A Simplified GUI for EEG Acquisition and Signal Pre-Processing

By

Woon Tian Lok

16067

Dissertation submitted in partial fulfillment of

the requirements of

Bachelor of Engineering (Hons)

(Electrical and Electronic)

JANUARY 2016

Universiti Teknologi PETRONAS
32610 Seri Iskandar,
Perak Darul Ridzuan,
Malaysia.

CERTIFICATION OF APPROVAL

A Simplified GUI for EEG Acquisition and Signal Pre-Processing

By

Woon Tian Lok
16067

A project dissertation submitted to the
Electrical and Electronic Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL AND ELECTRONIC)

Approved by,

_____
(Dr. Syed Saad Azhar Ali)

UNIVERSITI TEKNOLOGI PETRONAS
Seri Iskandar, PERAK
January 2016

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgments, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

_____

WOON TIAN LOK

ABSTRACT

Brain-computer interface (BCI) is a direct communication pathway between a human and an external device. However, for highly motor-disabled people, it is the only means of communication to the outside world. By using it to provide people a way to command some devices, which could expand the abilities of a person. Thus, giving highly disabled people back some of their autonomy. The main objective of this project is to design a Graphic User Interface (GUI) to acquire and pre-processing the raw Electroencephalogram (EEG) signal in MATLAB for further brain analysis. As a result, a MATLAB GUI has been implemented that can get raw data from the EEG headset to perform pre-processing of the signal. Also, the GUI could be further tested, improved and considerations are given to enhance the development of EEG signal processing.

ACKNOWLEDGEMENT

First and foremost, I would like to Universiti Teknologi PETRONAS for providing me an opportunity to pursue Final Year Project(FYP) as a partial fulfillment of the requirement for me to graduate from engineering course. Special thank my project supervisor, Dr. Syed Saad Azhar Ali for taking the effort to shared his valuable knowledge with me and guided me patiently, without him, I would not able to finish my FYP on time. Last but not least, I would like to express my appreciation and gratitude to all those who have helped me to complete my dissertation report.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLE

CHAPTER 1

INTRODUCTION

1.1 Background

In recent times, development of Brain-Computer Interface (BCI) in areas corresponding to medical purposes and in research has become more widespread by exploration of the physiological signals like Electroencephalogram(EEG) [1].

A variety of signal processing techniques has applied for online EEG signal processing, such as Matlab, LabVIEW, and Python. These methods required external tools to allow routine processing of EEG data, and also, make possible the analysis of multimodal data collected using more complex experimental designs than previous analysis methods allowed[2].

The existing EEG setup at CISIR based on the dedicated software provided by the manufacturer. Hence, a proposed interface is required to acquire data from EEG. This interface proposed interface will serve as a platform to perform online EEG signal processing. It is vital to have the EEG data available for online processing in MATLAB environment to provide a reliable algorithm resources for further analysis[3]

Emotiv EPOC has a total of 14 EEG sensors, and two references nodes, due to the low-cost, higher bit rate, better resolution and user-friendly of Emotiv EPOC headset, it has become one of the few highly affordable BCI equipment available today as a basis to design a brain [1, 4].

1.2 Problem Statement

The current BCI software in Universiti Teknologi PETRONAS is provided by the manufacturer of Emotiv EPOC, which has limited development of EEG signal processing that can be done. The education edition of the Emotiv EPOC software only offer the user to get raw EEG data using SDK, and conduct research such as detection of emotion, measures and displays a wide range of subjective emotional responses using their application program interface (API) [5].

Hence, a new BCI software is required to conduct a further study on the brain signal and to perform signal processing. Additional studies of BCI are very important for a medical application such as become a method of diagnosis for tumors, stroke, and other focal brain disorders.

In-depth knowledge such as understanding the meaning of brain wave and their categorization are required to develop a new BCI. Hence, the aim of this project is to focus on how to acquire and pre-processing raw EEG signals online.

1.3 Objectives and Scope of Study

The main objectives of this project are to design an interface capable of acquiring EEG signals from existing EEG hardware and perform pre-processing analysis on the raw EEG signals. Understanding of the architecture of BCI system is very vital to develop a GUI. The apprehend meaning of brain wave and how to develop a BCI application using Emotiv EEG headset is needed.[6]

Along with this, the GUI for raw EEG signal pre-processing will be done by some MATLAB coding and the MATLAB environment will be tested and evaluated through experiment.

CHAPTER 2

LITERATURE REVIEW

2.1 Human Brain Structure

The fundamental of an Electroencephalogram (EEG) technology is to study the human brain working principle. The human brain can be separate into right and left hemispheres. Figure 2.1.1 shows the brain anatomy of a human brain, and there are four different lobes, which are Frontal, Temporal, Parietal and Occipital [7], each of the lobes has different functions.

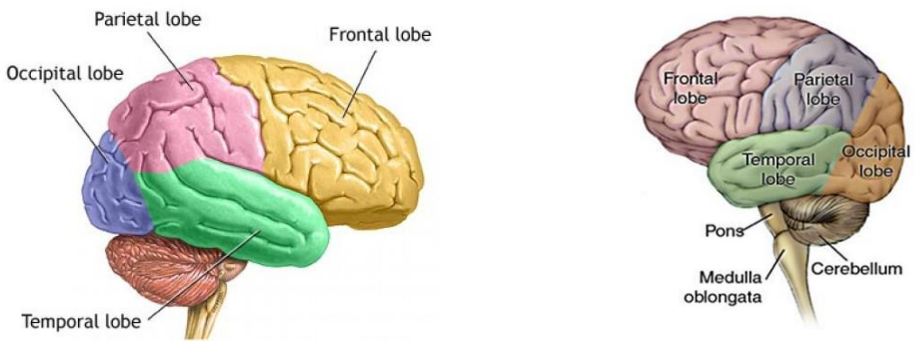


Figure 2.1.1: Brain Anatomy

The human brain stem structures consist of the pons, medulla oblongata, and cerebellum. The control of vital processes, such as respiration and heart rate is regulated by the medulla oblongata and pons. On the order hand, the cerebellum is a large muscle coordination controls balancing of our body movement while walking or writing[8].

Figure 2.1.2 shows the structure of a single neuron. The human brain consist of a huge number of complex neurons network, each of them is interconnected together and sending out an electrical impulse to communicate with each other. The feedback field potentials produce by the neurons can be captured and measured by using EEG electrodes.[7]

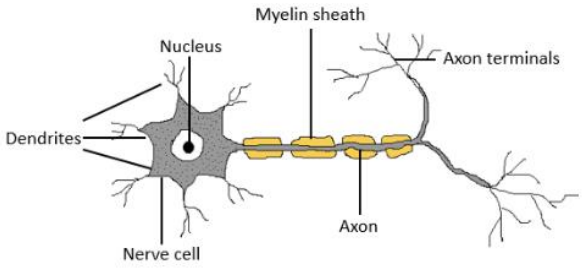


Figure 2.1.2 : Structure of a single neuron

2.2 EEG (Electroencephalogram)

The EEG measures brain waves of different frequencies within the brain. Different tasks will trigger different frequencies which the frequencies are extracted from the EEG [9]. The raw EEG has usually been described in terms of frequency bands delta < 4Hz (deep sleep stage), theta 4–7 Hz (sleep stage), alpha 8–12 Hz (quiet waking), beta 12–30 Hz (activated cortex), and gamma 30–60Hz ("cognitive" frequency band) [10] as shown in Figure 2.2.1, 2.2.2, 2.2.3, 2.2.4 and 2.2.5. The electrical potentials generated by activity in the brain is very weak, however, straight placed of electrodes on the scalp can help to the degree those weak signals.

1. Delta waves (< 4Hz) – It is a very diffuse slow wave, generally seen in deep sleep.

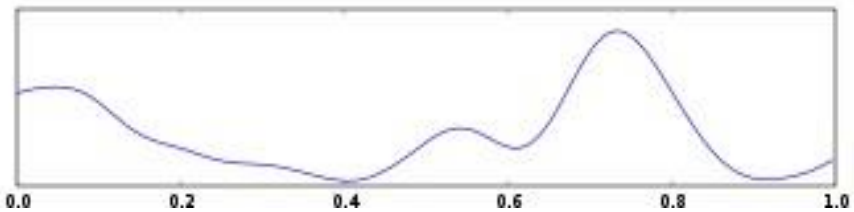


Figure 2.2.1 : Delta wave

2. Theta waves (4–7 Hz) – It normally seen in young children and also be seen in meditation.

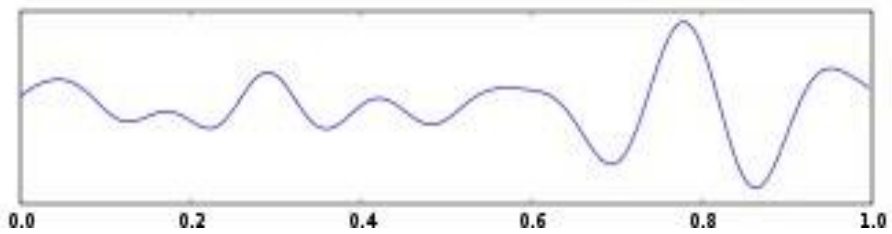


Figure 2.2.2 : Theta wave

3. Alpha waves (8-12 Hz) – It is the resting rhythm of the visual system.

.

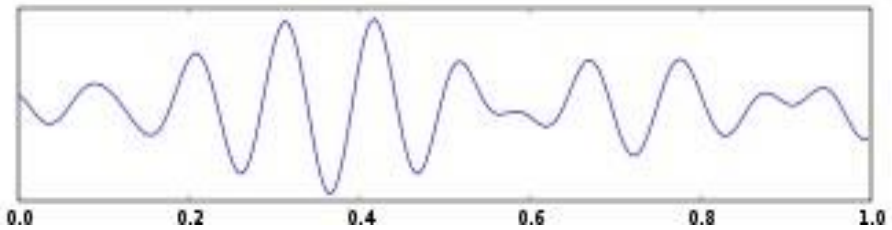


Figure 2.2.3 : Alpha wave

4. Beta waves (12–30 Hz)
   - Low Beta Band (12-15Hz)
     - Do math's problems or reading in this state.
   - Beta Band (15-20 Hz)
     - Memory recalls when reading.
   - High Beta Band (20-30 Hz)
     - Muscular activity (EMG) can sneak in.
     - Intensive thinking, worrying about things.

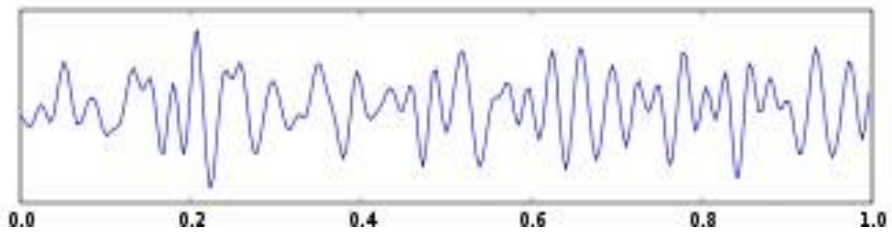


Figure 2.2.4 : Beta wave

5. Gamma wave (30-100Hz) – Indicates the high level of sensory & perceptual binding.

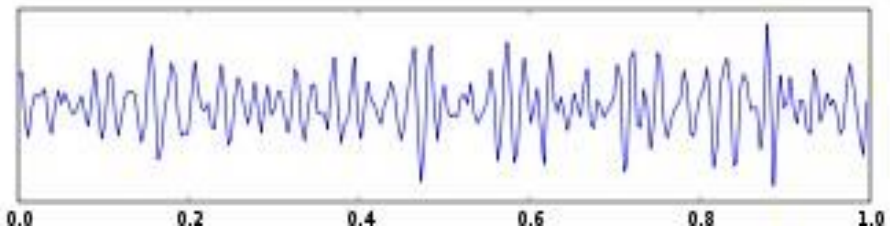


Figure 2.2.5 : Gamma wave

Initially, EEG technology is mostly used in a therapeutic application for severe motor incapacities treatment. Presently, more EEG acquisition unit is coming out to the market trend, with the wearable wireless headset ability and convenient to the user which makes the EEG technology to applied in many other expenses such as control system and entertainment rather than medical application.[11]

2.3 Brain-Computer Interfaces (BCI)

The primary focused of BCI is mainly because of growing number of individuals with severe motor disabilities. The researchers hope that the BCI system would help to recover their life in the meantime to reduce the cost and time for intensive care[8].

