BME3500 – Report

Report Type: Homework Report
Name and Surname: Arif Selimhan Çetinkale
Student Number: 18017003

RESEARCH

EEG

EEG signals are generated by the electrical activity of neurons in the brain, and they are typically measured at the scalp using a set of electrodes placed on the head. The voltage fluctuations recorded by the electrodes are typically in the microvolt range, and they reflect the summated activity of large populations of neurons in the brain. EEG signals are time-varying and can be decomposed into different frequency bands, each of which is associated with specific brain states and activities. For example, alpha waves (8-12 Hz) are typically observed during relaxation and meditation, beta waves (13-30 Hz) are associated with cognitive tasks and alertness, and gamma waves (30-100 Hz) are involved in sensory processing and perception. EEG signals are also sensitive to various physiological and environmental factors, such as eye movements, muscle activity, and sleep stage. These factors can produce artifacts in the EEG data that can obscure the underlying brain activity and make it difficult to interpret the data. Therefore, it is important to carefully control for these factors during EEG recording and analysis. EEG signals can be analyzed using various techniques, such as time-frequency analysis, independent component analysis, and machine learning methods. These techniques can be used to extract meaningful information from the EEG data and to identify changes in brain activity associated with different cognitive states or pathological conditions. EEG signals have a wide range of applications, including research on brain function and cognition, diagnosis of neurological disorders, and brain-computer interface systems. They are also used in clinical settings to monitor brain activity during surgery and to assess brain damage in patients with head injuries.

EEG signal processing and Matlab

MATLAB is a widely used programming language and software platform that provides a range of tools for signal processing and analysis. It has a large number of built-in functions and libraries for EEG signal processing, such as the Signal Processing Toolbox and the EEGLAB toolbox. These tools allow users to perform a variety of tasks, including filtering, epoching, re-referencing, artifact removal, feature extraction, and classification. EEG signal processing in MATLAB typically involves several steps: Preprocessing: This step involves preparing the raw EEG data for further analysis. It may include filtering the data to remove unwanted frequencies, epoching the data into time segments, and re-referencing the data to adjust the reference point. Artifact removal: This step involves correcting for external sources of noise, such as eye movements, muscle activity, and electrical interference. There are several techniques that can be used for artifact removal, such as independent component analysis and regression-based methods. Feature extraction: This step involves extracting relevant features from the EEG data that can be used to characterize the underlying brain activity. There are several techniques that can be used for feature extraction, such as time-frequency analysis, independent component analysis, and wavelet decomposition. Classification: This step involves using the extracted features to classify the EEG data.

