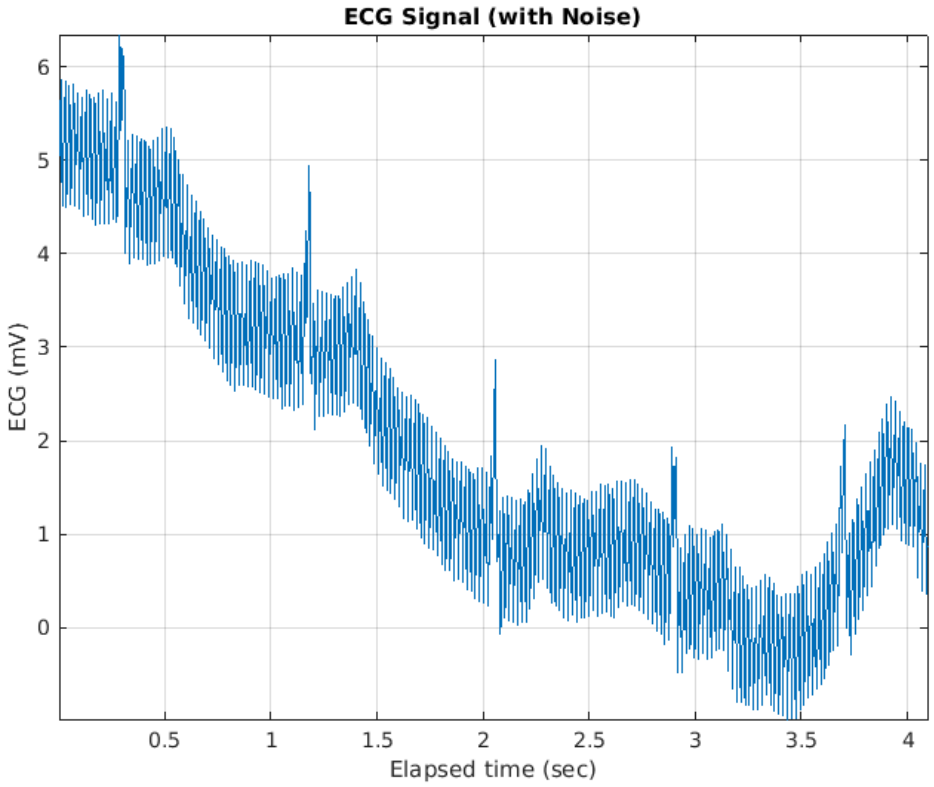
```
% BME671L Lab #10: filtering

% Your name: Ravitashaw Bathla (rb369)

clear;

% In this lab we will look at an electrocardiogram and the effects of
% filtering.
```

Q1: electrocardiogram.txt has column headers, and two columns separated by a tab. Open the file using a text editor and take a look. The first column contains a time vector and the second contains the electrical signal in mV. Use 'dlmread' to import the raw data using the appropriate options. Store the time series data as 'tt' and the electrical signal as 'ecg'.

```
ecg_file = dlmread('electrocardiogram.txt', '\t', 3, 0);
tt = ecg_file(:, 1);
ecg = ecg_file(:, 2);
```

Q2: Plot the signal. This graph should be 'publishing quality' with all axis labeled, units, ect... What types of noise are in this signal? Describe how the noise contributes to the general shape of the signal.

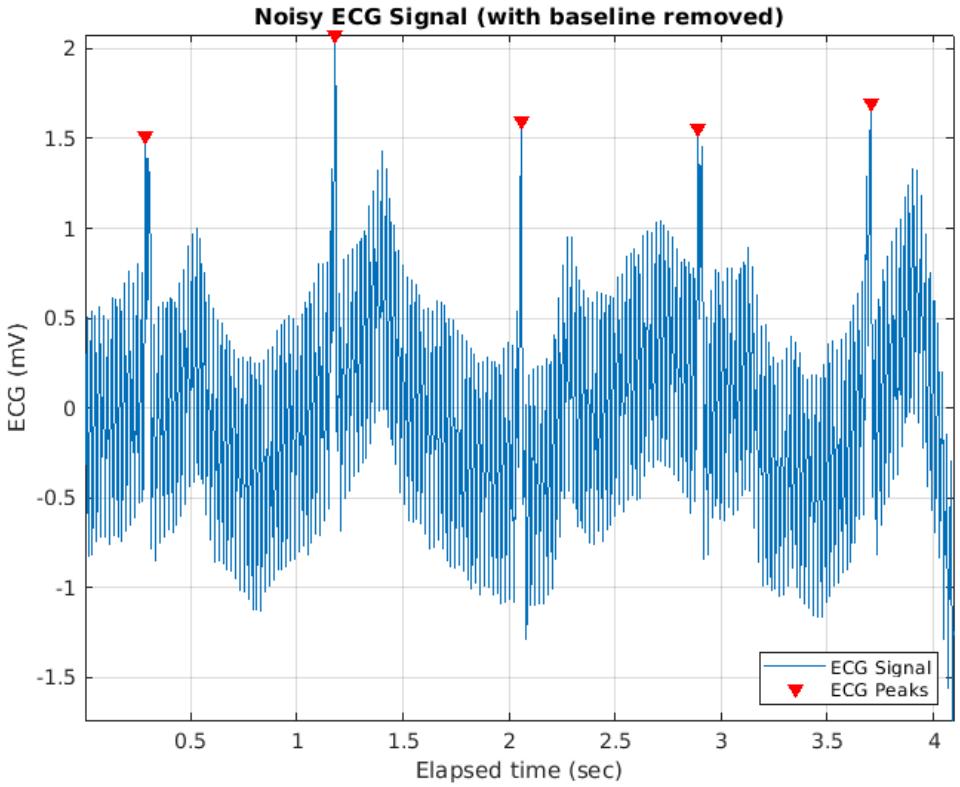
```
% YOUR ANSWER:
figure(1), clf;
plot(tt, ecg);
title('ECG Signal (with Noise)');
ylabel('ECG (mV)');
xlabel('Elapsed time (sec)');
grid on;
axis tight;

% There can be several components that might add noise to the ECG
 signal.
% It can be from interference from other electronic devices,
 instrument noise,
% motion artifcats, electrosurgical noise etc.
% A clean continious ECG signal will have very slight fluctuation in
 the
% signal. However in this case, there appears to be a very large
% fluctuation in the signal. This reflecss that there is significant
 noise
% in the provided signal.
```