For this project, non-invasive BCI acquisition EEG headset is used as it is low cost and the safest compares to invasive and partially invasive acquisition techniques[8]. The invasive technique has been used in BCI acquisition tool to direct insert on the brain through critical surgery. On the other hand, partially invasive technique is used to insert the BCI devices onto the top of human brain skull.[8]

2.3.1 Architecture of a BCI system

The ideas of Brain-Computer Interface (BCI) is to connect the of brain waves with an output device through some interface. From the figure 2.3.1.1, its shows the three major physical components in a BCI system which are[7]:

1) Signal Acquisition
    a) Transforms of electrode signal into numerical values that can be manipulated and read by a PC.

2) Signal Processing:
    a) Analysis and organize of EEG data.
        i) Data Pre-processing
        ii) Feature Extraction
        iii) Data Classification

3) Device Receiver: Respond to the command from the Signal Processing.

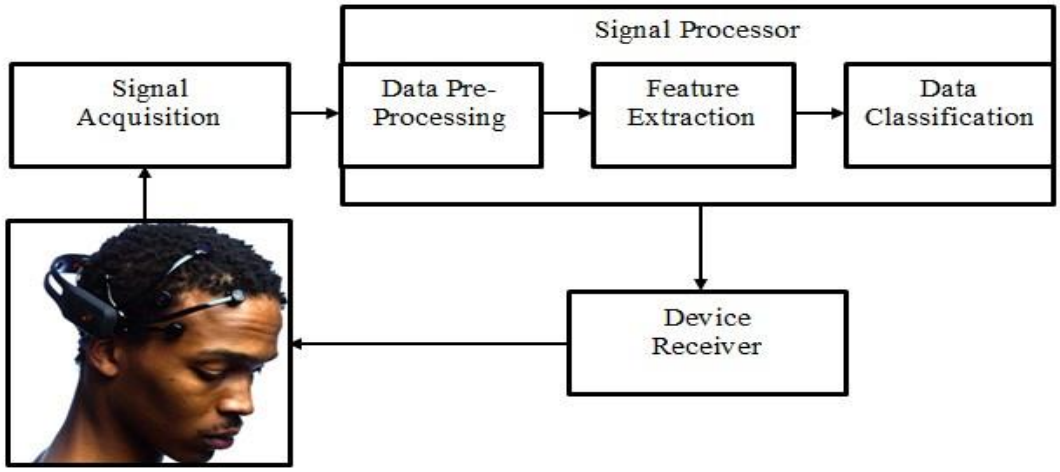


Figure 2.3.1.1: BCI system diagram

2.4 Emotiv EPOC EEG headset (Signal Acquisition Tool)

Figure 2.4.1 : Emotiv EPOC EEG headset

Figure 2.4.1 is an Emotiv EPOC EEG headset that had been chosen as a signal acquisition hardware used by this project. From the comparison of the current EEG acquisition hardware, Emotiv EPOC EEG headset possesses higher bit rate and have better resolution compare to other acquisition tools. [12] The abbreviation of each EEG sensor can be represented by the Table 1.

Table 1: EEG sensor abbreviation [13]

AF3	Attention	FC6	Left Body controller
AF4	Judgment	T7	Verbal memory
F3	Motor planning	T8	Emotional memory
F4	Motor planning for left upper	P7	Verbal understanding
F7	Verbal Expression	P8	Understanding, Motivation
F8	Anger, Happy	O1	Visual processing
FC5	Right Body controller	O2	Visual processing

2.5 Data Pre-processing

Removing of DC Offset

EEG data is stored as fluctuating point values which is directly transformed from the unspecified 14-bit ADC output from the Emotiv EPOC headset. The DC level of the raw signal occurs at roughly 4000 uV, the negative voltages are conveyed as +ve values less than the average level, and positive voltages are conveyed as +ve values greater than the average. It is necessary to apply DC offset removal before performing the analysis. One of the methods was to remove the mean value from the EEG data.[14]

Extract of Specific EEG frequency

Figure 2.5.1 shows the frequency response of the Butterworth Filter. This filter is selected to extract the EEG frequency because of its maximally flat characteristic in the passband. In addition, no ripple is present in the stopband. On the other hands, the passbands and stopbands are maximally flat, thus resulting in a quality output signal for the different frequency bands.

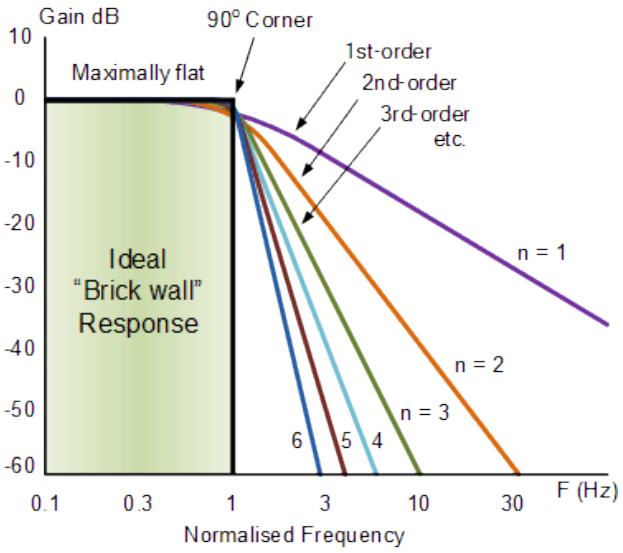


Figure 2.5.1: Ideal Frequency Response for a Butterworth Filter

2.6 Simple EEG Classification

Detecting Close eye-brain wave pattern

The EEG sensor at nodes O1 and O2 are used for visual processing[15]. During close eye period, a different pattern of brain wave will have received by sensor O1 and O2. That wave pattern can be classified as close eye movement as shows in figure 2.6.1.

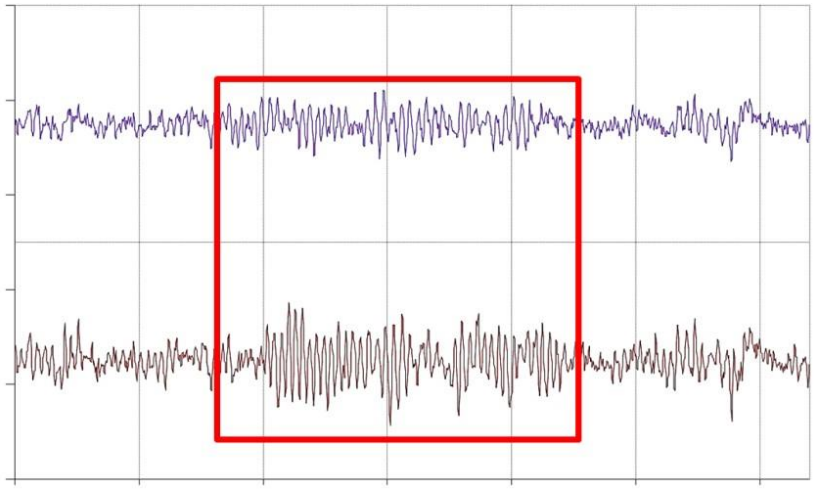


Figure 2.6.1 : Brainwave during close eye at O1 and O2 sensor

Figure 2.6.2 shows the duration of blink eye stage when sensor AF3, and AF4 received a different pattern of brain wave that can classify as blinking eye [15].

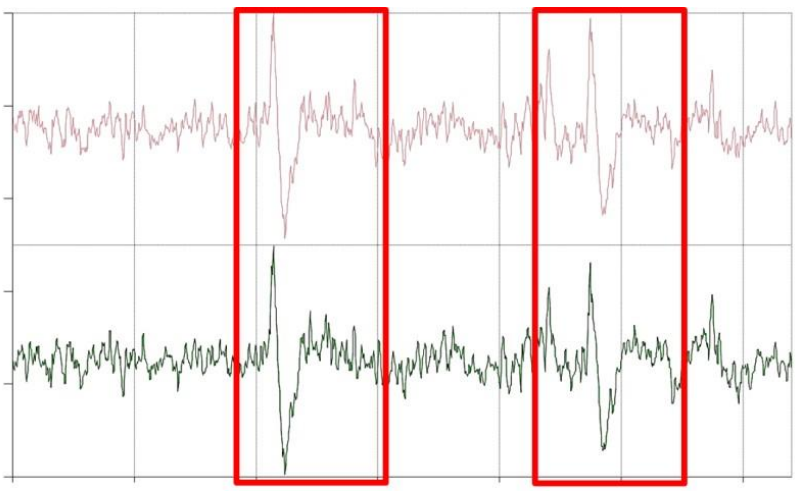


Figure 2.6.2: Brainwave during blink at AF3 and AF4 sensor

Table 2: Types of EEG Online Signal Processing Software

Software	Strengths	Weakness
MATLAB	➢ Easy to function ➢ Many Toolbox provided ➢ Easy to get ➢ Implementation very fast ➢ plotting function	➢ Expensive ➢ Need update maintenance cost
OCTAVE	➢ Use C language ➢ Free	➢ Implementation very slow
PYTHON	➢ Interpreted language like MATLAB ➢ Support advanced features ➢ Free	➢ Less easy as in MATLAB ➢ Not a popular language in the industry
C/C++	➢ High level of computational speed and optimization ➢ Use Four distinct languages	➢ C++ is extremely huge ➢ Various features interact in horribly complex ways

2.7 Selection of EEG Online Signal Processing Software

From Table 2, among all the programming software, MATLAB is the best choice for designing a Brain Computer Interface (BCI) environment for online Electroencephalogram (EEG) signal processing because of its powerful built-in features. Compare with other software like C++ and Octave, all of them need more complex programming code. Although the price for MATLAB itself is not low, however, MATLAB can easily to be accessed in Universiti Teknologi PETRONAS.

Hence, MATLAB has been chosen as my primary software to design a GUI that can acquire EEG data online for signal processing. In addition, the portable Emotiv EPOC EEG Headset will be used as an EEG Signal Acquisition tool for transfering the raw EEG data to MATLAB for online signal processing.

Table 3: Available MATLAB Toolbox for EEG Signal Processing

TOOLBOX	FEATURES	WEAKNESSES
FieldTrip	-Online and offline EEG analysis -Spectral and connectivity analysis	-Does not have (GUI).
BCILab	-Online and offline EEG analysis -Non-linear classification spectrum analysis	-Toolbox of EEGLab -Leverage utility from existing software
EEGLab	-Offline analysis only	-Can’t access online EEG signal data
BioSig	-Accepting and generating signals in multiple data formats	- Don’t have a real-time BCI system implemented in MATLAB and Simulink

2.8 Critical Analysis

Once the implemented software has been finalized, a review for different existing EEG signal processing toolbox has been started before designing a MATLAB interface. MATLAB offers some open valuable tools like EEGLab and BCILab for both online and offline signal analysis for BCI approaches[16].

Table 3 showed the comparison of the features of each toolbox and listed out their weaknesses. From all of the toolbox, BCILab is the most powerful and have many features, but it has a lesser focus on real-time acquisition application, but integrate with existing software like BCI2000 and ERICA. Besides that, FieldTrip is also a good toolbox. Unfortunately, it doesn't have GUI interface which means that the user needs to know basic MATLAB function in order to run the software.

