Department of Electronics and Telecommunication Engineering

University of Moratuwa

BM4151 - Biosignal Processing

MATLAB Assignment 1

Digital Filters

Name	**Index**
H. D. M. Prematilaka	180497C

This report is submitted in partial fulfillment of the requirements for the module BM 4151 – Biosignal Processing.

Date of Submission: November 28, 2022

Table of Contents

List of Figures

List of Tables

1 Smoothing Filters

1.1 Moving Average (MA) Filter

According to the definition, an equally weighted MA filter of order M is given by,

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k] \tag{1}$$

In the Laplace domain, the transfer function could be expressed as,

$$H[z] = \frac{Y[z]}{X[z]} = \frac{1}{M} \sum_{k=0}^{M-1} z^{-k} \tag{2}$$

1.1.1 Preliminaries

1.1.1.1 Loading the ECG Template

The following parameters were used during the acquisition of the given ECG template.

Sampling Frequency	500Hz
Amplitude Range	mV

Table 1: Acquisition Parameters of the ECG Template

'Load' command of MATLAB was used for this task and the loaded ECG template was assigned to a dummy variable.

1.1.1.2 Plotting the Loaded Signal

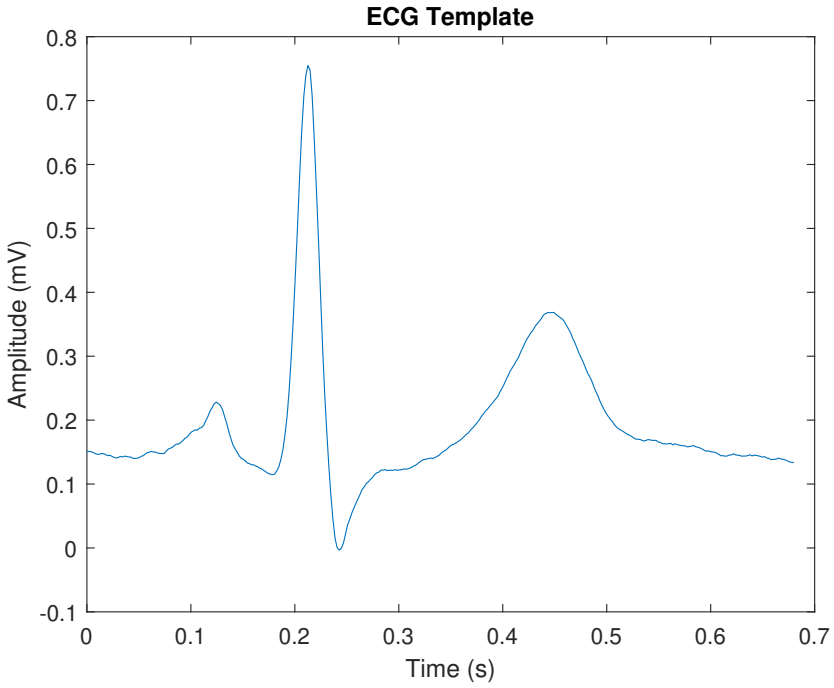

Figure 1: ECG Template

After close observation, we could roughly identify the starting and ending points of different segments and intervals as well as the P wave and T wave in the ECG wavelet.

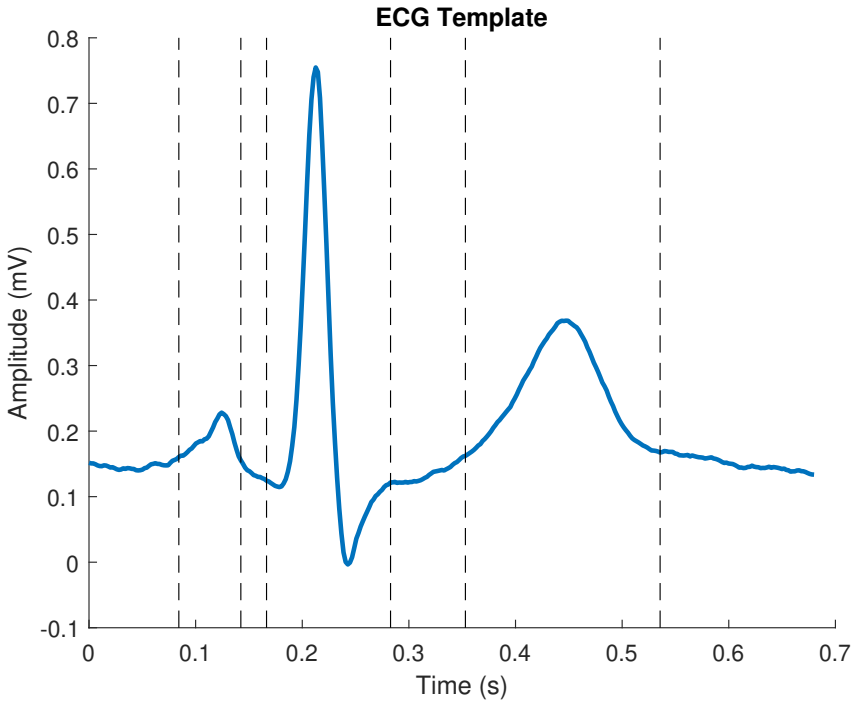

Figure 2: ECG Template with Features

The approximated time stamps are,

- PR Interval: 0.08425 - 0.1665 (s)
- PR Segment: 0.1424 - 0.1665 (s)
- QRS Complex: 0.1665 - 0.2828 (s)
- QT Interval: 0.1665 - 0.5356 (s)
- ST Segment: 0.2828 - 0.353 (s)

1.1.1.3 Adding White Gaussian Noise of 5dB

Now, we will observe the changes in the original ECG signal after the introduction of a 5dB white noise.

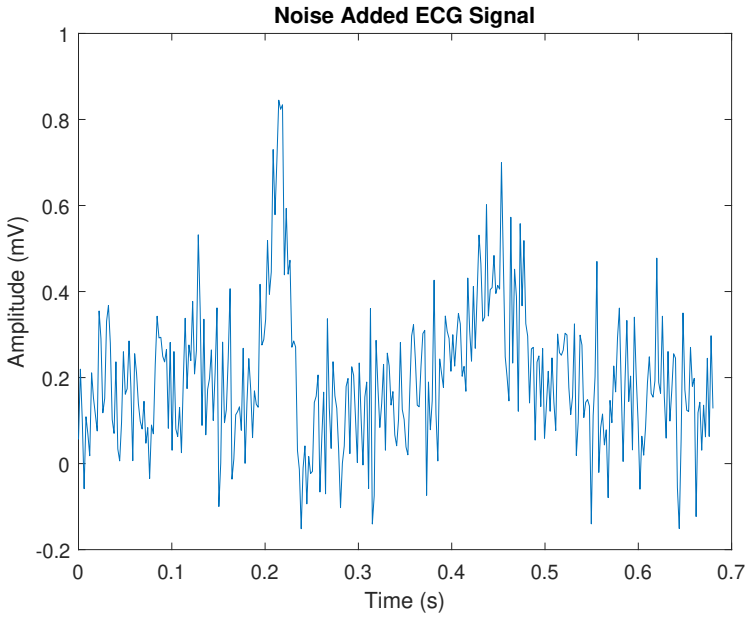

Figure 3: ECG Signal after a Noise Addition of 5dB

To get a better understanding how noise has corrupted the signal, we will overlay the original signal on top of the noisy signal.

