Signals & systems project (phase_1)
Dr. karbalaei
second Semester (98_99)

Signals and systems project (phase_1)

1398_99

Student Information

student	Email	Student number
Arshia Afzal	arshiaafzal1378@gmail.com	97109943
Nahaleh Fatemi	nahaleh.fatemi@yahoo.com	97102141

Description

To replace any placeholder text (such as Course Name above), just tap it and type.

Expectations and Goals

finding out how EEG signals work and clustering the brain electrodes.

EEG signals

- Difference between invasive and noninvasive, EEG signals

An **electroencephalogram (EEG)** is the recording of the brain electrical activity. A set of electrodes are placed on the scalp of the subject. This technique is **non-invasive** since no surgery is required. **EEG** is a fast and cheap technique. The main drawback is that by recording the electrical activity far away from the source (i.e. the neurons inside the skull), the signal we pick up is distorted and its amplitude reduced. Moreover **EEG** is often contaminated with artefacts.

The next technique in the increasing invasiveness scale is called **Electrocorticogram (ECoG)**. This technique requires opening the skull of the subject and placing an electrode array on the exposed brain. The quality of the signal (in terms of distortion and amplitude) is much better than **EEG**, but obviously surgery is required. In principle, this technique does not damage the brain, or at least not too much. It is often used in severe epileptic patients before undergoing surgery to localize the brain focus of the seizures.

Invasive EEG recordings are those recordings that are made with electrodes that have been surgically implanted on the surface or within the depth of the brain. Occasionally, the patient's epilepsy syndrome may require the use of **invasive EEG** recordings before epilepsy surgery can be considered.

Gathered from: https://www.neuroelectrics.com/blog/2014/12/18/invasive-vs-non-invasive-eeg-ready-to-become-a-cyborg/. And https://my.clevelandclinic.org/health/diagnostics/17144-invasive-eeg-monitoring .

- N100, P300, P600

1. What is p300 EEG?

The **P300** is an event-related potential (ERP) endogenous component that has a positive deflection that occurs in the scalp-recorded electroencephalogram (**EEG**) and typically elicited approximately 300 ms after the presentation of an infrequent stimulus (such as visual, auditory, or somatosensory). The presence, magnitude, topography and timing of this signal are often used as metrics of cognitive function in decision-making processes. While the neural substrates of this ERP component still remain hazy, the reproducibility and ubiquity of this signal makes it a common choice for psychological tests in both the clinic and laboratory.

2. What does p600 mean?

The P600 is an event-related potential (ERP), or peak in electrical brain activity measured by electroencephalography (EEG). It is a language-relevant ERP and is thought to be elicited by hearing or reading grammatical errors and other syntactic anomalies. it is a language-relevant ERP and is thought to be elicited by hearing or reading grammatical errors and other syntactic anomalies. Therefore, it is a common topic of study in neurolinguistics experiments investigating sentence processing in the human brain.

3. What does N100 mean?

In neuroscience, the N100 or N1 is a large, negative-going evoked potential measured by electroencephalography (its equivalent in magnetoencephalography is the M100); it peaks in adults between 80 and 120 milliseconds after the onset of a stimulus, and is distributed mostly over the fronto-central region of the scalp. It is elicited by any unpredictable stimulus in the absence of task demands. It is often referred to with the following P200 evoked potential as the "N100-P200" or "N1-P2" complex. While most research focuses on auditory stimuli, the N100 also occurs for visual (see visual N1, including an illustration), olfactory, heat, pain, balance,[4] respiration blocking, and somatosensory stimuli.

The auditory N100 is generated by a network of neural populations in the primary and association auditory cortices in the superior temporal gyrus in Heschl's gyrus[7] and planum temporale. It also could be generated in the frontal and motor areas.[9] The area generating it is larger in the right hemisphere than the left.

Gathered from:

https://en.wikipedia.org/wiki/N100 and https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5269602/ and https://en.wikipedia.org/wiki/P600_(neuroscience).

- Different bandwidths for EEG signals

- Delta: has a frequency of 3 Hz or below. It tends to be the highest in amplitude and the slowest waves. It is normal as the dominant rhythm in infants up to one year and in stages 3 and 4 of sleep.
- Theta: has a frequency of 3.5 to 7.5 Hz and is classified as "slow" activity. It is perfectly normal in children up to 13 years and in sleep but abnormal in awake adults.
- Alpha: has a frequency between 7.5 and 13 Hz. Is usually best seen in the posterior regions of the head on each side, being higher in amplitude on the dominant side. It appears when closing the eyes and relaxing, and disappears when opening the eyes or alerting by any mechanism (thinking, calculating). It is the major rhythm seen in normal relaxed adults. It is present during most of life especially after the thirteenth year.
- Beta: beta activity is "fast" activity. It has a frequency of 14 and 38 Hz. It is usually seen on both sides in symmetrical distribution and is most evident frontally. It may be absent or reduced in areas of cortical damage. It is generally regarded as a normal rhythm. It is the dominant rhythm in patients who are alert or anxious or have their eyes open.
- Gamma: Gamma brainwaves are the fastest of brain waves (high frequency, between 38 to 42 Hz), and relate to simultaneous processing of information from different brain areas. Gamma brainwaves pass information rapidly and quietly. The subtlest of the brainwave frequencies, the mind has to be quiet to access gamma. It is speculated that gamma rhythms modulate perception and consciousness, and that a greater presence of gamma relates to expanded consciousness.