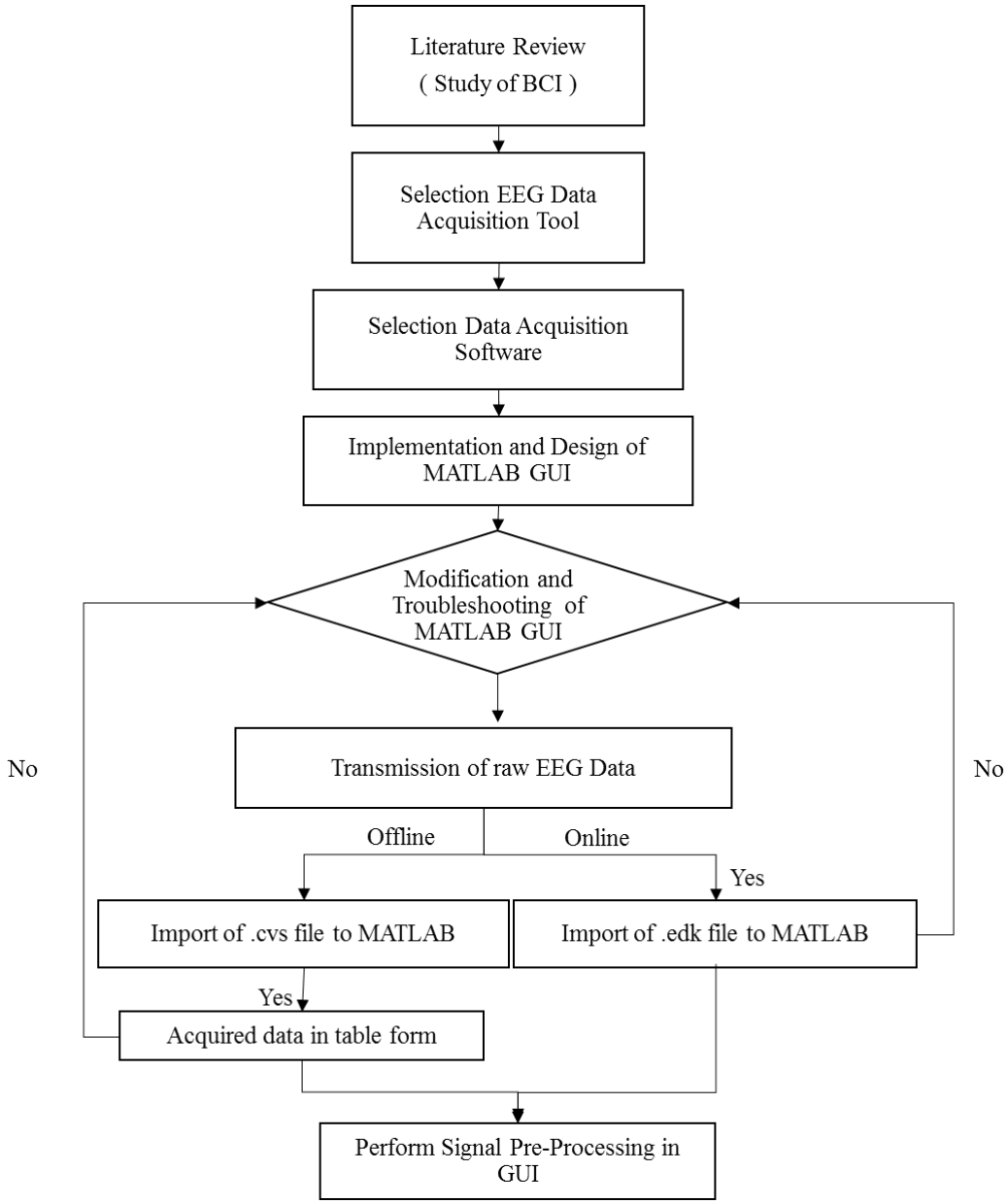
The new designation of MATLAB GUI must be able to get raw data from 14-channel electrode sensors of Emotiv EPOC EEG headset. Besides that, the BCI environment needs to have a filter. This help the user to extract the specific EEG frequency that is needed. As the consequence, this BCI interface will become the foundation for research in future.

CHAPTER 3

METHODOLOGY

3.1 Design of MATLAB Interface Process

The flow of the project methodology is shows in the figure below.



3.2 Development of BCI on PC with Emotiv EEG headset

` Emotiv Education Edition SDK is a suite of development tool to help the developer to create BCI application on PC. It includes the following:

- Emotiv Control Panel: Program for setting and testing the headset.
- Testbench: Program for load and save the experiment data.
- EmoEngine(edk.dll): Library files that needed for processing EEG data.

3.3 Online BCI Design groundwork

❖ The Emotiv SDK library files were imported into the same folder, afterward, make sure the Emotiv Control Panel was opened and connected to Emotiv headset so that the library files below could read by the MATLAB.

1. ***edk.dll***
2. ***edk.h***
3. ***edk_utils.dll***
4. ***edkErrorCode.h***
5. ***EmoStateDLL.h***

❖ Type mex -setup in the command window as shows in figure 3.3.1. (Noted: Visual Studio 2013 Pro was installed in MATLAB).

```
>> mex -setup
MEX configured to use 'Microsoft Visual C++ 2013 Professional (C)' for C language compilation.
Warning: The MATLAB C and Fortran API has changed to support MATLAB
     variables with more than 2^32-1 elements. In the near future
     you will be required to update your code to utilize the
     new API. You can find more information about this at:
     http://www.mathworks.com/help/matlab/matlab_external/upgrading-mex-files-to-use-64-bit-api.html.

To choose a different language, select one from the following:
 mex -setup C++
 mex -setup FORTRAN
fx >>
```

Figure 3.3.1: Command Window (Type mex -setup)

❖ Command mex -setup C++ as shown in Figure 3.3.2. (Noted: Compiler was chosen at this stage)

```
>> mex -setup C++
MEX configured to use 'Microsoft Visual C++ 2013 Professional' for C++ language compilation.
Warning: The MATLAB C and Fortran API has changed to support MATLAB
     variables with more than 2^32-1 elements. In the near future
     you will be required to update your code to utilize the
     new API. You can find more information about this at:
     http://www.mathworks.com/help/matlab/matlab_external/upgrading-mex-files-to-use-64-bit-api.html.
fx >>
```

Figure 3.3.2: Command Window (Type mex -setup C++)

❖ After that, the "EmotivEEG headset toolbox" was downloaded from Mathworks file exchange and the toolbox was put under the directory of MATLAB. After that, the library files provided by the Emotiv SDK was all added to the Emotiv headset toolbox.

❖ Open *EmotivEEG.m*, *ExampleUsage.m* (Noted: Provided by the EmotivEEG headset toolbox) using MATLAB and connect the Emotiv headset to the Emotiv Control Panel.

3.4 Offline BCI Design groundwork

- Offline data was recorded and saved in edf. files format using Emotiv Testbench. This format is commonly used EEG processing applications such as FieldTrip and BCILab. The edf. files will then be converted to CSV file using the Emotiv Testbench itself in order to process for reusability as shown in Figure 3.4.1.

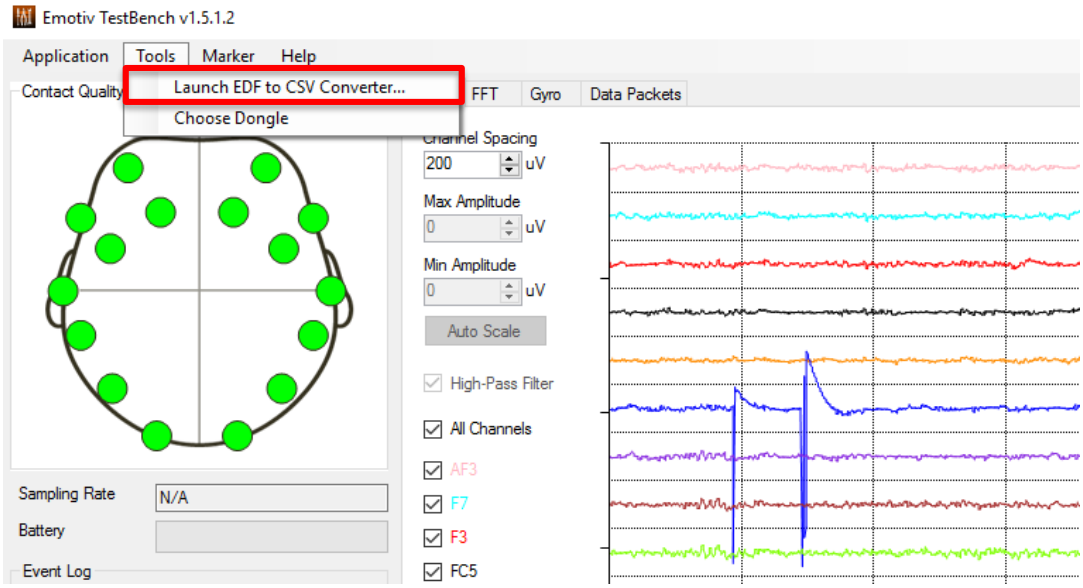


Figure 3.4.1: Convert EDF to CSV

- CSV. file is imported to MATLAB by selecting TABLE mode, and table function were generated to .m file as shown in Figure 3.4.2.

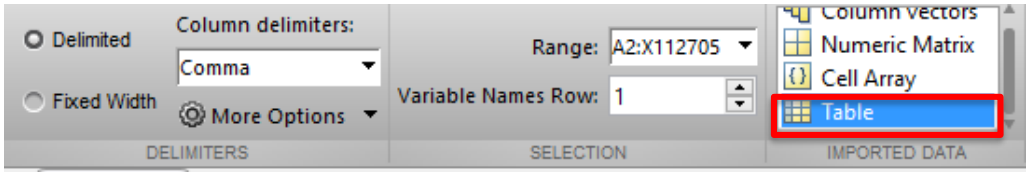


Figure 3.4.2: Transform data to TABLE form

- The function that was generated will be transferred into GUI as callback function as shown in Figure 3.4.3.

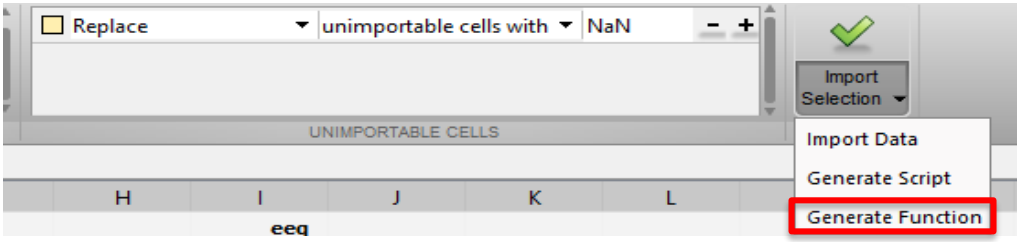
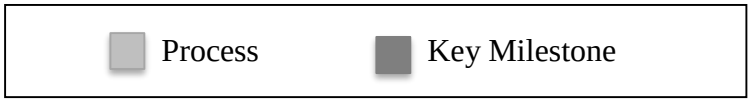


Figure 3.4.3: Generate Function from the csv file

3.5 Grant Chart



DETAIL/WEEK	FYP 1														FYP 2													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Topic Chosen																													
Literature Review																													
Extended Proposal																													
Getting Familiar with MATLAB and EEG Headset																													
Proposal Defense																													
Find ways to query raw EEG data in MATLAB																													
Conceptual Design of GUI Layout																													
Understanding the Architecture of BCI systems																													
Submission of draft Interim Report																													
Submission of Interim Report																													
Advanced Study of MATLAB GUI coding																													
Code writing in for element in MATLAB GUI																													
Design of Butterworths filter																													
Implementation and Design of MATLAB GUI																													
Loading raw Data in MATLAB GUI																													
Modification and Troubleshooting of MATLAB GUI																													
Transmission of raw EEG Data																													
Perform Signal Pre-Processing																													

CHAPTER 4

RESULT & DISCUSSION

The understanding of Electroencephalogram (EEG) signal theory is very important in order to design a Graphic User Interface (GUI). There are certain behaviors and techniques involve in creating an algorithm for acquisition and pre-processing of raw EEG signal. The layout of GUI designed need to be clean and have a clear instruction that can be easily understood by the user. This GUI will serve as a basic software for further analysis in another field which can be defined by the user. Therefore, the GUI need to have acquisition function and simple pre-processing function.

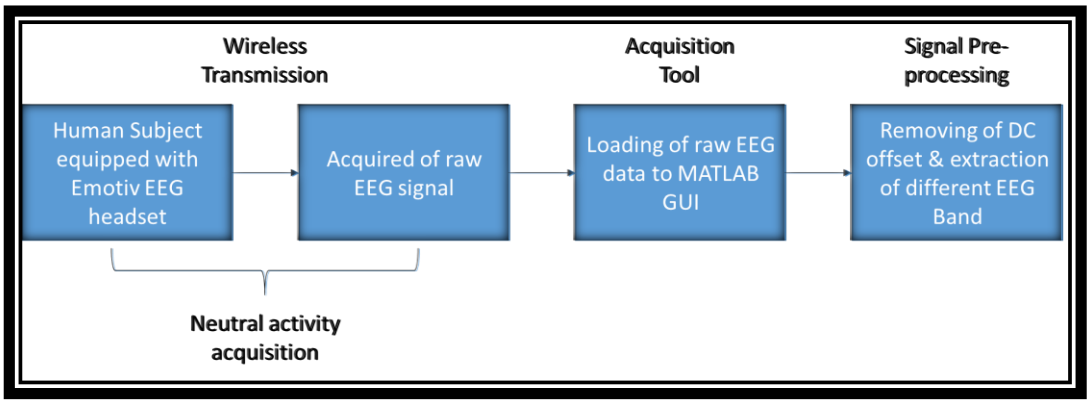


Figure 4.1: Block Diagram of EEG MATLAB GUI Acquisition System

Figure 4.1 shows the block diagram of EEG acquisition system. First, the wireless transmission of human subject equipped with Emotiv EEG headset to MATLAB GUI interface. After that, pre-processing of EEG was done by removing the DC offset, and different EEG band was filtered out.

