Department of Electronic & Telecommunication Engineering

University of Moratuwa

BM4151 : Biosignal processing

MATLAB Assignment 1 – Digital Filters

Name	Index number
D.S Weerapperuma	180685C

This is submitted as a partial fulfillment for the module BM4151 Biosignal Processing

Department of Electronic and Telecommunication Engineering University of Moratuwa

Date of Submission:

30/11/2022

Table of contents

List of figures

1 Smoothing Filters

1.1 Moving average filter

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(n - k)$$

The moving average filter is simple and effective. We used the above equation to implement the moving average filter. First, we observed the characteristics of the given ECG data included in the "ECG_template.mat" file, and then we added 5dB Gaussian noise to that data. Then we applied the moving average filter we implemented to that noisy data and finally, we observed the characteristics of the filtered data.

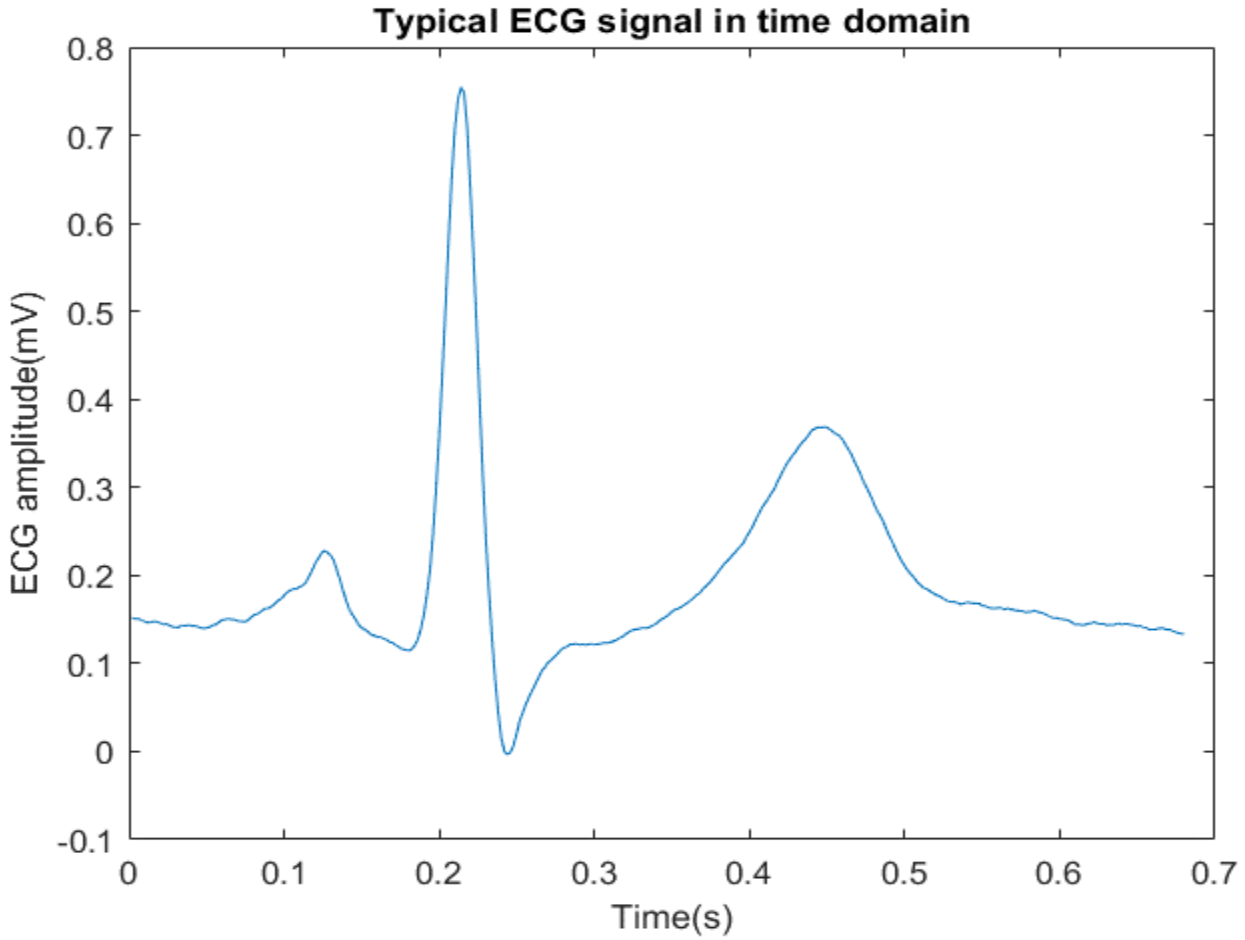

Figure 1.1 Typical ECG signal characteristics

Figure 1.1 shows the loaded ECG signal with the adjusted time scale (according to the sampling frequency of 500Hz). From that figure, we can clearly observe the P wave, QRS complex, and T wave of a typical ECG signal. We added white Gaussian noise of 5 dB using the "**awgn(x, snr, 'measured'**)" command and named the noise-added signal "nECG". Figure 1.2 shows both the noisy signal and the reference signal in the same plot.

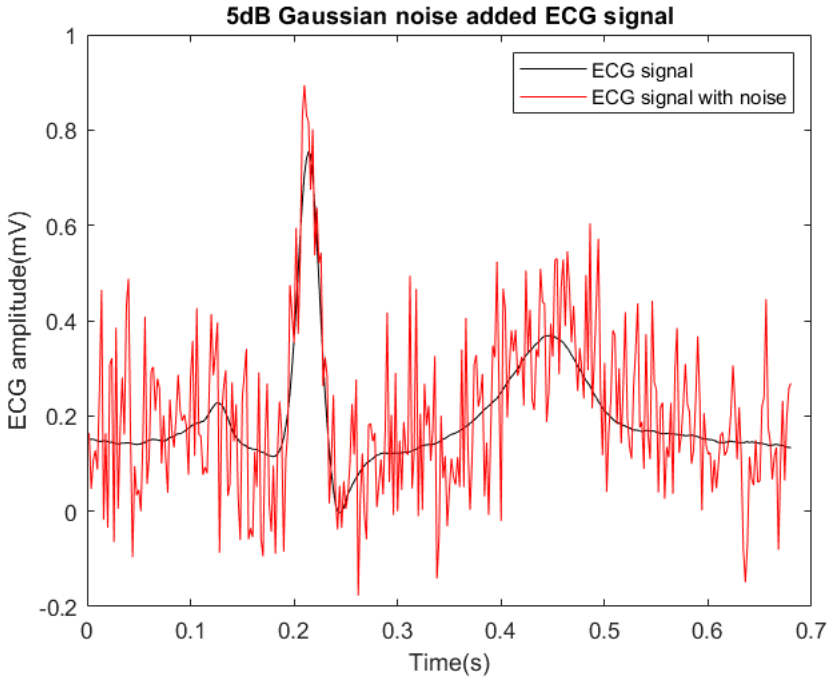

Figure 1.2 Noisy ECG signal and ideal ECG signal

We used "**periodogram(x,window,nfft,fs)"** to generate PSD estimates of the ECG signals. Figure 1.3 shows the power spectral density (PSD) estimate of the noisy ECG.

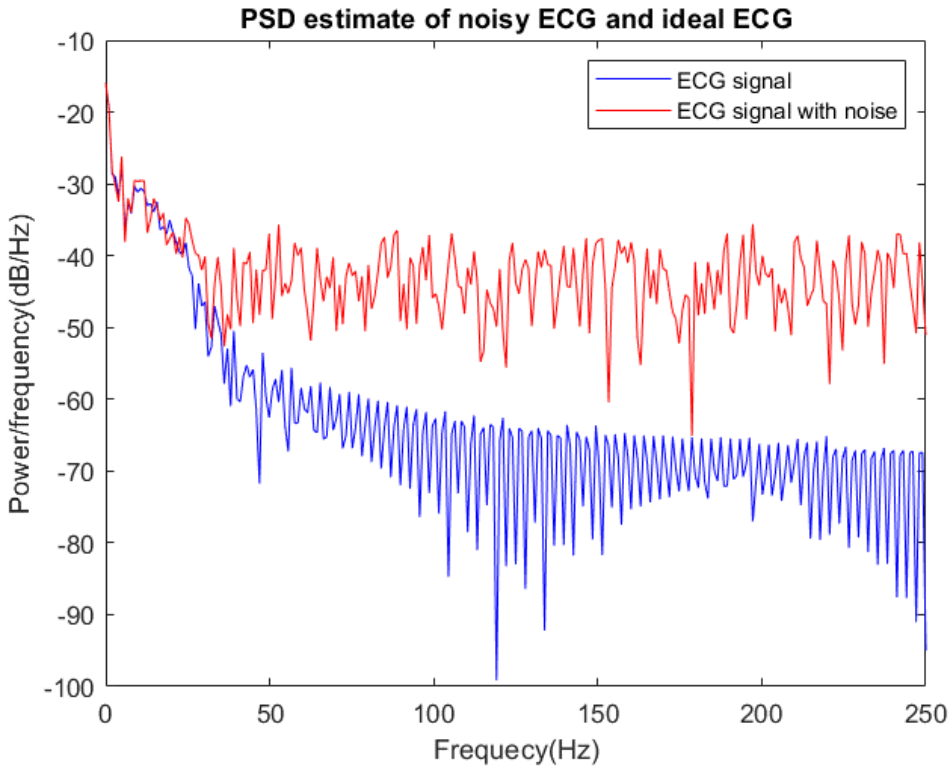

Figure 1.3 PSD estimate of ideal ECG and noisy ECG

According to figure 1.3, we can see that lower frequencies (less than 40Hz) have more power compared to the higher frequencies in the original EEG signal. The noisy ECG signal has more power compared to the original EEG signal at higher frequencies because of the additional noise.

MA(3) filter implementation with a customized script

Derive the Group Delay

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(n-k) \quad \xrightarrow{\mathcal{F}} \quad Y(\omega) = \frac{1}{N}\sum_{k=0}^{N-1} X(\omega)\, z^{-k}$$

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{1}{N}\sum_{k=0}^{N-1} z^{-k}$$

$$H(\omega) = \frac{e^{-j(\frac{N-1}{2})\omega}}{N} \sum_{k=-(N-1)/2}^{(N-1)/2} e^{-j\omega k}$$

$using\ \ e^{j\theta} = \cos\theta + j\sin\theta\ , we\ can\ simplify\ above\ equation\ as\ follows,$

$$H(\omega) = \frac{\cos\left(\frac{N-1}{2}\right)\omega - j\sin(\frac{N-1}{2})\omega}{N} \sum_{k=-(N-1)/2}^{(N-1)/2} 2\cos k\omega$$

$$Arg\big(H(\omega)\big) = tan^{-1}\left(\frac{Im\big(H(\omega)\big)}{Re\,\big(H(\omega)\big)}\right)$$

$$Arg\big(H(\omega)\big) = tan^{-1}\left(\frac{\cos\left(\frac{N-1}{2}\right)\omega}{-\sin\left(\frac{N-1}{2}\right)\omega}\right)$$

$$Arg\big(H(\omega)\big) = tan^{-1}\left(\tan\left(\pi + \left(\frac{N-1}{2}\right)\omega\right)\right)$$

$$Arg\big(H(\omega)\big) = \pi + \left(\frac{N-1}{2}\right)\omega \qquad \longrightarrow \qquad (1)$$

The definition of the group delay as follows.

$$Group\ Delay\ (\tau_g) = \frac{\partial\, Arg\big(H(\omega)\big)}{\partial\omega} \qquad \longrightarrow \qquad (2)$$

From (1) and (2) ,

$$Group\ Delay\ (\tau_g) = \frac{N-1}{2}$$

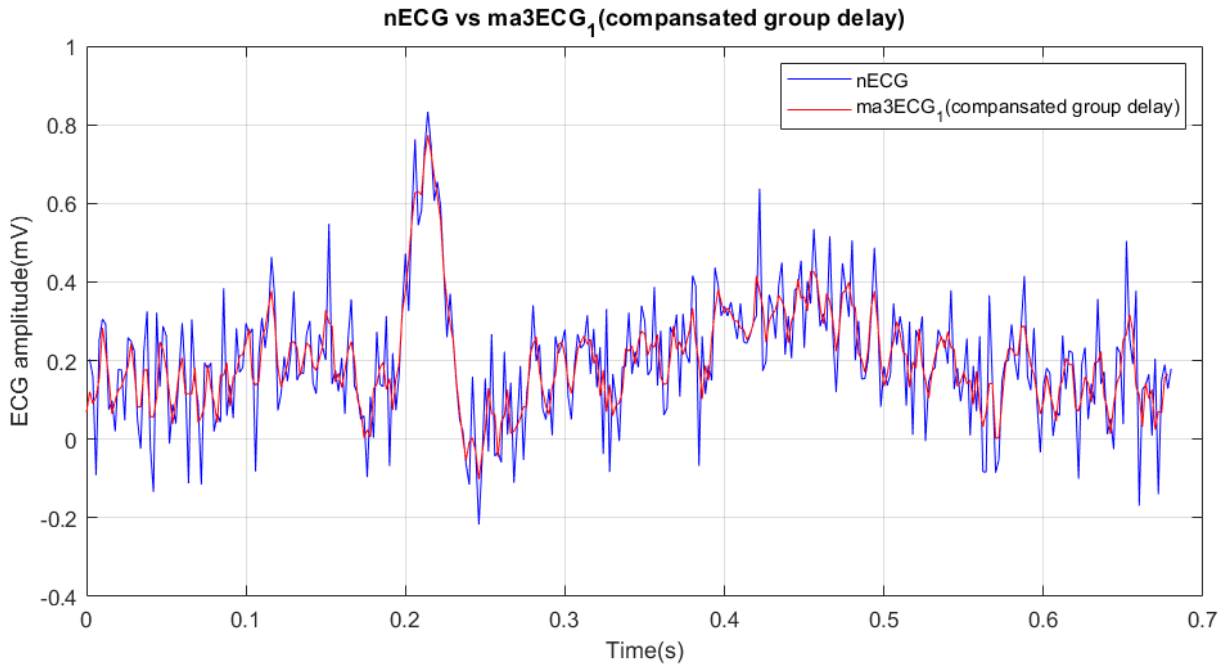

Figure 1.4 noisy ECG and delay compansated MA(3) output signal

Since the filter order is 3 the group delay will be 1 (according to the derived group delay). When we look at figure 1.4 we can see that the filtered signal is smoothened and the large dynamic amplitude changes are reduced.

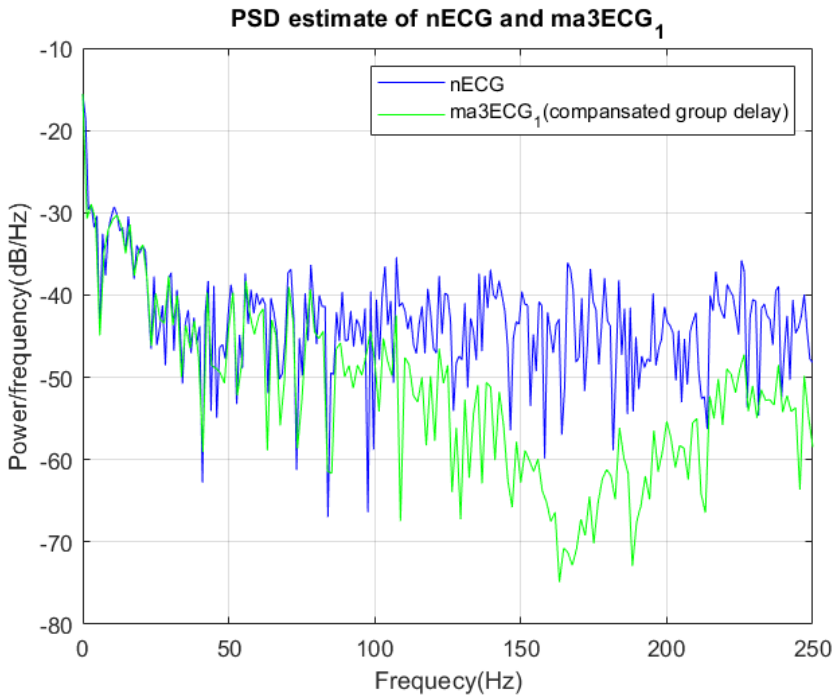

Figure 1.5 PSD estimate of noisy ECG and delay compansated filter output

From the above PSD plot, we can clarify that the moving average filter is work as a smoothing filter. It shows that the filter attenuated the high-frequency components of the noisy signal and was not affected by the lower frequencies(less than 40HZ). So the MA(3) filter preserves the necessary data of the ECG signal and reduced the noise.

MA(3) filter implementation with the MATLAB built-in function

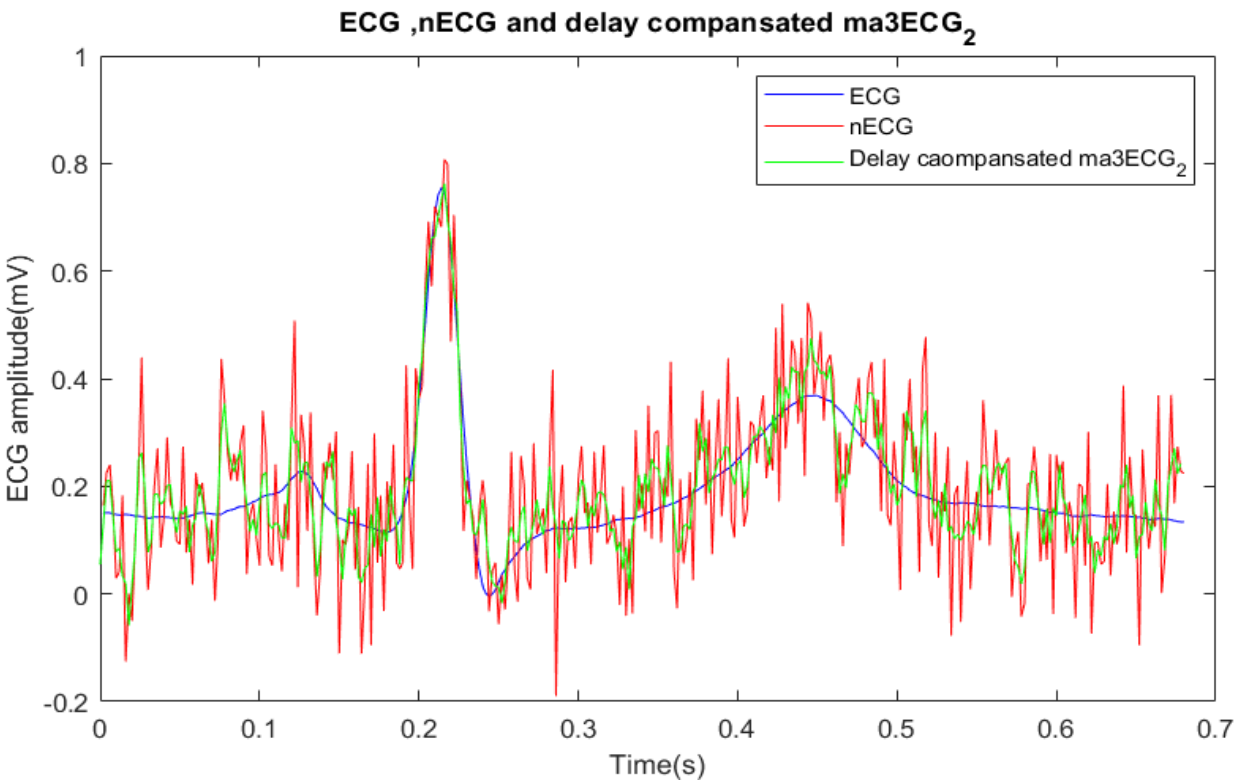

Figure 1.6 original ECG, noisy ECG ,ma3ECG_2

We can simply compare the original signal, filtered signal, and filtered signal using figure 1.6. The filtered signal is smoothened compared to the noisy signal but still, it has variations concerning the original signal.

