Table of Contents

Please note that this project is from a course by Mike x Cohen

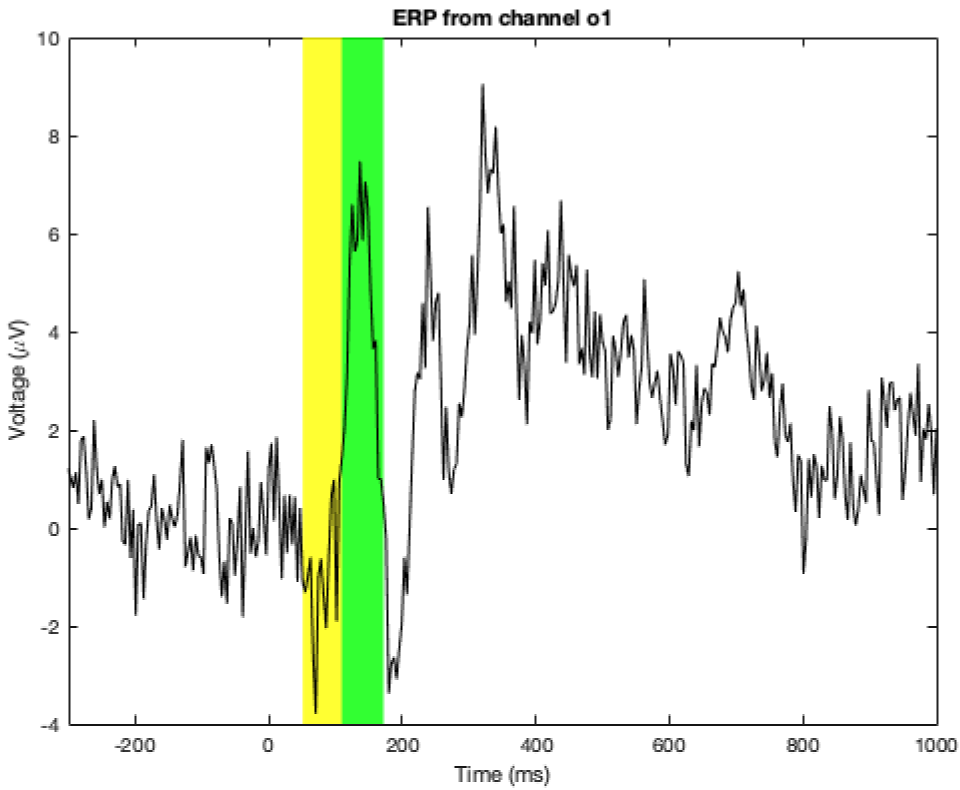
Load data

```
load sampleEEGdata.mat

% channel to pick
chan2use = 'o1';

% time window for negative peak
negpeaktime = dsearchn(EEG.times',[  50 110 ]')';
pospeaktime = dsearchn(EEG.times',[ 110 170 ]')';


% find channel index
chanidx = strcmpi({EEG.chanlocs.labels},chan2use);
```

compute ERP

```
erp = double( mean(EEG.data(chanidx,:,:),3) );

% plot ERP
figure(1), clf
plot(EEG.times,erp,'k','linew',1)
set(gca,'xlim',[-300 1000])


% plot patches over areas
ylim = get(gca,'ylim');
ph = patch(EEG.times(negpeaktime([1 1 2 2])),ylim([1 2 2 1]),'y');
set(ph,'facealpha',.8,'edgecolor','none')

ph = patch(EEG.times(pospeaktime([1 1 2 2])),ylim([1 2 2 1]),'g');
set(ph,'facealpha',.8,'edgecolor','none')

% move the patches to the background
set(gca,'Children',flipud( get(gca,'Children') )) %changes the colored
 boxes to be in the background
```