Q3: In order to properly analyze our signal we need to know what frequencies we are interested in. First predict the expected frequency based on your knowledge of the human heart rate. Then determine the fundamental frequency and period of this signal in s and Hz. Estimate the peak height (don't forget to subtract the baseline).

```
% HINT: I like the command 'findpeaks', or you can also do this
% by hand.

% YOUR ANSWER
% Predicted frequency: 1.33(Hz) at 80bpm
% Frequency: 1.1363Hz
% Period: 0.88 seconds (difference between two peaks)
% Peak height: 1.6 (average of all the peaks) and 1.5 for the first
 peak

% removed the baseline by fitting a low order polynomial to detrend
 the
% data
[p,s,mu] = polyfit(tt,ecg, 6);
baseline = polyval(p,tt,[],mu);
ecg_no_baseline = ecg - baseline;

% find peaks of the signal with a threshold
[peaks,locs_peaks] = findpeaks(ecg_no_baseline,
 tt, 'MinPeakHeight',1.5);
```

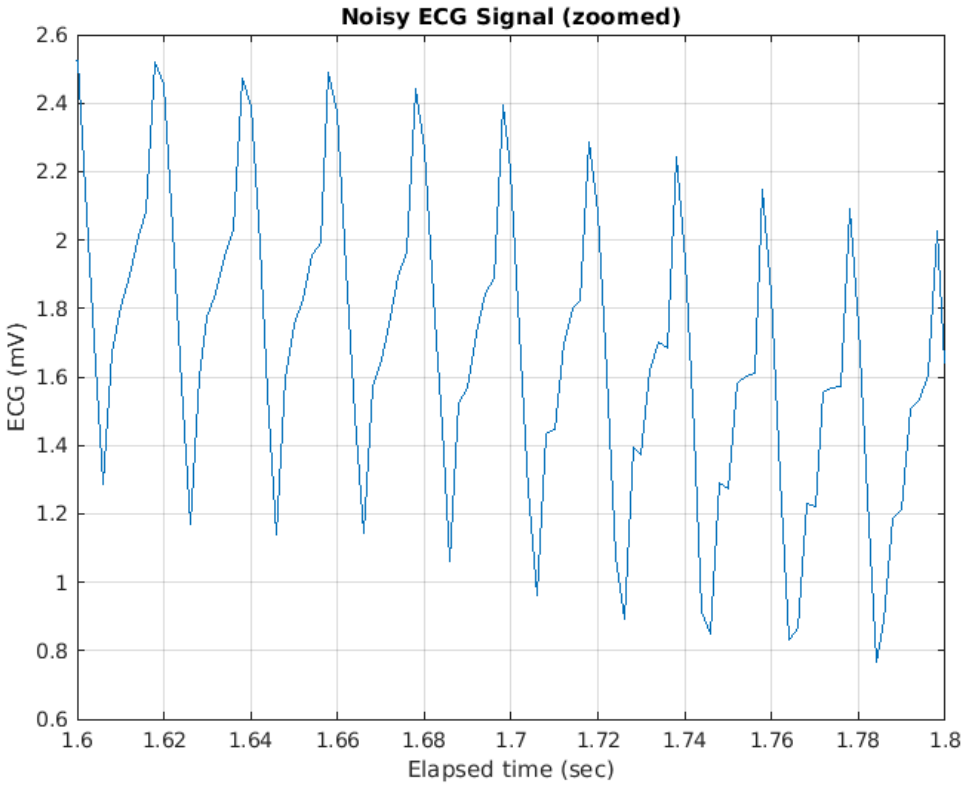
```
% plot the result
figure(2), clf;
plot(tt, ecg_no_baseline);
hold on;
plot(locs_peaks, peaks, 'rv','MarkerFaceColor','r');
title({'Noisy ECG Signal (with baseline removed)'});
ylabel('ECG (mV)');
xlabel('Elapsed time (sec)');
grid on;
axis tight;
legend({'ECG Signal','ECG Peaks'}, 'Location','southeast');
```



Q4: Next we will characterize the high frequency noise in the time domain. Make a plot of a small portion of the signal that allows you to visualize the noise. Using a small portion of the signal approximate the frequency and period of the noise.