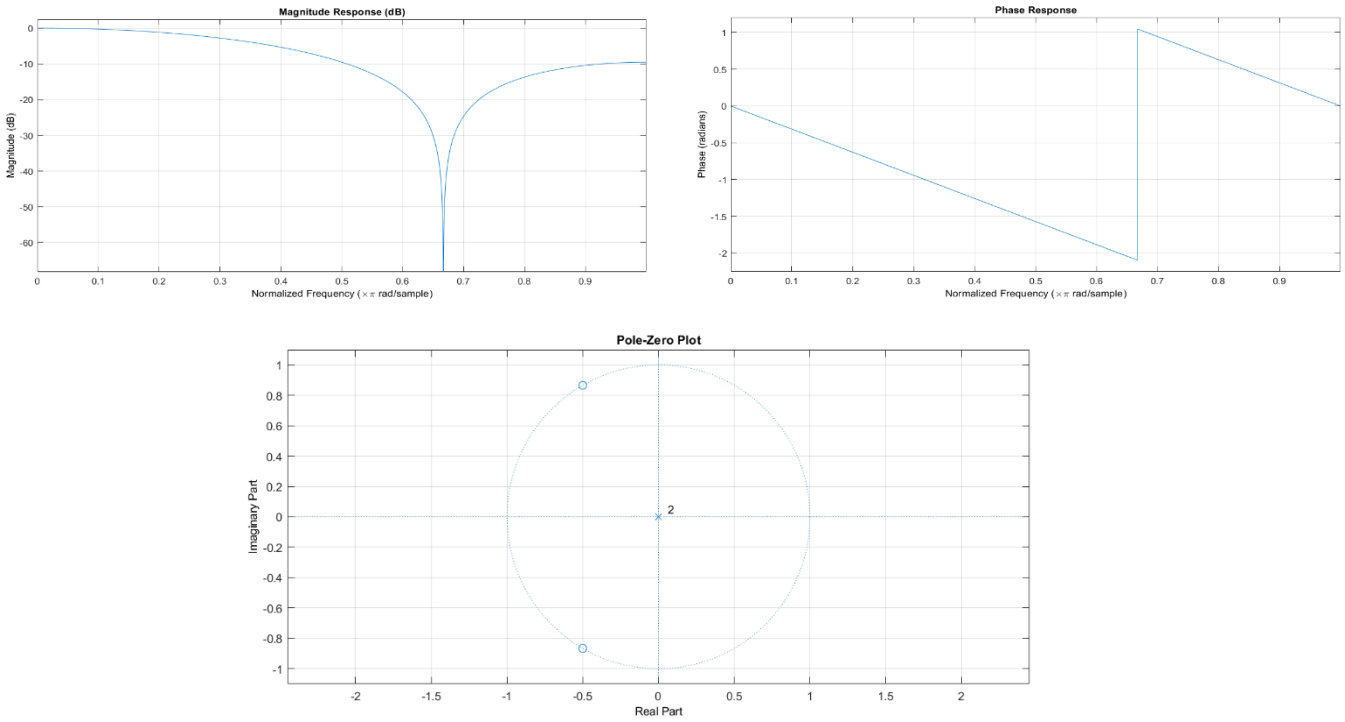

Figure 1.7 Magnitude Response, Phase Response & Pole - Zero Plot of MA(3) filter

There are only two zeros on the unity circle since this was a third-order filter.

MA(10) filter implementation with the MATLAB built-in function

Since MA(10) has higher order than MA(3) it is obvious that MA(10) reduces more amount of noise compared to the MA(3) filter. We can clarify that by observing the following figure 1.8 and compare with figure 1.7.

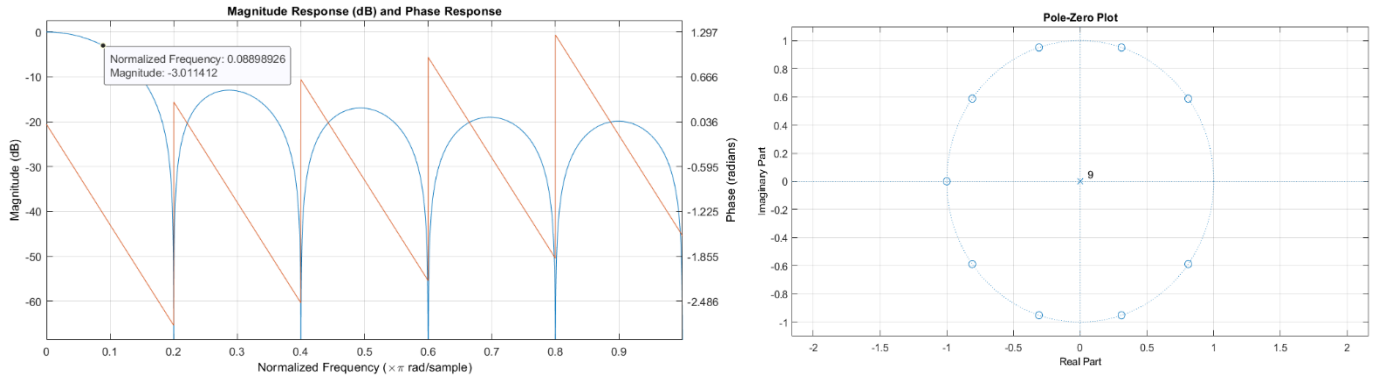

Figure 1.8 Magnitude Response, Phase Response & Pole - Zero Plot of MA(10) filter

By observing Figure 1.8 of the MA(10) we can identify that the threshold of -3dB attenuation is occur around 0.08 of the normalized frequency. For MA(3) filter this frequency will be 0.3 of the normalized frequency. So that MA(10) reduced a large amount of higher frequency noises w.r.t. to MA(3).

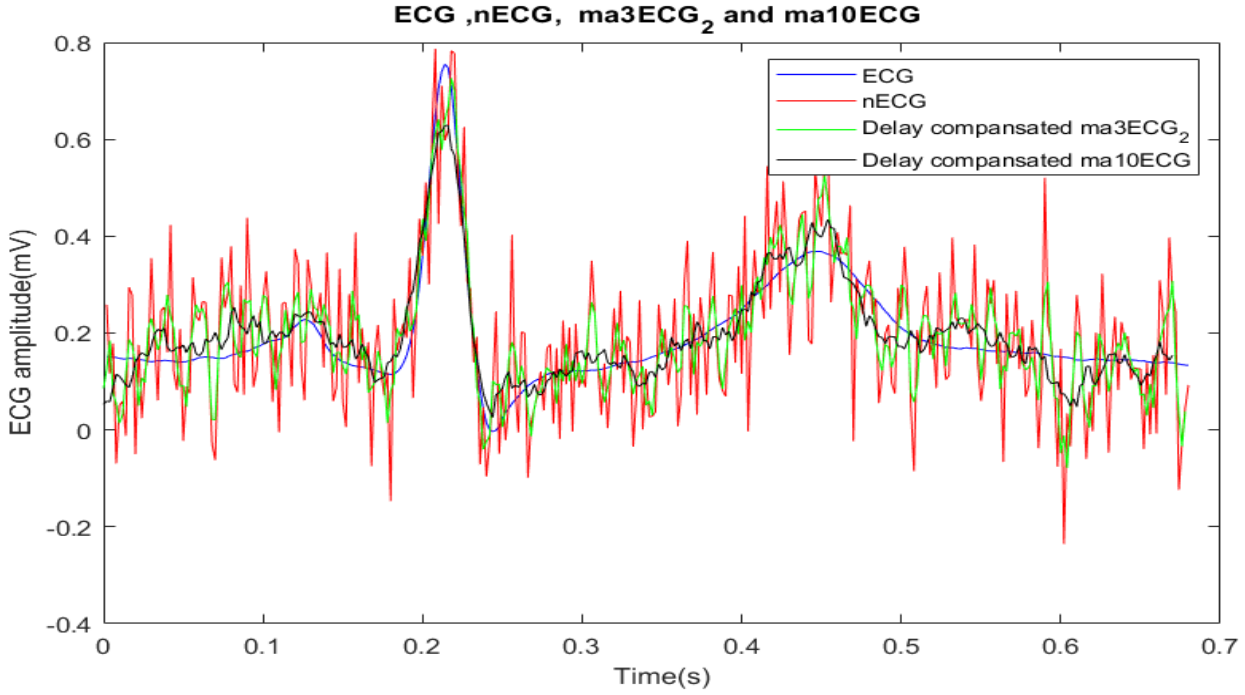

Figure 1.9 Original ECG, noisy_ECG, ma3ECG_2 and ma10ECG

According to Figure 1.9, we can clarify that MA(10) filter generates a smoother noise signal compared to the MA(3) filter as stated before when comparing magnitude plots. Since the necessary details of the ECG signal are contained in the lower frequencies, most of the higher frequencies are related to the noises. Because MA(10) attenuates a large number of higher frequencies from a greater amount, the output of this filter is smoothened compared to MA(3) and more close to the original ECG signal. But because of the averaging process, MA(10) smooths the peaks of the output signals which can be affected negatively at some points.

Optimum MA(N) filter order

Reducing the order of the MA filter cause to produce noisy output. Also increasing the order of the filter cause to smooths the peaks of the original non-noisy signal which may contain important details. So that it is very important to find the optimum order for the MA filter which gives a more clear and more detailed output. We can find the optimum filter order by observing the following graph figure 1.10.

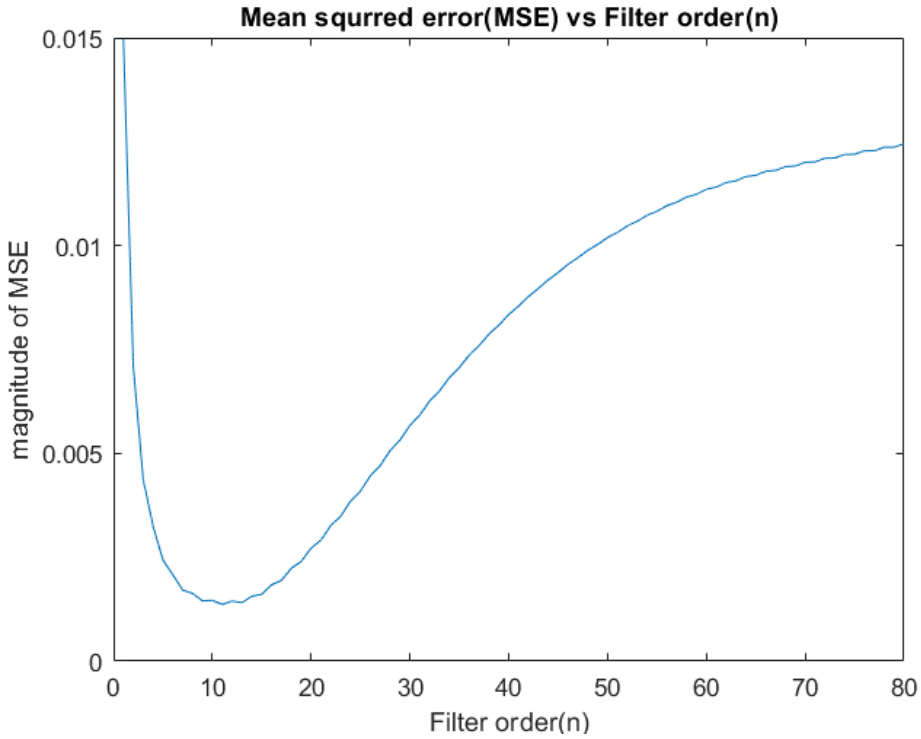

Figure 1.10 MA filter order vs MSE

Order of optimum MA filter = 11

When we reduce the order of the MA filter the number of frequencies attenuated will be decreased. So that the low pass cut-off will be increased. As a result, the signal will contain a considerable amount of noise. When we increase the order of the filter it will attenuate a larger amount of high frequencies so that the cut-off will be reduced. This may cause the removal of the important frequency components and the details from the original signal. From both these ways(increasing order or decreasing order) the output signal will more deviate from the original signal. So MSE values will be larger in lower orders and higher orders(figure 1.10).

1.2 Savitzky-Golay SG(N,L) filter

Savitzky-Golay filter fits a polynomial of order N to an odd number of data points L'=2L+1(where L' is an odd integer) in the predefined window in a least-squares sense. A unique solution requires $N \leq 2L$. We implemented SG filter using sgolayfilt(x,N, L') in Matlab and Figure 1.11 shows the comparative results.

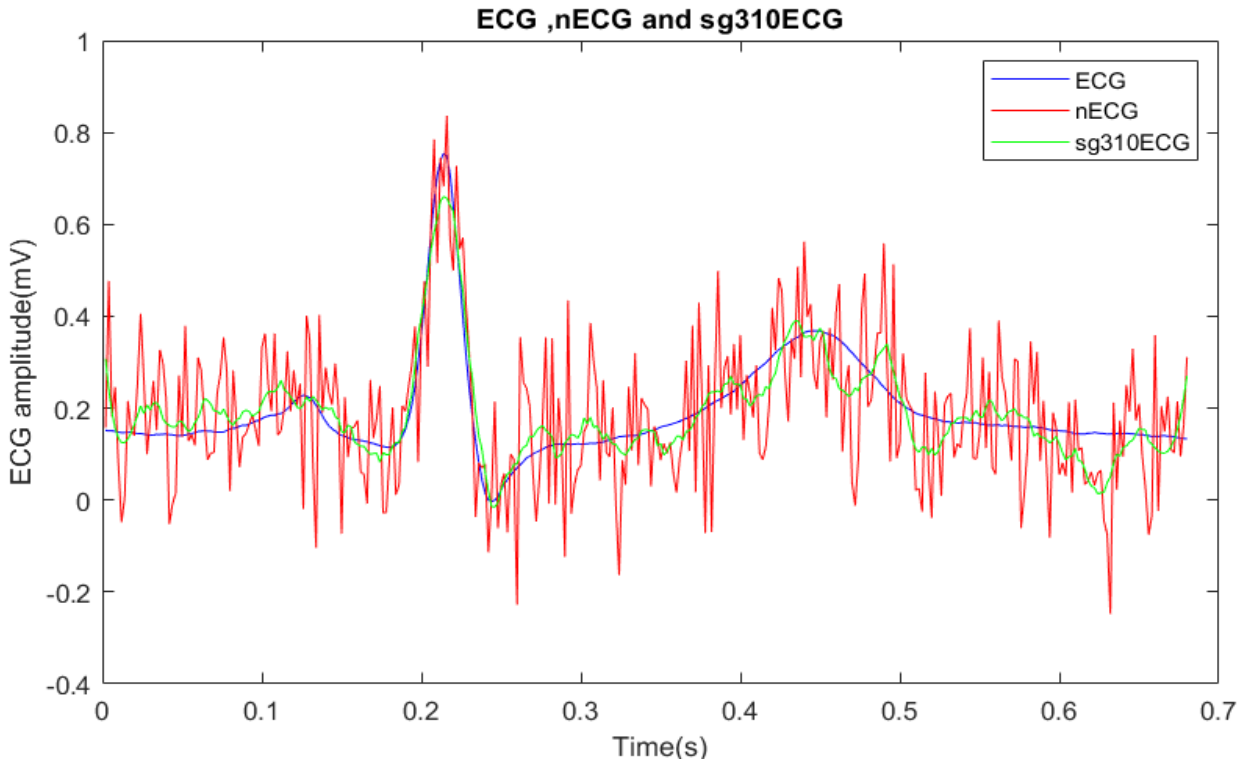

Figure 1.11 nECG , ECG template and sg310ECG signals

By observing the above figure we can verify that the Savitzky-Golay filter is work as a smoothing filter since it is reducing the noise from the original ECG signal.

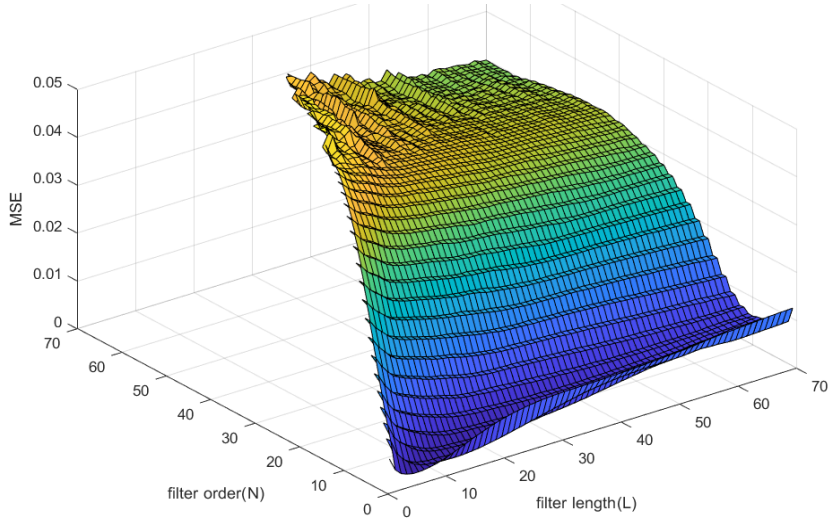

Figure 1.12 MSE vs Order vs Length of Saviztsky Golay Filters

Optimum Polynomial Order(N) = 2

Optimum Length of the Window (L) = 12

When we plotted SG(3,10) and optimum SG in the same plot we can see that the optimum SG filter reduces more noise than SG(3,10) and smoothen the signal. Also, it detects and smooths more peaks than SG(3,10) filter.

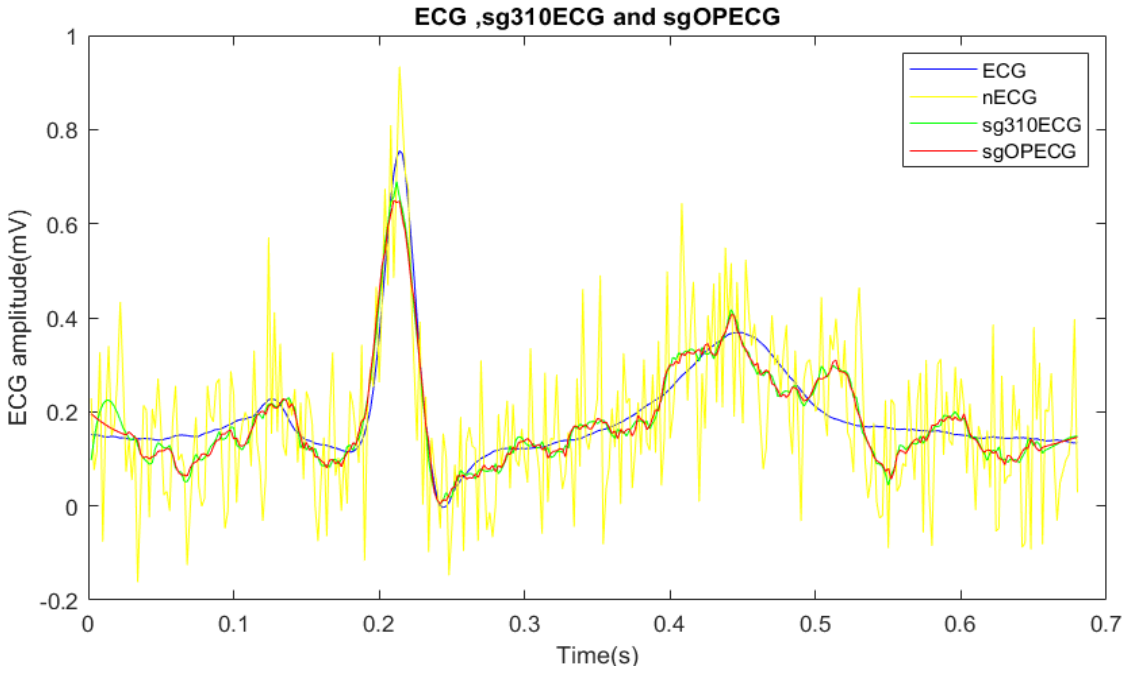

Figure 1.13 ECG_template, sg310ECG and opt_sgECG

We can compare the Optimum outputs of the MA filter and SG filter using the following figure and time elapsed values.

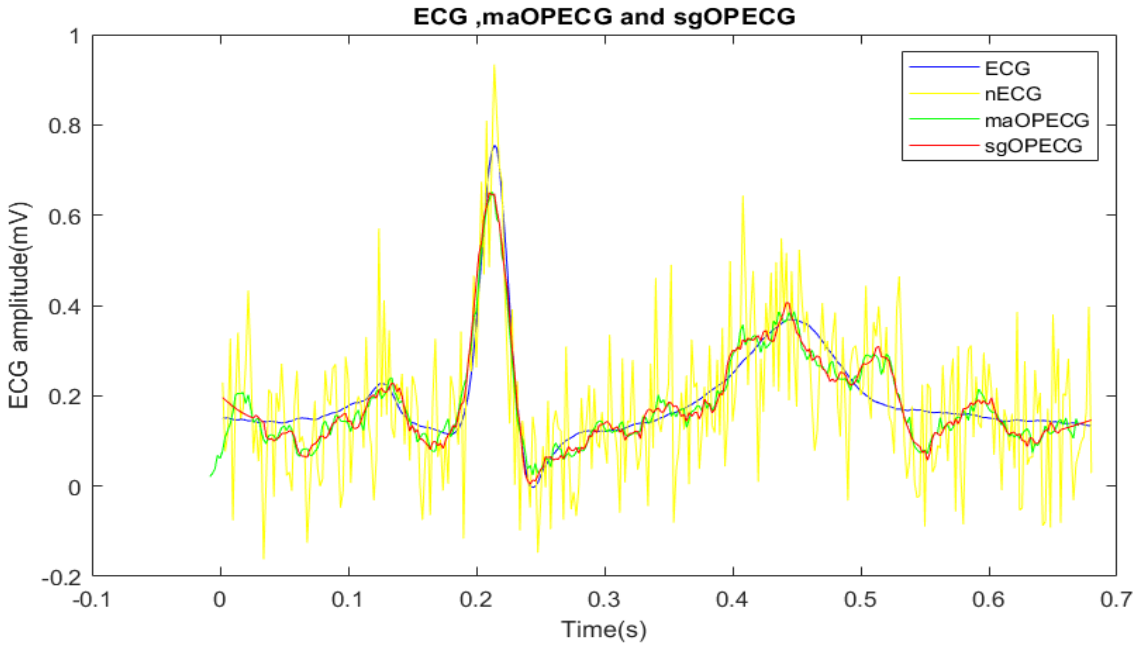

Figure 1.14 ECG, nECG, optimum MA and SG filter

Time elapsed for optimum **MA Filter** = 0.000080 S

Time elapsed for optimum **SG Filter** = 0.000477 S

According to figure 1.14, we can see that the SG filter generated more smooth signal compared to the MA filter. But in the time domain, the MA filter is more efficient than the SG filter since it consumes a lower time duration for the filtering process compared to the SG filter(6 times faster than the SG filter).