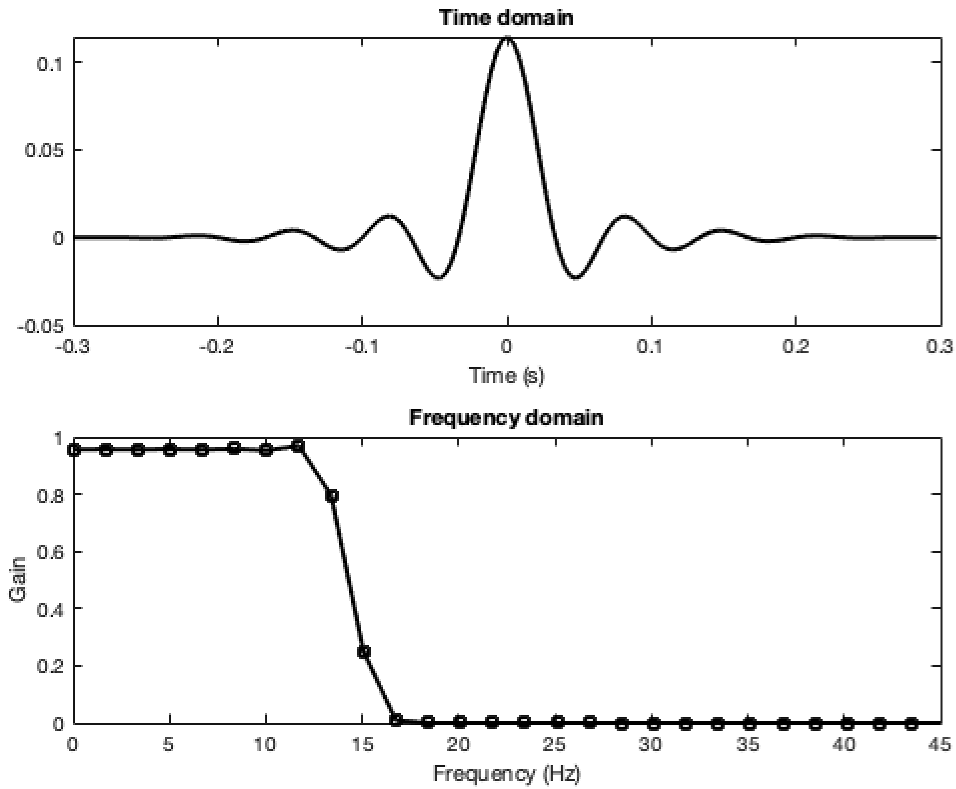
```
% axis labels, etc
xlabel('Time (ms)')
ylabel('Voltage (\muV)')
title([ 'ERP from channel ' chan2use ])
```



first low-pass filter (windowed sinc function)

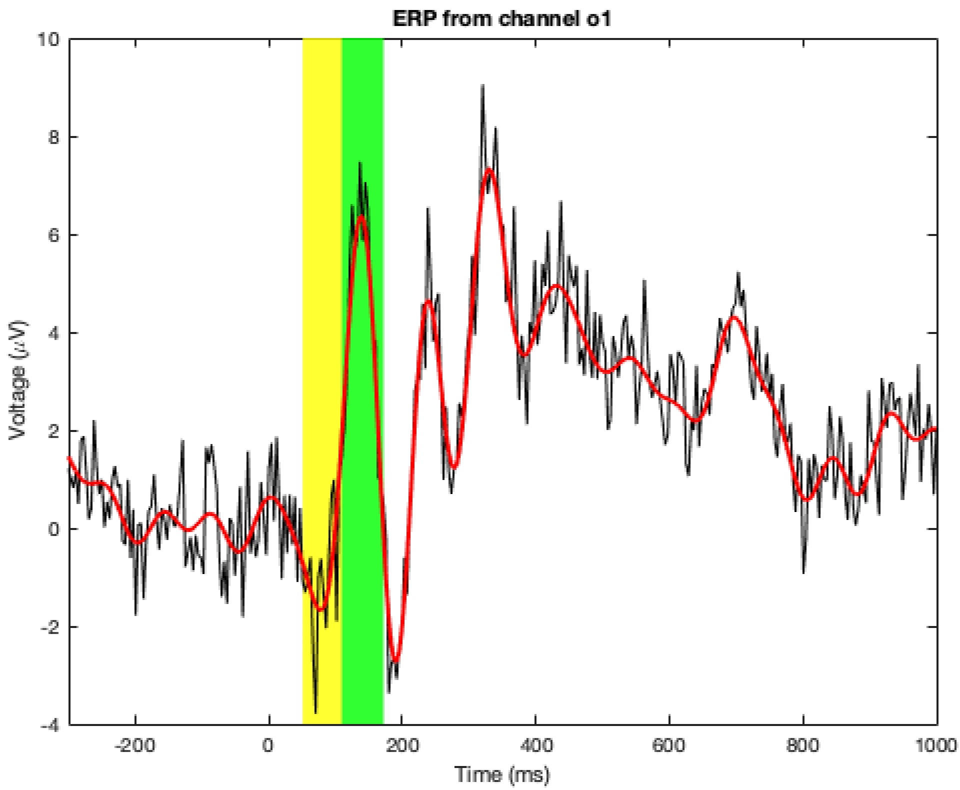
```
lowcut = 15;
filttime = -.3:1/EEG.srate:.3;
filtkern = sin(2*pi*lowcut*filttime) ./ filttime;

% adjust NaN and normalize filter to unit-gain
filtkern(~isfinite(filtkern)) = max(filtkern);
filtkern = filtkern./sum(filtkern);

% windowed sinc filter
filtkern = filtkern .* hann(length(filttime))';



% inspect the filter kernel
figure(2), clf
subplot(211)
plot(filttime,filtkern,'k','linew',2)
```

```
xlabel('Time (s)')
title('Time domain')


subplot(212)
hz = linspace(0,EEG.srate,length(filtkern));
plot(hz,abs(fft(filtkern)).^2,'ks-','linew',2)
set(gca,'xlim',[0 lowcut*3])
xlabel('Frequency (Hz)'), ylabel('Gain')
title('Frequency domain')
```