CODE

```
%----------------------------------------------------Q %1.1------
-----------------------------------------------------
% Loading data files
Q1_Data1 = load('Q1_Data1.mat');
Q1_Data2 = load('Q1_Data2.mat');
Q1_Data3 = load('Q1_Data3.mat');

% Extracting specification details for each dataset
sampling_rate_Q1_data_1 = Q1_Data1.EEG.srate % sampling rate
for Q1_Data1
num_channels_Q1_data_1 = Q1_Data1.EEG.nbchan % number of
channels for Q1_Data1
num_trials_Q1_data_1 = Q1_Data1.EEG.trials  % number of trials
for Q1_Data1
trial_length_Q1_data_1 = size(Q1_Data1.EEG.data, 2) %  /
sampling_rate1 ; % length of each trial in seconds for Q1_Data1

sampling_rate_Q1_data_2 = Q1_Data2.EEG.srate % sampling rate
for Q1_Data2
num_channels_Q1_data_2 = (Q1_Data2.EEG.nbchan) % number of
channels for Q1_Data2
num_trials_Q1_data_2 = (Q1_Data2.EEG.trials) % number of trials
for Q1_Data2
trial_length_Q1_data_2 = size(Q1_Data2.EEG.times, 2) % /
sampling_rate2; % length of each trial in seconds for Q1_Data2

sampling_rate_Q1_data_3 = Q1_Data3.EEG.srate % sampling rate
for Q1_Data3
num_channels_Q1_data_3 = Q1_Data3.EEG.nbchan % number of
channels for Q1_Data3
num_trials_Q1_data_3 =  Q1_Data3.EEG.trials % number of trials
for Q1_Data3
trial_length_Q1_data_3 = size(Q1_Data3.EEG.times, 2) % /
sampling_rate3; % length of each trial in seconds for Q1_Data3

%------------------------------------------------------Q 1.2-----
---------------------------------------------------------

% Extracting first channel for each dataset
Q1_Data1_channel1 = Q1_Data1.EEG.data(1, :, :); % first channel
for Q1_Data1
Q1_Data2_channel1 = Q1_Data2.EEG.data(1, :, :); % first channel
for Q1_Data2
Q1_Data3_channel1 = Q1_Data3.EEG.data(1, :, :); % first channel
for Q1_Data3
```

```
% Finding average ERP signal for each dataset using averaging
ERP_Q1_Data1 = mean(Q1_Data1_channel1, 3); % average ERP signal
for Q1_Data1
ERP_Q1_Data2 = mean(Q1_Data2_channel1, 3); % average ERP signal
for Q1_Data2
ERP_Q1_Data3 = mean(Q1_Data3_channel1, 3); % average ERP signal
hz2for Q1_Data3

% Seting time axis for plot
time_Q1_data_1 = 0:1/Q1_Data1.EEG.srate:(length(ERP_Q1_Data1)-
1)/Q1_Data1.EEG.srate; % time axis for Q1_Data1
time_Q1_data_2 = 0:1/Q1_Data2.EEG.srate:(length(ERP_Q1_Data2)-
1)/Q1_Data2.EEG.srate;
time_Q1_data_3 = 0:1/Q1_Data3.EEG.srate:(length(ERP_Q1_Data3)-
1)/Q1_Data3.EEG.srate;

hz_Q1_data_1=(0:1500-1)*500/1500;
hz_Q1_data_2=(0:1600-1)*400/1600;
hz_Q1_data_3=(0:2400-1)*600/2400;

% Ploting average ERP signal in time domain and power spectrum
of average ERP signal using subplot
figure
subplot(2,3,1)
plot(time_Q1_data_1, ERP_Q1_Data1)
xlabel('Time (s)')
ylabel('Amplitude (\muV)')
title('Average ERP Signal in Time Domain - Q1\_Data1')
hold on
plot(get(gca,'xlim'),[0 0],'k--')
plot([0 0],get(gca,'ylim'),'k--')
plot([0 0]+.5,get(gca,'ylim'),'k--')


subplot(2,3,4)
plot(hz_Q1_data_1, abs(fft(ERP_Q1_Data1)),"linew",2)
xlabel('Frequency (Hz)')
ylabel('Power')
title('Power Spectrum of Average ERP Signal - Q1\_Data1')

subplot(2,3,2)
plot(time_Q1_data_2, ERP_Q1_Data2)
xlabel('Time (s)')
ylabel('Amplitude (\muV)')
title('Average ERP Signal in Time Domain - Q1\_Data2')
hold on
plot(get(gca,'xlim'),[0 0],'k--')
plot([0 0],get(gca,'ylim'),'k--')
```

```
plot([0 0]+.5,get(gca,'ylim'),'k--')

subplot(2,3,5)
plot(hz_Q1_data_2, abs(fft(ERP_Q1_Data2)),"linew",2)
xlabel('Frequency (Hz)')
ylabel('Power')
title('Power Spectrum of Average ERP Signal - Q1\_Data2')

subplot(2,3,3)
plot(time_Q1_data_3, ERP_Q1_Data3)
xlabel('Time (s)')
ylabel('Amplitude (\muV)')
title('Average ERP Signal in Time Domain - Q1\_Data3')
hold on
plot(get(gca,'xlim'),[0 0],'k--')
plot([0 0],get(gca,'ylim'),'k--')
plot([0 0]+.5,get(gca,'ylim'),'k--')


subplot(2,3,6)
plot(hz_Q1_data_3, abs(fft(ERP_Q1_Data3)),"linew",2)
xlabel('Frequency (Hz)')
ylabel('Power')
title('Power Spectrum of Average ERP Signal - Q1\_Data3')


%-------------------------------------------------------Q2.1---
------------------------------------

% Loading data file
Q2_Data = load('Q2_Data.mat');
% Extracting first channel
Q2_Data_channel1 = Q2_Data.EEG.data(1, :, :); % first channel
for Q2_Data
% Finding average ERP signal using averaging
ERP_Q2_Data = mean(Q2_Data_channel1, 3); % average ERP signal
for Q2_Data
% Seting time axis for plot
time_Q2 = 0:1/Q2_Data.EEG.srate:(length(ERP_Q2_Data)-
1)/Q2_Data.EEG.srate; % time axis for Q2_Data
% Ploting average ERP signal in time domain
figure
plot(time_Q2, ERP_Q2_Data)
hold on
plot(get(gca,'xlim'),[0 0],'k--')
plot([0 0],get(gca,'ylim'),'k--')
plot([0 0]+.5,get(gca,'ylim'),'k--')
xlabel('Time (s)')
ylabel('Amplitude (\muV)')
```

```
title('Average ERP Signal in Time Domain - Q2\_Data (Channel
1)')

%-----------------------------------------------------------
Q2.2----------------------------------------


% Loading data file
Q2_Data = load('Q2_Data.mat');

% Extracting first channel
Q2_Data_channel1 = Q2_Data.EEG.data(1, :, :); % first channel
for Q2_Data
% Finding average ERP signal in time domain
ERP_Q2_Data = mean(Q2_Data_channel1, 3); % average ERP signal
in time domain
% Seting time axis for plot
time_Q2 = 0:1/Q2_Data.EEG.srate:(length(ERP_Q2_Data)-
1)/Q2_Data.EEG.srate; % time axis for Q2_Data
% Calculating power spectrum of first trial
trial1_fft_Q2 = fft(Q2_Data_channel1(:,:, 1)); % Fourier
Transform of first trial
trial1_power_spectrum_Q2 =
abs(trial1_fft_Q2).^2/length(trial1_fft_Q2); % power spectrum
of first trial

frequencyRepSeparate_Q2 =
fft(squeeze(Q2_Data_channel1(:,:,1)))/length(time_Q2);
powspectSeparate_Q2 =
mean((2*abs(frequencyRepSeparate_Q2)).^2,2);
% Calculating power spectrum of averaged trials in time domain