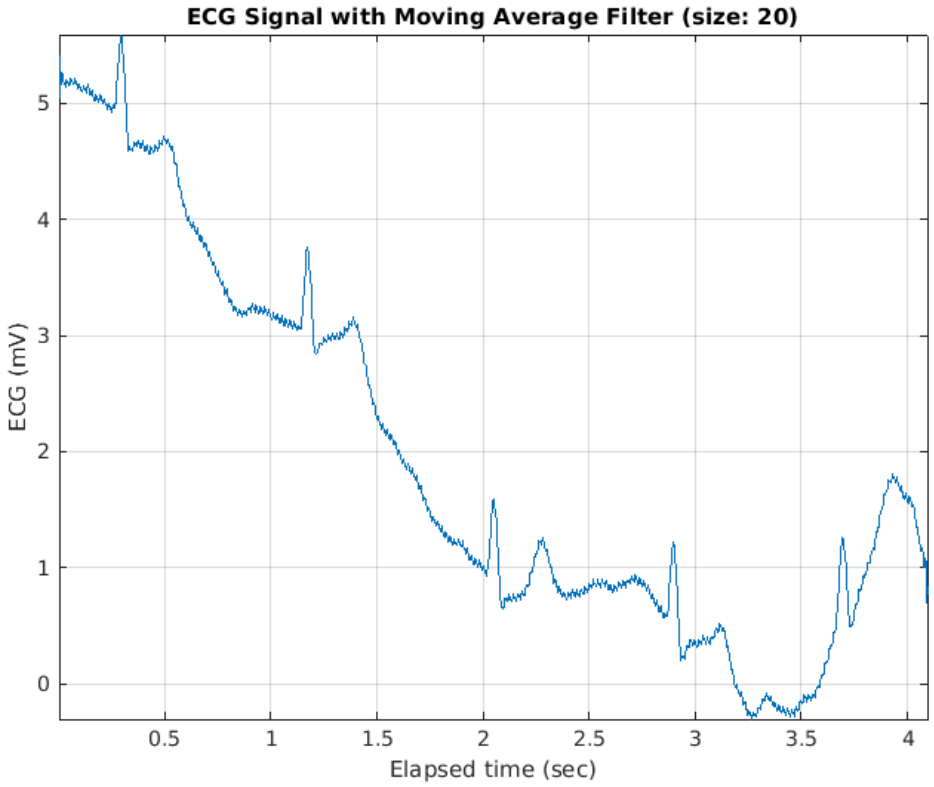
```
% YOUR ANSWER
% Frequency:
% Period:
figure(3), clf;
plot(tt,ecg);
xlim([1.6, 1.8]);
title({'Noisy ECG Signal (zoomed)'});
ylabel('ECG (mV)');
xlabel('Elapsed time (sec)');
grid on;
```

```
% The time period of the noise seems to be 0.02 seconds. The frequency
 of
% the noise is therefore, 1/0.02 = 50Hz.
% ******************************************************************
% SHOW YOUR IMAGE FOR Q4 TO THE TA TO RECEIVE CREDIT FOR THE LAB IF
 YOU
% ARE NOT PRESENT AT THE DISCUSSION SESSION ON FRIDAY.
% ******************************************************************
```



Q5: One low pass filter we discussed during the semester is a moving average filter. Using "smooth", implement a moving average filter that covers 20 data points. Plot the result.

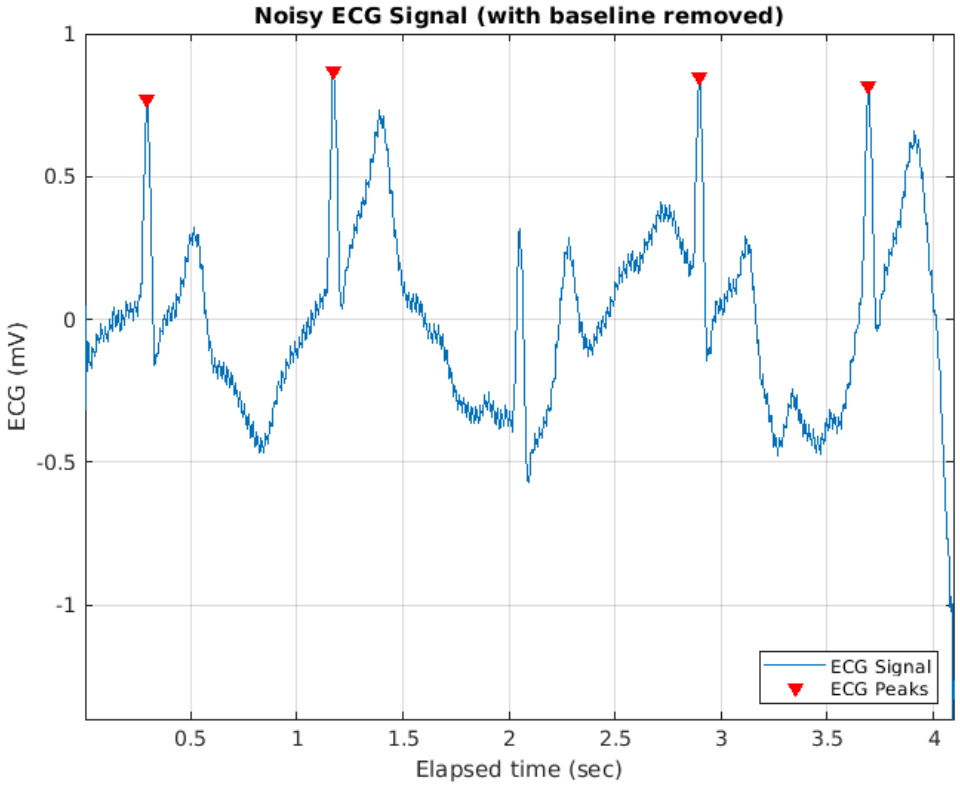
```
ecg_smooth = smooth(ecg, 20);
figure(4), clf;
plot(tt, ecg_smooth);
title({'ECG Signal with Moving Average Filter (size: 20)'});
ylabel('ECG (mV)');
xlabel('Elapsed time (sec)');
grid on;
axis tight;
```



Q6: Estimate the filtered peak height. Explain any possible down sides to filtering using a moving average filter.

```
% YOUR ANSWER:
% The average fitlered peak height is approximately 0.8 mV.
% If the signal has lot of large high frequency noise, it will require
 a
% large window size. The Moving average filter have a temporal
 autocorrelation
% and large window size induces latency making it difficult to filter
 out
% all the noise. In addition, the amplitude of the peak has decreased
 in
% comparision to original signal without filtering. Therefore, the
% intensity of the original signal seems to be reduced as well.

[p,s,mu] = polyfit(tt,ecg_smooth, 6);
smooth_baseline = polyval(p,tt,[],mu);
ecg_smooth_no_baseline = ecg_smooth - smooth_baseline;

% find peaks of the signal with a threshold
[peaks_smooth,smooth_locs_peaks] = findpeaks(ecg_smooth_no_baseline,
 tt, 'MinPeakHeight',0.75);

% plot the result
figure(5), clf;
```

