Laboratory of signal processing and medical images

Report Lab 4

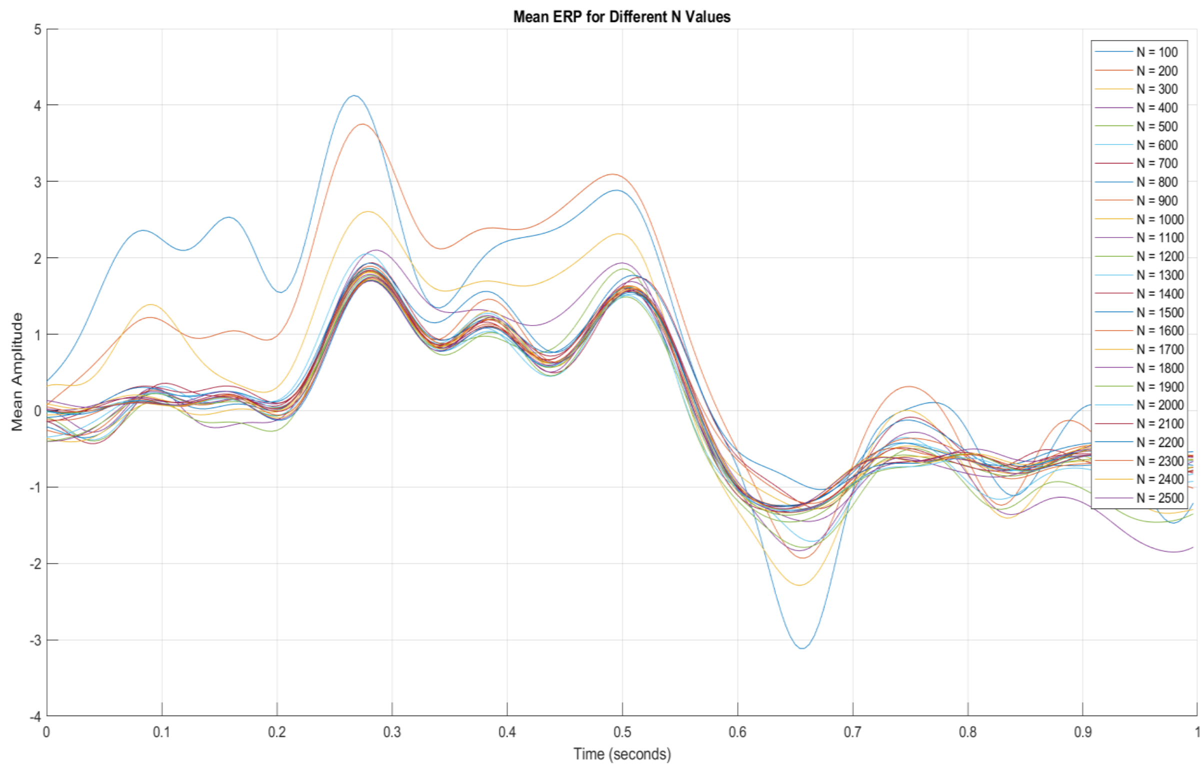
Sara Rezanejad – 99101643

MohamadHosein Faramarzi - 99104095

Ali Khosravipour – 99101502

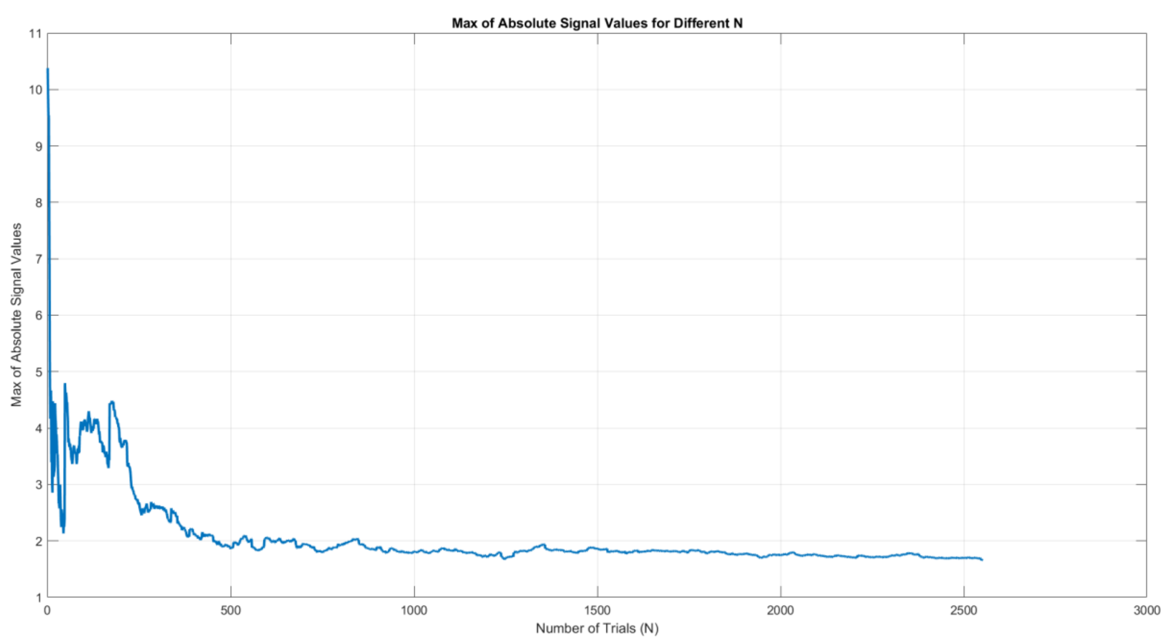
Part 1

1.