time_domain_average_fft_Q2 = fft(ERP_Q2_Data); % Fourier
Transform of average ERP signal in time domain
time_domain_average_power_spectrum_Q2 =
abs(time_domain_average_fft_Q2).^2/length(time_domain_average_f
ft_Q2); % power spectrum of average ERP signal in time domain
% Calculating power spectrum by averaging the Fourier
representations of individual trials
trials_fft_Q2 = fft(Q2_Data_channel1(:,:,:)); % Fourier
Transform of all trials      , [], 2
trials_power_spectrum_Q2 = mean(abs(trials_fft_Q2).^2,
3)/size(trials_fft_Q2, 2); % power spectrum by averaging the
Fourier representations of individual trials
% our hz axis
hz_Q2=(0:1200-1)*400/1200;
% Ploting powear spectrums
figure
subplot(3, 1, 1)
plot(hz_Q2, trial1_power_spectrum_Q2,"linew",2)
```

```
xlim([0 100])
title('Power Spectrum of First Trial')

subplot(3, 1, 2)
plot(hz_Q2, time_domain_average_power_spectrum_Q2,"linew",2)
xlim([0 100])
title('Power Spectrum of Averaged Trials in Time Domain')

subplot(3, 1, 3)
plot(hz_Q2, trials_power_spectrum_Q2,"linew",2)
xlim([0 100])
title('Power Spectrum Calculated by Averaging the Fourier
Representations of Individual Trials')

% Loading data file
%--------------------------------------------Q3.1--------------
--------------------------------
Q3_data = load('Q3_Data.mat');
sampling_rate = Q3_data.EEG.srate;
Q3_data_channels = Q3_data.EEG.data(1, :, :);
ERP_Q3_data = mean(Q3_data_channels,3);
tQ3 = 0:1/Q3_data.EEG.srate:(length(ERP_Q3_data)-
1)/Q3_data.EEG.srate;

figure
subplot(2,1,1)
plot(tQ3, ERP_Q3_data,"linew",1.25)
xlabel('Time (s)')
ylabel('Amplitude (\muV)')
title('Average ERP Signal in Time Domain - Q3/data')

[a,b] = xcorr(ERP_Q3_data,ERP_Q3_data); % Autocorrelation for
center frequency
subplot(2,1,2)
plot(b,a,"linew",1.25)
xlabel('Autocorelated f center spect')
ylabel('F value')
title('Centered Freq Spect')

Bandwidth = rms(ERP_Q3_data); % Root mean square for bandwith
Bandwidth

%-----------------------------------------------Q3.2-----------
-------------------------------------

trial1_fft_Q3_data = fft(Q3_data_channels(:,:, 1));
trial1_power_spectrum_Q3_data =
abs(trial1_fft_Q3_data).^2/length(trial1_fft_Q3_data);
```

```
frequencyRepSeparate_Q3_data =
fft(squeeze(Q3_data_channels(:,:,1)))/length(tQ3);
powspectSeparate_Q3_data =
mean((2*abs(frequencyRepSeparate_Q3_data)).^2,2);

time_domain_average_fft_Q3_data = fft(ERP_Q3_data);
time_domain_average_power_spectrum_Q3_data =
abs(time_domain_average_fft_Q3_data).^2/length(time_domain_aver
age_fft_Q3_data);

trials_fft_Q3_data = fft(Q3_data_channels(:,:,:));
trials_power_spectrum_Q3_data =
mean(abs(trials_fft_Q3_data).^2, 3)/size(trials_fft_Q3_data,
2);

hz_Q3_data=(0:1200-1)*400/1200;

figure

subplot(3, 1, 1)
plot(hz_Q3_data, trial1_power_spectrum_Q3_data,"linew",1.25)
xlim([0 100])
title('Power Spectrum of First Trial')

subplot(3, 1, 2)
plot(hz_Q3_data,
time_domain_average_power_spectrum_Q3_data,"linew",1.25)
xlim([0 100])
title('Power Spectrum of Averaged Trials in Time
Domain',"linew",1.25)

subplot(3, 1, 3)
plot(hz_Q3_data, trials_power_spectrum_Q3_data,"linew",1.25)
xlim([0 100])
title('Power Spectrum Calculated by Averaging the Fourier
Representations of First Trials')

%--------------------------------------------Q4.1--------------
-------------------------------

%loading data
Q4_Data = load('Q4_Data.mat');

% Taking first channel data
Q4_Data_channel1 = Q4_Data.EEG.data(1, :, :);

% ERP for each trials
ERP_Q4_Data = mean(Q4_Data_channel1, 3);

% Making time axis
time_Q4 = 0:1/Q4_Data.EEG.srate:(length(ERP_Q4_Data)-
1)/Q4_Data.EEG.srate;
```

```
% spectrogram f-graph making
figure
subplot(2,3,1)
spectrogram(ERP_Q4_Data, hamming(256), 240, 1000, 400);
colorbar;
xlabel('Time (s)')
ylabel('Frequency (Hz)')
title('Spectrogram of Average ERP Signal with Hamming Window')

subplot(2,3,2)
spectrogram(ERP_Q4_Data, hann(256), 240, 1000, 400);
colorbar;
xlabel('Time (s)')
ylabel('Frequency (Hz)')
title('Spectrogram of Average ERP Signal with Hann Window')


subplot(2,3,3)
spectrogram(ERP_Q4_Data, gausswin(256), 240, 1000, 400);
colorbar;
xlabel('Time (s)')
ylabel('Frequency (Hz)')
title('Spectrogram of Average ERP Signal with Gausian Window')


subplot(2,3,4)
spectrogram(ERP_Q4_Data, rectwin(256), 250, 1000, 400);
colorbar;
xlabel('Time (s)')
ylabel('Frequency (Hz)')
title('Spectrogram of Average ERP Signal with Rectangular
Window')

subplot(2,3,5)
spectrogram(ERP_Q4_Data, blackman(250), 240, 1000,400);
colorbar;
xlabel('Time (s)')
ylabel('Frequency (Hz)')
title('Spectrogram of Average ERP Signal with Blackman Window')


subplot(2,3,6)
plot(time_Q4, ERP_Q4_Data,"linew",1.25)
xlabel('Time (s)')
ylabel('Amplitude (\muV)')
title('Average ERP Signal in Time Domain - Q4')
hold on
plot(time_Q4, Q4_Data_channel1(1,:,1))
```

```
hz_Q4=(0:1200-1)*400/1200;

[~, time_idx] = max(abs(ERP_Q4_Data)); % Find the index of the time point with the maximum absolute value
time_point_Q4 = time_Q4(time_idx) % Convert the index to the time point in seconds

%finding the peak frequency of the transient activity, you can use the following code:

[~, frequency_idx_Q4] = max(abs(ERP_Q4_Data)); % Find the index of the frequency with the maximum absolute value
peak_frequency_Q4 = hz_Q4(frequency_idx_Q4) % Convert the index to the frequency in Hz

%finding the bandwidth of the transient activity, you can use the following code:

[~, frequency_idx_Q4] = max(abs(ERP_Q4_Data)); % Find the index of the frequency with the maximum absolute value
bandwidth_Q4 = hz_Q4(frequency_idx_Q4) - hz_Q4(1) % Calculate the bandwidth by subtracting the lowest frequency from the peak frequency
```

