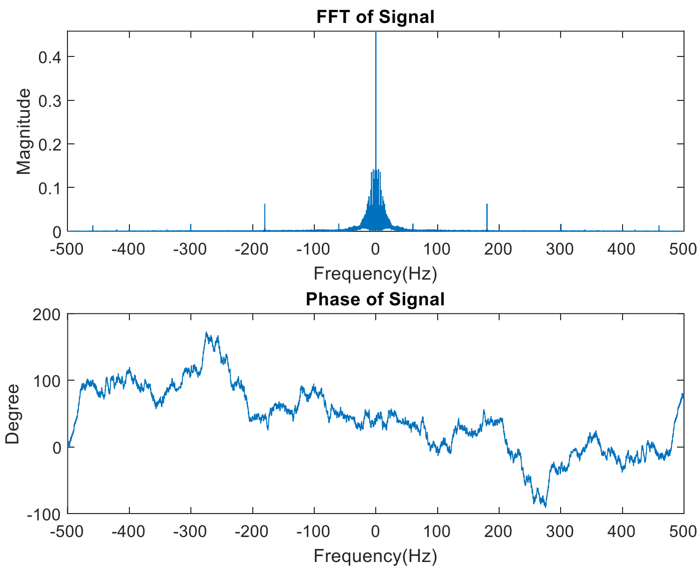
EE434 Biomedical Signal Processing MT Take Home Exam

Utku Acar 250206062

Codes of corresponding part can be found on APPENDIX at page 7-9 with corresponding part names

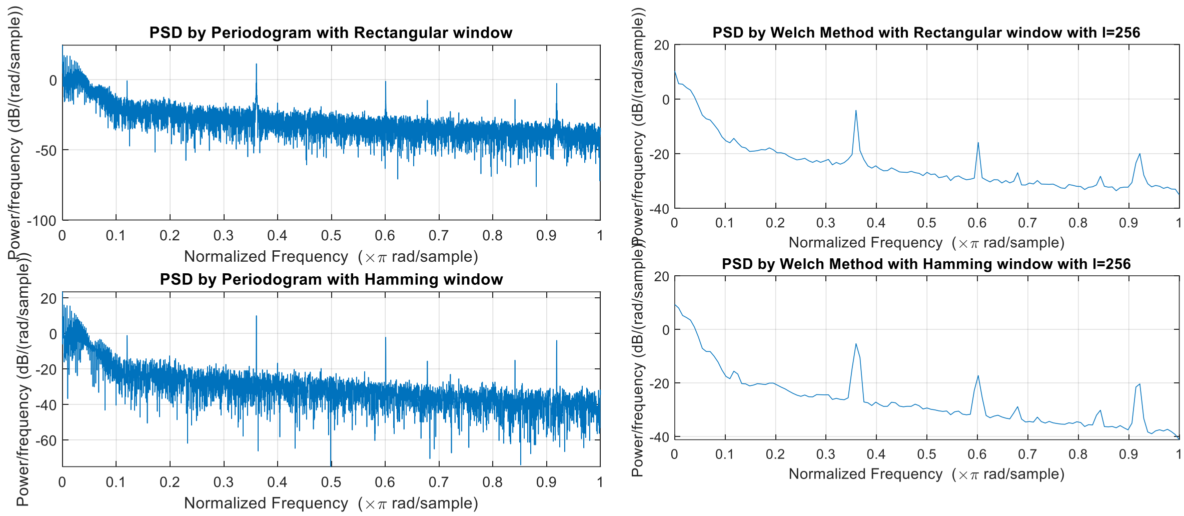
Eg: `%% a`

a)