4.1 Technical error

After connecting the Emotiv EEG headset to the computer through Bluetooth, Emotiv Control Panel and MATLAB were initiated simultaneously. “Emotiv EEG” was input in the MATLAB command window in order to get real-time EEG data. However, there were some warnings popped up which cause the Emotiv system to shut down as shown in Figure 4.1.1.

```
>> EmotivEEG
Warning: The function 'EE_GetSecurityCode' was not found in the library
> In loadlibrary (line 431)
  In EmotivEEG (line 73)
Warning: The function 'EE_CheckSecurityCode' was not found in the library
> In loadlibrary (line 431)
  In EmotivEEG (line 73)
EDK library loaded
Warning: Some error disconnecting from Emotiv Systems-5
> In EmotivEEG/delete (line 105)
Error using calllib
Parameter can not be converted to a string

Error in EmotivEEG (line 80)
            if calllib('edk','EE_EngineConnect',int8([self.unitIdentifier 0]))
```

Figure 4.1.1: EDK library loaded (Run *EmotivEEG.m*)

4.2 Offline BCI Result

Figure 4.2.1 is the GUI interface designed for EEG acquisition and signal pre-processing. Selection of offline and online analysis tool needed to be done at the initial stage. For online analysis, Emotiv EEG headset needs to be connected whereas for offline analysis, data saved earlier will be loaded in MATLAB GUI in order to perform signal pre-processing. After that, users can select the EEG band that they want to filter out. Lastly, select the nodes needed to plot the EEG signal. The figure at the left-hand sides will show node selected by the user.

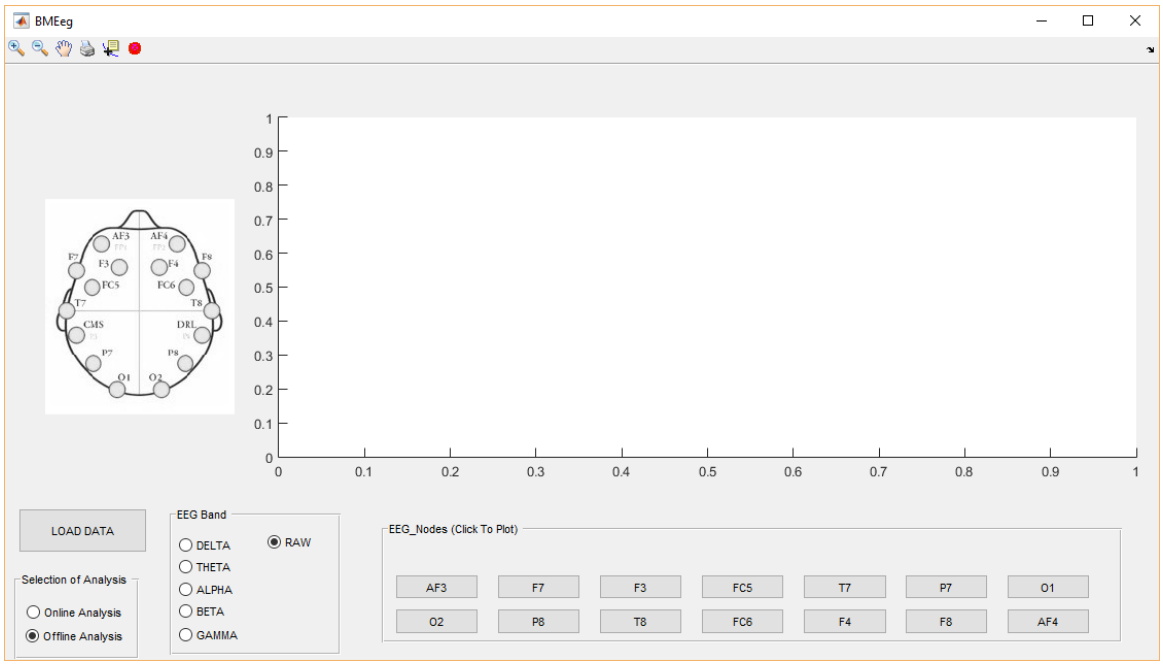


Figure 4.2.1: GUI for EEG Acquisition and Signal pre-Processing

Plotting of offline data in animated form with removing of dc offset for the raw EEG signal was shown in figure 4.2.2.

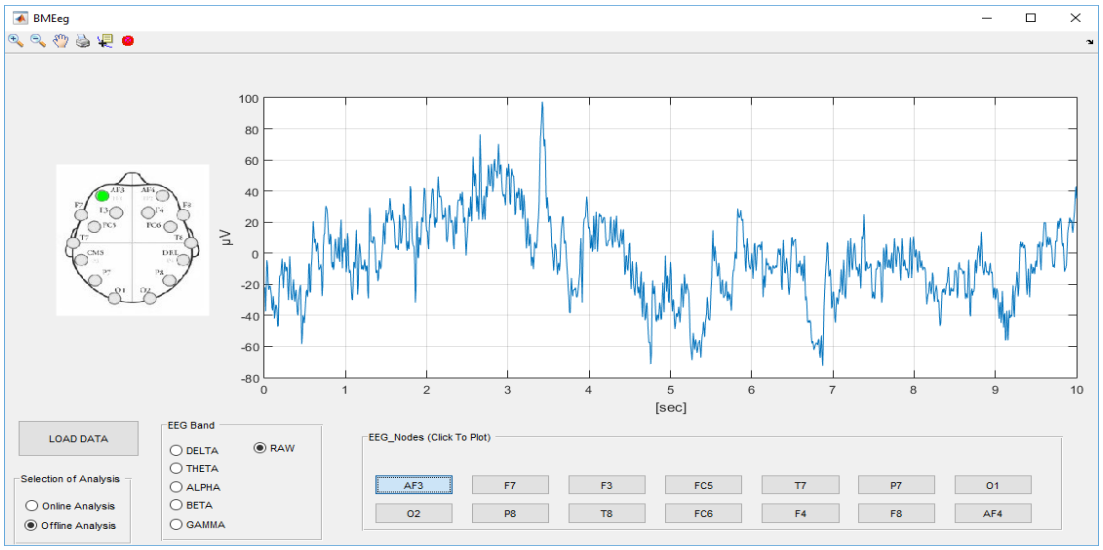


Figure 4.2.2: EEG pre-processing of raw data (node AF3)

4.3 Band Pass Filter Design

Table 4: Characteristic of EEG Band

	Gamma	**Beta**	**Alpha**	**Theta**	**Delta**
Sampling Frequency (Hz)	128	128	128	128	128
Filter Order (n)	4	4	4	4	4
Lower Cutoff frequency (Hz)	30	12	8	4	-
Higher Cutoff frequency (Hz)	100	30	12	8	4

As can be observed in Table 4, the sampling frequency of the EEG band was at 128Hz, and the order of the filter n was set to 4. Besides that, the cutoff frequency of each EEG band was listed down so that the Butterworth bandpass filter can be easily designed by using Signal Processing Toolbox.

4.4 Plotting of different wave patterns (node AF3)

The results as shown in Figure 4.4.1, 4.4.2, 4.4.3, 4.4.4 and 4.4.5 were plotted according to the type of EEG Band selected by users.

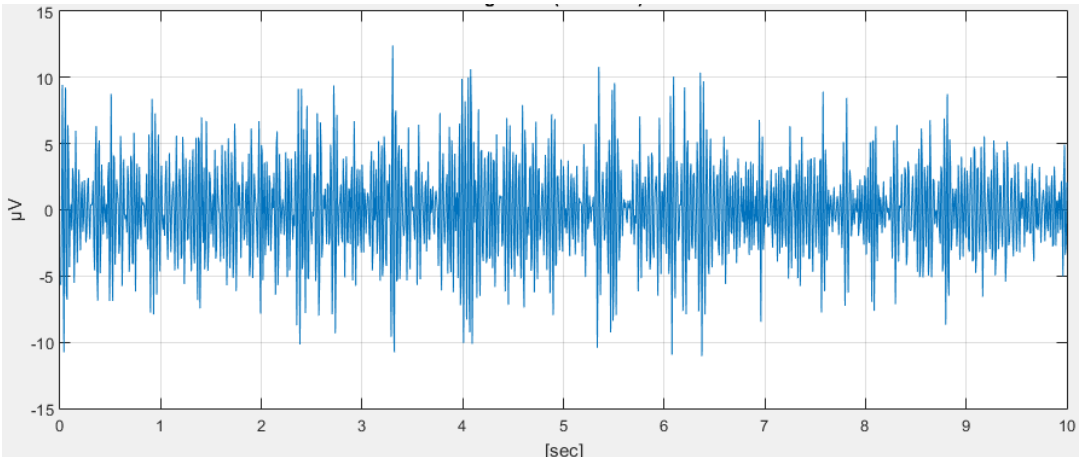


Figure 4.4.1: Gamma Band (node AF3)

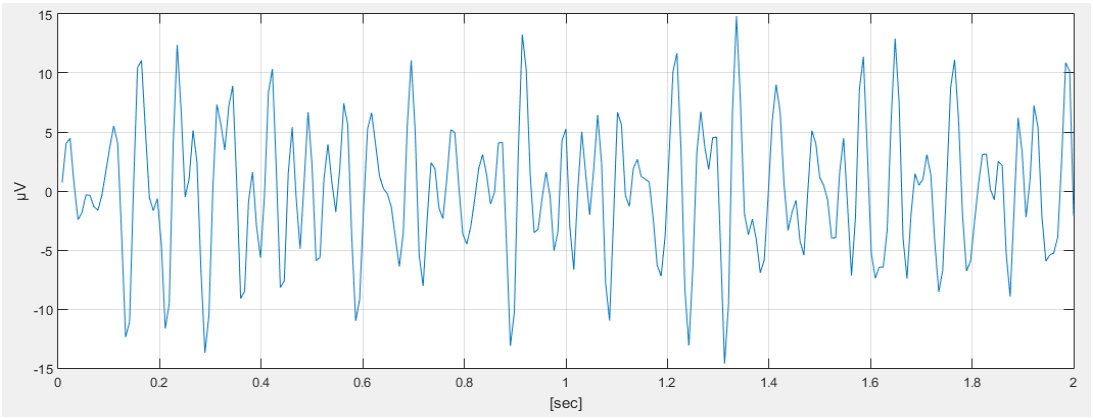


Figure 4.4.2: Beta Band (node AF3)

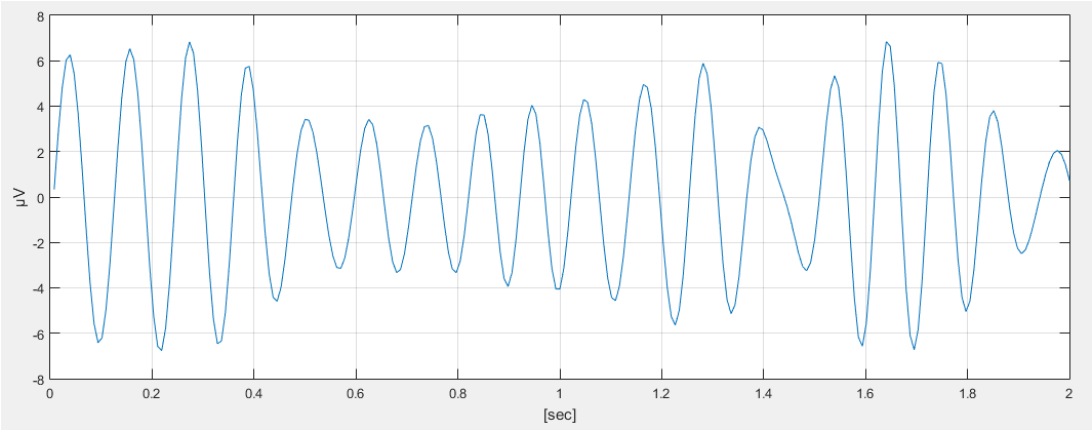


Figure 4.4.3: Alpha Band (node AF3)

