BE 521: Homework 1

Exploring Neural Signals

Spring 2020

Andrew Clark

Due: Tuesday 1/28/2020 11:59 PM

Objective: Working with the IEEG Portal to explore different Neural signals.
Collaborators: Vishal Tien, Alex Silva.

1 Seizure Activity (16 pts)

The dataset `I521_A0001_D002` contains an example of human intracranial EEG (iEEG) data displaying seizure activity. It is recorded from a single channel (2 electrode contacts) implanted in the hippocampus of a patient with temporal lobe epilepsy being evaluated for surgery. In these patients, brain tissue where seizures are seen is often resected. You will do multiple comparisons with this iEEG data and the unit activity that you worked with in Homework 0 (`I521_A0001_D001`). You will have to refer to that homework and/or dataset for these questions.

1. Retrieve the dataset in MATLAB using the IEEGToolbox and generate a *session* variable as before (No need to report the output this time). What is the sampling rate of this data? What is the maximum frequency of the signal content that we can resolve? (2 pts)

```
%Create new session
session = IEEGSession('I521_A0001_D002', 'andrewc', '/Users/andrewclark/Downloads/ieeg_password.bin' );

%Get info on the session
session.data
```

```
Warning: Objects of edu/upenn/cis/db/mefview/services/TimeSeriesDetails class
exist - not clearing java
Warning: Objects of edu/upenn/cis/db/mefview/services/TimeSeriesInterface class
exist - not clearing java
IEEGSETUP: Found log4j on Java classpath.
URL: https://www.ieeg.org/services
Client user: andrewc
Client password: ****

ans =

  <a href="matlab:help('IEEGDataset')">IEEGDataset</a>:

         snapName: 'I521_A0001_D002'
          montage: As Recorded
           filter: 'No Filter'
         resample: 'Not resampled'
       sampleRate: 200
```

```
           values: 
    channelLabels: [1x2 cell]
         annLayer: []
      rawChannels: [1x1 IEEGTimeseries]
      allMontages: [1x44 IEEGMontage]

  <a href="matlab:methods(IEEGDataset)">Methods</a>, <a href="matlab:IEEGObject.openPortalSite()">main.ieeg.
```

The sampling rate is 200 Hz. The maximum frequency of the signal content that we can resolve is 200 Hz / 2 = 100 Hz.

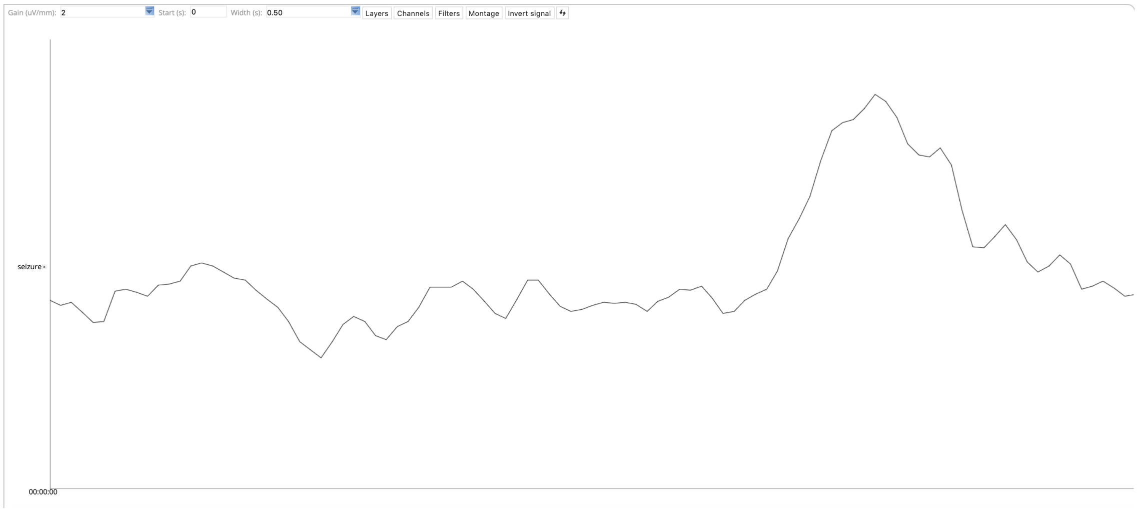
2. How does the duration of this recording compare with the recording from HW0 (`I521_A0001_D001`)? (2 pts)

```
%Calculate the duration
durationInUSec = session.data(1).rawChannels(1).get_tsdetails.getDuration;
durationInSec = durationInUSec / 1e6
```

```
durationInSec =

  644.9950
```