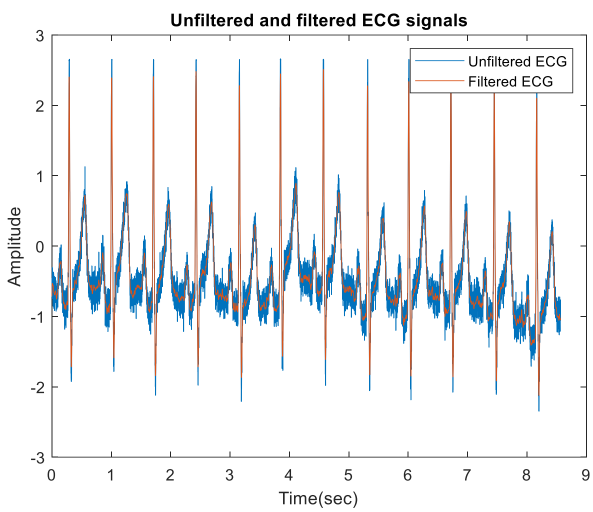


b) This is an ECG signal which means it shows heart behavior so our signal should begin from 0.05 Hz and goes to the 150Hz component. But we have 180Hz, 300 Hz, and 460 Hz components in our signal too. So, they could be noise. So, we have high frequency noises. This can be seen in figure above.

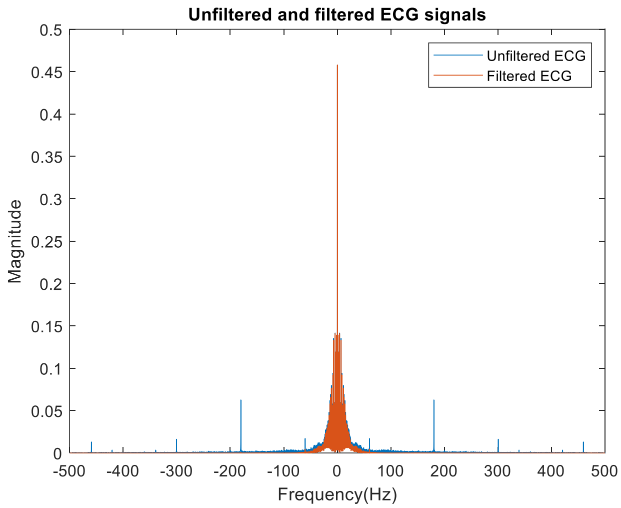
c) We can see that Welch method has less oscillatory behavior in PSD. We cannot change window length in periodogram (It should be equal to window length). I did not also divide the signal into parts because if I did it and take average of them it becomes Barlett method. Since window length is equal to signal length, we have terrible time resolution nearly any, but we have excellent frequency resolution. In Welch method averages overlapped windows and the size of the window can be adjustable in Welch method. When we increase size of the window the lines get more peaks(oscillations). Hamming windows gives smoother PSD graph than rectangular window. This can be seen in figure below.



d)



Moving Average filter averages signal by moving kernel and dividing to length of the kernel. 'Moving' feature is provided by filter command which is convolution operation. When we increase size of the kernel, we lose peak values and while we decrease the size of the kernel, we get similar output as our signal which is unnecessary computation since the main purpose of this technique is noise removing. 10 is ideal for kernel size since it is removing enough noise without losing peak values. As we can see in graph above the signal has denoised as possible. This can be seen in figure above.



We can see that when we set kernel size 10, high frequency components decayed nearly 10 times while low frequency components (0-150) Hz are not affected much (2% at max). While we are setting kernel size as 80, we lose low frequency components too. This can be seen in figure above.

e) The selection of windows and their parameters are the best because I have tried different options for them and if we suppose moving average with kernel size 10 is the best denoised signal as possible (10 times decayed at frequency 180 Hz) then we can calculate mean square error for each option of window parameters separately. This assumption based on the idea that MA works good for FIR filters since they have only zeros (Deep valleys). (They have poles at 0 or infinity)