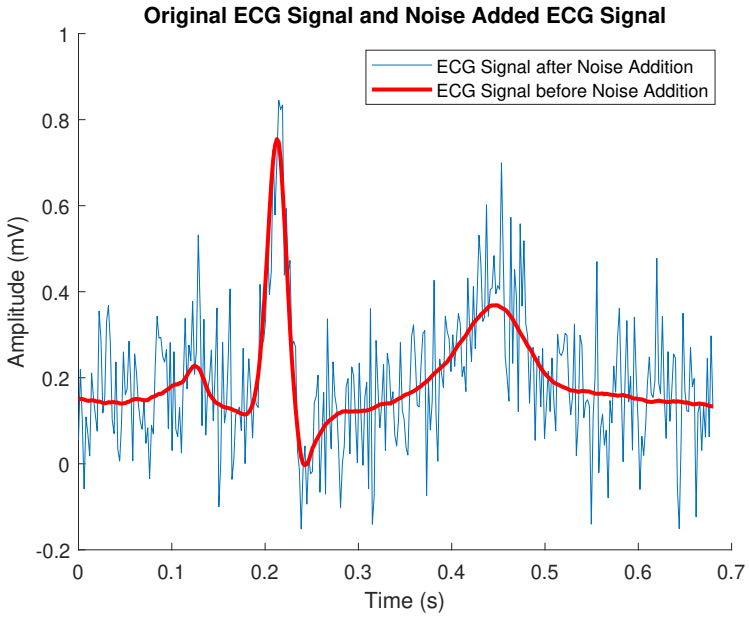

Figure 4: Comparison between the Initial and Noisy ECG Signals

1.1.1.4 Plotting the Power Spectral Density

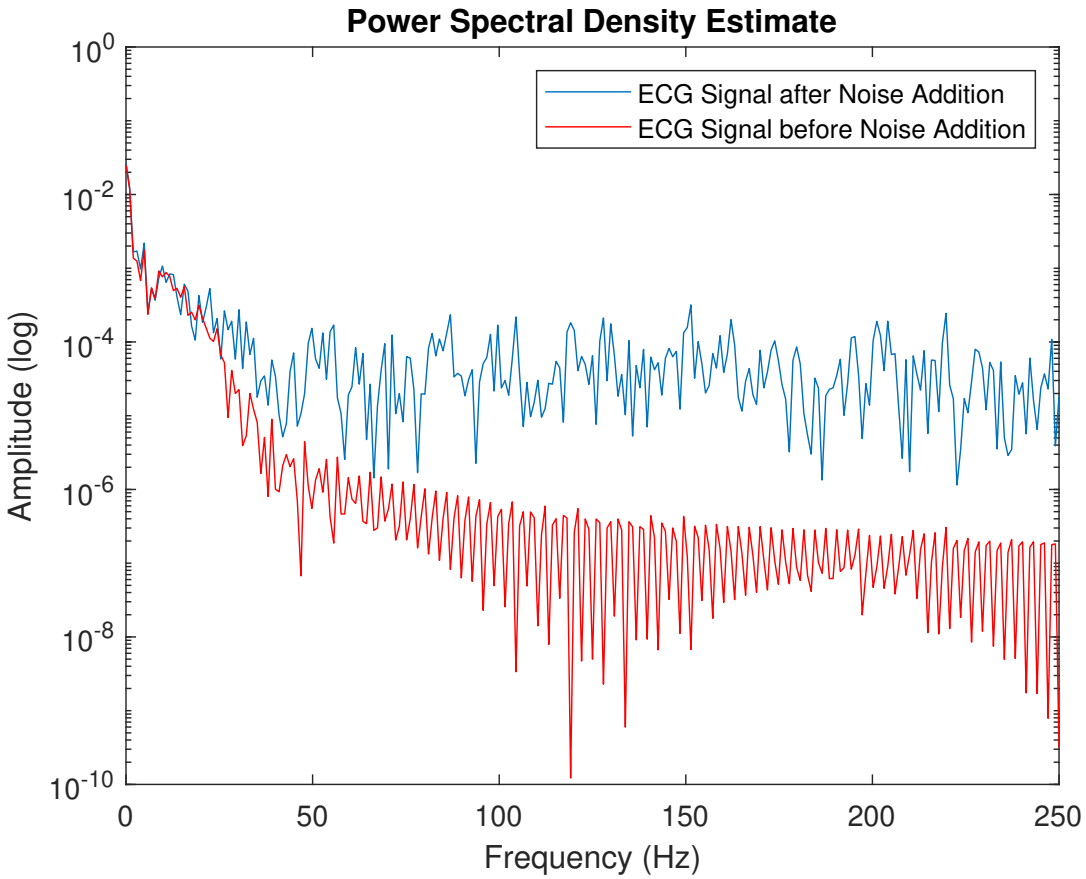

Figure 5: Comparison between PSDs of the Initial and Noisy ECG Signals

Considering Fig. 5, in the initial ECG template, the power distribution is skewed towards the lower frequency components. This is true to a certain extent for the noise-added ECG signal as well. However, the energy distribution of the noise-added ECG signal in higher frequency components is much higher than that of the original ECG signal. That being said, we may incorporate a low pass filter to suppress these higher-frequency noise components.

1.1.2 MA(3) Filter Implementation with a Custom Script

1.1.2.1 Writing a Script for the MA(3) Filter

The code for the custom script is found in the customMovingAvg.m file. The first $(order - 1)$ terms are not averaged and are taken directly from the input signal. The moving average is implemented starting from the $(order)^{th}$ term.

1.1.2.2 Deriving the Group Delay

In the Laplace domain, the transfer function is given by Eq. 2. To obtain the group delay, we will consider the Fourier domain and the conversion $z = e^{j\omega T_s}$ will be used.

$$
\begin{aligned}
H(\omega) &= \frac{1}{N}\sum_{k=0}^{N-1} e^{-jk\omega T_s} \\
&= \frac{e^{-j\left(\frac{N-1}{2}\right)\omega T_s}}{N} \sum_{k=\frac{-(N-1)}{2}}^{\frac{(N-1)}{2}} e^{-jk\omega T_s} \\
&= \frac{cos\left(\frac{(N-1)\omega T_s}{2}\right) - jsin\left(\frac{(N-1)\omega T_s}{2}\right)}{N} \left[\left(\sum_{k=1}^{\frac{(N-1)}{2}} 2cos(k\omega T_s)\right) + 1\right]
\end{aligned}
$$

Note that, we have used the relation, $e^{j\omega T_s} = cos(\omega T_s) + jsin(\omega T_s)$. Moreover, notice that, $\frac{\left[\left(\sum_{k=1}^{\frac{(N-1)}{2}} 2cos(k\omega T_s)\right)+1\right]}{N}$ is constant (c). Thus,

$$
\begin{aligned}
H(\omega) &= c\left(cos\left(\frac{(N-1)\omega T_s}{2}\right) - jsin\left(\frac{(N-1)\omega T_s}{2}\right)\right) \\
Arg(H(\omega)) &= tan^{-1}\left(\frac{-sin\left(\frac{(N-1)}{2}\right)\omega T_s}{cos\left(\frac{(N-1)}{2}\right)\omega T_s}\right) \\
&= -\frac{N-1}{2}\omega T_s \\
\tau_g &= \frac{-\partial Arg(H(\omega))}{\partial\omega} = \frac{N-1}{2}T_s
\end{aligned} \tag{3}
$$

When $N = 3$, the group delay would be $\tau_g = T_s = 1/500 = 0.002$.

1.1.2.3 nECG Signal vs Delay Compensated ma3ECG_1 Signal

In the previous part, we calculated that the delay was 0.002s. Before comparing the nECG signal and the delay-compensated signal, we will first check the uncompensated and delay-compensated ma3ECG_3 signals.

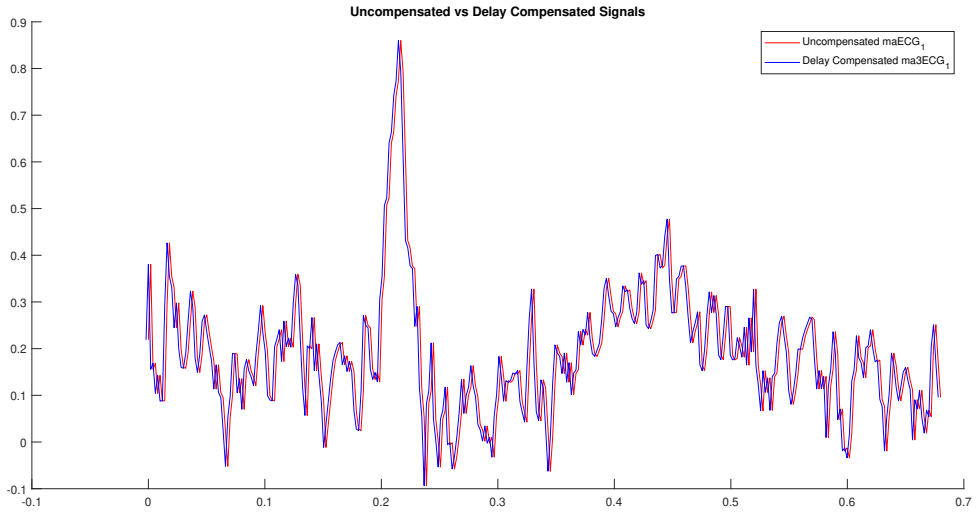

Figure 6: Comparison between Uncompensated and Delay Compensated ma3ECG_1 Signals

For better comparison, let's zoom in on the plot a bit. As we can see, the difference between the shown black dotted lines is 0.002s.

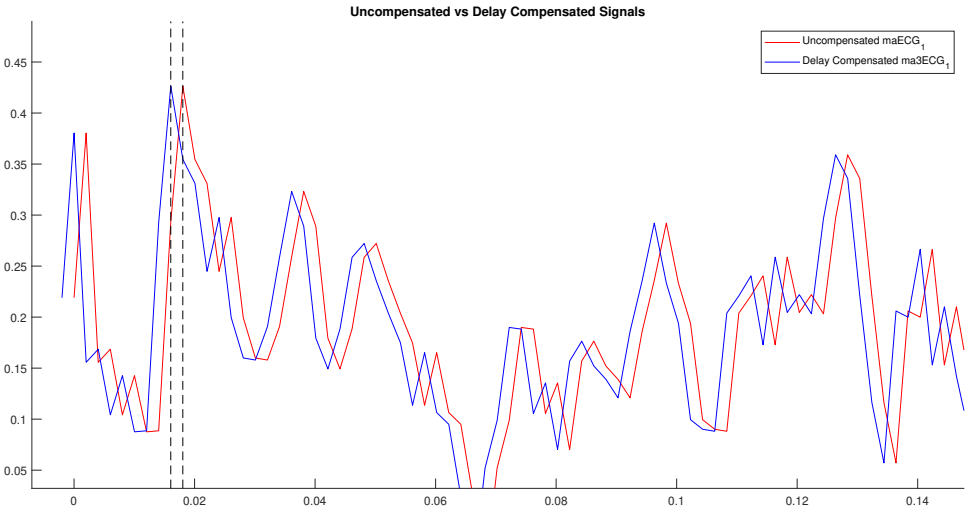

Figure 7: Comparison between Uncompensated and Delay Compensated ma3ECG_1 Signals (Zoomed in)

Now, we will compare the nECG signal and the delay compensated ma3ECG_1 signal. As indicated in the plot below, it is evident that rapid variations in the nECG signal are suppressed to a greater extent in the ma3ECG_1 signal.