COMMENTS ABOUT THE CODE

At the beginning of the code sections prepared separately for each question, we first load our data and make the necessary definitions. Specifically for the first question, we wanted the output of information such as sampling frequency, number of channels, number of trials and data length. Then we applied the ERP analysis mean function for the first channel of all three data and plotted the signal. We determined our intervals such as time and frequency. Then, we applied power spectrum analysis to the signals with fourier transform. and we drew the outputs.

In the second question, we made the ERP analysis of the first channel of the signal. For the first attempt, we took fft and drew the power spectrum, adjusting the range of 1-100. Afterwards, we adjusted and plotted the average power spectrum in the frequency range of 1-100. Then we drew the averaged power spectrum in the frequency range of 1-100. For question three, we plotted the averaged erp signal as in 2. We plotted the power spectrum of the first trial of channel 1, the power spectrum of the mean of the trials, and the power spectrum of the mean of the fourier of the trials in the 1-100 frequency domain. for the last one we also plotted the center frequencies and then suppressed the bandwidth. For the 4th question, we plotted the erp signal of channel 1 and the spectrogram of the power spectrum, using the spectrogram function with different window methods.

RESULTS

Q1.1

```
sampling_rate_Q1_data_1 =
500

num_channels_Q1_data_1 =
4

num_trials_Q1_data_1 =
40

trial_length_Q1_data_1 =
1500

sampling_rate_Q1_data_2 =
400

num_channels_Q1_data_2 =
5

num_trials_Q1_data_2 =
30

trial_length_Q1_data_2 =
1600

sampling_rate_Q1_data_3 =
600

num_channels_Q1_data_3 =
3

num_trials_Q1_data_3 =
60

trial_length_Q1_data_3 =
2400
```