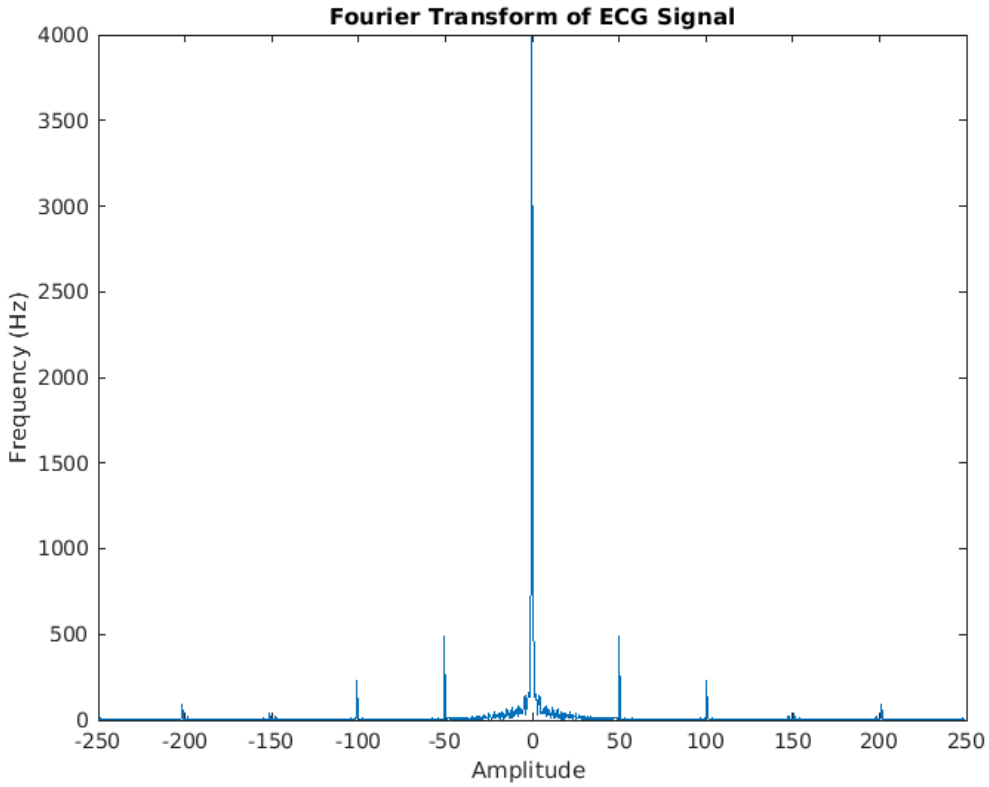
```
plot(tt, ecg_smooth_no_baseline);
hold on;
plot(smooth_locs_peaks, peaks_smooth, 'rv','MarkerFaceColor','r');
title({'Noisy ECG Signal (with baseline removed)'});
ylabel('ECG (mV)');
xlabel('Elapsed time (sec)');
grid on;
axis tight;
ylim([min(ecg_smooth_no_baseline), 1]);
legend({'ECG Signal','ECG Peaks'}, 'Location','southeast');
```



Q7: Throughout the semester we have described signals in both the time and frequency domain. Take the Fourier transform of the signal and store it as f_ecg. Plot the magnitude of the Fourier transform of the ecg signal with the x-axis in Hz. Using this representation, estimate the frequency of the noise present in the signal.

```
% YOUR ANSWER
% Frequency:
f_ecg = fftshift(fft(ecg));

ts = mean(diff(tt));
fs = 1/ts;

L = length(ecg);

ff = ((-L/2):(L/2)-ts)*fs/L;
```

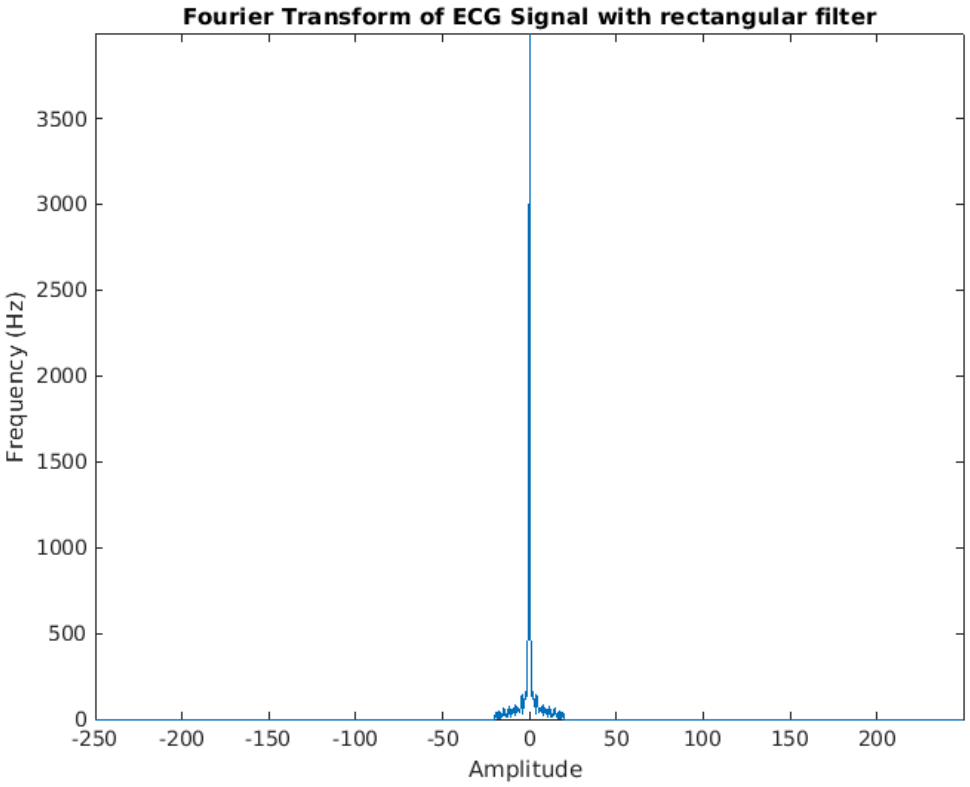
```
figure(6), clf;
plot(ff, abs(f_ecg));
title('Fourier Transform of ECG Signal');
xlabel('Amplitude ');
ylabel('Frequency (Hz)');

% From the plot, it appears that the noise is present the fundamental
% frequency of +/- 50.3Hz. This is likely the noise in the signal
 because
% as anticipated in q4, the noise has frequency of 50Hz. In q1, we
% anticipate that the predicted frequency is 1.33Hz for clean ECG
 signal.
```

