11][30]. In the context of this study the brain wave of interest is a superposition of the Alpha, Beta, and Mu brain waves.



Sources: http://wolfcrow.com/blog/notes-by-dr-optoglass-motion-and-the-frame-rate-of-the-human-eye/

Figure 2.4: Brain Waves in EEG



Tab 1. The summary of the basic human brain waves.

Name of the wave	Frequency [Hz]	Typical amplitude $[\mu V]$
Delta	0.1 – 4	100 – 200
Theta	4 – 8	higher than 30
Alpha	8 – 13	30 – 50 or higher
Beta	13 – 30	2 – 20 or higher
Gamma	40 – 80	3 – 5 or higher

Sources: http://slideplayer.com/slide/5855779/

Table 2.5: Frequency and Amplitudes of Brain Waves

2.5 Magnetic Resonance Imaging (MRI):

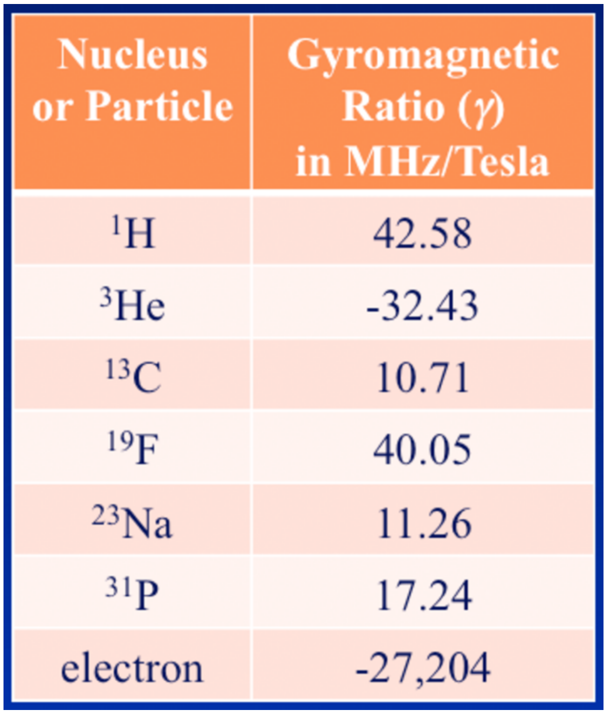
Magnetic Resonance Imaging is an imaging modality that exploits the nuclear spin resonance of atoms in the body. The MRI relies on the concept that when a magnet is exposed to an external magnetic field (B_0 field), it tends to orient itself in the same direction as the external magnetic field. The first stage of MRI acquisition is to expose the participant to the static magnetic field called the B_0 field. This causes the hydrogen atoms to align either parallel or anti-parallel to the B_0 field. The second stage is to send a Radio-Frequency (RF) pulse with the same frequency as that of particles spin to make their procession synchronous to each other. The frequency needed to make the atoms' phase aligned is determined by the Larmor frequency. The Larmor frequency is proportional to the gyromagnetic ratio and the magnetic field strength of the MRI (Equation 2.6) [29]

$$\omega_0 = \gamma B_0$$

Sources: https://www.hdiac.org/node/1910

Equation 2.6: Larmor Equation

The gyromagnetic ratio (MHz/T) of a particle is a measure of the ratio between the particles magnetic moment to its angular momentum (ω) [29]. Different particles used in MRI have different gyromagnetic ratios (Table 2.7). B_0 field is the magnitude of the static magnetic field in the MRI.



Nucleus or Particle	Gyromagnetic Ratio (γ) in MHz/Tesla
^{1}H	42.58
^{3}He	-32.43
^{13}C	10.71
^{19}F	40.05
^{23}Na	11.26
^{31}P	17.24
electron	-27,204

Sources: http://mriquestions.com/gyromagnetic-ratio-gamma.html

Table 2.7: Gyromagnetic ratio of different particles

The final stage is to introduce a gradient magnetic field, which varies proportionally in space. The gradient magnetic field causes predictable variations in the Larmor frequencies given off by the synchronous processions of the particle of interest [29]. Therefore, one can determine where in the human body a given frequency comes from. By introducing a gradient magnetic field, the Larmor equations becomes Equation 2.8 [29].

$$\omega(\mathbf{r},t) = \omega_0 + \gamma\mathbf{G}(t)\cdot\mathbf{r}$$

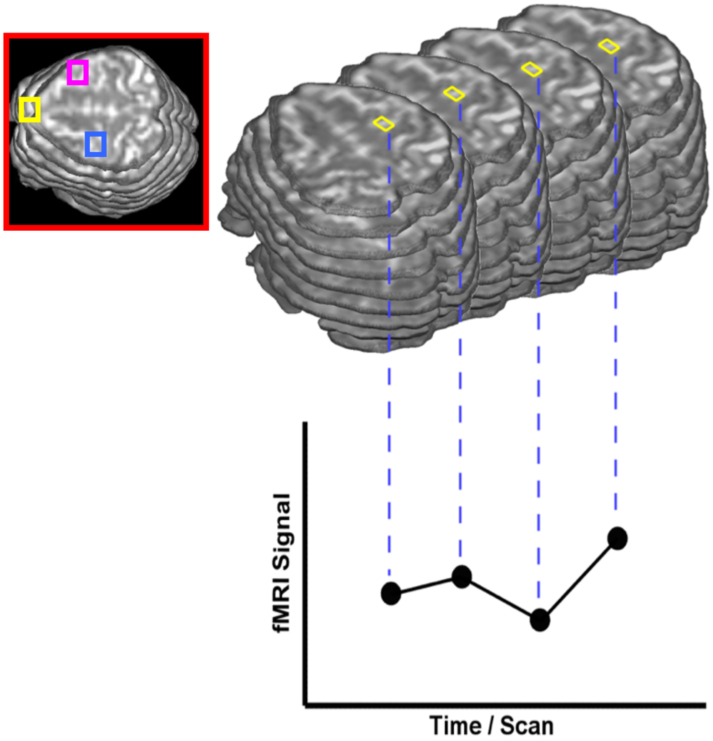
Sources: https://www.hdiac.org/node/1910

Equation 2.8: Modify Larmor equation

The G(t) term is the magnetic field gradient which causes predictable frequency fluctuations of the signal. This basic template has been used to obtain MRI images with excellent soft tissue contrast [29]. For instance, this property of MRI enables radiologist to observe difference in the nuclear relaxation times of health and cancerous tissue [29].

2.6 Functional Magnetic Resonance Imaging (fMRI):

Functional Magnetic Resonance Imaging (fMRI) is a popular imaging modality in the field of neuroscience. This tool enables the study of neuronal activity by relying on the bidirectional interaction between both neuronal and hemodynamic activities [1][7][8][10]. fMRI detects Blood Oxygen Dependent changes (BOLD) to infer on neural activity [1][7][8][10]. The magnetic properties of cerebral blood are determined by the oxygen concentration attached to hemoglobin. During neuronal metabolic activation, oxygen rich hemoglobin supplies the neurons with oxygen necessary for their survival. The intensity of a single voxel unit in the fMRI represent the oxygen supplied to a region of the brain at an instance of time. Therefore, the time series, known as the BOLD signal, can be acquired by taking the mean voxel intensity of the same brain region for various sequentially acquired fMRI volumes (Figure 2.9).



Sources: www.brainvoyager.com

Figure 2.9: Representation of Bold signal.

2.7 Simultaneous EEG-fMRI:

EEG-fMRI data allows a researcher to obtain information on fast voltage changes and hemodynamic activity synchronously [4]. The main advantage of acquiring EEG inside the scanner is that it provides an excellent spatial-temporal resolution [4]. However, the simultaneously recorded EEG signal is subjected to the presence of various artifacts (Figure 2.10 a). Two of the most studied artifacts are the Gradient Artifact (GA) and the Pulsatile Artifact (PA) [2][3].

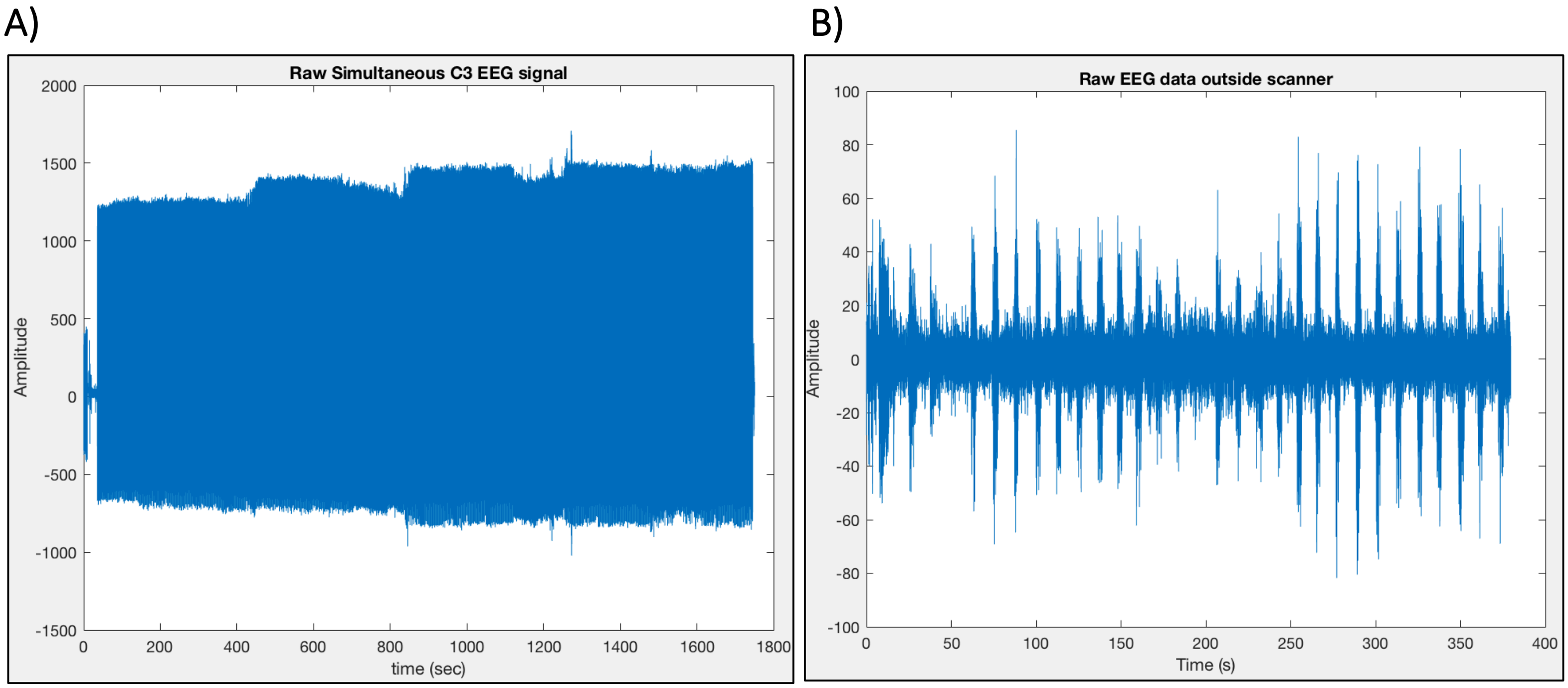


Figure 2.10: (A) Is Raw Simultaneous EEG recording (with GA) as demonstrated by the high amplitudes in the order of 1500 $\mu Volts$ and (B) is the Raw EEG signal recorded outside the scanner with no Gradient Artifact.

2.8 Gradient Artifact (GA):

The Gradient Artifact (or Imaging Artifact) is introduced in the EEG due to the switching of the gradient field in the scanner (Figure 2.11). In echo planar imaging (EPI), the switching of the gradient field occurs every time a new MR slice is acquired [2][12][29]. MR slices tend to be acquired in the range of 50-150 msec, it is during this time that dthe gradient fields and Radio-Frequency (RF) pulses induces a repetitive artifact waveform in the electrical potential of the EEG signal that can be approximated as the differential form of the gradient pulse [2][12]. The magnitude of the gradient artifact can be up to 10^4 μV higher than the neuronal EEG signal [2][12].

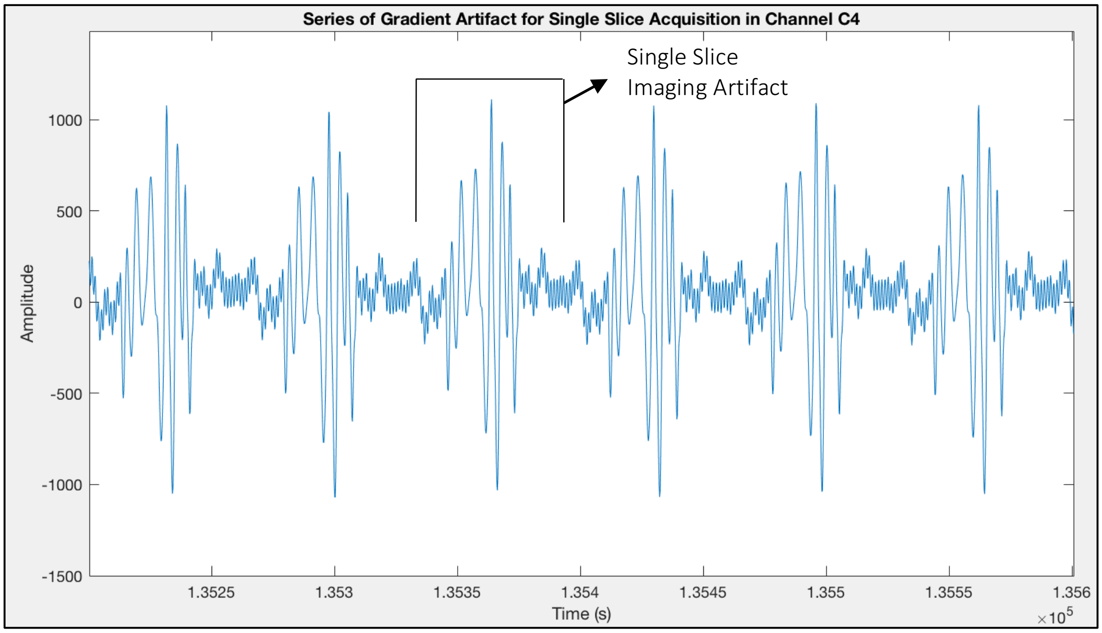
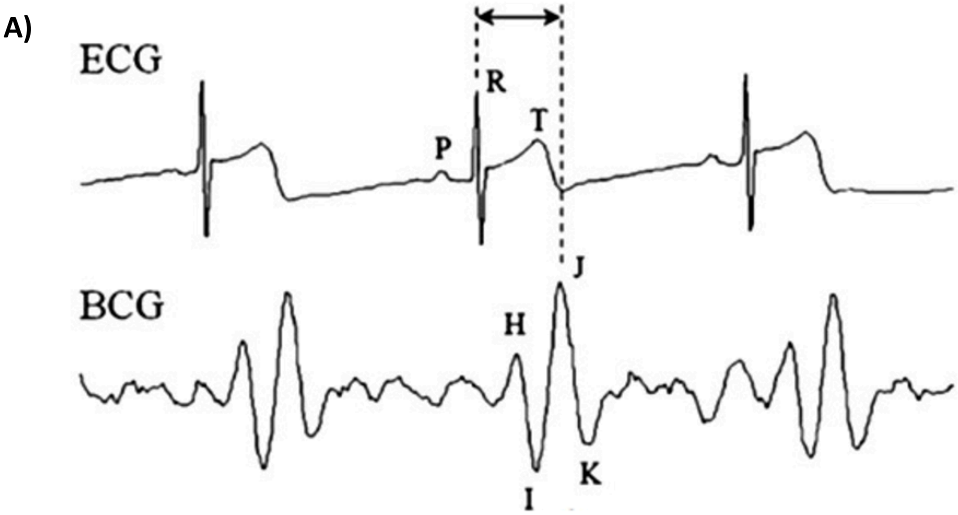


Figure 2.11: Gradient Artifacts present in Simultaneous EEG

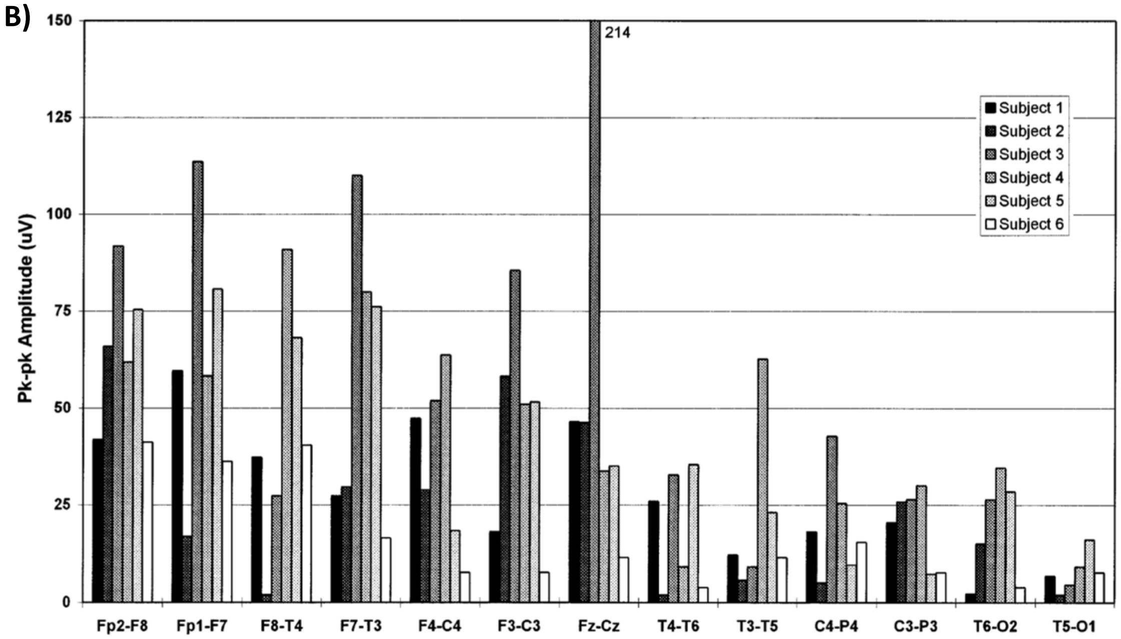
2.9 Pulsatile Artifact (PA):

The presence of the Pulsatile Artifact (PA) is due to the electromotive force induced by head movements that arise from cardiac pulsation while the subject lays inside the MR scanner [3]. The shape of the PA differs from the shape of a typical Ballistocardiogram. Even when the PA is caused by electromotive forces, a typical Ballistocardiogram is determined by heart dynamics, ventricular forces, and directional fluctuations in blood ejection from the heart [3][29] (Figure 2.12). The PA greatly distorts and hides the physiological activity present in the EEG signal. A property of this artifact is that it possesses high power in the frequency range of 4-10 Hz [3][13][21]. The amplitude for the PA artifact varies between channels and subjects, but they can exceed amplitudes of 50 μV (Figure 2.13). Therefore, the presence of this artifact makes it difficult to detect or classify the alpha (α) and mu (μ) brain waves in both time and frequency domains.



Sources: http://abrc.snu.ac.kr/korean/viewforum.php?f=172

Figure 2.12: Comparison of Balliscardiogram in ECG and Pulsatile Artifact in EEG.



Sources: Allen, 1998

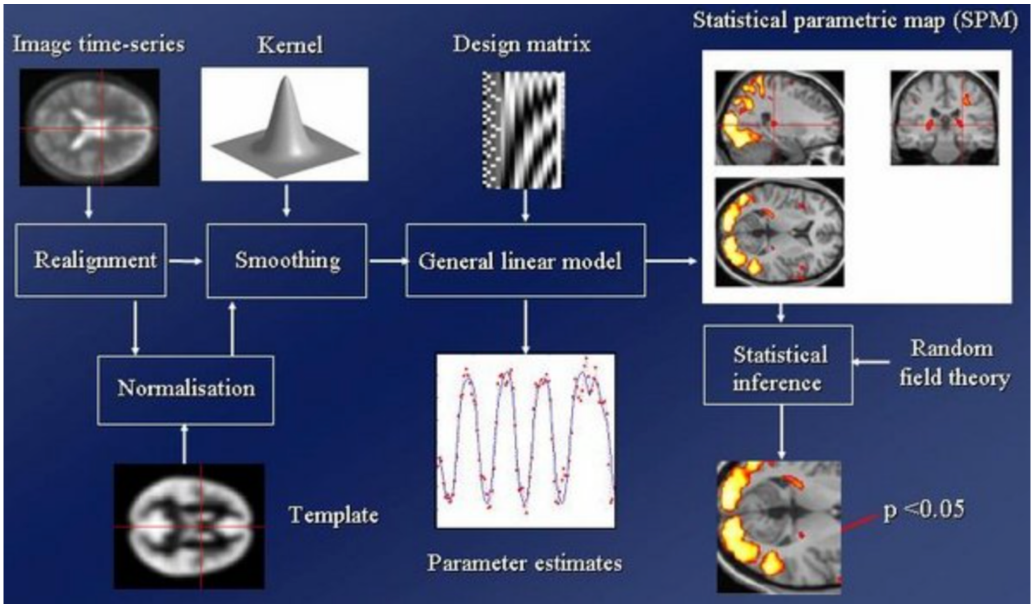
Figure 2.13: Chart of peak-peak amplitude of Pulsatile Artifact.

Theoretical Background:

This section highlights the mathematical background of the techniques that are currently being used in the field and the newly implemented by this study for the analysis of the simultaneous FMRI data, removal of the artifacts from the simultaneous EEG, and classification of brain waves from EEG data. The first subsection introduces the mathematics of different stages of fMRI data analysis. The second subsection deals with different signal processing techniques to remove the two different artifacts present in simultaneous EEG recording. The final subsection aim is to introduce the technique used in this project for both the removal of the PA artifact and the classification of the brain waves of interest.

3.1 fMRI Processing:

fMRI processing is divided in multiple stages. These stages are realignment, normalization, smoothing, parameter estimation, statistical parametric mapping, and statistical inference (Figure 3.1) [1].



Sources: http://slideplayer.com/slide/4918616/

Figure 3.1: fMRI processing

3.2 Affine Transforms:

Affine transformations are the simplest models used in fMRI that uses of linear operations. A property of these transformations is that any set of points that fall in a line will continue to fall on a line after the transformation [9]. This prevents the image to be distorted because is not possible to make radical changes to shape of an object [9]. Affine transforms involve the combination of many linear transforms such as translation along each axis, rotation around each axis, scaling along each axis, and shearing along each axis [9] (Figure 3.2).

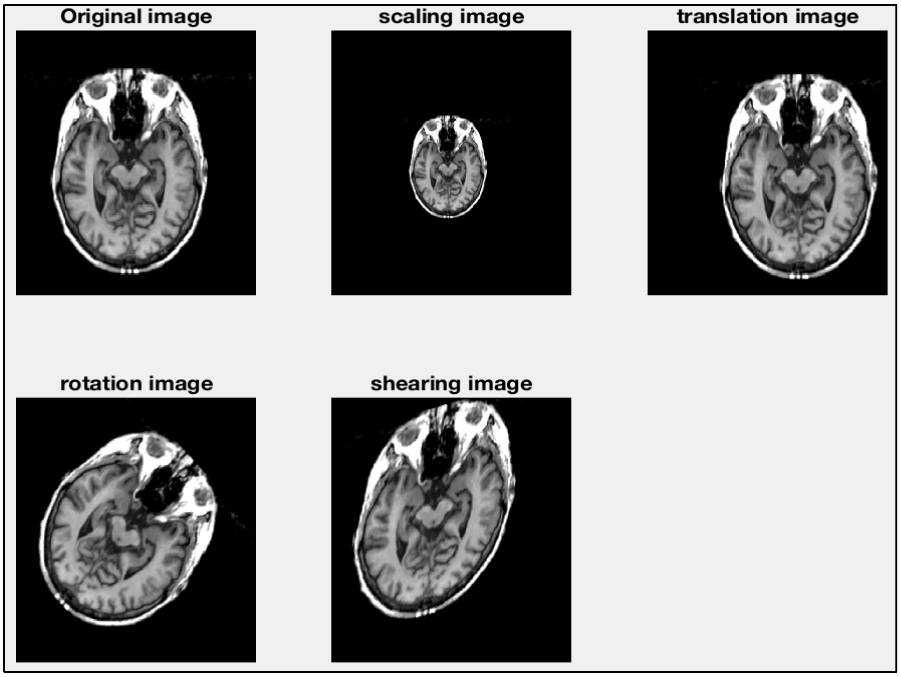


Figure 3.2: Demonstration of affine transformations on a MRI slice

3.3 Mathematics of affine transforms:

Affine transforms involve the applications of a linear operator to the coordinate system of an image (Equation 3.3).

$$C_{transformed} = T*C_{orig}$$

Equation 3.3: Representation of Affine transform

In Equation 3.3, $C_{transformed}$ represents the transformed coordinates, C_{orig} are the original coordinates, and T is the transformation matrix [9]. The transformation matrix can be individual as previously shown in Figure 3.4 or any combination of those linear transforms. For two-dimensional coordinate image the transformation matrixes can be defined as follows (Equations 3.4):

(a)

$$T_{scale} = \begin{bmatrix} Scale_x & 0 & 0 \\ 0 & Scale_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(b)

$$T_{Trans} = \begin{bmatrix} 1 & 0 & Trans_x \\ 0 & 1 & Trans_y \\ 1 & 0 & 1 \end{bmatrix}$$

(c)

$$T_{rotat} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(d)

$$T_{Shear} = \begin{bmatrix} 1 & Shear_x & 0 \\ Shear_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Equations 3.4: Transformation matrix for the (a) Scaling, (b) Translation, (c) Rotation, and (d) Shearing of a 2D image.

3.4 Cost functions:

Motion Correction of fMRI deals with minimizing the differences between the FMRI volumes and reference structural MRI volume. To estimate the parameters for the affine transform model to best align the fMRI images to the reference structural MRI images, the difference between the images need to be determine [9]. The difference between images is referred to as the cost function. A proper cost function should be sufficiently small when two images are well-aligned and larger as they become misaligned. The type of cost function depends on types of images being registered or aligned. If the images are of the same type, like the realignment of fMRI voxels units across different time-points, then the cost function can be determined by minimizing the difference between intensity values [9]. However, when dealing with images of different contrast the optimal alignment will not result in similar intensity values across images [9]. This is known as "between-modality" registration [9]. Therefore, other methods are used that are sensitive to the relative intensities of different sets of voxels [9]. Two of the most commonly used cost functions are the least square and the normalized correction.

The least square cost function measures the average squared difference between voxel intensities in each of the images begin registered (Equation 3.5)

$$\text{C} = \sum_{v=1}^{n}(A_v - B_v)^2$$

Equation 3.5: Least Square Cost Function

A_v and B_v represent the intensities of the vth voxels in the images A and B, respectively [9]. This technique measures similarities of intensity values at each voxel unit. Therefore, it is only appropriate to use this method for within-modality registrations.

The Normalized correction considers the linear relationship between voxel intensities in the two images being aligned [9] (Equation 3.6).

$$\text{C} = \frac{\sum_{v=1}^{n}(A_v B_v)}{\sqrt{\sum_{v=1}^{n} {A_v}^2}\sqrt{\sum_{v=1}^{n} {B_v}^2}}$$

Equation 3.6: Normalized correlation cost function

3.5 Generalized Linear Model:

The generalize linear model is a common technique in FMRI to perform voxel-based correlation analysis. This model relates a single continuous dependent variable to one or more continuous or categorical independent variables [1][7][8](Equation 3.7).

$$y_i = \beta_1\, x_{i1} + \beta_2\, x_{i2} + \beta_3\, x_{i3} + \ldots\, x_{ij}\, \beta_j + \varepsilon_i$$

Equation 3.7: The Generalized Linear Model is a statistical tool that enables one to determine the level of influence predictive variables are having on a observation. The terms y_i, $X_{i,j}$, β_j, and ε_i represent the observation i, value i for predictor variable j (known as a regressor), parameter estimate for predictor variable j, and error for observation i; respectively.

The error term ε, typically is written as $\varepsilon \sim N(0, \sigma^2 I)$, where N is the multivariate normal distribution and I is the identity matrix [8].

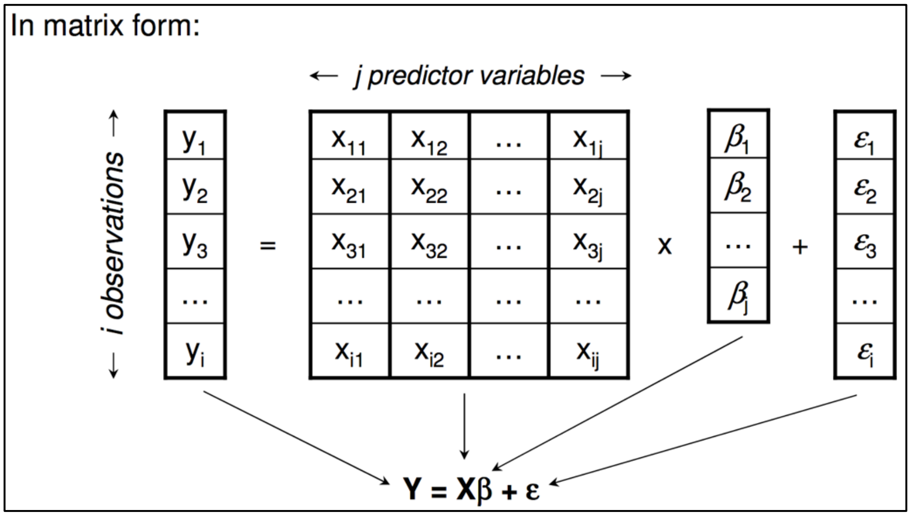


Figure 3.8: Matrix form of GLM. For FMRI data the ith observation would contain the observer bold signals for a given MR voxels unit. The j predictor variables are the modeled hemodynamic functions for a given testing set up. The vector of β represent the level that each of the design HRF influence the observed bold signal. While, the error terms would represent any Gaussian distributed component of the observed bold signal not accounted by the modelled HRF.

In terms of FMRI analysis, the observed data is the BOLD signal of each voxel unit in the FMRI volume [2][8]. Each of the time series of individual voxels are treated as a separate column vector of data [1][8]. The design matrix contains various explanatory components that describes a given observation [1][8]. The β values represent the degree of influence a particular explanatory component has in the description of the observed BOLD signal for a given voxel unit [1][8]. The error is the amount of variability in the observe BOLD signal that cannot be accounted with by the explanatory variables in the design matrix [1][7][8].

In practice, Equation 3.7 cannot be fully solved with $\varepsilon = 0$. The main objective is to estimate parameters β, such that mean square error between the model and the observe data is minimized [1][7][8]. Linear algebra enables such estimations by finding the minimum value to the residual sum of squares (Equation 3.9).

$$\boxed{S = \sum_{j=1}^{N} e_j^2 = \underline{e}^T \underline{e} = \sum_{j=1}^{N} \left(Y_j - x_{j1}\tilde{\beta}_1 - \ldots - x_{jL}\tilde{\beta}_L\right)^2}$$

Equation 3.9: The Residual Sum of Squares represents the sum differences between the best estimated models and given observation squared.

The minimization of the residual sum of square can be derived by taking the derivative and setting it equal to zero (Equation 3.10). The expression in Equation 3.10 can be solved analytically.

$$\boxed{\frac{\partial S}{\partial \tilde{\beta}_l} = 2\sum_{j=1}^{N} (-x_{jl}) \left(Y_j - x_{j1}\tilde{\beta}_1 - \ldots - x_{jL}\tilde{\beta}_L\right) = 0}$$

Equation 3.10: Derivative of Residual Sum of Squares

For the general analytical solution one needs to consider that Equation 3.9 can be restated as Equation 3.11 by using the Equation 3.12 [17].

$$\sum_{j=1}^{N} e_j^2 = e^T e = (Y - X\tilde{\beta})^T (Y - X\tilde{\beta})$$

Equation 3.11: Matrix form of Residual Sum of Squares

$$e = Y - X\tilde{\beta}$$

Equation 3.12: Vector of residuals

$$e^T e = Y^T Y - XY^T\,\tilde{\beta} - X^T\tilde{\beta}^T Y + \tilde{\beta}^T X^T X\tilde{\beta}$$

Since the term $e^T e$ is a scalar, then the term $X^T\tilde{\beta}^T Y$ is also a scaler. Use the fact that the transpose of a scalar is also a scaler [17], one arrives at the following expression (1). By substituting expression (1) into Equation 3.12, one arrives at the expression (2). The Final solution is obtained by taking the derivative of (2), setting it equal to zero (3), and solving for β (4) (5).

$$X\tilde{\beta}Y^T = (X\tilde{\beta}Y^T)^T = X^T Y\tilde{\beta}^T \qquad (1)$$

$$S = e^T e = Y^T Y - 2X^T Y\,\tilde{\beta}^T + X^T X\tilde{\beta}^2 \qquad (2)$$

$$\frac{\partial S}{\partial \tilde{\beta}} = -2X^T Y + 2X^T X\tilde{\beta} = 0 \qquad (3)$$

$$X^T X\tilde{\beta} = X^T Y \qquad (4)$$

$$(X^T X)^{-1}(X^T X)\,\tilde{\beta} = (X^T X)^{-1} X^T Y \qquad (5)$$

The analytical solution is given by Equation 3.13, assuming $X^T X$ is invertible.

$$I\tilde{\beta} = \tilde{\beta} = (X^T X)^{-1} X^T Y$$

Equation 3.13: Least Squares Solution

3.6 Statistical Parametric Maps:

Statistical Parametric Maps (SPMs) is the construction of spatial statistical processes to test hypothesis about specific effects within a region [8][9][10]. SPMs are processes with voxel values that are under the null hypothesis, which distributed with a known probability density function [7][8]. In FMRI analysis the most commonly used SPMs used are T and F maps [1][7][8]. In this study only T maps were generated for the FMRI analysis because the main goal is to examine the net affect that a delay and no delay HRF have on the correlation between voxels.