Figure 1 – Question 1.1

Q1.2

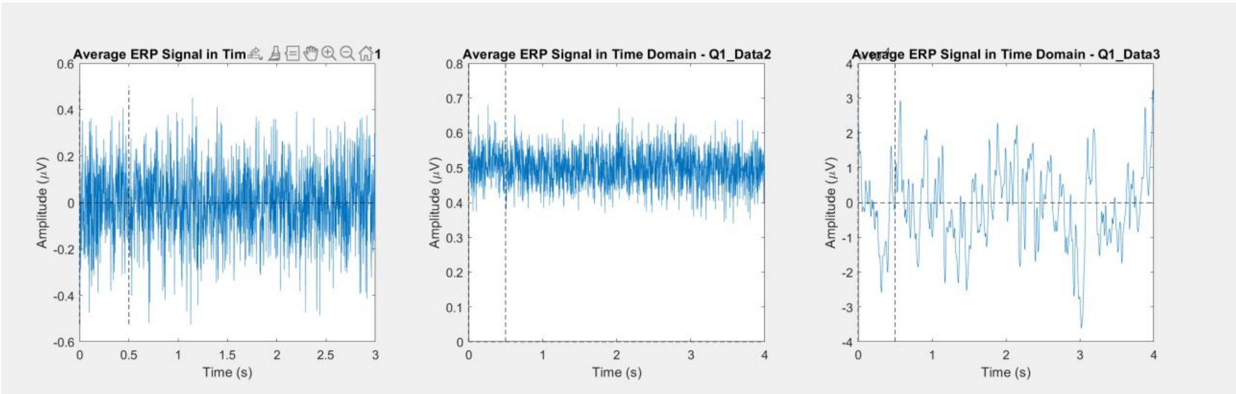

Figure 2 – Question 1.2

Q1.3

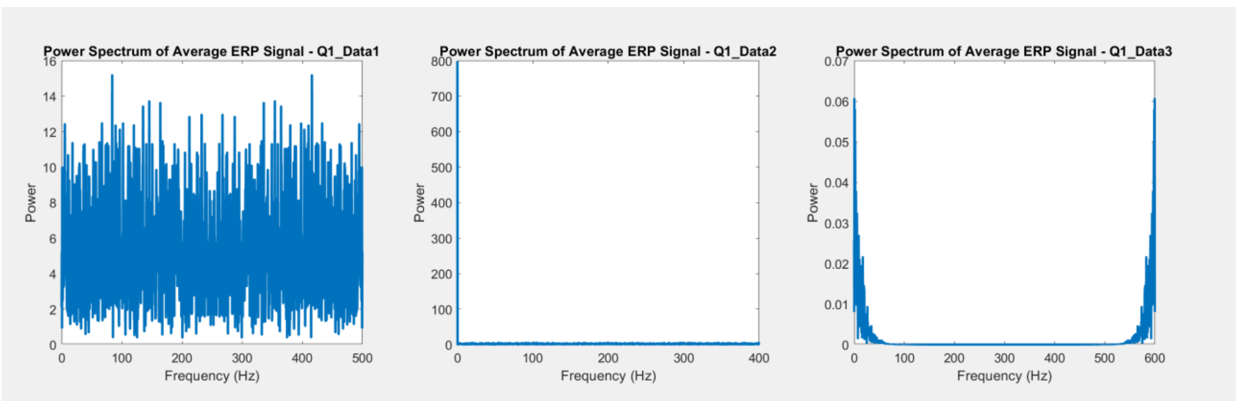

Figure 3 – Question 1.3

1 Uniformly Distributed   2 White Gaussian Noise   3 Pink noise

Q2.1

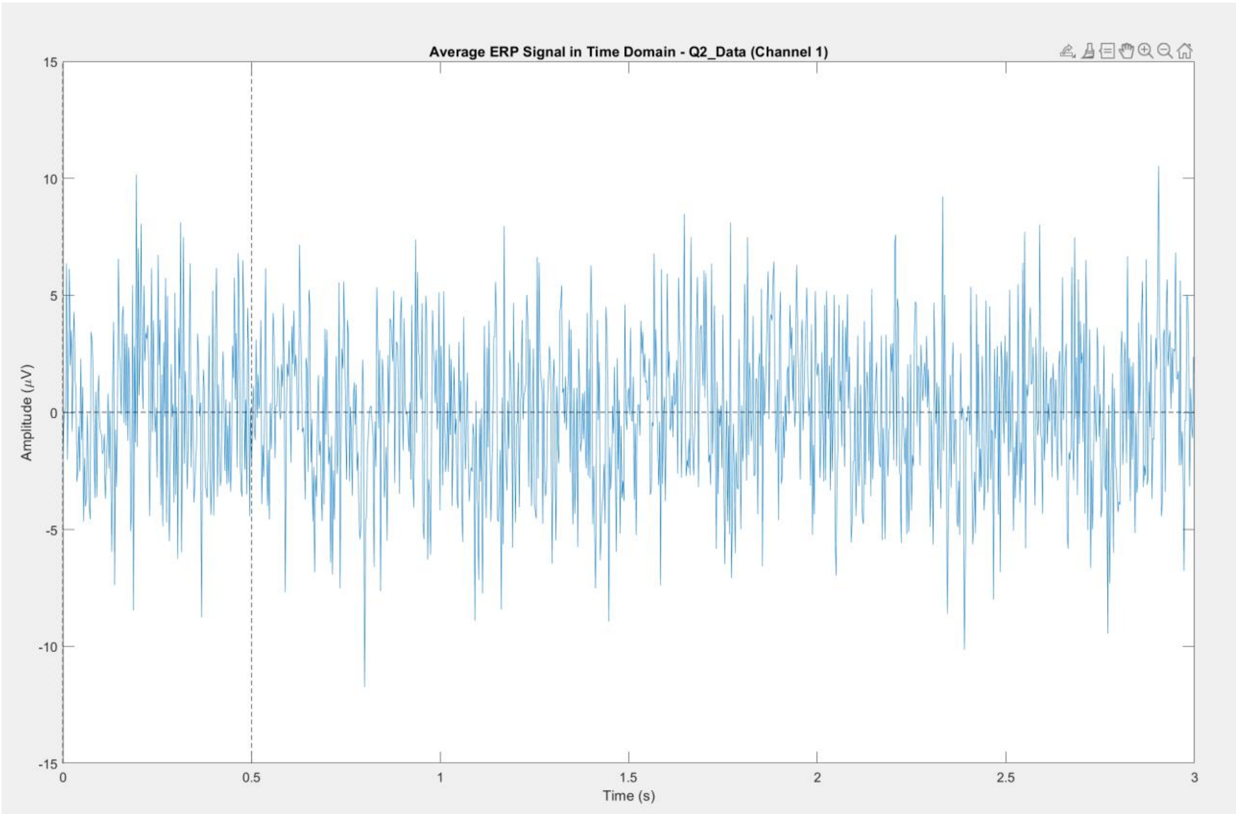

Figure 4 – Question 2.1

Q2.2

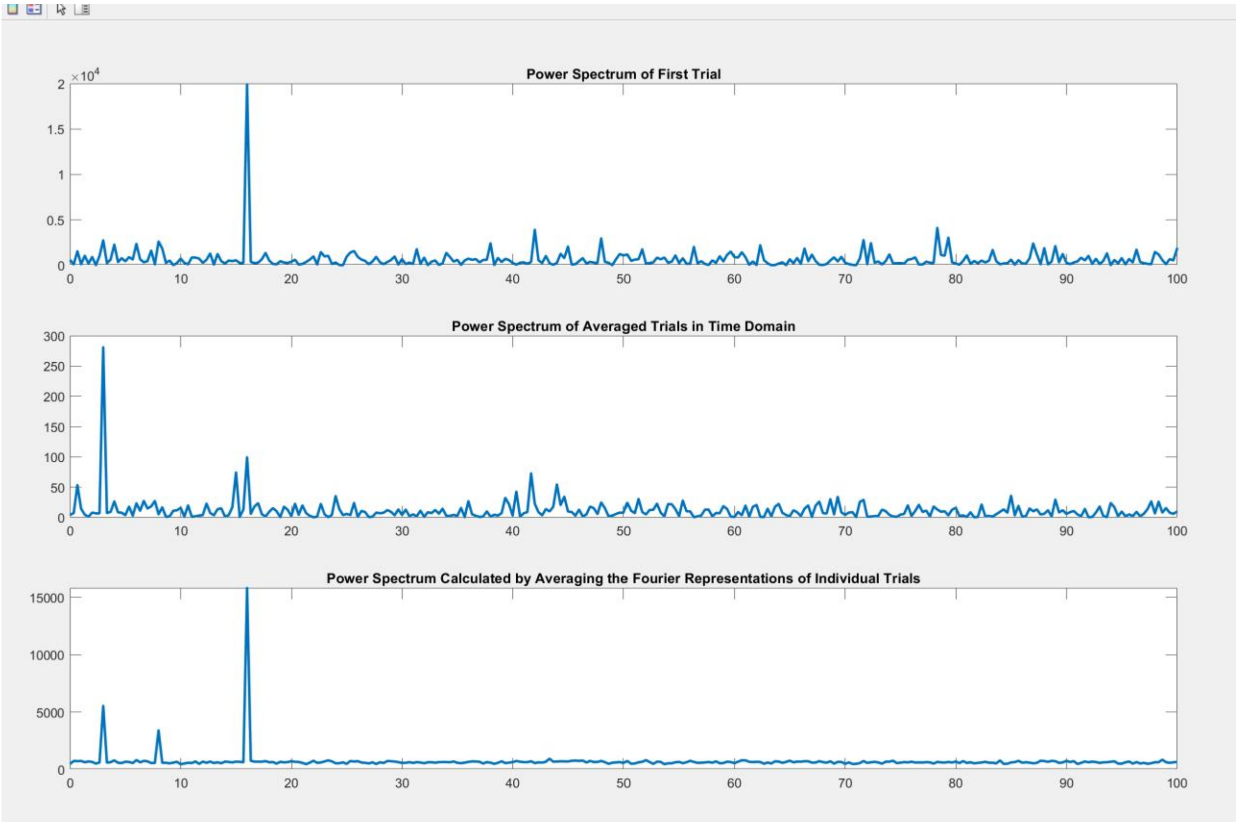

Figure 5 – Question 2.2

Q3.1

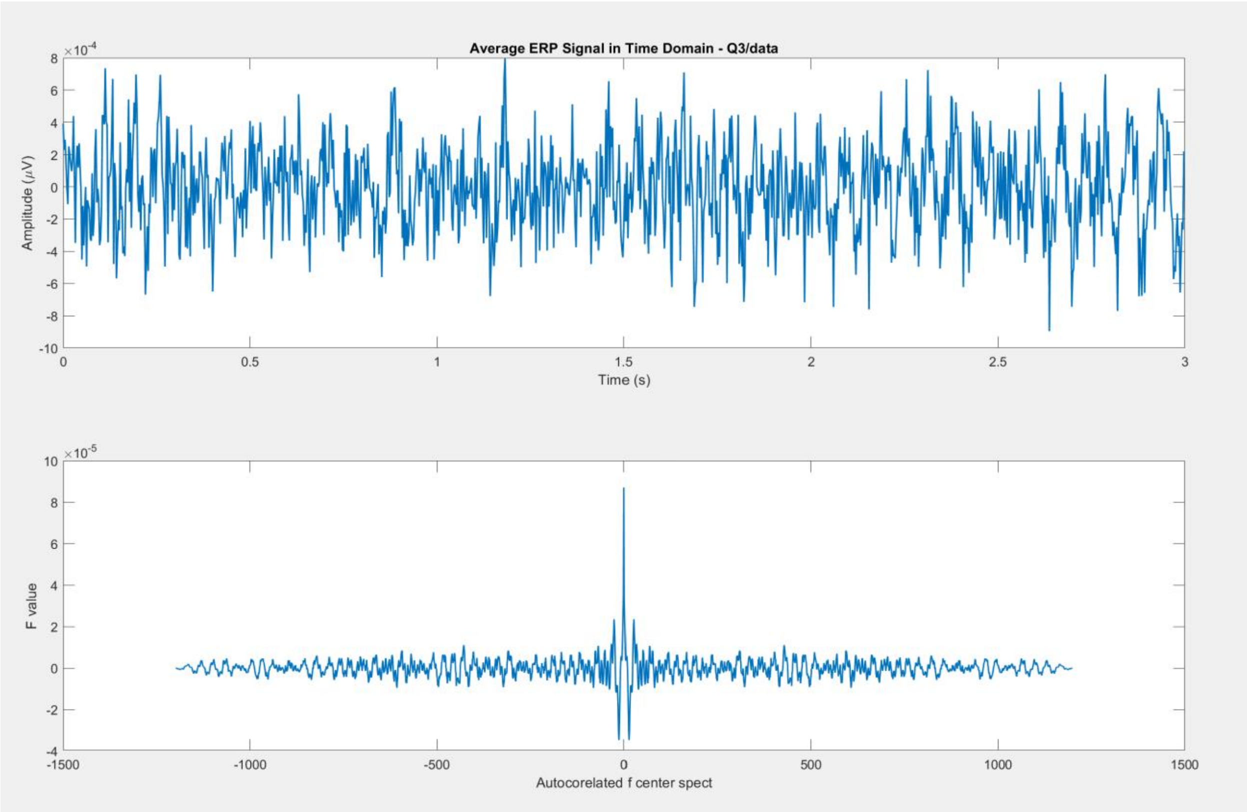

Figure 6 – Question 3.1

Q3.2

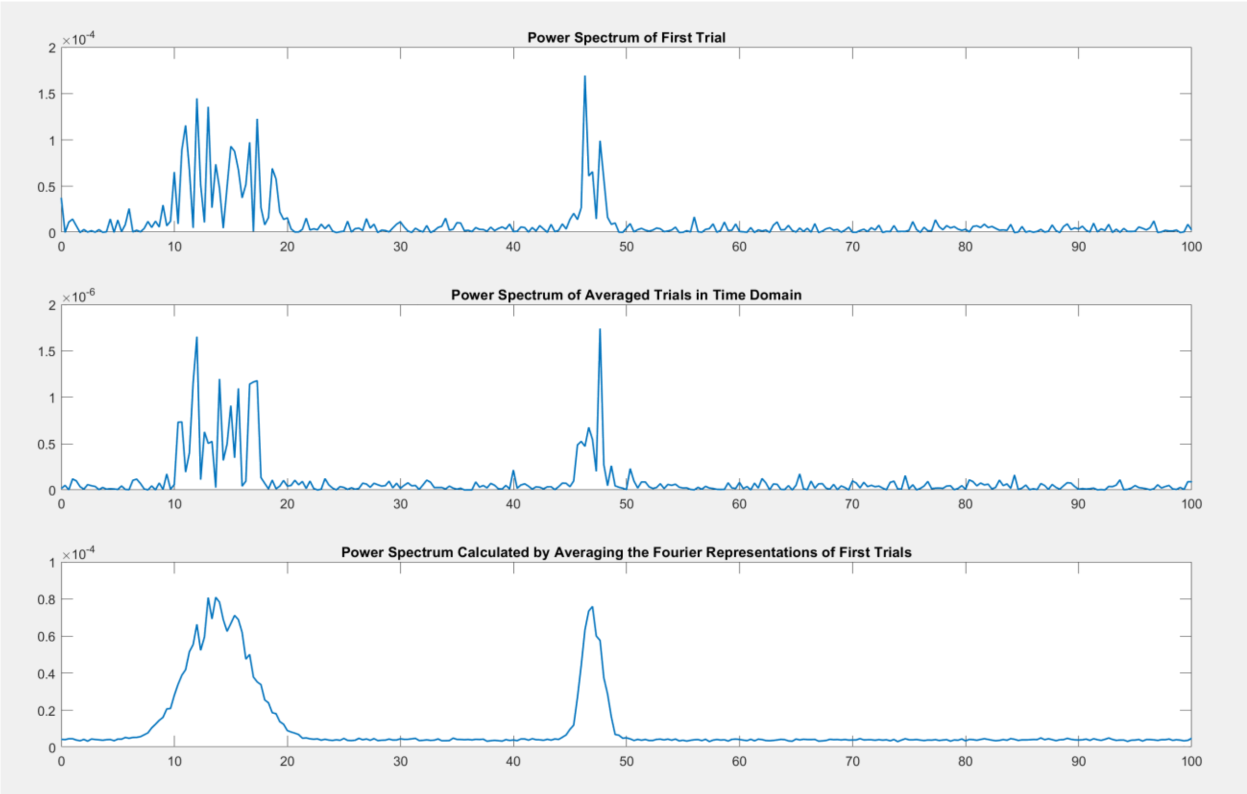

Figure 7 – Question 3.2

Q4.1

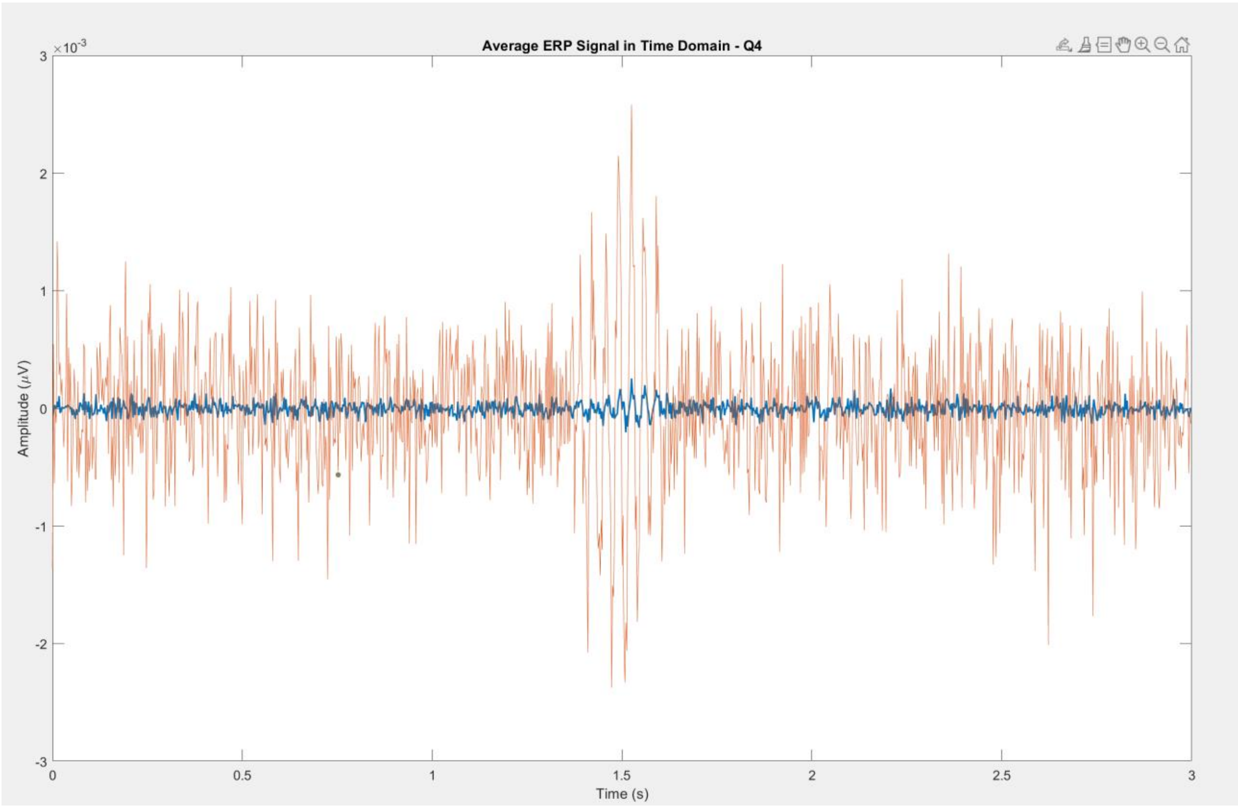

Figure 8 – Question 4.1

First channel trials of data 4 as a comparing results and blue one as a ERP

Q4.2

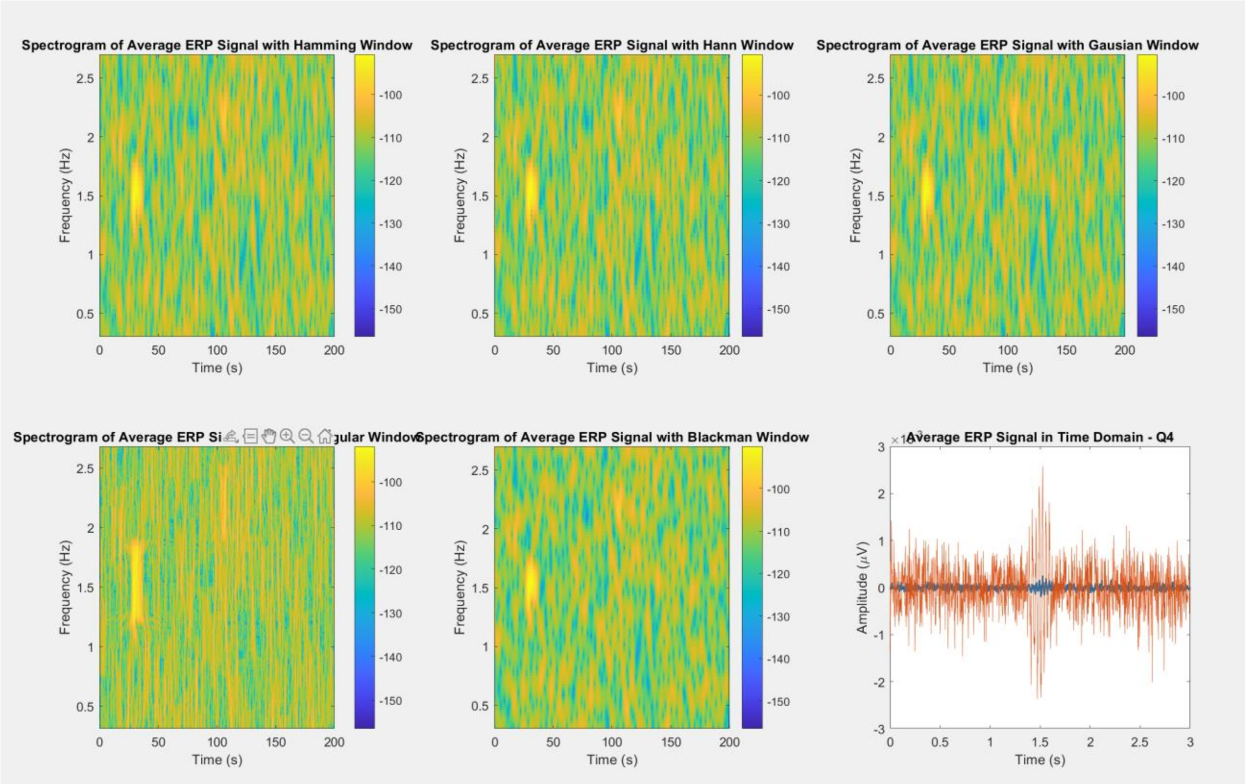

Figure 9 – Question 4.2

Q4.3

```
time_point_Q4 =

    1.5250


peak_frequency_Q4 =

  203.3333


bandwidth_Q4 =

  203.3333
```

Figure 10 – Question 4.3

COMMENTS ABOUT THE RESULTS

In figure 1, we can see the desired results in the first part of the first question, in the form of sampling rate, number of channels, number of trials and length for each data. In figure 2, we observe the output of the average erp signals. In the third figure, there are power spectra, these spectra give us information about the frequencies of the signals in our signal. Since our signals contain different noises, it is normal to see different frequency responses. If we need to explain