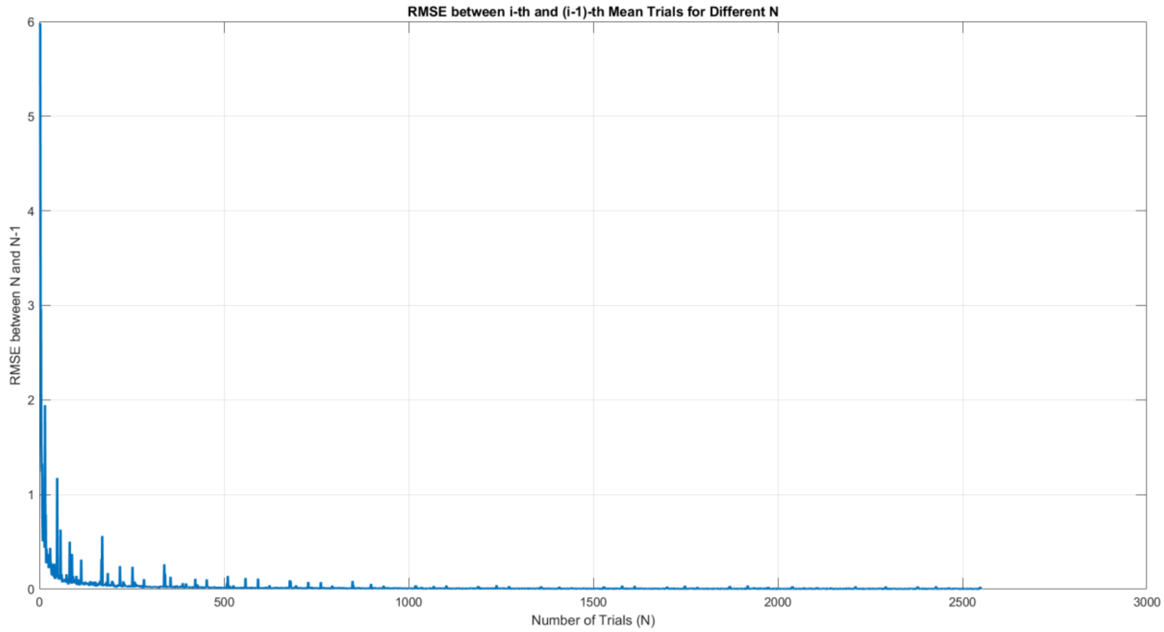


As N increases, graphs converge to the same shape.

2.



3.



The RMSE can be defined mathematically as:

$$\mathrm{RMSE}_{N-1} = \sqrt{\frac{1}{1}\sum_{i=1}^{1} (\mu_c - \mu_p)^2}$$

Since the means are single values (not arrays), the summation simplifies to just the squared difference between the two means. Therefore, the formula for RMSE can be simplified to:

$$\mathrm{RMSE}_{N-1} = \sqrt{(\mu_c - \mu_p)^2}$$

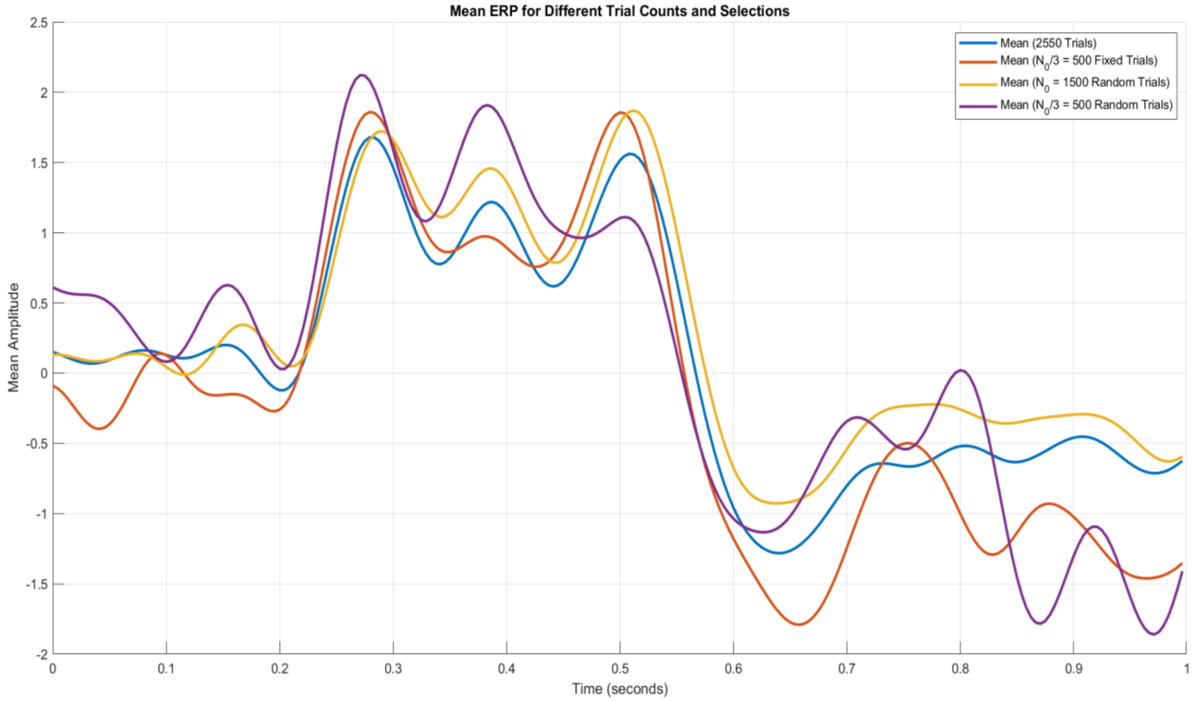
This further simplifies to:

$$\mathrm{RMSE}_{N-1} = |\mu_c - \mu_p|$$

4.

According to the obtained 3 graphs, it can be seen that for $N_0 = 500$, the graph converges somewhat, and from this value onwards, averaging in more samples does not affect the result noticeably.

5.



As it is known, the graph does not differ significantly for larger Ns. According to the choice of random N, which here is 1500 and its one-third value can be seen in its one-third value, the peaks cannot be separated well.

6.

To investigate real experiments based on the 300P signal (especially in the field of brain-computer interfaces), it is necessary to carefully analyze the measured parameters and beams.

The number of repetitions of the pattern is 300P

1. Number of replicates: In 300P-based experiments, usually the number of replicates of the 300P pattern depends on the type of study and its specific purpose. In general, to ensure the accuracy and reduce the error in

the calculations, the number of iterations can even reach several hundreds to more than a thousand. For example, in some experiments the number of iterations may reach 1000 or even 2000 to ensure that a proper and meaningful averaging of the signals is performed.

2. Compatibility with previous results: Our previous results suggest that for $N_0 = 500$, the graph converges somewhat, and beyond this value, averaging over more samples does not significantly affect the result. This shows that in some cases, high iterations can be useful, but it reaches a point where it has less effect on the final results.

Explanation of the differences

The difference between the number of repetitions used in the actual experiments and the results obtained may depend on several factors:

1. Data Quality: In real data, there is noise and various disturbances that can affect the results. Therefore, the number of repetitions may need to be higher to ensure the accuracy of the results.

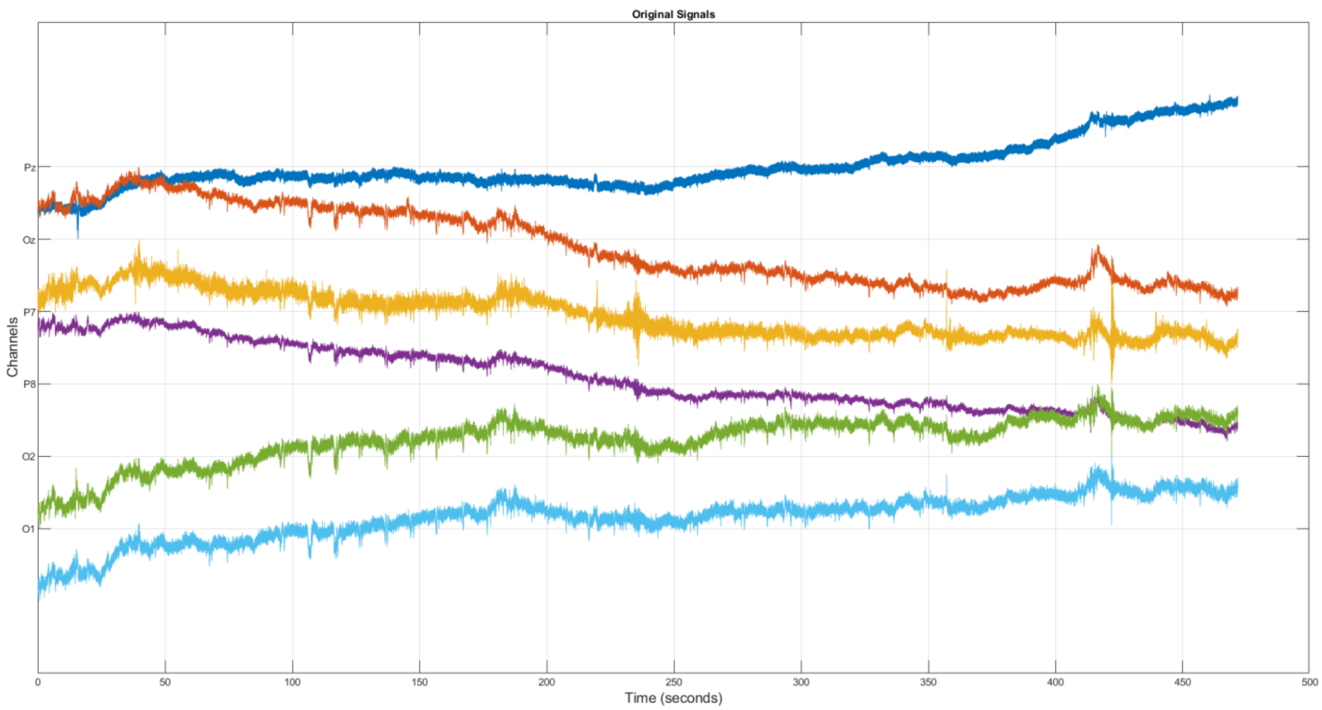
2. Type of Analysis: The type of analysis performed (eg, frequency or time analysis) also affects the number of iterations. In some analyses, more replications may be necessary to detect more precise and valid patterns.