In this section, the mathematical background of t statistics will be discussed. The T-test can be used to test the significance of a particular contrast of $\beta_{i's}$. For a vector of β_0, β_1, β_2, …., β_n, a contrast vector of length n+1 needs to be specified [1][9]. The contrast vector can be stated to determine if a β_i value is different from 0 or if values β_i and $\beta_{j \cong i}$ are different from each other [1][9]. For each of the respective cases, if i = 1 and j = 2 then the following contrast vectors can be specified c = [0 1 0 … 0] or c = [0 1 -1 0 …. 0]. The null hypothesis for the previous given conditions are H_0: $c\tilde{\beta} = \beta_1 = 0$ and H_0: $c\tilde{\beta} = \beta_1 - \beta_2 = 0$, respectively. It can be demonstrated that the distribution of $c\tilde{\beta}$ is normally distributed with properties of $N(c\beta,\ c(X^TX)^{-1}c^T\sigma^2)$ [1][9]. Therefore, under the null hypothesis, $c\tilde{\beta} \sim N(0,\ c(X^TX)^{-1}c^T\sigma^2)$ [9]. However, a normal distribution cannot be used to conduct the hypothesis test since σ^2 is unknown. Instead, a t statistic is used to perform the hypothesis test (Equation 3.14).

$$t_{values} = \frac{c\tilde{\beta}}{\sqrt{c(X^TX)^{-1}c^T\hat{\sigma}^2}} = \frac{c\tilde{\beta}}{std\ error}$$

Equation 3.14: t statistic equation

In Equation 3.14, the standard error (std error) = mean square error = $\frac{\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{m} X_{i,j}\tilde{\beta}_j)^2}{n}$

3.7 Current techniques for Artifact removal:

The two most studies artifacts present in simultaneous EEG recording are the Gradient Artifact (or Imaging Artifact) and the Pulsatile Artifact (PA). The most widely used technique used for the removal of the GA is Average Artifact Subtraction (AAS) [2].

3.8 Average Artifact Subtraction for GA removal (AAS$_{GA}$):

This technique involves taking the average of a fixed quantity of epochs to calculate an average artifact template [2][12]. The average artifact template represents the gradient artifact waveform, which is subtracted from the original EEG for each of the epochs [2][12]. An epoch can be defined as either one volume scan or as one slice acquisition [2][12]. The most appropriate definition of an epoch will depend on how the FMRI was acquired. If the fMRI sequence was periodic with delays between volume scans, then the epoch is better defined by the volume repetition time (TR). The TR being the time it takes to acquire one fMRI volume scan [2][12]. On the other hand, if the fMRI was acquired continuously without delay between volumes, then an epoch is best characterized by the acquisition time (TA). The TA being the time it takes to acquire a single fMRI slice [2][12]. To optimize the calculation of the average artifact each of the defined epochs are interpolated by a Sinc function [2][12]. In addition, prior to the template formation all the epochs are maximally aligned to each other using of cross-correlation (Equation 3.15). After the formation of the GA template, the template is subtracted from each of the interpolated epochs. Finally, the EEG recording is usually down-sampled at a sampling frequency typically used for EEG (~200 – 250 Hz) [2][12].

$$R_{xy} = \sum_{n=0}^{N-1} x(n)y(n-k)$$

Equation 3.15: Discrete Cross-Correlation

Many other signal processing techniques have been proposed for the removal of the GA. However not only is AAS$_{GA}$ is the simplest techniques to implement, but previous works have shown that it delivers one of the best results for the removal of the GA [2][12]. For these reasons, AAS$_{GA}$ is the only method introduced in this report for the removal of the GA.

The removal of the GA not only uncovers some EEG activity but also uncovers the presence of the PA artifact. Some of the PA artifact removal techniques presented here are AAS_{PA}, Independent Component Analysis (ICA), Optimal Basis Set (OBS) using Principle Component Analysis (PCA), and Adaptive Filtering [29].

3.9 Average Artifact Subtraction for Pulsatile Artifact (PA) removal (AAS_{PA}):

AAS_{PA} simplest and most frequently used technique for the removal of the PA [3][29]. In this method, the cardiac cycles from ECG are used to determine EEG epochs containing PAs. To compute the average PA template in each EEG channel, sections of the EEG signal ∓ ½ the average of R-R interval centered about the R amplitude plus a time delay (~0.21 s) are average [12] [13]. The R-R interval is defined as the distance between an initial maximum peak to the consecutive max peak in ECG data. The R peaks are used because previous works have found that the PA tend to occur after the presence of R peaks in an ECG recoding [13].

This technique has the capability to improve the probability of PA Artifact peak detection [13]. However, the method has major problems in the complete removal of the Artifacts. In previous studies, it has been examined that this technique has only reduced the number of Pulsatile artifact by ~45% [13]. This is mainly due to the underlining assumption made by AAS which is that Pulsatile Artifacts are stationary over a fixed period of cardiac cycles [13]. This assumption is highly non-descriptive of the properties of PA as they tend to vary continuously in form, time, and amplitude [13]. Another major drawback of this methodology, is that the ECG signal used for the PA template formation are also subjected to the presence of PA [13]. Therefore, it is rather difficult to distinguish true R peaks from the peak of a PA to properly capture the presence of the PA in the EEG signal.

3.10 Independent Component Analysis (ICA):

Independent Component Analysis (ICA) is a statistical method use to remove artifacts or noise from a signal interest. It is only able to remove those artifacts that are linearly and independently mixed with the signal of interest [30]. ICA can extract independent sources in a signal if the source signal is considered to be a linear mix of signals [30]. Utilizing this method, the data can model as Equation 3.16.

$$\boxed{x_i = \sum_{j=1}^{p} a_{ij} s_j \qquad \text{or} \qquad \mathbf{x} = \mathbf{A}\mathbf{s}}$$

Sources: Jutten & Herault, 1991; Comon, 1994

Equation 3.16: ICA model tries to return independent sources from a bunch of noise linearly mixed observation.

In Equation 3.16, the terms X, S, and A represent a stack of row vectors representing data from each EEG channels, the stack of row vectors representing each of the independent sources, and the mixing matrix, respectively [30]. Given an input X, the ICA algorithm can deliver estimates for parameters A and S [30]. Then, the estimated parameter S (the independent components) can be used to reject artifacts from the signal of interest [30]. The two major assumptions in the application of ICA to EEG signals are summarized by Maki, 2015. Firstly, the ICA component projections are summed linearly at the scalp electrodes [24]. Secondly, the time series of EEG activity and artifacts are statistically independent [24]. Partially, these assumptions impose a major drawback for the elimination of PA artifact from the EEG signal. This two assumptions imply the PA in each EEG channel are linear independent mixtures form isolated PA sources, which is not the case [24][29][30]. Another major drawback imposed on the extent of PA elimination by ICA is that of under-determined [24]. The concept of under-determined refers to the inability of ICA to separate more than N sources, were N represents the number of electrodes [24]. These prevents the proper separation of independent sources because EEG signals are generated by various synapses, which vastly outnumbers N [24].

3.11 Adaptive Filtering:

There have been multiple approaches to implement Adaptive Filtering for PA Artifact removal. The first method deals with the property of PA artifact to diffuse more spatially than a neuronal signal of interest [29]. In this technique, N EEG electrodes are used to derive a spatial filter by projecting the multiple channel signal into a single dimensional space [29]. The second approach deals with the use of a piezoelectric transducer placed directly in a participant's temporal artery. The motion artifact noise detected by the piezoelectric transducer is used along with a Wiener or Kalman filter for PA artifact removal [29]. The method has proven to be particularly good at alpha detection since it is capable of reducing noise in the frequency band of interest [29]. Previous works have shown that this technique is better than AAS because it does not assume prior knowledge related the artifact waveform [29]. The third technique involves the use of a reference layer adaptive filter. This involves the use of an EEG cap, which contain electrodes positioned directly above the EEG electrodes [22][23]. These extra set of electrodes are the reference electrodes, which are not in contact with the scalp. Therefore, the reference electrodes will detect the presence of the artifact for each of the individual EEG electrode positions. The signal from both the EEG and reference electrodes are processed using both AAS_{GA} and AAS_{PA} [23]. The residual artifacts in the reference electrodes are used with an adaptive filter to remove remaining residual artifacts in the EEG signal [23].

3.12 Machine Learning Algorithm:

Machine Learning (ML) Algorithm can be either Supervised, Unsupervised, Reinforced, Semi-Supervised, and Active learning [20]. The report elaborates on Unsupervised and Supervised Learning.

3.13 Unsupervised Learning:

Unsupervised Learning does not involve the prediction of any outcome (dependent) variable [20]. This type of learning is only dependent on the intrinsic correlation among the input data [20]. These types of ML Algorithms are used to uncover significant patterns or features in the input data without the use of labels [20]. Unsupervised Learning is suitable for biological learning since it does not depend on the utilization of labels, but rather uses primitives like neural competition and cooperation [20]. Unsupervised Gaussian Mixture Model – Expectation Maximization (GMM-EM) Algorithm was employed to learn a prior of the brain wave of interests from the EEG recorded outside the MR scanner. The prior will serve as a basis for Wiener filter to extract the variability in the simultaneous EEG signal, which mostly resembles that of the learned prior.

3.14 Gaussian Mixture Model with Expectation Maximization:

The technique of GMM-EM will be elaborated as in Maki, 2015, but in the time domain. The data from a multi-channel EEG signal relating the kth event of interest from a given set of channels C is represented as $E_k(t) = [E_{k,1}(t), E_{k,2}(t), \ldots, E_{k,C}(t)]^T$ at time t in the time domain. This can also be represented in the following expression (Equation 3.17):

$$E_k(t) = \sum_{l \in A_k} h_l s_l(t)$$

$$h_l = [h_{1,l}, h_{2,l}, \ldots, h_{C,l}]$$

Equation 3.17: Representation of kth event in EEG as the linear mixture of active sources

In Equation 3.17, the terms $h_{i,l}$, s_l(t), and $A_k$represent the transfer function from the lth source to the C channel as long as assumption $0 \leq h_{C,l} \leq 1$ holds for the source signal from various sources and the subset of sources active in the k-*th* event, respectively [24]. The observed multi-channel EEG signal is $X(t) = [x(t)_1, x(t)_2, x(t)_3, \ldots, x(t)_C]$ [24]. The observed signal can be also expressed as the superposition of the various $E_k(t)$ (Equation 3.18), where K is the number of events.

$$\mathrm{X(t)} = \sum_{k=1}^{K} E_k(t)$$

Equation 3.18: Superposition of kth events

It was assumed that the probability density function (PDF) of $s_l(t)$ can be model by a zero-mean Gaussian Distribution (Equation 3.19)

$$P(s_l(t)) = \mathcal{N}_E(s_l(t); 0, v_k(\mathrm{t})) \quad l \in A_k$$

Equation 3.19: Gaussian Distribution

In Equation 3.19, the kth dependent v_k(t) term represents the variance of the complex Gaussian in the time domain. Further, under the assumption that the source signals are non-correlated to each other then the PDF for $E_k(t)$ can be model by a multivariate Gaussian distribution as follows (Equation 3.20) [24].

$$P(E_k(t)) = \mathcal{N}_E(E_k(t); 0, R_{c_k}(\mathrm{t}))$$

$$R_{c_k}(\mathrm{t}) = v_k(\mathrm{t})R_k$$

Equation 3.20: Multivariate Gaussian Distribution

The spatial covariance matrix R_{c_k}(t) is the composed of the product of both the time invariant matrix R_k and the time variant component v_k(t) [24]. Also, under the assumption that a single event signal is active in each time slot as expressed in Equation 3.21, then the PDF of the observed signal can be model by a Gaussian Mixture Model (Equation 3.22).

$$\mathrm{X(t)} = E_{z(t)}(t)$$

Equation 3.21: Active events in each slot

$$p(X|\theta) = \prod_t p(X(t)|\theta)$$

$$= \prod_t \sum_{k=1}^{K} \alpha_k \mathcal{N}_E(X(t); 0, R_{E_k}(t))$$

Equation 3.22: Gaussian Mixture Model

In Equation 3.21, the term $z(t)$ represents the index of the active event signal [24]. While, in Equation 3.22 the terms α_k is the prior probability of a given kth event to be active [24]. The parameter θ = {α_k, ν_k(t), R_{E_k}} of each mixture component [24][27]. Given an observation X, the parameter θ is estimated by maximizing the likelihood function of the GMM (Equation 3.23) [20][24][27].

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(X|\theta)$$

Equation 3.23: Maximization of Likelihood Function

This problem is tackled by using the Expectation-Maximization (EM) Algorithm. The Expectation step computes the posterior probability for every observation X with respect to each of the components in the GMM [20][27]. The Maximization step takes the computed membership posterior probabilities as weights to provide estimates of the means, covariance matrix, and mixing proportion for each component utilizing maximum likelihood [20][27]. The EM algorithm iterates over these two steps until convergence [20][27].

3.15 Implemented Adaptive Filtering:

A Weiner Filter can be represented as follows (Equation 3.24):

$$\hat{E}_k(t) = \hat{R}_{E_k}(t)\hat{R}_X^{-1}X(t)$$

$$\widehat{W} = \hat{R}_{E_k}(t)\hat{R}_X^{-1}$$

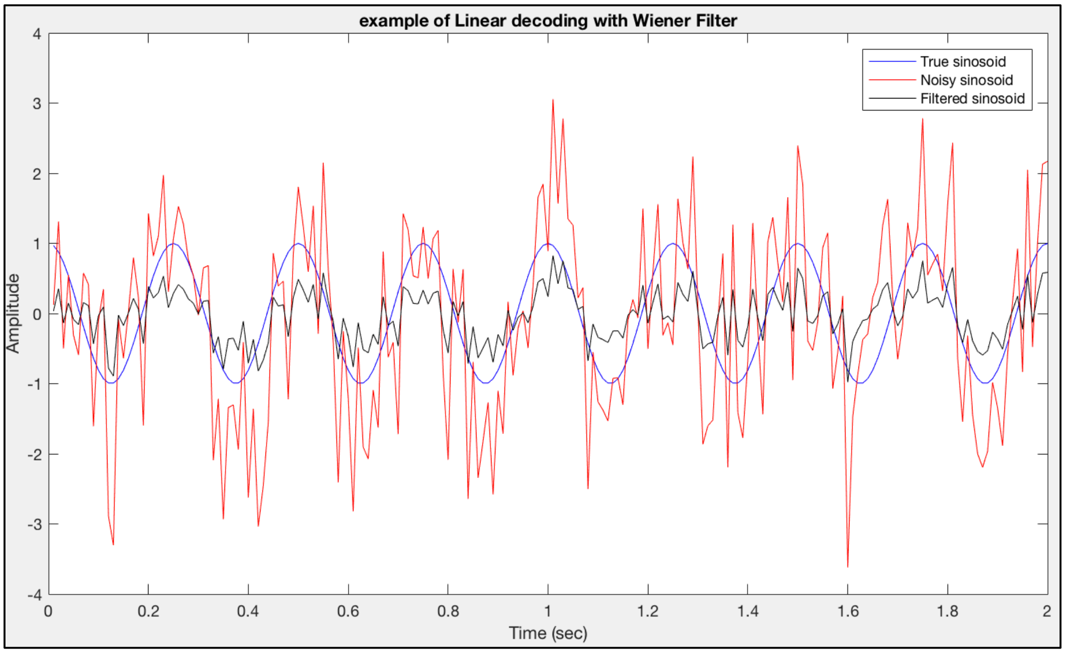
Equation 3.24: Wiener Filter Equation

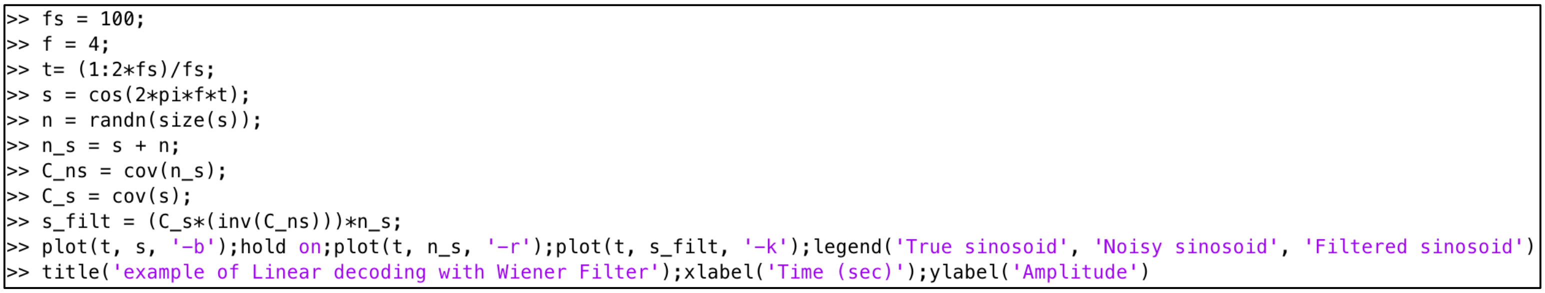
The term $\widehat{W}$, introduced in Equation 3.25, represents the estimated weighting necessary to minimize the square difference between the noisy signal and the desired filtered signal (Equation 3.25).

$$\widehat{W} = \underset{\widehat{W}}{\operatorname{argmin}} \left|\left|\hat{E}_k(t) \; - \; X(t)\right|\right|^2$$

Equation 3.25: Minimization of Mean Squares Error

An example code was developed to illustrate the basic concept behind the use of covariance matrixes in a Wiener filter (Figure 3.26).





```
>> fs = 100;
>> f = 4;
>> t= (1:2*fs)/fs;
>> s = cos(2*pi*f*t);
>> n = randn(size(s));
>> n_s = s + n;
>> C_ns = cov(n_s);
>> C_s = cov(s);
>> s_filt = (C_s*(inv(C_ns)))*n_s;
>> plot(t, s, '-b');hold on;plot(t, n_s, '-r');plot(t, s_filt, '-k');legend('True sinosoid', 'Noisy sinosoid', 'Filtered sinosoid')
>> title('example of Linear decoding with Wiener Filter');xlabel('Time (sec)');ylabel('Amplitude')
```

Figure 3.26: Illustration for Wiener Filter with Covariance Matrix. The red, blue, and black sinusoid indicate the noise, true, and filtered sinusoid, respectively. The goal of the Wiener filter

is to reduce the power of the noise, such as the mean square error between the noise signal and the signal of interest is reduced.

The assumptions for a Wiener Filter:

1. The noise being suppressed by the filter has to be additive (Equation 3.27) [25].

$$x(t) = s(t) + \eta(t)$$

Equation 3.27: Representation of Observed Noise

2. Both $s(t)$ and additive $\eta(t)$ have well-defined power spectra over some period of interest [25][26].
3. $s(t)$ and additive $\eta(t)$ are stationary linear stochastic processes [25][26].

The primary challenge of Wiener filtering is the estimation of two district power spectrum from a single signal $x(t)$. The reason being, as opposed to the example in Figure 3.26, the power spectrum of the actual signal of interest is often unknown. The Wiener filter can be either applied in the time and/or frequency domain [25][26]. Currently, two mathematical tools for relating second-order statistical properties of mixtures with deterministic and stochastic finite-power signals are autoconvolution and panorama [25]. The learned covariance prior from the mixture of features represents the variability of the EEG signal that accounts for the power that a brain wave has in the overall power of an EEG signal. However, GMM-EM gives a covariance matrix from a higher dimensional feature space rather than from a single channel signal.

For $\forall\, f \in V$, where V is a vector of features, GMM-EM can provide a covariance matrix with dimensions equal to V for each independent kth component. The general covariance matrix, computed from the class data covariance of the EEG, can be used for a Multichannel Wiener filter [19]. The space covariance can be computed by averaging all same class data covariance of the multichannel EEG [19]. Therefore, the covariance matrix of all set of features, used to distinguish each of the independent components, can be averaged to calculate the space covariance of each component (Equation 3.28). Then, the general covariance of the two independent components (or the two class) can be expressed as a sum between the space covariance of each component [19] (Equation 3.29).

$$\bar{R}_1 = \frac{R_{1,f_1} + R_{1,f_2} + R_{1,f_3} + \cdots + R_{1,f_n}}{n} \;;\; \bar{R}_2 = \frac{R_{2,f_1} + R_{2,f_2} + R_{2,f_3} + \cdots + R_{2,f_n}}{n}$$

Equation 3.28: Space Covariance for a two-class system

$$R_c = \bar{R}_1 + \bar{R}_2 = \sum_{k=1}^{K=2} \bar{R}_k$$

Equation 3.29: General Covariance for a two-class system

Knowing the general covariance for a two-class (component) Gaussian Mixture Model, the following Multichannel Wiener filter can be implemented (Equation 3.30):

$$\hat{X}_c(t) = \hat{R}_c(t)\hat{R}_X{}^{-1}X(t)$$

Equation 3.30: Multi-Channel Wiener Filter

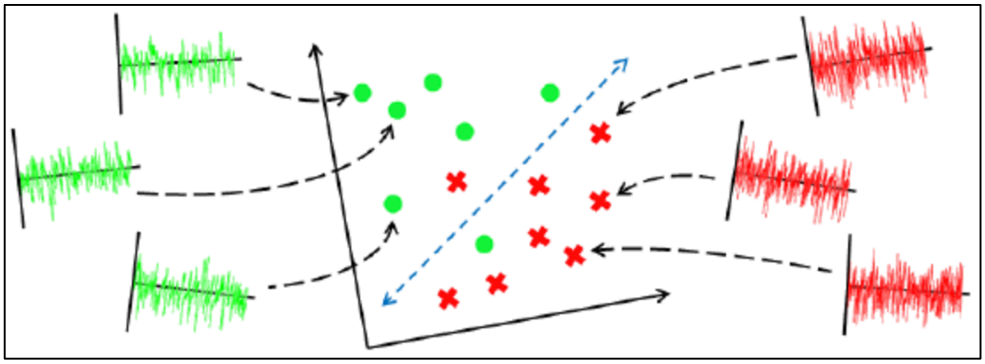
In Equation 3.3.13, the term $\hat{X}_c(t)$ is the filtered Multichannel EEG recorded inside the scanner by using the learned $\hat{R}_c(t)$ from a Multichannel EEG recorded outside the scanner. Equation 3.3.12 and Equation 3.3.13, can be generalize for a multi-class GMM but for the purposes of this project only a two-class GMM was used.

3.16 Supervised Learning:

Supervised Learning involves target (or dependent) variables which is to be predicted from a given set of predictors (independent variables) [20]. Using the set of variables, the algorithm can learn a mapping from inputs to a desired output. The training process continues iteratively until the model reaches a desired accuracy on the training data [20]. Two common supervised classification algorithms commonly use in Brain Computer Interfaces (BCI) are Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM) [16]. In this project, a Supervised Support Vector Machine Algorithm was utilized for the classification of different brain waves of interest.

3.17 Support Vector Machine (SVM):

Support Vector Machines uses a discriminate hyperplane to identify different classes [11][14][20]. The selection of the hyperplane is the one that maximizes the margins (Figure 3.31). In SVM, the margins are defined by the distance from the nearest training point [11][14][20]. Something that is highly crucial is the design and selection of descriptive features of the particular brain waves of interest [11][14][20]. The extracted features used for the classification in this report will be emphasize in the following chapter.



Sources: http://shm.ucsd.edu/Site/research/shmalgorithms.html

Figure 3.31: Illustration of SVM classification in Higher Dimensional Feature Space

3.18 Cross-Validation:

Training a classifier requires the assessment of its performance. The accuracy of the classifier is quantified by comparing the number of correctly label data to its original assigned label. Cross-Validation is a standard model-selection method in statistics [20]. In this method, the total data set is randomly partitioned into a training and testing set [20]. The major part of the partitioning is incorporated in the training set, which is utilized to train the ML algorithm [20]. Typically, 10-20% of the whole data set is included in the testing set and is used to examine the overall accuracy of the classifier [20]. The testing set is different from the training set and so the testing set is completely independent from the set used to estimate the model [20].

A popular type of cross-validationthat is often used to diagnose the efficiency of the ML Algorithm is K-folds [20]. In K-folds cross-validation, the whole dataset is split into K non-overlapping data subsets with the same dimensions [20]. Each data subset can be partition into testing and training sets. The K-fold cross-validation estimation error is the average value of the errors for each of the K fold subsets. This error estimation depends on both the training set and the K-fold portioning [20].

Materials and Methods:

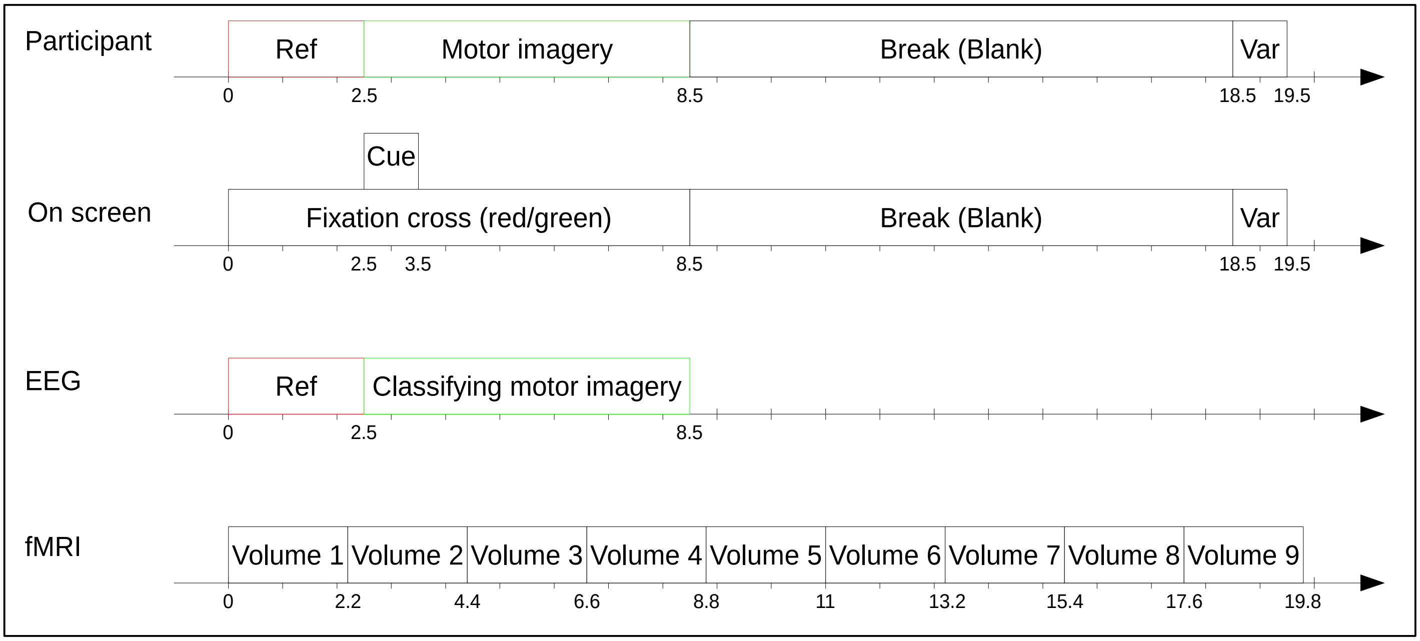
The aim of the study is to examine the performance of statistical/machine learning method in the analysis of simultaneous EEG-FMRI data. These techniques were implemented for both the analysis of FMRI images and the effective removal of artifacts in the simultaneous EEG recoding.

4.1 EEG-FMRI Data:

The two datasets acquired were Simultaneous EEG-fMRI data and the EEG data recorded outside the scanner. Both of the datasets were obtained from the same participant. The data used in this study was provided by the Institute of Neural Engineering at TU Graz. The data provided was collected from a single 23-year-old male participant [21]. The participant resided inside an fMRI scanner, while receiving instructions for performing a Motor Imagery (MI) task [21].

4.2 Motor Imagery (MI) Paradigm Description:

This paradigm applies for both the simultaneously acquired EEG recording and the EEG recorded outside the FMRI scanner. At the start of each trial, a fixed cross appeared on the screen. The fixed cross remained on the screen for ~ 8.5 seconds. The first 2.5 seconds were used as a reference period, while the remaining 4 seconds were used to label either right hand or right feet MI [21]. The indication of performing either right hand or right feet MI was randomized [21]. After indication for MI, the participant was asked to maintain the MI until the screen appeared blank [21]. The break duration for each trial lasted from a range of 8 – 10 seconds. The total number of runs performed during the simultaneous recording and non-simultaneous recording were 4 and 7, respectively. Every run in the simultaneous EEG and non-simultaneous EEG recording contained 20 and 30 trials, respectively. During this trials, the participant performed either right hand or right feet MI. The following paradigm is illustrated by Figure 4.1.

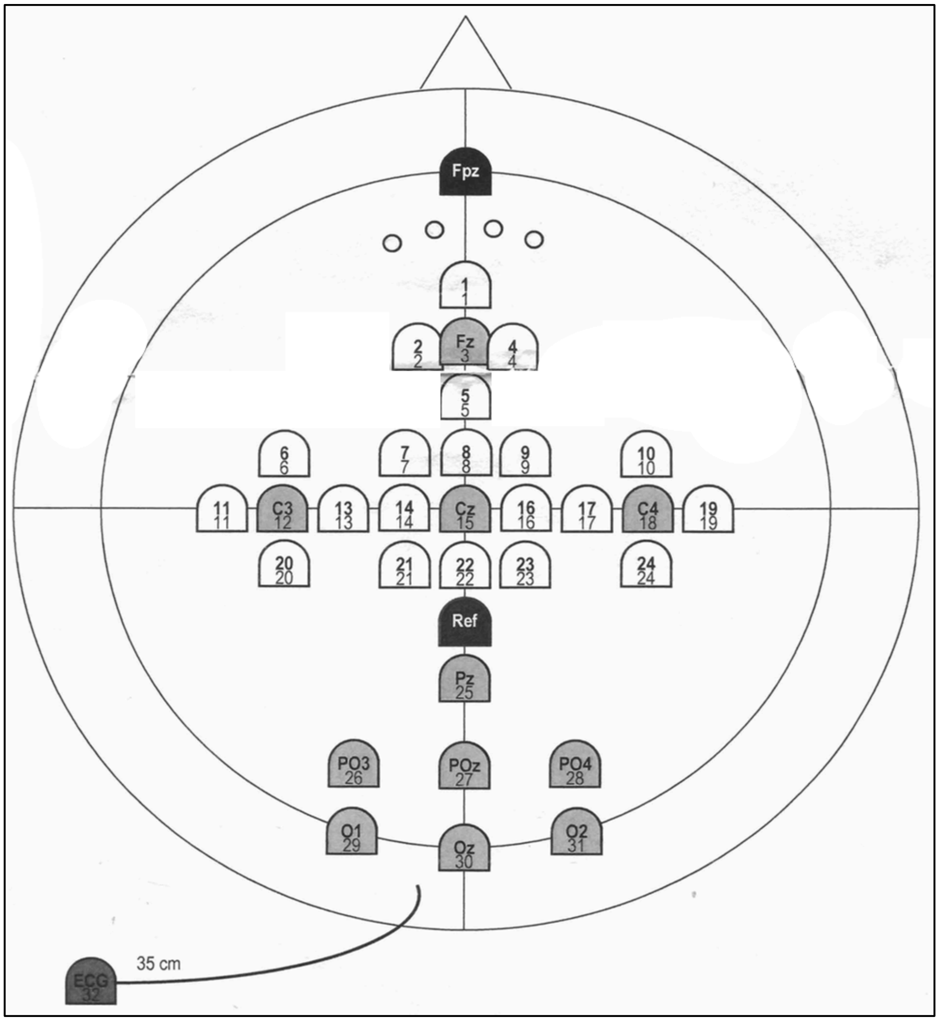


Sources: Institute of Neural Engineering of TU Graz (by David Steyrl)

Figure 4.1: Illustration of Motor Imagery (MI) paradigm. Figure shows how the time, in which the fMRI volumes were acquired, was broken down. The color of the fixation cross represented whether the participant needed to perform either right hand or right feet motor imagery. The cue on the screen markes the time were the EEG signal need to consider as Motor Imagery. The time prior to the cue marker, highlighted in red (ref), indicates the time were the EEG signal does not represent any MI activity. Once the screen goes blnk, the participant could stop performing MI.