2 Ensemble averaging

2.1 Signal with multiple measurements

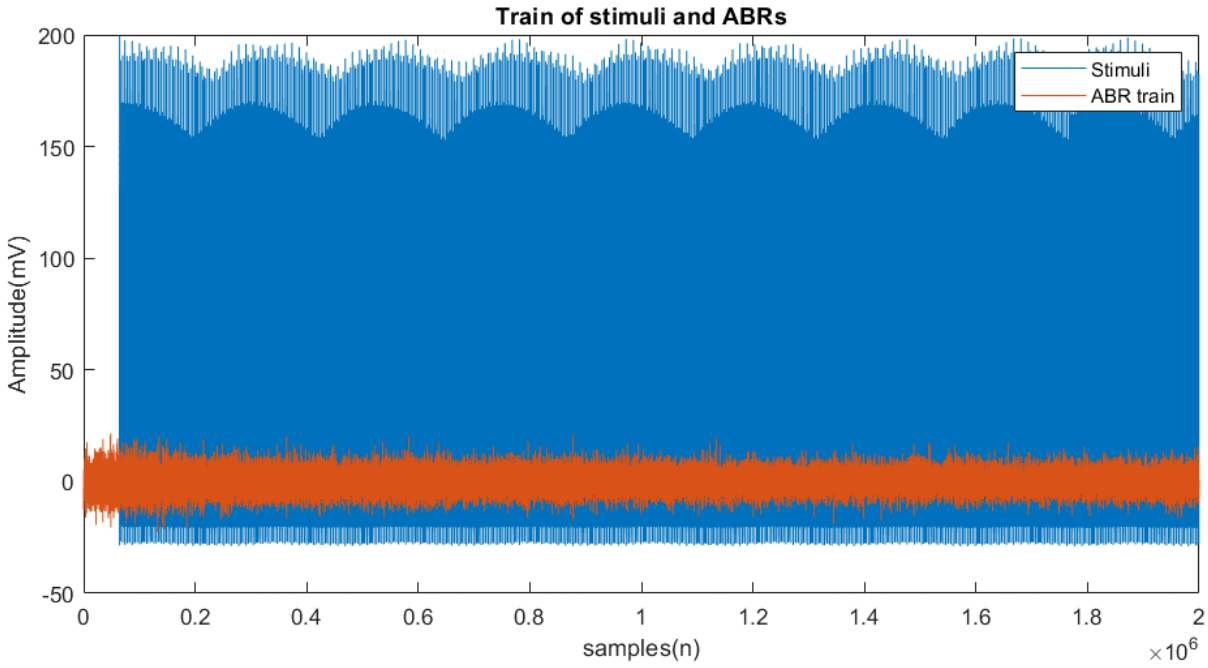

Figure 2.1 Train of ABR recording and Audio Pulse Stimuli

It is difficult to observe ABR and pulse trains from Figure 2.1. Therefore We zoomed the graph along the axis as follows, so that it enables us to get clear observations.

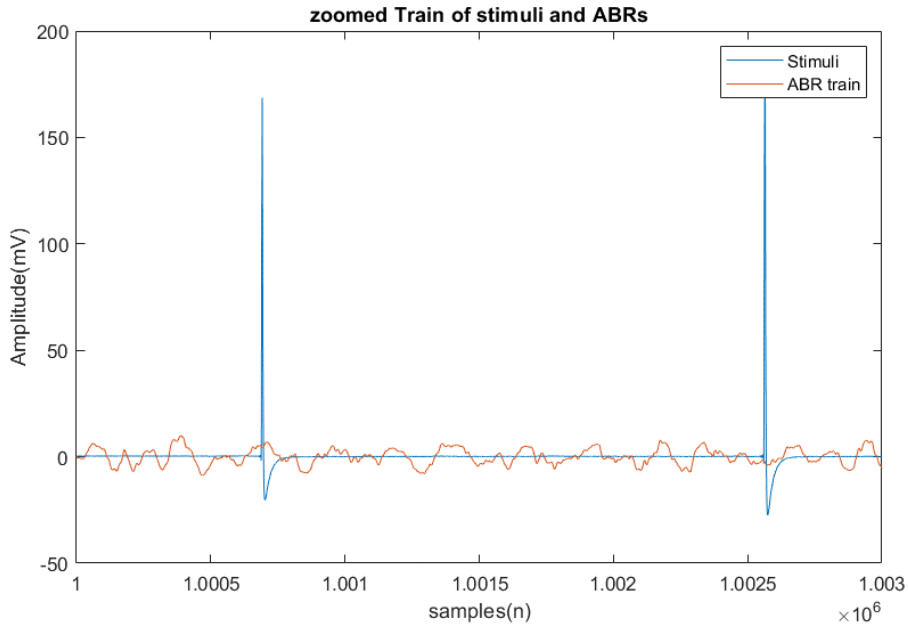

Figure 2.2 Zoomed Train of ABR recording and Audio Pulse Stimuli

From Figure 2.2 we can see the respective ABR signal recording and the stimulus separately.

It is necessary to extract epochs from the recording in a proper way to calculate the ensemble average. After determining a voltage threshold to automatically detect the likely stimuli occurrences we window ABR epochs according to the extracted stimulus points. We Considered the window length to be 12ms (-2ms to 10ms) with reference to the stimulus time point (i.e. -80 to 399 sample points at fs=40 kHz). Then we calculated the ensemble average from those epochs and plotted the final results in the following figure.

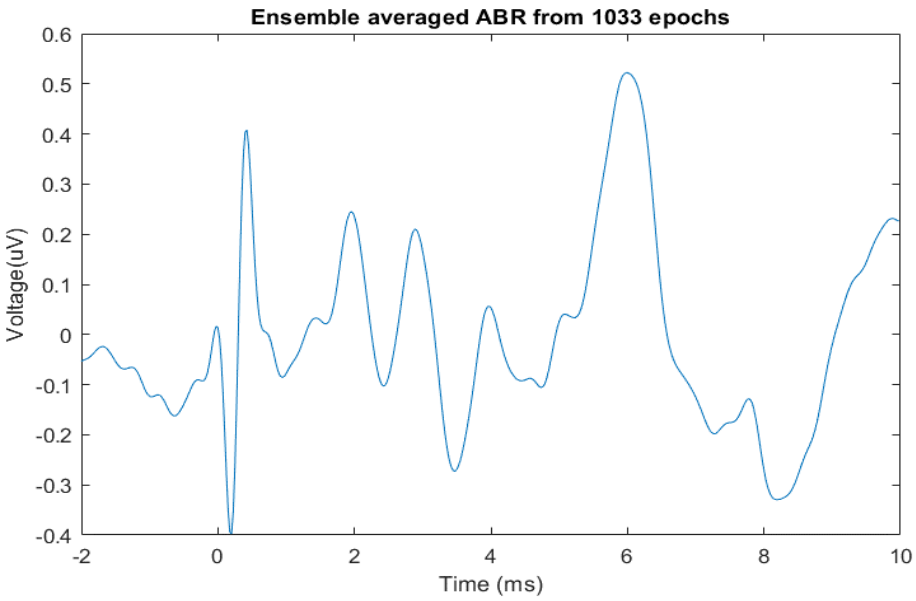

Figure 2.3 Ensemble averaged ABR

Improvement of the SNR

We implemented a code that calculates progressive MSEs for the given number of epochs. Then we plotted a graph of MSE_k against k for analyzing purposes as follows.

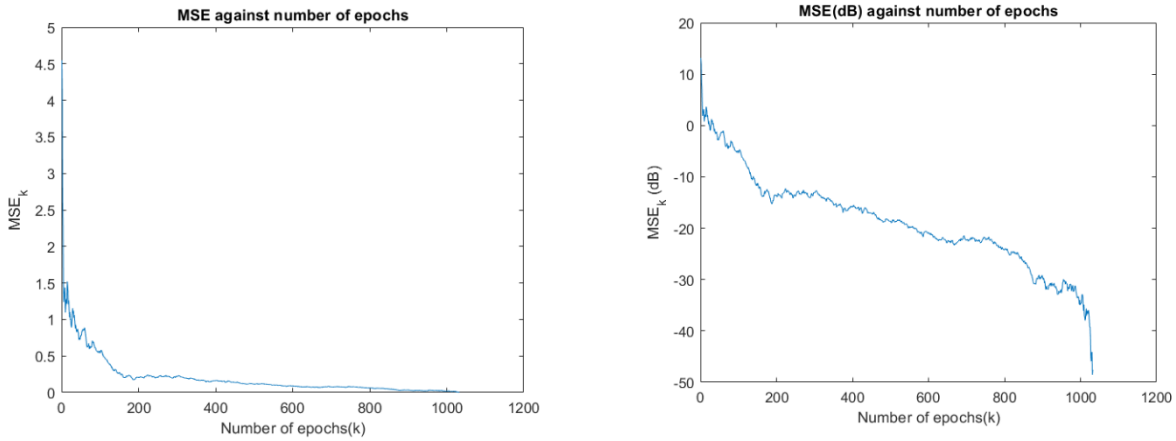

Figure 2.4 MSE vs number of epochs

According to Figure 2.4, we can see the MSE will converge to 0 when we increase the number of epochs to calculate the ensemble average. MSE is almost zero when we use a very large number of epochs (>1000). This was also validated by observing the logarithmic plot as well. Because when the number of epochs is very high these practical values are reaching to theoretical values (expected values).

Let's derived an expression for SNR theoretically since it is very helpful for better understanding.

According to the assumptions of the ensemble averaging let's say that signal S_j same for every epoch. Therefore total summation of signals(S_k) from k number of epochs,

$$S_k = \sum_{j=1}^{k} S_j = kS_j$$

Since we have the same signal in every epoch the variance depends on the noise. We can calculate the variance of the summed signal(σ_k^2) using the variance of the noise (σ_j^2),

$$\sigma_k^2 = \sum_{j=1}^{k} \sigma_j^2 = k\,\sigma_j^2 \Rightarrow \sigma_k = \sqrt{k}\,\sigma_j$$

Therefore SNR can be calculated as follows.

$$SNR_k = \frac{S_k}{\sigma_k} = \frac{kS_j}{\sqrt{k}\,\sigma_j} = \sqrt{k}\,\frac{S_j}{\sigma_j} = \sqrt{k}\,SNR_j$$

According to the equation we derived for the SNR, we can observe that SNR is proportional to the number of epochs we use for the calculate ensemble average. We can convert this equation to a logarithmic scale as follows.

$$SNR_k = \left(\frac{1}{2}\right) 20\log k + 20log\,(SNR_j) = 10\log k + 20log\,(SNR_j)$$

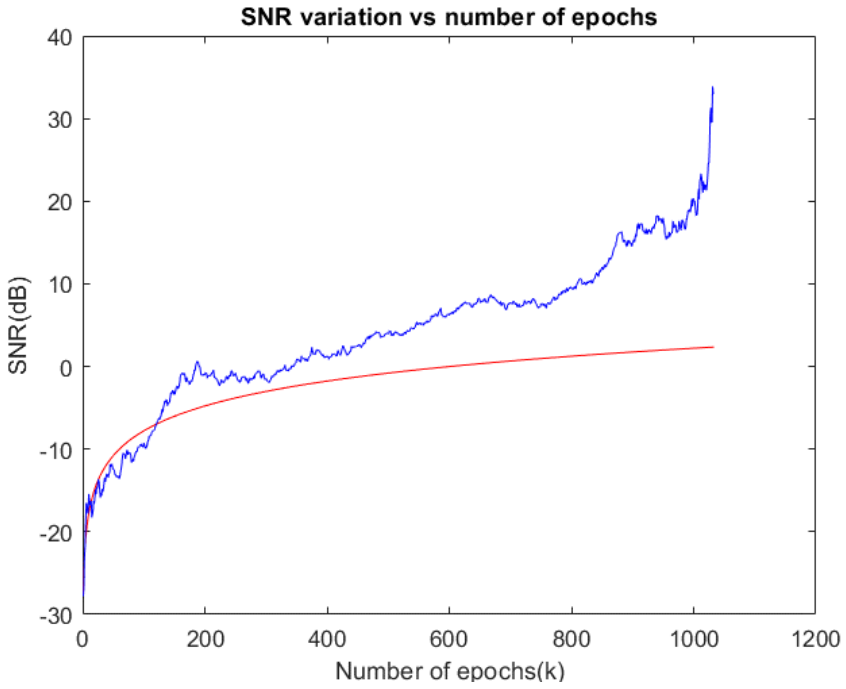

Figure 2.5 SNR variation vs number of epochs

2.2 Signal with repetitive patterns

Viewing the signal and adding Gaussian white noise

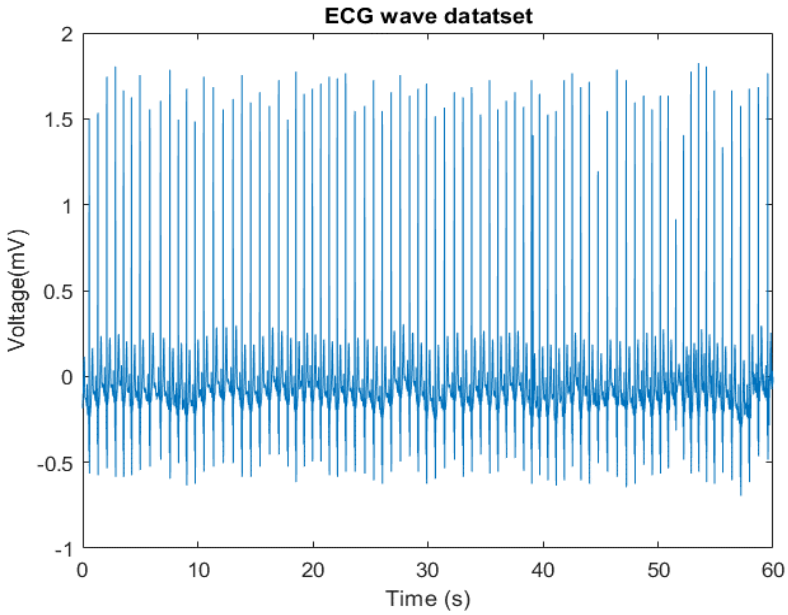

Figure 2.6 Complete ECG recording

For observation purposes and the next steps, we extracted 1.6 S time duration. This extracted portion is shown in figure 2.7.

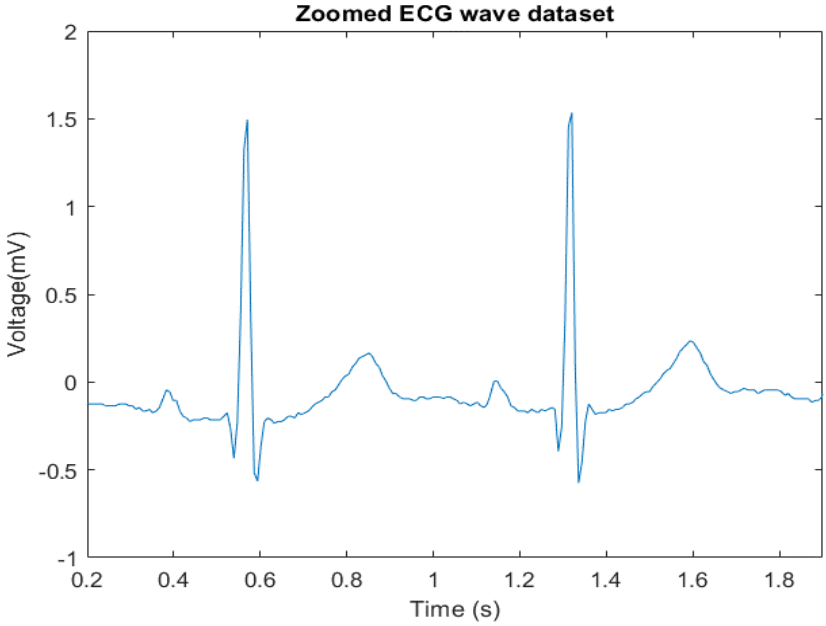

Figure 2.7 ECG waveform in zoomed time axis

We extracted a single PQRST waveform by observing the previous figure. So that we used the 2nd peak of that figure and there are 34 data points before the peak point and 65 data points after the peak point includes to the PQRST wave. Therefore selected waveform started at 134 sample points and end at 233 sample points(233 -134 + 1 = 100 points). This selected waveform is shown in figure 2.8.

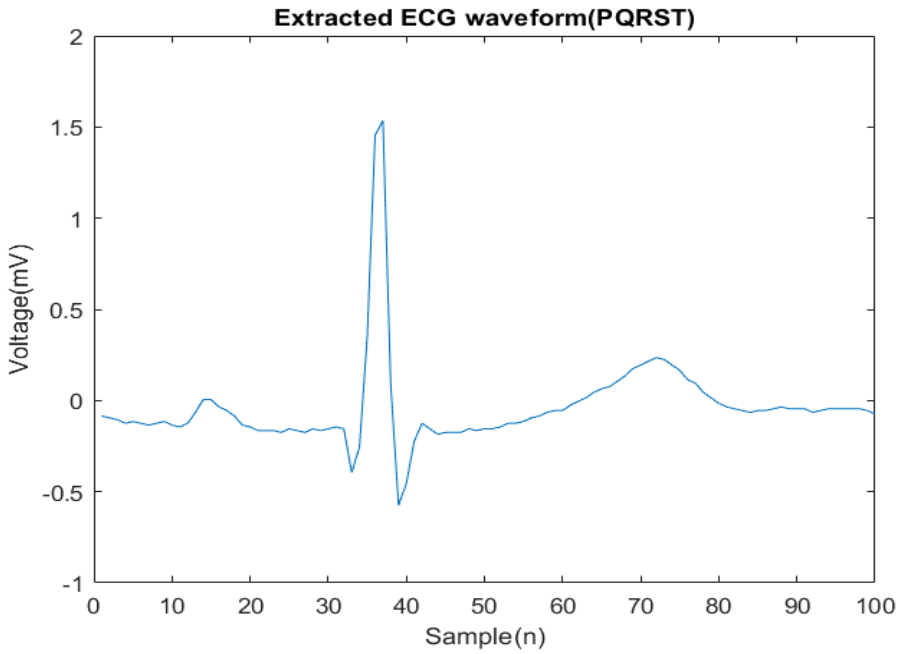

Figure 2.8 Extracted waveform for the template

Now we add 5dB of noise to the ECG recording so that we can use that noisy signal to evaluate the ensemble filter. This noisy signal shown figure in figure 2.9.

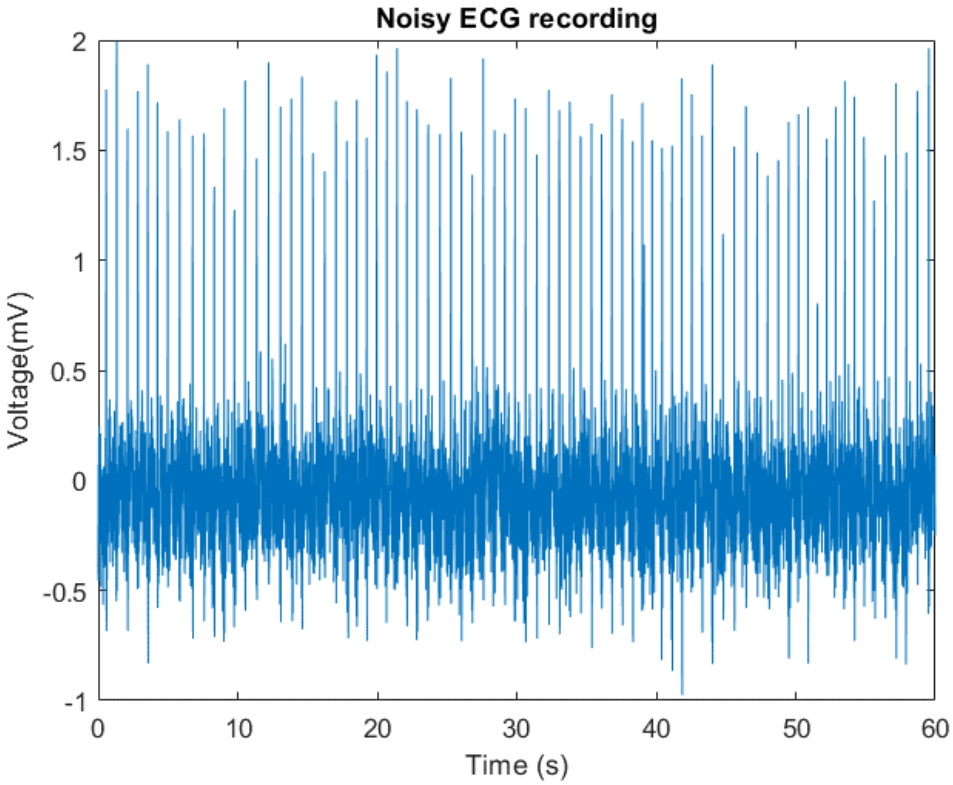

Figure 2.9 ECG recording with noise

Segmenting ECG into separate epochs and ensemble averaging

given that an ECG template already exists, it is a better method to use points of maximum correlation with a noisy ECG pulse train rather than merely detecting the R-wave to segment the ECG pulse train into separate epochs. So that we calculated the normalized cross-correlation between the ECG_template and the nECG using "xcorr(x,y) " with some modifications. Figure 2.10 shows the normalized cross-correlation values against the adjusted lag axis converted to the time axis.