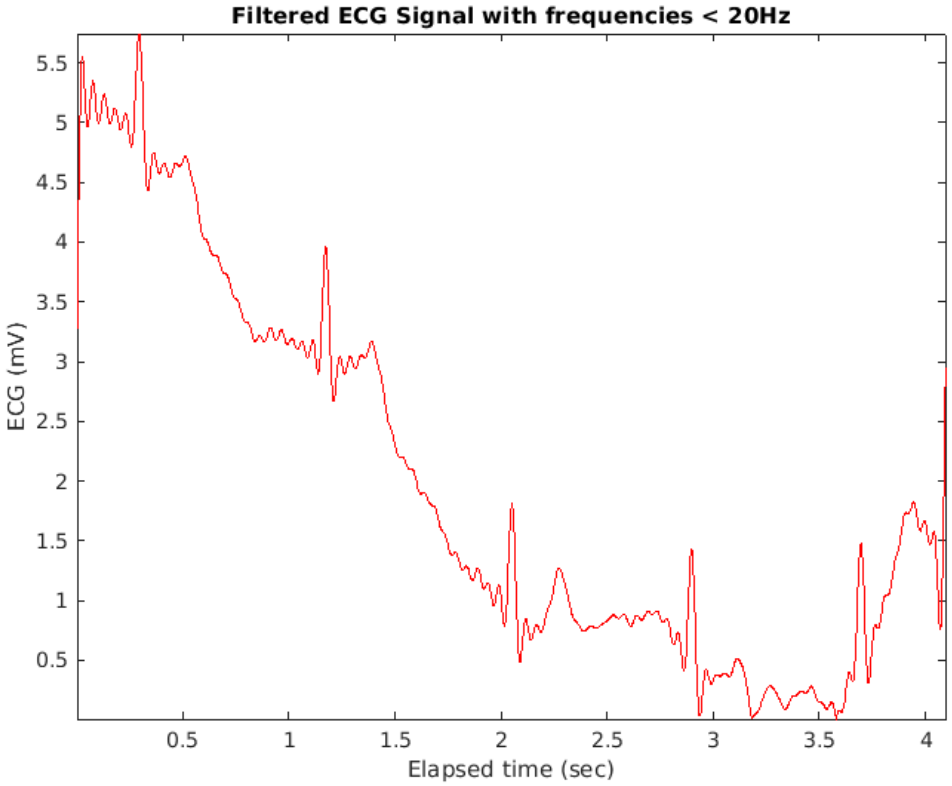


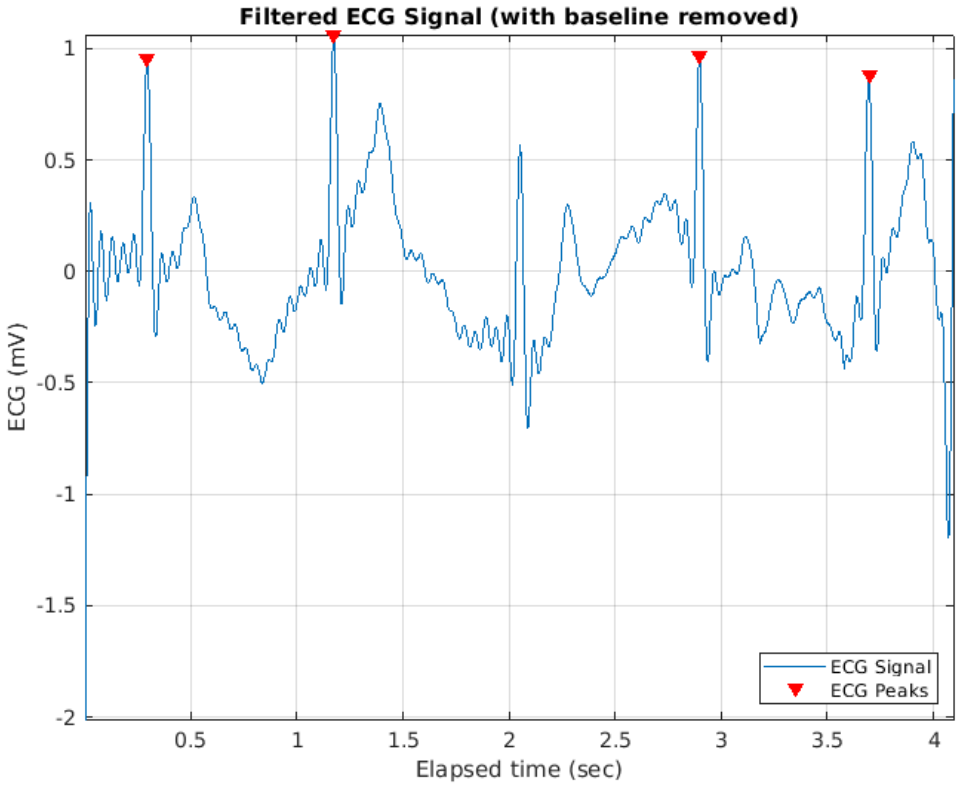
Q8: The simplest way to get rid of the noise is to use a rectangular (rect) low pass filter and set all values of the Fourier transform to 0 above the cut-off frequency. Design a rect filter, f20, in the frequency domain that has a value of 1 when abs(f) < 20 Hz, and 0 otherwise. Implement the filter using multiplication in the frequency domain. Take the inverse Fourier transform of the filtered signal and plot the result in the time domain. Again, estimate the peak height.

```
% YOUR ANSWER:
% Peak height: the mean peak height is 0.9216 and the height of the
 first
% peak is at 0.9491

f20 = abs(ff) < 20;
rect_ecg = f_ecg .* f20';
```

```
figure(7), clf;
plot(ff, abs(rect_ecg));
title('Fourier Transform of ECG Signal with rectangular filter');
xlabel('Amplitude ');
ylabel('Frequency (Hz)');
axis tight;

ecg_filt = ifft(rect_ecg);
figure(8), clf;
plot(tt, abs(ecg_filt), 'r-');
title('Filtered ECG Signal with frequencies < 20Hz');
ylabel('ECG (mV)');
xlabel('Elapsed time (sec)');
axis tight;

% find peak height
[p,s,mu] = polyfit(tt, abs(ecg_filt), 6);
rect_baseline = polyval(p,tt,[],mu);
ecg_rect_no_baseline =  abs(ecg_filt) - rect_baseline;

% find peaks of the signal with a threshold
[peaks_rect,rect_locs_peaks] = findpeaks(ecg_rect_no_baseline,
 tt, 'MinPeakHeight',0.8);
figure(9), clf;
plot(tt, ecg_rect_no_baseline);
hold on;
plot(rect_locs_peaks, peaks_rect, 'rv','MarkerFaceColor','r');
title({'Filtered ECG Signal (with baseline removed)'});
ylabel('ECG (mV)');
xlabel('Elapsed time (sec)');
grid on;
axis tight;
legend({'ECG Signal','ECG Peaks'}, 'Location','southeast');
```