now filter the ERP and replot

```
% apply filter
erpFilt = filtfilt(filtkern,1,erp);

% plot on top of unfiltered ERP
figure(1), hold on
plot(EEG.times,erpFilt,'r','linew',2)
```



peak-to-peak voltages and timings

```
%%%% first for unfiltered ERP

% find minimum/maximum peak values and peak times
[erpMin,erpMinTime] = min(erp(negpeaktime(1):negpeaktime(2)));
[erpMax,erpMaxTime] = max(erp(pospeaktime(1):pospeaktime(2)));

% ERP timings
erpMinTime = EEG.times( erpMinTime+negpeaktime(1)-1 );
erpMaxTime = EEG.times( erpMaxTime+pospeaktime(1)-1 );

% get results (peak-to-peak voltage and latency)
erpP2P = erpMax - erpMin;
erpP2Plat = erpMaxTime - erpMinTime;


%%%% then for low-pass filtered ERP

% find minimum/maximum peak values and peak times
[erpFMin,erpFMinTime] = min(erpFilt(negpeaktime(1):negpeaktime(2)));
[erpFMax,erpFMaxTime] = max(erpFilt(pospeaktime(1):pospeaktime(2)));

% ERP timings
erpFMinTime = EEG.times( erpFMinTime+negpeaktime(1)-1 );
```

```
erpFMaxTime = EEG.times( erpFMaxTime+pospeaktime(1)-1 );

% get results (peak-to-peak voltage and latency)
erpFP2P = erpFMax - erpFMin;
erpFP2Plat = erpFMaxTime - erpFMinTime;
```

Report the results in the command window

```
% clear the screen
clc

fprintf('\nRESULTS FOR PEAK POINT:')
fprintf('\n   Peak-to-peak on unfiltered ERP: %5.4g muV, %4.3g ms
 span.',erpP2P,erpP2Plat)
fprintf('\n   Peak-to-peak on filtered ERP:   %5.4g muV, %4.3g ms
 span.\n\n',erpFP2P,erpFP2Plat)
```

```
RESULTS FOR PEAK POINT:
   Peak-to-peak on unfiltered ERP: 11.27 muV, 66.4 ms span.
   Peak-to-peak on filtered ERP:    8.05 muV, 58.6 ms span.
```

repeat for mean around the peak

```
% time window for averaging (one-sided!!)
win = 10; % in ms
% now convert to indices
win = round( win / (1000/EEG.srate) );

%%%% first for unfiltered ERP

% find minimum/maximum peak times
[~,erpMinTime] = min(erp(negpeaktime(1):negpeaktime(2)));
[~,erpMaxTime] = max(erp(pospeaktime(1):pospeaktime(2)));

% adjust ERP timings
erpMinTime = erpMinTime+negpeaktime(1)-1;
erpMaxTime = erpMaxTime+pospeaktime(1)-1;

% now find average values around the peak time
erpMin = mean( erp(erpMinTime-win:erpMinTime+win) );
erpMax = mean( erp(erpMaxTime-win:erpMaxTime+win) );


% ERP timings
erpMinTime = EEG.times( erpMinTime );
erpMaxTime = EEG.times( erpMaxTime );

% get results (peak-to-peak voltage and latency)
erpP2P = erpMax - erpMin;
erpP2Plat = erpMaxTime - erpMinTime;
```

```
%%%% then for low-pass filtered ERP

% find minimum/maximum peak values and peak times
[~,erpFMinTime] = min(erpFilt(negpeaktime(1):negpeaktime(2)));
[~,erpFMaxTime] = max(erpFilt(pospeaktime(1):pospeaktime(2)));

% adjust ERP timings
erpFMinTime = erpFMinTime+negpeaktime(1)-1;
erpFMaxTime = erpFMaxTime+pospeaktime(1)-1;

% now find average values around the peak time
erpFMin = mean( erpFilt(erpFMinTime-win:erpFMinTime+win) );
erpFMax = mean( erpFilt(erpFMaxTime-win:erpFMaxTime+win) );

% adjust ERP timings
erpFMinTime = EEG.times( erpFMinTime );
erpFMaxTime = EEG.times( erpFMaxTime );

% get results (peak-to-peak voltage and latency)
erpFP2P = erpFMax - erpFMin;
erpFP2Plat = erpFMaxTime - erpFMinTime;
```

Report the results in the command window

```
fprintf('\nRESULTS FOR WINDOW AROUND PEAK:')
fprintf('\n   Peak-to-peak on unfiltered ERP: %5.4g muV, %4.3g ms
 span.',erpP2P,erpP2Plat)
fprintf('\n   Peak-to-peak on filtered ERP:   %5.4g muV, %4.3g ms
 span.\n\n',erpFP2P,erpFP2Plat)
```

```
RESULTS FOR WINDOW AROUND PEAK:
   Peak-to-peak on unfiltered ERP: 7.996 muV, 66.4 ms span.
   Peak-to-peak on filtered ERP:   7.442 muV, 58.6 ms span.
```

done.

Published with MATLAB® R2020b