Some example options of the options have been given below:

order(N)=5 window1= rectwin(N+1) window2= Kaiser(N+1,0.5) MSE=0.0428

order(N)=5 window1= rectwin(N+1) window2= Kaiser(N+1,1) MSE=0.0437

order(N)=5 window1= rectwin(N+1) window2= Kaiser(N+1,2.5) MSE=0.0476

order(N)=5 window1= hamming(N+1) window2= Kaiser(N+1,0.5) MSE=0.0521

order(N)=5 window1= hann(N+1) window2= Kaiser(N+1,0.5) MSE=0.0536

order(N)=10 window1= rectwin(N+1) window2= Kaiser(N+1,0.5) MSE=0.2803

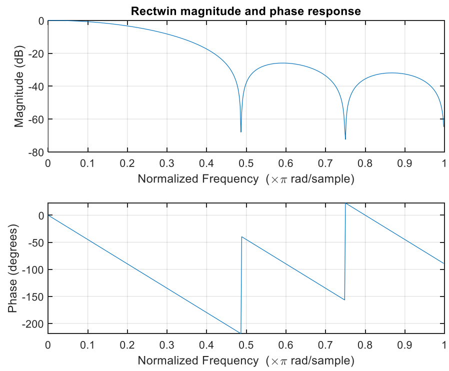
order(N)=4 window1= rectwin(N+1) window2= Kaiser(N+1,0.5) MSE=0.0477

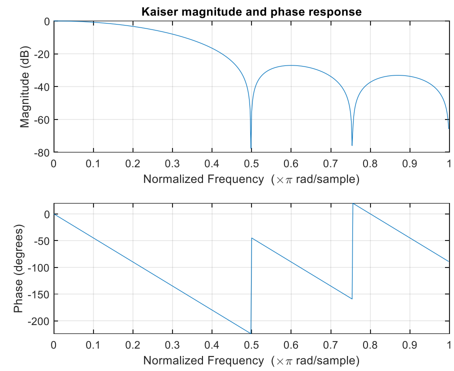
order(N)=5 window1= rectwin(N+1) window2= hamming(N+1) MSE=0.0515

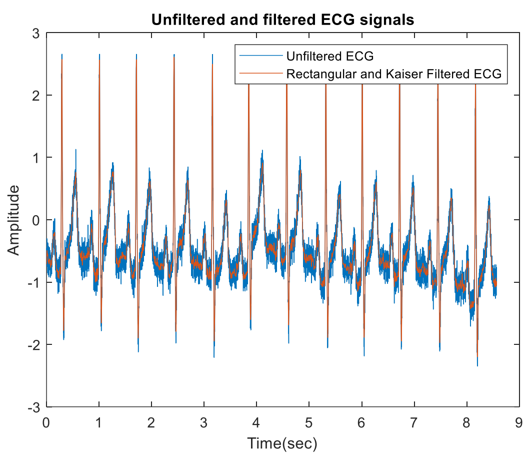
order(N)=5 window1= rectwin(N+1) window2= Hann(N+1) MSE=0.0530

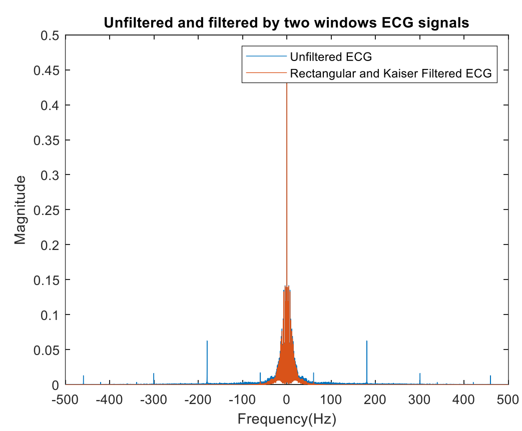
We can clearly see that best result is first one which is N=5, rectwin(N+1),Kaiser(N+1,0.5) MSE=0.0428.

The answer of why I choose rectangular window and Kaiser window is because rectangular window gives near perfect primary elimination of high frequencies (>150) then Kaiser gives me opportunity that discriminate main lobe from side lobes without changing filter size(N+1) and order with roll off factor beta (=0.5). As a result, I filtered high frequencies with rectangular window and improve filtering by Kaiser window without losing peak values much (%2) in time domain while reducing noise components 10 times. I also used filtfilt(.) to remove filter delay. Eventually signal got better MSE=0.0344 for best result above with rectangle and Kaiser window. Yes, it is denoised. Result of these process can be seen below.