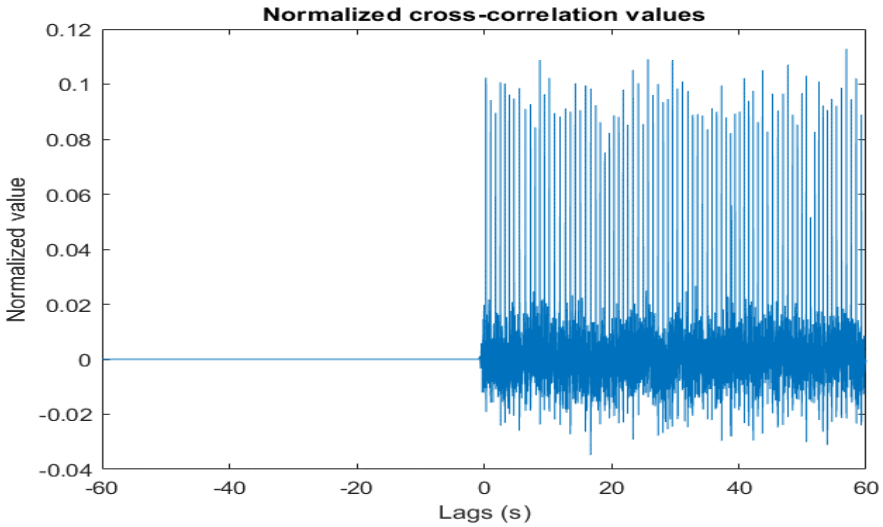

Figure 2.10 Normalized cross corrilations values aginst the adjsuted lag axis in the time domain

By observing the above graph carefully we decided the threshold as 0.08 for normalized cross-correlation to segmentation purposes. Using that threshold we extracted ECG epochs from the noisy ECG dataset. If there are adjustment cross-correlation values that have cross-correlation more than the threshold we selected the highest point of correlation to the segmentation process.

After that we calculate and plot the improvement in SNR as the number of ECG pulses included in the ensemble average is increasing. The result is shown in figure 2.11

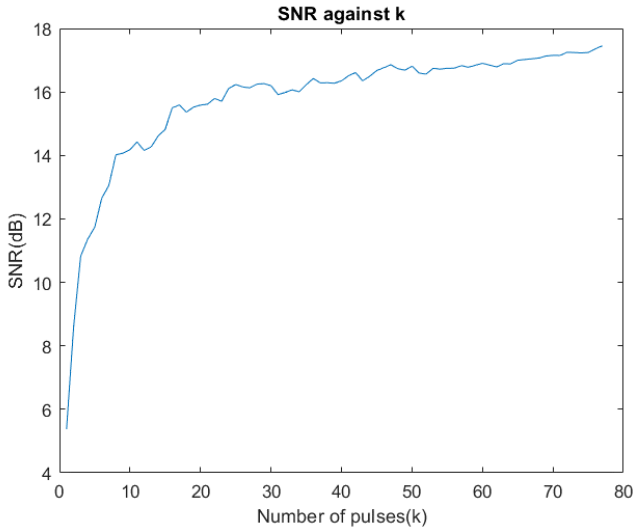

Figure 2.11 Improvement of SNR when increasing number of pulses

We plotted ensemble arbitrarily selected averaged pulses(70 and 5 pulses used for calculating ensemble averages), noisy ECG pulses as well as template ECG pulses in the same graph for comparison purposes.

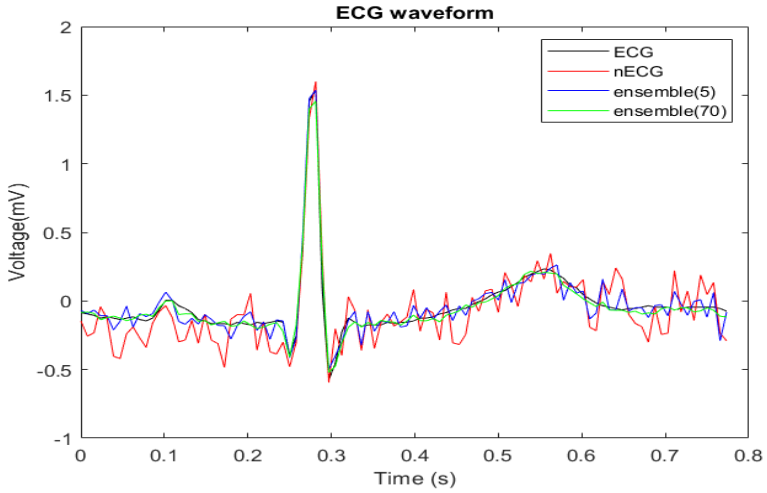

Figure 2.12 ECG Sample and Ensemble Averages

According to Figure 2.12, we can see that there is a more smooth ensemble(70) compared to the ensemble(5). Also, the output of the ensemble(70) is more similar to the ECG template. So that we can conclude that the increase in the number of pulses used for the ensemble average causes to generate a smoother and more similar version of the ECG template.

Why its better to use cross-correlation rather than peak detection?

- The length time window can vary at different PQRST waves
- There are some abnormal wave rejections

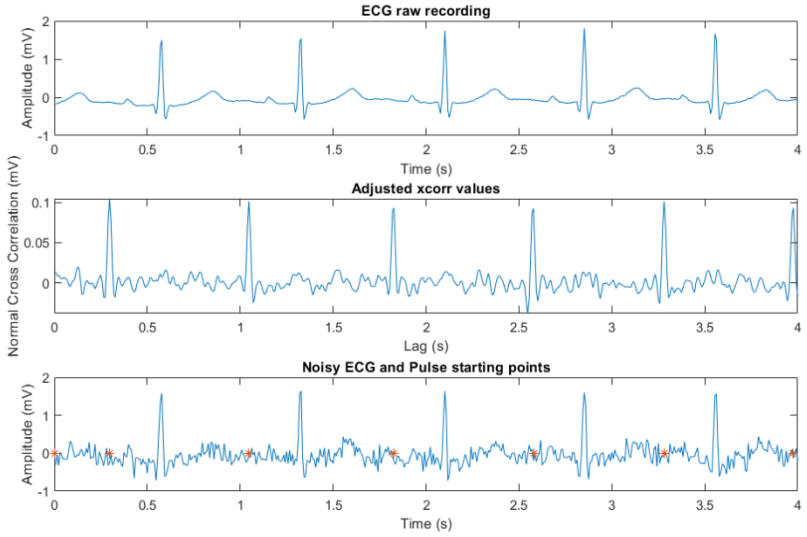

Figure 2.13 cross correlation pulse detection

According to Figure 2.13, we can clearly see that higher cross-correlation values are occurred at when the template signal more coincides with the noise signal. So as shown in the third figure of figure 2.13, by choosing the correct threshold and detecting peaks of cross-correlations we can easily detect all the pulses correctly.

3 FIR derivative filters

3.1 FIR derivative filter properties (use the fvtool(b,a))

We can design a different kinds of FIR filters by changing their filter coefficients. Depending on the coefficients all the filter characteristics will be changed. The moving average filter has the same coefficients with all the terms. This limits its potential for filtering so that it only performs well in some limited cases. The derivative-based filter is one kind of different coefficients filter based on the derivatives of the signal.

The first order derivative of discreet signal with sampling period T is

$$y(n) = \frac{x(n) - x(n-1)}{T}$$

Let's convert this time domain function to the frequency domain and let's see their coefficients.

$$Y(z) = \frac{X(z) - X(z).z^{-1}}{T}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1-.z^{-1}}{T} = \frac{1}{T}(1 - z^{-1})$$

In this case, $b_0 = 1$, $b_1 = -1$, and other coefficients are 0. As same with this, we can write a central difference equation with a Sampling period of T as follows.

$$y(n) = \frac{x(n+1) - x(n-1)}{2T}$$

If we ignore the time shift of the central difference and consider current and past values for 3 points central difference, we modify the above eq. as follows.

$$y(n) = \frac{x(n) - x(n-2)}{2T}$$

By following the process of generating the transfer function as we followed in the first-order derivative we can obtain the transfer function for the 3 points central difference shown as follow.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1-.z^{-2}}{2T} = \frac{1}{2T}(1 - z^{-2})$$

In this case, $b_0 = 1$, $b_1 = 0$, $b_2 = -1$, and other coefficients are 0.

Let's plot and compare 1st order and 3-point central difference filters.

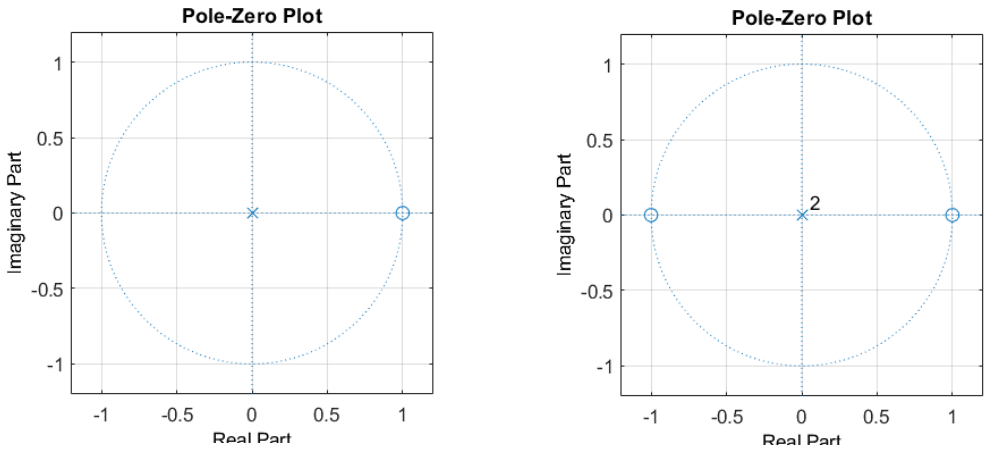

Figure 3.1 1st order and 3-point central different filters pole-zero plots

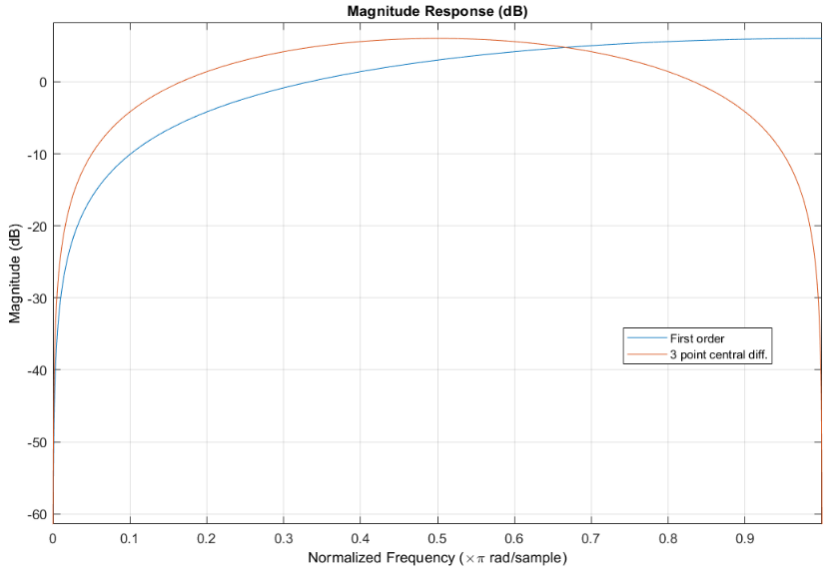

Figure 3.2 1st order and 3-point central diff. magnitude responses(dB)

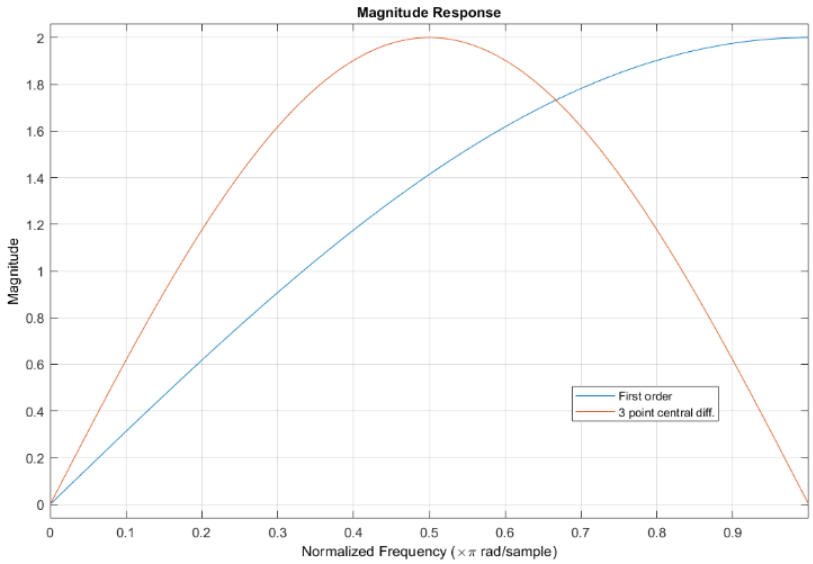

Figure 3.3 1st order and 3 point central diff. magnitude responses

By observing the above magnitude response figure, we can see that the first-order filter employs a high pass filter while 3 points central difference filter work as a bandpass filter. Also, we can see that there is some amplification in the pass band of the filter outputs. Since this amplification is not the same for all the frequencies in the pass band this leads to distortions in the pass band. Therefore, we can compensate for this amplification by introducing a scaling factor for this.

We can find the scaling factor using the following equation.

$$G = \frac{1}{max(|H(z)|)}$$

For the first-order derivative filter,

$$|H(z)| = \frac{1}{2T}|(1 - z^{-1})|$$

When Z = -1, |H(z)| will be maximum. Therefore,

$$G = \frac{1}{max(|H(z)|)} = \frac{T}{|(1-(-1)^{-1})|} = \frac{T}{2}$$

We can rewrite the time domain equation of the first-order derivative filter as follows.

$$y(n) = \frac{x(n) - x(n-1)}{2}$$

For 3 point central difference derivative filter,

$$|H(z)| = \frac{1}{2T}|(1 - z^{-2})|$$

When Z = j |H(z)| will be maximum. Therefore,

$$G = \frac{1}{max(|H(z)|)} = \frac{2T}{|(1-(j)^{-1})|} = T$$

We can rewrite the time domain equation of 3 points central difference derivative filter as follows.

$$y(n) = \frac{x(n) - x(n-2)}{2}$$

After compensating unnecessary amplification by including the scaling factors we can obtain following filter magnitude responses. Note that the maximum response of the filter is 0dB(no amplification).

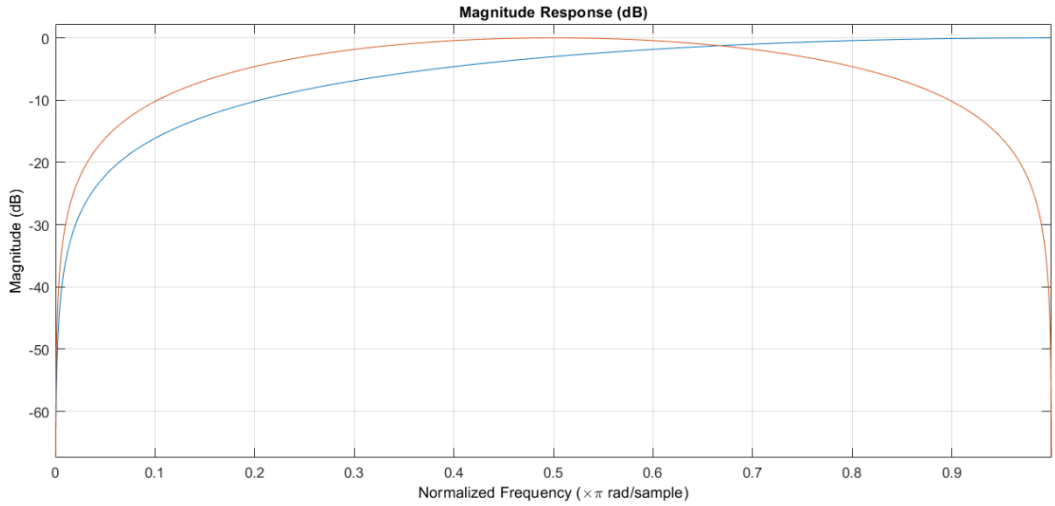

Figure 3.4 1st order and 3 point central diff. magnitude responses(dB) scaled

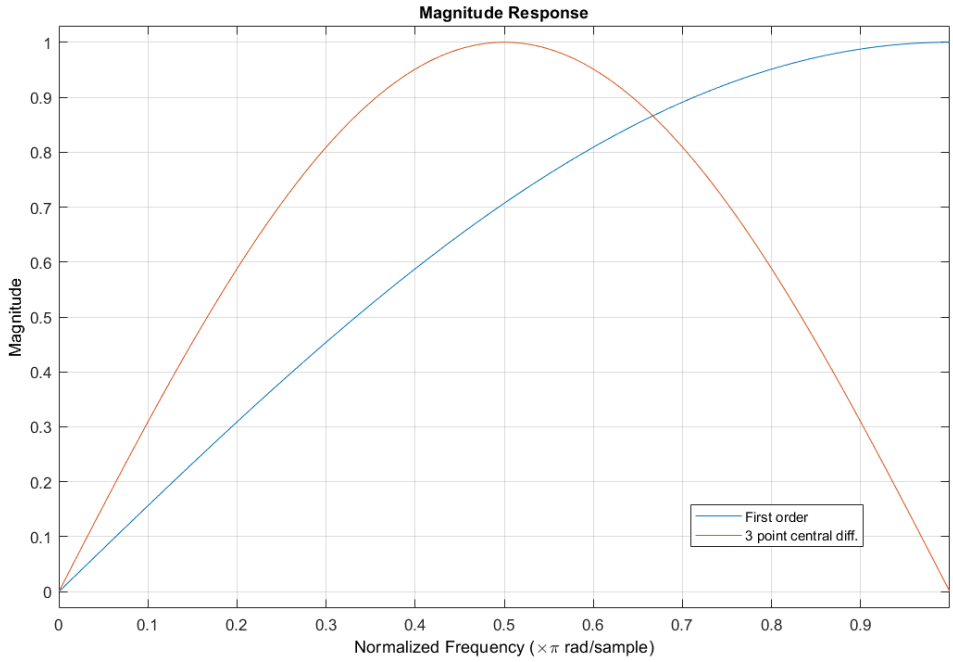

Figure 3.5 1st order and 3 point central diff. magnitude responses(linear) scaled

3.2 FIR derivative filter application

We added EMG and AWGN noises to the given ECG signal and the derived nECG which is shown in figure 3.6.

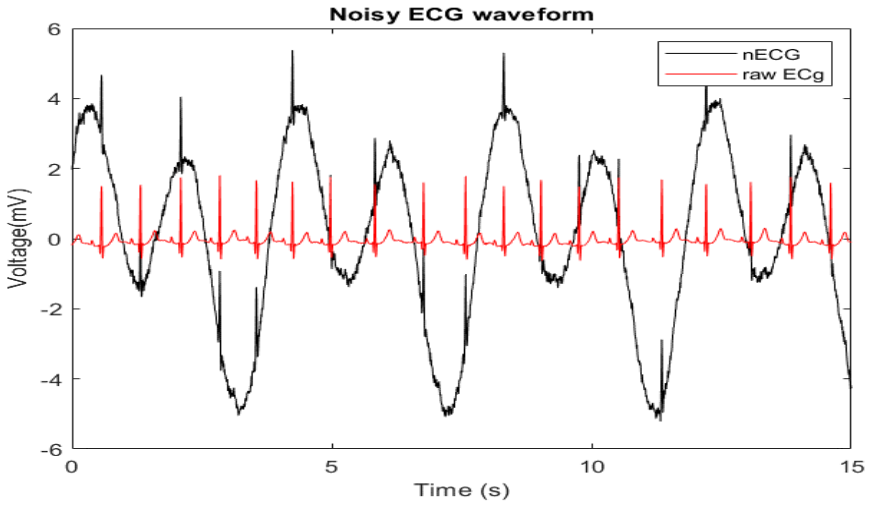

Figure 3.6 Noisy ECG raw ECG signal (zoomed first 15 seconds)

We applied first order and 3-point central diff. derivative filters on this noisy signal and we obtained the following graph.