- Sampling Frequency for each signal

According to Nyquist theorem, $F_s = 2f_{max}$

So:

- infra-low frequency band: $F_s = 1Hz$
- delta frequency band: $F_s = 6Hz$
- theta frequency band: $F_s = 15Hz$
- alpha frequency band: $F_s = 26Hz$
- beta frequency band: $F_s = 76Hz$
- gamma frequency band: $F_s = 84Hz$

Gathered from:

https://www.medicine.mcgill.ca/physio/vlab/biomed_signals/eeg_n.htm

- From loading the data we could easily figure out that Δt between two data points is about 0.0039s, so the sampling frequency is equal to:

$$Fs = 256Hz$$

- From the previous part and the Gamma's bandwidth we can find out the correct cutoff frequency for EEG signals which is about:

$$0.5 - 40\ Hz$$

- Now we want to plot our data's fft (Subject1.mat) using the function ***semibandfft.*** The result is:

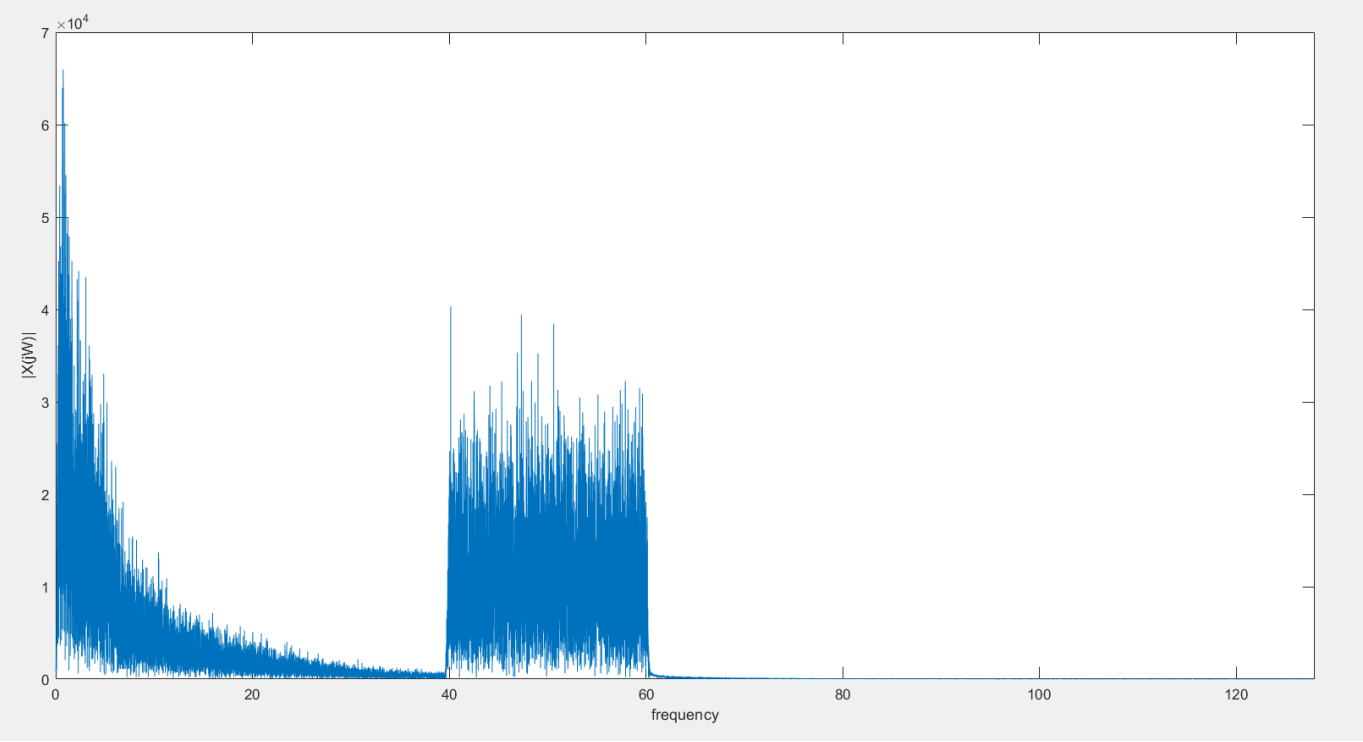


Figure 4# fft of the dataset using semi band function

- Now we want to find the bandwidth with energy analysis. By using the function *bandpower* we find out the energy of the signal between frequency f_1 =0 to f_2 =freq.
  Now we have to calculate the frequency in which the energy of our signal becomes 99 percent of total energy. For this purpose, we use a *for loop* to find the frequency **freq**
  And the answer is 60Hz as we were anticipating from the previous part result.
  But from the previous figure and the fact that Gamma's bandwidth is about 38 to 42 Hz we know that the frequencies from 40 to 60 Hz are noise frequencies so we represent the band width:

$$0.5 - 40\ Hz$$

- Cutoff frequency:

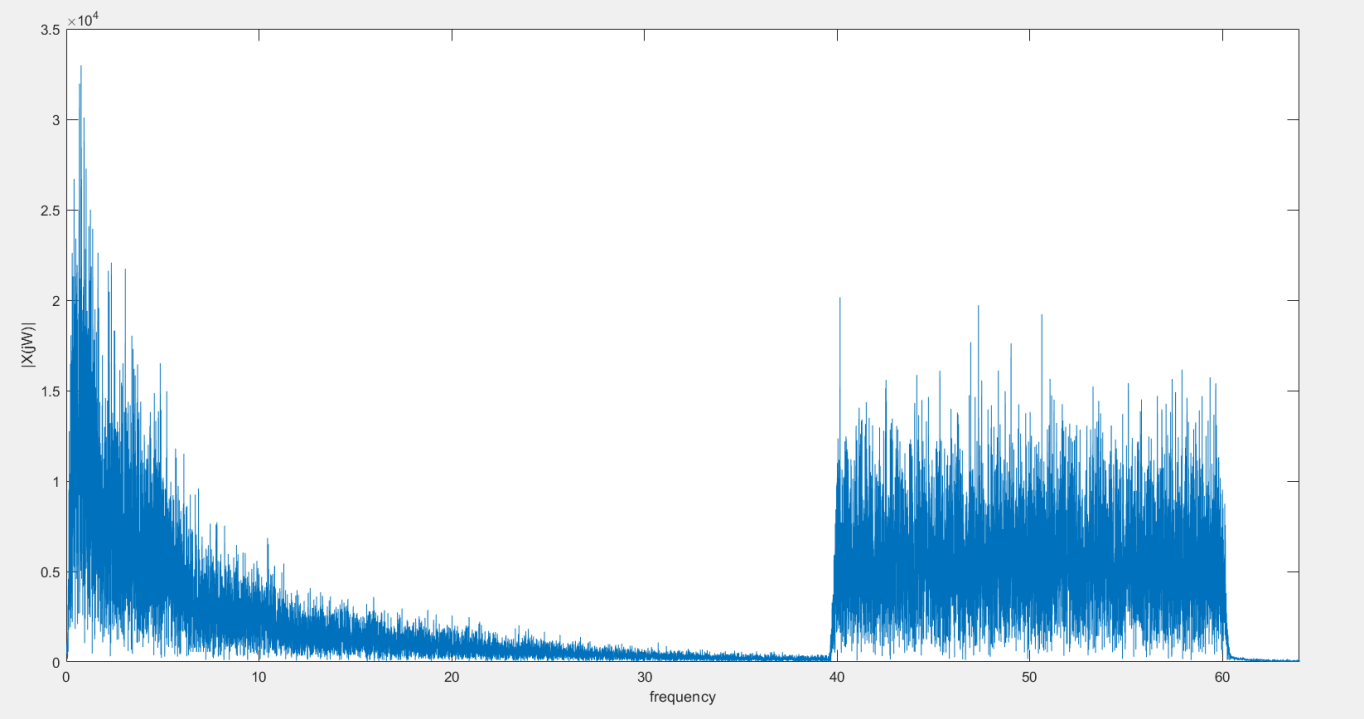
$$\boldsymbol{40\ Hz}$$

- **Mean of the signal**

We can't remove the DC part so easily because before filtering the signal we are removing the DC part of the noise too so first we have to filter the noise.

- **Reducing the sampling frequency**

Just like the second part of sampling section we use ***freqreducer*** to reduce the frequency of our signal.
And the result is:



#Figure 5# signal after reduction of Fs

As we were anticipating the magnitude of Fourier transform of our signal is multiplied by 0.5 just like the theory we know.

$$X(e^{j\Omega}) = \frac{1}{T}\sum_{r=-\infty}^{\infty} X_c(j(\frac{\Omega}{T} - \frac{2r\pi}{T}))$$

And what we have calculated from the 2nd part of the sampling section.

- **Epoching**

  Epoching is means to choose our data from the dataset from a time before the stimulation to a time after the stimulation.
  Now we build our matrix based on where stimulations(Trial) are from
  t=backward sample(0.2s) to t= forward sample(0.8s) we choose our data(Time) from 8 different channels (channels row). This is how epoching matrix actually works.

- Why can't we reduce sampling frequency before filtering?

  Because we don't know the exact Nyquist frequency (because of the noise frequency) to start reducing the sampling frequency. And even we may choose a low frequency for sampling because of noise and then our final fft signal has aliasing.

- What is the difference between epoching first and then filtering the signal and vice versa?
  Most filters for EEG signals are based on convolution, which is a linear operation, so changing the sequence of filtering and epoching won't affect the results. However, these filters in their purest form operate on infinite-duration waveforms, and nonlinear steps are often built into filters to deal with edges at the beginning and end of a finite-duration waveform. This non-linear behavior is most severe in high-pass filters, so it's better to filter the EEG data first and then epoch it. After filtering the EEG data, edges are still present, but since they are near the very beginning or the end of the signal, they are farther in time from the events in our interest.