As we can see from above the ECG signal has been denoised. It has less oscillatory behavior in time domain. It also could be excessive denoised because we have lost half of the amplitude of 60Hz component which is in interval (0,150) Hz but it could be also powerline noise so it’s unknown. This situation happened after I changed filter(.) function to filtfilt(.) function which makes zero phase correction which could be caused by filtering process. (Can be seen in frequency domain.). This can be seen in figure above. I choose eliminating 60 Hz component instead of saving it because I think filtering delay correction is more important problem than saving unknown signal which it could be noise also.

f) I selected Cheby2 filter over Cheby1 filter because we are caring about

low frequency components (<150Hz) even we lose sharpness of the response. Because

Type 1 Chebyshev filter allows low pass attenuation

Type 2 Chebyshev filter allows high pass attenuation

Since we have low frequency components in our ECG data, we should be using Type 2

Chebyshev to keep these low frequency values as they were.

order(N)=5 filter1= butter(.) window2= cheby2(30db ripple attenuation) MSE=0.0764

order(N)=5 filter1= butter(.) window2= cheby2(40db ripple attenuation) MSE=0.0711

order(N)=5 filter1= butter(.) window2= cheby2(50db ripple attenuation) MSE=0.0652

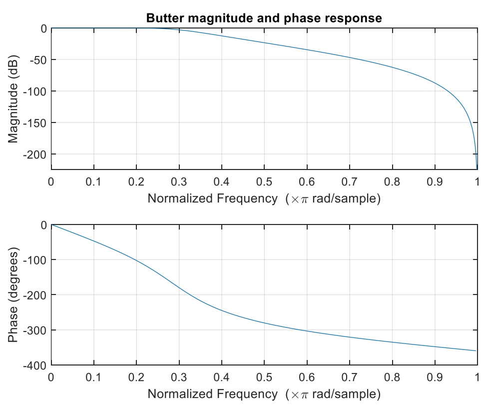
order(N)=5 filter1= butter(.) window2= cheby1(30db ripple attenuation) MSE=0.5521

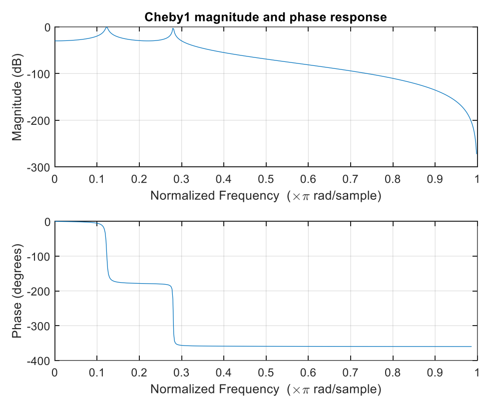
order(N)=4 filter1= butter(.) window2= cheby2(30db ripple attenuation) MSE=0.0547

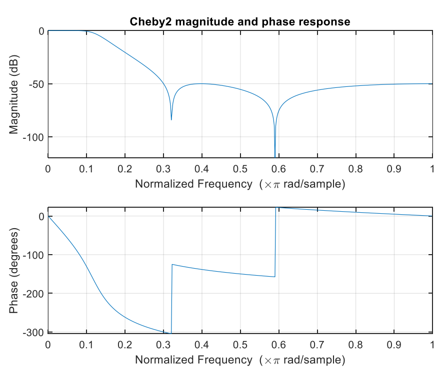
order(N)=3 filter1= butter(.) window2= cheby2(30db ripple attenuation) MSE=0.0345

order(N)=2 filter1= butter(.) window2= cheby2(30db ripple attenuation) MSE=0.2379