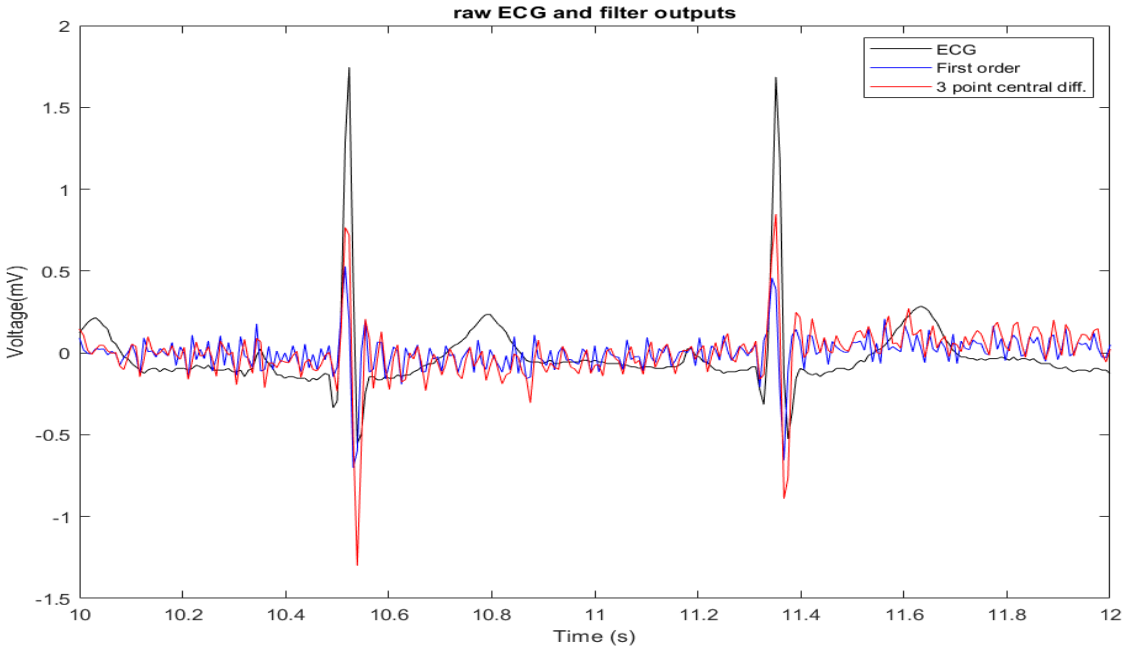

Figure 3.7 Outputs of the filters and raw ECG(zoomed 10 -12)

By observing the above figure(3.7) we can clarify that both filters removed the low noise(removed EMG and low noisy components of AWGN) components in nECG. But still, there are high-frequency noise components in both filtered signals. Also, we can see that the first-order filter has more smooth R complex than 3 points central difference filter so the R peak is lower in the first-order filter. Not only that first order filter attenuates a more noisy signal compared to the 3-point central difference derivative filter.

4 Designing FIR filters using windows

4.1 Characteristics of window functions (use the tool)

The rectangular window method is the simplest windowing method. It only extracts the window length amount of pulses from the original ideal filter impulse response.

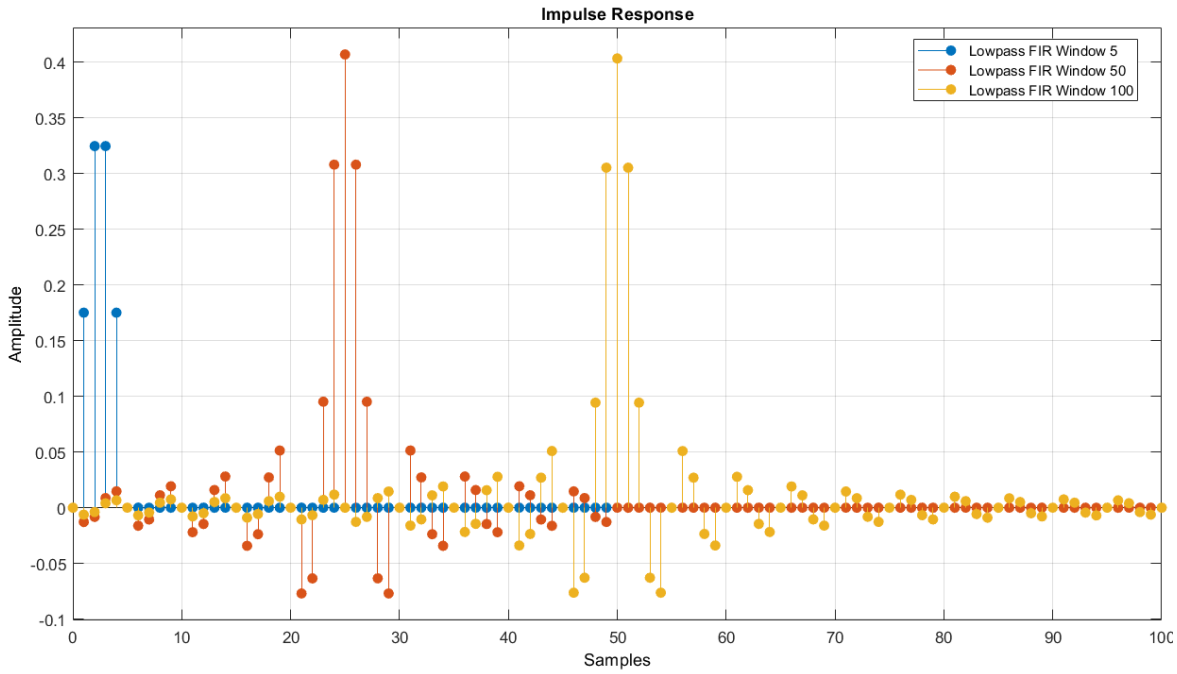

Figure 4.1impulse responses for M=5,50,100 rect. window lengths

According to the above figure 4. 1 we can see that when the window length is increasing the number of pulses extracted from the impulses response of the ideal filter is increased. Therefore when the length of the window function(M) increases the filter approximates to ideal filter behavior.

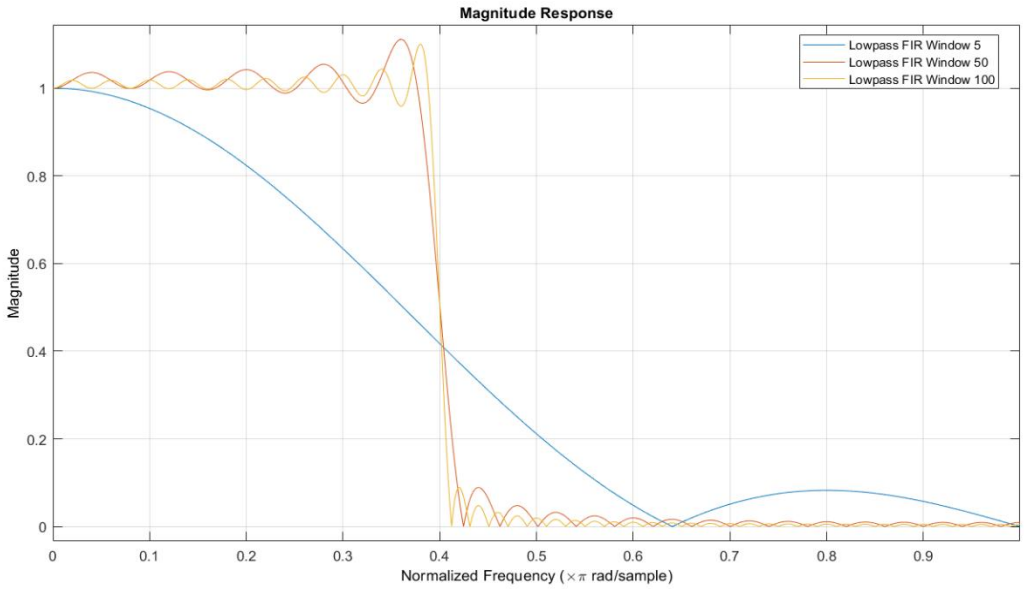

Figure 4.2 magnitude responses rect. windows(5,50,100)

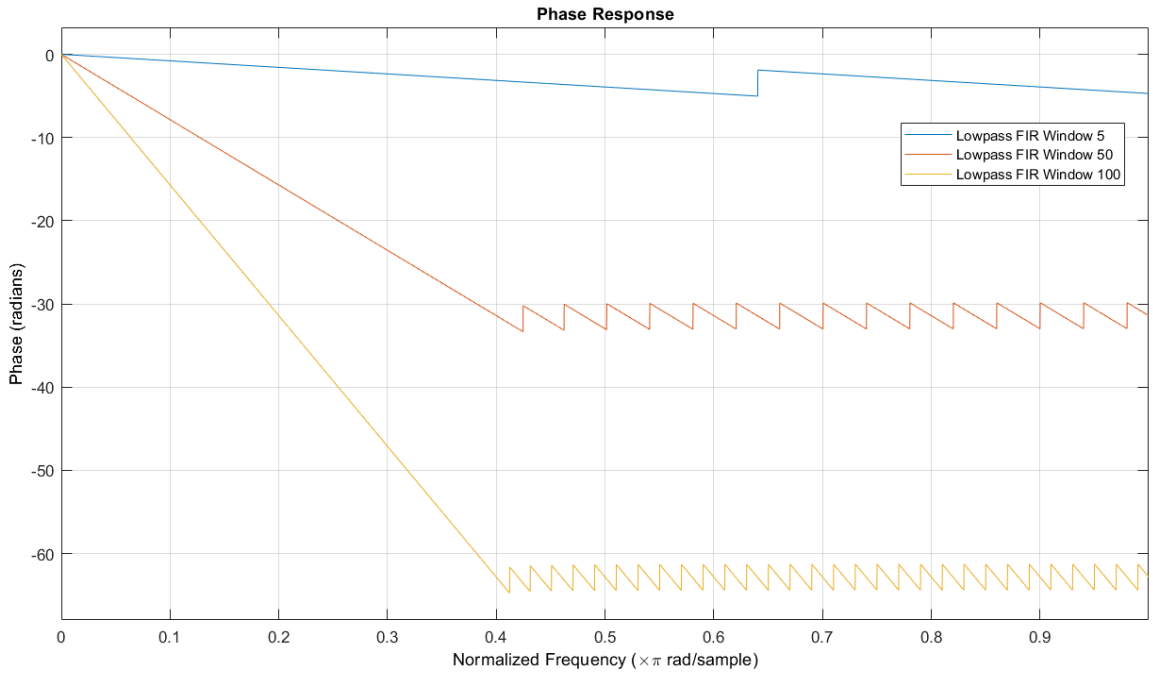

Figure 4.3 phase responses rect. windows(5,50,100)

By looking above figures we can observe that when we increase the window length, the width of the transition period reduces so the filter has a very sharp cutoff. Also, there is a very higher attenuation that occurs when we increase the window length. The discontinuities at the edges of the rectangular window cause large ripples in the pass band. But increasing the window length reduces the amplitude of most of those ripples.

To improve the magnitude response, smoother windows have been designed to remove discontinuities of the rectangular window.

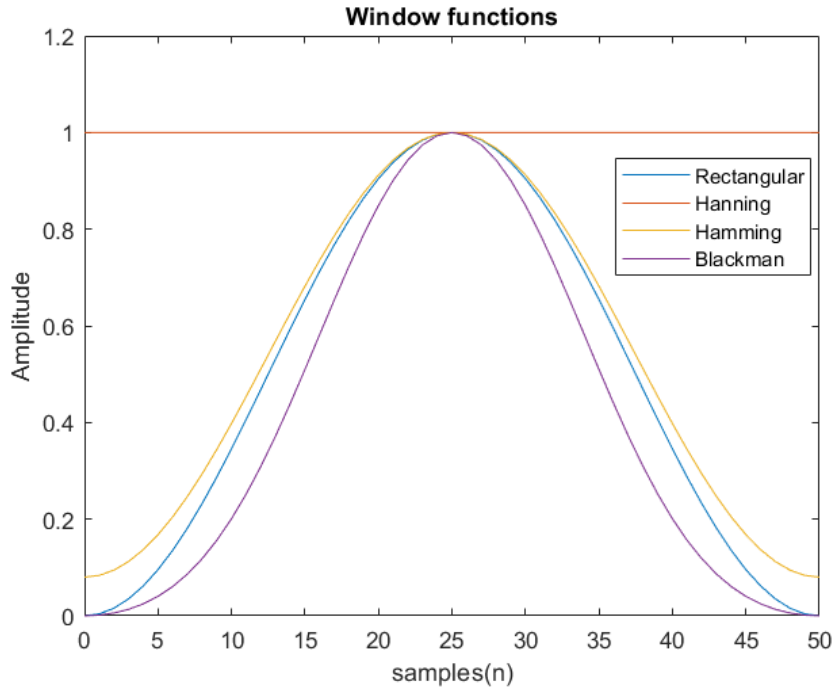

Figure 4.4 Morphology of different windows

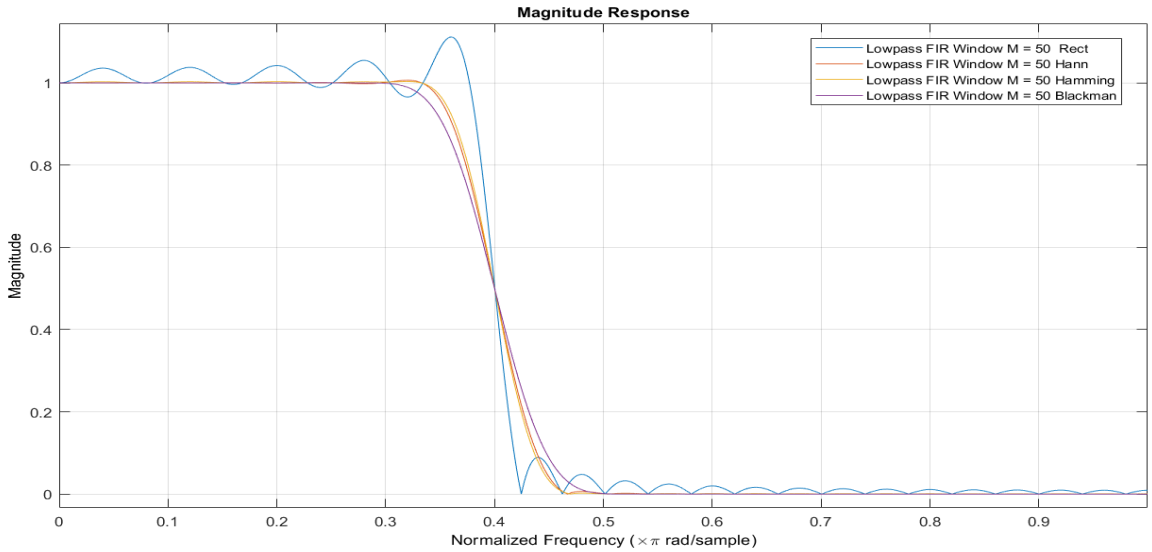

Figure 4.5 Magnitude response with a linear magnitude scale of different filters

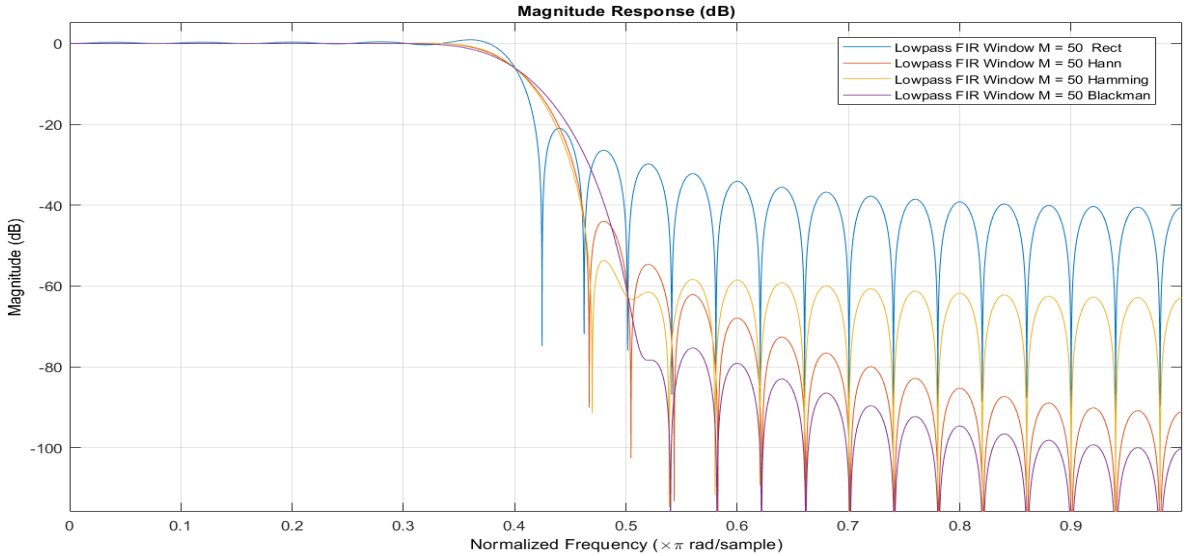

Figure 4.6 Magnitude response with a logarithmic magnitude scale for different filters

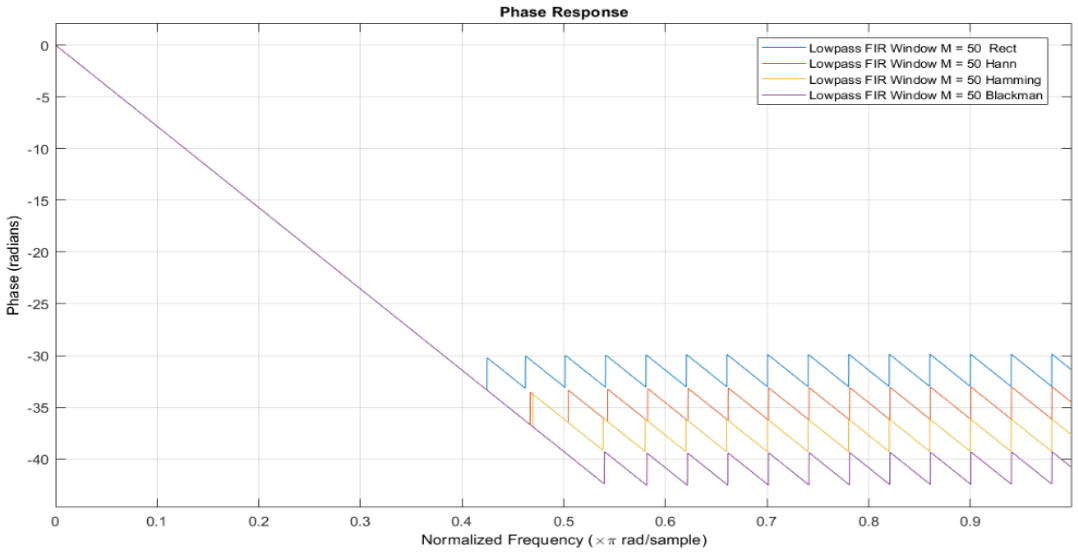

Figure 4.7 Phase responses of different filters

From figure 4.4 we can observe that the rectangular window is the only function that has constant height values through the window length. That is why it is named a rectangular window. But the other windows mentioned in the figure are bell-shaped curves and they are symmetric windows. This is the main reason for higher ripples of the rect. window-based filters and others have lower ripples. All the filters have the same phase change for the pass band since we choose the same size for all window lengths.

4.2 FIR Filter design and application using the Kaiser window

In most of the windows we met before, there is a trade-off between ripple amplitudes in the pass band and the stop band and transition width around the cut-off frequency for a certain window length. The Kaiser window is a generic window function that can approximate a variety of windows by varying the shaping parameter **β** and the window length M. It provides more flexibility to the user to generate customized windows according to the requirements of ripple amplitudes as well as transition widths. In this section, a lowpass and a highpass filter using the Kaiser window and an FIR comb filter we designed.

Let's see the time domain representation and frequency domain representation(PSD) of noise ECG embedded in an ECG signal sampled at 500Hz.