- Why do we have to wait before starting the EEG processing after we have placed the EEG_hat?

  Because after placing the EEG_hat neurons start spiking because of placing the hat and that stimulation is not the correct stimulation for us to study. So we have to wait a bit until our noise is gone and test conditions become steady.
  And as said above, it is better to wait for a few seconds before starting the stimulations to avoid the edge artifacts from extending into our time period of stimuli.

- **Freqband (function)**

  In this part we want to explain the function *freqband* which helps us choose a self-defined bandwidth (Bandpass1-Bandpass2) of our input signal. For this function we used *bandpass* function from Mat-Lab.

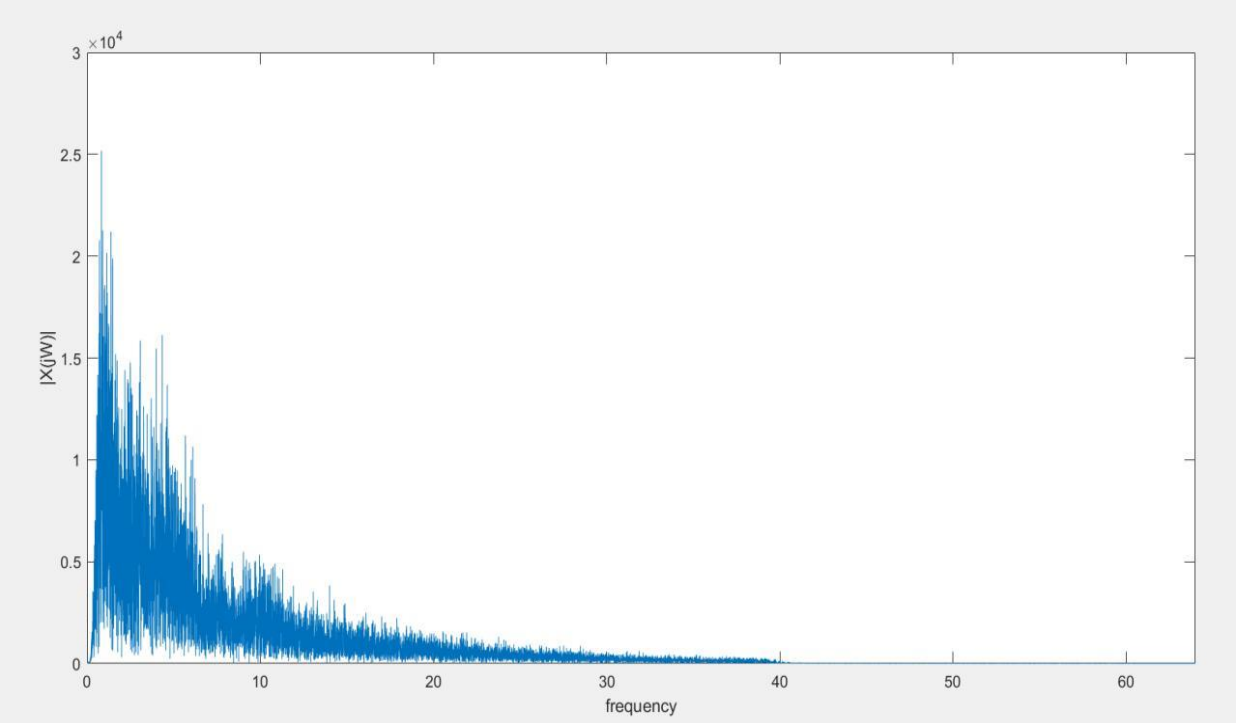
!!**But this** function's run time is too much.

So we use the *BPF function* to build our new filter with an order of 100 and with sampling frequency about 128 Hz. And by using this filter to filter our epoch matrix rows (all trials for all channels) run time reduces significantly.

Like the code below:

```
x(i,:,j) = filter(BPF(100,10,20,Fs/2),1,epoch(i,:,j));
```

The FFT of our signal looks like this after filtering and removing the DC part:



#Figure 6# signal after filtering

- Calculating energy of our signal
  For this part we have calculated the energy of each bond (alpha,betha,…) in a matrix form for all channels and trials then we have calculated them separately and at least the total energy for each bond is:

  (Delta:3.2065 theta:1.9245 alpha:1.2948 betha:0.6252 gamma:4.8041)*10^7

Total Energy = 1.1855e+08

Clustering

- ***Correlation-matrix***
- Show that r_{xy} is between -1,1:

$$\int_{-\infty}^{\infty} \left(x_{(t)} - y_{(t)}\right)^2 dt \geq 0 \rightarrow \int_{-\infty}^{\infty} {x_{(t)}}^2 dt + \int_{-\infty}^{\infty} {y_{(t)}}^2 dt - 2\int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt \geq 0$$

$$\rightarrow \int_{-\infty}^{\infty} {x_{(t)}}^2 dt + \int_{-\infty}^{\infty} {y_{(t)}}^2 dt \geq 2\int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt$$

$$\rightarrow \frac{\int_{-\infty}^{\infty} {x_{(t)}}^2 dt + \int_{-\infty}^{\infty} {y_{(t)}}^2 dt}{2} \geq 2\int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt \quad (1)$$

$$\int_{-\infty}^{\infty} {x_{(t)}}^2 dt \equiv a \mid \int_{-\infty}^{\infty} {y_{(t)}}^2 dt \equiv b \;\; \left(\sqrt{a} - \sqrt{b}\right)^2 \geq 0 \rightarrow \frac{a+b}{2} \leq \sqrt{ab}$$

$$\rightarrow \frac{\int_{-\infty}^{\infty} {x_{(t)}}^2 dt + \int_{-\infty}^{\infty} {y_{(t)}}^2 dt}{2} \leq \sqrt{\int_{-\infty}^{\infty} {x_{(t)}}^2 dt \int_{-\infty}^{\infty} {y_{(t)}}^2 dt} \quad (2)$$

$$from\ 1\ and\ 2 \rightarrow \sqrt{\int_{-\infty}^{\infty} {x_{(t)}}^2 dt \int_{-\infty}^{\infty} {y_{(t)}}^2 dt} \geq \int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt$$

$$\frac{\left|\int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt\right|}{\sqrt{\int_{-\infty}^{\infty} {x_{(t)}}^2 dt \int_{-\infty}^{\infty} {y_{(t)}}^2 dt}} \leq 1 \rightarrow$$

$$\boxed{-1 \leq r_{xy} \leq 1}$$

- If r_{xy} =1:

$$\rightarrow \int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt \times \int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt = \int_{-\infty}^{\infty} {x_{(t)}}^2\, dt \int_{-\infty}^{\infty} {y_{(t)}}^2\, dt$$

$$\rightarrow \frac{\int_{-\infty}^{\infty} {x_{(t)}}^2\, dt}{\int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt} \times \frac{\int_{-\infty}^{\infty} {y_{(t)}}^2\, dt}{\int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt} = 1 \rightarrow \frac{\int_{-\infty}^{\infty} {x_{(t)}}^2\, dt}{\int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt} = \alpha \; and \; \frac{\int_{-\infty}^{\infty} {y_{(t)}}^2\, dt}{\int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt} = \frac{1}{\alpha}$$

So:

$$\int_{-\infty}^{\infty} {x_{(t)}}^2\, dt = \alpha \int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt \rightarrow \int_{-\infty}^{\infty} {x_{(t)}}^2\, dt - \alpha \int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt = 0$$

$$\rightarrow \int_{-\infty}^{\infty} x_{(t)} \left(x_{(t)} - \alpha y_{(t)}\right) dt = 0 \; (1)$$

$$\alpha \int_{-\infty}^{\infty} {y_{(t)}}^2\, dt = \int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt \rightarrow \alpha \int_{-\infty}^{\infty} {y_{(t)}}^2\, dt - \int_{-\infty}^{\infty} x_{(t)}\, y_{(t)} dt = 0$$

$$\rightarrow \int_{-\infty}^{\infty} y_{(t)} \left(x_{(t)} - \alpha y_{(t)}\right) dt = 0 \; (2)$$

$$(1) - \alpha(2) \rightarrow \int_{-\infty}^{\infty} x_{(t)} \left(x_{(t)} - \alpha y_{(t)}\right) dt - \alpha \int_{-\infty}^{\infty} y_{(t)} \left(x_{(t)} - \alpha y_{(t)}\right) dt = 0$$

$$\rightarrow \int_{-\infty}^{\infty} \left(x_{(t)} - \alpha y_{(t)}\right) \left(x_{(t)} - \alpha y_{(t)}\right) dt = 0 \rightarrow \int_{-\infty}^{\infty} \left(x_{(t)} - \alpha y_{(t)}\right)^2 dt = 0$$

because $\left(x_{(t)} - \alpha y_{(t)}\right)^2$ is always positive so :

$$\text{if} : \int_{-\infty}^{\infty} \left(x_{(t)} - \alpha y_{(t)}\right)^2 dt = 0 \rightarrow \left(x_{(t)} - \alpha y_{(t)}\right)^2 = 0 \rightarrow$$

$$\boxed{x_{(t)} = \alpha y_{(t)}}$$

- Because this parameter is so similar to Internal multiplication of two vectors and actually it is Internal multiplication for two functions and because this parameter is only equal to 1 when $x_{(t)} = \alpha y_{(t)}$ it is a good standard to check the similarity of two signals.

  If this parameter is near 1 two signals are similar and if its near 0 two signals are independent. (if = 0 they are completely independent).

- Removing the signal noises

  as we have calculated first we remove our noise then start the epoching progress so we expect no noisy behavior for our epoched signal.