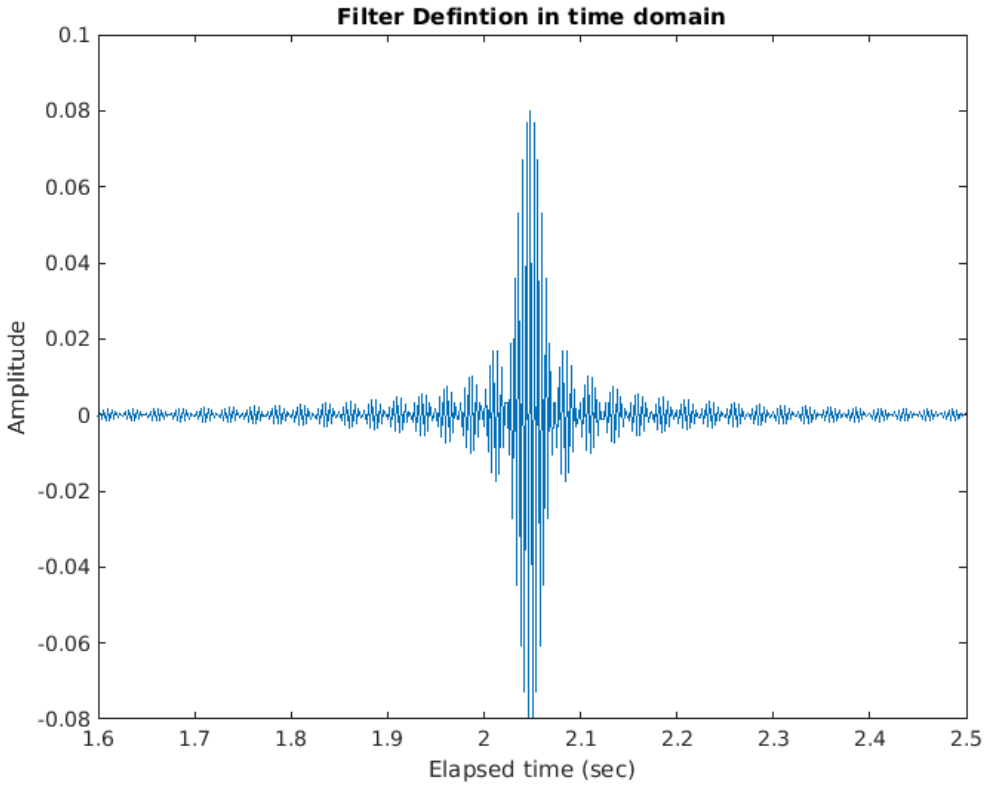

Fourier Transform of ECG Signal with rectangular filter
Frequency (Hz)
3500
3000
2500
2000
1500
1000
500
0
-250
-200
-150
-100
-50
0
50
100
150
200
Amplitude


Filtered ECG Signal with frequencies < 20Hz
ECG (mV)
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0.5
1
1.5
2
2.5
3
3.5
4
Elapsed time (sec)



Q9: To examine the effect of the filter in the time domain, take the inverse Fourier transform of the filter and plot the results in the time domain. Use 'xlim' to examine the signal from 1.6 to 2.5s. What is multiplication in the frequency domain equivalent to in the time domain?

```
% YOUR ANSWER:
f20_i = fftshift(ifft(f20));
figure(10), clf;
plot(tt, f20_i);
xlim([1.6 2.5]);
title('Filter Defintion in time domain');
ylabel('Amplitude');
xlabel('Elapsed time (sec)');

% Multiplication in a frequency domain is equivallent to convolution
 in the
% time domain.
```