The duration of this recording is 644.9950 seconds, however the duration of the recording from HW0 was 10 seconds (as calculated in HW0).

3. Using the time-series visualization functionality of the IEEG Portal, provide a screenshot of the first 500 ms of data from this recording. (2 pts)



Gain (uV/mm): 2 Start (s): 0 Width (s): 0.50

4. Compare the activity in this sample with the data from HW0. What differences do you notice in the amplitude and frequency characteristics? (2 pts)

```
%Pull in all data for HW1 data
nr = ceil((session.data.rawChannels(1).get_tsdetails.getEndTime)/1e6*session.data.sampleRate);
allData = session.data.getvalues(1:nr,1);
sample_rate_D002=200;
```

```
first_500_ms_HW1=allData(1:0.5*sample_rate_D002);
max_peak_HW1=max(first_500_ms_HW1)
```

```
max_peak_HW1 =

   167
```

```
%Start HW0 data
session_HW0 = IEEGSession('I521_A0001_D001', 'andrewc', '/Users/andrewclark/Downloads/ieeg_password.bin' );
```

```
Warning: Objects of edu/upenn/cis/db/mefview/services/TimeSeriesDetails class
exist — not clearing java
Warning: Objects of edu/upenn/cis/db/mefview/services/TimeSeriesInterface class
exist — not clearing java
IEEGSETUP: Found log4j on Java classpath.
URL: https://www.ieeg.org/services
Client user: andrewc
Client password: ****
```

```
%Compute metrics for comparing amplitude and frequency
session_HW0.data
sample_rate_D001=32051;
nr = ceil((session_HW0.data.rawChannels(1).get_tsdetails.getEndTime)/1e6*session_HW0.data.sampleRate);
allData_HW0 = session_HW0.data.getvalues(1:nr,1);
sample_rate_D002=200;
first_500_ms_HW0=allData_HW0(1:0.5*sample_rate_D001);
max_peak_HW0=max(first_500_ms_HW0)
```

```
ans =

  <a href="matlab:help('IEEGDataset')">IEEGDataset</a>:

         snapName: 'I521_A0001_D001'
          montage: As Recorded
           filter: 'No Filter'
         resample: 'Not resampled'
       sampleRate: 32051
           values:
    channelLabels: [1x2 cell]
         annLayer: []
      rawChannels: [1x1 IEEGTimeseries]
      allMontages: [1x44 IEEGMontage]

  <a href="matlab:methods(IEEGDataset)">Methods</a>, <a href="matlab:IEEGObject.openPortalSite()">main.ieeg.

Warning: Integer operands are required for colon operator when used as index.

max_peak_HW0 =

  131.8830
```

```
average_amp_D001=mean(first_500_ms_HW0)
average_amp_D002=mean(first_500_ms_HW1)
```

```
average_amp_D001 =

    0.0041


average_amp_D002 =

   −7.7000
```