- Sampling frequency reduction

  With first plotting the data it is obvious that we can reduce the sampling frequency 4 times without losing any data. So just like previous Fs reduction we have:

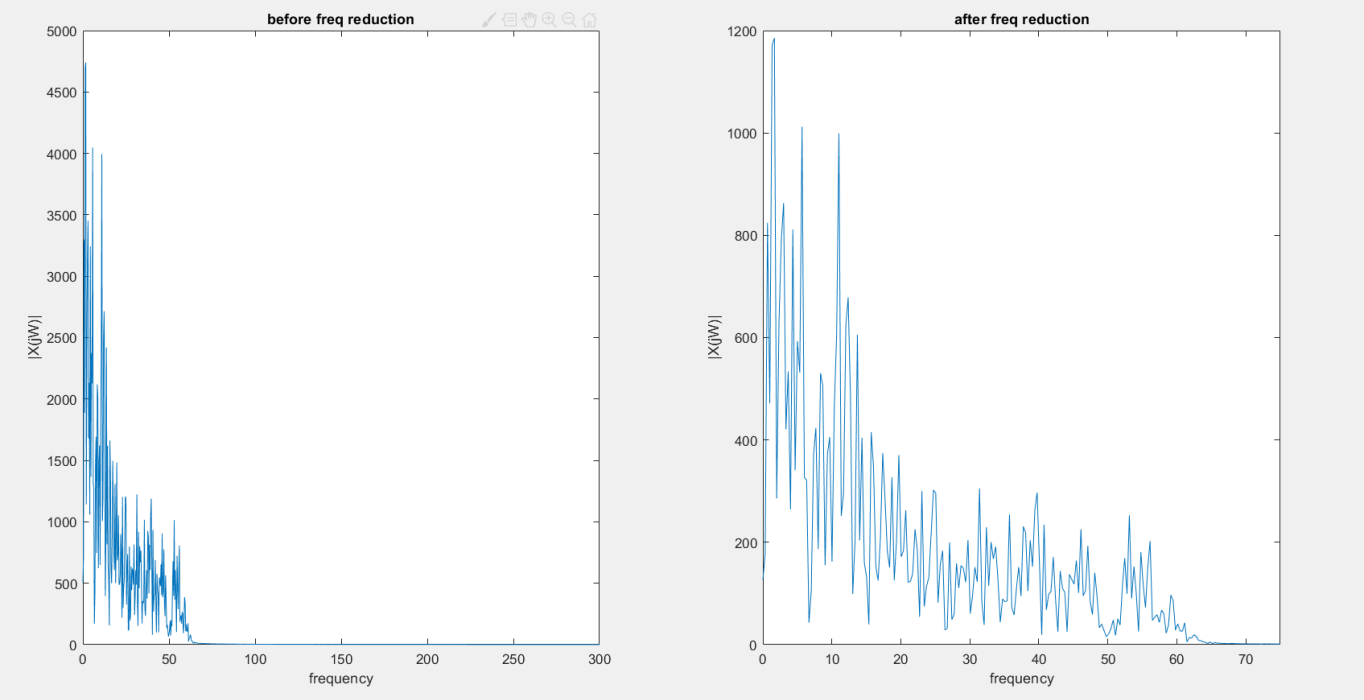


Figure 7# signal before and after Fs reduction

- Building the correlation matrix

- ***Clustering***

  First we have to present some equation for the distance between two channels we use the equation below:

  $d=1-|R_{xy}|$

which d is the distance matrix and *R* is the correlation matrix and now d is a good parameter for measuring the distance between two channels and for the distance between two different clusters we use two different algorithms which they are known as upgama and wpgama algorithms like bellow:

Upgama

The UPGMA algorithm constructs a rooted tree (dendrogram) that reflects the structure present in a pairwise similarity matrix (or a dissimilarity matrix). At each step, the nearest two clusters are combined into a higher-level cluster. The distance between any two clusters A and B, each of size (*i.e.*, cardinality) |A| and |B|, is taken to be the average of all distances $d_{(x,y)}$ between pairs of objects x in A and y in B, that is, the mean distance between elements of each cluster:

$$\frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x, y)$$

In other words, at each clustering step, the updated distance between the joined clusters $A \cup B$ and a new cluster X is given by the proportional averaging of the $d_{(A,X)}$ and $d_{(B,X)}$ distances:

$$d_{(\mathcal{A} \cup \mathcal{B}),X} = \frac{|\mathcal{A}| \cdot d_{\mathcal{A},X} + |\mathcal{B}| \cdot d_{\mathcal{B},X}}{|\mathcal{A}| + |\mathcal{B}|}$$

Which is our first algorithm and equation for distance.

One example from:
https://en.wikipedia.org/wiki/UPGMA#:~:text=UPGMA%20(unweighted%20pair%20group%20method,weighted%20variant%2C%20the%20WPGMA%20method.

- **First clustering**

Let us assume that we have five elements (a, b, c, d, e) and the following matrix D_1 of pairwise distances between them :

	a	b	c	d	e
a	0	17	21	31	23
b	17	0	30	34	21
c	21	30	0	28	39
d	31	34	28	0	43
e	23	21	39	43	0

In this example, $D_1(a, b) = 17$ is the smallest value of D_1, so we join elements a and b.