4.3 Simultaneous EEG-FMRI Acquisition:

A 32 channels MRI compatible EEG cap (BrainCap-MR with Multitrodes, EASYCAP GmbH, Herrsching, Germany) (Figure 4.2) was used for recording. The EEG cap was synchronized with the gradient field clock of the MR scanner [22]. All 32 EEG electrodes were made of Ag/AgCl with safety resistors [22]. To fully capture the waveform of the GA the EEG activity was acquired at the sampling rate of 5000 Hz [22]. The cut off frequency of analogue high pass and low pass filters were set to 0.016 Hz and 250 Hz, respectively. During recording, the voltage range was set to ∓ 16.384 mV [22]. The EEG recording was conducted within a 3T Skyra (Siemens, Erlangen, Germany) MRI scanner [22]. The MR scanner was equipped with standard 24 channel head coils [22]. During the activation time of the scanner an echo planar imaging (EPI) sequence was running with repetition time (TR) of 2200 *m*s. To reduce the motion of the EEG electrode cables sand bags were used to hold them in place [22].



Source: EASYCAP GmbH, Herrsching, German

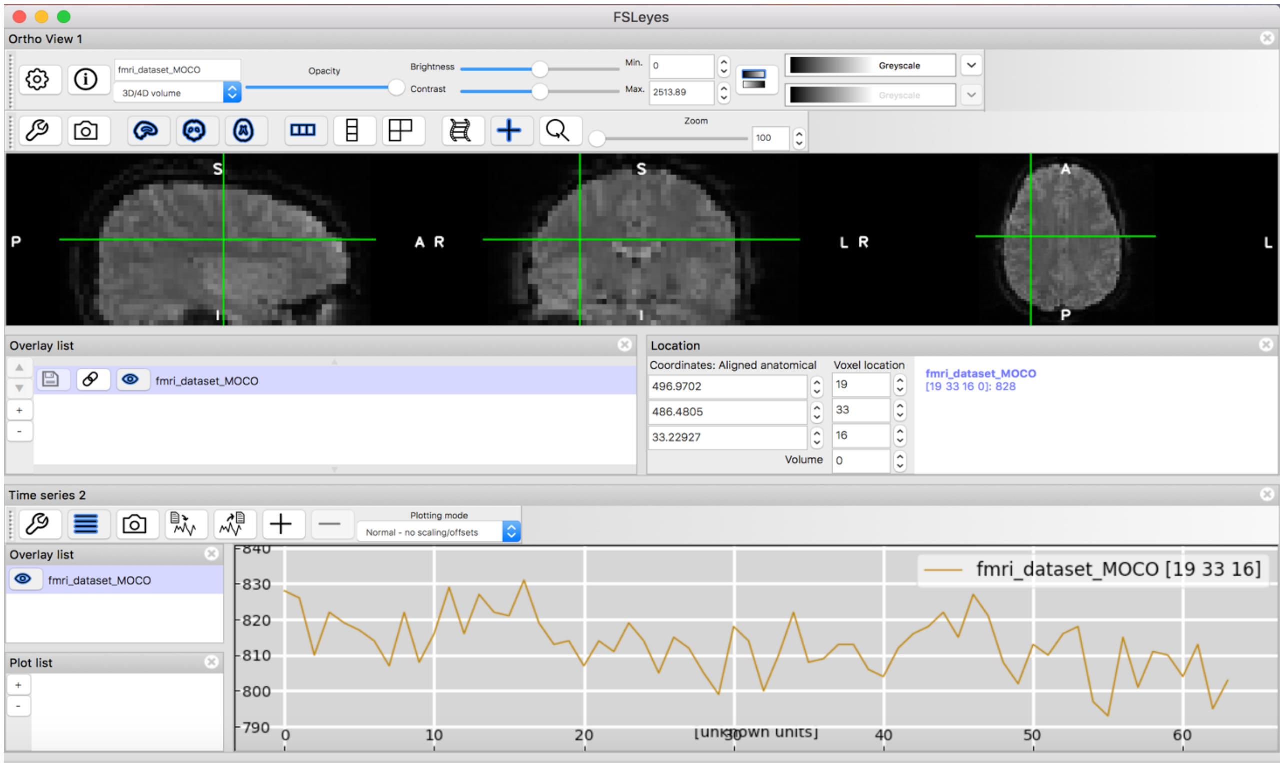
Figure 4.2: 32 channels MRI compatible EEG cap

4.4 Non-Simultaneous EEG Acquisition:

A “BrainCap-MR” electrode system by EasyCap (EASYCAP GmbH, Herrsching, German) was used to record EEG activity outside the MRI scanner [21]. The cap consists of one ground, one reference, one electrocardiogram (ECG) and 63 EEG electrodes [21]. The EEG electrodes were regularly place on the scalp with a distance of 2.5 cm from each other [21]. The rectangular grid was centered on a 10-20 system positions C3, Cz, and C4 [21]. To simulate the same conditions as the simultaneous EEG, the participates rested in a face up position while the non-simultaneous EEG activity was recorded. The sampling frequency used for recording was of 200 Hz.

4.5 Data processing:

The raw FMRI data provided were formatted in .IMA format. The FMRI dataset contained 64 consecutive acquired volumes form one participant. Each of the 64 volumes were imported into the free FSL software (Analysis Group, FMRIB, Oxford, UK) version 5.0 and were motion corrected with the default Normalized correlation cost function. Six degrees of freedom (DOF) were used for FMRI registration, with a structural MR reference volume, rather than 12 DOF. Six DOF (3 translations and 3 rotations) is reasonable because during image realignment the brain does not vary in size or shape. The voxel time series (or Bold signal) were obtained by taking the mean intensity for same voxel unit in the different volumes and plotting them over time (Figure 4.3).


FSLeyes

Ortho View 1

fmri_dataset_MOCO
3D/4D volume
Min. 0
Max. 2513.89
Greyscale
Zoom 100

Overlay list
fmri_dataset_MOCO

Location
Coordinates: Aligned anatomical: 496.9702, 486.4805, 33.22927
Voxel location: 19, 33, 16
Volume 0
fmri_dataset_MOCO
[19 33 16 0]: 828

Time series 2
Plotting mode: Normal - no scaling/offsets

Overlay list
fmri_dataset_MOCO

Plot list

Figure 4.3: The co-registered fMRI images from left to right, display the sagittal, coronal, and traverse views of the brain. The position of green cursor indicated coordinates of a voxel unit of interests and the plot displayed directly beneath the images is the observed bold signal for voxel units indicated by the cursor.

4.6 FMRI Voxel-Based Correlation Procedure:

1. After motion correction, the skull was stripped off the brain images by applying a 0.6 fractional intensity threshold.
2. After skull stripping, a Full Wight Half Maximum Gaussian kernel of 4mm was applied to the MOCO fMRI. This spatially smooths out the intensity to increase the signal to noise ratio (SNR).
3. A mask was created to remove the surrounding voxels activity which lie outside brain regions.
4. The design matrix for the GLM model uses Mark Cohen's Gamma-variate function along with its first and second temporal derivatives (Equation 4.4) convoluted with a non-shifted and shifted (6-seconds) box-car function. The box-car function was assigned a value of 1 during the time segments were either right hand or right feet MI was performed and zeros during the other time periods. The purpose of including both first and second temporal derivatives of the HRF is to capture peaks that occur earlier or later in the bold signal and to capture narrower or wider responses in the signal, respectively.

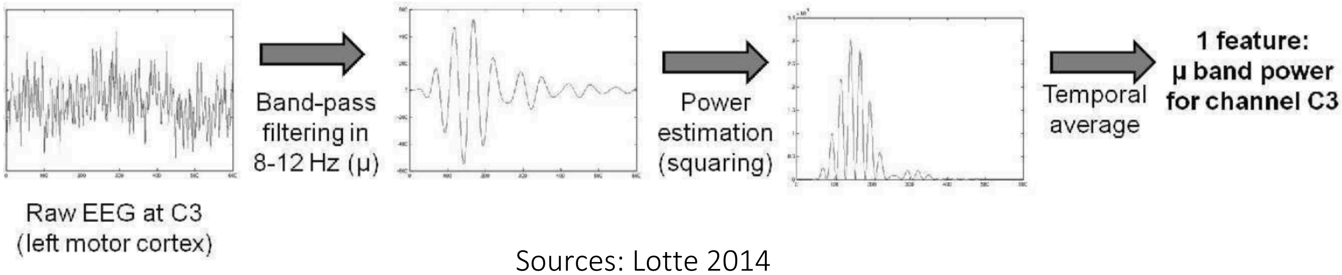
   $$\mathrm{r(t)} = t^{b}e^{-\frac{t}{c}}$$

   Equation 4.4: Mark Cohen's Gamma-variate function used to represent Hemodynamic Response Function (HRF). The terms b and c are constant with values of 8.6 and 0.547, respectively.

5. The Ordinary Least Square was used to solve for the best β estimates for the GLM. T maps were calculated for each of the FMRI slices. The T Maps are then overlay to its appropriate fMRI slice.

4.7 Procedure for MI classification from non-Simultaneous EEG:

1. The EEG was bandpass filtered, using a Finite Impulse Response (FIR) filter of order 1000, from 1Hz to 80 Hz. The high pass at 1 Hz removes low-frequency drifts, while the low pass at 80Hz removes any frequency that is not part of the EEG signal.
2. A Butterworth Notch filter of 8th order and zero phase was used to remove 50 Hz powerline noise in the EEG signal.
3. A second FIR bandpass filter of order 1000 was applied from 8 Hz to 30 Hz. This was done to enhance MI features which have been found to predominate in the *mu* and beta ranges.
4. One second long epochs were extracted from channels C3, Cz, Pz, and CP1. Every extracted epoch began from the time when the cue to perform either right hand or right feet MI appeared on the screen.
5. The SVM classifier was train using the time domain feature vector from the extracted epochs of channels C3 and Cz. After training, the performance was quantified by calculating the proportionality of the correctly labeled MI to the total number of epochs extracted from both Pz and CP1 channels.
6. Repeat steps 1-4, then a power estimation was obtained by squaring each of the 1 second long epochs extracted from channels C3, Cz, Pz, and CP1 (Figure 4.5). Step 5 was repeated but using the new power band feature vector.



Sources: Lotte 2014

Figure 4.5: Illustration of band power feature extraction from channel C3. The left most plot indicated a one second long epoch containing MI activity. The center plot is the bandpass filtered (8-12 Hz) raw EEG signal containing the MI activity. The right plot is the power estimation obtained by squaring the bandpass filtered signal.

4.8 Procedure for MI classification from Simultaneous EEG:

1. Implement an AAS$_{GA}$ algorithm for the removal of the GA from the Multichannel EEG signal. The algorithm extracts EEG epochs defined by the time intervals when the fMRI volumes were acquired. For each individual channel, the epochs were interpolated to 5 times the original sampling frequency of the EEG.
2. All the epochs belonging to a given channel were cross-correlated with one another. Then, they were shifted in time until the epochs were maximally cross-correlated with each other.
3. For all the cross-correlated epochs, the average artifact template was obtained by averaging 50 epochs before and 50 epochs after a given epoch. This process helps to accounts for extract variability in the GA waveform due to the participant's motion.
4. After obtaining the average artifact template for a given epoch, the least square error between the epoch and the template was minimized using ordinary least squares. The template was then shifted in time until it was maximally cross-correlated to the given epoch and subtracted. Steps 1 – 4 will repeat for all the EEG channels.
5. The AAS$_{GA}$ EEG data is low pass at 70 Hz and subsequently down-sampled at a new sampling frequency of 250 Hz.

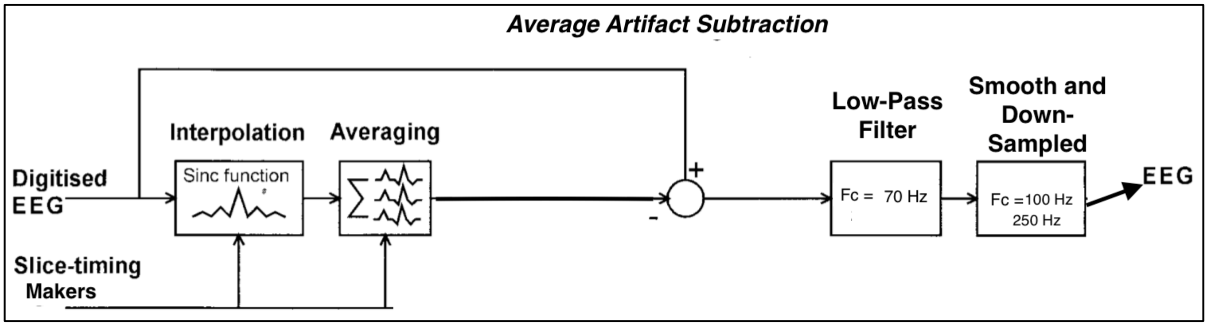


Figure 4.6: Illustration showing the general process of the implemented AAS$_{GA}$

6. To remove the PA artifact, the algorithm described in section 3.14 and 3.15 was employed. The first 100 ms of each of the C3 and Cz extracted epochs were used to learn the spatial covariance prior.

7. After the removal of the PA artifact, a Butterworth Notch filter of 8th order and zero phase was used to remove 50 Hz powerline noise in the EEG signal.
8. Steps 3 – 6 are repeated for the non-simultaneous multichannel EEG data with channels C3, Cz, C21, and C22.

Results:

5.1 Regressors used for GLM:

A total of 6 regressors were used in the GLM model, excluding the baseline regressor. The first 3 regressors were the HRF and its first and second temporal derivates without delay with respect to the time of MI activity (Figure 5.1). The second set of 3 regressors used were the same HRF and its temporal derivatives with an induced 6 second delay with respect to the period of MI activity (Figure 5.2).

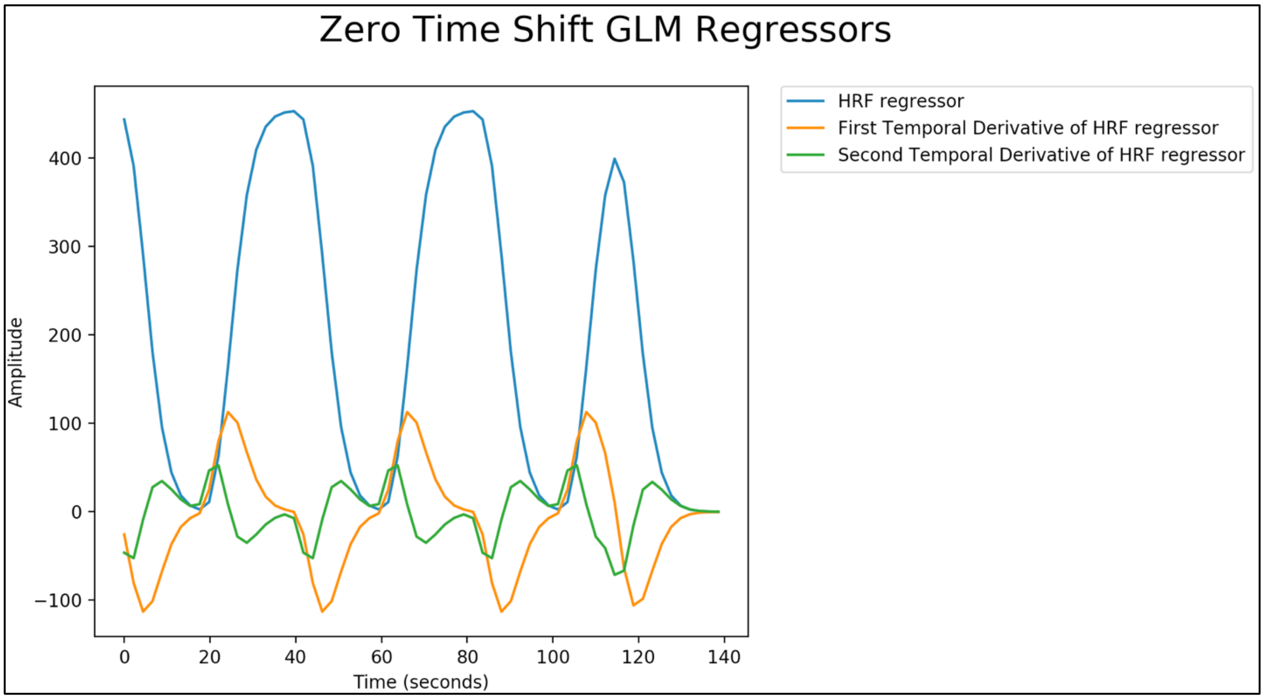


Figure 5.1: GLM regressors with no induced time delay with respect to MI activity

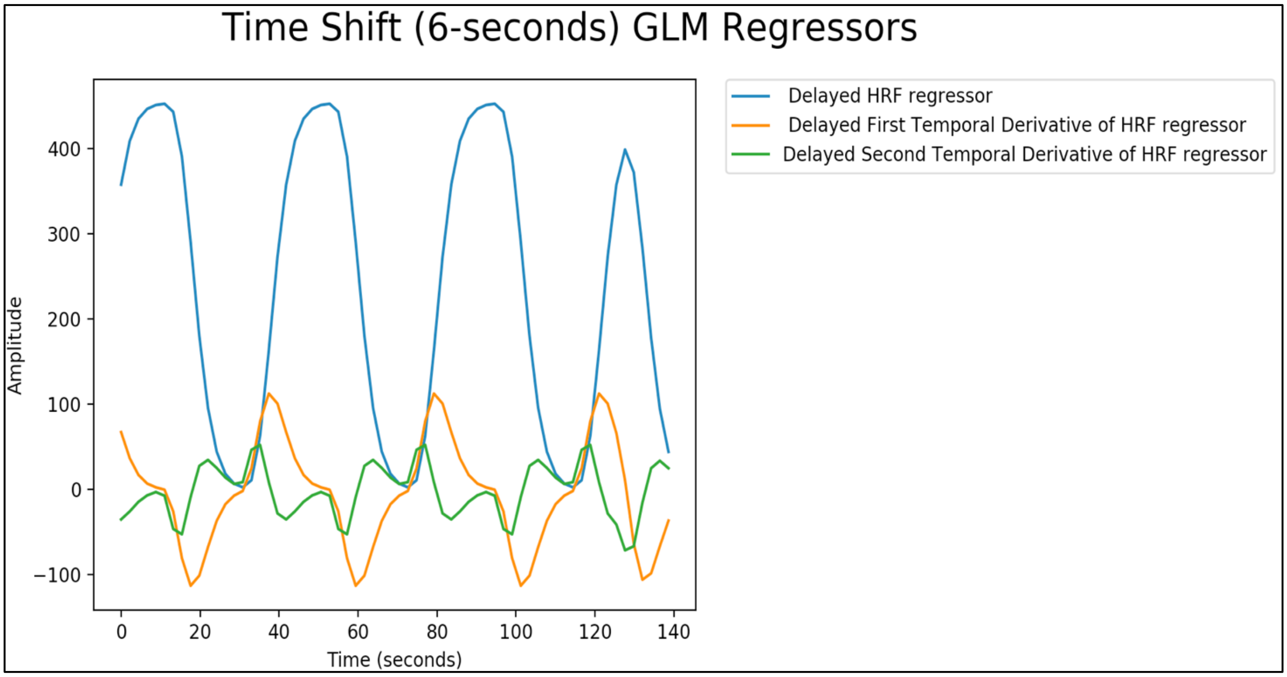


Figure 5.2: GLM regressors with 6 seconds induced time delay with respect to MI activity.

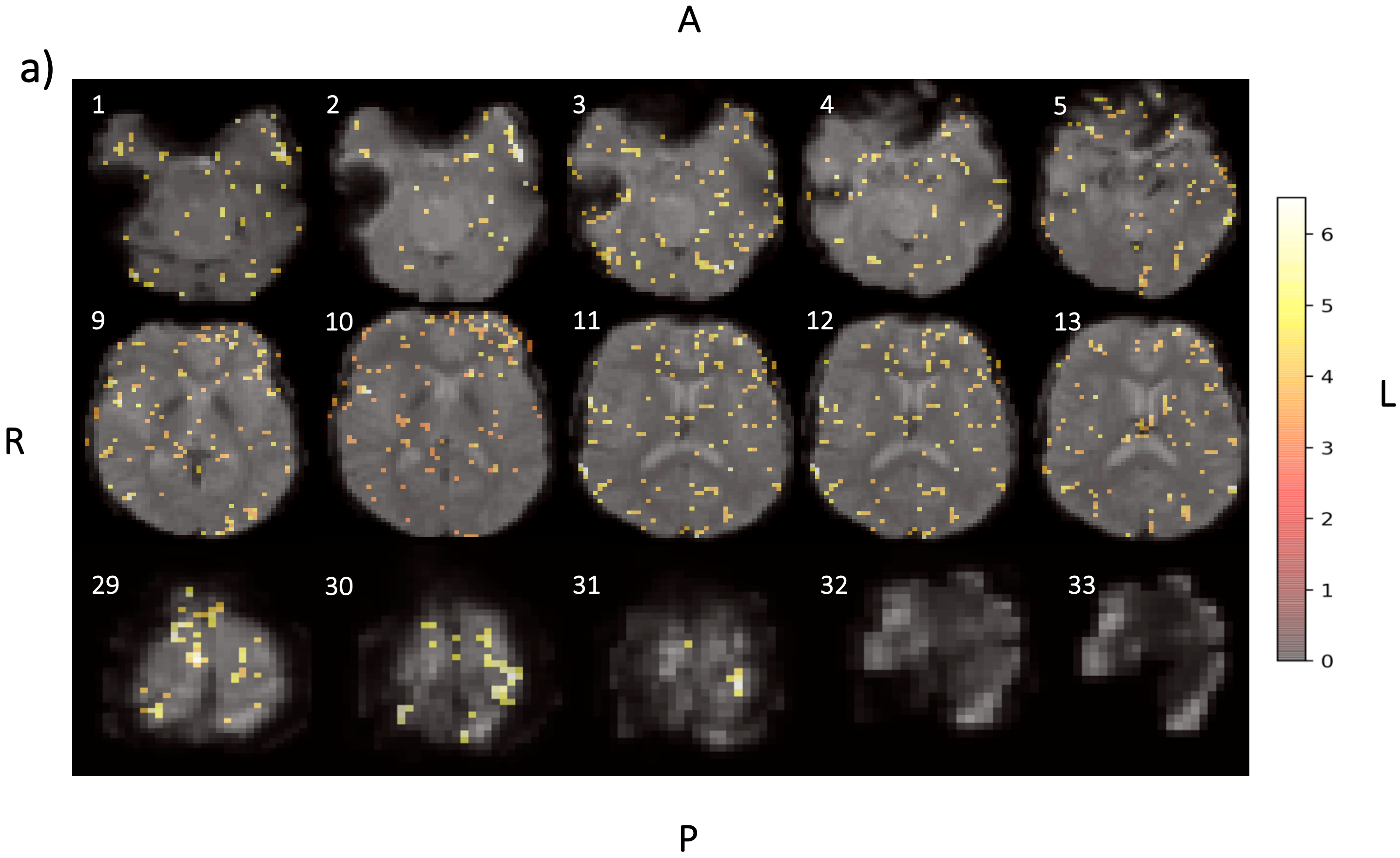
5.2 T-Maps from GLM:

The t-value maps were calculated for each individual FMRI voxel. To evaluate the level of correlation each voxel had with a given contrast vector (or null hypothesis) the T maps were threshold at a value of 1.5 times the interquartile range (IQR) plus the 50th percentile. The null hypothesis in question were:

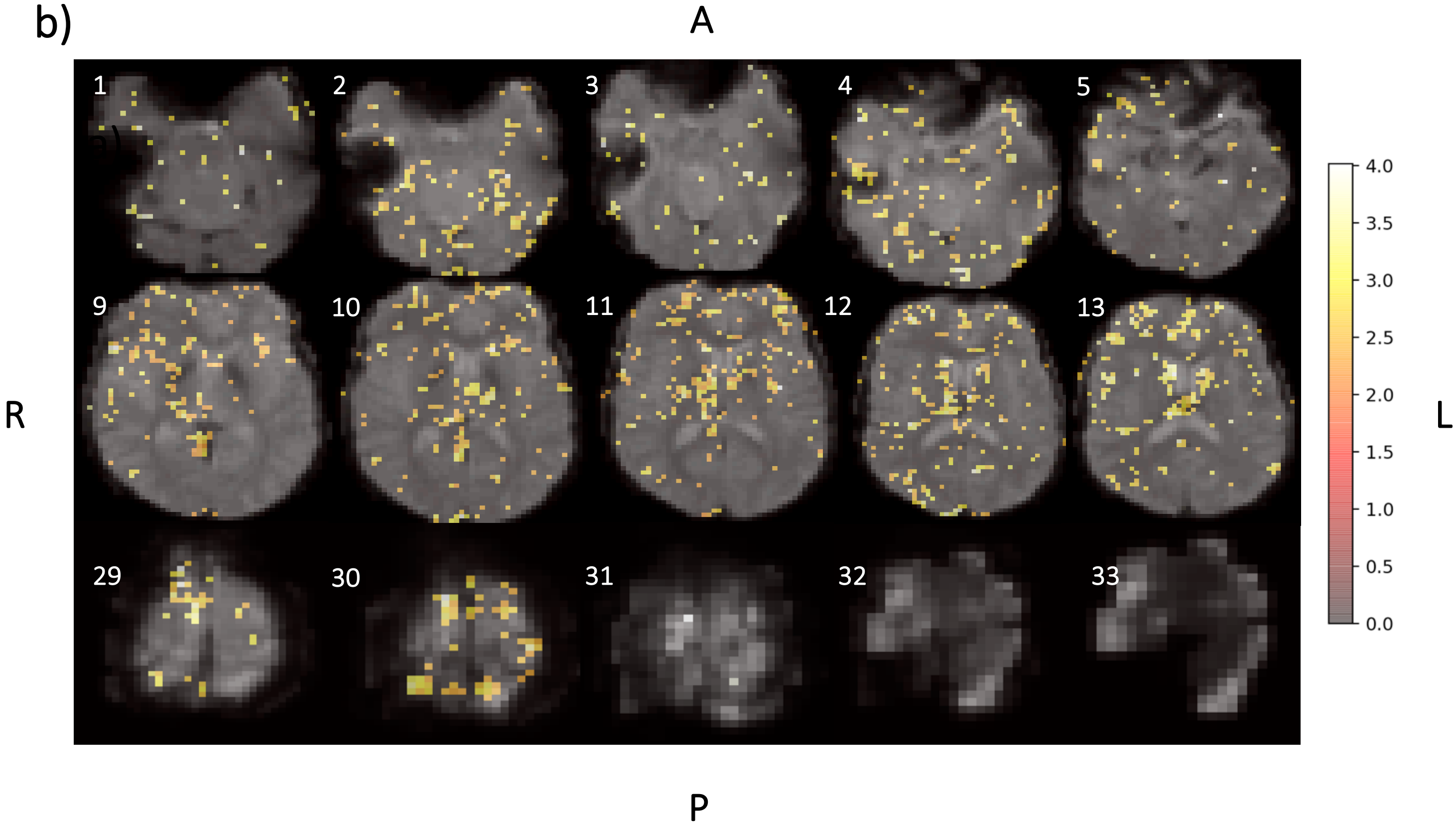
1. If the combination effects of the zero time-delayed regressors are equal to zero.
2. If the combination effects of the 6 second time-delayed regressors are equal to zero.
3. If the difference between the effects of the 6 second time-delayed regressors and the effects of the zero time-delayed regressors are equal to zero.

The resulted threshold T-maps were overlay on to appropriate 2D FMRI slices. A few of the transverse slices are shown in Figure 5.3, showing motor related brain regions, for the three scenarios. The colored heat map shows the voxels with t-values greater than the 50th percentile plus 1.5 times the IQR. The colormap show color representation given to each t-value.

a)


A
1
2
3
4
5
9
10
11
12
13
29
30
31
32
33
R
L
P
6
5
4
3
2
1
0

b)


A
1
2
3
4
5
9
10
11
12
13
29
30
31
32
33
R
L
P
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0

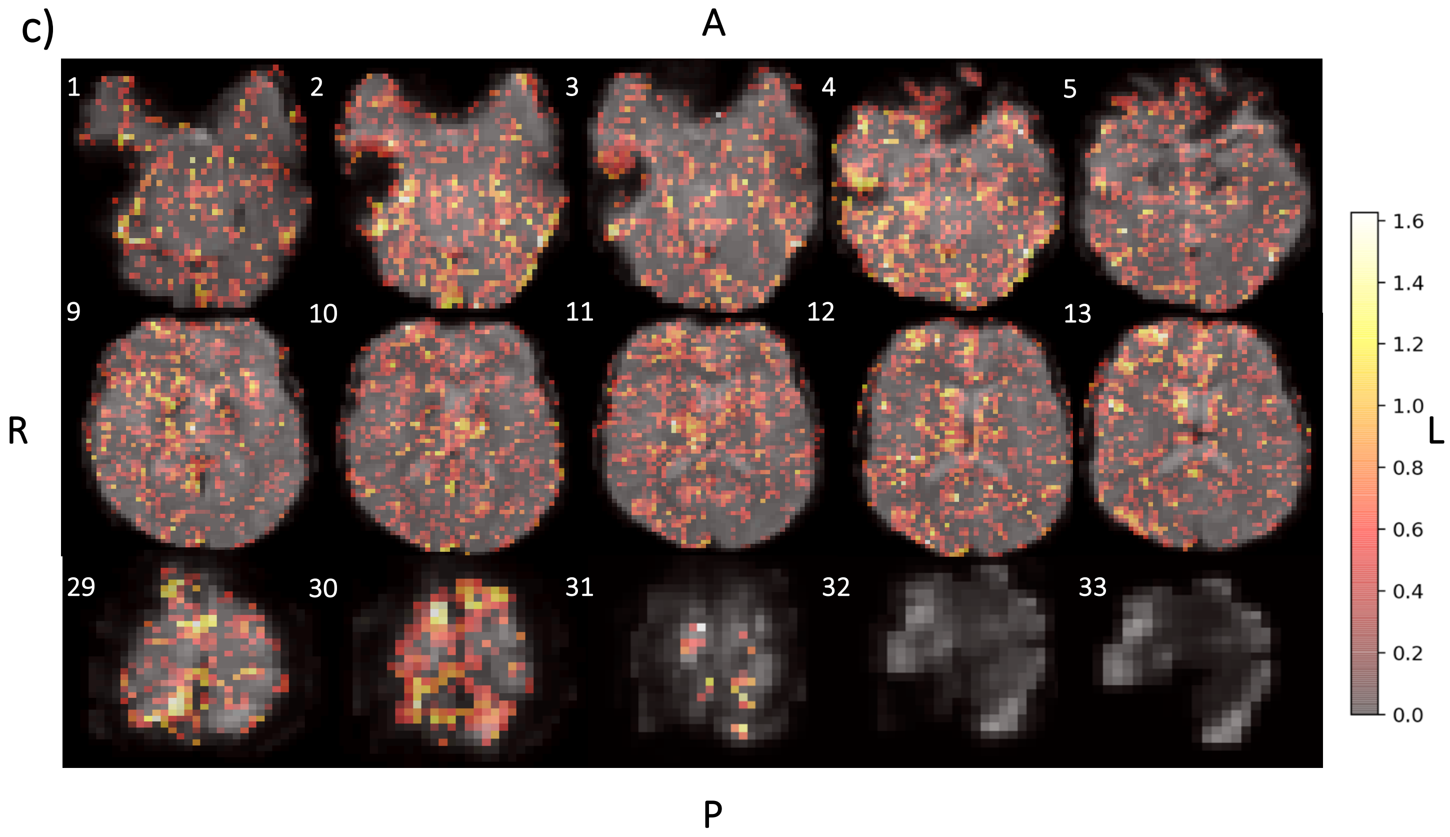
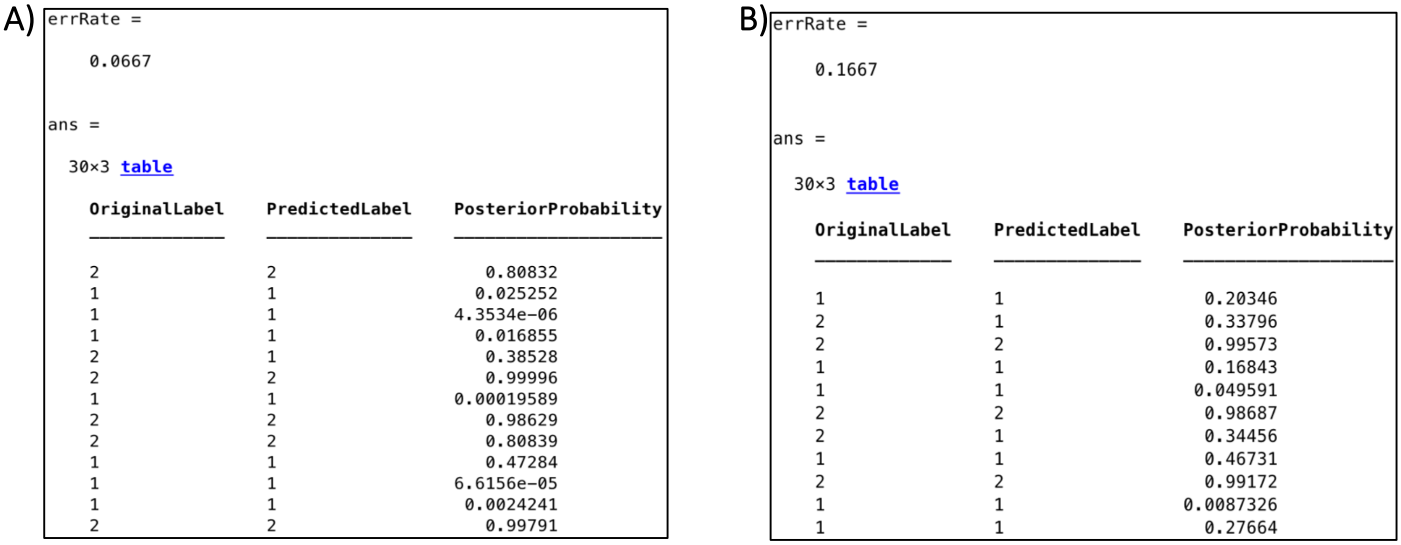


Figure 5.3: Displays the overlay T maps slices 1-5, 9-13, and 29-33 onto the FMRI slices (Transverse axis). Each of the displayed collage of t-maps have labels A, P, R, and L which indicate the Anterior, Posterior, Right, and Left sides of the brain. The colored voxels are those with t-values greater than the 50th percentile of the entire distribution of t-values plus 1.5 the in IQR. This is shown for (a) Regressors with no time delay with respect to the times of MI activity, (b) Regressors with an induced 6 second time delay with respect to times of MI activity, and (c) The difference in influence between (b) and (a).