The amplitude and frequency characteristics of activity in this sample (first 500 ms of the I521_A0001_D002 dataset) are different compared to the data from the same time period in the HW0 dataset (I521_A0001_D001). In terms of amplitude characteristics, the spike of the HW1 dataset (I521_A0001_D002) was 167 uV during the first 500ms (found by computing the max ampltiude of the signal in this clip), while the spike height of the I521_A0001_D001 dataset from HW0 was 131.9uV. Thus, the maximum amplitude of the HW1 dataset was higher than the maximum amplitude of the HW0 dataset. However, when we look simply at the overall average amplitude of the entire HW1 I521_A0001_D002 dataset, this measure is equal to -7.700 uV, while the overall average amplitude of the entire HW0 I521_A0001_D001 dataset was 0.0041 uV. Thus, the spike in the I521_A0001_D002 dataset had a higher amplitude, but the overall average ampltiude of the I521_A0001_D001 dataset was higher (for the first 500ms clip we are analyzing here). In terms of frequency content, the maximum frequency that can be resolved (due to the sampling rate chosen) in this dataset is 200 Hz / 2 = 100 Hz, while in the HW0 dataset the maximum frequency that could be resolved was 32051 Hz / 2 = 16026 Hz. Thus, assuming those conducting the experiment did not sample at a rate that would allow aliasing, the data in the HW1 dataset has much lower frequency content; the signal oscillates a lot more slowly compared to the data from HW0. The lower frequency I521_A0001_D002 dataset also has small ridges during these slow increases and decreases in voltage, while the data from HW0 had smoother, faster increases and decreases.

5. The unit activity sample in (I521_A0001_D001) was high-pass filtered to remove low-frequency content. Assume that the seizure activity in (I521_A0001_D002) has not been high-pass filtered. Given that the power of a frequency band scales roughly as $1/f$, how might these differences in preprocessing contribute to the differences you noted in the previous question? (There is no need to get into specific calculations here. We just want general ideas.) (3 pts)

   This difference in preprocessing would contribute to the differences noted because since the I521 A0001 D001 was high pass filtered, the only frequency content that we can observe in this signal is high frequency content. As such, when we look at the I521 A0001 D001 dataset it appears to have many more oscillations in it compared to the I521 A0001 D002 dataset. If the I521 A0001 D002 was also high pass filtered and the low frequency content was removed, we would see only a higher rate of oscillations in amplitude.

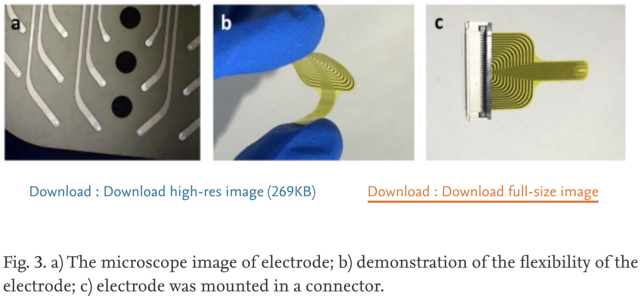
   Furthermore, if the I521 A0001 D002 dataset was also high pass filtered we would see fewer higher power signals (the waves would be flatter and have lower amplitudes). The signal would have lower amplitudes because the power (and thus the amplitude, as power is proportional to amplitude squared) of a frequency band scales roughly as 1/f, which means lower frequency bands have higher power and higher frequency bands have lower power (Hass, 2018). Thus, removing the lower frequency content from dataset I521 A0001 D002 would decrease power, making it even flatter (the frequency bands would have a lower amplitude) compared to the I521 A0001 D001 dataset. It follows that if the I521 A001 D001 was not high pass filtered it would have more sharp increases in amplitude and more spikes, since more low frequency content would be included in the signal.

