clear all

close all

clc

%%

% Load each file and extract epochs, put in relevant structures

%

% Define the EEG file(s) of interest

fname = {'data\_files/c30o30\_s4\_t1.bdf', 'data\_files/c30o30\_s4\_t2.bdf'};

% Initialise epoch counter

N\_epochs = 0;

for fi=1:length(fname)

% Load the EEG file (fname{fi}). Note the semicolon to supress echo

% of S and HDR to the command window. Don't worry about warnings

% concerning overflow detection or the lack of a status channel.

%

% For more info type:

%

% >> help sload

%

[S HDR] = sload(fname{fi});

Fs=HDR.SampleRate; %if you open the HDR structure you will find the samplerate. In this way, even if you have data with different s.r. your code will work anyway

% for i=1:17

% figure

% plot(S(:,i))

% end

% Implements supplementary\_code/part\_2/extract\_epochs.m

% Find length of signal

L = length(S);

% Differentiate trigger channel

delta\_trig = S(2:L,17)-S(1:L-1,17);

% Look for rising edges

trig = find(delta\_trig>0);

% Debounce if required

% I.e. remove spurious short triggers, if any

% these can occur when using a manual switch-based trigger

% e.g. look at the trigger channel in 'data\_files/c30o30\_s4\_t1.bdf'

T\_debounce = 0.5; % (in seconds) - set as desired

dt\_trig = trig(2:end) - trig(1:end-1);

delete\_short\_trigs = find(dt\_trig < Fs\*T\_debounce) + 1;

trig(delete\_short\_trigs) = [];

% How many triggers were extracted in total, after debouncing?

N\_trigs = length(trig)-1;

% Ensure we have an odd number of triggers (ignore final incomplete

% epoch if not).

% Trigger 1 = start of experiment

% Triggers 2, 4, 6, 8... (2k) = eyes closed

% Triggers 3, 5, 7, 9... (2k+1) = eyes open

if mod(N\_trigs, 2) == 0

N\_trigs = N\_trigs - 1;

end

disp(cell2mat(strcat({'Found '}, int2str(N\_trigs), {' triggers.'})));

N\_epochs = N\_trigs;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Extract triggers

% throw an error % insert your own code here to extract the triggers

% (see previous part of the lab)

% Discard 2 seconds of data before & after each cue

discard = 2\*Fs;

% Extract each epoch of data and store in the relevant structure for

% each class

% - ec: Eyes closed

% - eo: Eyes open

this\_epochs = 0;

Epoch\_length = 24; % in seconds

samples = Epoch\_length\*Fs;

for i=1:N\_trigs/2

ec\_extracted{i} = S(trig(2\*i)+discard:trig(2\*i+1)-discard,1:16);

eo\_extracted{i} = S(trig(2\*i+1)+discard:trig(2\*i+2)-discard,1:16);

maxlen = min(length(ec\_extracted{i}), length(eo\_extracted{i}));

N\_loop = floor(maxlen / samples);

for j = 1:N\_loop

N\_epochs = N\_epochs + 1;

this\_epochs = this\_epochs + 1;

ec{N\_epochs} = ec\_extracted{i}((j-1)\*samples+1:j\*samples,:);

eo{N\_epochs} = eo\_extracted{i}((j-1)\*samples+1:j\*samples,:);

end

end

disp(cell2mat(strcat({'Extracted '}, int2str(this\_epochs), {', '},...

int2str(Epoch\_length), {' second epochs of each class from '})));

end

disp(cell2mat(strcat({'Extracted a total of '}, int2str(N\_epochs),...

{', '}, int2str(Epoch\_length), {' second epochs of each class.'})));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Plot FFT of raw data eo = eyes open

%% eo\_fft = fft(eo{N\_epochs});

eo\_fft = ssfft(eo{1}); %you have to use the ssfft function. Keep in mind that you have 5 epochs.

ec\_fft = ssfft(ec{1}); Unless you did it already in the previous script..

% Implements supplementary\_code/part\_3/calc\_psd.m

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Pre-filter data and calculate PSD for every epoch and every electrode.

