Jason Wong 1090292616 13/12/17

load ECoG\_data.mat

**%Exercise 3.1.1**

%To find the period of time between each samples, use diff(times)

period = diff(times);

%To find the average sampling period, use mean(period)

avgPeriod = mean(period);

%To find the sampling rate in Hz, use 1/avgPeriod

samplingRate = 1/avgPeriod;

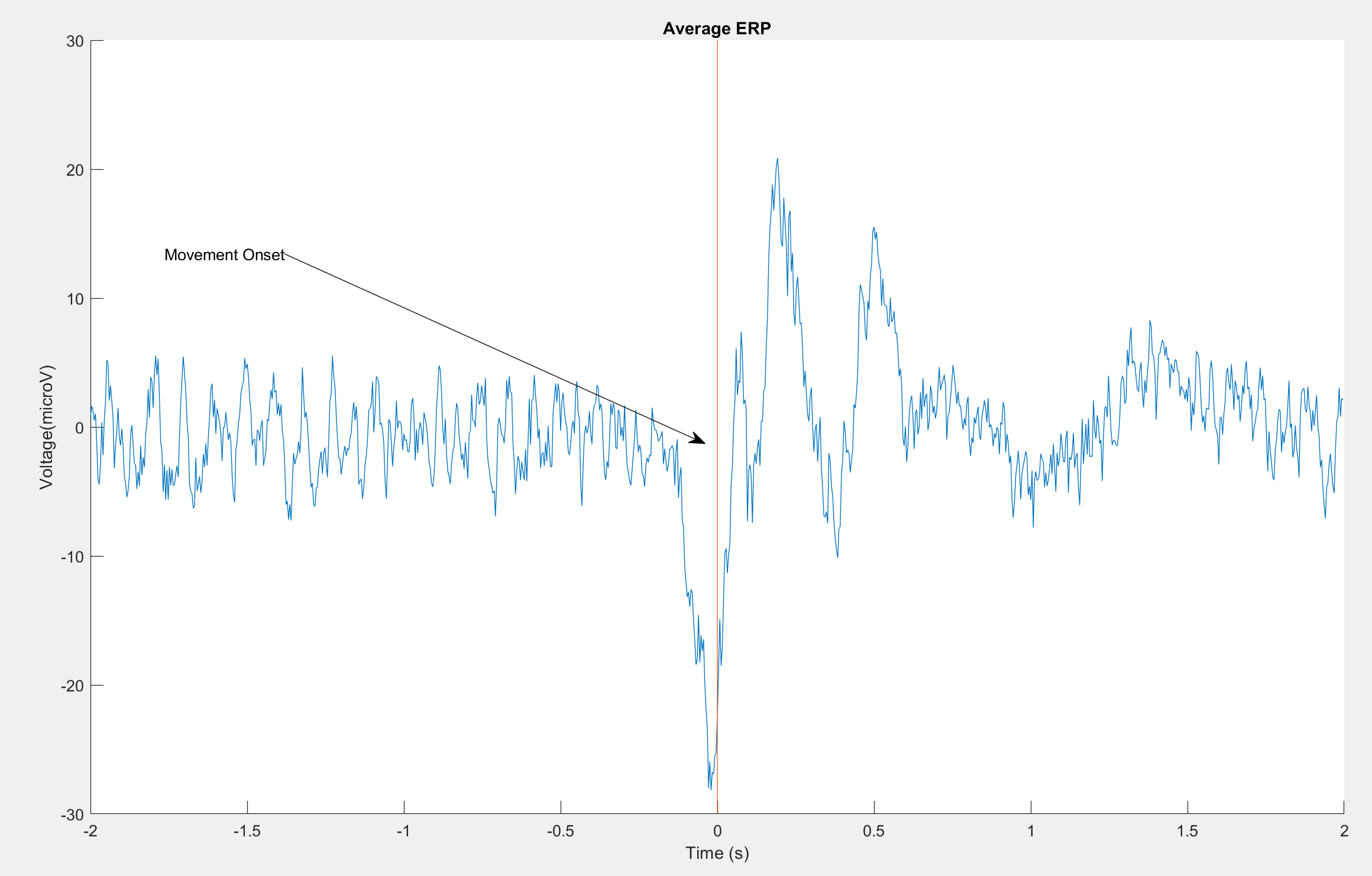
**%samplingRate = 250Hz**

**%Exercise 3.1.2**

%To find data for Channel 18, use ECoG(:,18)