them, it is possible to choose our response frequencies of our uniformly distributed signal in our first graph. Our 2nd graph is a straight line, which means it contains white noise. Third, we have a frequency response that decreases at certain intervals, which allows us to say that it contains pink noise. In our 4th figure, we can observe the ERP of the signals in the first channel of our Q2 signal. In figure 5, we can examine the power spectrum of the first experiment in the first plot, the average power spectrum of the trials in the second, and the power spectrum obtained by averaging the fourier's in the third plot. The power spectra in these graphs can help us understand patterns of brain activity about the frequency content of trials in general, and other characteristics of the signals. For example, as we saw in the first plot, we can see an intense signal activity between 10-20. this deviation is also visible in the graphs when we take the average, we can say that there is a signal spectrum in the 0-20 hz band in general, which gives information to comment on various activities. Maybe we can call it alpha beta waves. In our 6th figure, we see an ERP signal and the central frequency section. In the other graph, we can observe what value our center frequency has reached. The center frequency is the frequency at which the signal has the most power. Often used as a measure of the dominant frequency of the signal, power can be obtained from the psectrum. It can be useful for identifying broadband or narrowband signals and determining the relative strength of different frequency components within the signal. In general, the center frequency of an EEG signal can provide valuable information about the dominant frequency of the signal and can be useful for understanding brain activity patterns and other characteristics of the signal. In our 7th figure, we have a frequency graph obtained from the power spectra. From the graph of the first experiment, we see that there is a movement between 10-20 and an explosion between 45-50. When we take the averages, we can observe that there is a movement in these intervals for each signal in the same way. In our 8th figure, we see the first trial of the ERP and the first channel in the same way, an image that can be useful for comparison. When the average is taken, we can see how much the signal is damped and the active band is also evident even in the average. Finally, when we come to the spectrograms, we have 5 differently windowed spectrograms that allow us to visually examine at which frequencies our signal is intense. here we used hamming hann gausian blackman and rectangular windows. each presented its unique spectrogram to us. For example, we can see low signals more commonly in a rectangular chart. in others it seems