-- % % Video 2: Electrode locations in 2D and 3D % %
-- %

Load the data

```
% load data
load SSVEPdata.mat


% inspect the structure
whos
```

```
  Name      Size              Bytes  Class     Attributes

  EEG       1x1           156572974  struct    global
```

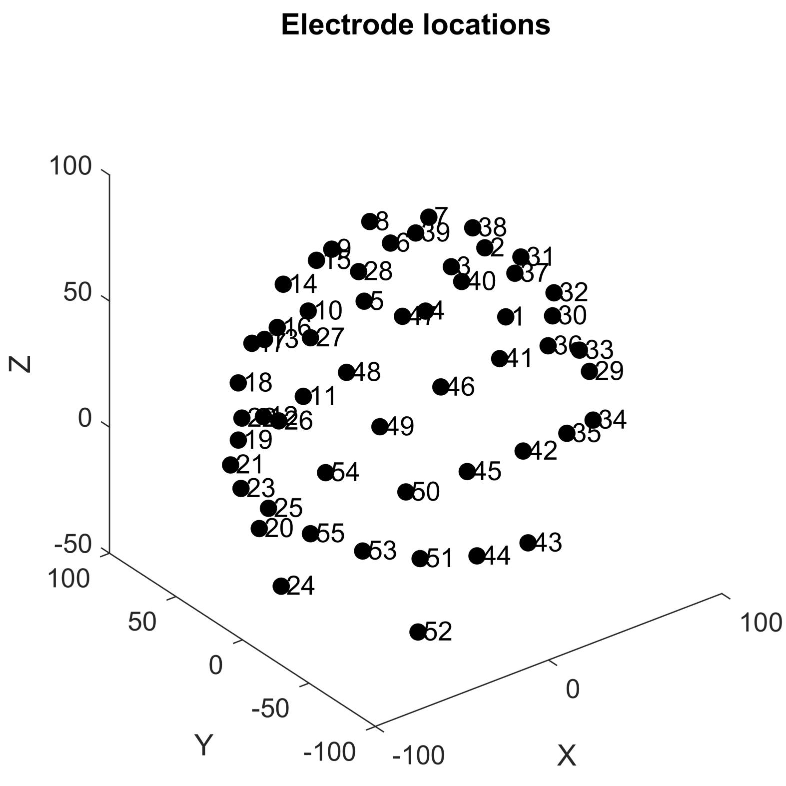
```
EEG
```

```
EEG = struct with fields:
             setname: 'raw epochs resampled'
            filename: ''
            filepath: ''
             subject: ''
               group: ''
           condition: ''
             session: []
            comments: []
              nbchan: 55
              trials: 346
                pnts: 2048
               srate: 512
                xmin: -1.5000
                xmax: 2.4990
               times: [-1500 -1.4980e+03 -1.4961e+03 -1.4941e+03 -1.4922e+03 -1.4902e+03 -1.4883e+03 -1.4863e+03 -1.
                data: [55×2048×346 single]
              icaact: []
             icawinv: []
           icasphere: []
          icaweights: []
         icachansind: []
            chanlocs: [1×55 struct]
          urchanlocs: []
            chaninfo: [1×1 struct]
                 ref: 'averef'
               event: []
             urevent: []
    eventdescription: {}
               epoch: [1×346 struct]
    epochdescription: {}
              reject: []
               stats: []
            specdata: []
          specicaact: []
          splinefile: ''
       icasplinefile: []
              dipfit: []
             history: []
```

```
            saved: 'no'
              etc: []
```

plot of electrode locations in 3D

```
figure(1), clf

% plot the electrode positions in 3D
plot3([EEG.chanlocs.X],[EEG.chanlocs.Y],[EEG.chanlocs.Z],'ko','markerfacecolor','k')

% draw the text labels
hold on
for i=1:EEG.nbchan
    text(EEG.chanlocs(i).X+3,EEG.chanlocs(i).Y,EEG.chanlocs(i).Z,num2str(i))
end

% make the plot look nicer and more interactive
xlabel('X'), ylabel('Y'), zlabel('Z')
title('Electrode locations')
rotate3d on
axis square
```



plot of electrode locations in 2D

```
% plot ERPs for dimension-specific averaging
figure(2), clf
```

```
% show an empty topoplot
topoplotIndie(zeros(EEG.nbchan,1),EEG.chanlocs,'electrodes','numbers');
title('2D topographical map')
```