%Code has been adapted from Zenas Chao's Week 7 lesson

%In this graph you can see that the neuron changes voltage before the monkey even moves



figure(1);

dataChannel = ECoG(:,18);

dataChannel = dataChannel'; %Make Channel 18 a horizontal array instead a vertical matrix

Num\_trial = 133; %Number of trials is 133

ERP=[]; % Step1 : declare an empty matrix

for tr=1:Num\_trial % Step2 : a FOR loop to segment each trial

Cue\_time=Movement\_onset(tr);

Trial\_index=find(times>=Cue\_time-2 & times<Cue\_time+2); %Includes 2 seconds before and after onset

ERP=[ERP ; dataChannel(Trial\_index)]; % Step3: append data to the matrix

end

avgERP = mean(ERP)\*1000;

hold on

plot(times(Trial\_index)-Cue\_time, avgERP)

plot([0 0], [-30 30]) %Plots the line of movement onset

title('Average ERP')

ylabel('Voltage(microV)');

xlabel('Time (s)');

x = [0.25 0.51];

y = [0.7 0.5];

annotation('textarrow',x,y,'String','Movement Onset') %Labels the movement onset

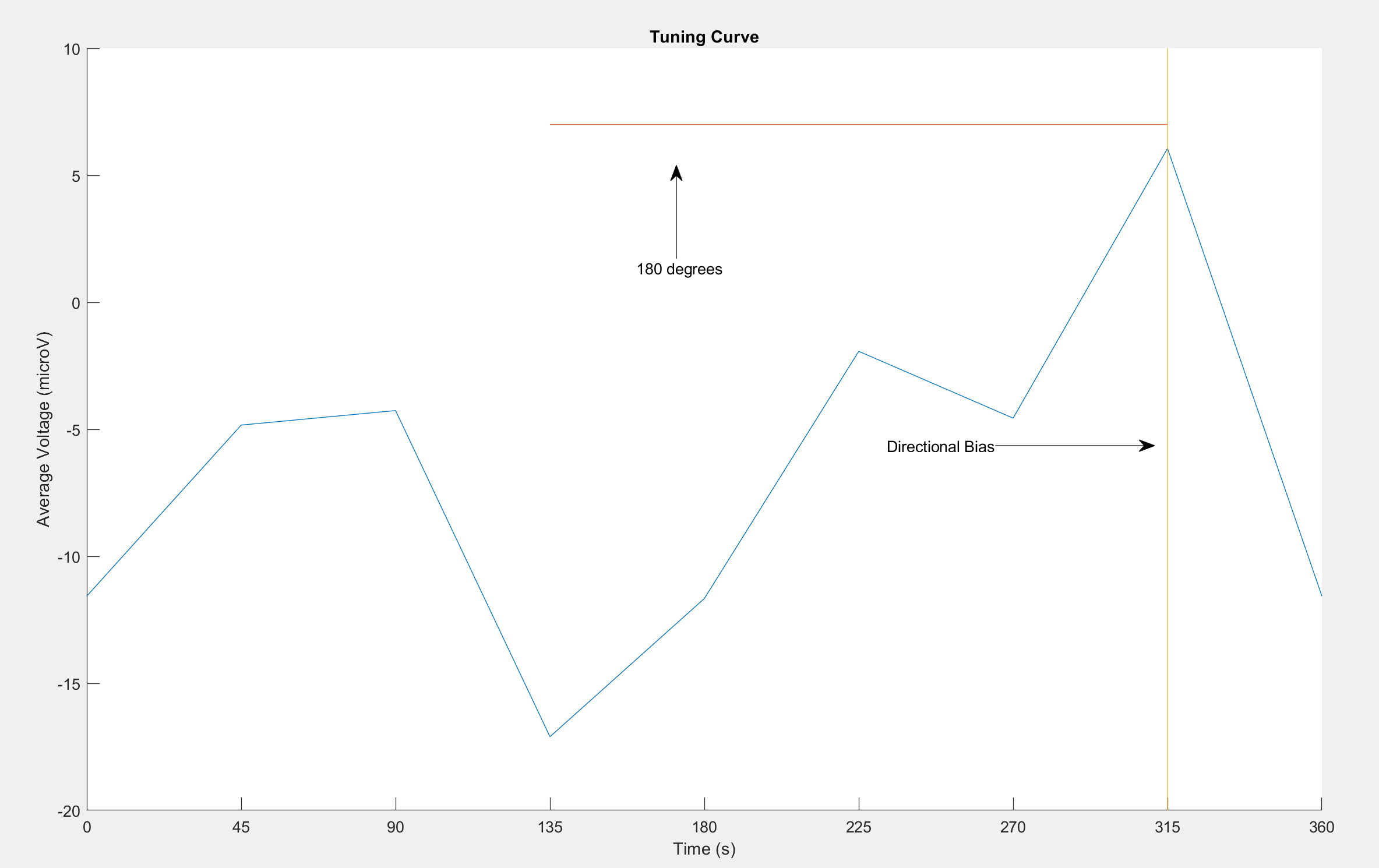
hold off