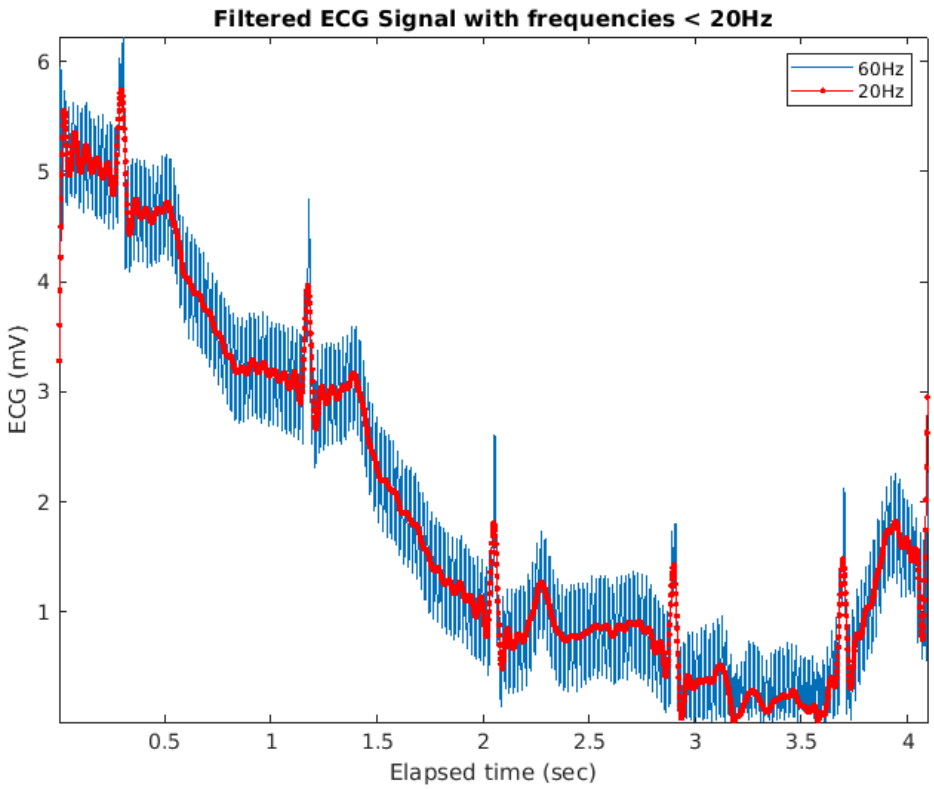
Q10: The Fourier transform of a rect function is a sinc function. By definition, the sinc function extends from -inf in inf, although it does decrease in magnitude. Therefore, applying a rect filter in the frequency domain can introduce ringing in the time domain. This ringing is most prominent in places where there is a fast, large change in the signal such as the start of the signal (going from 0 to a value) or near the peaks. This is because those changes are comprised of extremely high frequencies. Is it possible to implement a true frequency rect filter in the time domain?

```
% YOUR ANSWER: Sinc function is an "ideal" low pass filter and a real
% system can only approximate the results. Therefore, it is
 theoritically
% possible to have a true frequency rect filter in time domain but the
 result can be
% approximated to a close in real-time.
%
```

Q11: One method to reduce the ringing is to use a wider cut-off filter to include additional higher frequencies. In the same figure, plot the time signal corresponding to the 20 Hz rect filter and a 60 Hz rect filter. Describe the time-domain differences between these two filters. How would the time-domain representation of the filter change if an even wider frequency cut-off was used?

```
% YOUR ANSWER:
% As mentioned before, the rectangular Sinc low pass filter introduces
% ringing in the signal in the time-domain. This can be clearly
 observed
% with frequency of 60Hz (in blue). There are osciallations in the
 singal
```

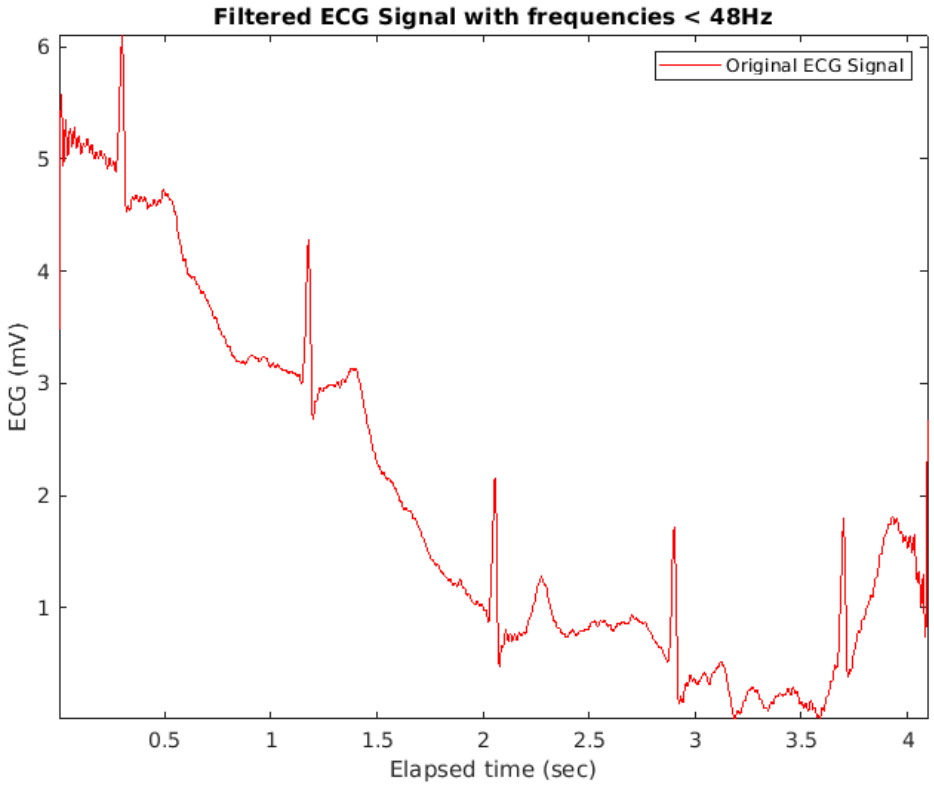
```
% with higher frequency. If wider range of frequency cut-off is used,
 it
% may result in higher ringing pattern and reduce the amplitude of the
% original ECG signal.

f60 = abs(ff) < 60;
rect_ecg60 = f_ecg .* f60';

ecg_filt60 = ifft(rect_ecg60);
figure(11), clf;
plot(tt, abs(ecg_filt60));
hold on;
plot(tt, abs(ecg_filt), 'r.-');
title('Filtered ECG Signal with frequencies < 20Hz');
ylabel('ECG (mV)');
xlabel('Elapsed time (sec)');
legend({'60Hz','20Hz'}, 'Location','northeast');
axis tight;
```