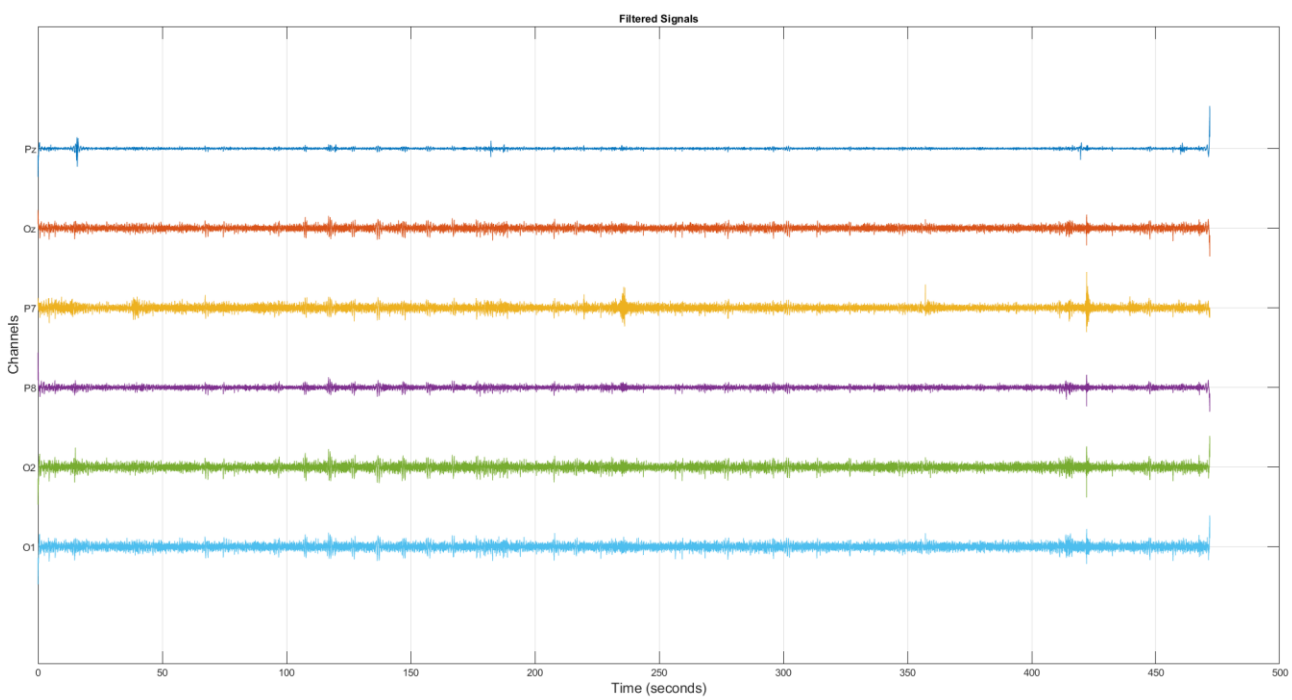
3. pattern dimensions: In actual experiments with larger and more complex patterns, the number of iterations

may increase. For the 300P signal, more systems and more time are required to obtain accurate results.

Part 1

1.