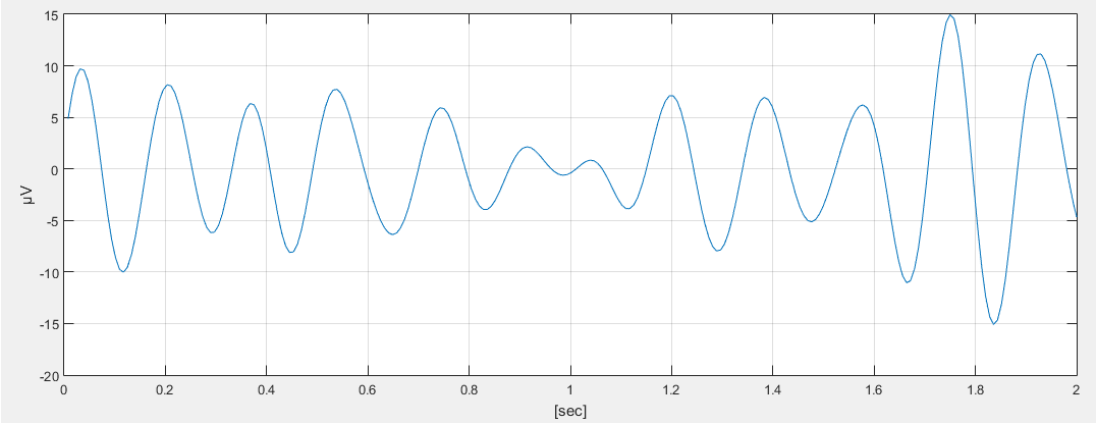


Figure 4.4.4: Theta Band (node AF3)

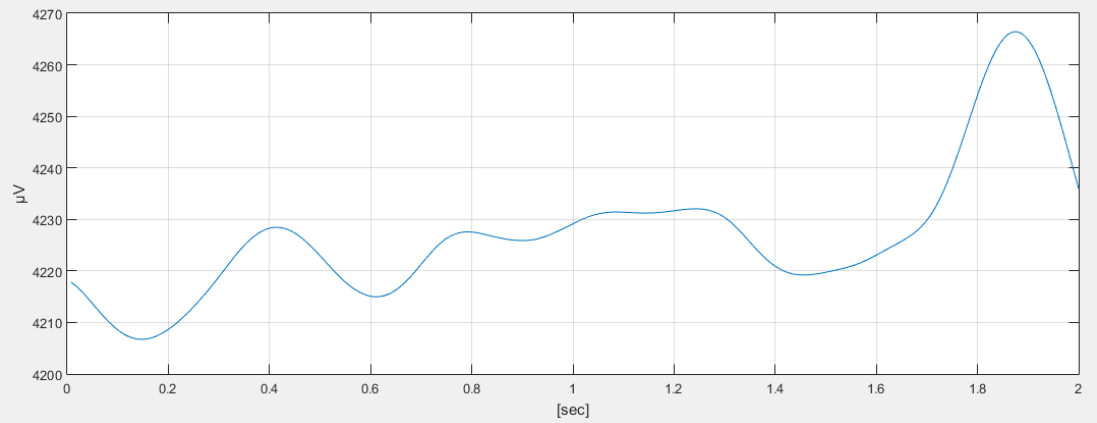


Figure 4.4.5: Delta Band (node AF3)

CHAPTER 5

Conclusion and Recommendations

5.1 Conclusion

GUI design for EEG signal acquisition and pre-processing begins with the selection of EEG acquisition tool, and software follows by designing a user friendly interface to perform pre-processing of selected raw signal and with the plotting of selected band and nodes. The acquired brain signals were sent to the computer via Bluetooth and saved as a .csv file to perform signal pre-processing. Operations like removing of DC offset and the extraction of specific EEG frequency were implemented to collect EEG signals. In this project, the selection of filter is very important to get a high-quality output. Butterworth IIR filter was selected as a pre-processing tool because of its passband and stopbands are maximally flat, thus resulting in a quality output signal for the different frequency bands. Furthermore, a reasonably sharp fall-off can be reached.

The challenges that faced when implementing the Brain Computer Interface (BCI) system are variability in the acquired online EEG signals via the Emotiv headset to MATLAB GUI. The latest result showed that the function "EE_GetSecurityCode" is in the header file of the toolbox but not included in the edk. library, so when the MATLAB called the function, the error "not found in the library" appeared . By removing the “EE_GetSecurityCode“line from the header file will prevent the error that causes faulty during the transmission of raw data from headset to MATLAB. However, the offline data was successfully loaded in the MATLAB GUI, and the MATLAB GUI can easily perform pre-processing of the signal.

5.2 Recommendation

For further development of this project, improvement can be made to the algorithm, signal processing, and classification procedures with addition of featuring extensions. This algorithm could be further enhanced to control remote applications. A system builds with signals consisting of more number of electrodes would have high precision in classification of the signals. They are of great help for improving the quality of life for physically impaired people to perform physically challenging tasks with the aid of BCI system.

References

[1] Rashima Mahajan, Dipali Bansal, and S. Singh, "<A-Real-Time-Set-Up-for-Retrieval-of-Emotional-States-from-Human-Neural-Responses.pdf>," *International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering*, vol. 8, p. 6, 2014.

[2] A. Delorme, T. Mullen, C. Kothe, Z. Akalin Acar, N. Bigdely-Shamlo, A. Vankov, *et al.*, "EEGLAB, SIFT, NFT, BCILAB, and ERICA: new tools for advanced EEG processing," *Comput Intell Neurosci*, vol. 2011, p. 130714, 2011.

[3] Jonathan R. Wolpaw and C. B. Boulay, *Brain-Computer Interfaces*: Springer Berlin Heidelberg, 2010.

[4] H. Ekanayake, "P300 and Emotiv EPOC: Does Emotiv EPOC capture real EEG?," *Research Use of Emotiv EPOC*, January 5 2015.

[5] (2015, 26 Oct). *Wearables for your brain | EEG*. Available: https://emotiv.com/

[6] Y. Liu, O. Sourina, and M. K. Nguyen, "Real-Time EEG-Based Human Emotion Recognition and Visualization," pp. 262-269, 2010.

[7] S. Munian, S. Sivalingam, and V. Jayaraman, "Analysis of Real Time EEG Signals," Master, Electrical Engineering, Linnaeus University, Sweden, 2014.

[8] A. E. Hassanien and A. T. A. Editors, *Brain-Computer Interfaces* vol. 74, 2015.

[9] P. Teich, "Designing a Brain Computer Interface Using an Affordable EEG Headset.," Master, Mathematics and Computer Science, Free University Berlin, Berlin, 2015.

[10] D. T. V. U. K. Kumar, "Generation and reconstruction of EEG signals using matlab," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, 2013.

[11] Yisi Liu, Olga Sourina, and M. K. Nguyen, "Real-time EEG-based Human Emotion Recognition and Visualization," presented at the 2010 International Conference on Cyberworlds, Nanyang Technological University, Singapore, 2010.

[12] S. Sontisirkit, "SPECIAL STUDY ON GUILDLINES TO DEVELOP BRAIN COMPUTER INTERFACE," Master of Computer Science Asian Institute of Technology, Thailand, 2013.

[13] GUIDELINE TO DEVELOP BRAIN COMPUTER INTERFACE [Online]. Available: http://sralife.com/?page=1_workblog_eeg-2014

[14] Emotiv, "TestBench™ User Manual," ed, 2016.

[15] M. Adelson, "<ExperimenterReport.pdf>," PRINCETON UNIVERSITY2011.

[16] A. Delorme, C. Kothe, A. Vankov, N. Bigdely-Shamlo, R. Oostenveld, T. Zander, *et al.*, "MATLAB-Based Tools for BCI Research," in *Brain-Computer Interfaces*, ed: Springer London, 2010, pp. pp 241-259.

APPENDICES

Setup Environments

1) The Emotiv Education Edition SDK and Visual Studio 2010 were installed in the PC.
2) EPOC EEG headset was prepared.
3) The foam-tipped sensors were wet with contact lens solution provided in the EMOTIV headset toolbox before insert into the black plastic headset arms.

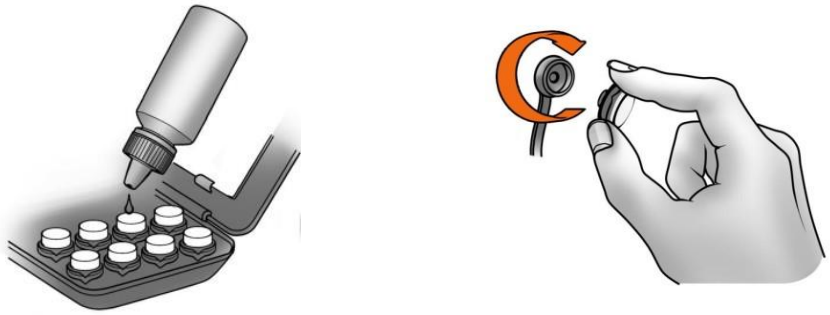


Figure 1 : Preparation for Emotiv EEG headset

4) Pairing the neuroheadset to the USB receiver, and then wear it in the right position.

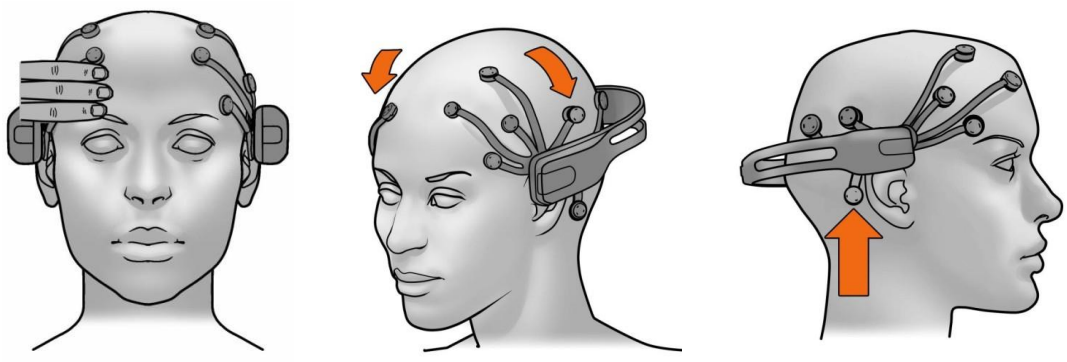


Figure 2: Position of Emotiv EEG headset

5) After connected Emotiv neuroheadset to the USB receiver, Emotiv Control Panel was initiated to check the strength of the brain signal. There are 5 colors of signal quality:

- Black: No signal
- Yellow, Orange, Red: Poor signal
- Green: Good signal

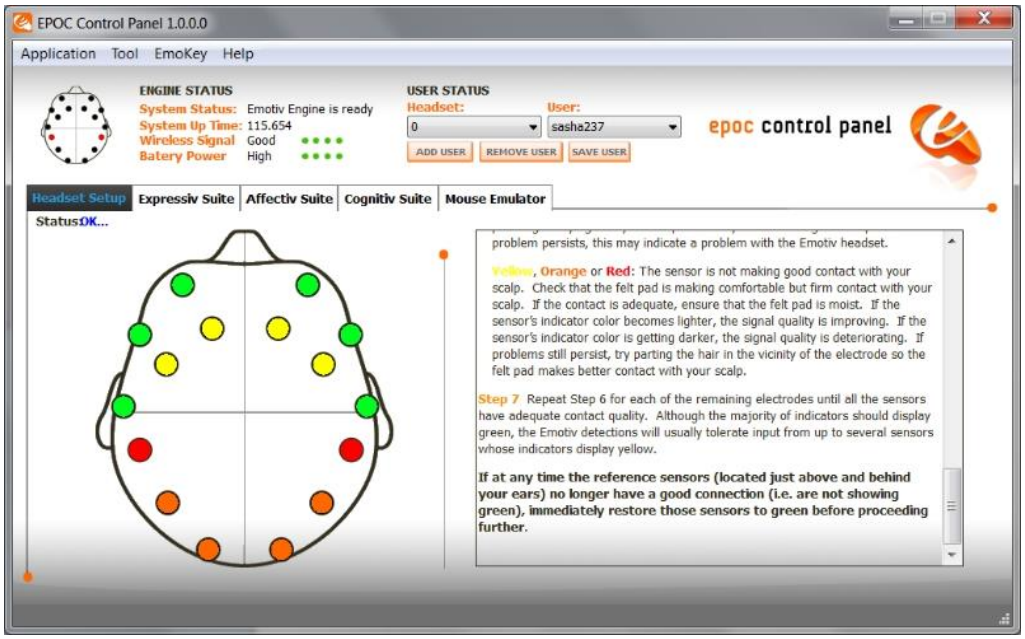


Figure 3: EPOC Control Panel

BMEeg.exe MATLAB Code

```
function varargout = nodes(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @nodes_OpeningFcn, ...
                   'gui_OutputFcn',  @nodes_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before nodes is made visible.
function nodes_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for nodes
handles.output = hObject;
axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\non.jpg');


% Update handles structure
guidata(hObject, handles);

% UIWAIT makes nodes wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = nodes_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;


function
[COUNTER,AF3,F7,F3,FC5,T7,P7,O1,O2,P8,T8,FC6,F4,F8,AF4,GYROX,GYROY,TIMESTAMP,FUNC_ID,FUNC_VALU
E,MARKER,SYNC_SIGNAL,TIME,VarName24] = csvread(filename, startRow, endRow)
%% Initialize variables.
delimiter = ',';
if nargin<=2
   startRow = 2;
   endRow = inf;
end

formatSpec = '%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%s%s%[^\n\r]';
fileID = fopen(filename,'r');

textscan(fileID, '%[^\n\r]', startRow(1)-1, 'ReturnOnError', false);
dataArray = textscan(fileID, formatSpec, endRow(1)-startRow(1)+1, 'Delimiter', delimiter, 'EmptyValue' ,NaN,'ReturnOnError',
false);
for block=2:length(startRow)
   frewind(fileID);
   textscan(fileID, '%[^\n\r]', startRow(block)-1, 'ReturnOnError', false);
   dataArrayBlock = textscan(fileID, formatSpec, endRow(block)-startRow(block)+1, 'Delimiter', delimiter,
'EmptyValue' ,NaN,'ReturnOnError', false);
   for col=1:length(dataArray)
      dataArray{col} = [dataArray{col};dataArrayBlock{col}];
   end
end

fclose(fileID);
```

```
COUNTER = dataArray{:, 1};
AF3 = dataArray{:, 2};
F7 = dataArray{:, 3};
F3 = dataArray{:, 4};
FC5 = dataArray{:, 5};
T7 = dataArray{:, 6};
P7 = dataArray{:, 7};
O1 = dataArray{:, 8};
O2 = dataArray{:, 9};
P8 = dataArray{:, 10};
T8 = dataArray{:, 11};
FC6 = dataArray{:, 12};
F4 = dataArray{:, 13};
F8 = dataArray{:, 14};
AF4 = dataArray{:, 15};
GYROX = dataArray{:, 16};
GYROY = dataArray{:, 17};


% --- Executes on button press in online.
function online_Callback(hObject, eventdata, handles)

 errordlg('Please Connect EEG Headset','Error');


% --- Executes on button press in offline.
function offline_Callback(hObject, eventdata, handles)


% --- Executes on button press in load.
function load_Callback(hObject, eventdata, handles)
structure with handles and user data (see GUIDATA)

[filename,pathname] = uigetfile('*.csv');
handles.filename = [ pathname filename ];

guidata(hObject, handles)

% --- Executes on button press in AF3.
function AF3_Callback(hObject, eventdata, handles)

axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\AF3.jpg');

filename = handles.filename;
[~, AF3] = csvread(filename, 2, inf);
QQ = transpose(AF3);
fs = 128;
timeBase = 10;
N = length(QQ);
n = 4;
hAx = handles.eeg;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);

ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);
```

```
            if get(hObject,'value')

                set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
                set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

                drawnow;

                xlabel('[sec]');
                ylabel('µV');
                grid on;

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.gamma          % (30-100Hz)

    W3 = 60/fs;
    W4 = 100/fs;
    Wn_a = [W3 W4];
    [g,h] = butter(n,Wn_a);
    gamma = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
                drawnow
                title ('gamma (30-100Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.beta           % (12-30Hz)


    W5 = 24/fs;
    W6 = 60/fs;
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
```

```
            title ('beta (12-30Hz)');
            xlabel('[sec]');
            ylabel('µV');
            grid on;
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

         else
            break;
         end
      end
   end

case handles.alpha          % (8-12Hz)


   W3 = 16/fs;
   W4 = 24/fs;
   Wn_a = [W3 W4];
   [e,f] = butter(n,Wn_a);
   alpha = filter(e,f,QQ);

   if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

         if get(hObject,'value')

            set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
            title ('alpha (8-12Hz)')
            xlabel('[sec]');
            ylabel('µV');
            grid on;
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

         else
            break;
         end
      end
   end

case handles.theta          % (4-7Hz)

   W1 = 8/fs;
   W2 = 14/fs;
   Wn_t = [W1 W2];
   [c,d] = butter(n,Wn_t);
   theta = filter(c,d,QQ);

   if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

         if get(hObject,'value')

            set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
            title ('theta (4-7Hz)');
            xlabel('[sec]');
            ylabel('µV');
            grid on;
            drawnow
```

```
            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

        else
            break;
        end
    end
end

case handles.delta          %  (<4Hz)

    Wn_d = 8/fs;
    [b,a] = butter(n,Wn_d);
    delta = filter(b,a,QQ);
    delta = detrend (delta);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
                title ('delta (<4Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

end

 guidata(hObject, handles)

% --- Executes on button press in F7.
function F7_Callback(hObject, eventdata, handles)
% hObject    handle to F7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\F7.jpg');

filename = handles.filename;
[~, ~, F7] = csvread(filename);

QQ = transpose(F7);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);


ylim auto;
tic;
```

```
bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

            if get(hObject,'value')

               set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
               set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

               drawnow;

               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

   case handles.gamma            % (30-100Hz)

      W3 = 60/fs;
      W4 = 100/fs;
      Wn_a = [W3 W4];
      [g,h] = butter(n,Wn_a);
      gamma = filter(g,h,QQ);

      if get(hObject,'value')

         hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

         while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

               set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
               drawnow
               title ('gamma (30-100Hz)');
               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

   case handles.beta             % (12-30Hz)
```

```
    W5 = 24/fs;
    W6 = 60/fs;
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

       hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

       while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

             set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
             title ('beta (12-30Hz)');
             xlabel('[sec]');
             ylabel('µV');
             grid on;
             drawnow

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
       end
    end

case handles.alpha          % (8-12Hz)


    W3 = 16/fs;
    W4 = 24/fs;
    Wn_a = [W3 W4];
    [e,f] = butter(n,Wn_a);
    alpha = filter(e,f,QQ);

    if get(hObject,'value')

       hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

       while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

             set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
             title ('alpha (8-12Hz)')
             xlabel('[sec]');
             ylabel('µV');
             grid on;
             drawnow

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
       end
    end

case handles.theta          % (4-7Hz)

    W1 = 8/fs;
    W2 = 14/fs;
    Wn_t = [W1 W2];
    [c,d] = butter(n,Wn_t);
    theta = filter(c,d,QQ);
```

```
    if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

        if get(hObject,'value')

          set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
          title ('theta (4-7Hz)');
          xlabel('[sec]');
          ylabel('µV');
          grid on;
          drawnow

          t = toc;

          ind = round(t*fs);
          ind = max(ind,1);

        else
          break;
        end
      end
    end

  case handles.delta          %  (<4Hz)

    Wn_d = 8/fs;
    [b,a] = butter(n,Wn_d);
    delta = filter(b,a,QQ);
    delta = detrend (delta);

    if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

        if get(hObject,'value')

          set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
          title ('delta (<4Hz)');
          xlabel('[sec]');
          ylabel('µV');
          grid on;
          drawnow

          t = toc;

          ind = round(t*fs);
          ind = max(ind,1);

        else
          break;
        end
      end
    end

end

 guidata(hObject, handles)

% --- Executes on button press in F3.
function F3_Callback(hObject, eventdata, handles)
% hObject    handle to F3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\F3.jpg');

filename = handles.filename;
[ ~, ~,~, F3] = csvread(filename);
```

```
QQ = transpose(F3);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);


ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

            if get(hObject,'value')

               set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
               set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

               drawnow;

               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

   case handles.gamma          % (30-100Hz)

      W3 = 60/fs;
      W4 = 100/fs;
      Wn_a = [W3 W4];
      [g,h] = butter(n,Wn_a);
      gamma = filter(g,h,QQ);

      if get(hObject,'value')

         hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

         while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

               set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
               drawnow
               title ('gamma (30-100Hz)');
               xlabel('[sec]');
               ylabel('µV');
```

```
            grid on;

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

        else
            break;
        end
    end
end

case handles.beta           % (12-30Hz)


    W5 = 24/fs;
    W6 = 60/fs;
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
                title ('beta (12-30Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.alpha          % (8-12Hz)


    W3 = 16/fs;
    W4 = 24/fs;
    Wn_a = [W3 W4];
    [e,f] = butter(n,Wn_a);
    alpha = filter(e,f,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
                title ('alpha (8-12Hz)')
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
```

```
            ind = max(ind,1);

        else
            break;
        end
    end
end

case handles.theta          % (4-7Hz)

    W1 = 8/fs;
    W2 = 14/fs;
    Wn_t = [W1 W2];
    [c,d] = butter(n,Wn_t);
    theta = filter(c,d,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
                title ('theta (4-7Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.delta          %  (<4Hz)

    Wn_d = 8/fs;
    [b,a] = butter(n,Wn_d);
    delta = filter(b,a,QQ);
    delta = detrend (delta);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
                title ('delta (<4Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end
```

```
end

 guidata(hObject, handles)

% --- Executes on button press in FC5.
function FC5_Callback(hObject, eventdata, handles)
% hObject    handle to FC5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\FC5.jpg');

filename = handles.filename;
[~,~,~,~,FC5] = csvread(filename);

QQ = transpose(FC5);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);


ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

            if get(hObject,'value')

               set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
               set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

               drawnow;

               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

   case handles.gamma            % (30-100Hz)

      W3 = 60/fs;
      W4 = 100/fs;
      Wn_a = [W3 W4];
```

```
    [g,h] = butter(n,Wn_a);
    gamma = filter(g,h,QQ);

    if get(hObject,'value')

       hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

       while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

             set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
             drawnow
             title ('gamma (30-100Hz)');
             xlabel('[sec]');
             ylabel('µV');
             grid on;

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
       end
    end

 case handles.beta            % (12-30Hz)


    W5 = 24/fs;
    W6 = 60/fs;
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

       hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

       while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

             set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
             title ('beta (12-30Hz)');
             xlabel('[sec]');
             ylabel('µV');
             grid on;
             drawnow

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
       end
    end

 case handles.alpha           % (8-12Hz)


    W3 = 16/fs;
    W4 = 24/fs;
    Wn_a = [W3 W4];
    [e,f] = butter(n,Wn_a);
    alpha = filter(e,f,QQ);

    if get(hObject,'value')
```

```
        hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
                title ('alpha (8-12Hz)')
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.theta          % (4-7Hz)

    W1 = 8/fs;
    W2 = 14/fs;
    Wn_t = [W1 W2];
    [c,d] = butter(n,Wn_t);
    theta = filter(c,d,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
                title ('theta (4-7Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.delta          %  (<4Hz)

    Wn_d = 8/fs;
    [b,a] = butter(n,Wn_d);
    delta = filter(b,a,QQ);
    delta = detrend (delta);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
                title ('delta (<4Hz)');
```

```
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

end

 guidata(hObject, handles)

% --- Executes on button press in T7.
function T7_Callback(hObject, eventdata, handles)
% hObject    handle to T7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\T7.jpg');

filename = handles.filename;
[~,~,~,~,~,T7] = csvread(filename);

QQ = transpose(T7);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);


ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

            if get(hObject,'value')

               set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
               set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

               drawnow;

               xlabel('[sec]');
               ylabel('µV');
               grid on;
```

```
            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

        else
            break;
        end
    end
end

case handles.gamma          % (30-100Hz)

    W3 = 60/fs;
    W4 = 100/fs;
    Wn_a = [W3 W4];
    [g,h] = butter(n,Wn_a);
    gamma = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
                drawnow
                title ('gamma (30-100Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.beta           % (12-30Hz)


    W5 = 24/fs;
    W6 = 60/fs;
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
                title ('beta (12-30Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);
```

```
            else
               break;
            end
         end
      end

   case handles.alpha          % (8-12Hz)


      W3 = 16/fs;
      W4 = 24/fs;
      Wn_a = [W3 W4];
      [e,f] = butter(n,Wn_a);
      alpha = filter(e,f,QQ);

      if get(hObject,'value')

         hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