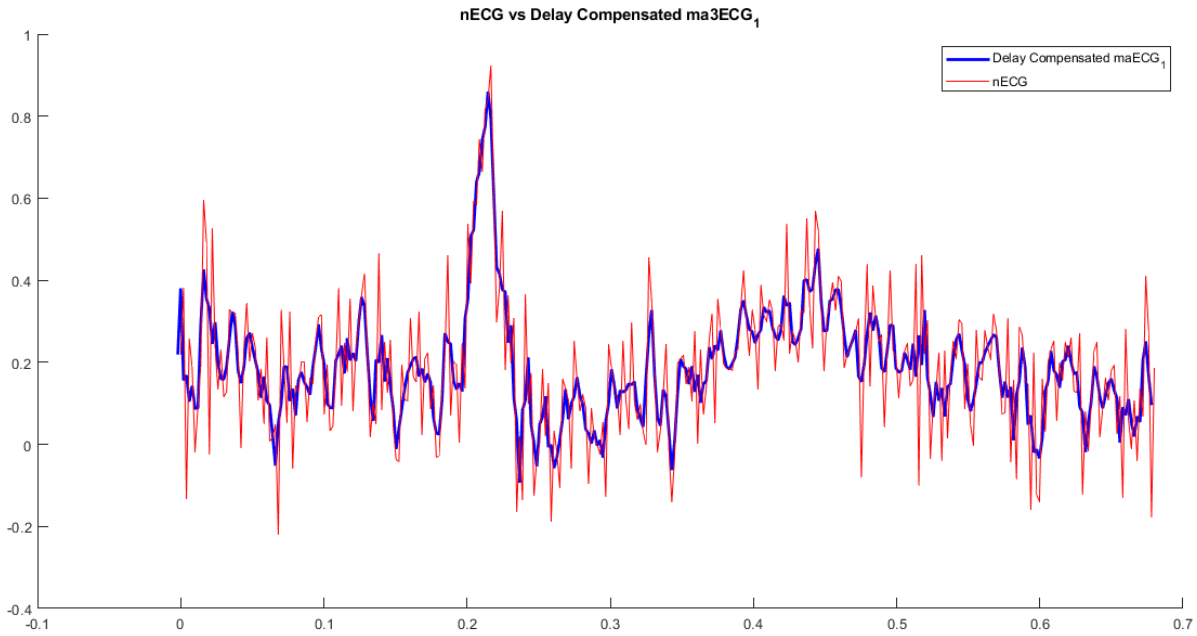

Figure 8: Comparison between nECG and Delay Compensated ma3ECG_1 Signals

1.1.2.4 Power Spectral Densities of nECG Signal and ma3ECG_1 Signal

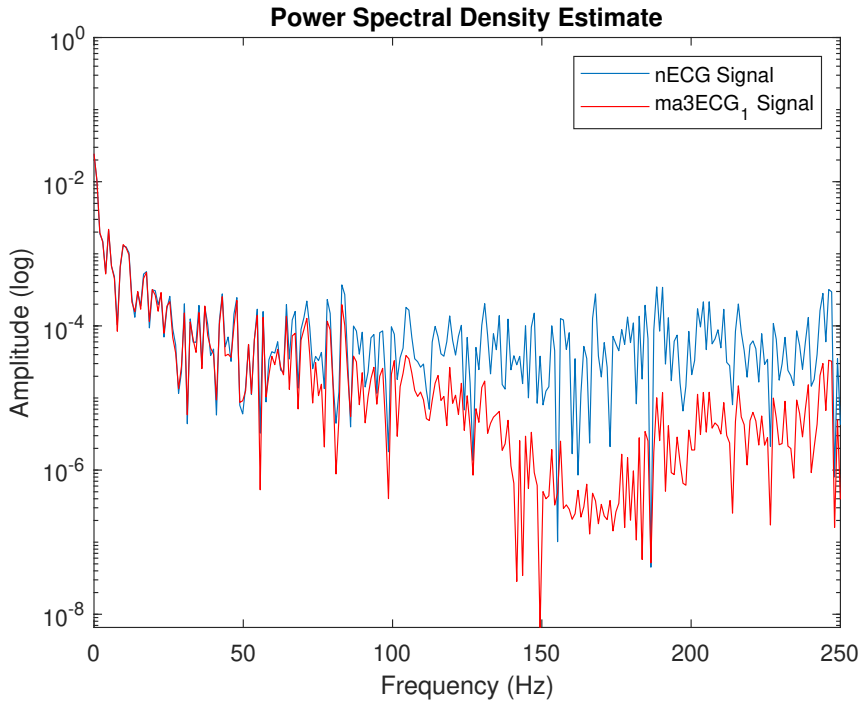

Figure 9: PSD Comparison between nECG and ma3ECG_1 Signals

From Fig. 9, we can clearly see that the magnitudes of the high-frequency components are suppressed to a considerable level.

1.1.3 MA(3) Filter Implementation with MATLAB Built-in Function

1.1.3.1 Using filter(b,a,x) Command

In the filter(b,a,x) command,

- a - Coefficients of the output signal in the difference equation
- b - Coefficients of the input signal in the difference equation
- x - Input signal

Since the order of the MA filter is 3, we can express the output signal as,

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$$
$$y[n] = \frac{1}{3}x[n] + \frac{1}{3}x[n-1] + \frac{1}{3}x[n-2]$$

Thus, $a = 1$ and $b = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$.

1.1.3.2 Plotting nECG, ECG_template and ma3ECG_2

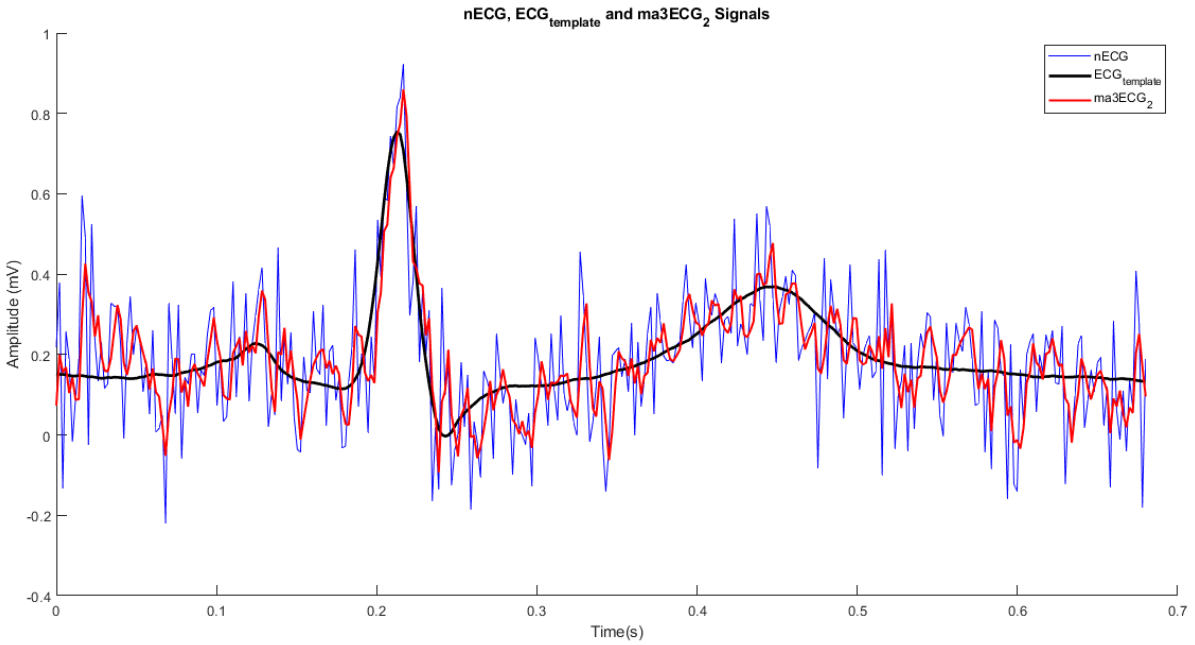

Figure 10: Comparison between nECG, ECG_Template and ma3ECG_2 Signals

MA Filter has managed to reduce the amplitudes of the high variation portions of the noisy ECG signal (blue). However, the filtered signal (red) is still quite different to the original ECG template (black) despite having the rough shape of the original signal.

1.1.3.3 Using fvtool(b,a) Command

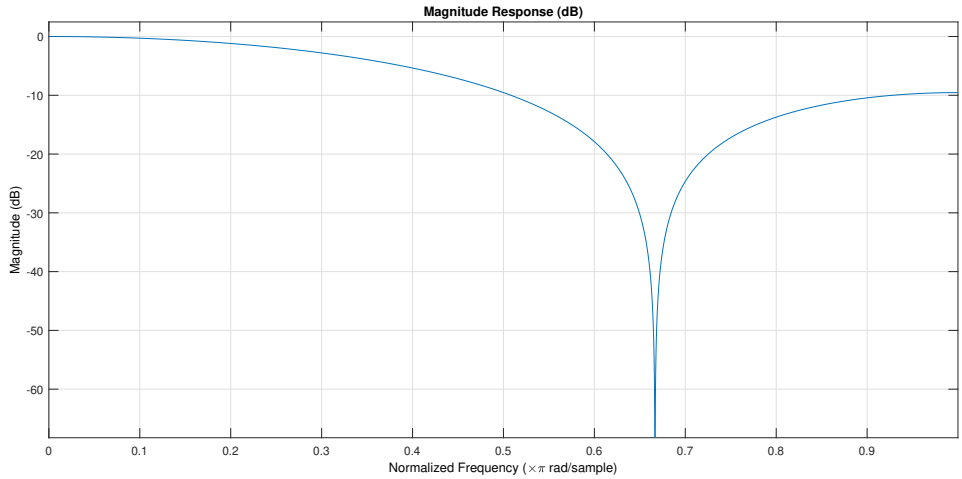

Figure 11: Magnitude Response of the MA(3) Filter

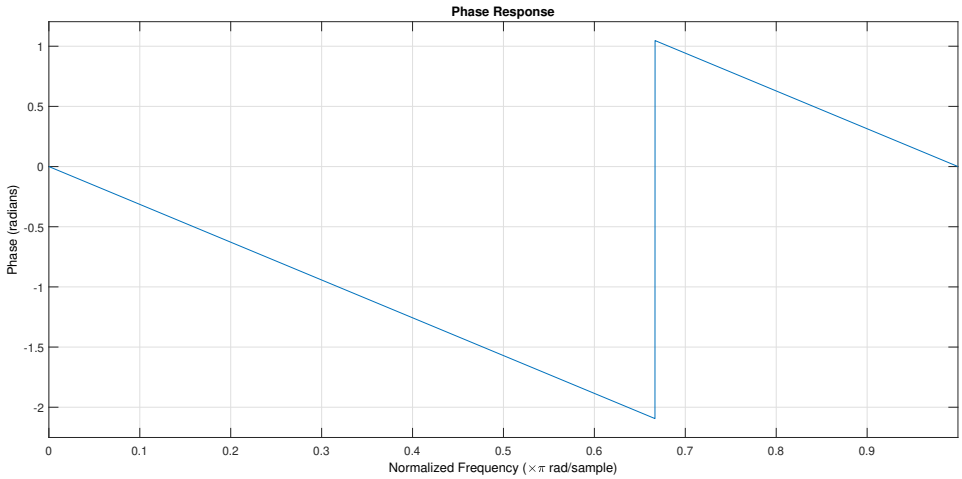

Figure 12: Phase Response of the MA(3) Filter

Let's combine the phase and magnitude response plots for a better insight.