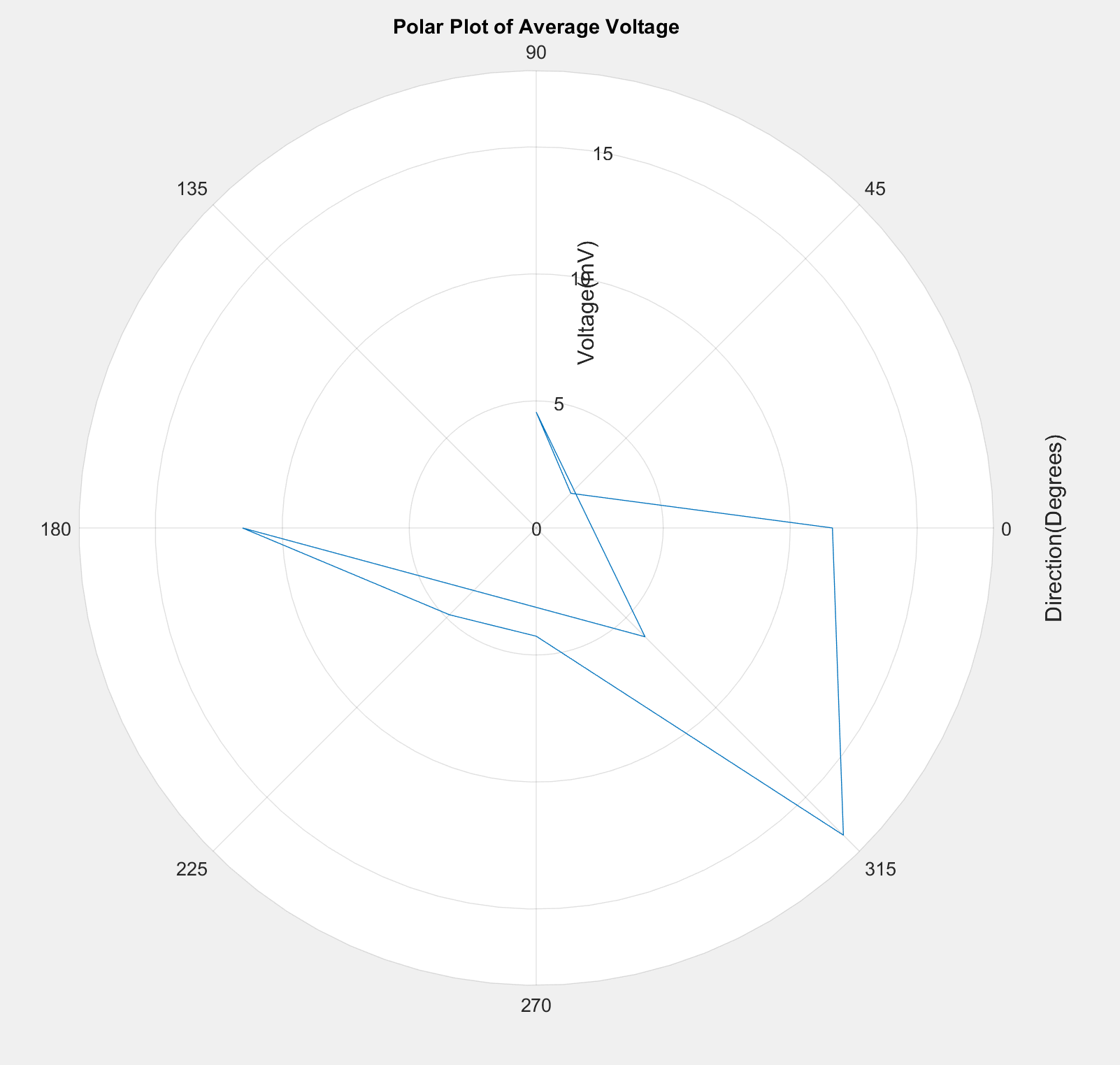
**%Exercise 3.2.1**

%Code adapted from Zenas Chao's Week 7 lecture

%Using Average Voltage as my measurement

%In the tuning curve, a clear peak and trough can be seen, both 180 degrees apart



%There is an obvious directional tuning towards 315 degrees. This will be explained in exercise 3.2.3

load LFP\_data.mat

Num\_trial = 463;

ERP=[]; % Step1 : declare an empty matrix

for tr=1:Num\_trial % Step2 : a FOR loop to segment each trial

Cue\_time=instruction(tr);

Trial\_index=find(times>=Cue\_time-0.5 & times<Cue\_time+2);

ERP=[ERP ; lfp(Trial\_index)]; % Step3: append data

end

ERP\_direction=[]; % Step1 : declare an empty matrix

for d=1:8 % Step2 : a FOR loop to find each direction

idx=find(direction==d);

ERP\_direction=[ERP\_direction; mean(ERP(idx,:),1)]; % Step3: append ERP

end

figure(2)

xticks(0:45:360)

avgVoltage = mean(ERP\_direction,2); %This finds the average voltage for the direction

hold on

%This code plots a tuning graph based on the average voltage for each

%direction

angles =[0:45:360];

voltages = [];

for d=0:8

voltages = [voltages avgVoltage(mod(d,8)+1)]

end

plot(angles,voltages);

title('Tuning Curve')

xlabel('Time (s)')

ylabel('Average Voltage (microV)')

plot([315 135], [7 7]); %Plots the distance between 180 degrees on the graph

annotation('textarrow',[0.5 0.5],[0.7 0.8],'String','180 degrees') %Labels the line

axis([0 360 -20 10])

plot([315 315], [-20 10]); %Plots the bias

annotation('textarrow',[0.7 0.8],[0.5 0.5],'String','Directional Bias') %Labels the line

figure(3)

ax = polarplot(voltages);

ax = gca

ax.RAxis.Label.String = 'Voltage(mV)';

ax.ThetaAxis.Label.String = 'Direction(Degrees)';

title('Polar Plot of Average Voltage')

ax.ThetaTick = 0:45:360;

%Exercise 3.2.3

%Yes, direction tuning can be seen with a bias towards the 315 degrees

%direction, this may be because the level must be pushed in the South East

%direction, or maybe the lever was to the bottom right of the monkey's

%vision

%the neuron has direction tuning to 315 degrees