Overall, the T Maps show more synchronized clustered activity when the HRF is delay (6 seconds) from the time of MI activity. Slices 9-13 in figure 5.3 (b) shows most on the underlining activity being concentrated in the right hemisphere of the brain. On the other hand, the same slices in figure 5.3 (a) show rather scattered activity.

5.3 Classification of MI activity:

The binary SVM classifier in MATLAB (Mathworks Inc., Natick, MA, USA) version 2017b was used to learned features that distinguishes right hand from right feet motor imagery (MI). The 2-class classification problem was defined by the identifier set {1,2} where 1 and 2 represented an event being right hand MI or right feet MI, respectively. To reduce the effects of the curse of dimensionality, due to low amounts training data, only channels C3 and Cz were used for training. These two channels were used since they lie directly above the parietal lobe of the brain. Two types of feature vectors were used individually to examine the overall affect that the set of features had in the discriminative power of the learning algorithm. The two individual feature vectors were a time domain and a power feature vector. The learning algorithm ran 7 times, one for each run of the non-simultaneous EEG acquisition. The error rate of the algorithm was quantified as the amount of times the algorithm misclassified a type of MI over the total size of the testing set. The overall accuracy was calculated as one minus the average of all the error rates (Figure 5.4 and 5.5). The difference between right hand and right feet MI was established by the likeliness of an observation, in the testing set, to be either hand or feet MI based on a learned set of features (or Posterior Probability). If the posterior probability of an observation in the testing set was ≥ than 0.5 then the event was considered a right feet MI, anything else was considered to be right hand MI.

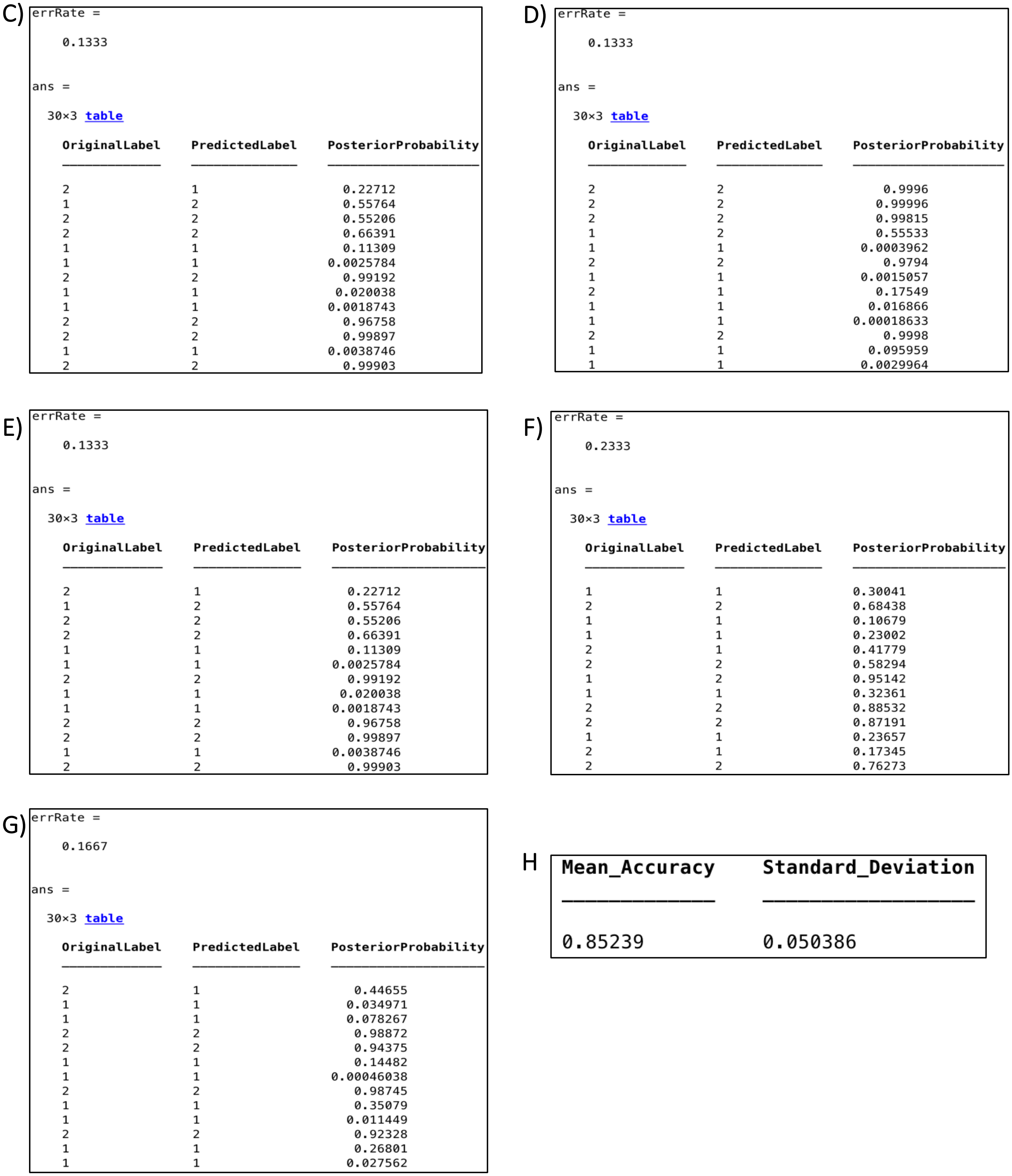


A)

```
errRate =

    0.0667


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                2                    0.80832
    1                1                   0.025252
    1                1                 4.3534e-06
    1                1                   0.016855
    2                1                    0.38528
    2                2                    0.99996
    1                1                 0.00019589
    2                2                    0.98629
    2                2                    0.80839
    1                1                    0.47284
    1                1                 6.6156e-05
    1                1                  0.0024241
    2                2                    0.99791
```

B)

```
errRate =

    0.1667


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                1                    0.20346
    2                1                    0.33796
    2                2                    0.99573
    1                1                    0.16843
    1                1                   0.049591
    2                2                    0.98687
    2                1                    0.34456
    1                1                    0.46731
    2                2                    0.99172
    1                1                  0.0087326
    1                1                    0.27664
```



C)

```
errRate =

    0.1333


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                1                    0.22712
    1                2                    0.55764
    2                2                    0.55206
    2                2                    0.66391
    1                1                    0.11309
    1                1                  0.0025784
    2                2                    0.99192
    1                1                   0.020038
    1                1                  0.0018743
    2                2                    0.96758
    2                2                    0.99897
    1                1                  0.0038746
    2                2                    0.99903
```

D)

```
errRate =

    0.1333


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                2                       0.9996
    2                2                      0.99996
    2                2                      0.99815
    1                2                      0.55533
    1                1                    0.0003962
    2                2                       0.9794
    1                1                    0.0015057
    2                1                      0.17549
    1                1                     0.016866
    1                1                   0.00018633
    2                2                       0.9998
    1                1                     0.095959
    1                1                    0.0029964
```

E)

```
errRate =

    0.1333


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                1                    0.22712
    1                2                    0.55764
    2                2                    0.55206
    2                2                    0.66391
    1                1                    0.11309
    1                1                  0.0025784
    2                2                    0.99192
    1                1                   0.020038
    1                1                  0.0018743
    2                2                    0.96758
    2                2                    0.99897
    1                1                  0.0038746
    2                2                    0.99903
```

F)

```
errRate =

    0.2333


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                1                 0.30041
    2                2                 0.68438
    1                1                 0.10679
    1                1                 0.23002
    2                1                 0.41779
    2                2                 0.58294
    1                2                 0.95142
    1                1                 0.32361
    2                2                 0.88532
    2                2                 0.87191
    1                1                 0.23657
    2                1                 0.17345
    2                2                 0.76273
```

G)

```
errRate =

    0.1667


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                1                     0.44655
    1                1                    0.034971
    1                1                    0.078267
    2                2                     0.98872
    2                2                     0.94375
    1                1                     0.14482
    1                1                  0.00046038
    2                2                     0.98745
    1                1                     0.35079
    1                1                    0.011449
    2                2                     0.92328
    1                1                     0.26801
    1                1                    0.027562
```

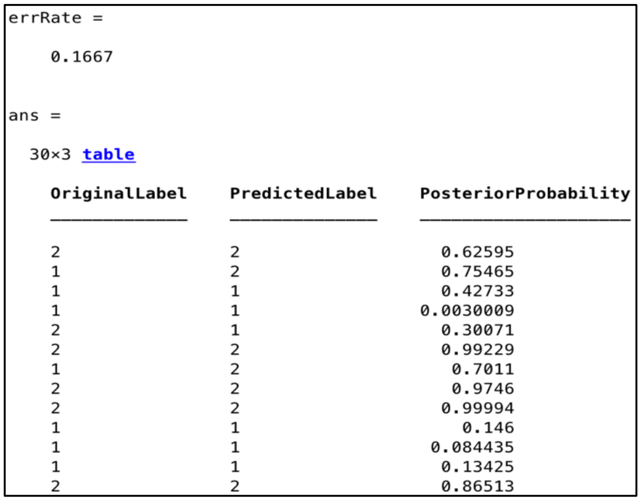
H

```
Mean_Accuracy    Standard_Deviation
_____________    __________________

0.85239          0.050386
```

Figure 5.4: Shown above are the error rates of the algorithm with respect to each of the 7 non-simultaneous EEG runs (A-G), when using a time feature vector. The 1 and 2 in both original and

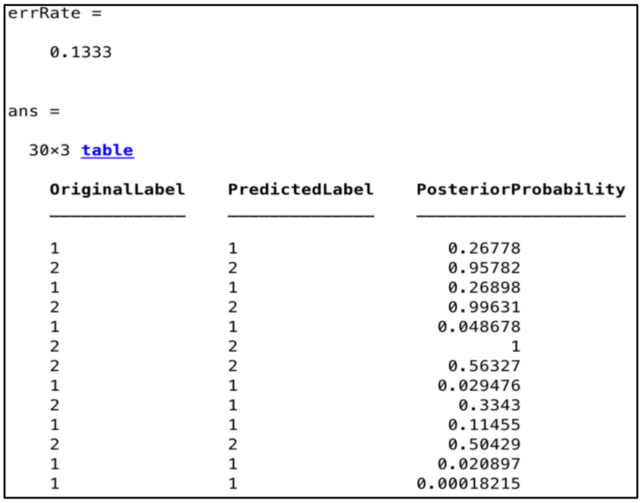
predicted labels represents a right hand MI and right feet MI, respectively. (H) Shows the overall accuracy of correctly classifying a given type of MI overall the 7 EEG runs.

A)



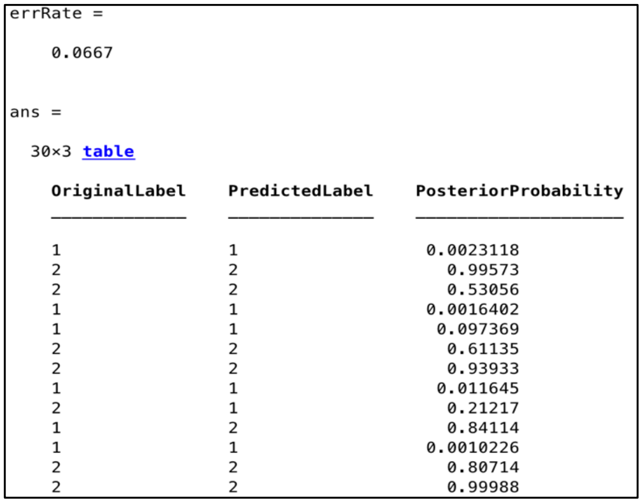
```
errRate =

    0.1667


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                2                    0.62595
    1                2                    0.75465
    1                1                    0.42733
    1                1                  0.0030009
    2                1                    0.30071
    2                2                    0.99229
    1                2                     0.7011
    2                2                     0.9746
    2                2                    0.99994
    1                1                      0.146
    1                1                   0.084435
    1                1                    0.13425
    2                2                    0.86513
```

B)



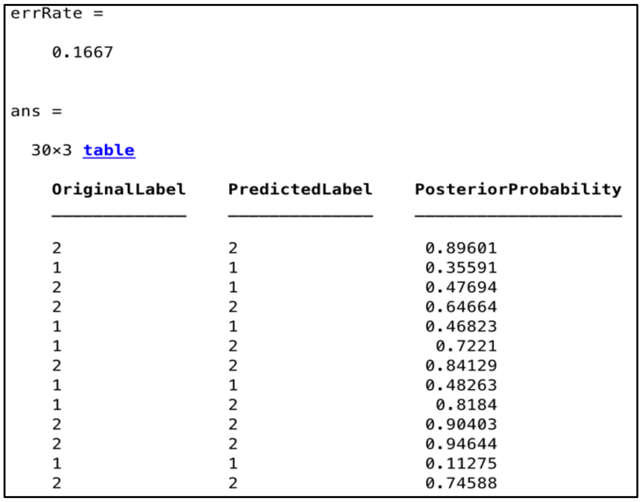
```
errRate =

    0.1333


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                1                    0.26778
    2                2                    0.95782
    1                1                    0.26898
    2                2                    0.99631
    1                1                   0.048678
    2                2                          1
    2                2                    0.56327
    1                1                   0.029476
    2                1                     0.3343
    1                1                    0.11455
    2                2                    0.50429
    1                1                   0.020897
    1                1                 0.00018215
```

C)



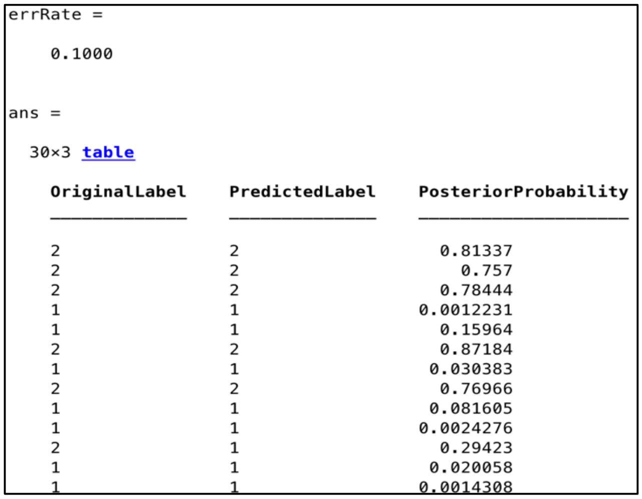
```
errRate =

    0.0667


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                1                  0.0023118
    2                2                    0.99573
    2                2                    0.53056
    1                1                  0.0016402
    1                1                   0.097369
    2                2                    0.61135
    2                2                    0.93933
    1                1                   0.011645
    2                1                    0.21217
    1                2                    0.84114
    1                1                  0.0010226
    2                2                    0.80714
    2                2                    0.99988
```

D)



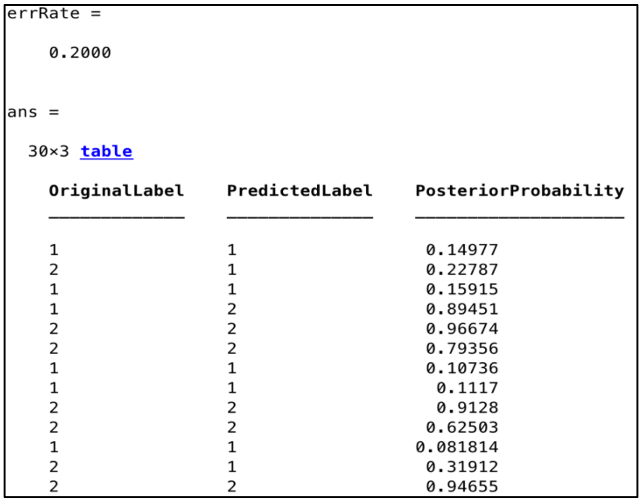
```
errRate =

    0.1667


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                2                    0.89601
    1                1                    0.35591
    2                1                    0.47694
    2                2                    0.64664
    1                1                    0.46823
    1                2                     0.7221
    2                2                    0.84129
    1                1                    0.48263
    1                2                     0.8184
    2                2                    0.90403
    2                2                    0.94644
    1                1                    0.11275
    2                2                    0.74588
```

E)

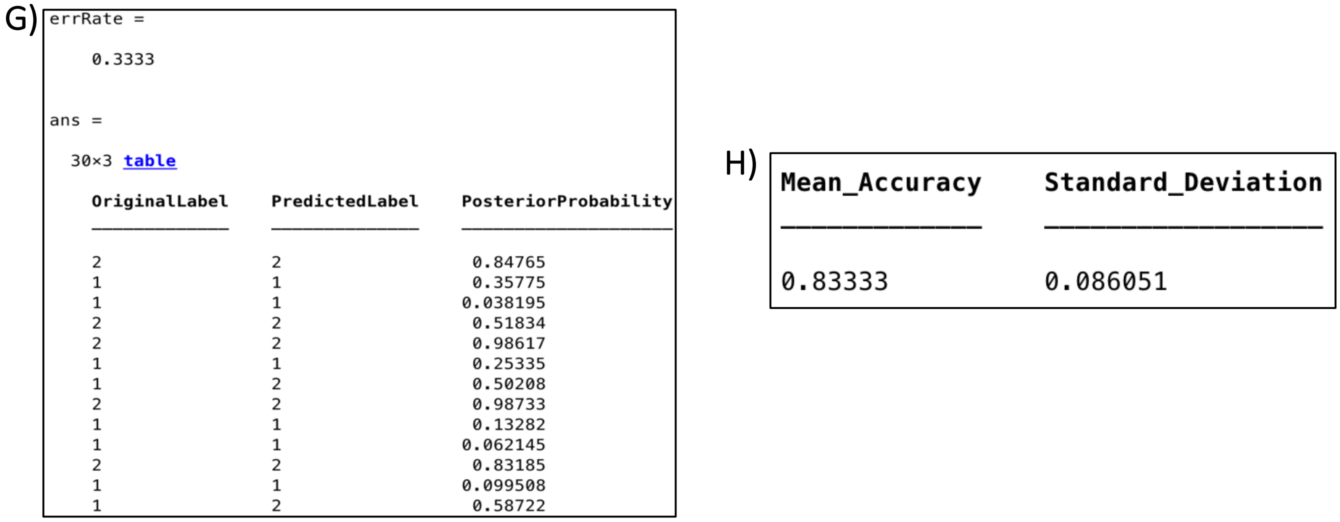


```
errRate =

    0.1000


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                2                    0.81337
    2                2                      0.757
    2                2                    0.78444
    1                1                  0.0012231
    1                1                    0.15964
    2                2                    0.87184
    1                1                   0.030383
    2                2                    0.76966
    1                1                   0.081605
    1                1                  0.0024276
    2                1                    0.29423
    1                1                   0.020058
    1                1                  0.0014308
```

F)



```
errRate =

    0.2000


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                1                    0.14977
    2                1                    0.22787
    1                1                    0.15915
    1                2                    0.89451
    2                2                    0.96674
    2                2                    0.79356
    1                1                    0.10736
    1                1                     0.1117
    2                2                     0.9128
    2                2                    0.62503
    1                1                   0.081814
    2                1                    0.31912
    2                2                    0.94655
```



G)

```
errRate =

    0.3333


ans =

  30×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                2                   0.84765
    1                1                   0.35775
    1                1                  0.038195
    2                2                   0.51834
    2                2                   0.98617
    1                1                   0.25335
    1                2                   0.50208
    2                2                   0.98733
    1                1                   0.13282
    1                1                  0.062145
    2                2                   0.83185
    1                1                  0.099508
    1                2                   0.58722
```

H)

```
Mean_Accuracy    Standard_Deviation
_____________    __________________

0.83333          0.086051
```

<u>Figure 5.5:</u> Shown above are the error rates of the algorithm with respect to each of the 7 non-simultaneous EEG runs (A-G), when using a power feature vector. The 1 and 2 in both original and predicted labels represents a right hand MI and right feet MI, respectively. (H) Shows the overall accuracy of correctly classifying a given type of MI overall the 7 EEG runs.

The overall mean accuracy obtained for both the time domain and power feature vectors were 0.8524 (or 85.24%) and 0.8333 (or 83.33%), respectively. The difference in the mean accuracies between both features was of 0.0196 (or 1.96 %). These indicates that there does not appear to be a difference in the mean accuracy of correctly classifying the two MI based on using either a time domain or power feature vector.

5.4 Implementation of AAS_{GA} Algorithm:

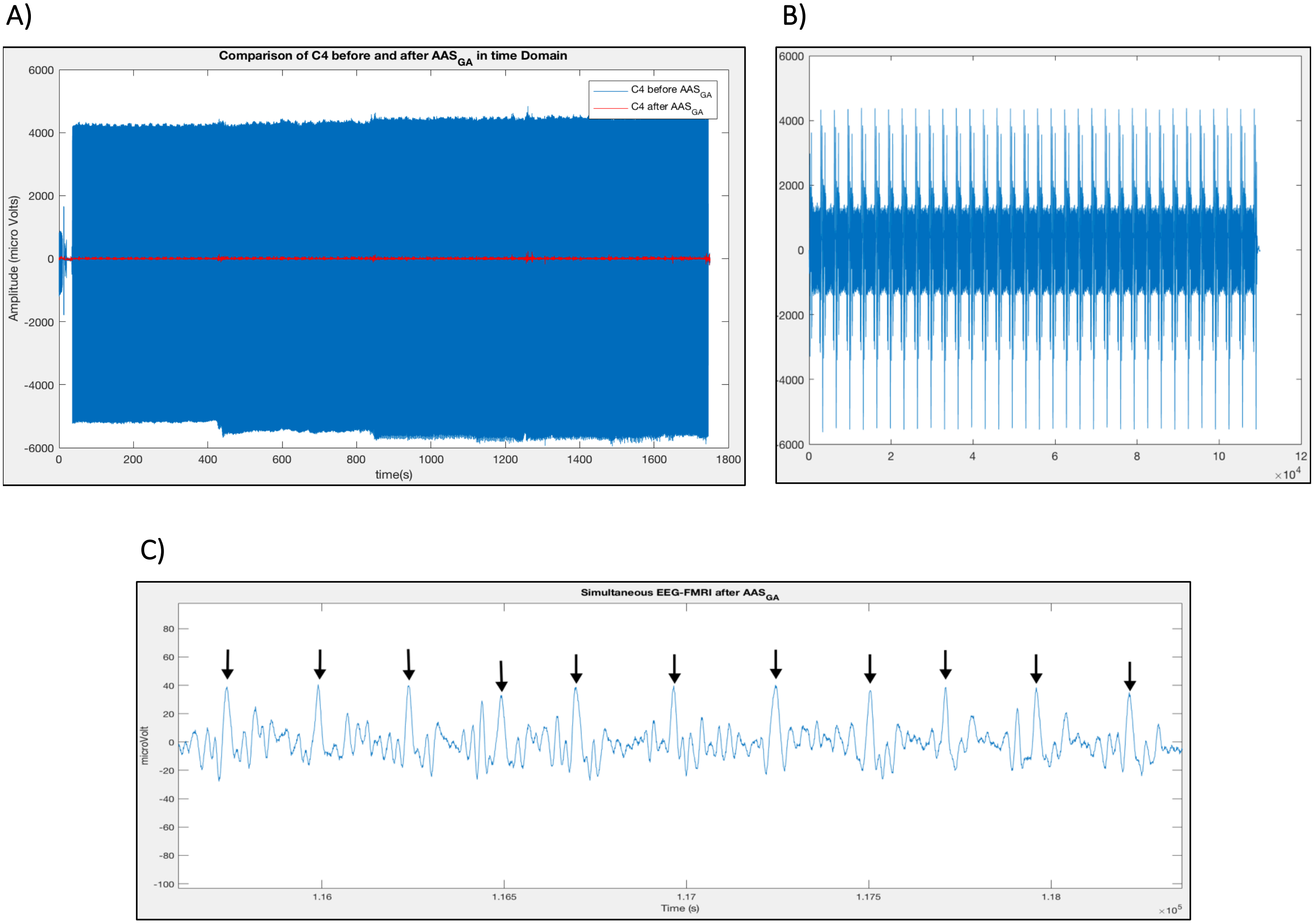


Figure 5.6: Generated plots show (A) a comparison, in the time domain, between the raw EEG signal in C4 channel (Blue) and the signal after Average Artifact Subtraction of GA (Red), (B) the calculated GA Template, and (C) the remaining Pulsatile artifacts (indicated by the black arrows) after the subtraction of the GA template from the signal.

5.5 Implementation of GMM-EM $_{\text{Wiener Filter}}$:

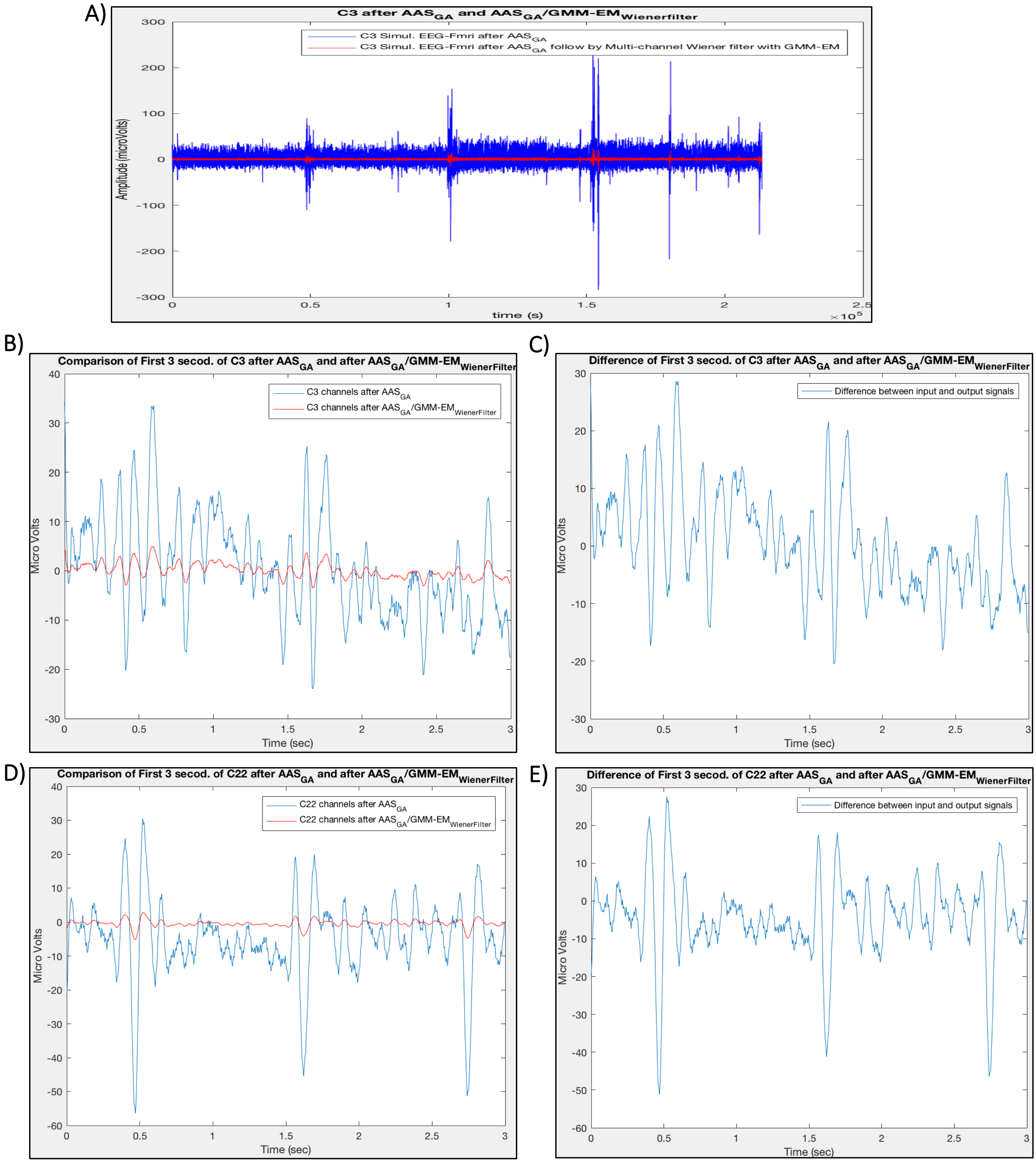
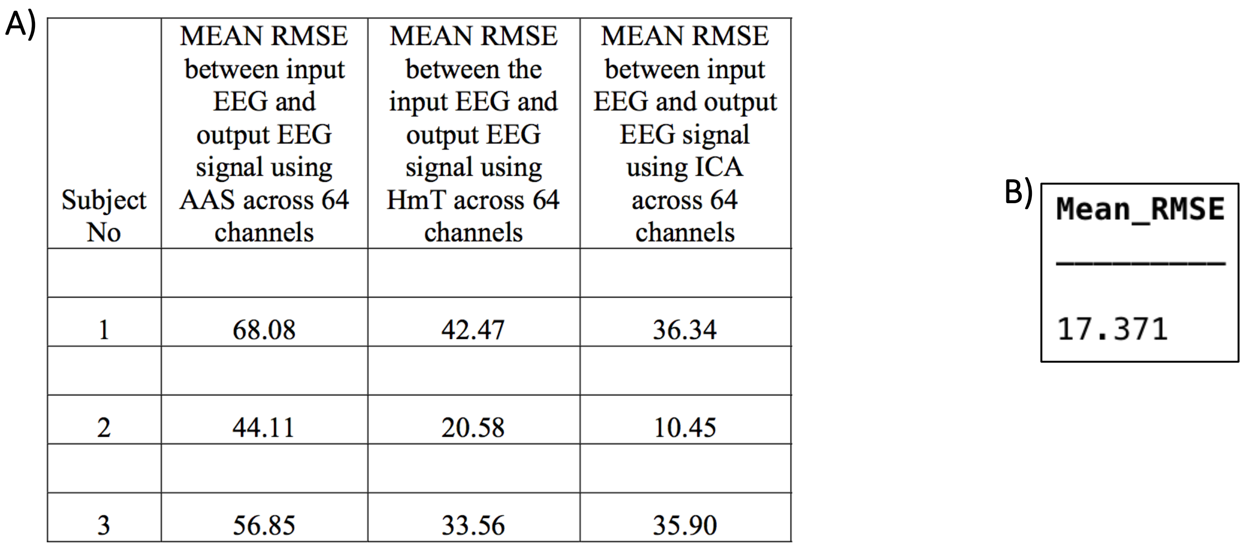


Figure 5.7: The Generated plot shows comparisons in the time domain of (A) the down-sampled C3 channel EEG signal after AAS_{GA} (Blue) and after AAS_{GA}/$\text{GMM-EM}_{\text{Wiener Filter}}$ (Red), (B) the first 3 seconds of the EEG signal at channel C3 after AAS_{GA} (Blue) and after AAS_{GA}/$\text{GMM-EM}_{\text{Wiener Filter}}$ (Red), (C) the difference between each of the signals in (B), (D) the first 3 seconds of the EEG signal at Channel C22 after AAS_{GA} (Blue) and after AAS_{GA}/$\text{GMM-EM}_{\text{Wiener Filter}}$ (Red), and (E) the difference between each of the signals in (D).

To quantitatively assess the effectiveness of $\text{GMM-EM}_{\text{Wiener Filter}}$ to suppress PA artifacts the root-mean-square-error (RMSE) was calculated for each EEG channel (Appendix). The mean RMSE, over all the EEG channels, was computed to compare it with the mean RMSE presented by Mahadevan, 2008 (Figure 5.8) [29].