         while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

               set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
               title ('alpha (8-12Hz)')
               xlabel('[sec]');
               ylabel('µV');
               grid on;
               drawnow

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

   case handles.theta          % (4-7Hz)

      W1 = 8/fs;
      W2 = 14/fs;
      Wn_t = [W1 W2];
      [c,d] = butter(n,Wn_t);
      theta = filter(c,d,QQ);

      if get(hObject,'value')

         hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

         while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

               set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
               title ('theta (4-7Hz)');
               xlabel('[sec]');
               ylabel('µV');
               grid on;
               drawnow

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end
```

```
   case handles.delta           %  (<4Hz)

      Wn_d = 8/fs;
      [b,a] = butter(n,Wn_d);
      delta = filter(b,a,QQ);
      delta = detrend (delta);

      if get(hObject,'value')

         hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

         while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

               set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
               title ('delta (<4Hz)');
               xlabel('[sec]');
               ylabel('µV');
               grid on;
               drawnow

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

end

 guidata(hObject, handles)

% --- Executes on button press in P7.
function P7_Callback(hObject, eventdata, handles)
% hObject    handle to P7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\P7.jpg');

filename = handles.filename;
[~,~,~,~,~,~,P7] = csvread(filename);

QQ = transpose(P7);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);


ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')
```

```
        hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
        hold on;
        hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


        while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

          if get(hObject,'value')

             set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
             set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

             drawnow;

             xlabel('[sec]');
             ylabel('µV');
             grid on;

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
        end
    end

case handles.gamma          % (30-100Hz)

    W3 = 60/fs;
    W4 = 100/fs;
    Wn_a = [W3 W4];
    [g,h] = butter(n,Wn_a);
    gamma = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

             set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
             drawnow
             title ('gamma (30-100Hz)');
             xlabel('[sec]');
             ylabel('µV');
             grid on;

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
        end
    end

case handles.beta           % (12-30Hz)


    W5 = 24/fs;
    W6 = 60/fs;
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));
```

```
        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
                title ('beta (12-30Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.alpha          % (8-12Hz)


    W3 = 16/fs;
    W4 = 24/fs;
    Wn_a = [W3 W4];
    [e,f] = butter(n,Wn_a);
    alpha = filter(e,f,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
                title ('alpha (8-12Hz)')
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.theta          % (4-7Hz)

    W1 = 8/fs;
    W2 = 14/fs;
    Wn_t = [W1 W2];
    [c,d] = butter(n,Wn_t);
    theta = filter(c,d,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
```

```
                title ('theta (4-7Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

  case handles.delta           %  (<4Hz)

    Wn_d = 8/fs;
    [b,a] = butter(n,Wn_d);
    delta = filter(b,a,QQ);
    delta = detrend (delta);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
                title ('delta (<4Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

end

 guidata(hObject, handles)

% --- Executes on button press in O1.
function O1_Callback(hObject, eventdata, handles)
% hObject    handle to O1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\O1.jpg');

filename = handles.filename;
[~,~,~,~,~,~,~,O1] = csvread(filename);

QQ = transpose(O1);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
```

```
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);


ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

            if get(hObject,'value')

               set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
               set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

               drawnow;

               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

   case handles.gamma          % (30-100Hz)

      W3 = 60/fs;
      W4 = 100/fs;
      Wn_a = [W3 W4];
      [g,h] = butter(n,Wn_a);
      gamma = filter(g,h,QQ);

      if get(hObject,'value')

         hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

         while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

               set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
               drawnow
               title ('gamma (30-100Hz)');
               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
```

```
            end
        end
    end

case handles.beta           % (12-30Hz)


    W5 = 24/fs;
    W6 = 60/fs;
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
                title ('beta (12-30Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.alpha          % (8-12Hz)


    W3 = 16/fs;
    W4 = 24/fs;
    Wn_a = [W3 W4];
    [e,f] = butter(n,Wn_a);
    alpha = filter(e,f,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
                title ('alpha (8-12Hz)')
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.theta          % (4-7Hz)
```

```
    W1 = 8/fs;
    W2 = 14/fs;
    Wn_t = [W1 W2];
    [c,d] = butter(n,Wn_t);
    theta = filter(c,d,QQ);

    if get(hObject,'value')

       hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

       while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

             set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
             title ('theta (4-7Hz)');
             xlabel('[sec]');
             ylabel('µV');
             grid on;
             drawnow

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
       end
    end

  case handles.delta          %  (<4Hz)

    Wn_d = 8/fs;
    [b,a] = butter(n,Wn_d);
    delta = filter(b,a,QQ);
    delta = detrend (delta);

    if get(hObject,'value')

       hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

       while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

             set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
             title ('delta (<4Hz)');
             xlabel('[sec]');
             ylabel('µV');
             grid on;
             drawnow

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
       end
    end

end

 guidata(hObject, handles)

% --- Executes on button press in O2.
function O2_Callback(hObject, eventdata, handles)
% hObject    handle to O2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\O2.jpg');

filename = handles.filename;
[~,~,~,~,~,~,~,~,O2] = csvread(filename);

QQ = transpose(O2);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);


ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

            if get(hObject,'value')

               set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
               set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

               drawnow;

               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

   case handles.gamma          % (30-100Hz)

      W3 = 60/fs;
      W4 = 100/fs;
      Wn_a = [W3 W4];
      [g,h] = butter(n,Wn_a);
      gamma = filter(g,h,QQ);

      if get(hObject,'value')

         hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

         while (ind < N-nSamples) && ishandle(hLine)
```

```
            if get(hObject,'value')

                set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
                drawnow
                title ('gamma (30-100Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.beta           % (12-30Hz)


    W5 = 24/fs;
    W6 = 60/fs;
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
                title ('beta (12-30Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.alpha          % (8-12Hz)


    W3 = 16/fs;
    W4 = 24/fs;
    Wn_a = [W3 W4];
    [e,f] = butter(n,Wn_a);
    alpha = filter(e,f,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
                title ('alpha (8-12Hz)')
                xlabel('[sec]');
```

```
            ylabel('µV');
            grid on;
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

         else
            break;
         end
      end
   end

case handles.theta          % (4-7Hz)

   W1 = 8/fs;
   W2 = 14/fs;
   Wn_t = [W1 W2];
   [c,d] = butter(n,Wn_t);
   theta = filter(c,d,QQ);

   if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

         if get(hObject,'value')

            set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
            title ('theta (4-7Hz)');
            xlabel('[sec]');
            ylabel('µV');
            grid on;
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

         else
            break;
         end
      end
   end

case handles.delta          %  (<4Hz)

   Wn_d = 8/fs;
   [b,a] = butter(n,Wn_d);
   delta = filter(b,a,QQ);
   delta = detrend (delta);

   if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

         if get(hObject,'value')

            set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
            title ('delta (<4Hz)');
            xlabel('[sec]');
            ylabel('µV');
            grid on;
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);
```

```
            else
               break;
            end
         end
      end

end

 guidata(hObject, handles)

% --- Executes on button press in P8.
function P8_Callback(hObject, eventdata, handles)
% hObject    handle to P8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\P8.jpg');

filename = handles.filename;
[~,~,~,~,~,~,~,~,~,P8] = csvread(filename);

QQ = transpose(P8);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);


ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

            if get(hObject,'value')

               set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
               set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

               drawnow;

               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
```

```
        end
    end

case handles.gamma          % (30-100Hz)

    W3 = 60/fs;
    W4 = 100/fs;
    Wn_a = [W3 W4];
    [g,h] = butter(n,Wn_a);
    gamma = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
                drawnow
                title ('gamma (30-100Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.beta           % (12-30Hz)


    W5 = 24/fs;
    W6 = 60/fs;
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
                title ('beta (12-30Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.alpha          % (8-12Hz)
```

```
    W3 = 16/fs;
    W4 = 24/fs;
    Wn_a = [W3 W4];
    [e,f] = butter(n,Wn_a);
    alpha = filter(e,f,QQ);

    if get(hObject,'value')

       hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

       while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

             set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
             title ('alpha (8-12Hz)')
             xlabel('[sec]');
             ylabel('µV');
             grid on;
             drawnow

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
       end
    end

case handles.theta          % (4-7Hz)

    W1 = 8/fs;
    W2 = 14/fs;
    Wn_t = [W1 W2];
    [c,d] = butter(n,Wn_t);
    theta = filter(c,d,QQ);

    if get(hObject,'value')

       hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

       while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

             set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
             title ('theta (4-7Hz)');
             xlabel('[sec]');
             ylabel('µV');
             grid on;
             drawnow

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
       end
    end

case handles.delta          %  (<4Hz)

    Wn_d = 8/fs;
    [b,a] = butter(n,Wn_d);
    delta = filter(b,a,QQ);
    delta = detrend (delta);

    if get(hObject,'value')
```

```
        hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

           if get(hObject,'value')

              set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
              title ('delta (<4Hz)');
              xlabel('[sec]');
              ylabel('µV');
              grid on;
              drawnow

              t = toc;

              ind = round(t*fs);
              ind = max(ind,1);

           else
              break;
           end
        end
     end

end

 guidata(hObject, handles)

% --- Executes on button press in T8.
function T8_Callback(hObject, eventdata, handles)
% hObject    handle to T8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\T8.jpg');

filename = handles.filename;
[~,~,~,~,~,~,~,~,~,T8] = csvread(filename);
QQ = transpose(T8);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);


ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

            if get(hObject,'value')

               set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
```

```
            set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

            drawnow;

            xlabel('[sec]');
            ylabel('µV');
            grid on;

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

         else
            break;
         end
      end
   end

case handles.gamma            % (30-100Hz)

   W3 = 60/fs;
   W4 = 100/fs;
   Wn_a = [W3 W4];
   [g,h] = butter(n,Wn_a);
   gamma = filter(g,h,QQ);

   if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

         if get(hObject,'value')

            set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
            drawnow
            title ('gamma (30-100Hz)');
            xlabel('[sec]');
            ylabel('µV');
            grid on;

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

         else
            break;
         end
      end
   end

case handles.beta             % (12-30Hz)


   W5 = 24/fs;
   W6 = 60/fs;
   Wn_b = [W5 W6];
   [g,h] = butter(n,Wn_b);
   beta = filter(g,h,QQ);

   if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

         if get(hObject,'value')

            set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
            title ('beta (12-30Hz)');
            xlabel('[sec]');
            ylabel('µV');
            grid on;
```

```
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

         else
            break;
         end
      end
   end

case handles.alpha          % (8-12Hz)


   W3 = 16/fs;
   W4 = 24/fs;
   Wn_a = [W3 W4];
   [e,f] = butter(n,Wn_a);
   alpha = filter(e,f,QQ);

   if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

         if get(hObject,'value')

            set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
            title ('alpha (8-12Hz)')
            xlabel('[sec]');
            ylabel('µV');
            grid on;
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

         else
            break;
         end
      end
   end

case handles.theta          % (4-7Hz)

   W1 = 8/fs;
   W2 = 14/fs;
   Wn_t = [W1 W2];
   [c,d] = butter(n,Wn_t);
   theta = filter(c,d,QQ);

   if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

         if get(hObject,'value')

            set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
            title ('theta (4-7Hz)');
            xlabel('[sec]');