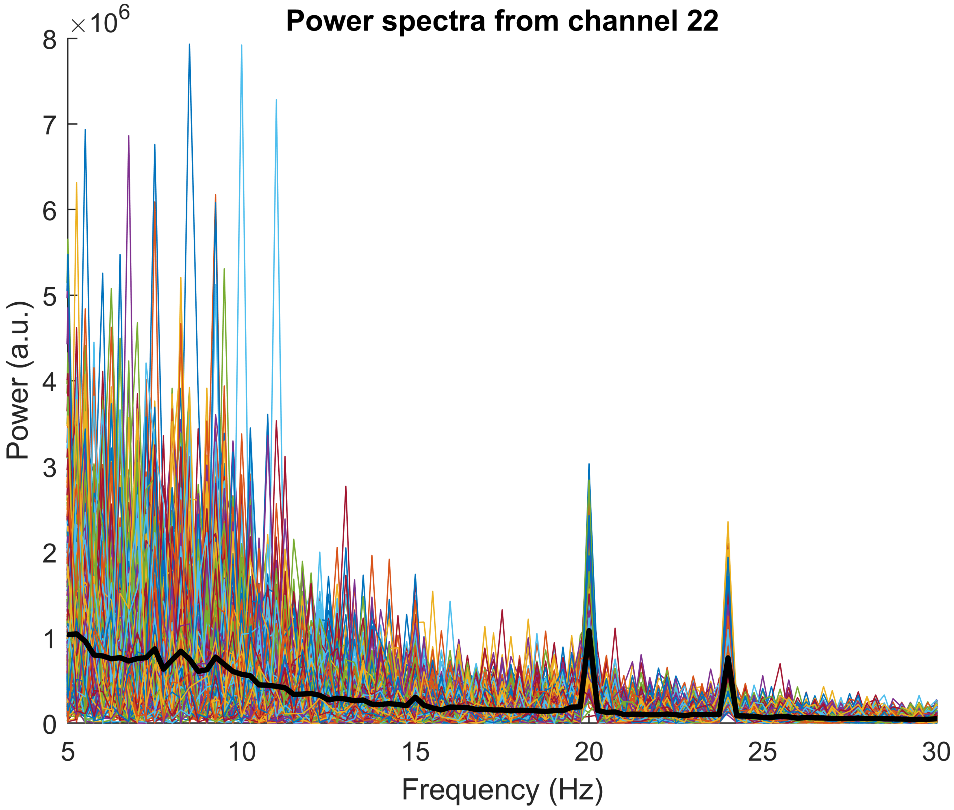
--- % % Video 3: Spectral analysis via the FFT % %
--- %

spectral analysis

```
% soft-code a channel to plot
chan2plot = 22;


% FFT of one channel
channelPower = zeros(EEG.pnts,EEG.trials);
for triali=1:EEG.trials
    channelPower(:,triali) = abs(fft(EEG.data(chan2plot,:,triali))).^2;
end

% without a loop
channelPower = squeeze(abs(fft(EEG.data(chan2plot,:,:),[],2)).^2);

% vector of frequencies
hz = linspace(0,EEG.srate/2,floor(EEG.pnts/2)+1);
```

