**BMEG 3330 Midterm Report**

**Group 1**

## 1. Overview

The midterm report aims to classify previously collected EEG data into α-band and eye-close conditions. We applied a bandpass filter to extract the α-features from the $C_Z$-channel EEG data. This report aims to elucidate the procedure of EEG preprocessing and the classification algorithm of data collected.

## 2. EEG preprocessing procedures

### 2.1 Raw EEG data acquisition

The EEG signal was collected by EEG electrodes connected to an EEG amplifier and a computer. The subject sat before the computer, clicked the "Eye Open" and "Eye Close" buttons before doing the corresponding action, and stopped after 10 seconds. The computer collected raw EEG data with a sampling rate of 256 Hz. EEG latency and condition were also recorded in the EEG experiment. After collection, the data was saved as a ***.daq*** file and a ***.mat*** file sent by TAs. We later loaded the data into a MATLAB analysis script for further processing.

### 2.2 Condition segmentation

After importing the EEG data, those corresponding to the eye-close and eye-open conditions were extracted. The different segments were identified with the EEG latency parameter, which is the eye-open or eye-close button that the desired α-band and eye-close segments by splitting the signal between the first and second button presses and the third and fourth button presses.

Below is the code for extracting the segments with the given latency values, where `i` is the condition number (1, 2), and `fs` is the sampling rate (i.e. 256 Hz).

```
1  latency = EEG_latency((2 * i - 1) : 2 * i);
2  start = latency(1); stop = latency(2);
3  eeg_cond{i} = EEGData(round(fs * start):round(fs * stop));
```

In line 1, the latency values for the start and end of a condition are obtained.
In line 3, the start and stop latency values are multiplied by the sampling rate, rounded off, to obtain the index of the desired sample points within the segments.

### 2.3 Bandpass filtering

A bandpass filter extracts and isolates the desired characteristics from the EEG data for both conditions. The 8-12 Hz α-frequency range is the target of interest. A Butterworth filter of order 6 is used to achieve this.

```
function returnValue = filterBandpass(InputData, fs, low, high)

Wn = [low, high];
order = 6;
[b, a] = butter(order, Wn / (fs * 0.5));
returnValue = filtfilt(b, a, InputData);

end
```

We give `butter()` our frequency band parameters: `low = 8, high = 12`, and it computes the filter coefficients `b` and `a`, which we then feed into `filtfilt()` to apply the coefficients to filter our EEG data.

This function is preferred over the `filter()` function as it provides zero-phase filtering, ensuring the filtered signal remains free from phase distortion. The phase relationship of the filtered data is reserved for effective and accurate data analysis. The α-features are successfully extracted without undesired signals to be further analysed or used for subsequent processing.

### 2.4 α-power extraction

After applying the bandpass filter, we can extract the α-power from the filtered EEG data, considering that the power spectrum of a function equals the magnitude squared of its Fourier transform: $\mathrm{PSD}(f) = |X(f)|^2$. We averaged the power values to summarise the powers in each respective condition and plotted *Figure 1*.

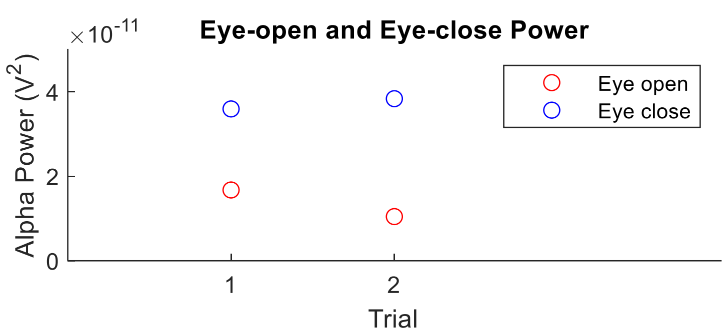


*Figure 1*
*α-power of each condition in both trials*

For a more accurate analysis, we also employed spectral estimation. Two methods are being used, namely periodogram and the Yule-Walker parametric autoregressive (AR) model of orders 10, 20, and 30, which refers to previous data points for prediction, to extract the α-power on the short segment of data collected. We can plot a graph showing power against frequency when extracting power values from desired EEG data corresponding to the eye-close and α-band segments. Power spectrum curves of varying orders are graphed to compare the power between the conditions and their classification.