   Reference: Hass, J. (2018). Introduction to Computer Music: Volume One. Retrieved January 28, 2020, from https://cecm.indiana.edu/etext/acoustics/chapter1_amplitude3.shtml

6. Two common methods of human iEEG are known as electrocorticography (ECoG) and stereoelectroencephalography (SEEG). For either of these paradigms (please indicate which you choose), find

and report at least two of the following electrode characteristics: shape, material, size. Please note that exact numbers aren't required, and please cite any sources used. (3 pts)

In the 2017 paper titled "Flexible ECoG electrode for implantation and neural signal recording applications" by Dong et al., a type of ECoG electrode is described in detail. To describe the material of these ECoG electrodes fabricated by Dong et al., these electrodes are based "based on a chromimum-silver-chromium (Cr-Ag-Cr) structure deposited on a 50 micrometer thick polymide (PI) foil" (Dong et al., 2017). These electrodes were fabricated into an array; the array is 16.3 mm wide by 24.8mm long by 60 micrometers thick, and each electrode is a circle 50 micrometers in diameter (Dong et al., 2017). The shape of the entire electrode array is best described as a collection of small arm-like structures with the circular electrodes at the end of these structures; the array and these structures are shown in the figure below. The idea behind the design is that electrical activity can be picked up by the electrodes at the end of each of these structures and this activity can be transferred to the solder pads at the other side of the array.



Download : Download high-res image (269KB)   Download : Download full-size image

Fig. 3. a) The microscope image of electrode; b) demonstration of the flexibility of the electrode; c) electrode was mounted in a connector.

The array of ECoG electrodes described above. Figure from Dong et al., 2017.

Reference: Dong, S., Chen, W., Wang, X., Zhang, S., Xu, K., Zheng, X. (2017). Flexible ECoG electrode for implantation and neural signal recording applications. Vacuum, 140, 96–100. doi: 10.1016/j.vacuum.2016.12.024

7. What is a local field potential? How might the characteristics of human iEEG electrodes cause them to record local field potentials as opposed to multiunit activity, which was the signal featured in HW0 as recorded from 40 micron Pt-Ir microwire electrodes? (2 pts)

   As stated in the paper "Neural Protheses for Reaching," the "local field potential is a summation signal of excitatory and inhibitory dendritic potentials from a large number of neurons near the recording site" (Scherberger, 2009). The characteristics of human iEEG electrodes might cause them to record local field potentials as opposed to multiunit activity because human iEEG electrodes are much larger in diameter compared to the smaller 40 micron Pt-Ir microwire electrodes, which causes the human iEEG electrodes to be able to pick up the local field potentials caused by an aggregate of neuronal potentials. In contrast, the smaller size of the 40 micron Pt-Ir microwire electrodes allows them to be able to pick up signals from multiunit activity. Additionally, another characteristic of the 40 micron Pt-Ir microwire electrodes is the higher sampling rate used in conjuction with them; this higher sampling rate allows them to pick up those multiunit signals, compared to the lower sampling rate used by the human iEEG electrodes to pick up the local field potentials.

   Reference: Scherberger, H. (2009). Neural Prostheses for Reaching. Encyclopedia of Neuroscience, 213–220. doi: 10.1016/b978-008045046-9.01114-1

2 Evoked Potentials (17 pts)

The data in I521_A0001_D003 contains an example of a very common type of experiment and neuronal signal, the evoked potential (EP). The data show the response of the whisker barrel cortex region of rat brain to an air puff stimulation of the whiskers. The `stim` channel shows the stimulation pattern, where the falling edge of the stimulus indicates the start of the air puff, and the rising edge indicates the end. The `ep` channel shows the corresponding evoked potential. Once again, play around with the data on the IEEG Portal, in particular paying attention to the effects of stimulation on EPs. You should observe the data with window widths of 60 secs as well as 1 sec. Again, be sure to explore the signal gain to get a more accurate picture. Finally, get a sense for how long the trials are (a constant duration) and how long the entire set of stimuli and responses are.

1. Based on your observations, should we use all of the data or omit some of it? (There's no right answer, here, just make your case either way in a few sentences.) (2 pts)

   Based on my observations, we should omit some of the data in this dataset. This is because I have observed that for some of the trials there are two spikes in the EP signal. These artifacts may be due to something else happening to the rat during the experiment that is causing a second spike in a single trial. The rat may have seen, heard, or felt something that cause this second spike to occur. For example, a loud noise or a bright light may have affected the rat. As such, the trials with two spikes should be omitted because these artifacts most likely constituent a violation of the experiment protocol.