A)

Subject No	MEAN RMSE between input EEG and output EEG signal using AAS across 64 channels	MEAN RMSE between the input EEG and output EEG signal using HmT across 64 channels	MEAN RMSE between input EEG and output EEG signal using ICA across 64 channels
1	68.08	42.47	36.34
2	44.11	20.58	10.45
3	56.85	33.56	35.90

B)

```
Mean_RMSE
_________

17.371
```

<u>Sources:</u> Mahadevan, 2008

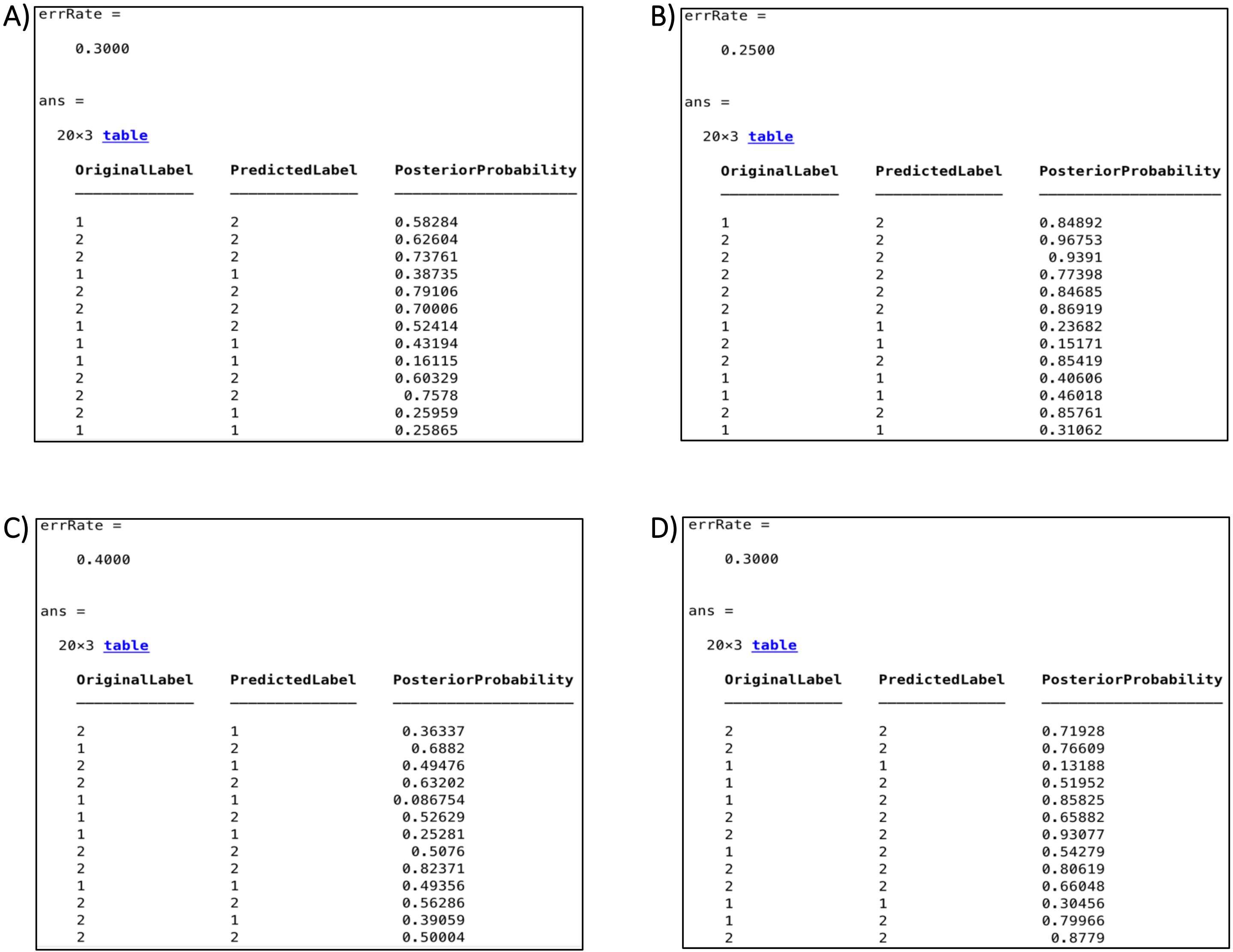
Figure 5.8: (A) Table shows the mean RMSE values of 64 EEG channels after applying AAS_{PA}, Dilated Discrete Hermite Transform Method, and ICA for PA removal of 3 different subjects' EEG activity (Mahadevan, 2008) [29]. (B) Shows the mean RMSE value of the 31 channels, of our subject, after implementation of the $\text{GMM-EM}_{\text{Wiener Filter}}$.

A statistical comparison between the efficiency of $\text{GMM-EM}_{\text{Wiener Filter}}$ technique to remove the PA and the techniques presented by Mahadevan, 2008 is not possible. The reason being that variation due to participant's EEG activity and electrode positions are not accounted for.

Therefore, one cannot establish the verdict that the GMM-EM $_{\text{Wiener Filter}}$ significantly reduces the PA compare to the other methods. However, the almost all the mean RMSE for each of the PA removal techniques examined by Mahadevan, 2008 are larger than the mean RMSE values calculated for the GMM-EM $_{\text{Wiener Filter}}$. Therefore, GMM-EM $_{\text{Wiener Filter}}$ could potentially be better than most techniques at suppressing the PA from simultaneous EEG signals.

<u>5.6 Classification of MI activity:</u>

To carry out classification, the same approach as described in section 5.3 was used. However, the learning algorithm ran 4 times, since the non-simultaneous EEG data only consisted of 4 runs rather than 7 runs. The classification involved the same two individual feature vectors of time and power. In this section, the average accuracies obtained after just AAS$_{\text{GA}}$ and AAS$_{\text{GA}}$/GMM-EM $_{\text{Wiener Filter}}$ were compared for the two types of features (Figure 5.9 and Figure 5.10).



A)

```
errRate =

    0.3000


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                2                 0.58284
    2                2                 0.62604
    2                2                 0.73761
    1                1                 0.38735
    2                2                 0.79106
    2                2                 0.70006
    1                2                 0.52414
    1                1                 0.43194
    1                1                 0.16115
    2                2                 0.60329
    2                2                  0.7578
    2                1                 0.25959
    1                1                 0.25865
```

B)

```
errRate =

    0.2500


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                2                 0.84892
    2                2                 0.96753
    2                2                  0.9391
    2                2                 0.77398
    2                2                 0.84685
    2                2                 0.86919
    1                1                 0.23682
    2                1                 0.15171
    2                2                 0.85419
    1                1                 0.40606
    1                1                 0.46018
    2                2                 0.85761
    1                1                 0.31062
```

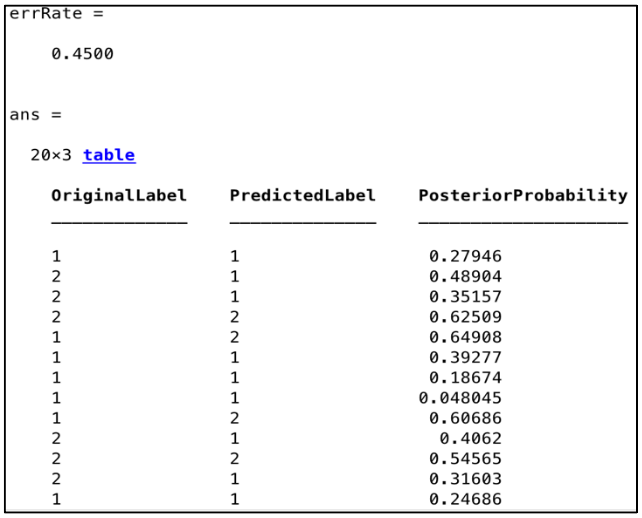
C)

```
errRate =

    0.4000


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                1                  0.36337
    1                2                   0.6882
    2                1                  0.49476
    2                2                  0.63202
    1                1                 0.086754
    1                2                  0.52629
    1                1                  0.25281
    2                2                   0.5076
    2                2                  0.82371
    1                1                  0.49356
    2                2                  0.56286
    2                1                  0.39059
    2                2                  0.50004
```

D)

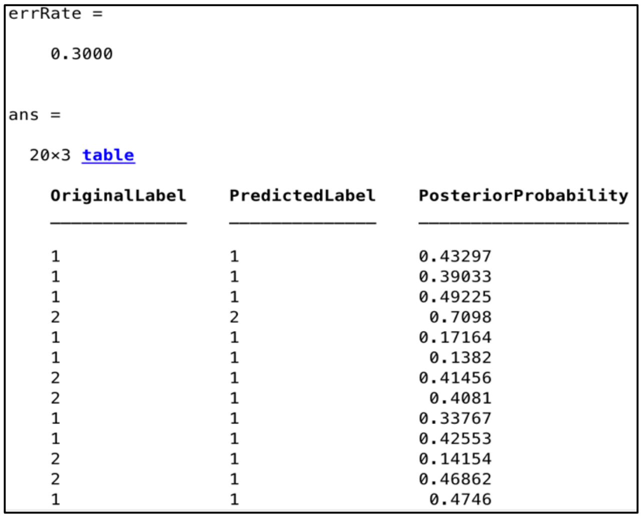
```
errRate =

    0.3000


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                2                 0.71928
    2                2                 0.76609
    1                1                 0.13188
    1                2                 0.51952
    1                2                 0.85825
    2                2                 0.65882
    2                2                 0.93077
    1                2                 0.54279
    2                2                 0.80619
    2                2                 0.66048
    1                1                 0.30456
    1                2                 0.79966
    2                2                  0.8779
```

E)



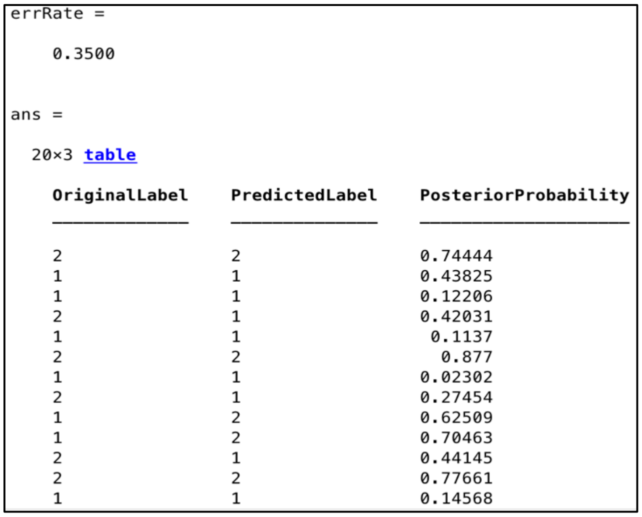
```
errRate =

    0.4500


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                1                   0.27946
    2                1                   0.48904
    2                1                   0.35157
    2                2                   0.62509
    1                2                   0.64908
    1                1                   0.39277
    1                1                   0.18674
    1                1                  0.048045
    1                2                   0.60686
    2                1                    0.4062
    2                2                   0.54565
    2                1                   0.31603
    1                1                   0.24686
```

F)



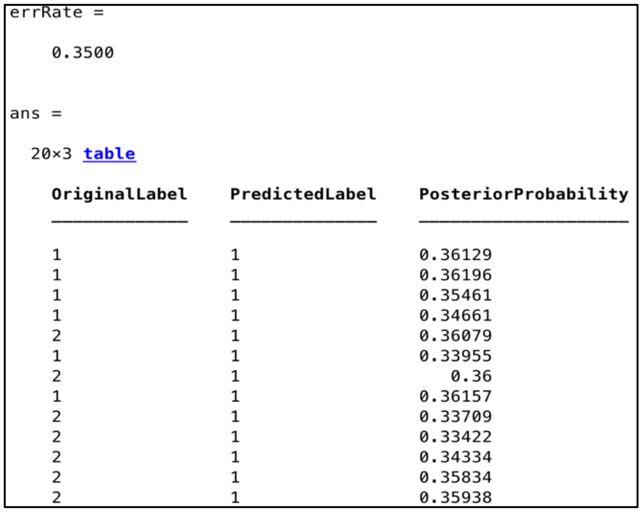
```
errRate =

    0.3000


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                1                  0.43297
    1                1                  0.39033
    1                1                  0.49225
    2                2                   0.7098
    1                1                  0.17164
    1                1                   0.1382
    2                1                  0.41456
    2                1                   0.4081
    1                1                  0.33767
    1                1                  0.42553
    2                1                  0.14154
    2                1                  0.46862
    1                1                   0.4746
```

G)



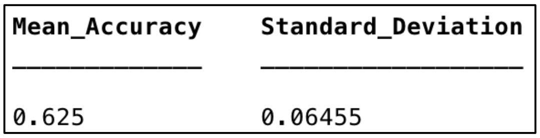
```
errRate =

    0.3500


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                2                  0.74444
    1                1                  0.43825
    1                1                  0.12206
    2                1                  0.42031
    1                1                   0.1137
    2                2                    0.877
    1                1                  0.02302
    2                1                  0.27454
    1                2                  0.62509
    1                2                  0.70463
    2                1                  0.44145
    2                2                  0.77661
    1                1                  0.14568
```

H)



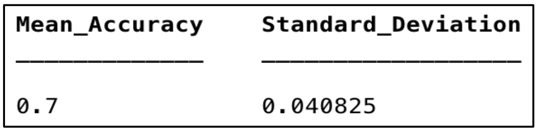
```
errRate =

    0.3500


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                1                  0.36129
    1                1                  0.36196
    1                1                  0.35461
    1                1                  0.34661
    2                1                  0.36079
    1                1                  0.33955
    2                1                     0.36
    1                1                  0.36157
    2                1                  0.33709
    2                1                  0.33422
    2                1                  0.34334
    2                1                  0.35834
    2                1                  0.35938
```

I)



```
Mean_Accuracy    Standard_Deviation
_____________    __________________

0.625            0.06455
```

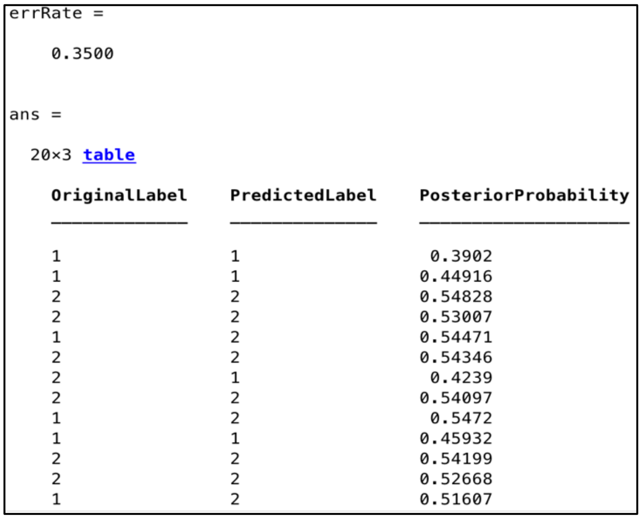
J)



```
Mean_Accuracy    Standard_Deviation
_____________    __________________

0.7              0.040825
```

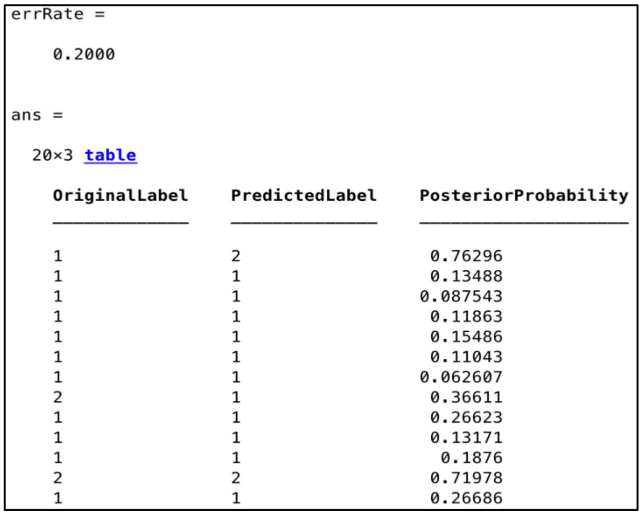
Figure 5.9: Shown above are the error rates of the algorithm with respect to each of the simultaneous EEG runs (A-J), when using a time feature vector. The 1 and 2 in both original and predicted labels represents a right hand MI and right feet MI, respectively. All the charts on the left side were the results from data after the implementation of AAS$_{GA}$ only. The charts on the right are results obtained after both AAS$_{GA}$ and GMM-EM $_{\text{Wiener Filter}}$ were performed. (I-J) Shows the overall accuracy of correctly classifying a given type of MI overall the 4 EEG runs.

A)



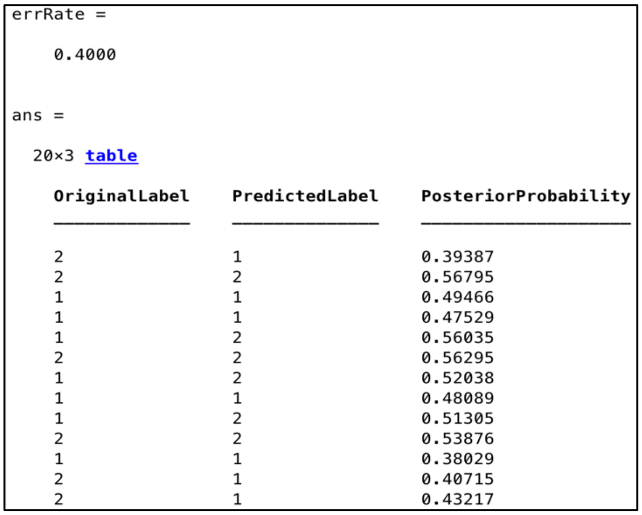
```
errRate =

    0.3500


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                1                  0.3902
    1                1                 0.44916
    2                2                 0.54828
    2                2                 0.53007
    1                2                 0.54471
    2                2                 0.54346
    2                1                  0.4239
    2                2                 0.54097
    1                2                  0.5472
    1                1                 0.45932
    2                2                 0.54199
    2                2                 0.52668
    1                2                 0.51607
```

B)



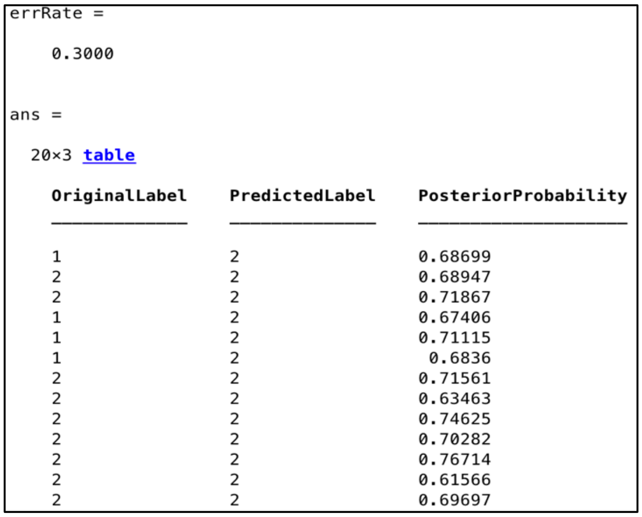
```
errRate =

    0.2000


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                2                  0.76296
    1                1                  0.13488
    1                1                 0.087543
    1                1                  0.11863
    1                1                  0.15486
    1                1                  0.11043
    1                1                 0.062607
    2                1                  0.36611
    1                1                  0.26623
    1                1                  0.13171
    1                1                   0.1876
    2                2                  0.71978
    1                1                  0.26686
```

C)



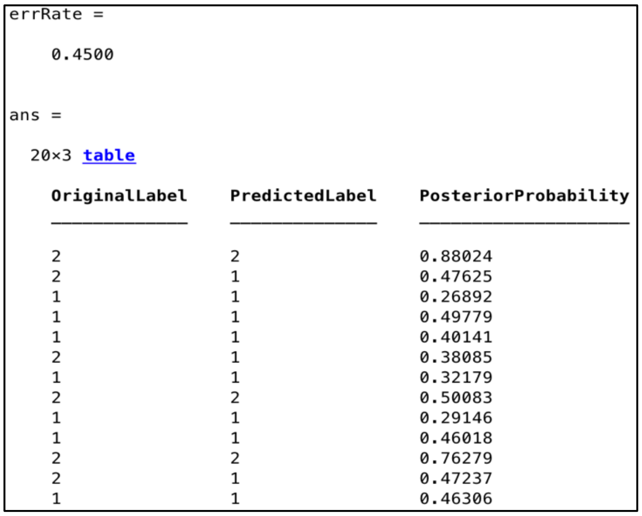
```
errRate =

    0.4000


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                1                 0.39387
    2                2                 0.56795
    1                1                 0.49466
    1                1                 0.47529
    1                2                 0.56035
    2                2                 0.56295
    1                2                 0.52038
    1                1                 0.48089
    1                2                 0.51305
    2                2                 0.53876
    1                1                 0.38029
    2                1                 0.40715
    2                1                 0.43217
```

D)



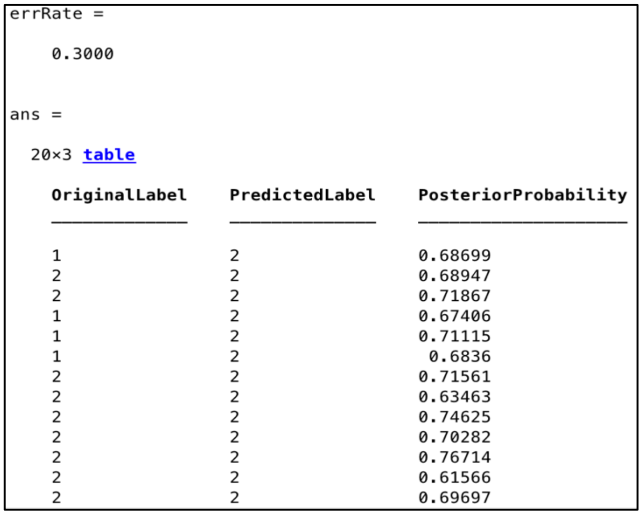
```
errRate =

    0.3000


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                2                 0.68699
    2                2                 0.68947
    2                2                 0.71867
    1                2                 0.67406
    1                2                 0.71115
    1                2                  0.6836
    2                2                 0.71561
    2                2                 0.63463
    2                2                 0.74625
    2                2                 0.70282
    2                2                 0.76714
    2                2                 0.61566
    2                2                 0.69697
```

E)

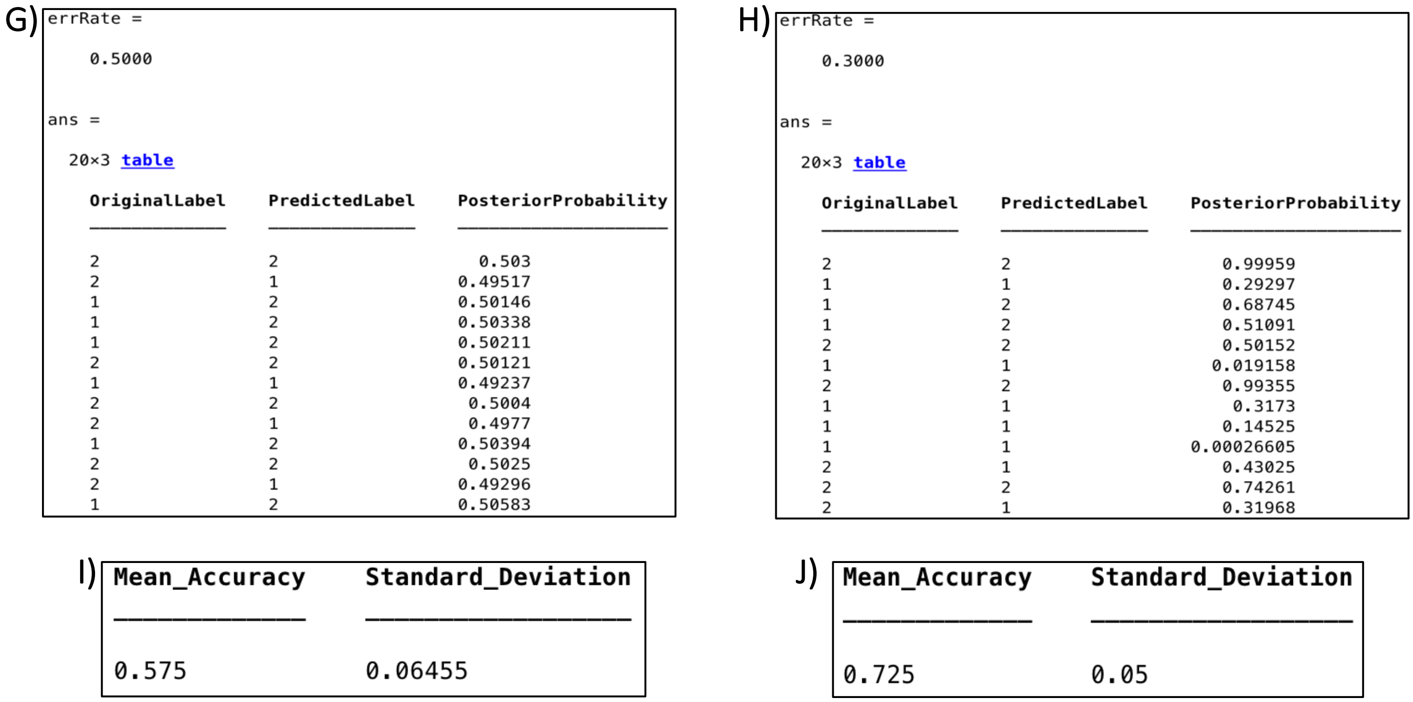


```
errRate =

    0.4500


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                2                 0.88024
    2                1                 0.47625
    1                1                 0.26892
    1                1                 0.49779
    1                1                 0.40141
    2                1                 0.38085
    1                1                 0.32179
    2                2                 0.50083
    1                1                 0.29146
    1                1                 0.46018
    2                2                 0.76279
    2                1                 0.47237
    1                1                 0.46306
```

F)



```
errRate =

    0.3000


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    1                2                 0.68699
    2                2                 0.68947
    2                2                 0.71867
    1                2                 0.67406
    1                2                 0.71115
    1                2                  0.6836
    2                2                 0.71561
    2                2                 0.63463
    2                2                 0.74625
    2                2                 0.70282
    2                2                 0.76714
    2                2                 0.61566
    2                2                 0.69697
```



G)

```
errRate =

    0.5000


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                2                   0.503
    2                1                 0.49517
    1                2                 0.50146
    1                2                 0.50338
    1                2                 0.50211
    2                2                 0.50121
    1                1                 0.49237
    2                2                  0.5004
    2                1                  0.4977
    1                2                 0.50394
    2                2                  0.5025
    2                1                 0.49296
    1                2                 0.50583
```

H)

```
errRate =

    0.3000


ans =

  20×3 table

    OriginalLabel    PredictedLabel    PosteriorProbability
    _____________    ______________    ____________________

    2                2                      0.99959
    1                1                      0.29297
    1                2                      0.68745
    1                2                      0.51091
    2                2                      0.50152
    1                1                     0.019158
    2                2                      0.99355
    1                1                       0.3173
    1                1                      0.14525
    1                1                   0.00026605
    2                1                      0.43025
    2                2                      0.74261
    2                1                      0.31968
```

I)

```
Mean_Accuracy    Standard_Deviation
_____________    __________________

0.575            0.06455
```

J)

```
Mean_Accuracy    Standard_Deviation
_____________    __________________

0.725            0.05
```

Figure 5.10: Shown above are the error rates of the algorithm with respect to each of the simultaneous EEG runs (A-J), when using a power feature vector. The 1 and 2 in both original and predicted labels represents a right hand MI and right feet MI, respectively. All the charts on the left side were the results from data after the implementation of AAS_{GA} only. The charts towards the right are results obtained after both AAS_{GA} and GMM-EM $_{\text{Wiener Filter}}$ were performed. (I-J) Shows overall accuracy of correctly classifying a given type of MI overall the 4 EEG runs.

The average classification accuracies for the AAS_{GA} and AAS_{GA}/GMM-EM $_{\text{Wiener Filter}}$ processed signals, using the time domain feature vector, were 0.625 (or 62.6%) and 0.700 (or 70.0%); respectively. The mean accuracies using an estimated power feature vector were 0.575 (or 57.5 %) and 0.725 (or 72.5%), respectively. The difference between the four categories may not seem substantial, since the standard deviation is rather large. This could be due to small amount of training set used for the classification. However, it is interesting to observe differences in overall classification performance between the two artifact removal stages for the same type feature vector. The differences in accuracies between artifact removal stages for a given time domain and power feature vector were 0.075 (or 7.5 %) and 0.150 (or 15.0 %), respectively. To determine whether a parametric or non-parametric significance test should be use a one-sided t test was performed to determine if the both populations came from a normal distribution. The test failed to reject the null hypothesis for the group using a time domain feature vector. On the other hand, the t-test did reject the null for the group using a power feature vector for classification. This means that the differences in accuracy between the two stages for the time domain and power feature vectors were normally distributed and non-normally distributed, respectively. The non-parametric Wilcoxon rank sum test was employed to examine if a significant difference between these two groups could be established. The test failed to reject the null hypothesis, with p-value of 0.143 (p-value > 0.05), establishing that there was no significant difference between the two groups.

Conclusion:

The results of the GLM does seem demonstrate high levels of activity in slices 29 and 30, which could represent the sections of the Parietal lobe. However, to conclusively determine if the correlated voxels are present in brain regions responsible for motor imagery the T-maps must be overlay onto a structural MRI brain volume and a 3D rendering of the maps most be created. A way to possibly obtain more representative T-maps of MI activity is to use the energy of the simultaneous EEG signal were MI was performed as a regressor for the GLM.

The offline classification of the two different MI activity in channels Pz and CP1 from the non-simultaneous EEG show to deliver good overall mean accuracies. The mean accuracies were in the 80's range, which lies in the higher range of possible offline classification accuracies as stated in Pfurtscheller, 2006. Although, lower average accuracies were obtained for C22 and C21 in the processed simultaneous EEG data, they still lied within the specified range of 50.8% to 93.3% [28]. These results are rather reasonable when accounting for the fact that, as of now, not a single technique can fully recuperate the EEG activity from the massive artifacts present in the EEG recording. Therefore, it can be said that more time needs to be invested in the development of new techniques. The technique employed in this report, for the Pulsatile Artifact (PA) removal, was GMM-EM $_{\text{Wiener Filter}}$. This technique seems to have potential to recover the underlining EEG activity within the mu and beta ranges. This idea is reflected through both the acquired mean RMSE value and acquired classification accuracies. The technique may be better at recuperating the underlining power of the EEG signal, than its waveform present in the time domain as demonstrated in figure 3.3.10. This, in principle, is because the Wiener Filter works by suppressing the power of the noise (or artifact) but cannot itself recover properties of the waveform of the signal that have been distorted by the noise [25]. If this is the case, then the classification of MI activity using power features should be significantly better the MI classification using time domain features. The non-parametic Wilcoxon significance test was used to see if there was a significant difference between the classification accuracies of obtained by the use of either a time domain or power features. The Wilcoxon test dictated that no significant difference was found between the accuracies obtain by using time domain or power features.

However, the non-significance differences between both set of feature might be a result of the small data set used to establish significance. Therefore, a similar study needs to be conducted on a larger dataset to establish if there is truly no significant difference between using a time domain feature vector or a power feature vector when classifying MI activity after the application of $\text{GMM-EM}_{\text{Wiener Filter}}$.

References:

[1] Robert W Cox, Functional MRI Analysis. “Absolute Beginner’s Guide to Anatomical and Functional MRI of the Brain”, *ISMRM*, pp. 1-8 (2011).

[2] Petra Ritter, Robert Becker, Frank Freyer, and Arno Villringer. EEG Quality: The Image Acquisition Artifact. EEG-fMRI Physiological Basis, Technique, and Application (2010). *Springer,* pp. 153-172

[3] Stafan Debener, Cornelia Kranczioch, and Ingmar Gutberlet. EEG Quality: Origin and Reduction of the EEG Cardiac-Related Artifact. EEG-fMRI Physiological Basis, Technique, and Application (2010). *Springer,* pp. 135-152.

[4] Christian-G. Bénar, Andrew P. Bagshaw, and Louis Lemieux. Experimental Design and Data Analysis Strategies. EEG-fMRI Physiological Basis, Technique, and Application (2010). *Springer,* pp. 221-258.

[5] Hongmi Lee and Brice A. Kuhl. “Reconstruction Perceived and Retrieved Faces from Activity Patterns in Lateral Parietal Cortex”, *Journal of Neuroscience,* Vol. 36, pp.6069-6082 (2016).

[6] John S. O’Brian and E. Lois Sampson. “Lipid composition of the normal human brain: gray matter, white matter, and myelin”, *Journal of Lipid Research*, Vol. 6, pp. 537-544 (1965).

[7] Rik Henson, “Analysis of fMRI Timeseries: Linear Time-Invarient Model, Events-related fMRI and Optimal Experimental Design” (2003). *In: Frackowiak, RSJ and Friston, KJ and Frith, C and Dolan, R and Price, C, (eds).* Human Brain Function (pp. 793-822). Elsevier: London.

[8] Ji Meng Loh, Martin A. Lindquist, and Tor D. Wager. “Residual Analysis for Detecting Mis-modeling in fMRI”, *Statistica Sinica,* Vol. 18, pp. 1421-1448 (2008).

[9] Russell A. Poldrack, Jeanette A. Mumford, Thomas E. Nichols. Image processing basics. Handbook of Functional MRI Data Analysis (2011). *Cambridge University* Press. pp. 13-33.