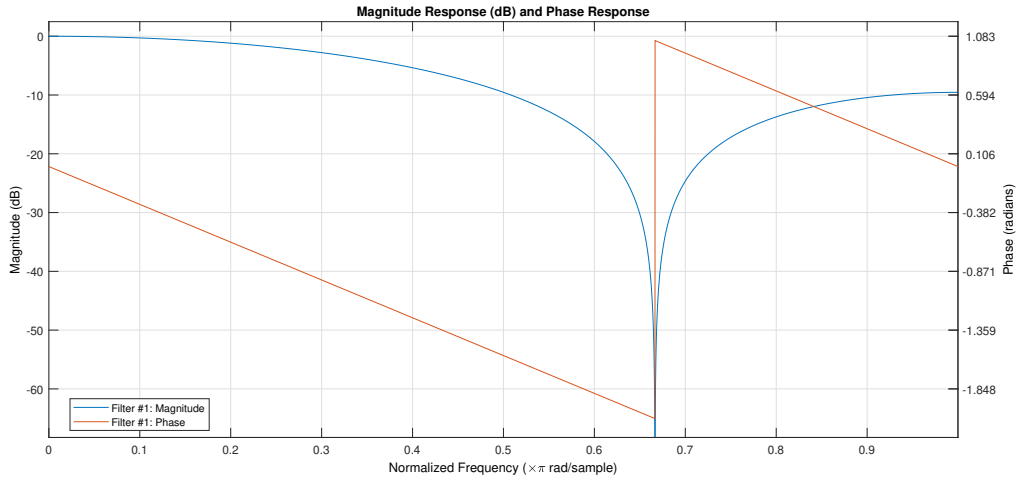

Figure 13: Magnitude and Phase Response of the MA(3) Filter

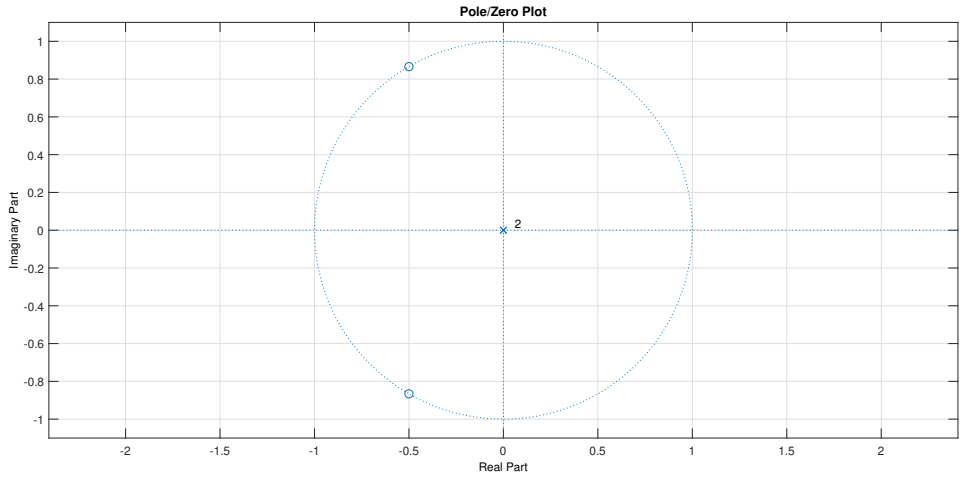

Figure 14: Pole/ Zero Plot of the MA(3) Filter

1.1.4 MA(10) Filter Implementation with MATLAB Built-in Function

1.1.4.1 Identifying the Improvement of MA(10) over the MA(3) Filter

When the order of the moving average filter is increased, the extent to which the signal is smoothened out would be increased. That is, sudden variations in the signal would be suppressed more. Accordingly, high frequency components would be subjected to more attenuation when the order is increased.

This difference could be observed by comparing the magnitude response of the MA filters of order 3 and 10 provided in Fig.11 and Fig.15 respectively. The half-power frequency of the MA(3) filter is close to 0.3 normalized frequency whereas that of the MA(10) filter is even less than 0.1 normalized frequency. Accordingly, we can conclude that higher frequency components are subjected to more attenuation when the order is increased.

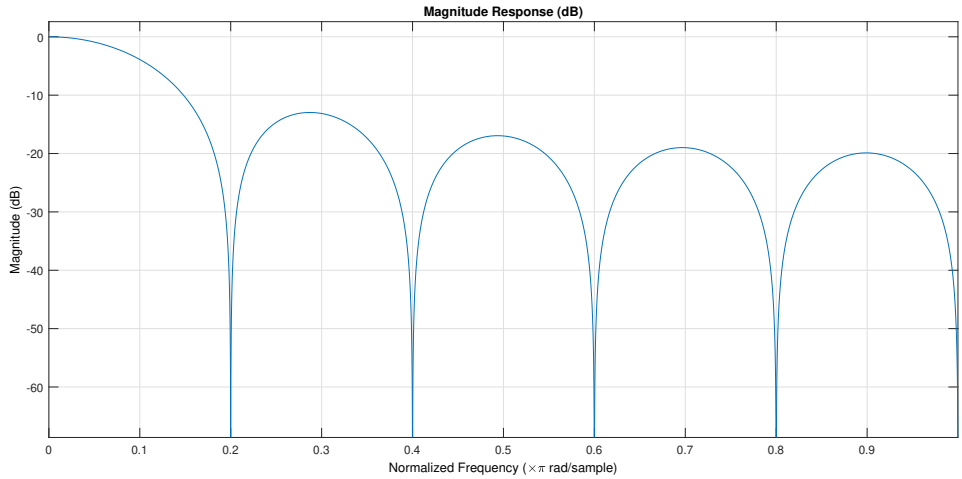

Figure 15: Magnitude Response of the MA(10) Filter

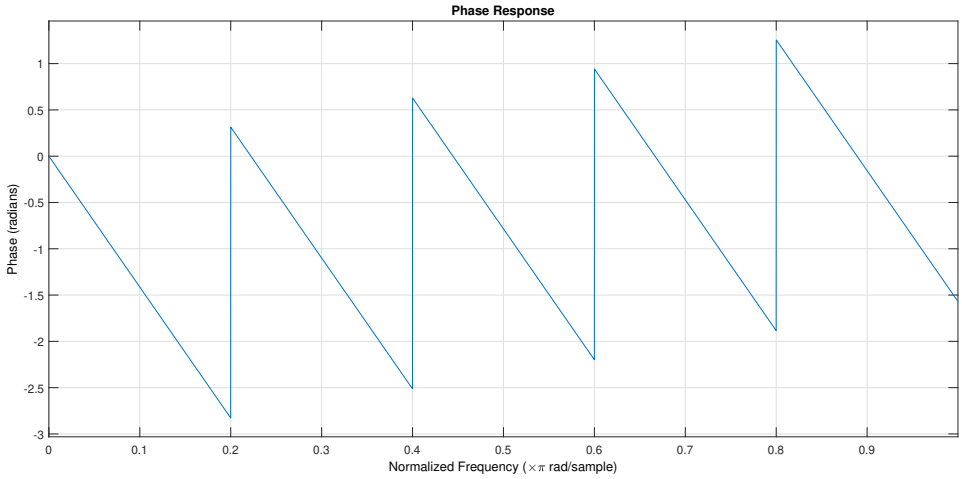

Figure 16: Phase Response of the MA(10) Filter

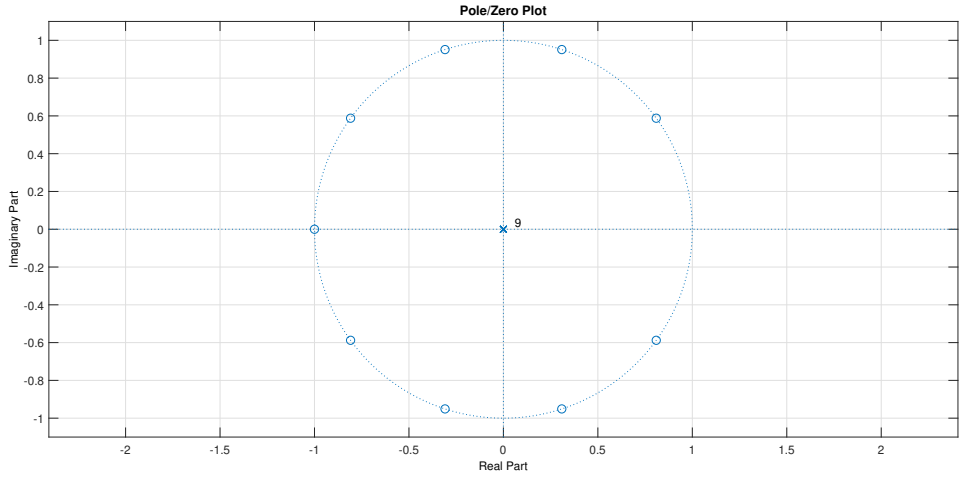

Figure 17: Pole/ Zero Plot of the MA(10) Filter

1.1.4.2 Filtering nECG Signal using MA(10) Filter

In the filter(b,a,x) command,

- a - Coefficients of the output signal in the difference equation
- b - Coefficients of the input signal in the difference equation
- x - Input signal

Since the order of the MA filter is 10, we can express the output signal as,

$$y[n] = \frac{1}{10}(x[n] + x[n-1] + x[n-2] + x[n-3] + x[n-4] + x[n-5] + x[n-6] + x[n-7] + x[n-8] + x[n-9])$$

Thus, $a = 1$ and $b = \left(\frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}\right)$.

1.1.4.3 Plotting nECG, ECG_template, ma3ECG_2 and ma10ECG

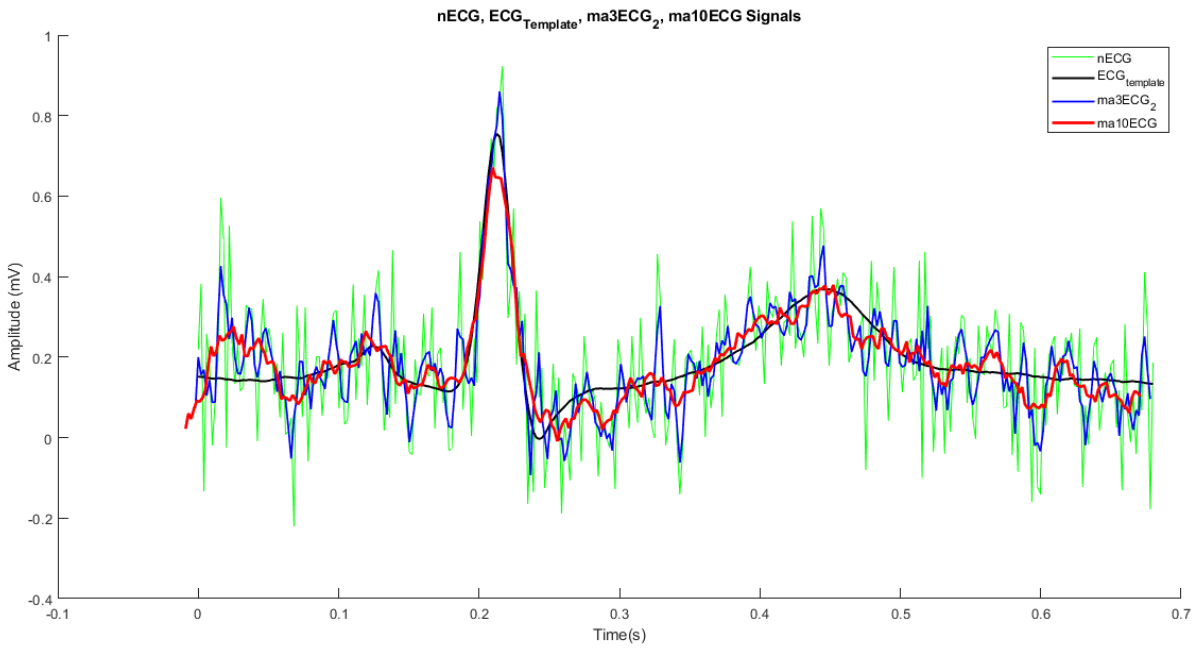

Figure 18: Comparison between nECG, ECG_Template, ma3ECG_2 and ma10ECG Signals

As shown in Fig.18, MA(10) filter has managed to further suppress the high frequency components of the noisy signal, compared to the MA(3) filter. That is, the MA(10) filtered signal is roughly tracing the given ECG template. However, we can see that the peaks and troughs of the signal are too being penalized causing the overall filtered signal to be flatter in nature compared to the original template.