2. Retrieve the `ep` and `stim` channel data in MATLAB. What is the average latency (in ms) of the peak response to the stimulus onset over all trials? (Assume stimuli occurs at exactly 1 second intervals)(3 pts)

```
%Create new session
session2=IEEGSession('I521_A0001_D003', 'andrewc', '/Users/andrewclark/Downloads/ieeg_password.bin' );
```

```
Warning: Objects of edu/upenn/cis/db/mefview/services/TimeSeriesDetails class
exist — not clearing java
Warning: Objects of edu/upenn/cis/db/mefview/services/TimeSeriesInterface class
exist — not clearing java
IEEGSETUP: Found log4j on Java classpath.
URL: https://www.ieeg.org/services
Client user: andrewc
Client password: ****
```

```
%Pull in all data
session2.data(1)

nr = ceil((session2.data.rawChannels(1).get_tsdetails.getEndTime)/1e6*session2.data.sampleRate);
allData_D003= session2.data.getvalues(1:nr, 1:2);
```

```
ans =

  <a href="matlab:help('IEEGDataset')">IEEGDataset</a>:

         snapName: 'I521_A0001_D003'
          montage: As Recorded
           filter: 'No Filter'
         resample: 'Not resampled'
       sampleRate: 2713
           values:
    channelLabels: [2x2 cell]
         annLayer: []
      rawChannels: [1x2 IEEGTimeseries]
```

```
     allMontages: [1x44 IEEGMontage]

  <a href="matlab:methods(IEEGDataset)">Methods</a>, <a href="matlab:IEEGObject.openPortalSite()">main.ieeg.
```

```
%Make EP and STIM Vectors
ep=allData_D003(:,1);
stim=allData_D003(:,2);
```

```
%Find total duration and sample rate
durationInUSec = session2.data(1).rawChannels(1).get_tsdetails.getDuration;
durationInSec = durationInUSec / 1e6;
sample_rate=session2.data.sampleRate;
```

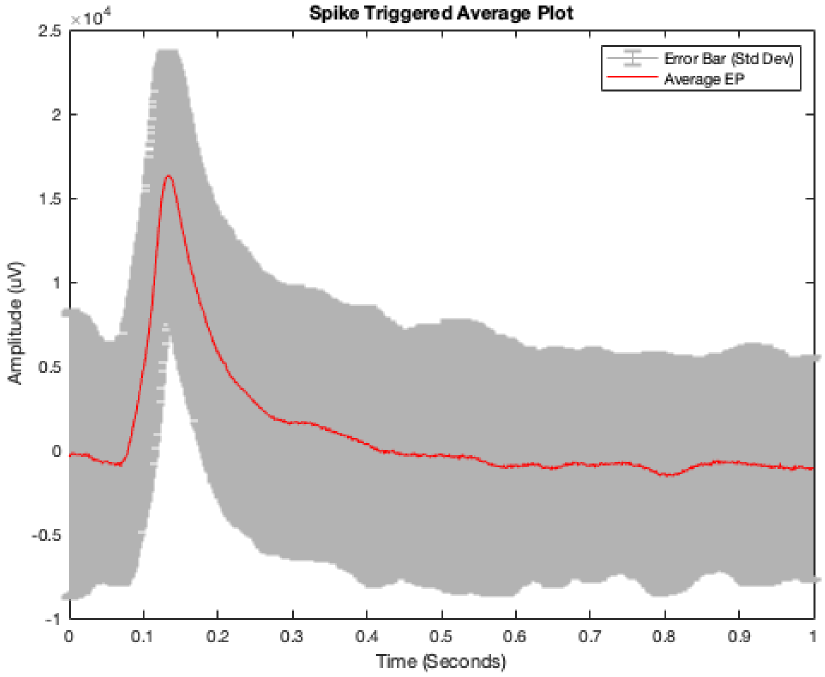
```
%Look at the data in terms of 1 second bins and then find the peak location
%in each bin. The TA John said this method is ok (it is ok to look at the data in 1 second
%bins).
stimulus_times=ceil(0:1:durationInSec);

%Make an empty vector for storing latencies for each trial
peak_locs_ep=[];

starter=1;
ender=sample_rate;

for i=1:length(stimulus_times)
    data_window=ep(starter:ender);

    [peak, index]=max(data_window);
    index=index/sample_rate; %divide the index by the sampling rate and that should be the time in seconds
    peak_locs_ep(i)=index;
    starter=ender+1;
    ender=ender+sample_rate;
end

%Calculate the mean latency
mean_latency_time= mean(peak_locs_ep)
```

```
mean_latency_time =

    0.1622
```