% The PSD should be resampled into 2Hz ‘bins’

% Band pass 2-40 Hz (use the filter you designed earlier)

b\_bp = band\_pass(Fs);

% Iterate through every extracted epoch

for i = 1:N\_epochs %% you are opening 2 for loops here

% Iterate through each electrode channel (Ei)

for Ei = 1:16 %%

% Apply bandpass filter to eyes open

eo\_bp(:,:,Ei) = filter(b\_bp, 1, eo{i}(:,Ei));

%??? NOW FILTER EYES CLOSED ???%

ec\_bp(:,:,Ei) = filter(b\_bp, 1, ec{i}(:,Ei));

% Calculate PSD for Ei using the Welch method

[psd freqs] = pwelch(eo\_bp(:,:,Ei), [], [], [], Fs);

N\_psd = length(psd) - 1;

temp\_psd = reshape(psd(1:N\_psd), N\_psd/(Fs / (2 \* 2)), []); %psd without\_eo

eo\_psd\_2hz(:,i,Ei) = mean(temp\_psd,1);

[psd freqs] = pwelch(ec\_bp(:,:,Ei), [], [], [], Fs);

N\_psd = length(psd) - 1;

temp\_psd = reshape(psd(1:N\_psd), N\_psd/(Fs / (2 \* 2)), []);

ec\_psd\_2hz(:,i,Ei) = mean(temp\_psd,1);

end %% you have to remember to close them as well and be sure you are using them correctly!

You did the part for eo but skipped the ec.

end

%point 5

% Calculate mean and standard deviation across all epochs of the PSD!!

%% This means also that you have to go through all the Channels. So, considering that the previous loop is closed you have to open a new one, hence:

for Ei = 1:16

eo\_psd\_2hz\_mean(:,Ei) = mean(eo\_psd\_2hz(:,i,Ei));

eo\_psd\_2hz\_std(:,Ei) = abs(eo\_psd\_2hz(:,i,Ei) - eo\_psd\_2hz\_mean(:,Ei));

plot(eo\_psd\_2hz\_std(:,Ei))

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% This point was before but ok

% Plot FFT of raw data eo = eyes open

% eo\_fft = fft(eo{N\_epochs}); %% look comment above

% Electrode montage on scalp

% Use your previous code to plot this on the entire montage

% Specify the electrode montage on scalp (the electrode numbers laid out

% in a matrix, according to their positions on the scalp and as we want

% them to be displayed

montage = [-1 -1 1 -1 -1; 2 3 4 5 6; 7 8 9 10 11; 12 13 14 15 16];

% The 10/20 labels for each electrode channel (again, corresponding to the

% montage above. In this case electrode 1 = Fz, electrode 2 = FC3,

% electrode 3 = FC1 etc.

electrode\_labels = {'Fz', 'FC3', 'FC1', 'FCz', 'FC2', 'FC4','C3', 'C1', 'Cz', 'C2', 'C4','CP3', 'CP1', 'CPz', 'CP2', 'CP4', 'P3', 'P1', 'PZ', 'P2', 'P4'}; %% You have 21 electrodes!!!

% Create a new figure (or select an existing one if you prefer)

figure

for j=1:4

for i=1:5

% Only create plots for electrodes that exist (ignore -1 values)

if montage(j, i) > 0

% which electrode is at this location in the montage matrix?

electrode = montage(j, i);

% select the correct (row, colum) --> what does the number 5 do

% here?

subplot(4, 5, i+(5\*(j-1)))

% plot something for the current electrode. You can plot

% whatever data you want here, but for example:

plotFFT2(eo\_fft(:,electrode), Fs)

% Set axes limits if necessary, e.g.

ylim([0 2000]) %%check limits

xlim([0 40])

% Label each subplot with the corresponding electrode poistion

title(electrode\_labels{electrode})

% Label the axes for the bottom left plot only (to prevent the

% figure getting too crowded and illegible)

if i == 1 && j == 4

xlabel('Frequency (Hz)');

ylabel('FFT (V)')

end

end

end

end

% Add a title to the entire plot - play around, as required

annotation('textbox', [0.1,0.75, 0.2, 0.2], 'String', 'Eyes Open','FontWeight', 'bold', 'LineStyle', 'none')

%%here (in theory) you plot the eyes open, then you have to do it again for eyes closed.