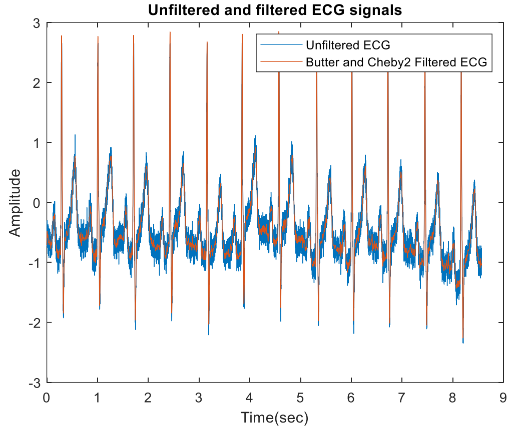
As we can see above best MSE is 0.0345 when filter order is 3 and cheby2 with 30 dB ripple attenuation. However, at this configuration we have lost some parts of our signal data (<150Hz) for example at 60 Hz we lost nearly all of it. So, we get better results (just 2% overshoot at max) while we are having filter order is 4 and ripple attenuation is 30 dB and MSE is 0.0547.

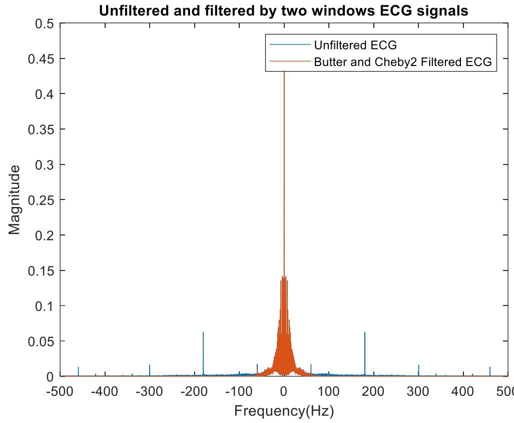






When we compare IIR and FIR filters most prominent differences in Phases of the filters. While FIR filters have linear phases, IIR filters has nonlinear phases. Butter filters has wider main lobe than FIR filters. Stability is also an important issue for IIR filter design while FIR filters are generally stable. This can be seen in figure above.





We can clearly identify that in time domain signal has been denoised without too much overshoot which means that filtered ECG has lesser oscillatory behavior which has been caused by noises earlier. This can be seen in figure above.

In frequency domain we see that low frequencies have been preserved and higher frequencies has been eliminated. Butter is for primary elimination and chebyshev2 for detailed elimination(discrimination) in my filter design. This can be seen in figure above.