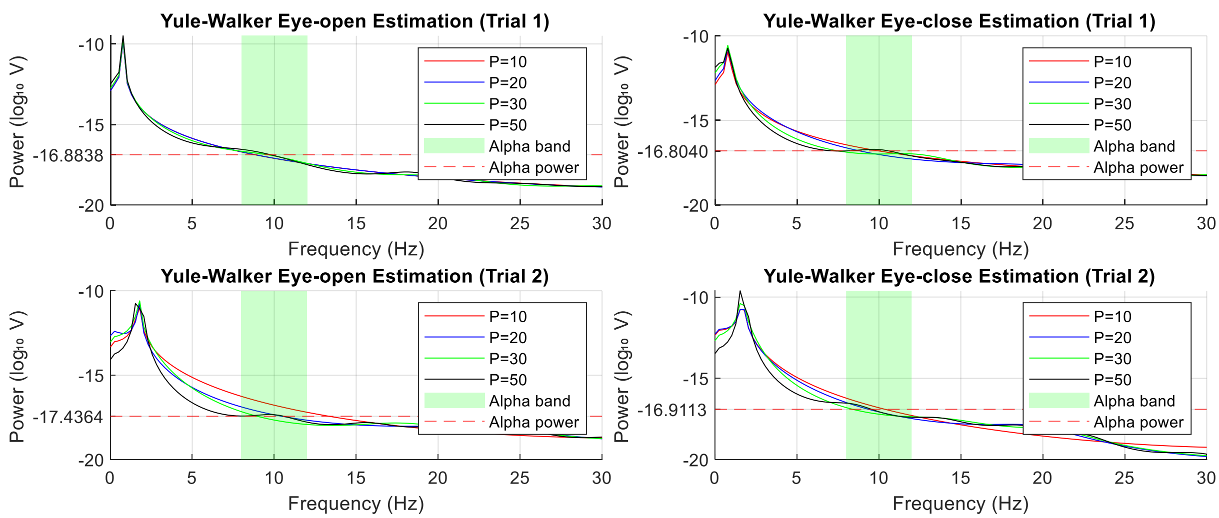


*Figure 2*
*Yule-Walker power estimations of each condition in each trial*

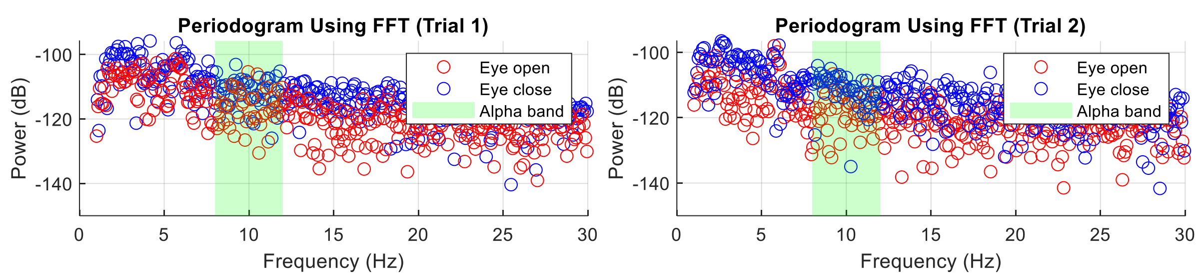


*Figure 3*
*Periodogram of both conditions in each trial*

## 3. Classification algorithm

The frequency range of EEG signals for both conditions, i.e. eye-open and eye-close, is within the α-band (8-12 Hz), and it is suppressed in the eye-open condition but not during the eye-close state when the individual is awake. Therefore, the α-band's power intensity is usually higher during eye-close than during eye-open. As observed in *Figure 2* and *Figure 3*, both methods showed that the α-power during eye-open for both trials is significantly lower than that during eye-close, as theoretically predicted.

```
eeg = EEGData(round(fs * start) : round(fs * stop));
eeg_filtered = filterBandpass(eeg, fs, 8, 12);
alpha_power = mean(abs(eeg_filtered).^2);
threshold = 3e-11;

if alpha_power <= threshold