- **First branch length estimation**

Let u denote the node to which a and b are now connected. Setting $\delta(a, u) = \delta(b, u) = D_1(a, b)/2$ ensures that elements a and b are equidistant from u. This corresponds to the expectation of the ultrametricity hypothesis. The branches joining a and b to u then have lengths $\delta(a, u) = \delta(b, u) = 17/2 = 8.5$ (*see the final dendrogram*)

- **First distance matrix update**

We then proceed to update the initial distance matrix D_1 into a new distance matrix D_2 (see below), reduced in size by one row and one column because of the clustering of a with b. Bold values in D_2 correspond to the new distances, calculated by **averaging distances** between each element of the first cluster (a, b) and each of the remaining elements:

$D_2((a, b), c) = (D_1(a, c) \times 1 + D_1(b, c) \times 1)/(1 + 1) = (21 + 30)/2 = 25.5$

$D_2((a, b), d) = (D_1(a, d) + D_1(b, d))/2 = (31 + 34)/2 = 32.5$

$D_2((a, b), e) = (D_1(a, e) + D_1(b, e))/2 = (23 + 21)/2 = 22$

Italicized values in D_2 are not affected by the matrix update as they correspond to distances between elements not involved in the first cluster.

And this was just for the first step.

Wpgama

The WPGMA algorithm constructs a rooted tree (dendrogram) that reflects the structure present in a pairwise distance matrix (or a similarity matrix). At each step, the nearest two clusters, say i and j, are combined into a higher-level cluster $i \cup j$. Then, its distance to another cluster k is simply the arithmetic mean of the average distances between members of k and i and k and j :

$$d_{(i \cup j),k} = \frac{d_{i,k} + d_{j,k}}{2}$$

Which is our second algorithm and equation for distance.

The WPGMA algorithm produces rooted dendrograms and requires a constant-rate assumption: it produces an ultrametric tree in which the distances from the root to every branch tip are equal. This ultrametricity assumption is called the molecular clock when the tips involve DNA, RNA and protein data.

This information and examples has been gathered from: https://en.wikipedia.org/wiki/WPGMA

- Self-defined Function “cluster”

This part is an introduction to our self –define function ***cluster***. With 3 inputs level (which shows the number of steps we want to go further) input signal and method
(which is our method to calculate our clustering matrix).
in this function first we change our matrix to a low triangular matrix because the arrays in the up triangular part are just as the same as the low triangular part so we change them to a number more than 1 (in this function for example 4) so they would not interference our computation now we use both methods above (based on what user has entered) to update our matrix so we find the minimum distance between two clusters (in step 1 between two arrays) and then put these two clusters in one new cluster(group). For this purpose, we have an output matrix named clustermat first this matrix is an (63*63) matrix with first row linspace (1,63,63) then in each iteration(levels) two channels merges and became in one cluster (group) and one of them (column) become all zero here is an example for a 4*4 matrix:

note: all of this results have been reduced in dimension for better view of matrix and for next parts.

This part of code has been commented well in functions.(just like previous parts ☺)

4	2	3
1	0	0

#Figure 8# 1st step

4	3
1	2

#Figure 8# 2nd step

note: In this function we have used levels instead of Distencemeasure.

4
3
2
1

#Figure 9# 3rd step

- How can we find the right clusters of our data?

  We can use two different standards to stop our algorithm. The first one is distance measurement, which means that if the distance between clusters is less than a certain value, we assume that we have found the right clusters. The second way of finding the right clusters, which we have used here, is changing the number of iterations and finding the right one. By the 55th iteration, the output is as seen bellow. And we can see that with this number of iterations the right clusters are obtained.

  In figure 11, the clusters are visualized for better understanding of the output. The graph seen in this figure is plotted using scatter function and the points are numbered using the graph present in the project's document.

43	56	51	52	63	60	58	62
34	55	50	19	61	59	0	0
33	54	47	0	57	32	0	0
10	53	46	0	31	30	0	0
5	49	45	0	29	28	0	0
4	48	42	0	24	0	0	0
3	44	41	0	0	0	0	0
2	39	40	0	0	0	0	0
1	35	38	0	0	0	0	0
0	27	37	0	0	0	0	0
0	26	36	0	0	0	0	0
0	25	22	0	0	0	0	0
0	23	18	0	0	0	0	0
0	21	17	0	0	0	0	0
0	20	16	0	0	0	0	0
0	15	14	0	0	0	0	0
0	11	13	0	0	0	0	0
0	6	12	0	0	0	0	0
0	0	9	0	0	0	0	0
0	0	8	0	0	0	0	0
0	0	7	0	0	0	0	0

Figure 10 # correlation matrix with 55 iterations

- Given the results of clustering and the location of the electrodes, is there a relationship between the electrodes of each cluster?

  Yes. According to the figures 10 and 11, the relation between electrodes and their location is evident. As we anticipated, the electrodes that are located closer together, are placed in one cluster.