1.1.5 Optimum MA(N) Filter Order

1.1.5.1 Writing a Script to Calculate the MSE

The code for the custom script is found in the customMSECalculation.m file. Let's take the noise-free signal to be $x[n]$ and the filtered signal to be $\hat{x}[n]$. The MSE is calculated by,

$$MSE = \frac{1}{L}\sum_{L}(\hat{x}[n] - x[n])^2$$

1.1.5.2 Determining the Optimum Filter Order

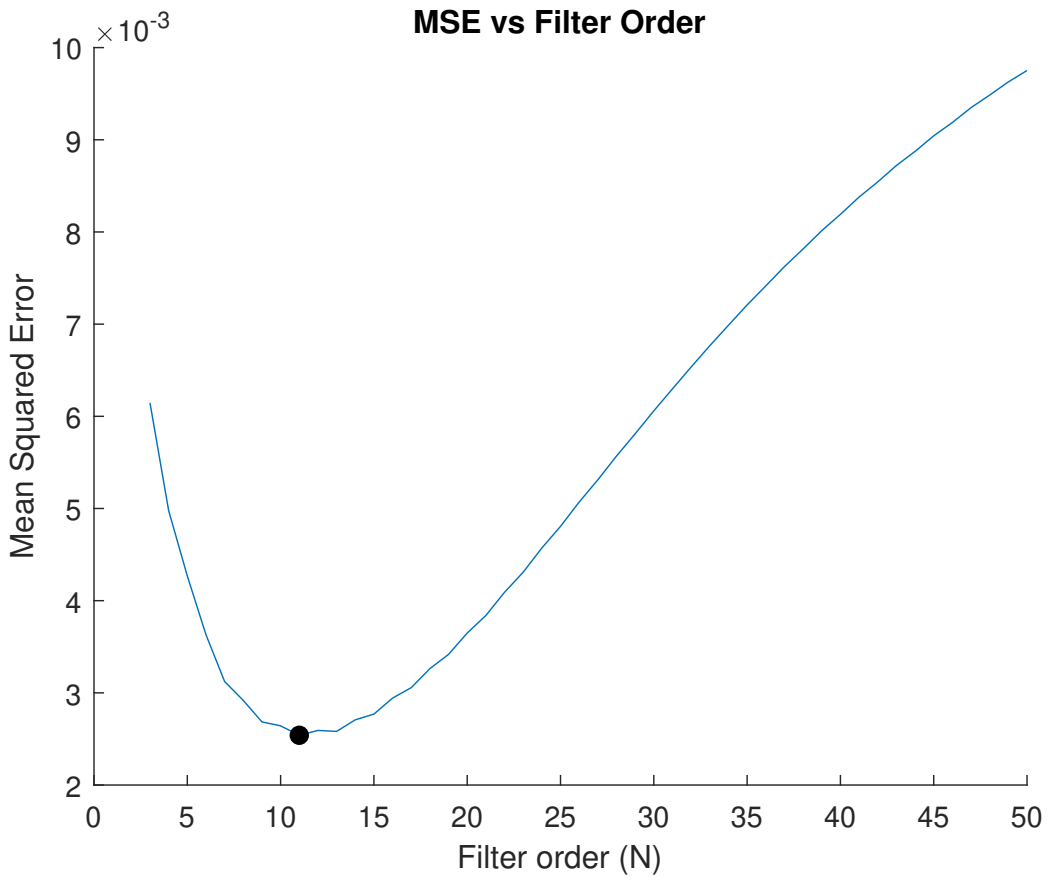

Figure 19: MSE vs Order (N) for the first 50 N values

We can observe from Fig.19 that the MSE value initially decreases with the increasing order value. However, after N = 11, the MSE starts to rise again with the increasing order value. In the following plot, the minimum MSE value and the corresponding order are marked.

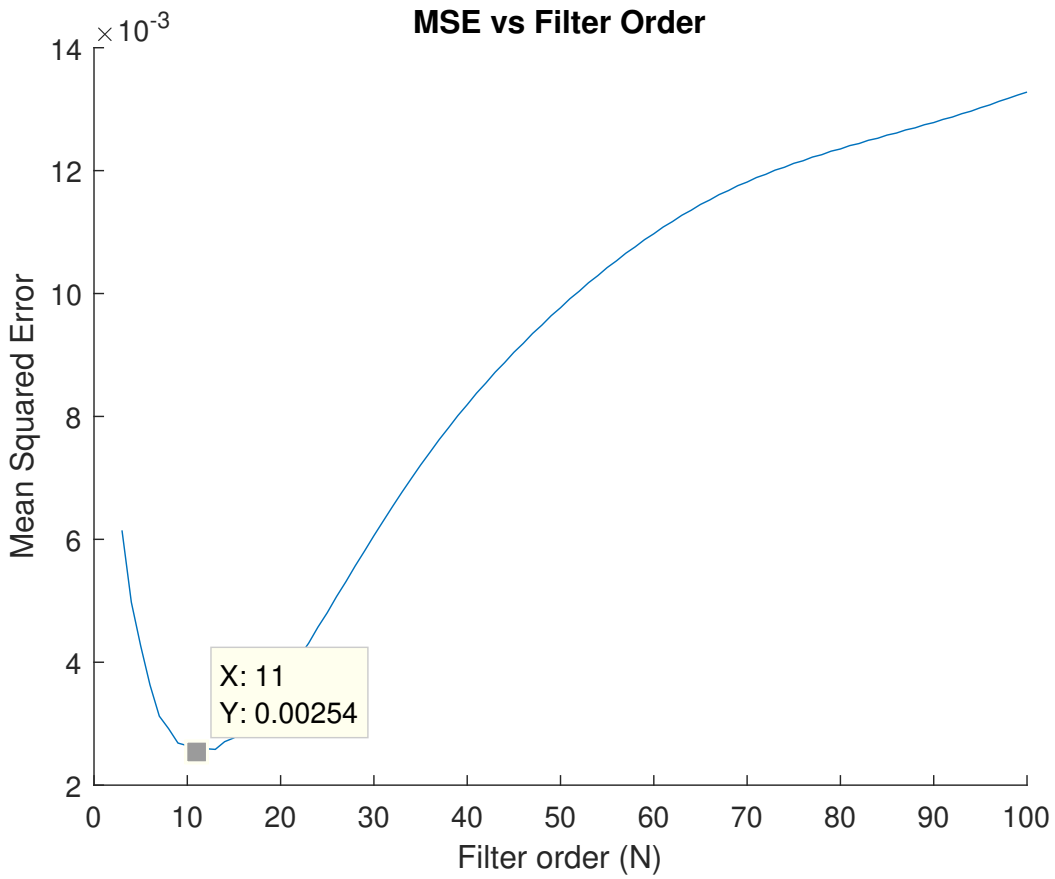

Figure 20: MSE vs Order (N) for the first 100 N values

1.1.5.3 Reasons for Large MSE Values at Higher and Lower Orders

At lower orders, the MA filter would not encapsulate sufficient information to smooth out the signal. Even though high-frequency components are suppressed to a certain extent, it is not enough. Because of this, the MSE error would be high.

On the other hand, when the order is too high, it would consider information over a large duration. As a result, the signal would have a flatter nature. Accordingly, the errors would be high at the peaks and troughs of the signal which eventually causes the MSE to rise.

1.2 Savitzky-Golay (SG) Filter

SG filters fit a polynomial of degree N to an odd number $L' = 2L+1$ of input data points with the objective of minimizing the squared error.

1.2.1 Application of SG(N,L)

1.2.1.1 Applying SG(3,11) Filter on nECG

In the sgolayfilt(x,N,L') command,

- x - Input signal
- N - Order of the polynomial
- L' - 2L + 1 where L is the no. of input data points

1.2.1.2 Plotting nECG, ECG_Template, sg310ECG

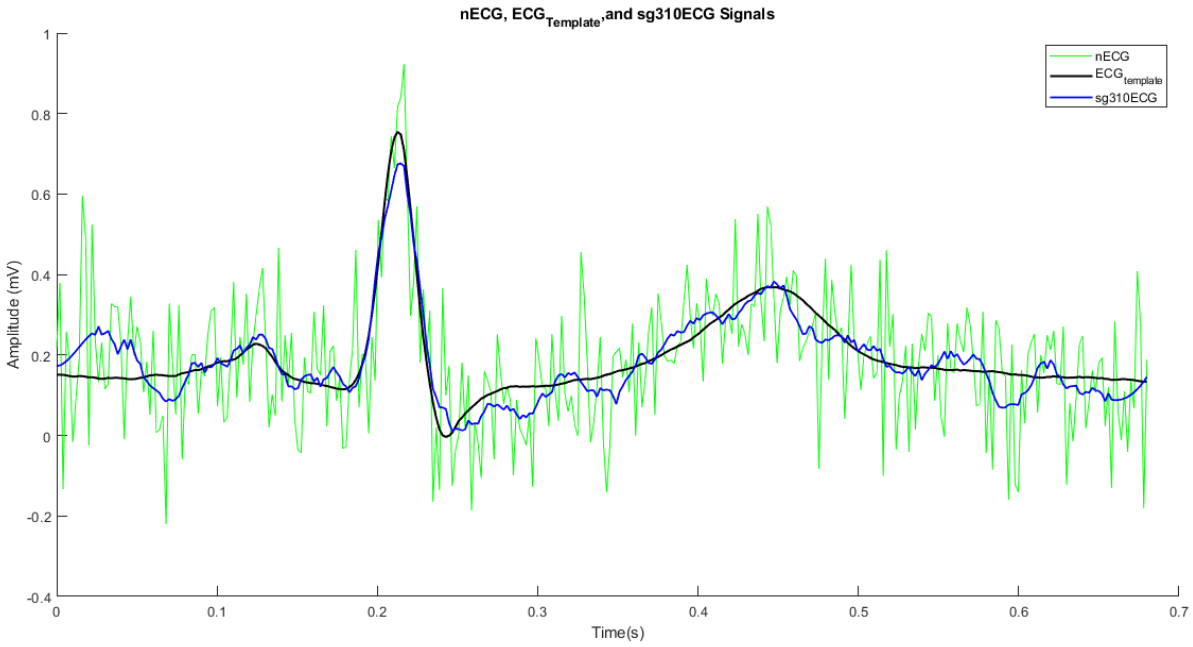

Figure 21: Comparison between nECG, ECG_Template, and sg310ECG Signals

From the above results, it is clear that SG(3,11) has filtered the noisy signal to a reasonable extent and the filtered signal roughly traces the ECG template signal. Moreover, the output from the SG(3,11) filter is quite similar to the MA(10) filter. However, the main difference between the outputs from those two filters is that SG(3,11) filter has not penalized the peaks and troughs of the signal as much as the MA(10) filter which is a plus point. This makes sense as the objective of SG filters is to minimize the squared error which in return would preserve key features of the signal such as peaks and troughs.