The mean latency time is calculated to be 0.162 seconds from the code above.

3. In neuroscience, we often need to isolate a small neural signal buried under an appreciable amount of noise. One technique to accomplish this is called the spike triggered average, sometimes called signal averaging. This technique assumes that the neural response to a repetitive stimulus is constant (or nearly so), while the noise fluctuates from trial to trial - therefore averaging the evoked response over many trials will isolate the signal and average out the noise. Construct a spike triggered average plot for the data in `I521_A0001_D003`. Plot the average EP in red. Using the commands `hold on` and `hold off` as well as `errorbar` and `plot`, overlay error bars at each time point on the plot to indicate the standard deviation of the responses at any given time point. Plot the standard deviation error bars in gray (RGB value: [0.7 0.7 0.7]). Make sure to give a proper legend along with your labels. (4 pts)

```
%Create Spike Triggered Average Plot
avg_mat=[];

starter=1;
ender=sample_rate;

for i=1:length(stimulus_times)
    data_window=ep(starter:ender);
    avg_mat(i,:)=data_window;
    starter=ender+1;
    ender=ender+sample_rate;
end

sized=size(avg_mat);
averaged=[];

%take the average of each column in the avg_mat matrix
aver=mean(avg_mat,1);

%compute the standard deviation of each column for error bars
std_dev=std(avg_mat,1);

%Create Plot
err=std_dev;
x_values=0:1/sample_rate:1-(1/sample_rate);
figure
errorbar(x_values,aver,err,'Color',[0.7 0.7 0.7])
hold on
plot(x_values,aver,'r')
xlabel('Time (Seconds)')
ylabel('Amplitude (uV)')
legend('Error Bar (Std Dev)','Average EP')
title('Spike Triggered Average Plot')
```



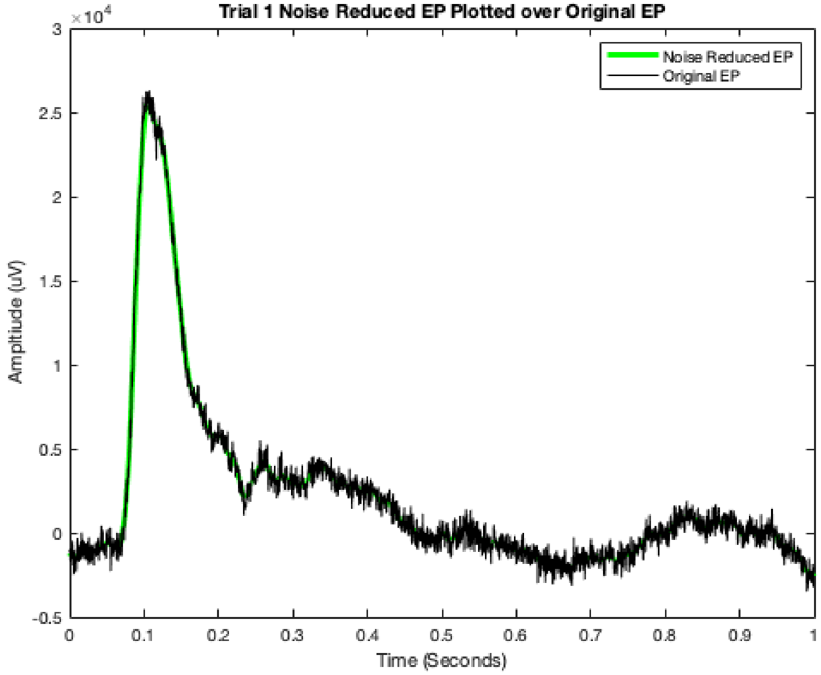
4. (a) We often want to get a sense for the amplitude of the noise in a single trial. Propose a method to