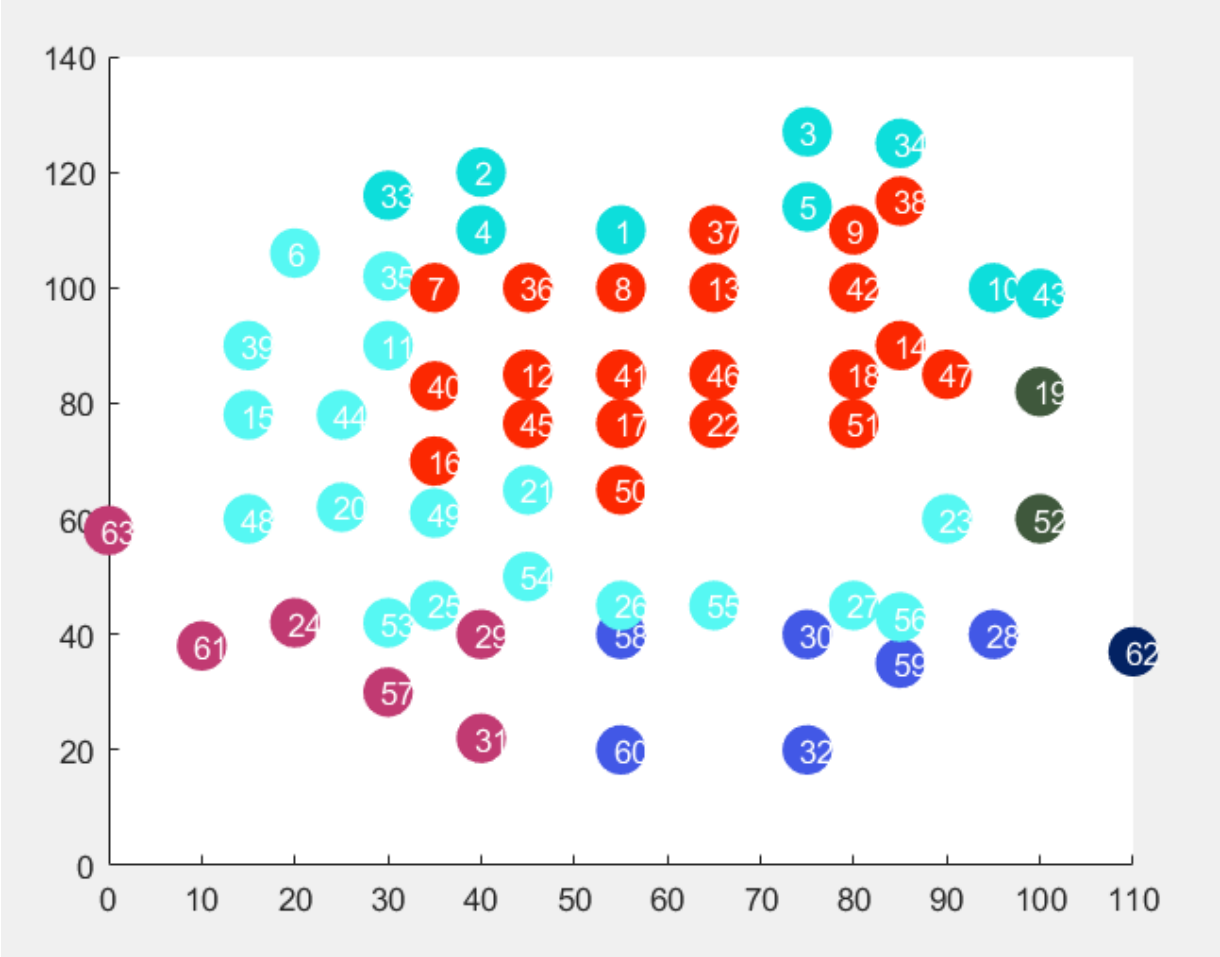


Figure 11 # visualization of the clusters

- After implementing this algorithm on the 8-electrode dataset, do we see a relation between the electrodes?

  After making sure that the algorithm above is working properly, we implemented it on the 8-electrode dataset and the results (the correlation matrix and the clusters) are as seen bellow:

1	0.3585	0.2016	0.2297	0.2158	0.0912	0.0960	0.1053
0.3585	1	0.2701	0.3564	0.2912	0.1540	0.1639	0.1783
0.2016	0.2701	1	0.3838	0.3104	0.3348	0.2847	0.2478
0.2297	0.3564	0.3838	1	0.3926	0.2822	0.3138	0.2902
0.2158	0.2912	0.3104	0.3926	1	0.2281	0.2644	0.3374
0.0912	0.1540	0.3348	0.2822	0.2281	1	0.3202	0.2605
0.0960	0.1639	0.2847	0.3138	0.2644	0.3202	1	0.3127
0.1053	0.1783	0.2478	0.2902	0.3374	0.2605	0.3127	1

Figure 12 # correlation matrix of the 8-electrode dataset

2	5	6	7	8
1	4	0	0	0
0	3	0	0	0

Figure 13 # cluster matrix of the 8-electrode dataset (5 clusters)

2	5	7	8
1	4	6	0
0	3	0	0

Figure 14 # cluster matrix of the 8-electrode dataset (4 clusters)