[10] K.J. Friston, A.P. Holmes, K.J. Worsley, J.-P. Poline, C.D. Frith, and R.S.J. Frackowiak. “Statistical Parametric Maps in Functional Imaging: A General Linear Approach”, *Human Brain Mapping*, Vol. 2, pp.189-210 (1995).

[11] Mohammad H. Alomari, Aya Samaha, and Khaled AlKamha. “Automated Classification of L/R Hand Movement EEG Signals using Advanced Feature Extraction and Machine Learning”,

International Journal of Advance Computer Science and Applications, Vol. 4, pp.207-212 (2013).

[12] Philip J. Allen, Oliver Josephs, and Robert Turner. "A Method for Removing Imaging Artifact from Continuous EEG Recorded during Functional MRI", *NueroImage*, Vol. 12, pp.230-239 (2000).

[13] Philip J. Allen, Giovanni Polizzi, Karsten Krakow, David R. Fish, and Louis Lemieux. "Identification of EEG Events in the MR Scanner: The Problem of Pulse Artifact and a Method for its Subtraction", *NeuroImage*, Vol. 8, pp. 229-239 (1998).

[14] Yvonne Höller, Jürgen Bergmann, Aljoscha Thomschewski, Martin Kronbichler, Peter Höller, Julia S. Crone, Elisabeth V. Schmid, Kevin Butz, Raffaele Nardone, Eugen Trinka. "Comparison of EEG-Features and Classification Methods for Motor Imagery in Patients with Disorders of Consciousness", *PLoS One,* Vol. 8, pp. 1-15 (2013).

[15] Wolfgang Taube, Michael Mouthon, Christian Leukel, Henri-Marcel Hoogewoud, Jean-Marie Annoni, and Martin Keller. "Brain activity during observation and motor imagery of different balance tasks: An fMRI study", *Cortex*, Vol. 64, pp. 102-114 (2015).

[16] Fabien Lotte. A Tutorial on EEG Signal Processing Techniques for Mental State Recognition in Brain-Computer Interfaces. Eduardo Reck Miranda; Julien Castet. Guide to Brain-Computer Music Interfacing, Springer, (2014). <hal-01055103>

[17] Michael J. Rosenfeld, OLS in Matrix Form, *NYU Lecture Notes*, (2013)

[18] François Lazeyras, Ivan Zimine, Olaf Blake, Stephen H. Perrig, and Margitta Seeck. "Functional MRI With Simultaneous EEG Recording: Feasibility and Application to Motor and Visual Activation". *Journal of Magnetic Resonance Imaging*, Vol 13, pp.943-948, (2001)

[19] Yaliang Ma, Xiaohui Ding, Qingshan She, Zhizeng Lou, Thomas Potter, and Yingchun Zhang. "Classification of Motor Imagery EEG Signals with Support Vector Machines and Particle Swarm Optimization", *Hindawi Publishing Corporation Computational and Mathematical Methods in Medicine,* Vol 1, pp. 1-8, (2016).

[20] K. –L. Du and M. N. S. Swamy, Fundamentals of Machine Learning, Neural Networks and Statistical Learning (2014), *Spring*, pp.15-67 (2014).

[21] Steyrl D, Wriessnegger S C, Müller-Putz G R. “Single Trial Motor Imagery Classification in EEG Measure During FMRI Image Acquisition – A First Glance”, *Biomed Tech*, Vol 58, pp. 1-3 (2013).

[22] David Steyrl, Franz Patz, Gunther Krausz, Günter Edlinger, and Gernot R. Müller-Putz. “Reduction of EEG Artifacts in Simultaneous EEG-FMRI Reference Layer Adaptive Filtering”, *IEEE,* pp. 3803- 3806 (2015).

[23] David Steryl, Gunther Krausz, Karl Koschutnig, Günter Edlinger, and Gernot R Müller-Putz. “Reference layer adaptive filtering for EEG artifact reduction in simultaneous EEG-FMRI”, *J. Neural Eng*, Vol. 14, pp. 1-20 (2017).

[24] Hayato Maki, Tomaki Toda, Sakriani Sakti, Graham Neubig, Satoshi Nakamura, “EEG Signal Enhancement using Multi-Channel Weiner Filter with a Spatial Correlation Prior”, *IEEE*, pp. 2639 – 2643 (2015)

[25] Scott C. Douglas and Danilo P. Mandic. “Single-Channel Wiener Filtering of Deterministic Signals in Stochastic Noise using the Panarame”, *IEEE,* pp. 4182-4186, (2017).

[26] Wiener Filters. Advance Digital Signal Processing and Noise Reduction (pp. 179 – 204), Second Edition. Saeed V. Vaseghi. *John Wiley & Sons Ltd*, (2000)

[27] Ramesh Sridharan. Gaussian mixture models and the EM algorithm. Introduction to Probability (pp. 1-11). Bertsekas & Tsitsiklis, (2008).

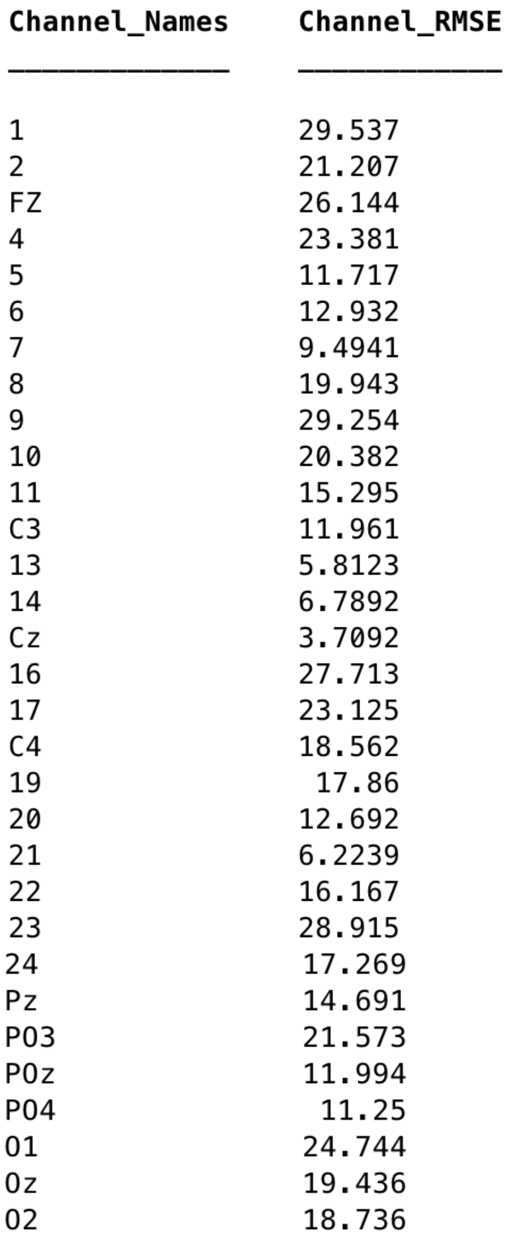
[28] G. Pfurtscheller, G.R. Müller-Putz, A. Schlögl, B.Graimann, R.Scherer, R.Leeb, C.Brunner, C. Kainrath, F. Lee, G. Townsend, C. Vadaurre, and C. Neuper. “15 Years of PA Research at Graz University of Technology: Current Projects”, *IEEE Transactions on Neural System and Rehabilitation Engineering,* Vol. 14, NO. 2, pp. 205 – 209, (2006)

[29] Anandi Mahabevan, “Real Time Ballistocardiogram Artifact Removal in EEG-FMRI using Dilated Discrite Hermite Transform”, *John Hopkins Medicine-School of Medicine*, pp. 1-128 (2008).

[30] Christoph Binder, Thomas Haslwanter, and Brain D. Schmit. “Measurement and analysis of cortical activity using EEG during upper limb stabilization”, *Marshall Plan Foundation*, pp. 1-56 (2015)

Appendix:

Results of Section 5.5:



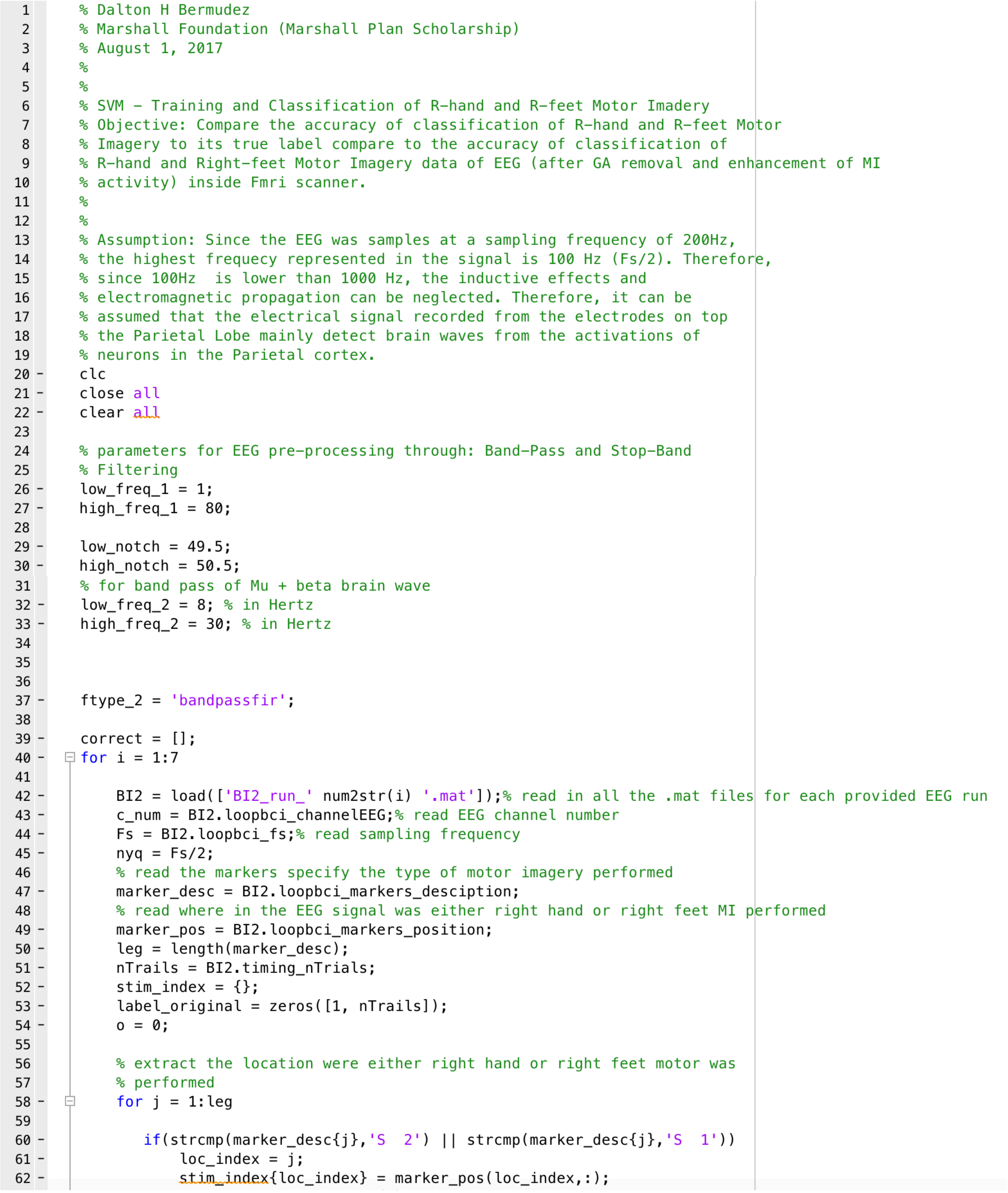
```
Channel_Names    Channel_RMSE
_____________    ____________

1                29.537
2                21.207
FZ               26.144
4                23.381
5                11.717
6                12.932
7                9.4941
8                19.943
9                29.254
10               20.382
11               15.295
C3               11.961
13               5.8123
14               6.7892
Cz               3.7092
16               27.713
17               23.125
C4               18.562
19                17.86
20               12.692
21               6.2239
22               16.167
23               28.915
24                17.269
Pz                14.691
PO3               21.573
POz               11.994
PO4                11.25
O1                24.744
Oz                19.436
O2                18.736
```

Developed Python and Matlab Scripts:

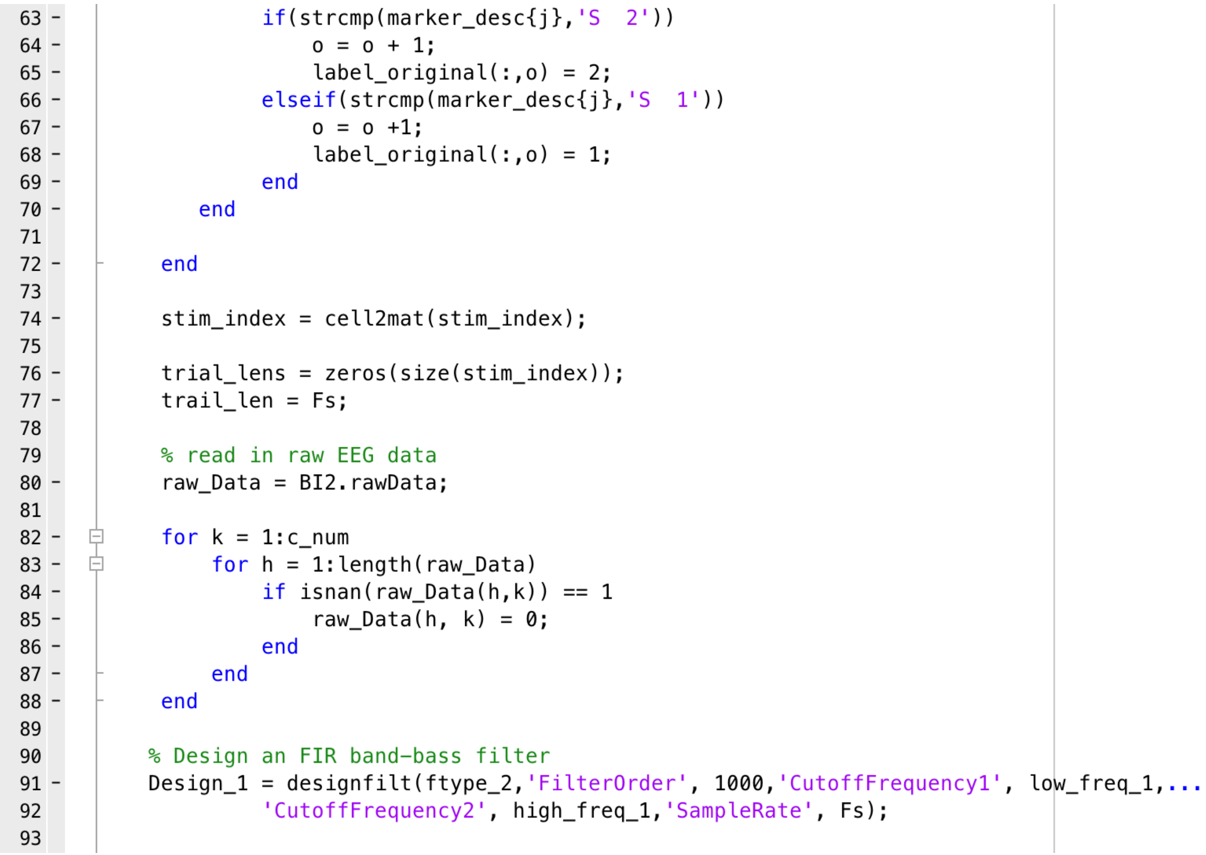
```
import scipy.fftpack as scp_fft
import scipy.io as sio
import matplotlib.pyplot as plt
import scipy.ndimage
from scipy.interpolate import griddata
from scipy import ndimage as nd

index_on = []
index_off = []
stimulation_times = []
stml_times = []
Beta_best = []
est_residuals = []
mean_square_residules = []


# skull stripping of the Fmri images
btr = fsl.BET()
btr.inputs.in_file = output_nifti_file
btr.inputs.frac = 0.6 # default from fsl (fractional intensity threshold)
btr.inputs.functional = True
btr.inputs.out_file = '...' # specify the output path were new threshold FMRI will be saved
res = btr.run()

# Spatial smoothing of the 4D fMRI to increase SNR
sus = fsl.SUSAN()
sus.inputs.in_file = '...' # Specify the path were the 4 dimensional FMRI file is located
sus.inputs.fwhm = 4.0 # FWHM of smoothing, in mm, gets converted using sqrt(8*log(2)) -single subject analysis
                      # While larger kernals in range of 6-10 mm should be consider for multi-subject analysis
                      # to account for intersubject variablility
sus.inputs.brightness_threshold = 2000 # Brightness threshold and should be greater than
                                             #noise level and less than contrast of edges to be preserved.
sus.inputs.out_file = '...'# specify the output path were new threshold FMRI will be saved
rlt = sus.run()

# read in motion corrected, skulled stripped and slice time corrected FMRI data
MOCO_fmri = nib.load('...') # specify path were the corrected FMRI 4-Dimensional volume is located
hdr_MOCO_fmri = MOCO_fmri.header # read the header of the 4-Dimensional FMRI volume
raw_func = hdr_MOCO_fmri.structarr
[dim_x_fmri, dim_y_fmri, dim_z_fmri, dim_t_fmri] = raw_func['dim'][1:5] #dimension sides of the 4-D FMRI data
data = MOCO_fmri.get_data() # read in the data from the nib files

# create a mask to remove sorroundign voxels that show little activity:
EPI_fmri_image = np.sum(data,3)
mask_image = np.float64(EPI_fmri_image > 18000)

Bold_brain = np.zeros((dim_x_fmri, dim_y_fmri, dim_z_fmri, dim_t_fmri,1))

# performing GLM on the fMRI activity: (refromaning matrix for GLM)
# reference slice by physiological structure from Brain activity during observation and motor imagery of different balance tasks:
# An fMRI study
for i in range(0, dim_x_fmri):
    for j in range(0, dim_y_fmri):
        for k in range(0, dim_z_fmri):
            Bold_brain[i,j,k,:,:] = (data[i,j,k,:].reshape(64, 1))

# creating a design matrix for GLM
base_line = np.ones(Bold_brain.shape[3:5])
# using Mark Cohen's gamma-variate function as HRF
t = np.arange(30) # typical duration of the Bold signal is at least 30 s
c = 0.547
b = 8.6

# canonical hemodynamic response function (HRF) without lag:
r_t = pow(t,b)*(np.exp(-1*t/c))
# temporal derivatives of the canonical hemodynamic response function (HRF) (no lag):
r_t_prime = pow(t,b)*((-1/c)*np.exp(-1*t/c)) + np.exp(-1*t/c) * (b*pow(t,b-1))
r_t_double_prime = pow(t,b)*((1/pow(c,2))*np.exp(-1*t/c)) + ((-1/c)*(np.exp(-1*t/c))) * \
                   (b*pow(t,b-1)) + np.exp(-1*t/c) * (b*(b-1)*pow(t,b-2)) + b*pow(t,b-1) * (-1/c*np.exp(-1*t/c))


# conical hemodynamic response function (HRF) with lag of 6s - (paper: Estimating the delay of the fMRI response):
r_t_delay = np.roll(r_t, 6)
r_t_prime_delay = np.roll(r_t_prime, 6)
r_t_double_prime_delay = np.roll(r_t_double_prime, 6)

file_path_paradigm = '...' # specify the text file containing the times of the paradigm were Motor Imagery was performd
with open(file_path_paradigm) as text:
    paradigm  = text.read().splitlines()
# read the times in the paradigm text file were MI was performed as indicated by the fixated cross
for s in range(0, len(paradigm)):
    if 'cross on' in paradigm[s]:
        index_on.append(s)
    elif 'cross off' in paradigm[s]:
        index_off.append(s)
```

```
for q in range(0, len(index_both)):
    stimulation_times.append(re.findall('([\d.]+)', paradigm[index_both[q]])) # find the times and duration of stimulus
    stml_times.append([round(float(x)) for x in stimulation_times[q]])

stml_times = np.concatenate(stml_times, axis = 0)
duration_of_paradigm = 64 # first trial
stimulus = np.zeros(duration_of_paradigm)

# create the stimulus signal from the experiments paradigm
# This created the box-car function, which will be convoluted with the Mark Cohen’s gamma-variate function
for k in range(0, 7, 2):
    for h in range(0, len(stimulus)):
        if h >= stml_times[k] and h <= stml_times[k+1]:
            stimulus[h] = 1

# non - delay regressor for GLM
h_t = (np.convolve(stimulus, r_t, mode = 'same')).reshape(len(stimulus), 1)
h_t_prime = (np.convolve(stimulus, r_t_prime, mode = 'same')).reshape(len(stimulus),1)
h_t_double_prime = np.convolve(stimulus, r_t_double_prime, mode = 'same').reshape(len(stimulus),1)


# delay of 6s commen delay for the hymodynamic response after neural activation
h_t_delay = (np.convolve(stimulus, r_t_delay, mode = 'same')).reshape(len(stimulus), 1)
h_t_prime_delay = (np.convolve(stimulus, r_t_prime_delay, mode = 'same')).reshape(len(stimulus),1)
h_t_double_prime_delay = np.convolve(stimulus, r_t_double_prime_delay, mode = 'same').reshape(len(stimulus),1)

# create design matrix with all the regressors of interest
design_matrix = np.concatenate((h_t,h_t_prime,h_t_double_prime, h_t_delay, h_t_prime_delay, h_t_double_prime_delay, base_line), axis = 1)
design_matrix_rank_order = np.linalg.matrix_rank(design_matrix) #full rank matrix


for i in range(0, dim_x_fmri):
    for j in range(0, dim_y_fmri):
        for k in range(0, dim_z_fmri):
            # Ordinary Least Squares (solve for the best beta values that reduce the least square errors
            Beta_best.append((np.dot(np.linalg.pinv(np.dot(design_matrix.T ,design_matrix)), np.dot(design_matrix.T,Bold_brain[i,j,k,:,:])))) # Ordinary Least Squares


# Beta best values obtained for each design regressor
Beta_best = (np.array(Beta_best)).reshape((Bold_brain.shape[0], Bold_brain.shape[1], Bold_brain.shape[2], 7))

# computation of the estimated residule
est_residuals = Bold_brain.reshape((dim_x_fmri, dim_y_fmri, dim_z_fmri, dim_t_fmri)) - np.dot(Beta_best, design_matrix.T)


c_1 = np.array([1, 1, 1, 0, 0, 0, 0])# contrast to test the hypothesis that the no delay regressors are equal to zeros
c_2 = np.array([0, 0, 0, 1, 1, 1, 0])#  contrast to test the hypothesis that the 6 second delay regressors are equal to zeros


std_error_1 = np.array(np.zeros((dim_x_fmri, dim_y_fmri, dim_z_fmri)))
std_error_2 = np.array(np.zeros((dim_x_fmri, dim_y_fmri, dim_z_fmri)))
mean_square_error = np.array(np.zeros((dim_x_fmri, dim_y_fmri, dim_z_fmri)))
t_values_c1 = np.array(np.zeros((dim_x_fmri, dim_y_fmri, dim_z_fmri)))
t_values_c2 = np.array(np.zeros((dim_x_fmri, dim_y_fmri, dim_z_fmri)))

for i in range(0, dim_x_fmri):
    for j in range(0, dim_y_fmri):
        for k in range(0, dim_z_fmri):
            if all(Beta_best[i,j,k] != 0):

                # computation of the mean square error for each voxels unit in the FMRI volume
                mean_square_error[i,j,k] = (np.sum(pow(est_residuals[i,j,k,:],2))/len(est_residuals[i,j,k,:]))
                # calculate the standard error to calculate the t_map
                std_error_1[i,j,k] = np.sqrt(np.dot(np.dot(np.dot(mean_square_error[i,j,k],c_1),np.linalg.pinv(np.dot(design_matrix.T ,design_matrix))),c_1.T))
                std_error_2[i,j,k] = np.sqrt(np.dot(np.dot(np.dot(mean_square_error[i,j,k],c_2),np.linalg.pinv(np.dot(design_matrix.T ,design_matrix))),c_2.T))
                # compute the t-values to determine the contribution that the regressors in the GLM have on the observed bold signal variability
                t_values_c1[i,j,k] = np.dot(c_1.T,np.abs(Beta_best[i,j,k]))/std_error_1[i,j,k]
                t_values_c2[i,j,k] = np.dot(c_2.T,np.abs(Beta_best[i,j,k]))/std_error_2[i,j,k]

# determine the t-values that lie within brain structures
# for both the specified contrast vectors
t_map_HRF_no_delay = t_values_c1*mask_image
t_map_HRF_six_second_delay = t_values_c2*mask_image

# determine the 25th, 50th, andt 75th quartile of the t-value distribution for the first contrast vector
Q_1_c1 = np.percentile(t_values_c1,25)
Q_3_c1 = np.percentile(t_values_c1,75)
IQR_c1 = Q_3_c1 - Q_1_c1
Q_2_c1 = np.percentile(t_values_c1, 50)
t_c1_thres = Q_2_c1 + 1.5*IQR_c1 #The termine the t-value that is 1.5*IQR greater than the 50th percentile

# determine the 25th, 50th, andt 75th quartile of the t-value distribution for the second contrast vector
Q_1_c2 = np.percentile(t_values_c2,25)
Q_3_c2 = np.percentile(t_values_c2,75)
IQR_c2 = Q_3_c2 - Q_1_c2
Q_2_c2 = np.percentile(t_values_c2, 50)
t_c2_thres = Q_2_c2 + 1.5*IQR_c2#The termine the t-value that is 1.5*IQR greater than the 50th percentile
# Create a mask that is 1 for t-values in the T-Maps that are greater than the 50th percentile+1.5*IQR and zero everywhere else
t_value_mask_1 = (t_map_HRF_no_delay > t_c1_thres)
t_value_mask_2 = (t_map_HRF_six_second_delay > t_c2_thres)

# Threshold the T-maps with the created mask so only the t-values greater than the 50th percentile+1.5*IQR are shown
t_map_HRF_no_delay_thershold = t_value_mask_1*t_map_HRF_no_delay
t_map_HRF_six_second_delay_thershold = t_value_mask_2*t_map_HRF_six_second_delay
```

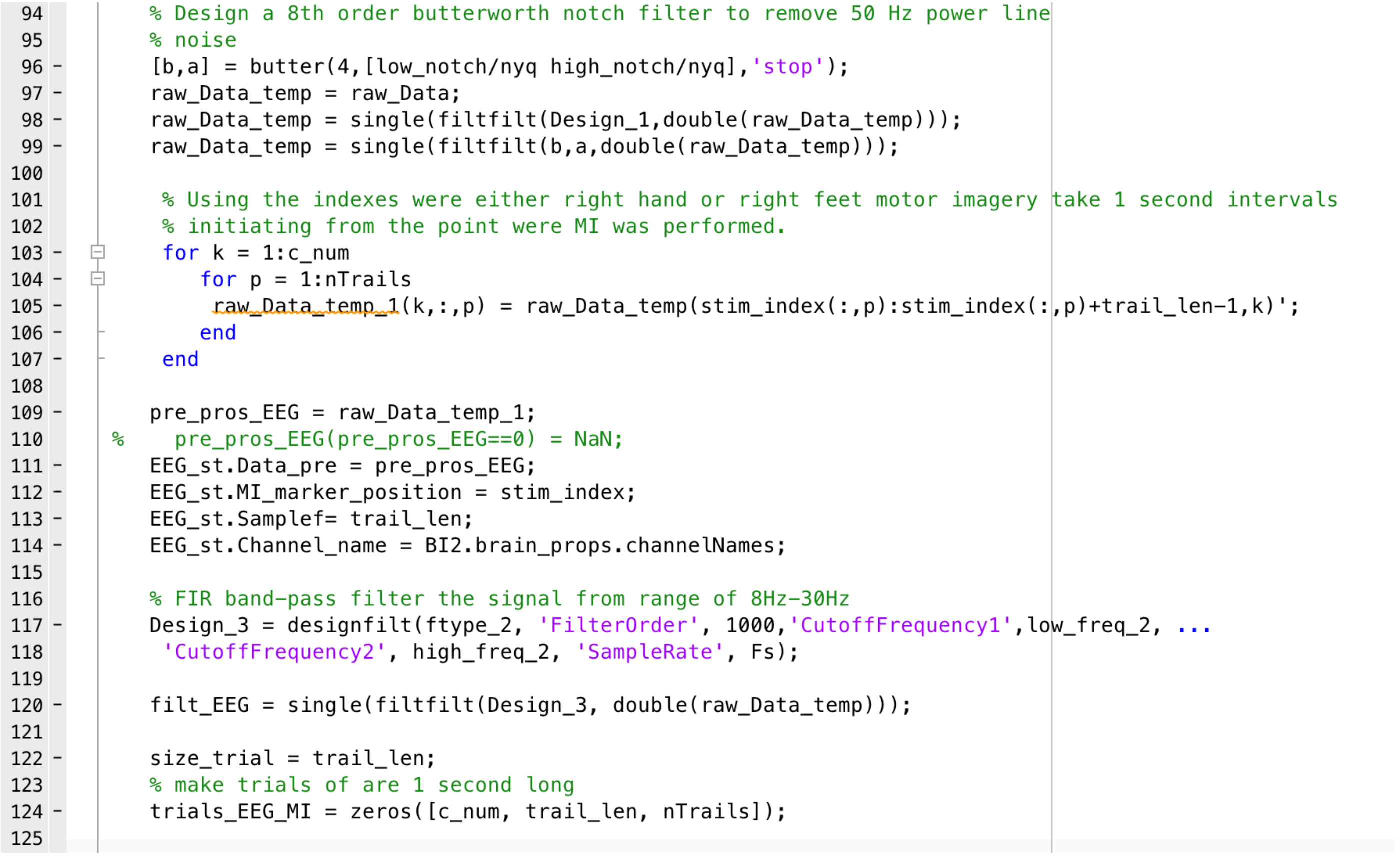
<u>Sources code of section 4.6:</u> The above python script was developed to performed the steps described in section 4.6. The script is an implementation of voxel-based correlation using Generalized Linear Model and Ordinary Least Squares. The script can only be used when analyzing data of a single subject.