1.2.2 Optimum SG(N,L) Filter Parameters

1.2.2.1 Determining the Optimum Filter Parameters

For a range of values of N and L, the MSE between the noise free signal and the filtered signal was calculated. The results are summarized in the following plot.

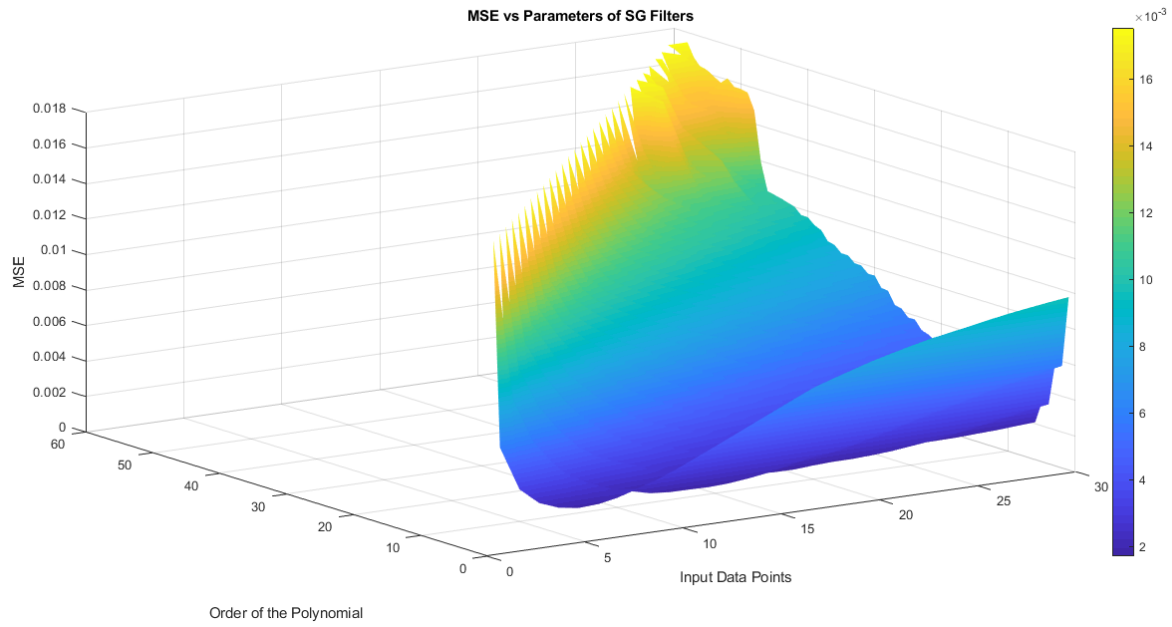

Figure 22: MSE vs Parameters of the SG Filter

The optimum parameters are N = 4 and L = 17. The plot indicating this optimum set of parameters is given below. (Same diagram as Fig.22 except the plot is rotated to get a better view of the minimum MSE point).

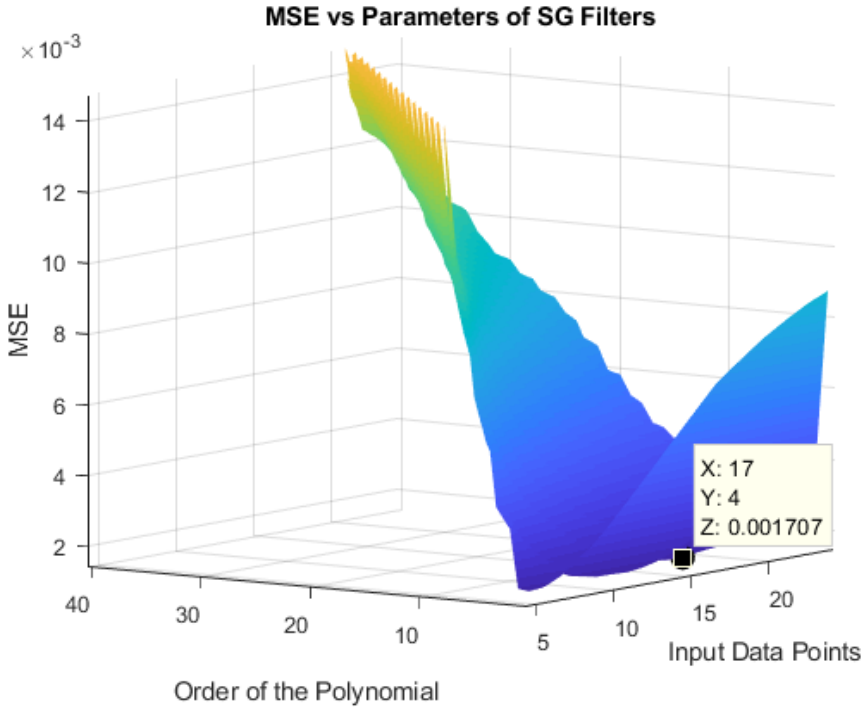

Figure 23: MSE vs Parameters of the SG Filter (with Optimum Parameters' Location)

1.2.2.2 Plotting ECG_Template, sg310ECG, sgECG_optimum

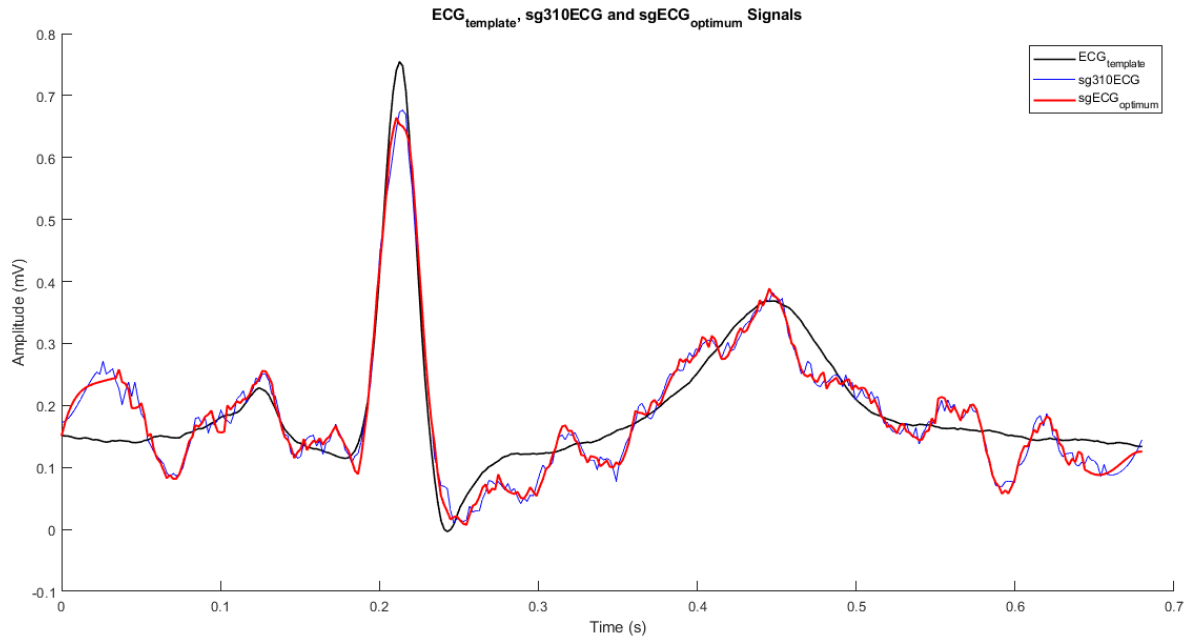

Figure 24: Comparison between ECG_Template, sg310ECG and sgECG_optimum Signals

Even though, SG(4,17) yields the minimum MSE, just by the looks of the plots, we could not observe a noticeable difference between the SG(3,11) filtered signal and the SG(4,17) filtered signal.

1.2.2.3 Comparison between MA(N) and SG(N,L) Filters

The optimum MA filter for the nECG signal was MA(11) filter and its corresponding MSE is 0.00254. The optimum SG filter was the SG(4,17) which had an MSE of 0.001707.

From Fig. 27, we can see that filtered signals obtained from MA(11) and SG(4,17) are nearly similar except for some subtle differences. SG(4,17) signal is slightly smoother than the MA(11) filtered signal. A jitter could be observed throughout the MA(11) filtered signal. Moreover, peaks and troughs are not penalized by the SG(4,17) filter as much as the MA(11) filter.

However, the computational complexity of the SG filter is higher than that of the MA filter as it is required to find the coefficients of a 4th-degree polynomial that minimizes the MSE whereas the MA filter simply adds up 11 terms and divide by 11 to get the signal amplitude at a given point.

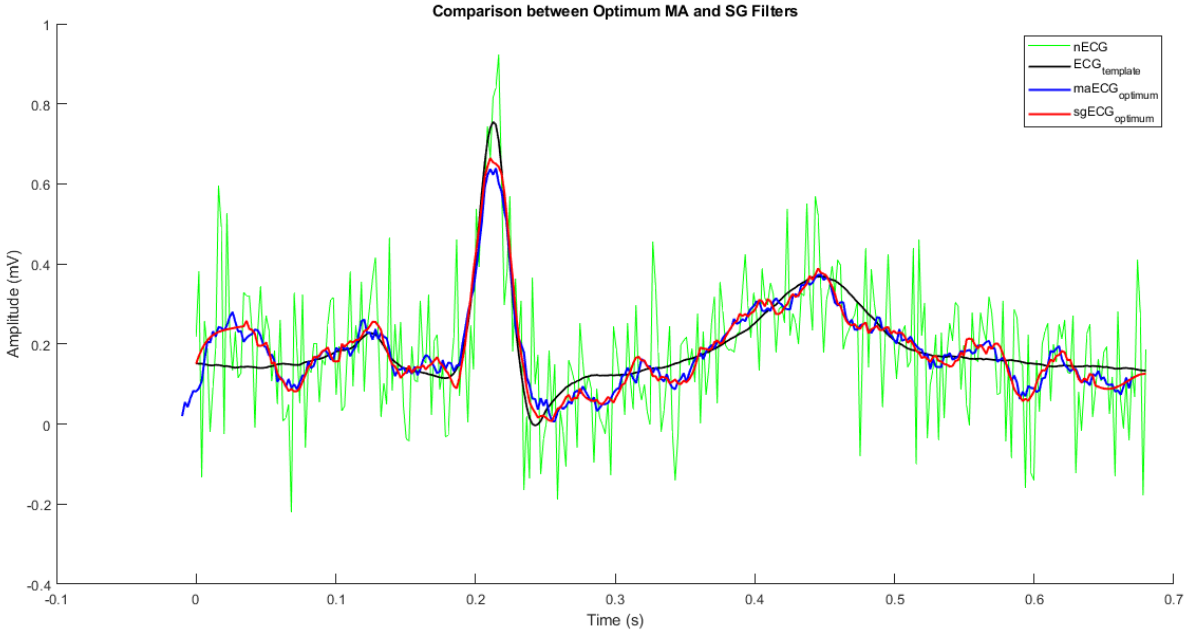

Figure 25: Comparison between nECG, ECG_Template, maECG_optimum and sgECG_optimum Signals

2 Ensemble Averaging

When the noise signal overpowers the input signal, conventional filters such as MA filters would penalize the input signal extensively while trying to get rid of the noise component. In such cases, we can turn towards synchronized averaging techniques. However, to get the best out of those approaches, we ideally require to have a signal with multiple measurements or repetitive patterns.