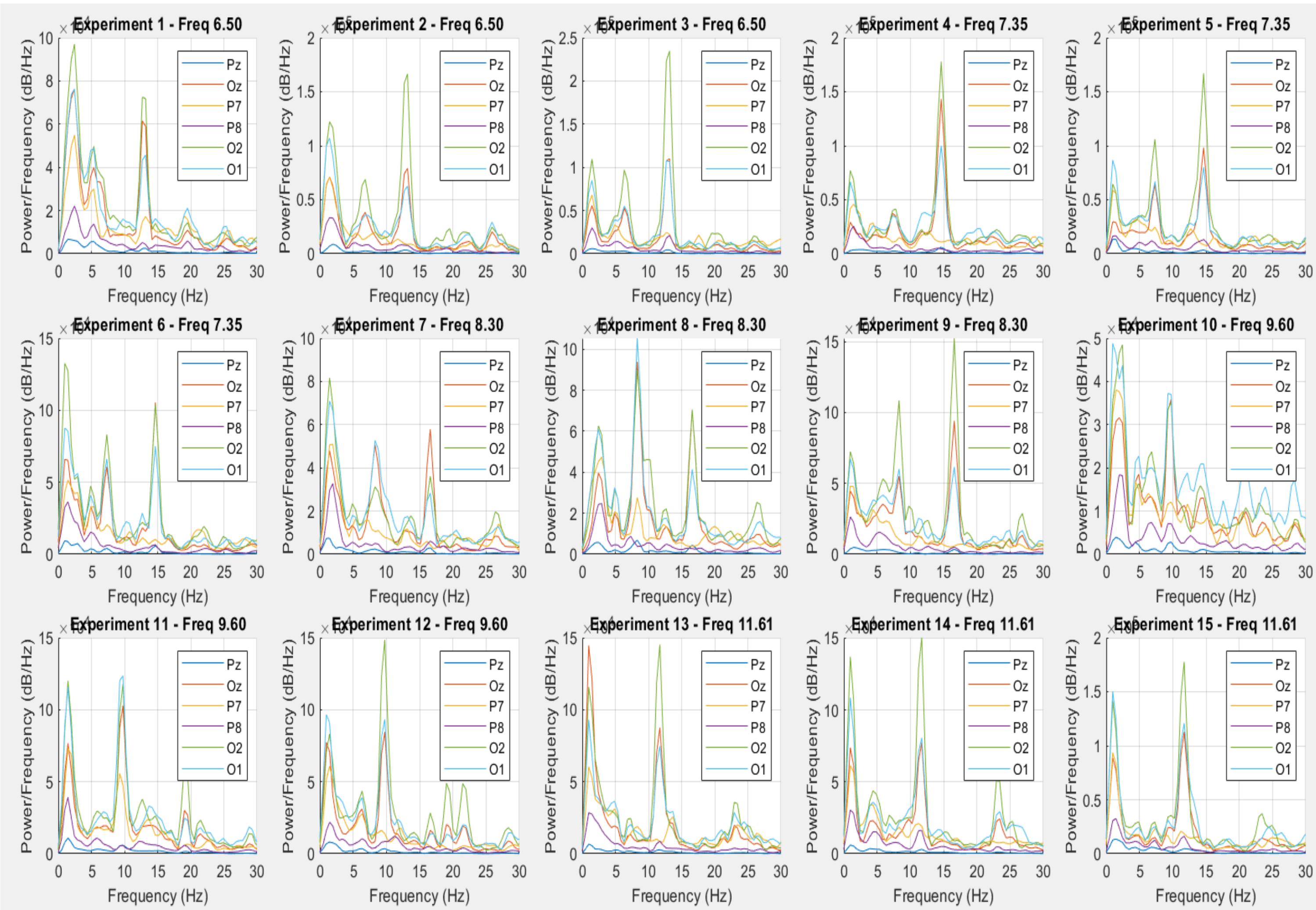




2.

```
%% 2.2 Window Extraction
winLength = 5 * fs;
eventWindows = [];
for i = 1 : size(Event_samples,2)
    start_idx = Event_samples(i);
    event_win = ssvep_filtered(:, start_idx:start_idx + winLength - 1);
    eventWindows = cat(3, eventWindows, event_win);
end
```

3.



4.

For a test based on SSVEP (spontaneous evoked potentials), the frequency content of the different channels is usually not the same. These differences may be due to the following factors:

1. Electrode positions: The position of the channels on the skull creates certain differences in receiving brain signals. For example, different brain activities may be stronger or weaker in different channels, depending on the region being stimulated.
2. Individual characteristics: Each person has specific brain patterns and characteristics that can affect the way the brain responds to stimulation.
3. Noise and Interference: Additional electrical signals and environmental noise may also affect different channels and lead to differences in frequency content.

5.

Certainly, observing the patterns indicates that there is a prominent peak where greater energy is concentrated. The peak frequencies are calculated as follows:

$$0.1 \times \frac{250}{2} = 12.5 \text{ Hz},\ 0.4 \times \frac{250}{2} = 50 \text{ Hz},\ 0.8 \times \frac{250}{2} = 100 \text{ Hz}$$

The dominant frequency corresponds to 50 Hz. This frequency is related to the electrical grid of the city, specifically 12.5 Hz associated with restful states. Frequencies around 100 Hz can be linked to concentration or mental strain, while the peak associated with these contexts implies specific neural activities.