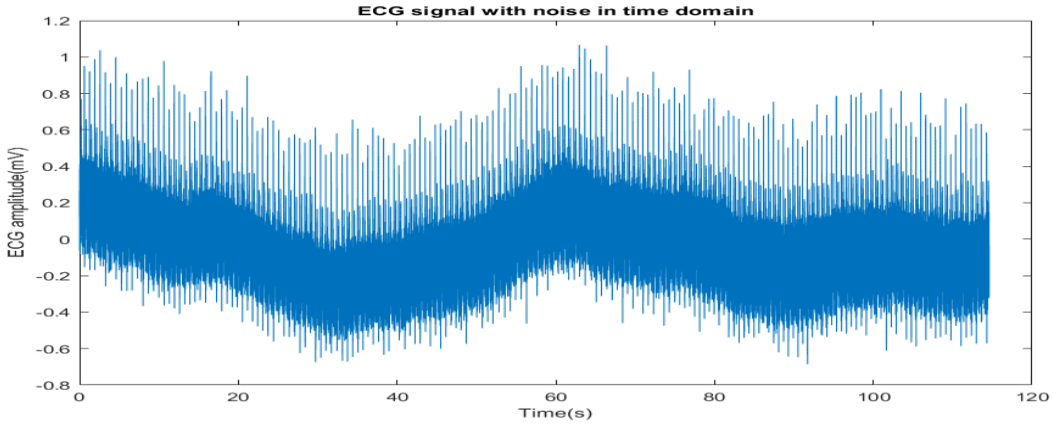

Figure 4.9 noisy ECG

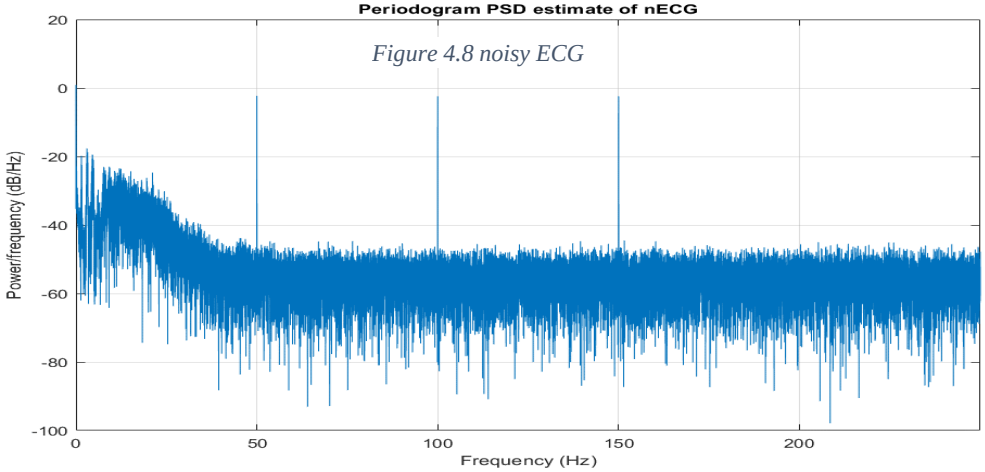

Figure 4.10 PSD of noisy ECG

Figure 4.8 noisy ECG

According to the above figure, we can observe that this ECG signal contains a lot of low and higher noisy components in the signal. This is mostly likely AWGN. However, we can see that there are some peaks in the PSD at 50,100, and 150 Hz. This is mostly due to power line noises.

Now we apply a bandpass filter and comb filter to remove these noises. We decided on parameters for the following filters that need to be applied on the above noisy ECG as follows.

	HPF	LPF
f_{pass}	7 Hz	123 Hz
f_c	5 Hz	125 Hz
f_{stop}	3 Hz	127 Hz
δ	0.001	0.001

Comb filter	
$f_{stop}1h$	50 Hz
$f_{stop}2$	100 Hz
$f_{stop}3$	150Hz

For simplicity, we use the same transition bandwidth and max ripple amplitude for both low-pass and high-pass filters. Now we have to calculate the necessary parameters(M, **β**) for the Kizer window as follows.

$$A = -20\log\delta = -20log(0.001) = 60dB$$

$$\Delta\omega = |f_{pass} - f_{stop}| \times \frac{2\pi}{f_s} = 0.050265$$

$$M = \frac{A - 8}{2.285\Delta\omega} = 452.74 \approx 453$$

$$\beta = \begin{cases} 0.1102(A - 8.7) & 50 < A \\ 0.5842(A - 21)\ 0.4 + 0.07886(A - 21) & 21 \le A \le 50 \\ 0 & A < 21 \end{cases}$$

Therefore $\beta = 5.65326$

This can summarize as follows.

	LPF	HPF
$\lvert f_{pass} - f_{stop}\rvert$	6 Hz	
A	60	
M(should even)	454	
β	5.65326	