2.1 Signal with Multiple Measurements

In this section, Auditory Brain Response (ABR) is used as the signal with multiple measurements and the properties of the signal are summarized in the following table.

Name of the recording	ABR_rec.mat
Sampling frequency	40kHz
Amplitude range	mV
Duration of the ABR	10ms from the point of simulation

Table 2: Properties of the ABR Pulse Train

2.1.1 Preliminaries

2.1.1.1 Clearing the Workspace and Command Window

The workspace variables were cleared using the 'clear all' command and the command window was cleaned using the 'clc' command.

2.1.1.2 Loading ABR_rec.mat

'Load' command of MATLAB was used for this task.

2.1.1.3 Plotting the Train of Stimuli and ABRs

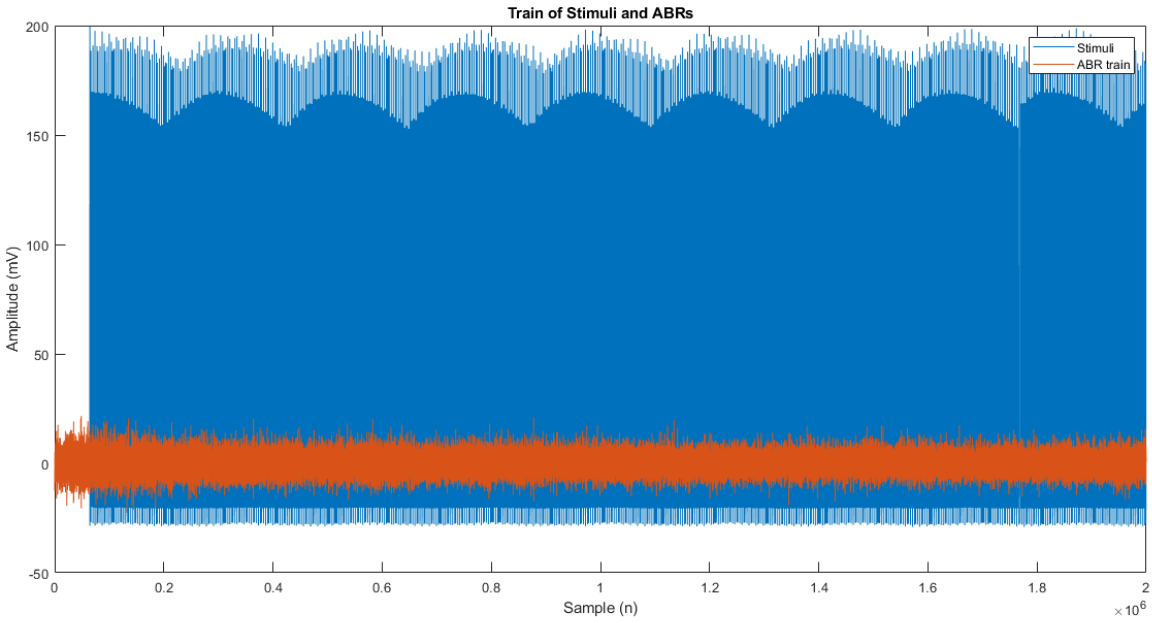

Figure 26: Train of Stimuli and ABRs

For better visualization, let's take a look at a slice of the above plot to clearly observe the stimuli and ABRs.

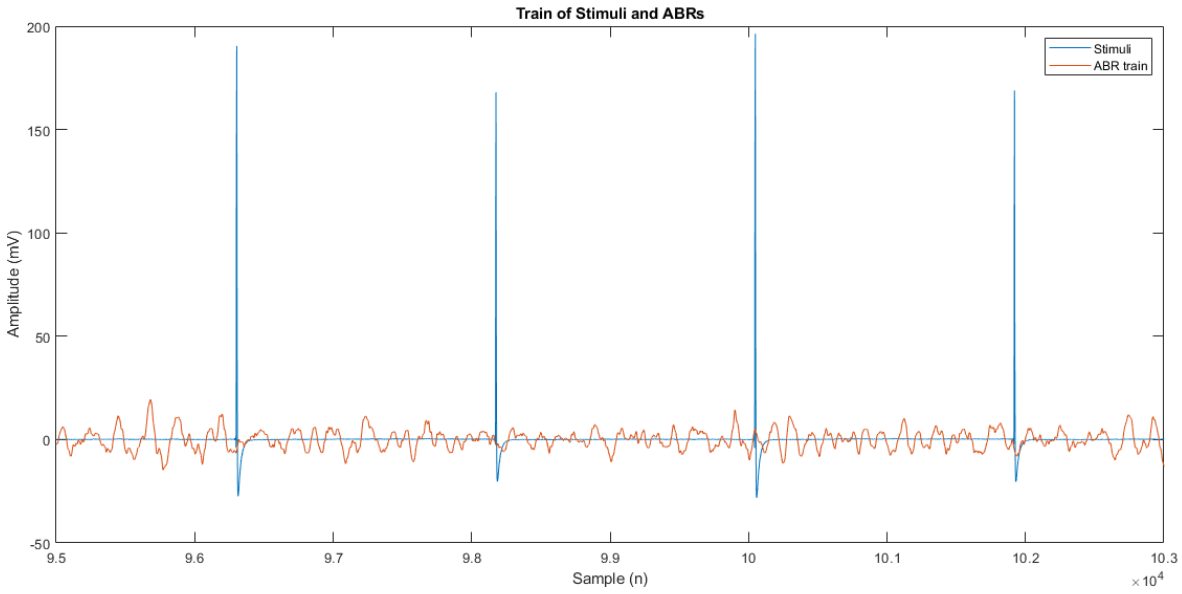

Figure 27: Train of Stimuli and ABRs (Zoomed-in)

2.1.1.4 Determining a Voltage Threshold

As given in the assignment, 50 was used as the threshold to automatically detect the likely stimuli occurrences. (Implemented in the code file)

2.1.1.5 Extracting Actual Stimulus Points

After determining estimated locations using an arbitrary threshold value, the resulting location vector (*thresh*) was subjected to a for loop to determine the actual stimulus points. Accordingly, if the difference between successive elements of *thresh* is greater than one, then the latter element is considered to be an actual stimulus point. Accordingly, 1033-D *stem_point* vector was obtained. (Implemented in the code file)

2.1.1.6 Windowing ABR Epochs

A window length of 12ms (-2ms to 10ms) with reference to the stimulus time point was selected for the -80 to 399 sample points which were sampled at a sampling frequency of fs=40 kHz. (Implemented in the code file)

2.1.1.7 Calculating the Average of All Epochs

Let the ensemble average vector be $\bar{x}$ (i.e. $\bar{x} \in \mathbb{R}^{480\times1}$), no. of stimulus points be n where n = 1033 and the epochs be $x \in \mathbb{R}^{480\times1033}$.

$$\bar{x} = \frac{1}{n}\sum_{n} x$$

2.1.1.8 Plotting the Ensemble Averaged ABR Waveform

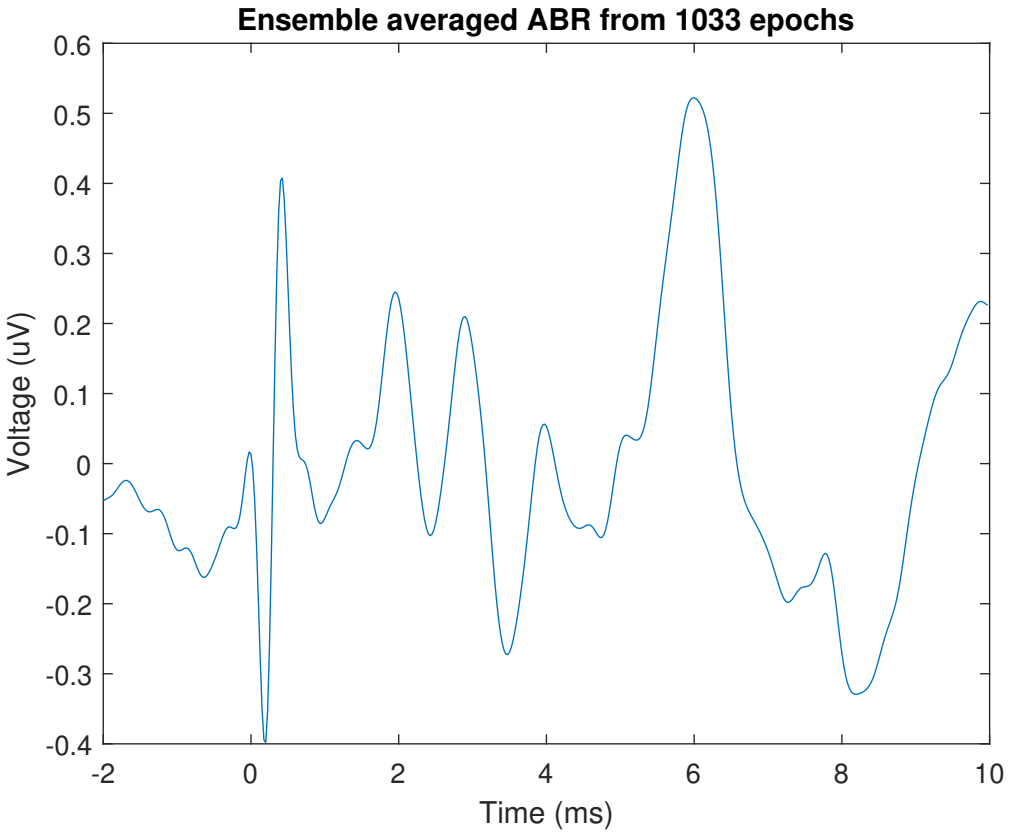

Figure 28: Ensemble Averaged ABR Waveform

2.1.2 Improvement of the SNR

2.1.2.1 Writing a MATLAB Script to Calculate Progressive MSEs

$$MSE_k = \sqrt{\frac{\sum_{n=1}^{N} \left(\hat{y}[n] - \tilde{y}_k[n]\right)^2}{N}} \qquad where \quad k = 1, 2, \ldots, M$$

$$\hat{y}[n] : Template$$

$$\tilde{y}_k[n] = \frac{\sum_{i=1}^{k} x_i[n]}{k} \qquad where \quad x_i[n] \ is \ a \ single \ ABR \ epoch$$

$$N : Length \ of \ a \ single \ ABR \ epoch$$

$$M : Epochs \ in \ the \ ensemble \ average$$

The progressive MSE calculation is implemented in the code file.