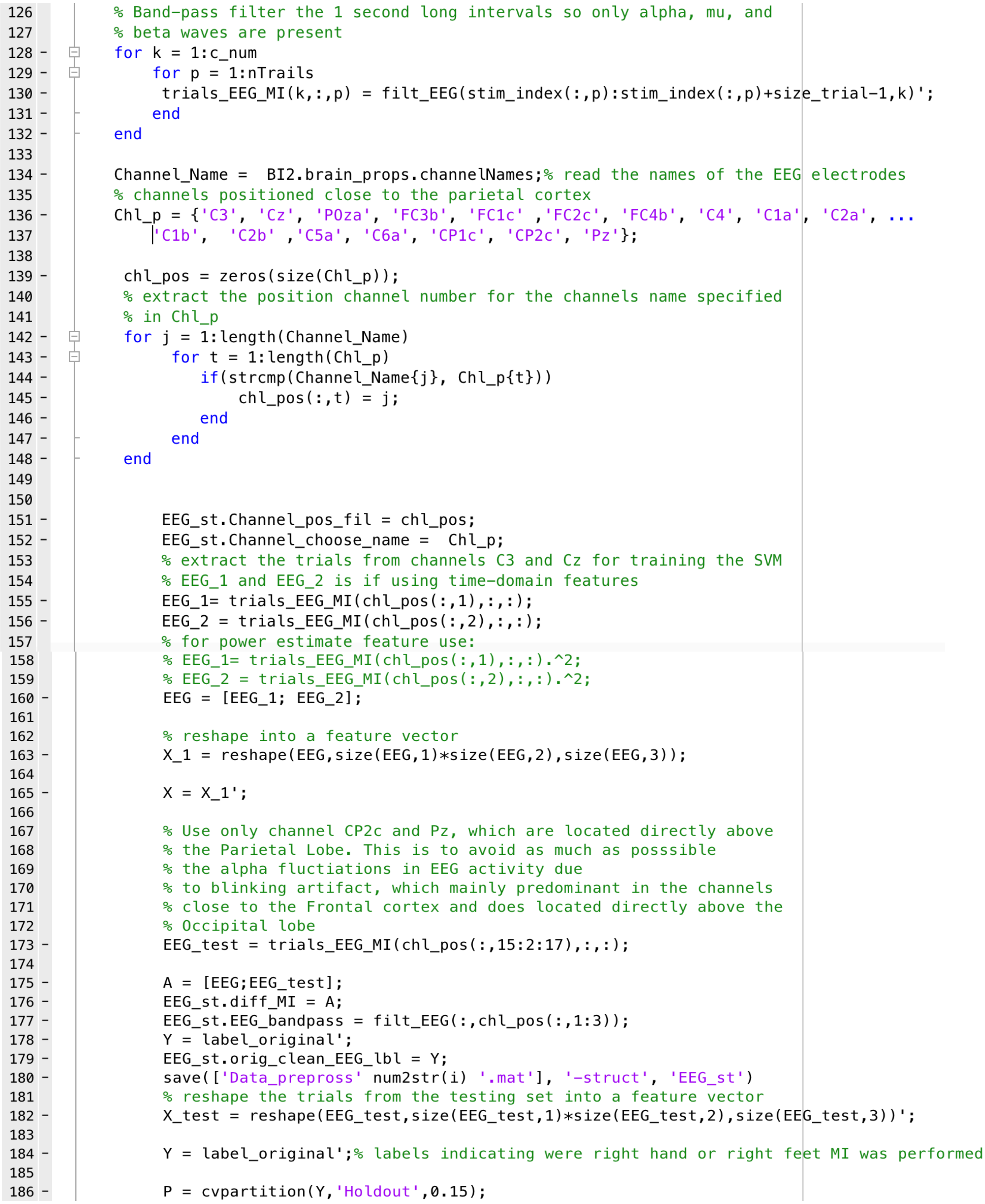
```
% Dalton H Bermudez
% Marshall Foundation (Marshall Plan Scholarship)
% August 1, 2017
%
%
% SVM - Training and Classification of R-hand and R-feet Motor Imadery
% Objective: Compare the accuracy of classification of R-hand and R-feet Motor
% Imagery to its true label compare to the accuracy of classification of
% R-hand and Right-feet Motor Imagery data of EEG (after GA removal and enhancement of MI
% activity) inside Fmri scanner.
%
%
% Assumption: Since the EEG was samples at a sampling frequency of 200Hz,
% the highest frequecy represented in the signal is 100 Hz (Fs/2). Therefore,
% since 100Hz  is lower than 1000 Hz, the inductive effects and
% electromagnetic propagation can be neglected. Therefore, it can be
% assumed that the electrical signal recorded from the electrodes on top
% the Parietal Lobe mainly detect brain waves from the activations of
% neurons in the Parietal cortex.
clc
close all
clear all

% parameters for EEG pre-processing through: Band-Pass and Stop-Band
% Filtering
low_freq_1 = 1;
high_freq_1 = 80;

low_notch = 49.5;
high_notch = 50.5;
% for band pass of Mu + beta brain wave
low_freq_2 = 8; % in Hertz
high_freq_2 = 30; % in Hertz



ftype_2 = 'bandpassfir';

correct = [];
for i = 1:7

    BI2 = load(['BI2_run_' num2str(i) '.mat']);% read in all the .mat files for each provided EEG run
    c_num = BI2.loopbci_channelEEG;% read EEG channel number
    Fs = BI2.loopbci_fs;% read sampling frequency
    nyq = Fs/2;
    % read the markers specify the type of motor imagery performed
    marker_desc = BI2.loopbci_markers_desciption;
    % read where in the EEG signal was either right hand or right feet MI performed
    marker_pos = BI2.loopbci_markers_position;
    leg = length(marker_desc);
    nTrails = BI2.timing_nTrials;
    stim_index = {};
    label_original = zeros([1, nTrails]);
    o = 0;

    % extract the location were either right hand or right feet motor was
    % performed
    for j = 1:leg

       if(strcmp(marker_desc{j},'S  2') || strcmp(marker_desc{j},'S  1'))
           loc_index = j;
           stim_index{loc_index} = marker_pos(loc_index,:);
```



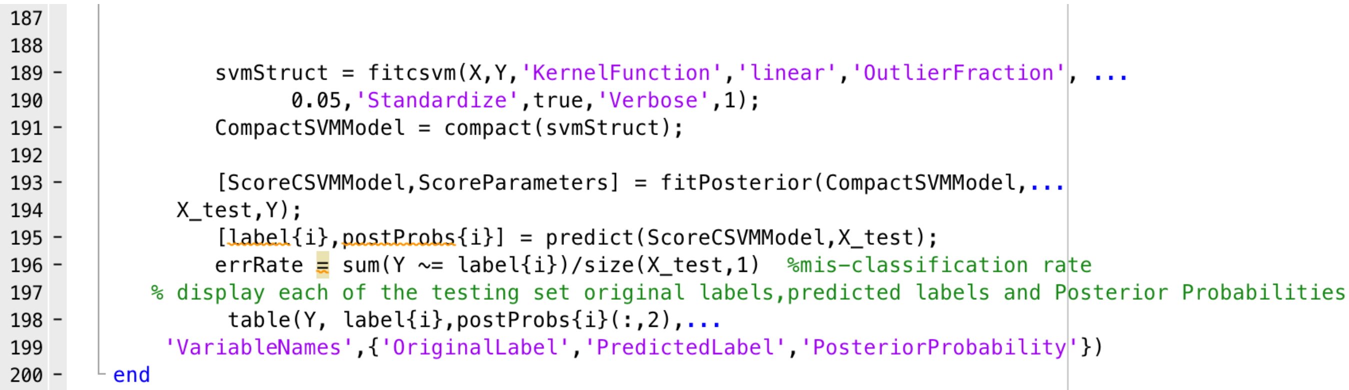
```
            if(strcmp(marker_desc{j},'S  2'))
                o = o + 1;
                label_original(:,o) = 2;
            elseif(strcmp(marker_desc{j},'S  1'))
                o = o +1;
                label_original(:,o) = 1;
            end
       end

    end

    stim_index = cell2mat(stim_index);

    trial_lens = zeros(size(stim_index));
    trail_len = Fs;

    % read in raw EEG data
    raw_Data = BI2.rawData;

    for k = 1:c_num
        for h = 1:length(raw_Data)
            if isnan(raw_Data(h,k)) == 1
                raw_Data(h, k) = 0;
            end
        end
    end

   % Design an FIR band-bass filter
   Design_1 = designfilt(ftype_2,'FilterOrder', 1000,'CutoffFrequency1', low_freq_1,...
            'CutoffFrequency2', high_freq_1,'SampleRate', Fs);

```



```
   % Design a 8th order butterworth notch filter to remove 50 Hz power line
   % noise
   [b,a] = butter(4,[low_notch/nyq high_notch/nyq],'stop');
   raw_Data_temp = raw_Data;
   raw_Data_temp = single(filtfilt(Design_1,double(raw_Data_temp)));
   raw_Data_temp = single(filtfilt(b,a,double(raw_Data_temp)));

    % Using the indexes were either right hand or right feet motor imagery take 1 second intervals
    % initiating from the point were MI was performed.
    for k = 1:c_num
       for p = 1:nTrails
        raw_Data_temp_1(k,:,p) = raw_Data_temp(stim_index(:,p):stim_index(:,p)+trail_len-1,k)';
       end
    end

   pre_pros_EEG = raw_Data_temp_1;
%    pre_pros_EEG(pre_pros_EEG==0) = NaN;
   EEG_st.Data_pre = pre_pros_EEG;
   EEG_st.MI_marker_position = stim_index;
   EEG_st.Samplef= trail_len;
   EEG_st.Channel_name = BI2.brain_props.channelNames;

   % FIR band-pass filter the signal from range of 8Hz-30Hz
   Design_3 = designfilt(ftype_2, 'FilterOrder', 1000,'CutoffFrequency1',low_freq_2, ...
    'CutoffFrequency2', high_freq_2, 'SampleRate', Fs);

   filt_EEG = single(filtfilt(Design_3, double(raw_Data_temp)));

   size_trial = trail_len;
   % make trials of are 1 second long
   trials_EEG_MI = zeros([c_num, trail_len, nTrails]);

```

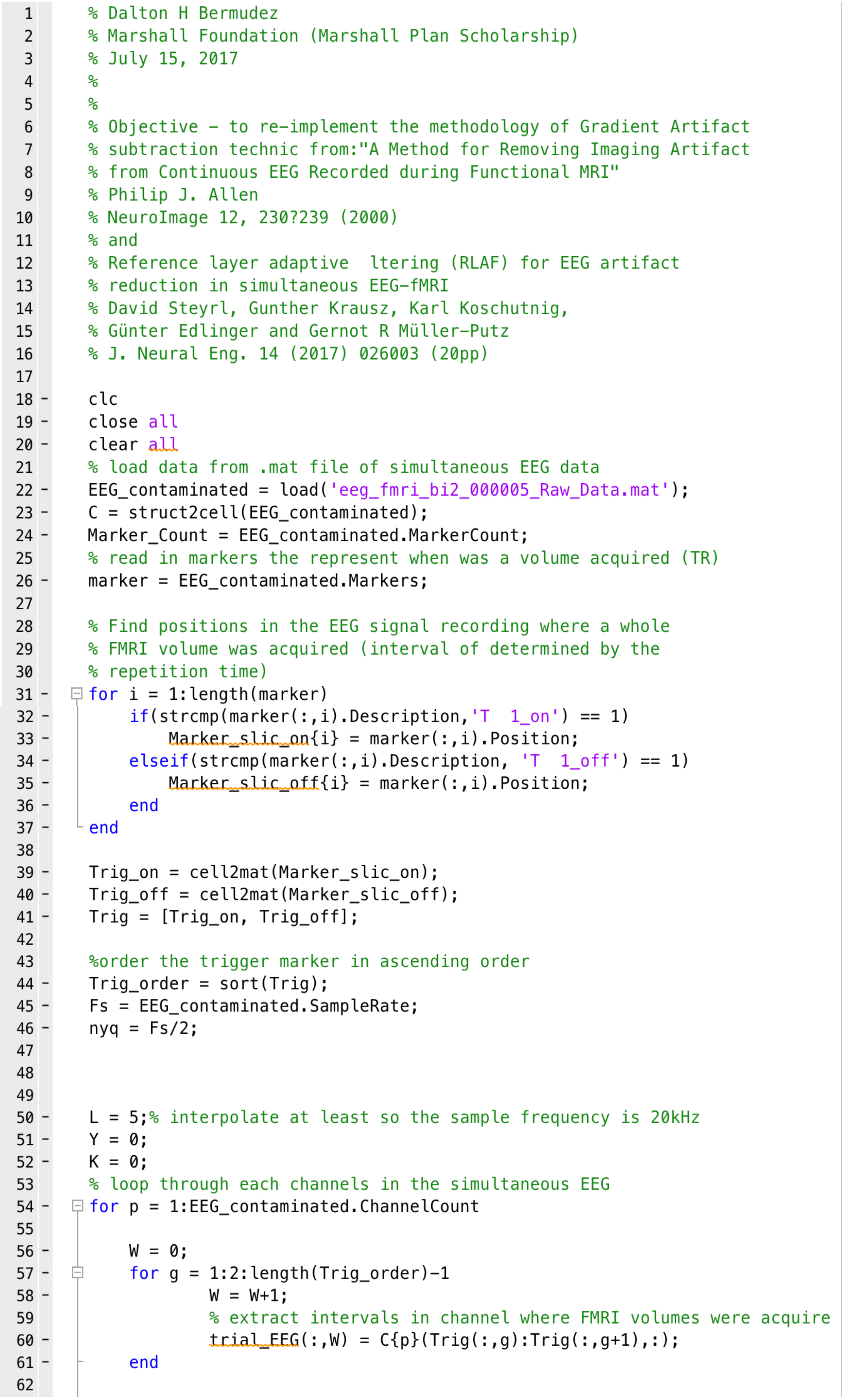


```
    % Band-pass filter the 1 second long intervals so only alpha, mu, and
    % beta waves are present
    for k = 1:c_num
        for p = 1:nTrails
         trials_EEG_MI(k,:,p) = filt_EEG(stim_index(:,p):stim_index(:,p)+size_trial-1,k)';
        end
    end

    Channel_Name =  BI2.brain_props.channelNames;% read the names of the EEG electrodes
    % channels positioned close to the parietal cortex
    Chl_p = {'C3', 'Cz', 'POza', 'FC3b', 'FC1c' ,'FC2c', 'FC4b', 'C4', 'C1a', 'C2a', ...
        'C1b',  'C2b' ,'C5a', 'C6a', 'CP1c', 'CP2c', 'Pz'};

     chl_pos = zeros(size(Chl_p));
     % extract the position channel number for the channels name specified
     % in Chl_p
     for j = 1:length(Channel_Name)
          for t = 1:length(Chl_p)
             if(strcmp(Channel_Name{j}, Chl_p{t}))
                 chl_pos(:,t) = j;
             end
          end
     end


         EEG_st.Channel_pos_fil = chl_pos;
         EEG_st.Channel_choose_name =  Chl_p;
         % extract the trials from channels C3 and Cz for training the SVM
         % EEG_1 and EEG_2 is if using time-domain features
         EEG_1= trials_EEG_MI(chl_pos(:,1),:,:);
         EEG_2 = trials_EEG_MI(chl_pos(:,2),:,:);
         % for power estimate feature use:
         % EEG_1= trials_EEG_MI(chl_pos(:,1),:,:).^2;
         % EEG_2 = trials_EEG_MI(chl_pos(:,2),:,:).^2;
         EEG = [EEG_1; EEG_2];

         % reshape into a feature vector
         X_1 = reshape(EEG,size(EEG,1)*size(EEG,2),size(EEG,3));

         X = X_1';

         % Use only channel CP2c and Pz, which are located directly above
         % the Parietal Lobe. This is to avoid as much as posssible
         % the alpha fluctiations in EEG activity due
         % to blinking artifact, which mainly predominant in the channels
         % close to the Frontal cortex and does located directly above the
         % Occipital lobe
         EEG_test = trials_EEG_MI(chl_pos(:,15:2:17),:,:);

         A = [EEG;EEG_test];
         EEG_st.diff_MI = A;
         EEG_st.EEG_bandpass = filt_EEG(:,chl_pos(:,1:3));
         Y = label_original';
         EEG_st.orig_clean_EEG_lbl = Y;
         save(['Data_preprocss' num2str(i) '.mat'], '-struct', 'EEG_st')
         % reshape the trials from the testing set into a feature vector
         X_test = reshape(EEG_test,size(EEG_test,1)*size(EEG_test,2),size(EEG_test,3))';

         Y = label_original';% labels indicating were right hand or right feet MI was performed

         P = cvpartition(Y,'Holdout',0.15);
```

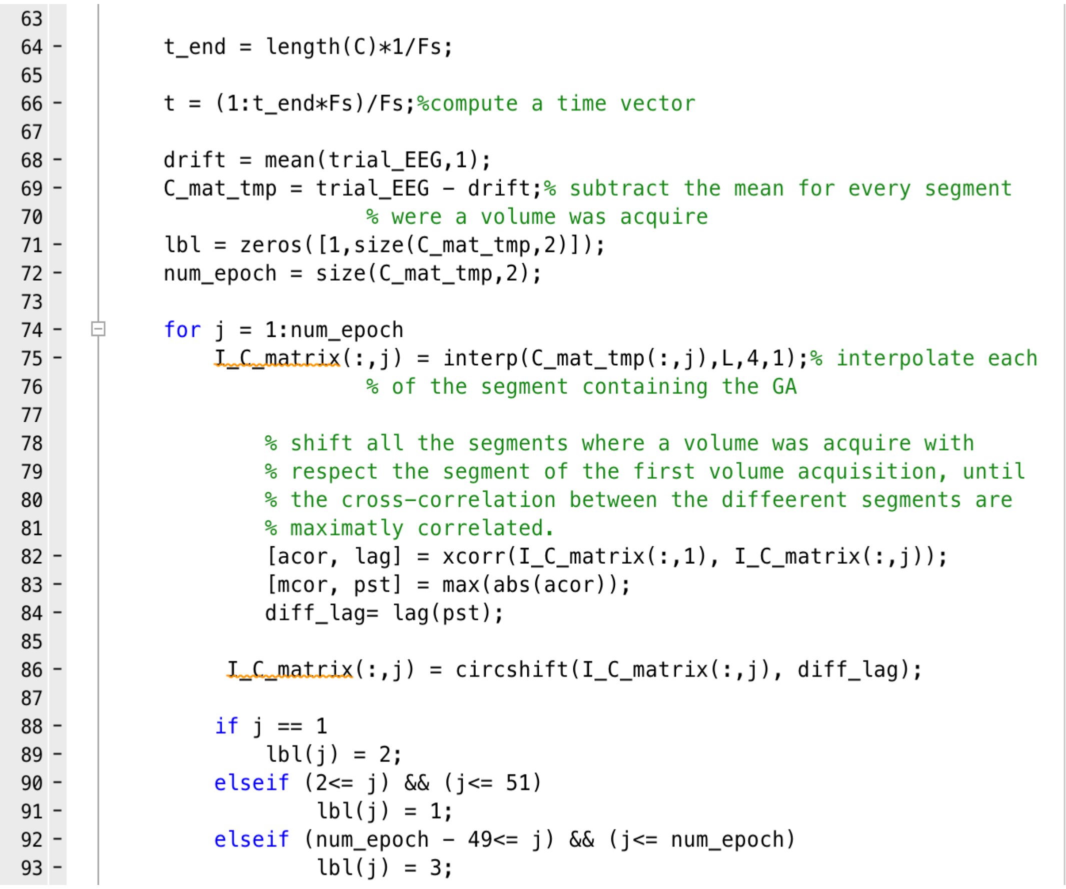


```
        svmStruct = fitcsvm(X,Y,'KernelFunction','linear','OutlierFraction', ...
              0.05,'Standardize',true,'Verbose',1);
        CompactSVMModel = compact(svmStruct);

        [ScoreCSVMModel,ScoreParameters] = fitPosterior(CompactSVMModel,...
     X_test,Y);
        [label{i},postProbs{i}] = predict(ScoreCSVMModel,X_test);
        errRate = sum(Y ~= label{i})/size(X_test,1)  %mis-classification rate
   % display each of the testing set original labels,predicted labels and Posterior Probabilities
         table(Y, label{i},postProbs{i}(:,2),...
    'VariableNames',{'OriginalLabel','PredictedLabel','PosteriorProbability'})
end
```

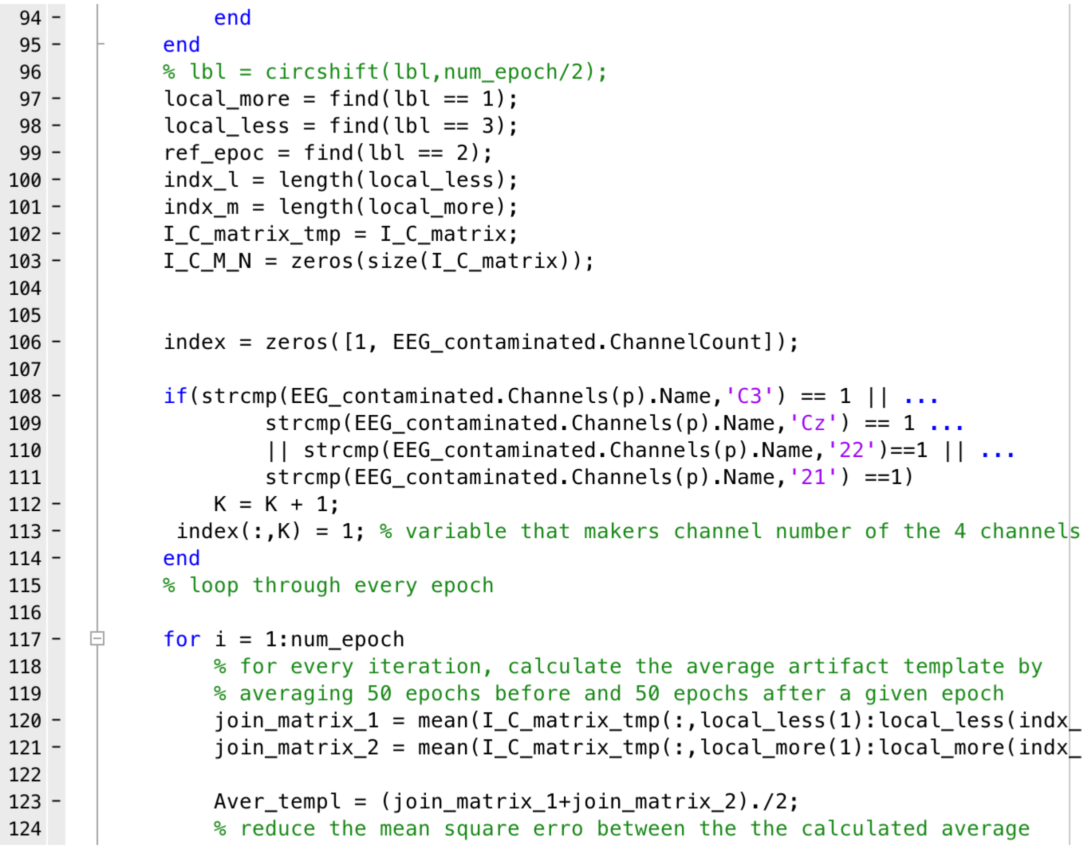
<u>Sources code of section 4.7:</u> The above Matlab script was developed to performed the steps described in section 4.7. The script is an implementation of Support Vector Machine classification using Matlab's fitsvm() function.



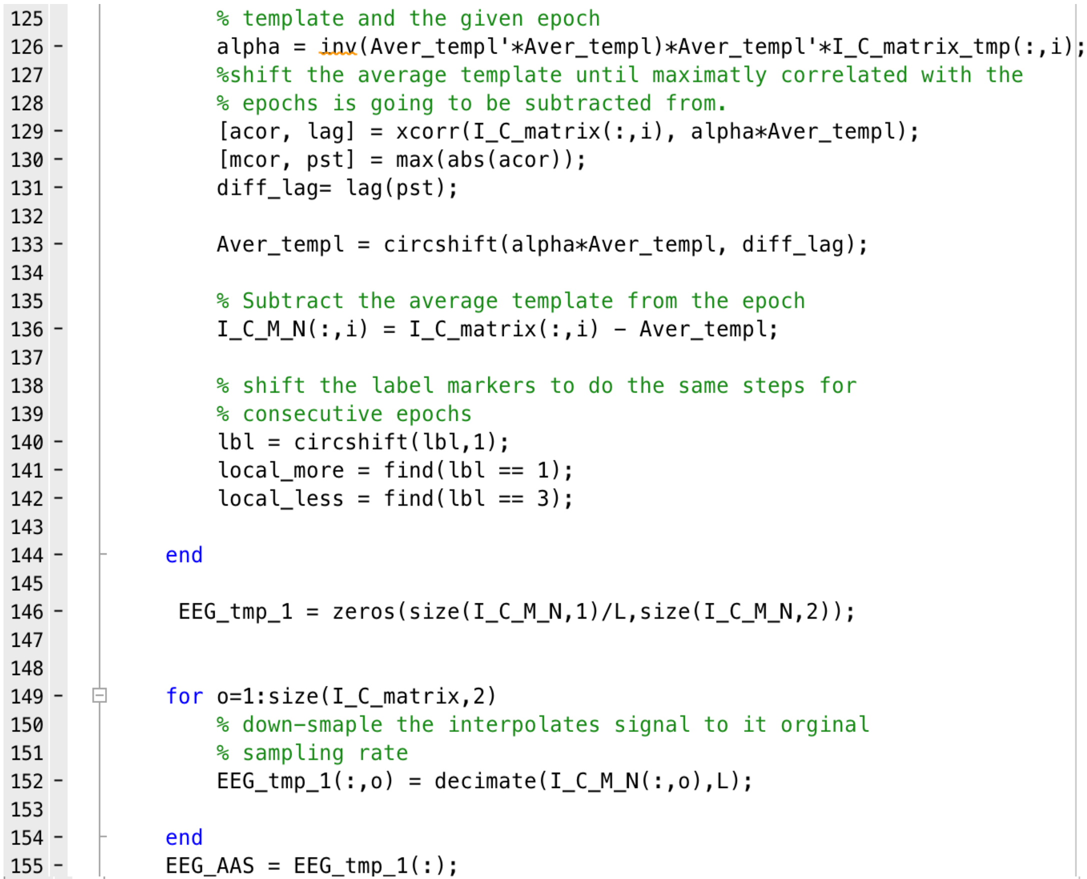
```
% Dalton H Bermudez
% Marshall Foundation (Marshall Plan Scholarship)
% July 15, 2017
%
%
% Objective - to re-implement the methodology of Gradient Artifact
% subtraction technic from:"A Method for Removing Imaging Artifact
% from Continuous EEG Recorded during Functional MRI"
% Philip J. Allen
% NeuroImage 12, 230?239 (2000)
% and
% Reference layer adaptive filtering (RLAF) for EEG artifact
% reduction in simultaneous EEG-fMRI
% David Steyrl, Gunther Krausz, Karl Koschutnig,
% Günter Edlinger and Gernot R Müller-Putz
% J. Neural Eng. 14 (2017) 026003 (20pp)

clc
close all
clear all
% load data from .mat file of simultaneous EEG data
EEG_contaminated = load('eeg_fmri_bi2_000005_Raw_Data.mat');
C = struct2cell(EEG_contaminated);
Marker_Count = EEG_contaminated.MarkerCount;
% read in markers the represent when was a volume acquired (TR)
marker = EEG_contaminated.Markers;

% Find positions in the EEG signal recording where a whole
% FMRI volume was acquired (interval of determined by the
% repetition time)
for i = 1:length(marker)
    if(strcmp(marker(:,i).Description,'T  1_on') == 1)
        Marker_slic_on{i} = marker(:,i).Position;
    elseif(strcmp(marker(:,i).Description, 'T  1_off') == 1)
        Marker_slic_off{i} = marker(:,i).Position;
    end
end

Trig_on = cell2mat(Marker_slic_on);
Trig_off = cell2mat(Marker_slic_off);
Trig = [Trig_on, Trig_off];

%order the trigger marker in ascending order
Trig_order = sort(Trig);
Fs = EEG_contaminated.SampleRate;
nyq = Fs/2;



L = 5;% interpolate at least so the sample frequency is 20kHz
Y = 0;
K = 0;
% loop through each channels in the simultaneous EEG
for p = 1:EEG_contaminated.ChannelCount

    W = 0;
    for g = 1:2:length(Trig_order)-1
            W = W+1;
            % extract intervals in channel where FMRI volumes were acquire
            trial_EEG(:,W) = C{p}(Trig(:,g):Trig(:,g+1),:);
    end

```



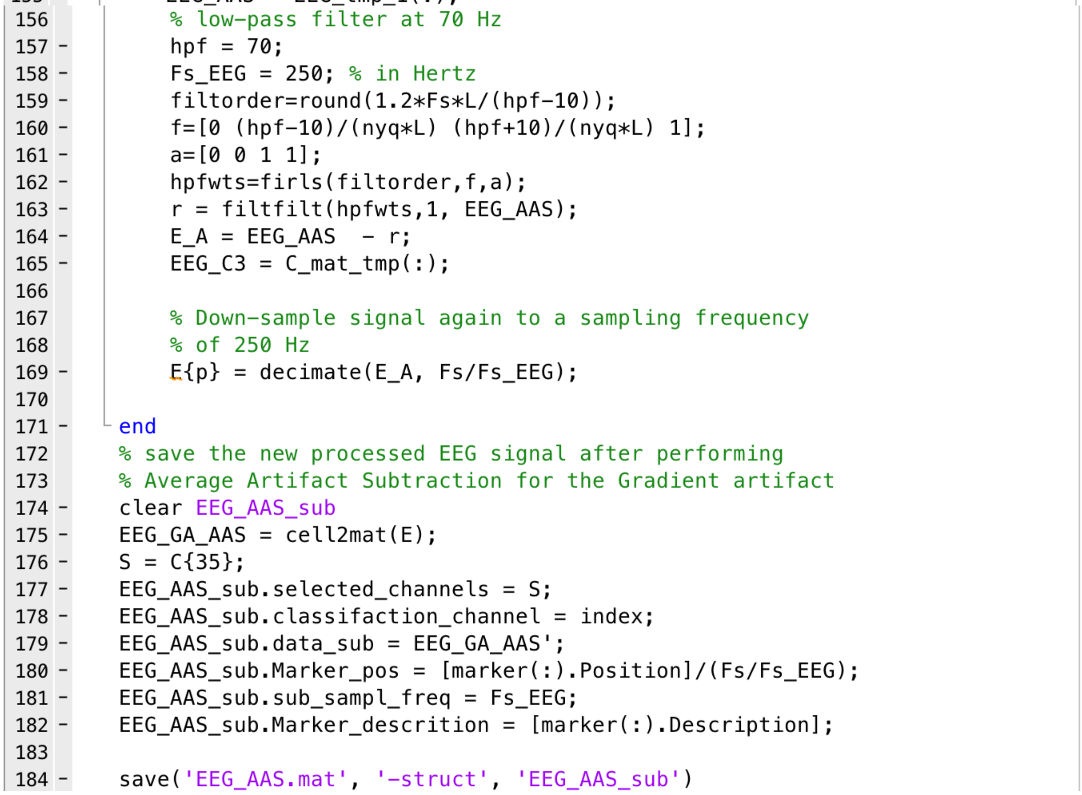
```
    t_end = length(C)*1/Fs;

    t = (1:t_end*Fs)/Fs;%compute a time vector

    drift = mean(trial_EEG,1);
    C_mat_tmp = trial_EEG - drift;% subtract the mean for every segment
                    % were a volume was acquire
    lbl = zeros([1,size(C_mat_tmp,2)]);
    num_epoch = size(C_mat_tmp,2);

    for j = 1:num_epoch
        I_C_matrix(:,j) = interp(C_mat_tmp(:,j),L,4,1);% interpolate each
                    % of the segment containing the GA

            % shift all the segments where a volume was acquire with
            % respect the segment of the first volume acquisition, until
            % the cross-correlation between the diffeerent segments are
            % maximatly correlated.
            [acor, lag] = xcorr(I_C_matrix(:,1), I_C_matrix(:,j));
            [mcor, pst] = max(abs(acor));
            diff_lag= lag(pst);

         I_C_matrix(:,j) = circshift(I_C_matrix(:,j), diff_lag);

        if j == 1
            lbl(j) = 2;
        elseif (2<= j) && (j<= 51)
                lbl(j) = 1;
        elseif (num_epoch - 49<= j) && (j<= num_epoch)
                lbl(j) = 3;
```



```
        end
    end
    % lbl = circshift(lbl,num_epoch/2);
    local_more = find(lbl == 1);
    local_less = find(lbl == 3);
    ref_epoc = find(lbl == 2);
    indx_l = length(local_less);
    indx_m = length(local_more);
    I_C_matrix_tmp = I_C_matrix;
    I_C_M_N = zeros(size(I_C_matrix));


    index = zeros([1, EEG_contaminated.ChannelCount]);

    if(strcmp(EEG_contaminated.Channels(p).Name,'C3') == 1 || ...
            strcmp(EEG_contaminated.Channels(p).Name,'Cz') == 1 ...
            || strcmp(EEG_contaminated.Channels(p).Name,'22')==1 || ...
            strcmp(EEG_contaminated.Channels(p).Name,'21') ==1)
        K = K + 1;
     index(:,K) = 1; % variable that makers channel number of the 4 channels of interest
    end
    % loop through every epoch

    for i = 1:num_epoch
        % for every iteration, calculate the average artifact template by
        % averaging 50 epochs before and 50 epochs after a given epoch
        join_matrix_1 = mean(I_C_matrix_tmp(:,local_less(1):local_less(indx_l)),2);
        join_matrix_2 = mean(I_C_matrix_tmp(:,local_more(1):local_more(indx_m)),2);

        Aver_templ = (join_matrix_1+join_matrix_2)./2;
        % reduce the mean square erro between the the calculated average
```