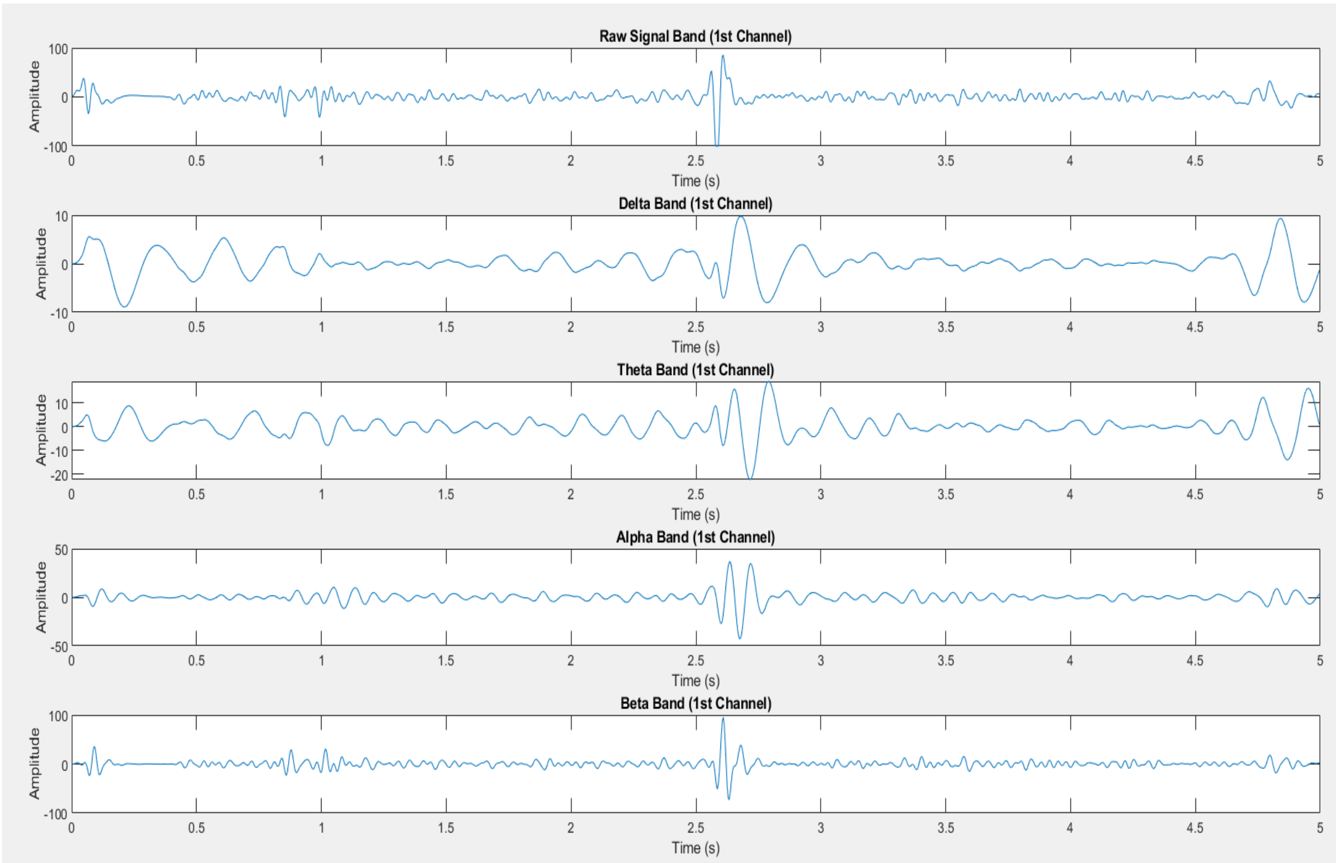
6.

There are several other methods for determining the dominant frequency in SSVEP signals that may give better results than using the frequency content of each channel:

1. Fourier Transform: This method is used to analyze signals and identify different frequencies.
2. Wavelet Transform analysis method: This method helps to analyze the signal in time and frequency and can help to better analyze the frequency fluctuations in different time bands.
3. Machine learning methods: Machine learning algorithms can identify complex patterns and identify which dominant frequency is evident in SSVEP signals.
4. Correlation matrix analysis: Correlation between channels can be used to identify specific frequency patterns.

Part 3

1.



2.

```
%% Part 3.2
seg_delta = zeros(200,30,2560);
seg_theta = zeros(200,30,2560);
seg_alpha = zeros(200,30,2560);
seg_beta = zeros(200,30,2560);
for i=1:200
    seg_delta(i,:,:) = filtered_delta(:,trial(i): (trial(i) + 2560) - 1);
    seg_theta(i,:,:) = filtered_theta(:,trial(i): (trial(i) + 2560) - 1);
    seg_alpha(i,:,:) = filtered_alpha(:,trial(i): (trial(i) + 2560) - 1);
    seg_beta(i,:,:) = filtered_beta(:,trial(i): (trial(i) + 2560) - 1);
end
```

This part of the code efficiently organizes filtered signal data into structured segments corresponding to different frequency bands and trials.