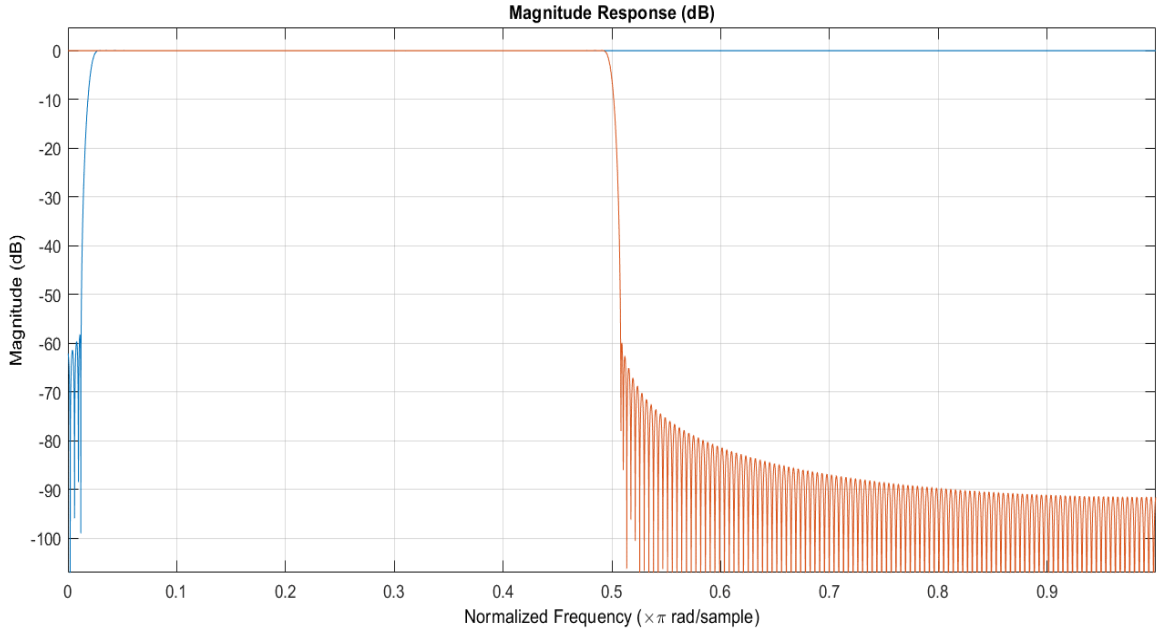

Figure 4.11 Magnitude Response of LP and HP Filter

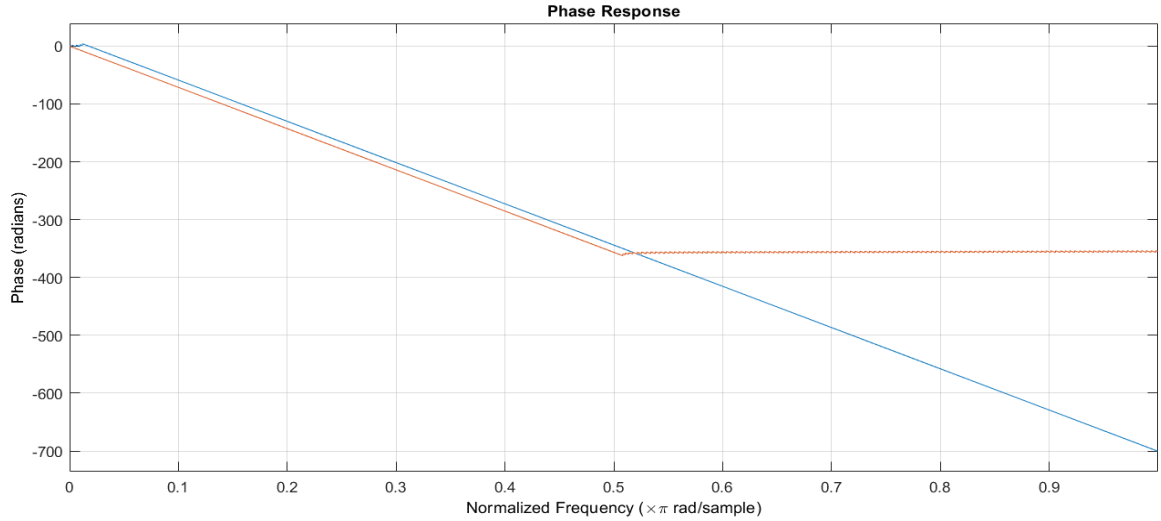

Figure 4.13 Phase Response of LP and HP Filter

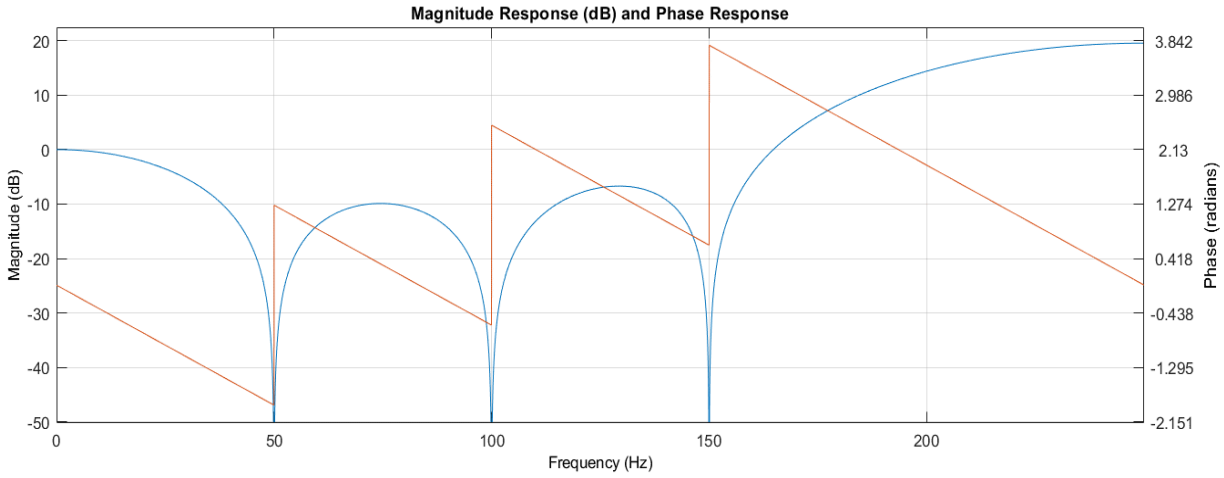

Figure 4.12 magnitude respose and phase response of comb filter

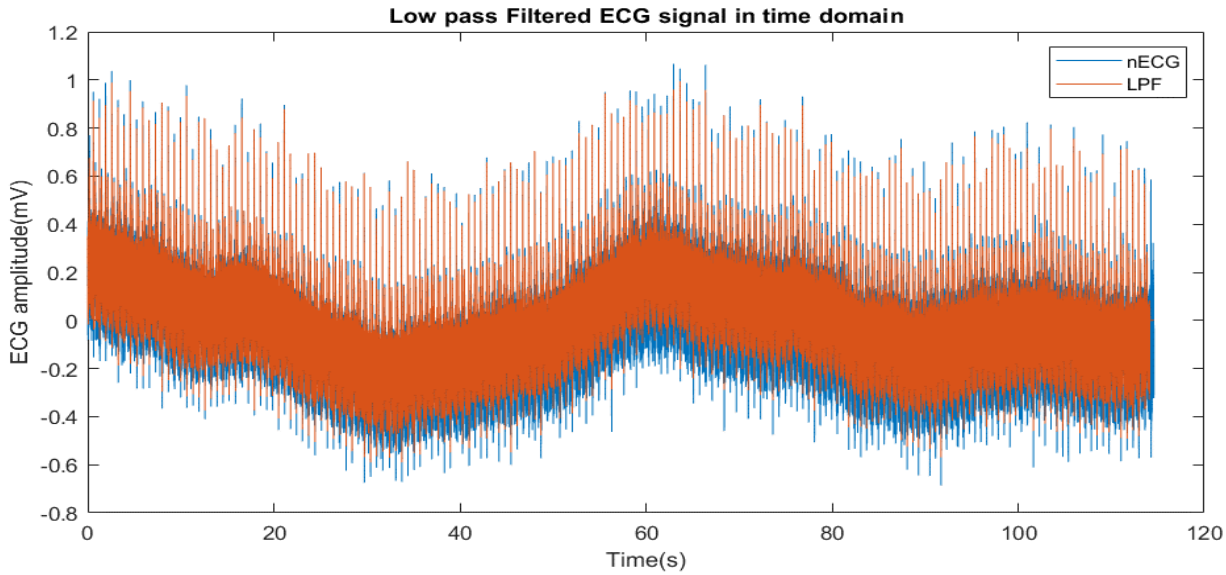

Figure 4.14 LPF response

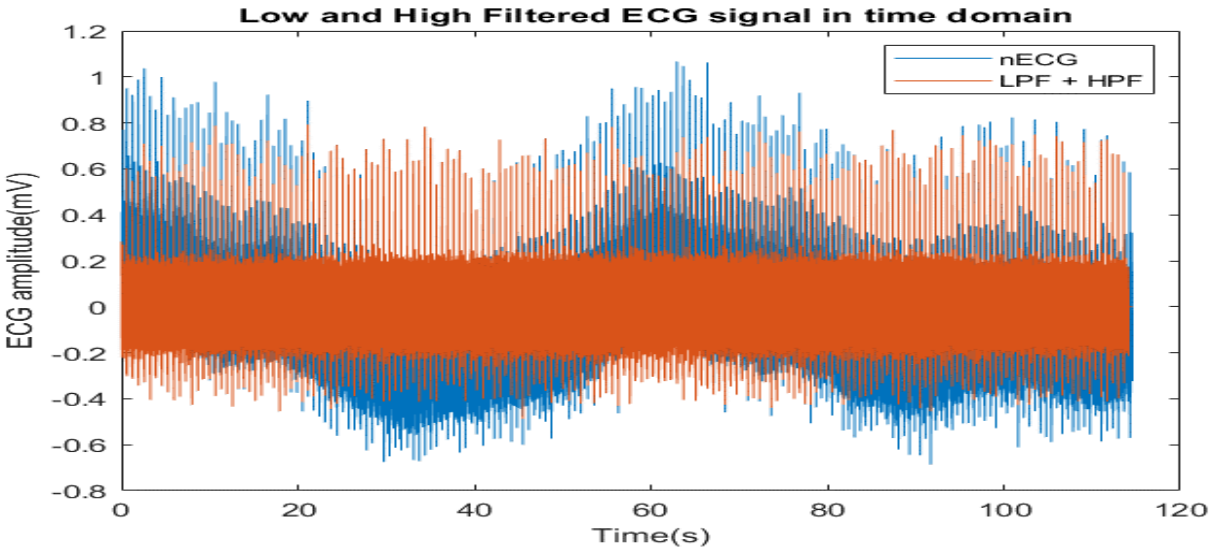

Figure 4.15 LPF + HPF applied

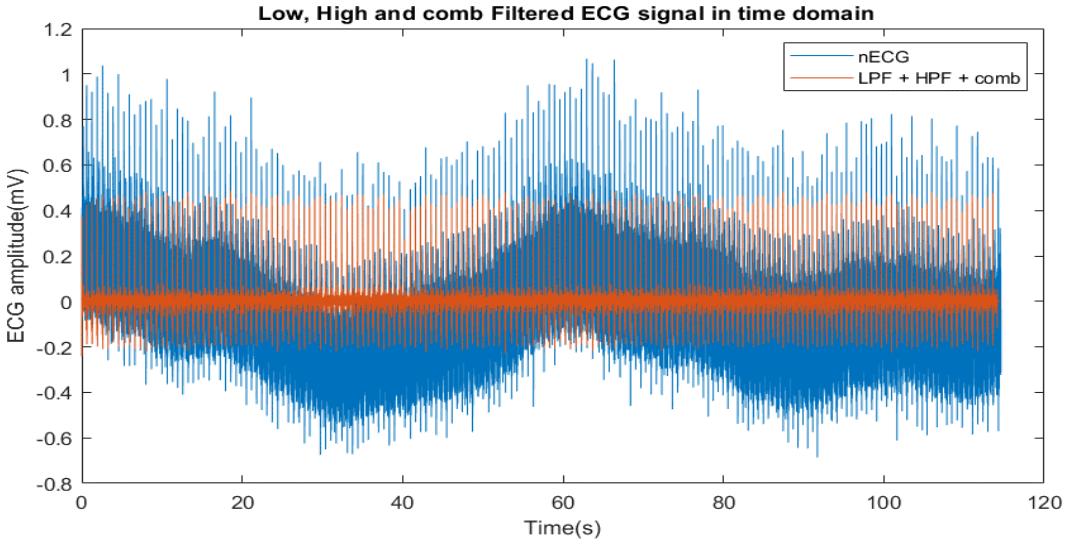

Figure 4.16 LPF + HPF + Comb filterd

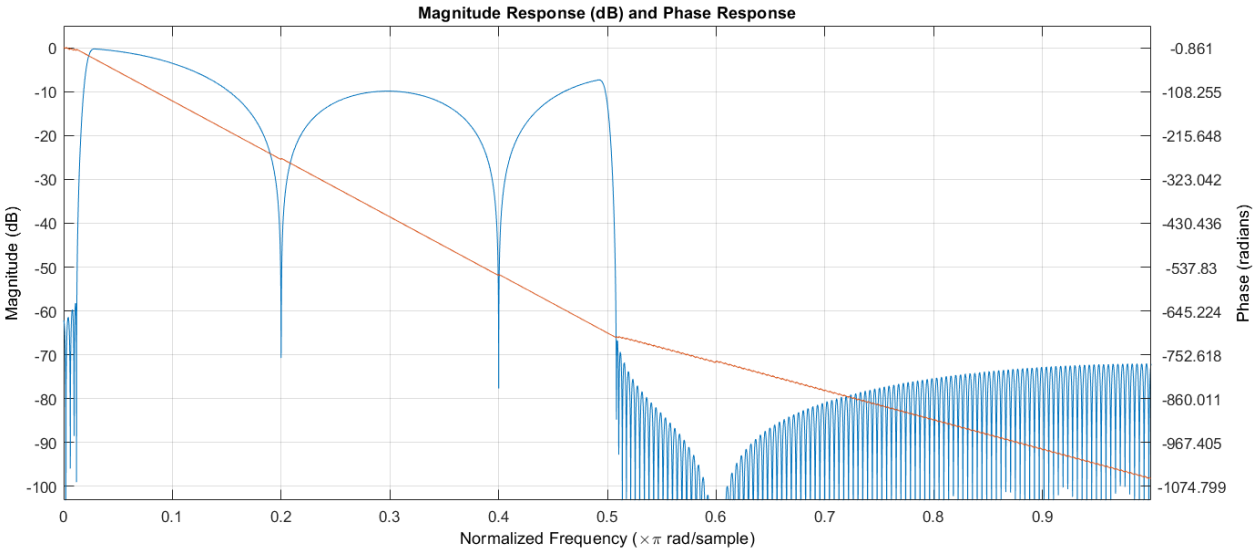

Figure 4.17 Combined filter magnitude response and phase response

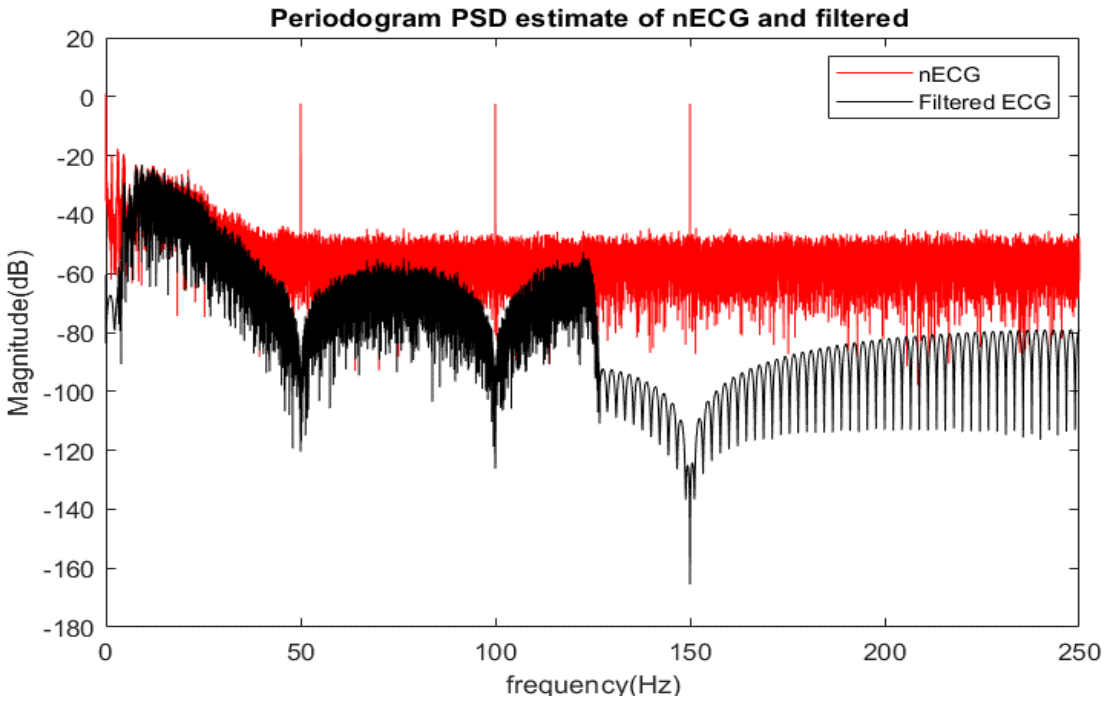

Figure 4.18 PSD of nECG and filtered ECG

5 IIR filters

5.1 Realizing IIR filter

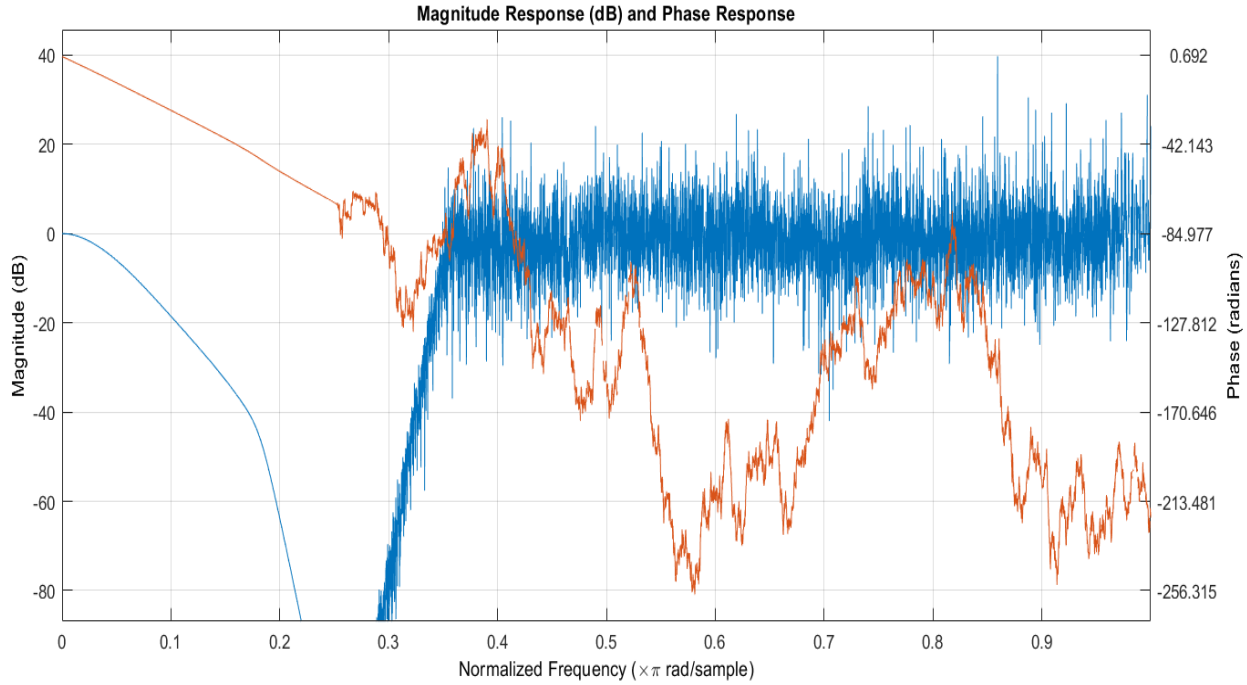

Figure 5.1 Butterworth filter(order =454) LP filter responses

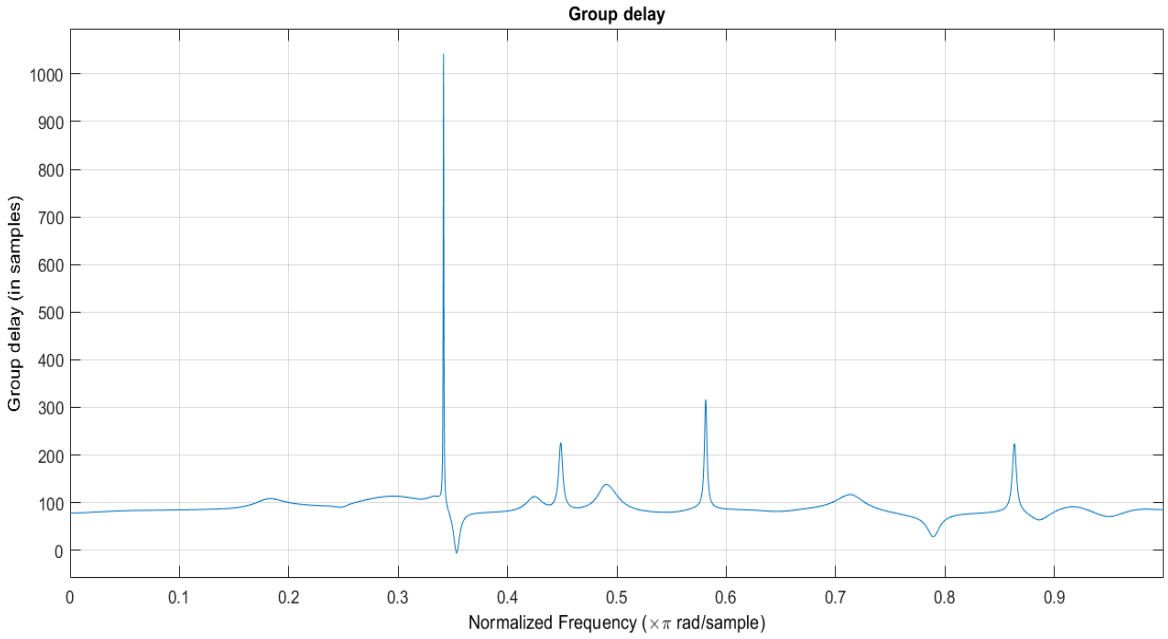

Figure 5.2 Butterworth filter(order =454) LP filter group delay

From the above figure, we can see that the generated Butterworth filter for order(N) = 454 is unstable and have a lot of frequency distortions. Therefore to analyze the characteristic of the IIR filter let's take M = 10 and continue this part.

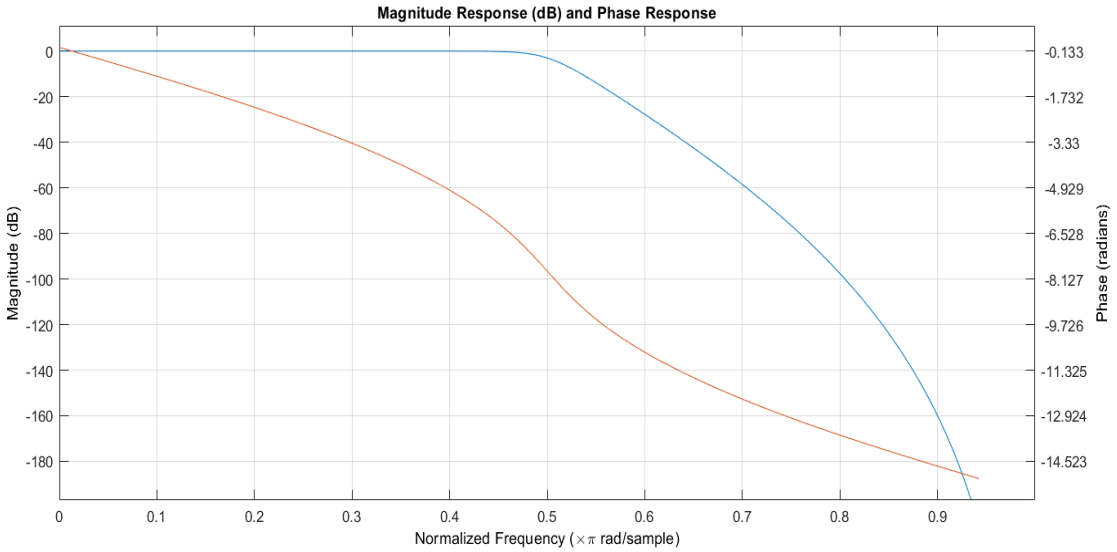

Figure 5.3 Butterworth filter(N =10) LP filter responses

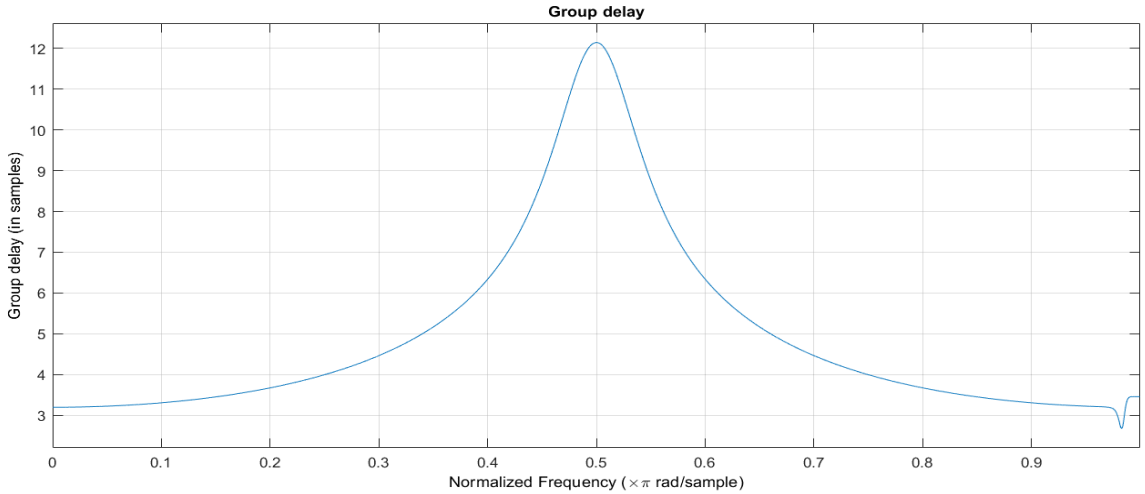

Figure 5.4 Butterworth filter(N =10) LP filter group delay

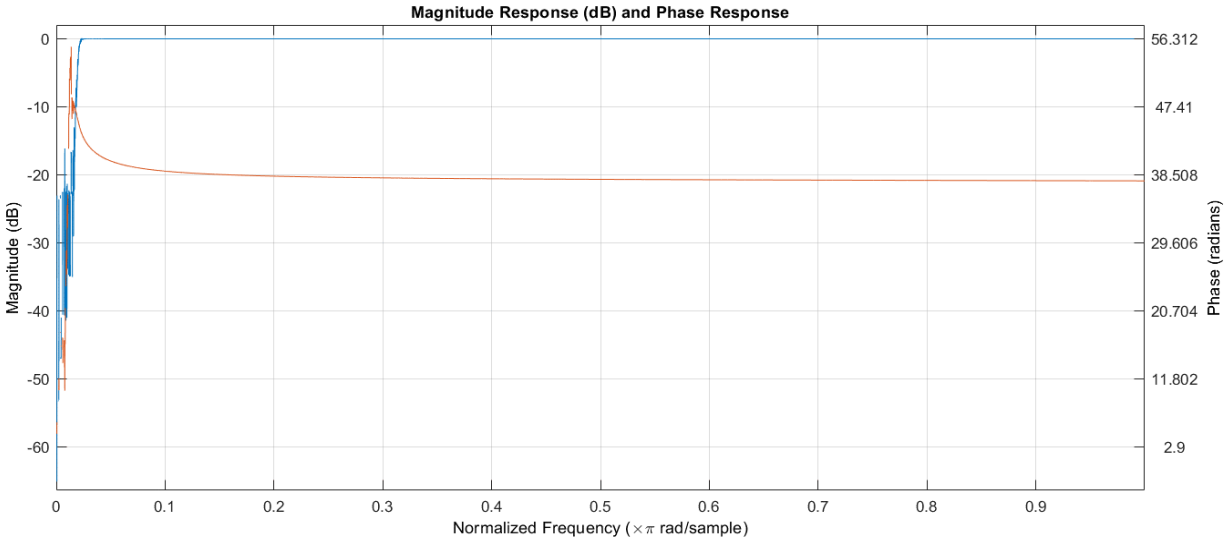

Figure 5.5 Butterworth filter(N =10) HP filter responses

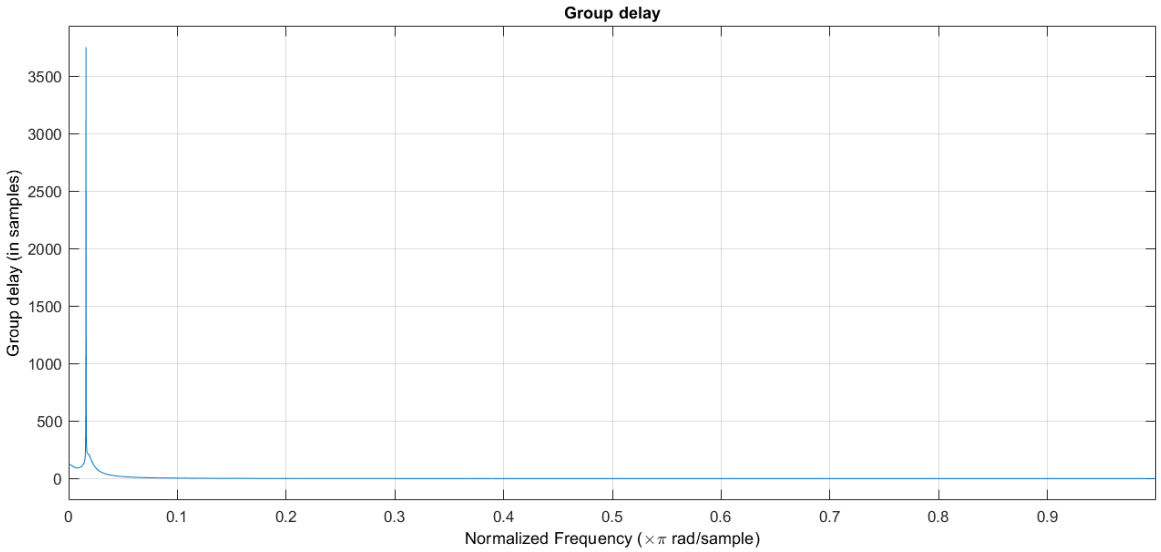

Figure 5.6 Butterworth filter(N =10) HP filter group delay

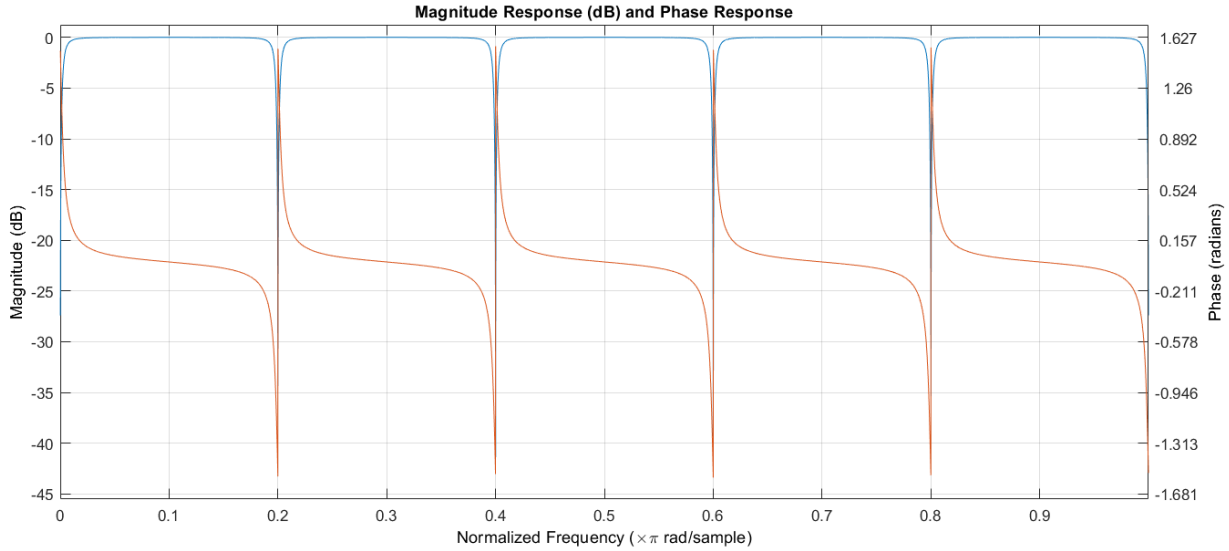

Figure 5.7 IIR Comb filter(N =10) responses

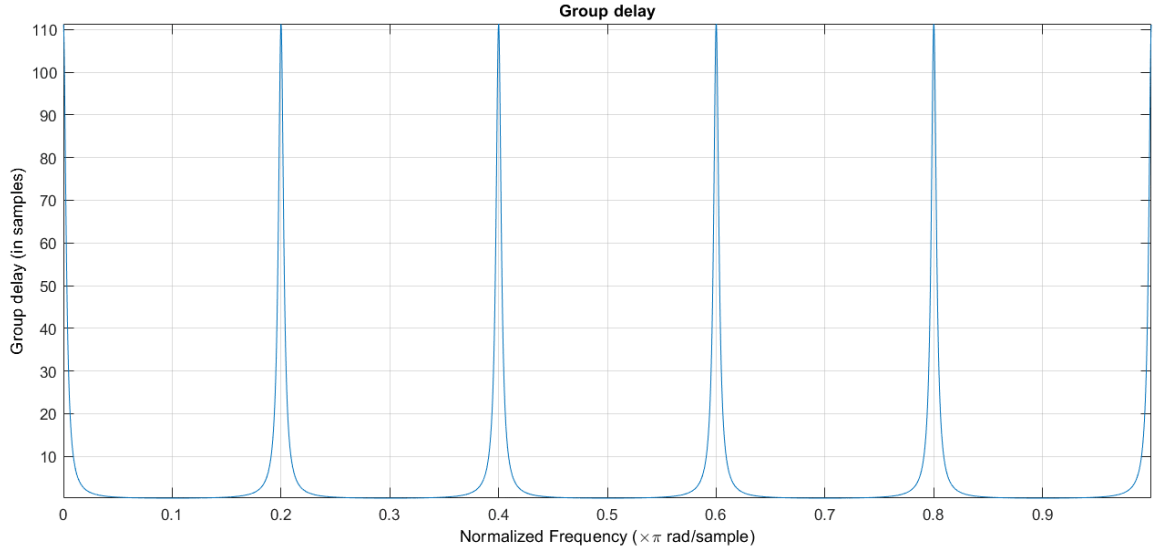

Figure 5.8 IIR Comb filter(N =10) Group delay

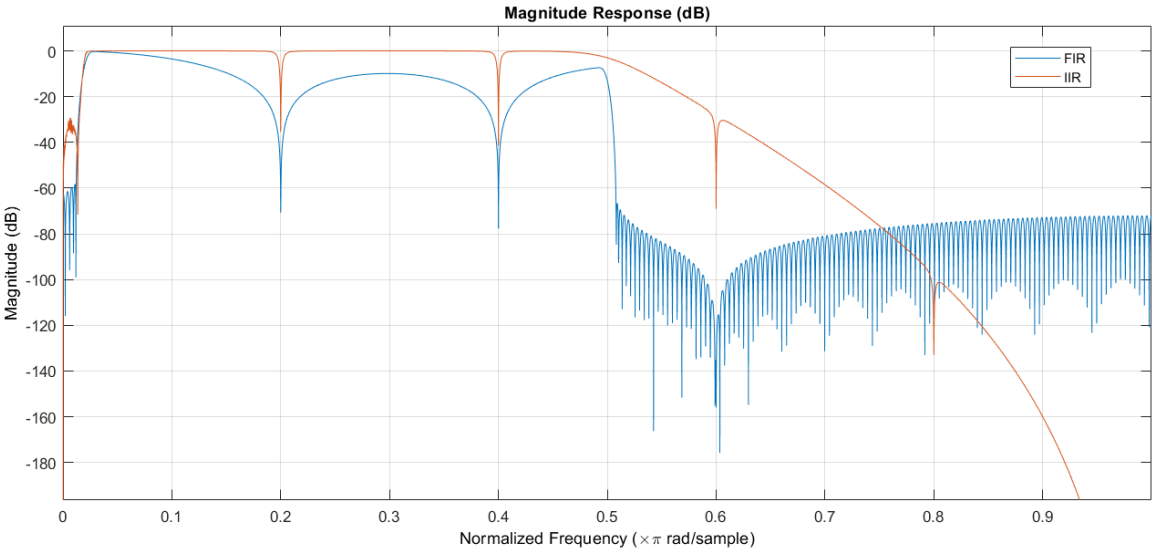

Figure 5.9 FIR and IIR filters magnitude and phase responses

According to above figure 5.9, we can see that the passband of cascaded FIR filters has higher attenuations compared to the cascaded IIR filter. Therefore cascaded IIR filters have a good magnitude response in the pass band. But in stop bands cascaded FIR filters perform well compared to IIR filters. Because attenuation of the stop band is higher in cascaded FIR filters and it has a very narrow transition band compared to the cascaded IIR filters. However compared to the cascaded IIR filter implementation, the Notch(comb) filter combined with cascaded IIR filters has a substantially smaller bandwidth.