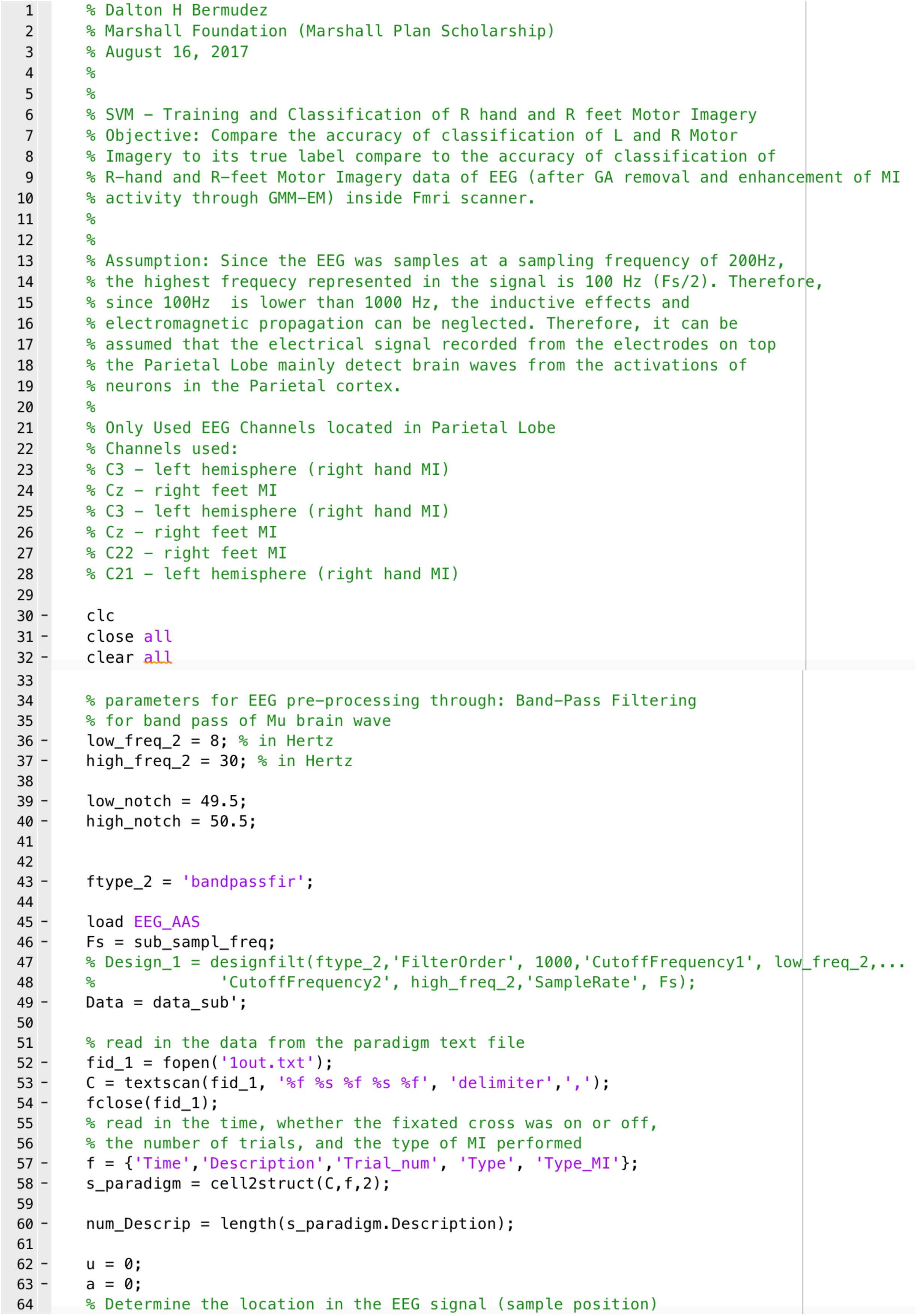


```
        % template and the given epoch
        alpha = inv(Aver_templ'*Aver_templ)*Aver_templ'*I_C_matrix_tmp(:,i);
        %shift the average template until maximatly correlated with the
        % epochs is going to be subtracted from.
        [acor, lag] = xcorr(I_C_matrix(:,i), alpha*Aver_templ);
        [mcor, pst] = max(abs(acor));
        diff_lag= lag(pst);

        Aver_templ = circshift(alpha*Aver_templ, diff_lag);

        % Subtract the average template from the epoch
        I_C_M_N(:,i) = I_C_matrix(:,i) - Aver_templ;

        % shift the label markers to do the same steps for
        % consecutive epochs
        lbl = circshift(lbl,1);
        local_more = find(lbl == 1);
        local_less = find(lbl == 3);

    end

     EEG_tmp_1 = zeros(size(I_C_M_N,1)/L,size(I_C_M_N,2));


    for o=1:size(I_C_matrix,2)
        % down-smaple the interpolates signal to it orginal
        % sampling rate
        EEG_tmp_1(:,o) = decimate(I_C_M_N(:,o),L);

    end
    EEG_AAS = EEG_tmp_1(:);
```

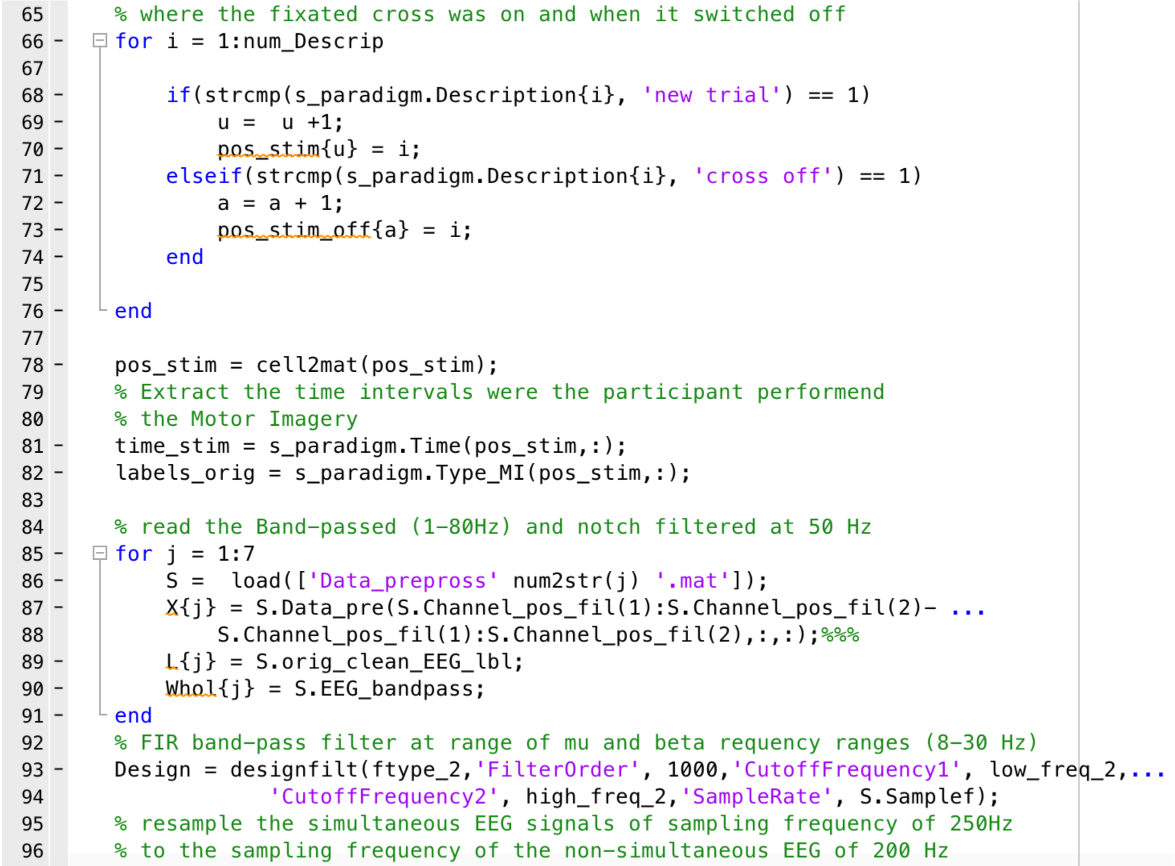


```
    % low-pass filter at 70 Hz
    hpf = 70;
    Fs_EEG = 250; % in Hertz
    filtorder=round(1.2*Fs*L/(hpf-10));
    f=[0 (hpf-10)/(nyq*L) (hpf+10)/(nyq*L) 1];
    a=[0 0 1 1];
    hpfwts=firls(filtorder,f,a);
    r = filtfilt(hpfwts,1, EEG_AAS);
    E_A = EEG_AAS  - r;
    EEG_C3 = C_mat_tmp(:);

    % Down-sample signal again to a sampling frequency
    % of 250 Hz
    E{p} = decimate(E_A, Fs/Fs_EEG);

end
% save the new processed EEG signal after performing
% Average Artifact Subtraction for the Gradient artifact
clear EEG_AAS_sub
EEG_GA_AAS = cell2mat(E);
S = C{35};
EEG_AAS_sub.selected_channels = S;
EEG_AAS_sub.classifaction_channel = index;
EEG_AAS_sub.data_sub = EEG_GA_AAS';
EEG_AAS_sub.Marker_pos = [marker(:).Position]/(Fs/Fs_EEG);
EEG_AAS_sub.sub_sampl_freq = Fs_EEG;
EEG_AAS_sub.Marker_descrition = [marker(:).Description];

save('EEG_AAS.mat', '-struct', 'EEG_AAS_sub')
```

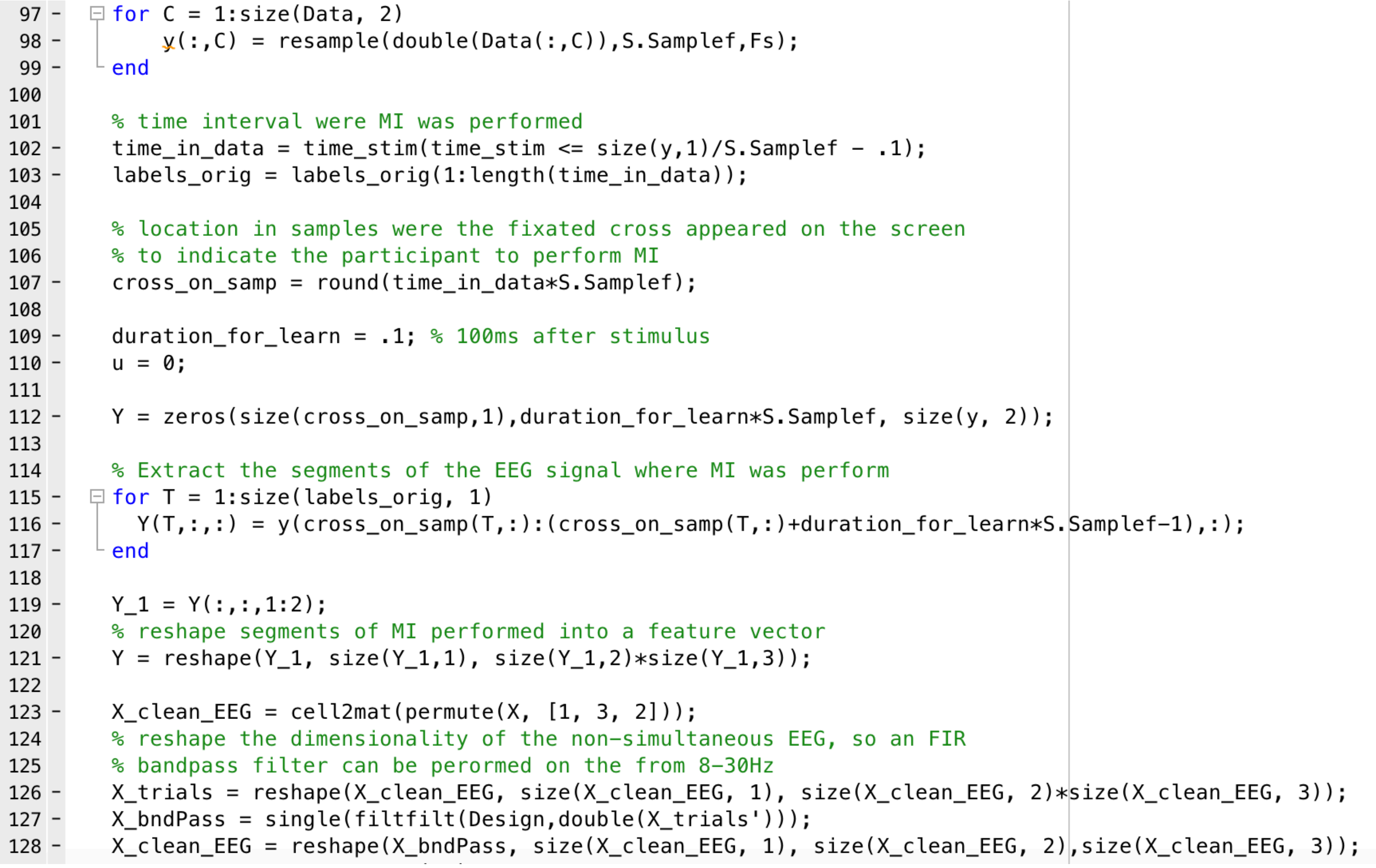
Sources code of section 4.8: The above Matlab script was developed to performed the technic AAS$_{GA}$ described by Allen (2000). The number of GA epochs averaged to calculate the Average Artifact Template was taken from Steyrl (2017).



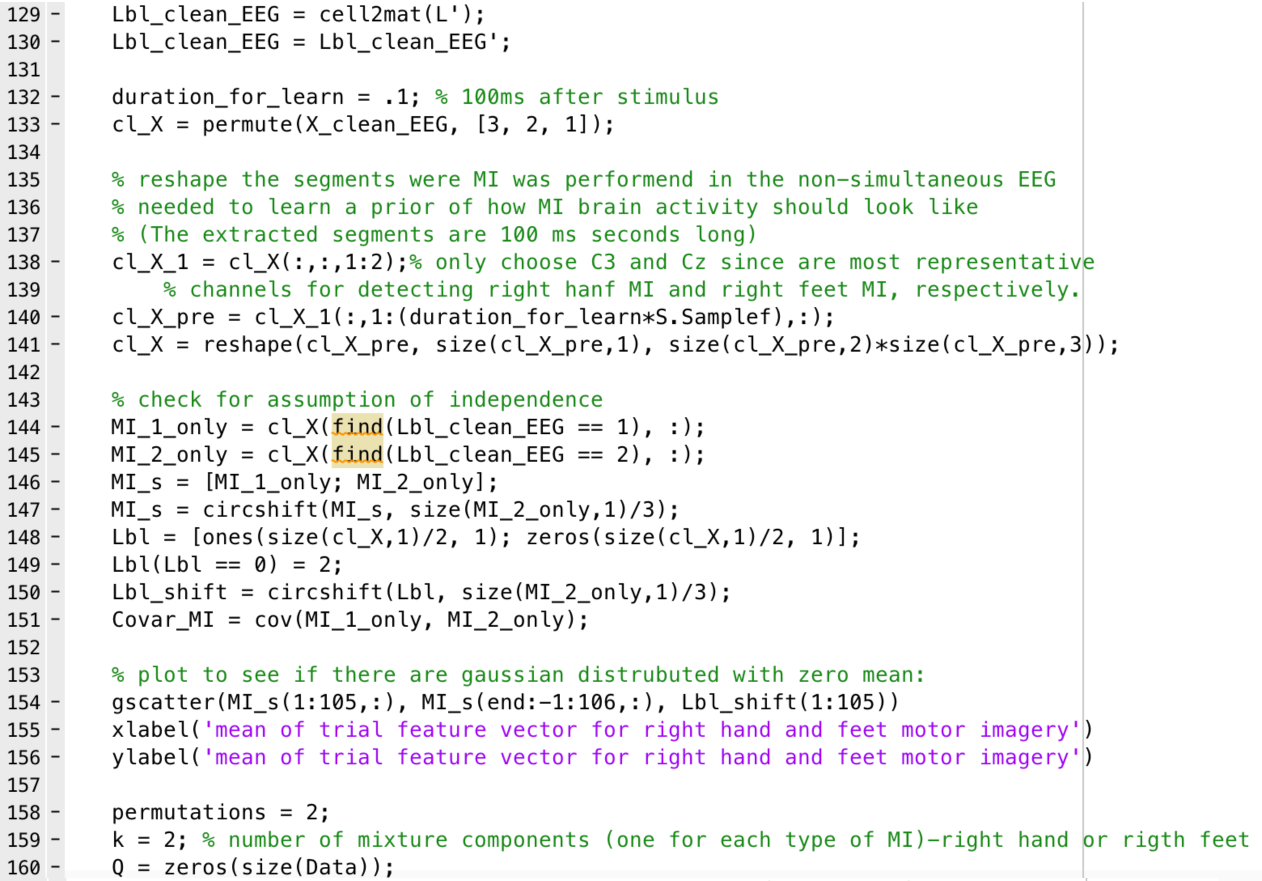
```
% Dalton H Bermudez
% Marshall Foundation (Marshall Plan Scholarship)
% August 16, 2017
%
%
% SVM - Training and Classification of R hand and R feet Motor Imagery
% Objective: Compare the accuracy of classification of L and R Motor
% Imagery to its true label compare to the accuracy of classification of
% R-hand and R-feet Motor Imagery data of EEG (after GA removal and enhancement of MI
% activity through GMM-EM) inside Fmri scanner.
%
%
% Assumption: Since the EEG was samples at a sampling frequency of 200Hz,
% the highest frequecy represented in the signal is 100 Hz (Fs/2). Therefore,
% since 100Hz  is lower than 1000 Hz, the inductive effects and
% electromagnetic propagation can be neglected. Therefore, it can be
% assumed that the electrical signal recorded from the electrodes on top
% the Parietal Lobe mainly detect brain waves from the activations of
% neurons in the Parietal cortex.
%
% Only Used EEG Channels located in Parietal Lobe
% Channels used:
% C3 - left hemisphere (right hand MI)
% Cz - right feet MI
% C3 - left hemisphere (right hand MI)
% Cz - right feet MI
% C22 - right feet MI
% C21 - left hemisphere (right hand MI)

clc
close all
clear all

% parameters for EEG pre-processing through: Band-Pass Filtering
% for band pass of Mu brain wave
low_freq_2 = 8; % in Hertz
high_freq_2 = 30; % in Hertz

low_notch = 49.5;
high_notch = 50.5;


ftype_2 = 'bandpassfir';

load EEG_AAS
Fs = sub_sampl_freq;
% Design_1 = designfilt(ftype_2,'FilterOrder', 1000,'CutoffFrequency1', low_freq_2,...
%             'CutoffFrequency2', high_freq_2,'SampleRate', Fs);
Data = data_sub';

% read in the data from the paradigm text file
fid_1 = fopen('1out.txt');
C = textscan(fid_1, '%f %s %f %s %f', 'delimiter',',');
fclose(fid_1);
% read in the time, whether the fixated cross was on or off,
% the number of trials, and the type of MI performed
f = {'Time','Description','Trial_num', 'Type', 'Type_MI'};
s_paradigm = cell2struct(C,f,2);

num_Descrip = length(s_paradigm.Description);

u = 0;
a = 0;
% Determine the location in the EEG signal (sample position)
```



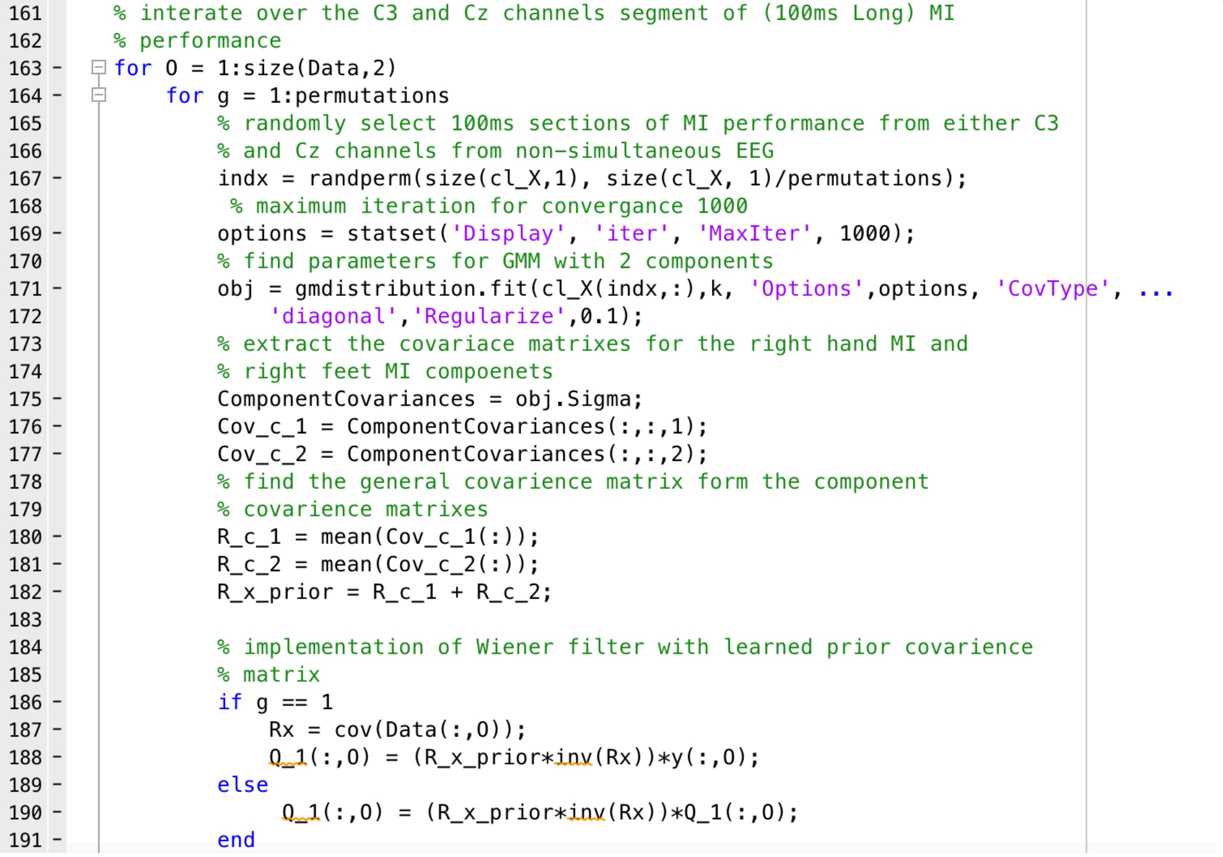
```
% where the fixated cross was on and when it switched off
for i = 1:num_Descrip

    if(strcmp(s_paradigm.Description{i}, 'new trial') == 1)
        u =  u +1;
        pos_stim{u} = i;
    elseif(strcmp(s_paradigm.Description{i}, 'cross off') == 1)
        a = a + 1;
        pos_stim_off{a} = i;
    end

end

pos_stim = cell2mat(pos_stim);
% Extract the time intervals were the participant performend
% the Motor Imagery
time_stim = s_paradigm.Time(pos_stim,:);
labels_orig = s_paradigm.Type_MI(pos_stim,:);

% read the Band-passed (1-80Hz) and notch filtered at 50 Hz
for j = 1:7
    S =  load(['Data_prepross' num2str(j) '.mat']);
    X{j} = S.Data_pre(S.Channel_pos_fil(1):S.Channel_pos_fil(2)- ...
        S.Channel_pos_fil(1):S.Channel_pos_fil(2),:,:);%%%
    L{j} = S.orig_clean_EEG_lbl;
    Whol{j} = S.EEG_bandpass;
end
% FIR band-pass filter at range of mu and beta requency ranges (8-30 Hz)
Design = designfilt(ftype_2,'FilterOrder', 1000,'CutoffFrequency1', low_freq_2,...
            'CutoffFrequency2', high_freq_2,'SampleRate', S.Samplef);
% resample the simultaneous EEG signals of sampling frequency of 250Hz
% to the sampling frequency of the non-simultaneous EEG of 200 Hz
```



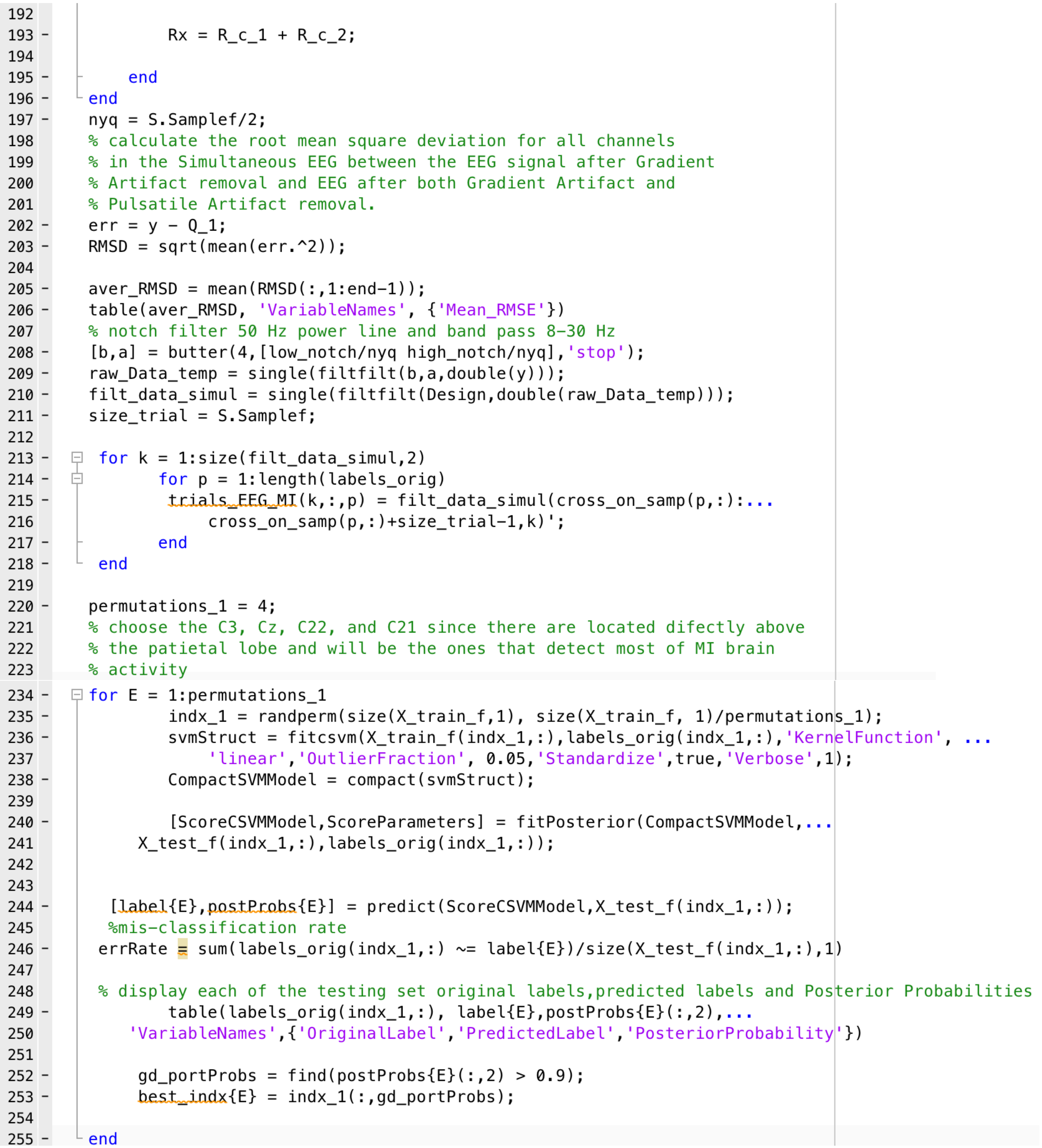
```
for C = 1:size(Data, 2)
    y(:,C) = resample(double(Data(:,C)),S.Samplef,Fs);
end

% time interval were MI was performed
time_in_data = time_stim(time_stim <= size(y,1)/S.Samplef - .1);
labels_orig = labels_orig(1:length(time_in_data));

% location in samples were the fixated cross appeared on the screen
% to indicate the participant to perform MI
cross_on_samp = round(time_in_data*S.Samplef);

duration_for_learn = .1; % 100ms after stimulus
u = 0;

Y = zeros(size(cross_on_samp,1),duration_for_learn*S.Samplef, size(y, 2));

% Extract the segments of the EEG signal where MI was perform
for T = 1:size(labels_orig, 1)
  Y(T,:,:) = y(cross_on_samp(T,:):(cross_on_samp(T,:)+duration_for_learn*S.Samplef-1),:);
end

Y_1 = Y(:,:,1:2);
% reshape segments of MI performed into a feature vector
Y = reshape(Y_1, size(Y_1,1), size(Y_1,2)*size(Y_1,3));

X_clean_EEG = cell2mat(permute(X, [1, 3, 2]));
% reshape the dimensionality of the non-simultaneous EEG, so an FIR
% bandpass filter can be perormed on the from 8-30Hz
X_trials = reshape(X_clean_EEG, size(X_clean_EEG, 1), size(X_clean_EEG, 2)*size(X_clean_EEG, 3));
X_bndPass = single(filtfilt(Design,double(X_trials')));
X_clean_EEG = reshape(X_bndPass, size(X_clean_EEG, 1), size(X_clean_EEG, 2),size(X_clean_EEG, 3));
```



```
    Lbl_clean_EEG = cell2mat(L');
    Lbl_clean_EEG = Lbl_clean_EEG';

    duration_for_learn = .1; % 100ms after stimulus
    cl_X = permute(X_clean_EEG, [3, 2, 1]);

    % reshape the segments were MI was performend in the non-simultaneous EEG
    % needed to learn a prior of how MI brain activity should look like
    % (The extracted segments are 100 ms seconds long)
    cl_X_1 = cl_X(:,:,1:2);% only choose C3 and Cz since are most representative
        % channels for detecting right hanf MI and right feet MI, respectively.
    cl_X_pre = cl_X_1(:,1:(duration_for_learn*S.Samplef),:);
    cl_X = reshape(cl_X_pre, size(cl_X_pre,1), size(cl_X_pre,2)*size(cl_X_pre,3));

    % check for assumption of independence
    MI_1_only = cl_X(find(Lbl_clean_EEG == 1), :);
    MI_2_only = cl_X(find(Lbl_clean_EEG == 2), :);
    MI_s = [MI_1_only; MI_2_only];
    MI_s = circshift(MI_s, size(MI_2_only,1)/3);
    Lbl = [ones(size(cl_X,1)/2, 1); zeros(size(cl_X,1)/2, 1)];
    Lbl(Lbl == 0) = 2;
    Lbl_shift = circshift(Lbl, size(MI_2_only,1)/3);
    Covar_MI = cov(MI_1_only, MI_2_only);

    % plot to see if there are gaussian distrubuted with zero mean:
    gscatter(MI_s(1:105,:), MI_s(end:-1:106,:), Lbl_shift(1:105))
    xlabel('mean of trial feature vector for right hand and feet motor imagery')
    ylabel('mean of trial feature vector for right hand and feet motor imagery')

    permutations = 2;
    k = 2; % number of mixture components (one for each type of MI)-right hand or rigth feet
    Q = zeros(size(Data));
```



```
    % interate over the C3 and Cz channels segment of (100ms Long) MI
    % performance
    for O = 1:size(Data,2)
        for g = 1:permutations
            % randomly select 100ms sections of MI performance from either C3
            % and Cz channels from non-simultaneous EEG
            indx = randperm(size(cl_X,1), size(cl_X, 1)/permutations);
             % maximum iteration for convergance 1000
            options = statset('Display', 'iter', 'MaxIter', 1000);
            % find parameters for GMM with 2 components
            obj = gmdistribution.fit(cl_X(indx,:),k, 'Options',options, 'CovType', ...
                'diagonal','Regularize',0.1);
            % extract the covariace matrices for the right hand MI and
            % right feet MI compoenets
            ComponentCovariances = obj.Sigma;
            Cov_c_1 = ComponentCovariances(:,:,1);
            Cov_c_2 = ComponentCovariances(:,:,2);
            % find the general covarience matrix form the component
            % covarience matrixes
            R_c_1 = mean(Cov_c_1(:));
            R_c_2 = mean(Cov_c_2(:));
            R_x_prior = R_c_1 + R_c_2;

            % implementation of Wiener filter with learned prior covarience
            % matrix
            if g == 1
                Rx = cov(Data(:,O));
                Q_1(:,O) = (R_x_prior*inv(Rx))*y(:,O);
            else
                 Q_1(:,O) = (R_x_prior*inv(Rx))*Q_1(:,O);
            end
```



```
        Rx = R_c_1 + R_c_2;

    end
end
nyq = S.Samplef/2;
% calculate the root mean square deviation for all channels
% in the Simultaneous EEG between the EEG signal after Gradient
% Artifact removal and EEG after both Gradient Artifact and
% Pulsatile Artifact removal.
err = y - Q_1;
RMSD = sqrt(mean(err.^2));

aver_RMSD = mean(RMSD(:,1:end-1));
table(aver_RMSD, 'VariableNames', {'Mean_RMSE'})
% notch filter 50 Hz power line and band pass 8-30 Hz
[b,a] = butter(4,[low_notch/nyq high_notch/nyq],'stop');
raw_Data_temp = single(filtfilt(b,a,double(y)));
filt_data_simul = single(filtfilt(Design,double(raw_Data_temp)));
size_trial = S.Samplef;

 for k = 1:size(filt_data_simul,2)
       for p = 1:length(labels_orig)
        trials_EEG_MI(k,:,p) = filt_data_simul(cross_on_samp(p,:):...
            cross_on_samp(p,:)+size_trial-1,k)';
       end
 end

permutations_1 = 4;
% choose the C3, Cz, C22, and C21 since there are located difectly above
% the patietal lobe and will be the ones that detect most of MI brain
% activity
for E = 1:permutations_1
        indx_1 = randperm(size(X_train_f,1), size(X_train_f, 1)/permutations_1);
        svmStruct = fitcsvm(X_train_f(indx_1,:),labels_orig(indx_1,:),'KernelFunction', ...
            'linear','OutlierFraction', 0.05,'Standardize',true,'Verbose',1);
        CompactSVMModel = compact(svmStruct);

        [ScoreCSVMModel,ScoreParameters] = fitPosterior(CompactSVMModel,...
     X_test_f(indx_1,:),labels_orig(indx_1,:));


  [label{E},postProbs{E}] = predict(ScoreCSVMModel,X_test_f(indx_1,:));
  %mis-classification rate
 errRate = sum(labels_orig(indx_1,:) ~= label{E})/size(X_test_f(indx_1,:),1)

 % display each of the testing set original labels,predicted labels and Posterior Probabilities
        table(labels_orig(indx_1,:), label{E},postProbs{E}(:,2),...
    'VariableNames',{'OriginalLabel','PredictedLabel','PosteriorProbability'})

     gd_portProbs = find(postProbs{E}(:,2) > 0.9);
     best_indx{E} = indx_1(:,gd_portProbs);

end
```

<u>**Sources code of section 3.14, 3.15, and 4.8:**</u> The above Matlab script first implements the GMM-EM $_{\text{Wiener Filter}}$ as described by sections 3.14 and 3.15. The Matlab function used for GMM-EM with

two components was gmdistribution.fit(). Then, the steps described in 4.8 for the classification of MI activity were perform on the processed simultaneous EEG data.