APPENDIX (CODE PART)

```
close all
clear
clc
warning off
ecg_2=load("ecg_2.mat");
ecg_2=ecg_2.ecg_hfn';
fs=1000;
record_time=length(ecg_2)/fs;%8.5680 second
t=0:1/fs:record_time-1/fs;

%% a
f=linspace(-fs/2,fs/2,length(ecg_2));
fft_signal=abs(fftshift(fft(ecg_2))/length(ecg_2));
phase_signal=unwrap(angle(fft(ecg_2)/length(fft_signal)));
figure;
subplot(211)
plot(f,fft_signal);
title('FFT of Signal');
xlabel('Frequency(Hz)');
ylabel('Magnitude');
subplot(212)
plot(f,phase_signal);
title('Phase of Signal');
xlabel('Frequency(Hz)');
ylabel('Degree');

%% c
figure;
subplot(211)
periodogram(ecg_2);
title('PSD by Periodogram with Rectangular window');
subplot(212)
periodogram(ecg_2,hamming(length(ecg_2)));
title('PSD by Periodogram with Hamming window');

figure;
subplot(211)
pwelch(ecg_2,rectwin(256));
title('PSD by Welch Method with Rectangular window with l=256');
subplot(212)
pwelch(ecg_2,hamming(256));
title('PSD by Welch Method with Hamming window with l=256');

%% d
kernel_length=10;
Movavg_kernel=ones(kernel_length,1)/kernel_length;
mov_filter_out=filtfilt(Movavg_kernel,1,ecg_2);
figure;
plot(t,ecg_2);
hold on
plot(t,mov_filter_out);
title('Unfiltered and filtered ECG signals');
xlabel('Time(sec)');
```

```
ylabel('Amplitude');
legend('Unfiltered ECG','Filtered ECG');
hold off
fft_signal_mov_filtered=abs(fftshift(fft(mov_filter_out))/length(mov_filter_out));
figure;
plot(f,fft_signal);
hold on
plot(f,fft_signal_mov_filtered);
title('Unfiltered and filtered ECG signals');
xlabel('Frequency(Hz)');
ylabel('Magnitude');
legend('Unfiltered ECG','Filtered ECG');
hold off

%% e
N    = 5;        % Order
Fc   = 150;       % Cutoff Frequency(For ECG)
flag = 'scale';  % Sampling Flag

% Create 2 window vectors for the design algorithm.
win1 = rectwin(N+1);
win2 = kaiser(N+1,0.5);
% Calculate the coefficients using the FIR1 function.
b1  = fir1(N, Fc/(fs/2), 'low', win1, flag);
b2  = fir1(N, Fc/(fs/2), 'low', win2, flag);
figure;
freqz(b1);
title('Rectwin magnitude and phase response');
figure;
freqz(b2);
title('Kaiser magnitude and phase response');
b1_filtered=filtfilt(b1,1,ecg_2);
b1_and_2_filtered=filtfilt(b2,1,b1_filtered);
figure;
plot(t,ecg_2);
hold on
plot(t,b1_and_2_filtered);
title('Unfiltered and filtered ECG signals');
xlabel('Time(sec)');
ylabel('Amplitude');
legend('Unfiltered ECG','Rectangular and Kaiser Filtered ECG');
hold off

fft_signal_b1_and_b2_filtered=abs(fftshift(fft(mov_filter_out))/length(mov_filter_out
));
figure;
plot(f,fft_signal);
hold on
plot(f,fft_signal_b1_and_b2_filtered);
title('Unfiltered and filtered by two windows ECG signals');
xlabel('Frequency(Hz)');
ylabel('Magnitude');
legend('Unfiltered ECG','Rectangular and Kaiser Filtered ECG');
hold off
msefir=sqrt(mean((mov_filter_out-b1_and_2_filtered).^2));
```

```
%% f
IIR_order=4;
Wp=2*Fc/fs;
[b3,a3]=butter(IIR_order,Fc/(fs/2));
figure;
freqz(b3,a3);
title('Butter magnitude and phase response');
[b4,a4]=cheby1(IIR_order,30,Wp);
figure;
freqz(b4,a4);
title('Cheby1 magnitude and phase response');
[b5,a5]=cheby2(IIR_order,50,Wp);
figure;
freqz(b5,a5);
title('Cheby2 magnitude and phase response');
butter_filtered_signal=filtfilt(b3,a3,ecg_2);
butter_cheby2_filtered_signal=filtfilt(b5,a5,butter_filtered_signal);
mseiir=sqrt(mean((mov_filter_out-butter_cheby2_filtered_signal).^2));

figure;
plot(t,ecg_2);
hold on
plot(t,butter_cheby2_filtered_signal);
title('Unfiltered and filtered ECG signals');
xlabel('Time(sec)');
ylabel('Amplitude');
legend('Unfiltered ECG','Butter and Cheby2 Filtered ECG');
hold off

fft_butter_cheby2_filtered_signal=abs(fftshift(fft(butter_cheby2_filtered_signal))/le
ngth(butter_cheby2_filtered_signal));
figure;
plot(f,fft_signal);
hold on
plot(f,fft_butter_cheby2_filtered_signal);
title('Unfiltered and filtered by two windows ECG signals');
xlabel('Frequency(Hz)');
ylabel('Magnitude');
legend('Unfiltered ECG','Butter and Cheby2 Filtered ECG');
hold off
```