5.2 Filtering methods using IIR filters

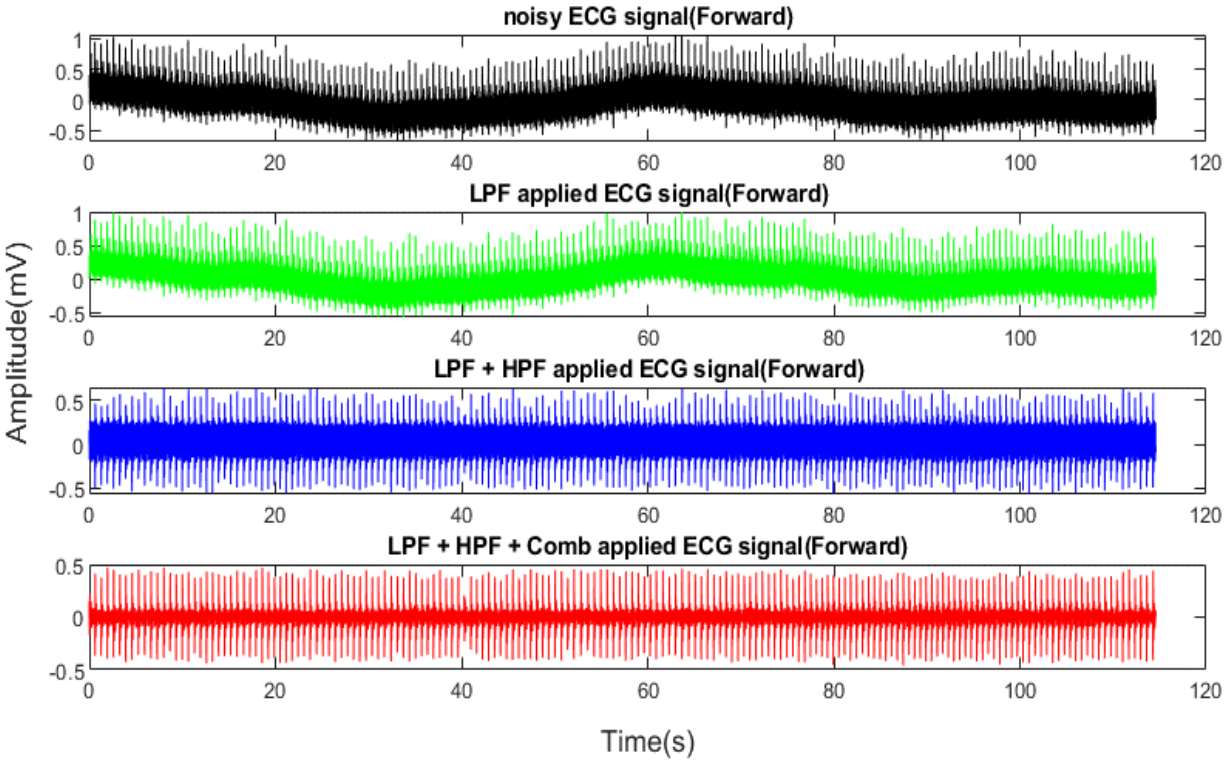

Figure 5.10 10 Forward filtered ECG

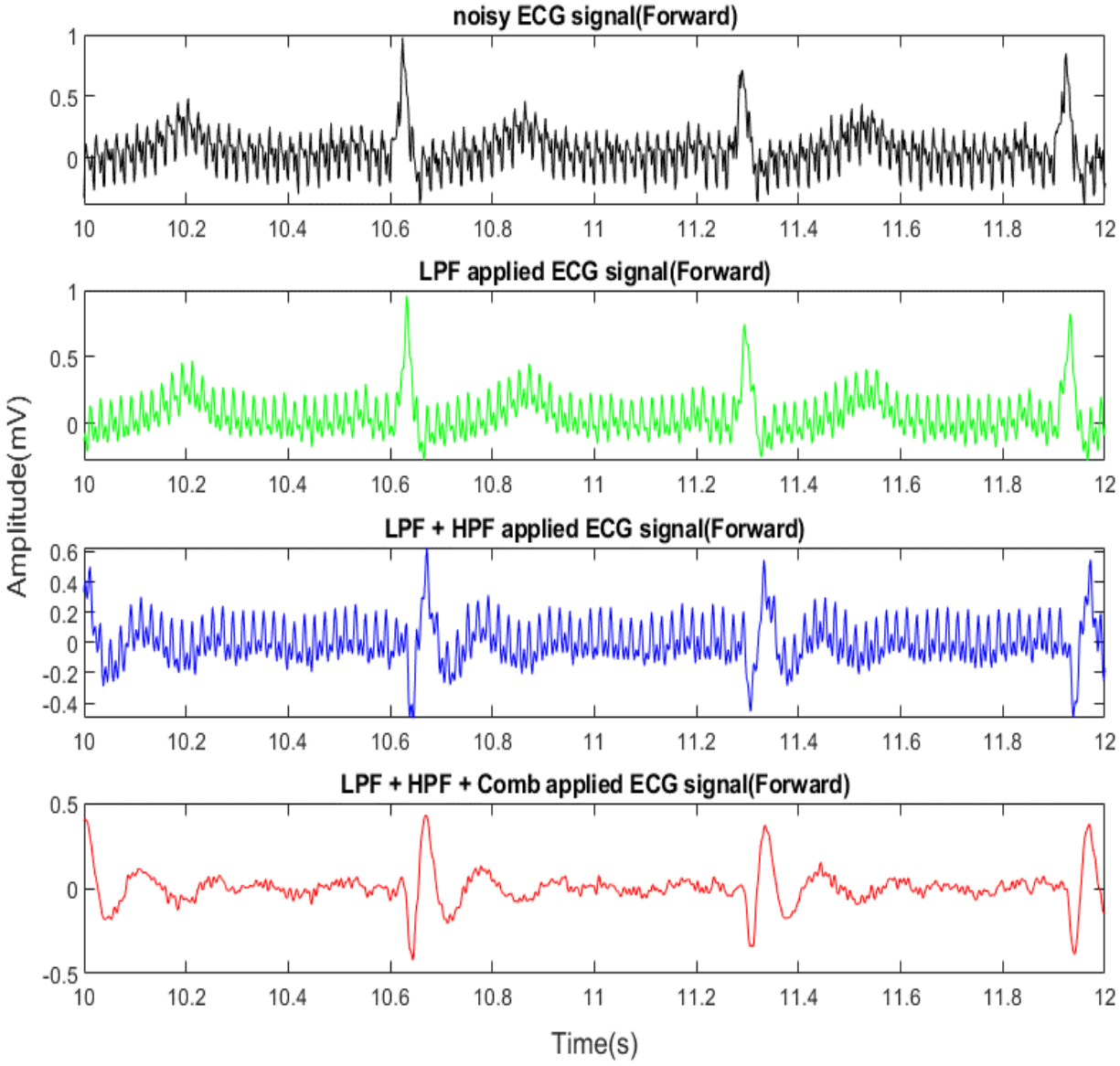

Figure 5.11 10 Forward filtered ECG (Zoomed)

From the above 2 figures, we can see how each filter affects the signal and reduce the noise components. But from the zoomed figure it is noticed that there are some misalignments in the time axis of the final filtered signal compared to the original signal. This is due to the phase distortion of the IIR filters. We can compensate for this by applying a forward-backward filtering method.

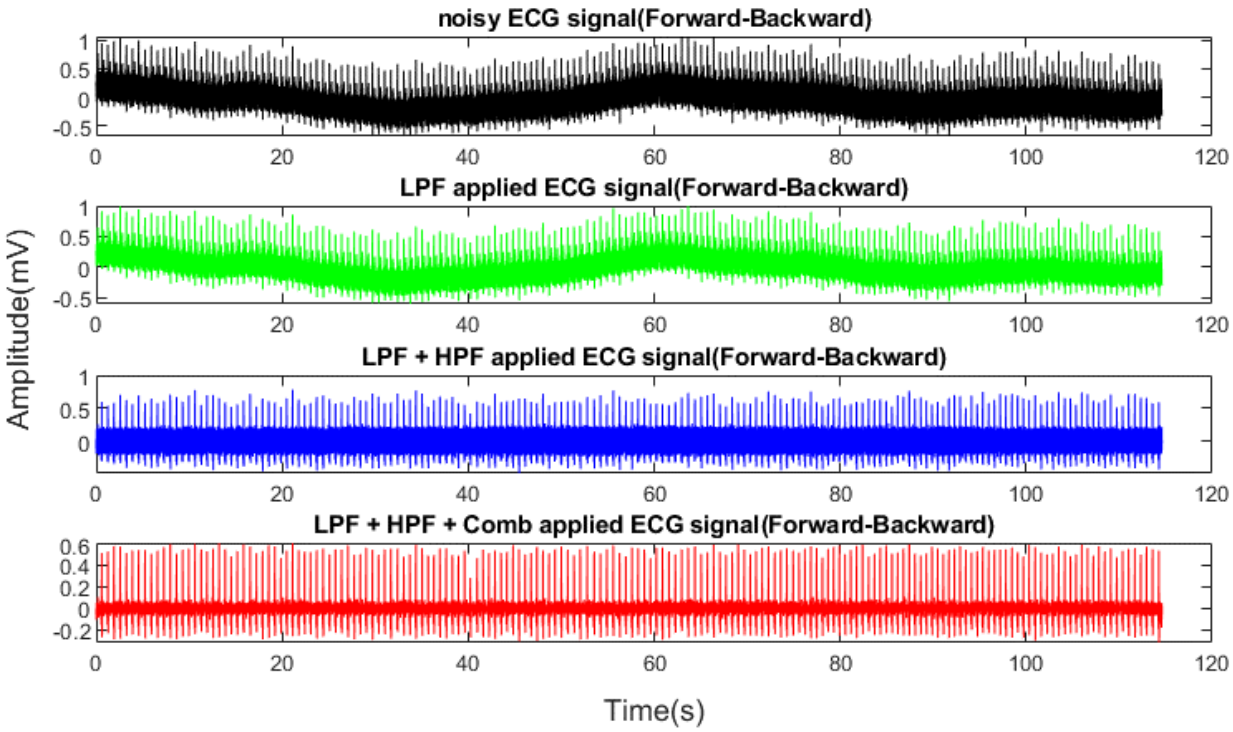

Figure 5.12 Forward and backward filtered ECG

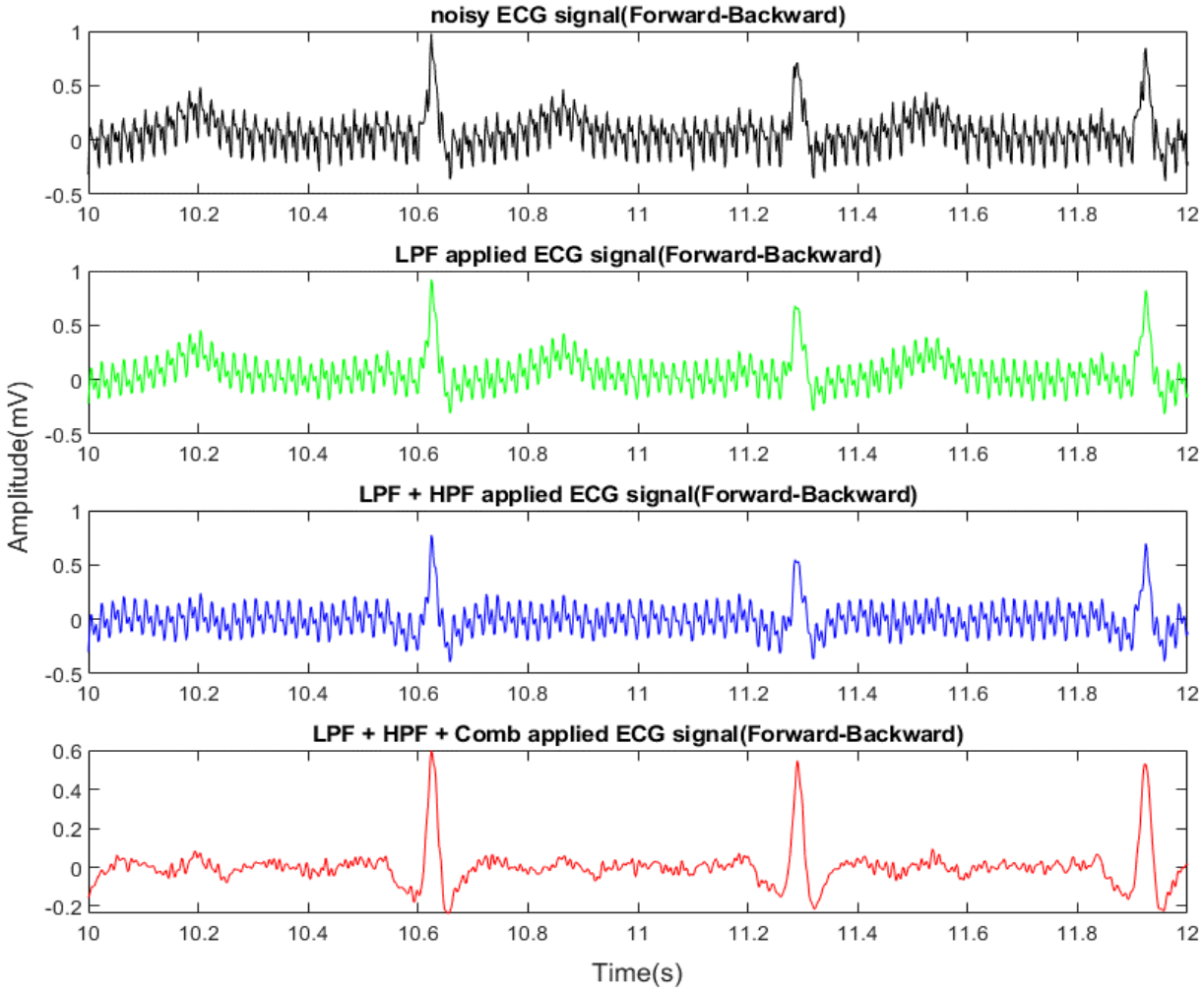

Figure 5.13 Forward and backward filtered ECG (Zoomed)

When applying forward-backward filtering to the given noisy ECG signal, we can see that there are no phase distortions in a filtered signal so that the filtered signal is perfectly aligned with the original noisy signal. So this forward-backward filtering it is possible to obtain zero-phase(0 group delay) IIR filtering.

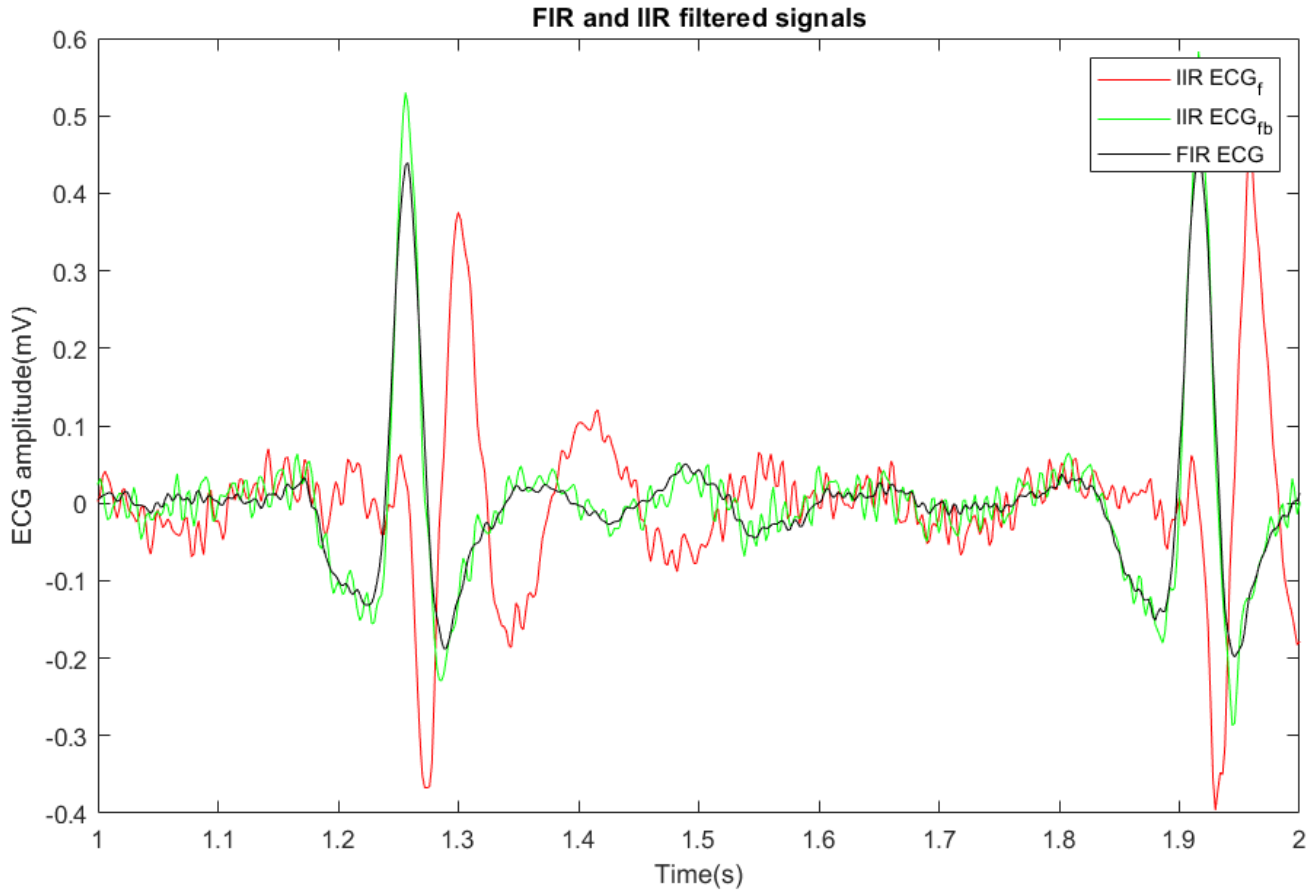

Figure 5.14 FIR and IIR filter outputs

From figure 5.14 we can see that the forward-backward filtered ECG signal is perfectly aligned with the delay-compensated output signal of the FIR filter. There are some high-frequency components included in the IIR filter signal because we chose the order of the IIR filter as 10. So in this case FIR filter attenuates more high-frequency noises compared to the IIR filter. But overall IIR filter performs very well after forward-backward filtering Also we can see that the forward-backward filter has the highest R waveform peaks since it doubles the filter coefficients and squared Magnitudes. It is not fair to compare FIR and IIR filters with different orders. However, in this case, due to the lower order of the IIR filter (10), several drawbacks can be seen. But when we increase the order of the IIR filter we can ensure that it will remove those drawbacks and generate the best output at lower order compared to the FIR filter.

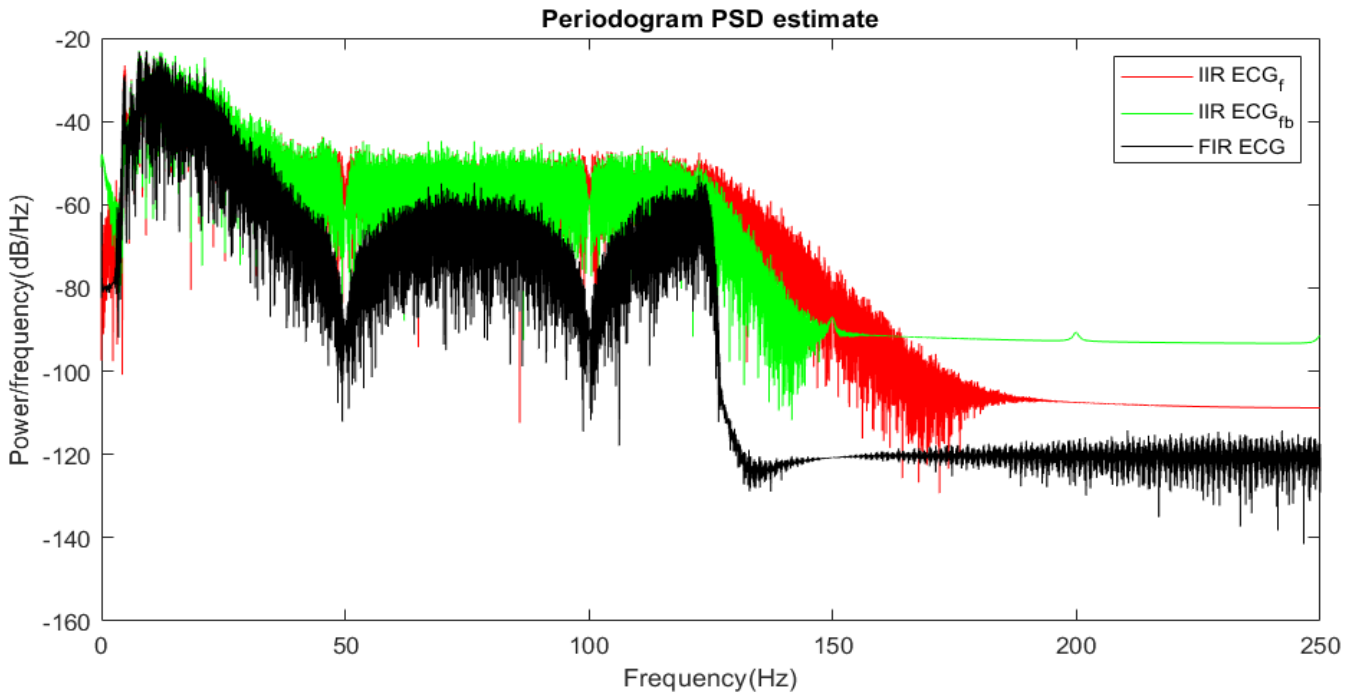

Figure 5.15 PSD's of FIR and IIR filtered signlas

From this figure 5.15, we can see the same interpretation we make from time domain figures. We can see that the FIR filter attenuates more signal power in the pass band compare to the IIR filters. From this figure, we can clarify that FIR filters have a narrow transition band and higher attenuation in the stop band compared to IIR filters. But This statement is not fair to compare FIR and IIR since FIR has a very higher filter order than IIR filter. Also If we try to increase the order of the IIR filter this causes instability.