            ylabel('µV');
            grid on;
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);
```

```
            else
               break;
            end
         end
      end

   case handles.delta            %  (<4Hz)

      Wn_d = 8/fs;
      [b,a] = butter(n,Wn_d);
      delta = filter(b,a,QQ);
      delta = detrend (delta);

      if get(hObject,'value')

         hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

         while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

               set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
               title ('delta (<4Hz)');
               xlabel('[sec]');
               ylabel('µV');
               grid on;
               drawnow

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

end

 guidata(hObject, handles)

% --- Executes on button press in FC6.
function FC6_Callback(hObject, eventdata, handles)
% hObject    handle to FC6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\FC6.jpg');

filename = handles.filename;
[~,~,~,~,~,~,~,~,~,~,~,FC6] = csvread(filename);

QQ = transpose(FC6);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);


ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);
```

```
switch bands

    case handles.raw

        if get(hObject,'value')

            hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
            hold on;
            hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


            while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

                if get(hObject,'value')

                    set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
                    set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

                    drawnow;

                    xlabel('[sec]');
                    ylabel('µV');
                    grid on;

                    t = toc;

                    ind = round(t*fs);
                    ind = max(ind,1);

                else
                    break;
                end
            end
        end

    case handles.gamma          % (30-100Hz)

        W3 = 60/fs;
        W4 = 100/fs;
        Wn_a = [W3 W4];
        [g,h] = butter(n,Wn_a);
        gamma = filter(g,h,QQ);

        if get(hObject,'value')

            hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

            while (ind < N-nSamples) && ishandle(hLine)

                if get(hObject,'value')

                    set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
                    drawnow
                    title ('gamma (30-100Hz)');
                    xlabel('[sec]');
                    ylabel('µV');
                    grid on;

                    t = toc;

                    ind = round(t*fs);
                    ind = max(ind,1);

                else
                    break;
                end
            end
        end

    case handles.beta           % (12-30Hz)


        W5 = 24/fs;
        W6 = 60/fs;
```

```
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

       hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

       while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

             set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
             title ('beta (12-30Hz)');
             xlabel('[sec]');
             ylabel('µV');
             grid on;
             drawnow

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
       end
    end

case handles.alpha          % (8-12Hz)


    W3 = 16/fs;
    W4 = 24/fs;
    Wn_a = [W3 W4];
    [e,f] = butter(n,Wn_a);
    alpha = filter(e,f,QQ);

    if get(hObject,'value')

       hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

       while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

             set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
             title ('alpha (8-12Hz)')
             xlabel('[sec]');
             ylabel('µV');
             grid on;
             drawnow

             t = toc;

             ind = round(t*fs);
             ind = max(ind,1);

          else
             break;
          end
       end
    end

case handles.theta          % (4-7Hz)

    W1 = 8/fs;
    W2 = 14/fs;
    Wn_t = [W1 W2];
    [c,d] = butter(n,Wn_t);
    theta = filter(c,d,QQ);

    if get(hObject,'value')
```

```
        hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

            set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
            title ('theta (4-7Hz)');
            xlabel('[sec]');
            ylabel('µV');
            grid on;
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

          else
            break;
          end
        end
      end

   case handles.delta           %  (<4Hz)

      Wn_d = 8/fs;
      [b,a] = butter(n,Wn_d);
      delta = filter(b,a,QQ);
      delta = detrend (delta);

      if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

          if get(hObject,'value')

            set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
            title ('delta (<4Hz)');
            xlabel('[sec]');
            ylabel('µV');
            grid on;
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

          else
            break;
          end
        end
      end

end

 guidata(hObject, handles)

% --- Executes on button press in F4.
function F4_Callback(hObject, eventdata, handles)
% hObject    handle to F4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\F4.jpg');

filename = handles.filename;
[~,~,~,~,~,~,~,~,~,~,~,~,F4] = csvread(filename);
QQ = transpose(F4);
```

```
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);

ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

            if get(hObject,'value')

               set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
               set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

               drawnow;

               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

   case handles.gamma            % (30-100Hz)

      W3 = 60/fs;
      W4 = 100/fs;
      Wn_a = [W3 W4];
      [g,h] = butter(n,Wn_a);
      gamma = filter(g,h,QQ);

      if get(hObject,'value')

         hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

         while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

               set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
               drawnow
               title ('gamma (30-100Hz)');
               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;
```

```
            ind = round(t*fs);
            ind = max(ind,1);

        else
            break;
        end
    end
end

case handles.beta           % (12-30Hz)


    W5 = 24/fs;
    W6 = 60/fs;
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
                title ('beta (12-30Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.alpha          % (8-12Hz)


    W3 = 16/fs;
    W4 = 24/fs;
    Wn_a = [W3 W4];
    [e,f] = butter(n,Wn_a);
    alpha = filter(e,f,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
                title ('alpha (8-12Hz)')
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
```

```
                break;
            end
        end
    end

  case handles.theta          % (4-7Hz)

    W1 = 8/fs;
    W2 = 14/fs;
    Wn_t = [W1 W2];
    [c,d] = butter(n,Wn_t);
    theta = filter(c,d,QQ);

    if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

        if get(hObject,'value')

          set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
          title ('theta (4-7Hz)');
          xlabel('[sec]');
          ylabel('µV');
          grid on;
          drawnow

          t = toc;

          ind = round(t*fs);
          ind = max(ind,1);

        else
          break;
        end
      end
    end

  case handles.delta          %  (<4Hz)

    Wn_d = 8/fs;
    [b,a] = butter(n,Wn_d);
    delta = filter(b,a,QQ);
    delta = detrend (delta);

    if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

        if get(hObject,'value')

          set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
          title ('delta (<4Hz)');
          xlabel('[sec]');
          ylabel('µV');
          grid on;
          drawnow

          t = toc;

          ind = round(t*fs);
          ind = max(ind,1);

        else
          break;
        end
      end
    end

end

 guidata(hObject, handles)
```

```
% --- Executes on button press in F8.
function F8_Callback(hObject, eventdata, handles)
% hObject    handle to F8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\F8.jpg');

filename = handles.filename;
[~,~,~,~,~,~,~,~,~,~,~,~,F8] = csvread(filename);

QQ = transpose(F8);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);

ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

            if get(hObject,'value')

               set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
               set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

               drawnow;

               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

   case handles.gamma            % (30-100Hz)

      W3 = 60/fs;
      W4 = 100/fs;
      Wn_a = [W3 W4];
      [g,h] = butter(n,Wn_a);
      gamma = filter(g,h,QQ);
```

```
    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
                drawnow
                title ('gamma (30-100Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.beta           % (12-30Hz)


    W5 = 24/fs;
    W6 = 60/fs;
    Wn_b = [W5 W6];
    [g,h] = butter(n,Wn_b);
    beta = filter(g,h,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
                title ('beta (12-30Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.alpha          % (8-12Hz)


    W3 = 16/fs;
    W4 = 24/fs;
    Wn_a = [W3 W4];
    [e,f] = butter(n,Wn_a);
    alpha = filter(e,f,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)
```

```
            if get(hObject,'value')

                set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
                title ('alpha (8-12Hz)')
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.theta          % (4-7Hz)

    W1 = 8/fs;
    W2 = 14/fs;
    Wn_t = [W1 W2];
    [c,d] = butter(n,Wn_t);
    theta = filter(c,d,QQ);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
                title ('theta (4-7Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

case handles.delta          %  (<4Hz)

    Wn_d = 8/fs;
    [b,a] = butter(n,Wn_d);
    delta = filter(b,a,QQ);
    delta = detrend (delta);

    if get(hObject,'value')

        hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

        while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

                set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
                title ('delta (<4Hz)');
                xlabel('[sec]');
                ylabel('µV');
                grid on;
```

```
                drawnow

                t = toc;

                ind = round(t*fs);
                ind = max(ind,1);

            else
                break;
            end
        end
    end

end

 guidata(hObject, handles)

% --- Executes on button press in AF4.
function AF4_Callback(hObject, eventdata, handles)
% hObject    handle to AF4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



axes(handles.brain);
imshow('C:\Users\hunte\Downloads\FYP\nodes\AF4.jpg');

filename = handles.filename;
[~,~,~,~,~,~,~,~,~,~,~,~,~,~,AF4] = csvread(filename);
QQ = transpose(AF4);
fs = 128;
timeBase = 10;
hAx = handles.eeg;
N = length(QQ);
n = 4;
nSamples = round(fs*timeBase);
ind = 1;
y = detrend(QQ);
yy = tsmovavg(y,'s',10,2);

ylim auto;
tic;

bands = get(handles.uipanel1,'selectedobject');
axes(handles.eeg);

switch bands

   case handles.raw

      if get(hObject,'value')

         hLine1 = plot(hAx,(1:nSamples)/fs,y(:,ind:ind+nSamples-1));
         hold on;
         hLine2 = plot(hAx,(1:nSamples)/fs,yy(:,ind:ind+nSamples-1));


         while(ind < N-nSamples) && ishandle (hLine1) && ishandle (hLine2);

            if get(hObject,'value')

               set(hLine1,'ydata',y(:,ind:ind+nSamples-1));
               set(hLine2,'ydata',yy(:,ind:ind+nSamples-1));

               drawnow;

               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);
```

```
            else
               break;
            end
         end
      end

   case handles.gamma            % (30-100Hz)

      W3 = 60/fs;
      W4 = 100/fs;
      Wn_a = [W3 W4];
      [g,h] = butter(n,Wn_a);
      gamma = filter(g,h,QQ);

      if get(hObject,'value')

         hLine = plot(hAx,(1:nSamples)/fs,gamma(:,ind:ind+nSamples-1));

         while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

               set(hLine,'ydata',gamma(:,ind:ind+nSamples-1));
               drawnow
               title ('gamma (30-100Hz)');
               xlabel('[sec]');
               ylabel('µV');
               grid on;

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end

   case handles.beta             % (12-30Hz)


      W5 = 24/fs;
      W6 = 60/fs;
      Wn_b = [W5 W6];
      [g,h] = butter(n,Wn_b);
      beta = filter(g,h,QQ);

      if get(hObject,'value')

         hLine = plot(hAx,(1:nSamples)/fs,beta(:,ind:ind+nSamples-1));

         while (ind < N-nSamples) && ishandle(hLine)

            if get(hObject,'value')

               set(hLine,'ydata',beta(:,ind:ind+nSamples-1));
               title ('beta (12-30Hz)');
               xlabel('[sec]');
               ylabel('µV');
               grid on;
               drawnow

               t = toc;

               ind = round(t*fs);
               ind = max(ind,1);

            else
               break;
            end
         end
      end
```

```
case handles.alpha          % (8-12Hz)


   W3 = 16/fs;
   W4 = 24/fs;
   Wn_a = [W3 W4];
   [e,f] = butter(n,Wn_a);
   alpha = filter(e,f,QQ);

   if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,alpha(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

         if get(hObject,'value')

            set(hLine,'ydata',alpha(:,ind:ind+nSamples-1));
            title ('alpha (8-12Hz)')
            xlabel('[sec]');
            ylabel('µV');
            grid on;
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

         else
            break;
         end
      end
   end

case handles.theta          % (4-7Hz)

   W1 = 8/fs;
   W2 = 14/fs;
   Wn_t = [W1 W2];
   [c,d] = butter(n,Wn_t);
   theta = filter(c,d,QQ);

   if get(hObject,'value')

      hLine = plot(hAx,(1:nSamples)/fs,theta(:,ind:ind+nSamples-1));

      while (ind < N-nSamples) && ishandle(hLine)

         if get(hObject,'value')

            set(hLine,'ydata',theta(:,ind:ind+nSamples-1));
            title ('theta (4-7Hz)');
            xlabel('[sec]');
            ylabel('µV');
            grid on;
            drawnow

            t = toc;

            ind = round(t*fs);
            ind = max(ind,1);

         else
            break;
         end
      end
   end

case handles.delta          %  (<4Hz)

   Wn_d = 8/fs;
   [b,a] = butter(n,Wn_d);
   delta = filter(b,a,QQ);
```

```
        delta = detrend (delta);

        if get(hObject,'value')

           hLine = plot(hAx,(1:nSamples)/fs,delta(:,ind:ind+nSamples-1));

           while (ind < N-nSamples) && ishandle(hLine)

              if get(hObject,'value')

                 set(hLine,'ydata',delta(:,ind:ind+nSamples-1));
                 title ('delta (<4Hz)');
                 xlabel('[sec]');
                 ylabel('µV');
                 grid on;
                 drawnow

                 t = toc;

                 ind = round(t*fs);
                 ind = max(ind,1);

              else
                 break;
              end
           end
        end

end

 guidata(hObject, handles)

% --------------------------------------------------------------------
function uipushtool2_ClickedCallback(hObject, eventdata, handles)
% hObject    handle to uipushtool2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

cla(handles.eeg);

guidata(hObject, handles)
```