    text{i} = ['- EEG Segment ', int2str(i), ' is eyeopen!'];
    disp(['- EEG Segment ', int2str(i), ' is eyeopen!']);
else
    text{i} = ['- EEG Segment ', int2str(i), ' is eyeclose!'];
    disp(['- EEG Segment ', int2str(i), ' is eyeclose!']);
end
```

Using the calculated α-power values, we can classify the EEG data into the two conditions using a simple threshold, where the verdict would be eye-close if the α-power exceeds the threshold and eye-open otherwise. The threshold value used for classification is set as $3 \times 10^{-11}\mathrm{V}^2$.

After testing the algorithm with the EEG data, the results, as seen in *Figure 4*, showed that they were all correctly classified.

```
Trial 1 classification:
- EEG Segment 1 is eyeclose!
- EEG Segment 2 is eyeopen!
Trial 2 classification:
- EEG Segment 1 is eyeopen!
- EEG Segment 2 is eyeclose!
```

*Figure 4*
*Classification results*

# 4. Appendix

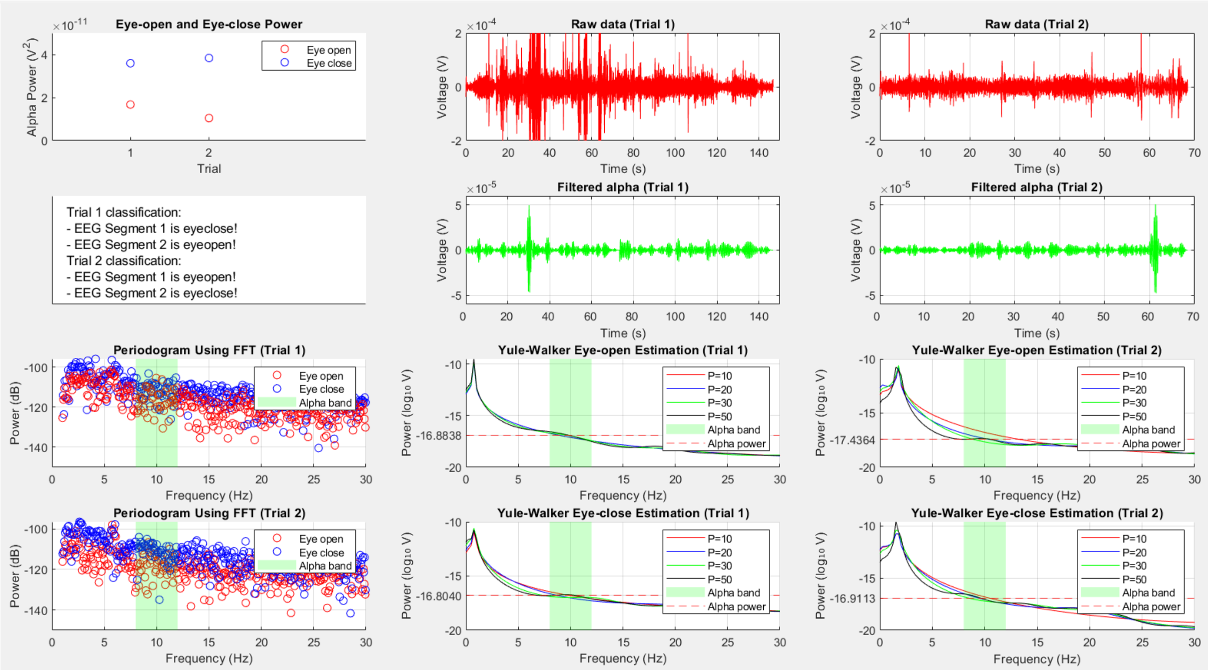


*Figure 5*
*MATLAB output window*