visualization

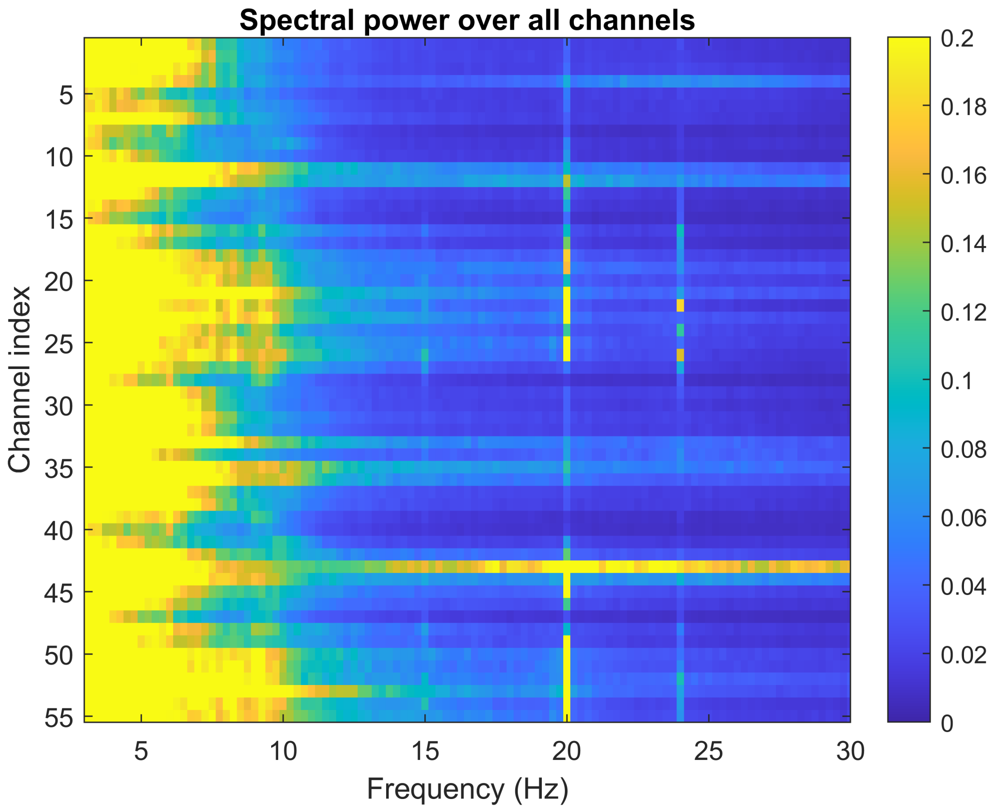
```
figure(3), clf, hold on
h = plot(hz,channelPower(1:length(hz),:));
plot(hz,mean(channelPower(1:length(hz),:),2),'k','linew',2)

% set all individual lines to black
%set(h,'color',.8*ones(3,1))

% pretty the plot
set(gca,'xlim',[5 30])
xlabel('Frequency (Hz)')
ylabel('Power (a.u.)')
title([ 'Power spectra from channel ' num2str(chan2plot) ])
```



--- % % Video 4: Image of channel spectra % %
--- %

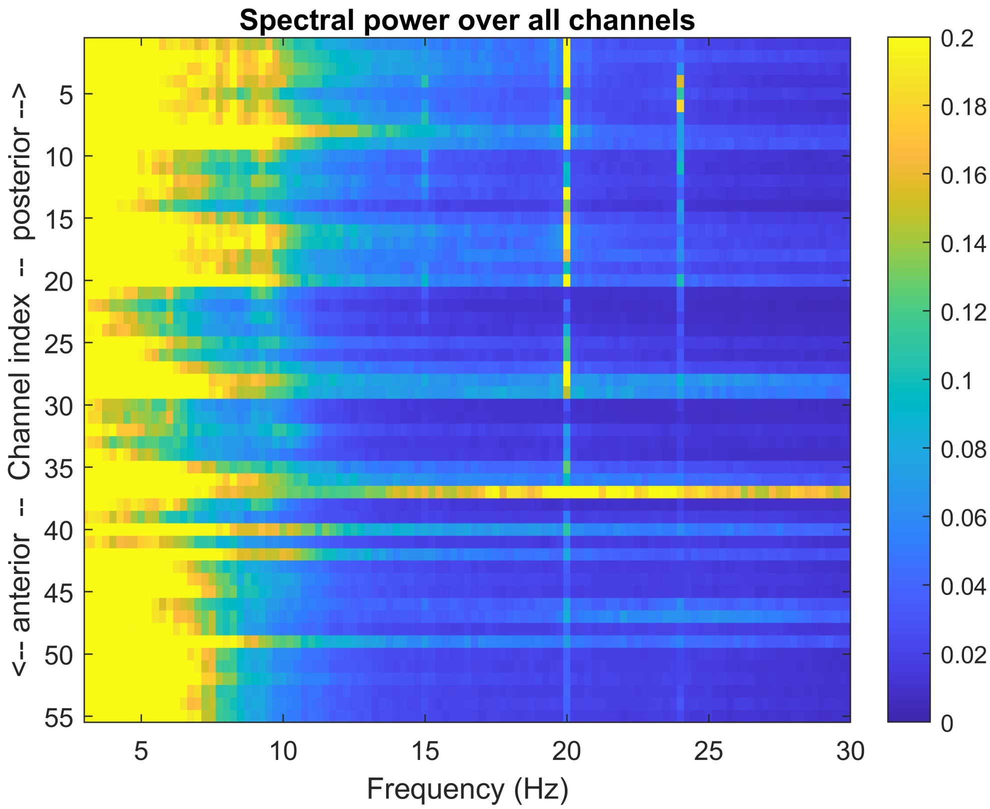
all channel spectra

```
% FFT of all channels at the same time,
% and then average over all trials.
allChannelPower = mean(abs(fft(EEG.data,[],2)/EEG.pnts).^2,3);

% and show in an image
figure(4), clf
imagesc(hz,[],allChannelPower(:,1:length(hz)))
set(gca,'xlim',[3 30],'clim',[0 .2])
xlabel('Frequency (Hz)')
ylabel('Channel index')
title('Spectral power over all channels')
colorbar
```