less so. we can observe that higher frequencies and resolution are better in other windowings. While making these graphics, we played with the intervals and overlap values of the windows. When we reduced the values, we encountered more distorted images and incomprehensible expressions, so the values were kept high.

Referances

Berry RB, Brooks R, Gamaldo CE, et al. for the American Academy of Sleep Medicine. The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications. Darien, IL: American Academy of Sleep Medicine; 2017. Version 2.4

Alvarez-Estevez D, Rijsman RM (2021) Inter-database validation of a deep learning approach for automatic sleep scoring. PLoS ONE 16(8): e0256111.

Alvarez-Estevez, D., & Fernández-Varela, I. (2020). Addressing database variability in learning from medical data: An ensemble-based approach using convolutional neural networks and a case of study applied to automatic sleep scoring. Computers in Biology and Medicine, 119, 103697. doi:10.1016/j.compbiomed.2020.103697

F. Renna, J. H. Oliveira, and M. T. Coimbra, "Deep convolutional neural networks for heart sound segmentation," IEEE Journal of Biomedical and Health Informatics, vol. 23, no. 6, pp. 2435–2445, 2019.

Xie, C., McCullum, L., Johnson, A., Pollard, T., Gow, B., & Moody, B. (2021). Waveform Database Software Package (WFDB) for Python (version 3.3.0). PhysioNet. https://doi.org/10.13026/g35g-c061