do this (there are a few reasonably simple methods, so no need to get too complicated). Note: do not assume that the signal averaged EP is the “true” signal and just subtract it from that of each trial, because whatever method you propose should be able to work on the signal from a single trial or from the average of the trials. (4 pts)

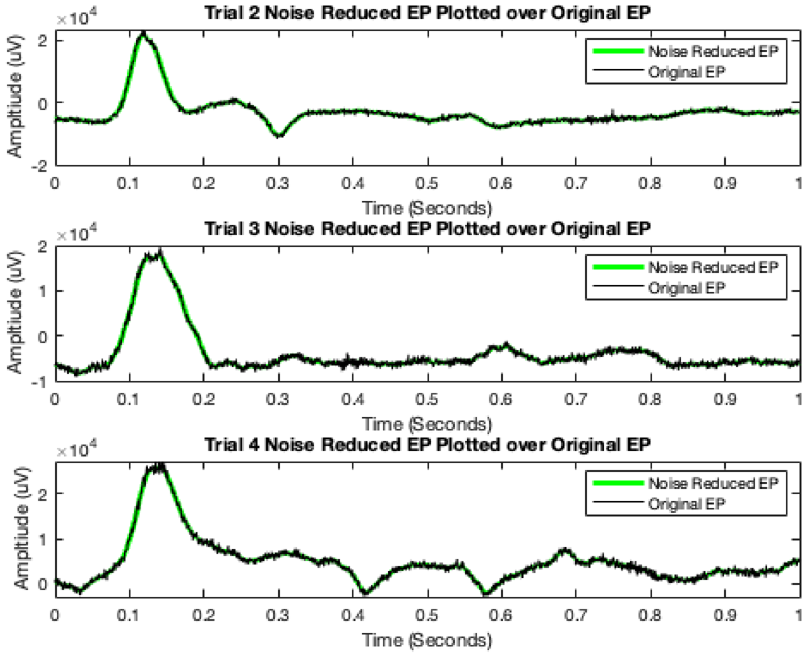
In order to reduce noise one method would be to take a moving average of the amplitude at each data point of a certain window size (this method would use the amplitudes of the 30 closest datapoints around each datapoint in the non-noise reduced signal and average them) for each trial and then store that as your noise reduced signal. Assuming that the noise is equidistant in ampltitude above and below the underlying signal this moving average will reduce noise and the resulting curve will be noise reduced. To quantify noise in each trial, the noise reduced signal can be subtracted from the non-reduced signal at each data point and the average of those resulting datapoints can give you an idea for the average noise in each trial. The average of the noise in each trial across all trials will give you an idea for the overall noise content across all trials.