now sort by channel X-coordinate

```
% we only need the sorting index, not the sorted values
[~,sortXidx] = sort([EEG.chanlocs.X]);

% same plot as above
figure(5), clf
imagesc(hz,[],allChannelPower(sortXidx,1:length(hz)))
set(gca,'xlim',[3 30],'clim',[0 .2])
xlabel('Frequency (Hz)')
ylabel('<-- anterior  --  Channel index  --  posterior -->')
title('Spectral power over all channels')
colorbar
```

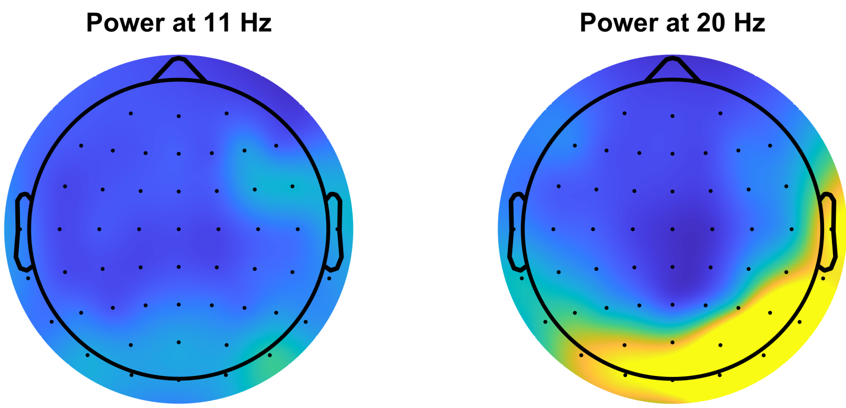


-- % % Video 5: Topographical maps % %
-- %

topoplot of 20 and 24 Hz activity

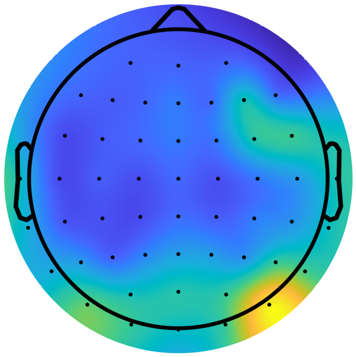
```
% find frequency boundaries
hzidx(1) = dsearchn(hz',11);
hzidx(2) = dsearchn(hz',20);


figure(6), clf
for i=1:2
    % specify the subplot
    subplot(1,2,i)

    % call the topographical map function
    topoplotIndie(allChannelPower(:,hzidx(i)),EEG.chanlocs,'numcontour',0);

    % set colorlimit and write the title
    set(gca,'clim',[0 .3])
    title([ 'Power at ' num2str(hz(hzidx(i))) ' Hz' ])
end
```

```
% explore some colormaps
colormap parula
```



--- % % Video 6: Endogenous alpha % %
--- %

```
% frequency boundaries
freqrange = [8 12];
alphaidx = dsearchn(hz',freqrange');

% topoplot of alpha over all time points
figure(7), clf
subplot(121)
topoplotIndie(mean(allChannelPower(:,alphaidx(1):alphaidx(2)),2),EEG.chanlocs,'numcontour',0);
title('Raw alpha power')
```