3.

```
%% Part 3.3
squared_data_delta = seg_delta.^2;
squared_data_theta = seg_theta.^2;
squared_data_alpha = seg_alpha.^2;
squared_data_beta = seg_beta.^2;
```

4.

This part of our code organizes data into different classes, calculates the mean for each class, and then aggregates those means based on the current iteration of the outer loop.

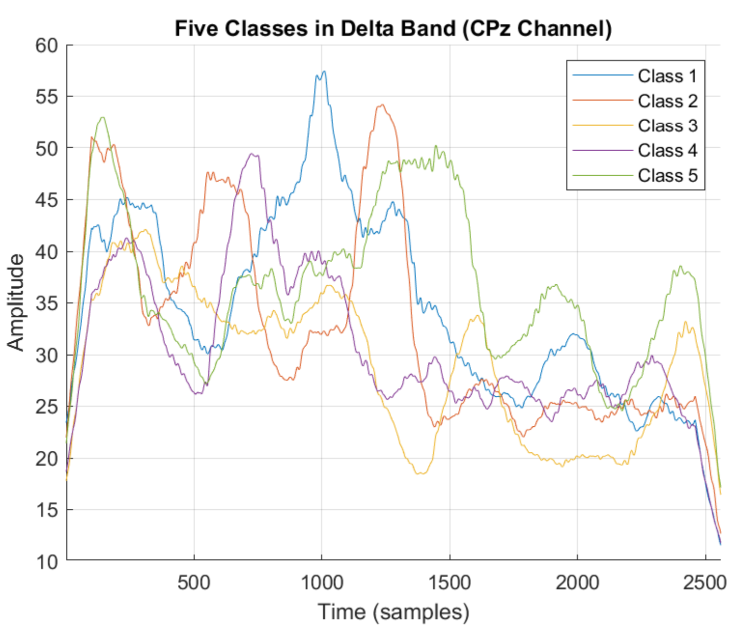
```
all_data = {squared_data_delta, squared_data_theta, squared_data_alpha, squared_data_beta};
for j=1:length(all_data)
    squared_data = all_data{j};
    for i = 1:200
        if y(i) == 1
            class1 = cat(1, class1, squared_data(i, :, :));
        elseif y(i) == 2
            class2 = cat(1, class2, squared_data(i, :, :));
        elseif y(i) == 3
            class3 = cat(1, class3, squared_data(i, :, :));
        elseif y(i) == 4
            class4 = cat(1, class4, squared_data(i, :, :));
        else
            class5 = cat(1, class5, squared_data(i, :, :));
        end
    end
    mean1 = mean(class1(:,:,:));
    mean2 = mean(class2(:,:,:));
    mean3 = mean(class3(:,:,:));
    mean4 = mean(class4(:,:,:));
    mean5 = mean(class5(:,:,:));
    if j==1
        Delta_X_avg = cat(1, mean1, mean2, mean3, mean4, mean5);
    elseif j==2
        Theta_X_avg = cat(1, mean1, mean2, mean3, mean4, mean5);
    elseif j==3
        Alpha_X_avg = cat(1, mean1, mean2, mean3, mean4, mean5);
    else
        Beta_X_avg = cat(1, mean1, mean2, mean3, mean4, mean5);
    end
end
```

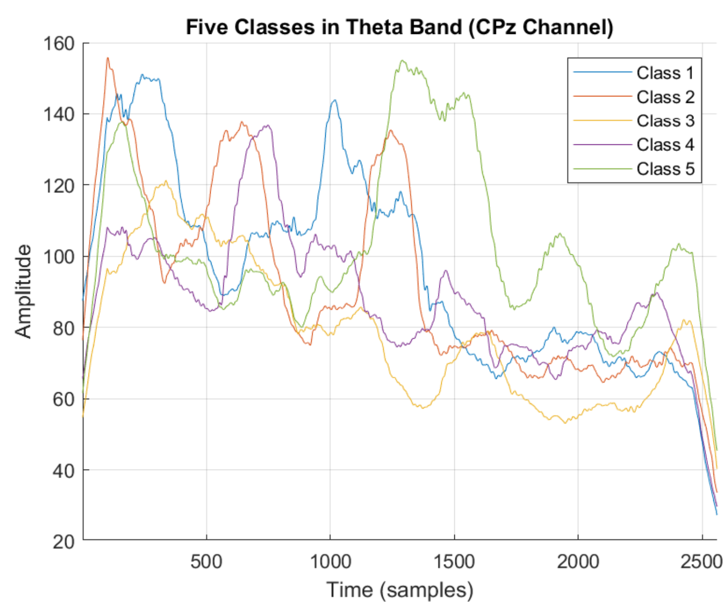
5.