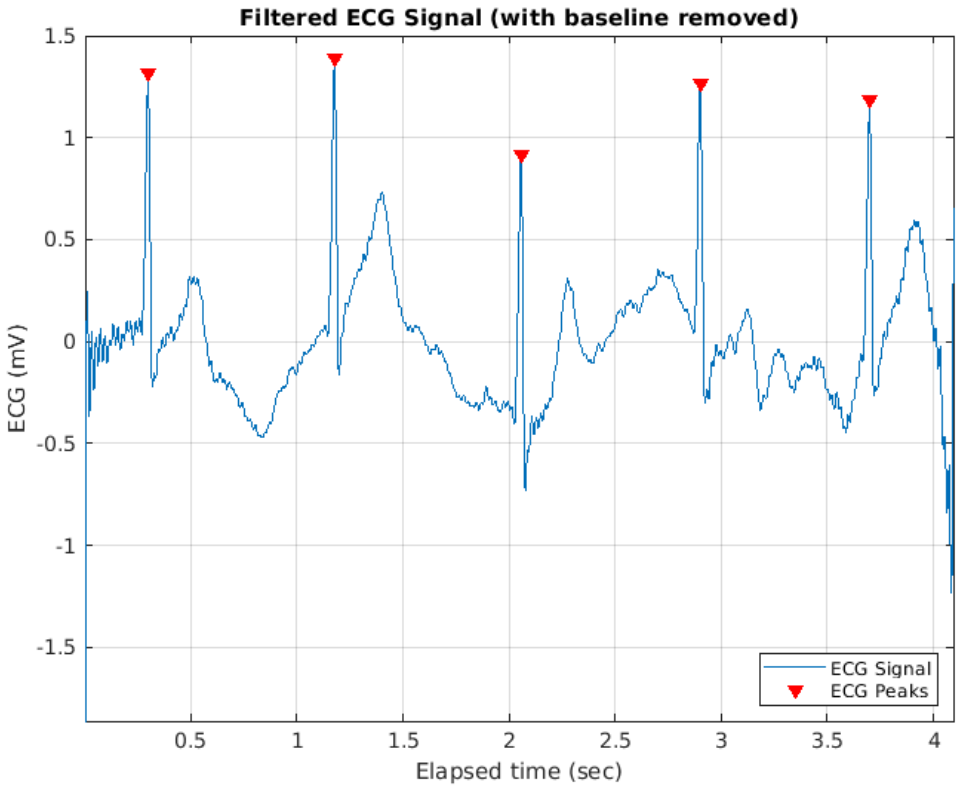


Q12: Play around with different cut off for the rect filter and choose one that you think does a decent job of reducing the noise while maintaining the original peak magnitudes. Make a plot of the filtered signal and report the cut-off frequency and peak height.

```
% YOUR ANSWER:
% Cut-off frequency: 48Hz
% Peak height: the mean peak height is 1.213 and the height of the
 first
```

```
% peak is 1.3163

f_opt= abs(ff) < 48;
rect_ecg_opt = f_ecg .* f_opt';

ecg_filt_opt = ifft(rect_ecg_opt);
figure(12), clf;
%plot(tt, abs(ecg));
%hold on;
plot(tt, abs(ecg_filt_opt), 'r-');
title('Filtered ECG Signal with frequencies < 48Hz');
ylabel('ECG (mV)');
xlabel('Elapsed time (sec)');
legend({'Original ECG Signal','Filtered ECG Signal
 (48Hz)'}, 'Location','northeast');
axis tight;

[p,s,mu] = polyfit(tt,abs(ecg_filt_opt), 6);
filtered_baseline = polyval(p,tt,[],mu);
ecg_filtered_no_baseline = abs(ecg_filt_opt) - filtered_baseline;

% find peaks of the signal with a threshold
[peaks_filtered, filtered_locs_peaks] =
 findpeaks(ecg_filtered_no_baseline, tt, 'MinPeakHeight',0.8);

% plot the result
figure(13), clf;
plot(tt, ecg_filtered_no_baseline);
hold on;
plot(filtered_locs_peaks, peaks_filtered, 'rv','MarkerFaceColor','r');
title({'Filtered ECG Signal (with baseline removed)'});
ylabel('ECG (mV)');
xlabel('Elapsed time (sec)');
grid on;
axis tight;
ylim([min(ecg_filtered_no_baseline), 1.5]);
legend({'ECG Signal','ECG Peaks'}, 'Location','southeast');

Warning: Ignoring extra legend entries.
```


Filtered ECG Signal with frequencies < 48Hz
Original ECG Signal
ECG (mV)
Elapsed time (sec)
6
5
4
3
2
1
0.5
1
1.5
2
2.5
3
3.5
4


Filtered ECG Signal (with baseline removed)
ECG Signal
ECG Peaks
ECG (mV)
Elapsed time (sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0.5
1
1.5
2
2.5
3
3.5
4

When you are done:

```
% * upload your script to Sakai
%   * upload a pdf containing your script and outputs
```

Published with MATLAB® R2020b