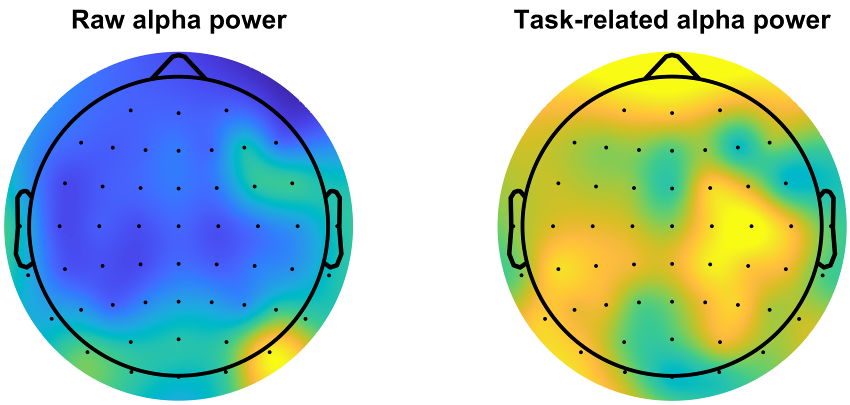


alpha change from baseline

```
% define time window and convert to indices
timerange = [ -1000 0 1000 ];
timeidx = dsearchn(EEG.times',timerange');

% compute new power matrices
powerPreStim = mean(abs(fft(EEG.data(:,timeidx(1):timeidx(2),:),EEG.pnts,2)/EEG.pnts).^2,3);
powerPstStim = mean(abs(fft(EEG.data(:,timeidx(2):timeidx(3),:),EEG.pnts,2)/EEG.pnts).^2,3);


% alpha power effect
alphaPstVsPre = 10*log10( mean(powerPstStim(:,alphaidx(1):alphaidx(2)),2) ./ ...
    mean(powerPreStim(:,alphaidx(1):alphaidx(2)),2) );
```

show the topographical map

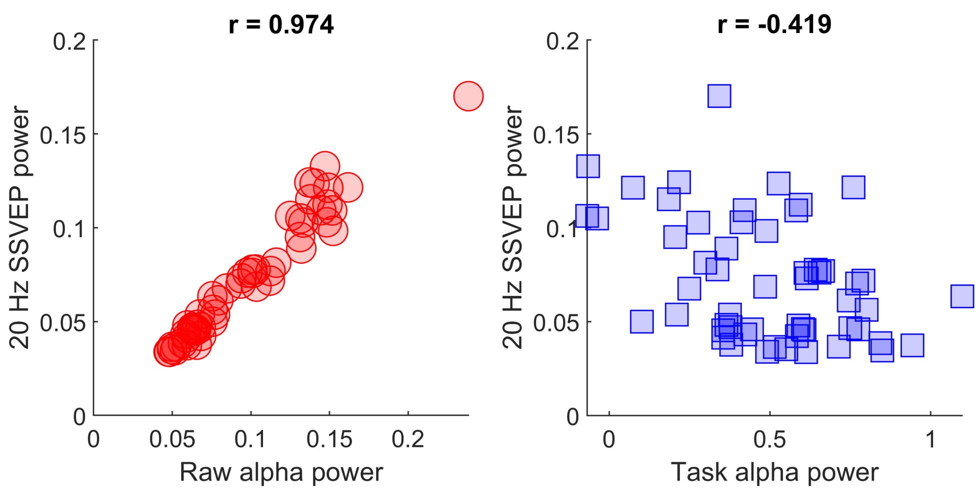
```
subplot(122)
topoplotIndie(alphaPstVsPre,EEG.chanlocs,'numcontour',0);
set(gca,'clim',[-1 1])
title('Task-related alpha power')
```



-- % % Video 7: Correlate alpha with SSVEP % %
-- %

```
figure(8), clf
subplot(121)

% create temp variables for convenience
x = mean(allChannelPower(:,alphaidx(1):alphaidx(2)),2);
y = allChannelPower(:,hzidx(1));

% scatter plot
scatter(x,y,120,'ro','markerfacecolor','r','markerfacealpha',.2)
axis square
xlabel('Raw alpha power')
ylabel('20 Hz SSVEP power')

% compute correlation and show in the title
r = corrcoef(x,y);
title([ 'r = ' num2str(r(2),3) ])


% more temp variables (x here is short so left out)
y = allChannelPower(:,hzidx(1));
```

```
% scatter plot
subplot(122)
scatter(alphaPstVsPre,y,120,'bs','markerfacecolor','b','markerfacealpha',.2)
axis square
xlabel('Task alpha power')
ylabel('20 Hz SSVEP power')

% correlation and title
r = corrcoef(alphaPstVsPre,y);
title([ 'r = ' num2str(r(2),3) ])
```



done.