(b) Show with a few of the EPs (plots and/or otherwise) that your method gives reasonable results. (1 pt)

```
%First trial noise filtering
starter_noise_red=1;
ender_noise_red=sample_rate;
one_trial=ep(starter_noise_red:ender_noise_red);
moving_avg_mat=movmean(one_trial,30);
figure
plot(x_values,moving_avg_mat,'Linewidth', 3, 'Color','g')
xlabel('Time (Seconds)')
ylabel('Ampltiude (uV)')
title('Trial 1 Noise Reduced EP Plotted over Original EP')
hold on
plot(x_values,one_trial,'Color','k')
legend('Noise Reduced EP','Original EP')
```



```
%Second EP Trial Noise Filtering

starter=ender+1;
ender=ender+sample_rate;

starter_noise_red_2=ender_noise_red+1;
ender_noise_red_2=ender_noise_red+sample_rate;
two_trial=ep(starter_noise_red_2:ender_noise_red_2);
moving_avg_mat_2=movmean(two_trial,30);
figure
subplot(3,1,1)
plot(x_values,moving_avg_mat_2,'Linewidth', 3, 'Color','g')
xlabel('Time (Seconds)')
ylabel('Ampltiude (uV)')
title('Trial 2 Noise Reduced EP Plotted over Original EP')
hold on
plot(x_values,two_trial,'Color','k')
legend('Noise Reduced EP','Original EP')
hold off

%Third EP Trial Noise Filtering
starter_noise_red_3=ender_noise_red_2+1;
ender_noise_red_3=ender_noise_red_2+sample_rate;
three_trial=ep(starter_noise_red_3:ender_noise_red_3);
moving_avg_mat_3=movmean(three_trial,30);
subplot(3,1,2)
plot(x_values,moving_avg_mat_3,'Linewidth', 3, 'Color','g')
xlabel('Time (Seconds)')
ylabel('Ampltiude (uV)')
title('Trial 3 Noise Reduced EP Plotted over Original EP')
hold on
plot(x_values,three_trial,'Color','k')
legend('Noise Reduced EP','Original EP')
hold off

%Fourth Trial Noise Filtering
starter_noise_red_4=ender_noise_red_3+1;
ender_noise_red_4=ender_noise_red_3+sample_rate;
four_trial=ep(starter_noise_red_4:ender_noise_red_4);
moving_avg_mat_4=movmean(four_trial,30);
subplot(3,1,3)
plot(x_values,moving_avg_mat_4,'Linewidth', 3, 'Color','g')
xlabel('Time (Seconds)')
ylabel('Ampltiude (uV)')
title('Trial 4 Noise Reduced EP Plotted over Original EP')
hold on
plot(x_values,four_trial,'Color','k')
legend('Noise Reduced EP','Original EP')
hold off
```



The plots above show that this method gives reasonable results.

(c) i. Apply your method on each individual trial and report the mean noise amplitude across all trials. (1 pt)

```
%Calculate mean noise amplitude across all trials

starter=1;
ender=sample_rate;
noise_vec=[];


for i=1:length(stimulus_times)
    data_window=ep(starter:ender);
    moving_avg=movmean(data_window,30);
    diff=mean(abs(moving_avg-data_window));
    noise_vec(i)=diff;

    starter=ender+1;
    ender=ender+sample_rate;
end

mean_noise_all_trials=mean(noise_vec)
mean_noise_trial_1=noise_vec(1);
```

```
mean_noise_all_trials =

  384.9384
```

As shown above, the mean noise amplitude across all trials using this moving average method was found to be 384.94 uV.

ii. Apply your method on the signal averaged EP and report its noise. (1 pt)

```
%Calculate noise for signal averaged EP
signal_avg_mov_avg=movmean(aver,30);
signal_averaged_noise=mean(abs(aver-signal_avg_mov_avg))
```

```
signal_averaged_noise =

   44.9340
```

As shown by the output of the code above the noise of the signal average EP was found to be 44.93 uV.

iii. Do these two values make sense? Explain. (1 pt)
These two values do make sense because the signal averaged EP has a much lower noise content, at only 44.93 uV, compared to the higher noise content of the mean amplitude noise across all trials, which was 384.94 uV. This makes sense because the signal averaged EP has already been averaged previously before noise was quantified, and this first averaging eliminated some of the noise already.