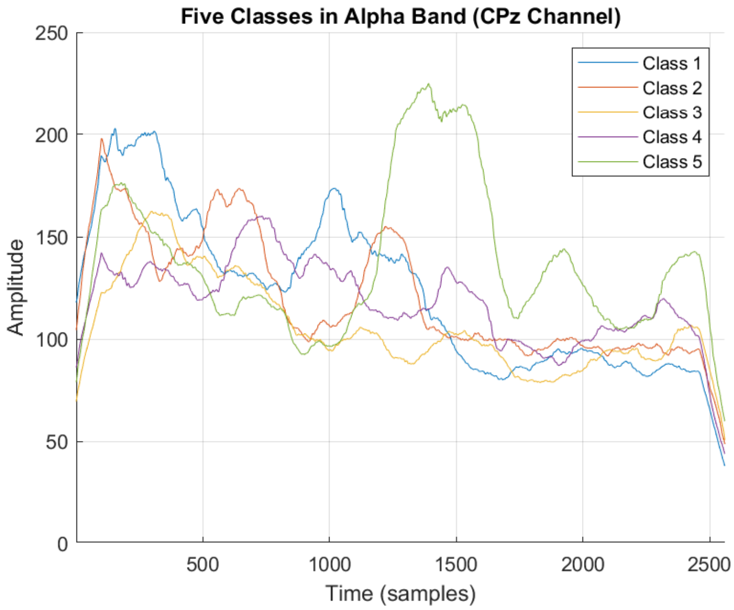
The overall goal of this part of our code is to smooth the average signal data (for different classes and frequency bands) by convolving it with a defined window.

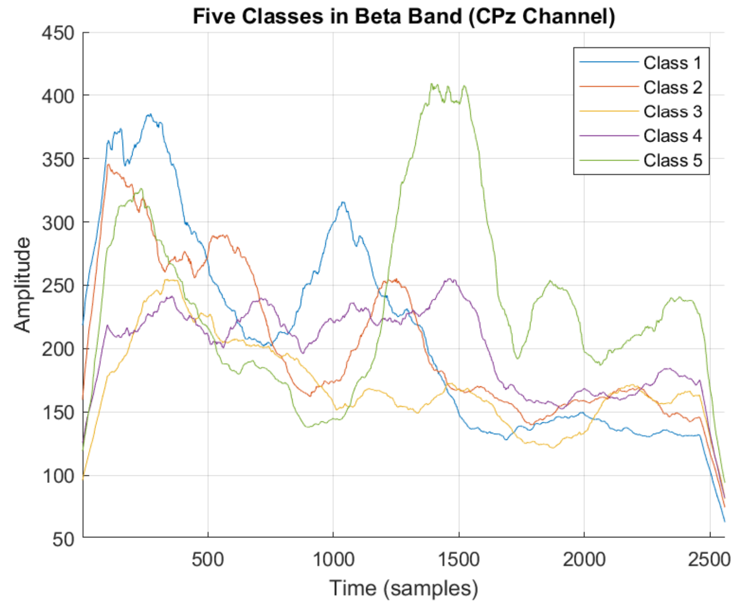
```
%% Part 3.5
newWin = ones(1,200)/sqrt(200);
Delta_X_avg_conv = zeros(size(Delta_X_avg));
Theta_X_avg_conv = zeros(size(Theta_X_avg));
Alpha_X_avg_conv = zeros(size(Alpha_X_avg));
Beta_X_avg_conv = zeros(size(Beta_X_avg));
for class = 1:size(Delta_X_avg, 1)
    for channel = 1:size(Delta_X_avg, 2)
        Delta_X_avg_conv(class, channel, :) = conv(squeeze(Delta_X_avg(class, channel, :)), newWin, 'same');
        Theta_X_avg_conv(class, channel, :) = conv(squeeze(Theta_X_avg(class, channel, :)), newWin, 'same');
        Alpha_X_avg_conv(class, channel, :) = conv(squeeze(Alpha_X_avg(class, channel, :)), newWin, 'same');
        Beta_X_avg_conv(class, channel, :) = conv(squeeze(Beta_X_avg(class, channel, :)), newWin, 'same');
    end
end
```

6.









7.

✓ Class 1: In the delta frequency band, there has been a more significant increase in power compared to before the intervention. While increases are also observed in other bands, the growth in the delta band is notably more pronounced.

✓ Class 2: This class experiences a decrease in power across all bands following the intervention, but shows a slight increase afterward, primarily in the delta frequency band.

✓ Class 3: After the intervention, this class shows a decline across all bands, with more pronounced changes in the delta band, where notable increases can also be seen in certain instances.

✓ Class 4: Following the intervention, this class will show an initial upward trend across all bands. The extent of this increase varies relatively across different bands, but is significantly lower in the theta band compared to others.

✓ Class 5: There is a consistent upward movement immediately after the intervention across all bands. The increase in the alpha band is greater than in others, making it generally experience a higher peak compared to the other classes, accompanied by increased power.