2.1.2.2 Plotting MSE_k vs k

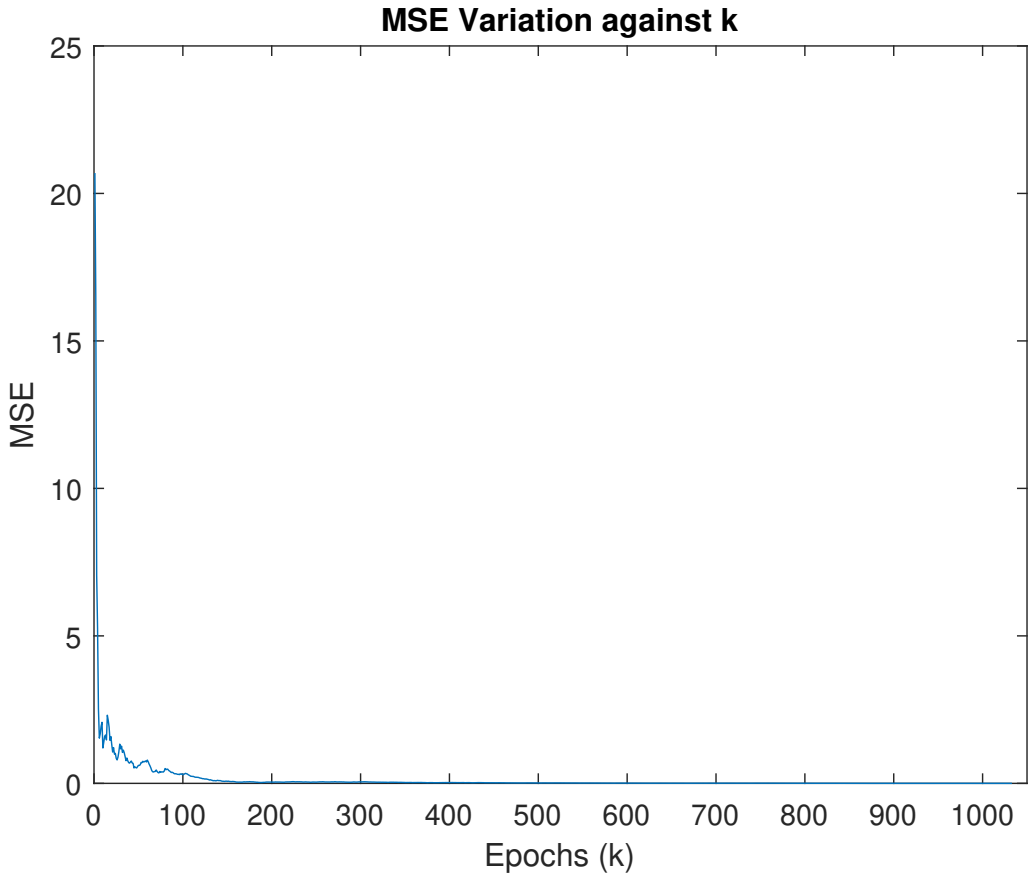

Figure 29: MSE_k vs k

For a better visualization, let's inspect the variation in the logarithmic domain.

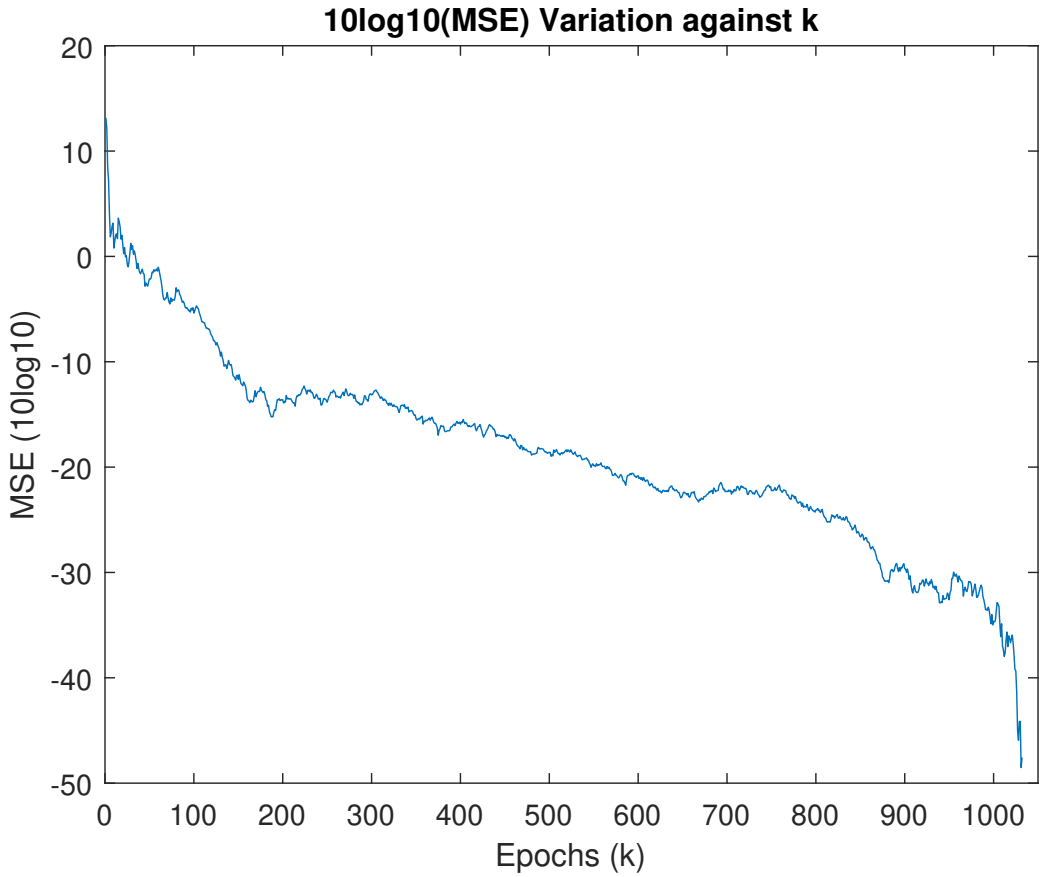

Figure 30: $10log_{10}(MSE_k)$ vs k

We can observe a rapid decrease in MSE as we increase k, initially. That's, the MSE converges at a rapid rate initially. Then, the gradient of the curve is gradually decreasing yet it is steadily converging to zero with increasing k values. This behaviour is clearly visible in the error plot drawn in the log domain.

2.2 Signal with Repetitive Patterns

In this section, an ECG pulse train is used as the signal with repetitive patterns and the properties of the signal are summarized in the following table.

Name of the recording	ECG_rec.mat
Recorded lead	II
Sampling frequency	128Hz
Amplitude range	mV

Table 3: Properties of the ECG Pulse Train

2.2.1 Viewing the Signal and Addition of AWGN

2.2.1.1 Clearing the Workspace and Loading ECG_rec.mat

The workspace variables were cleared using the 'clear all' command and the command window was cleaned using the 'clc' command. 'Load' command of MATLAB was used for loading the ECG_rec.mat file.

2.2.1.2 Plotting the Data and Observing Waveforms

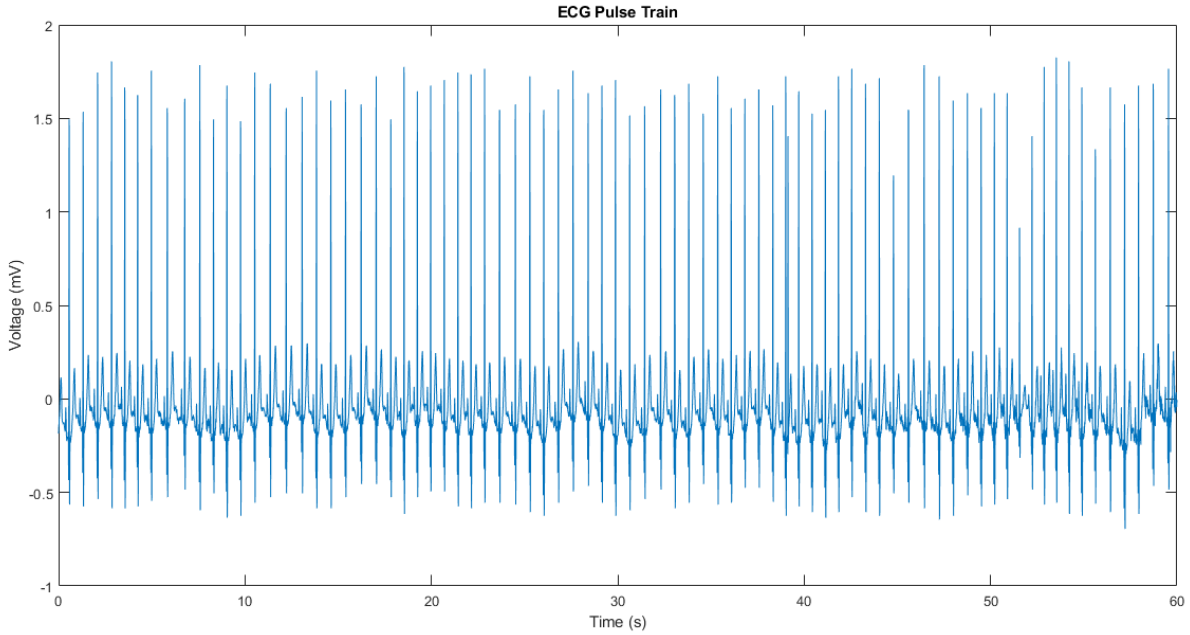

Figure 31: ECG Pulse Train

Let's visualize a few ECG pulses.

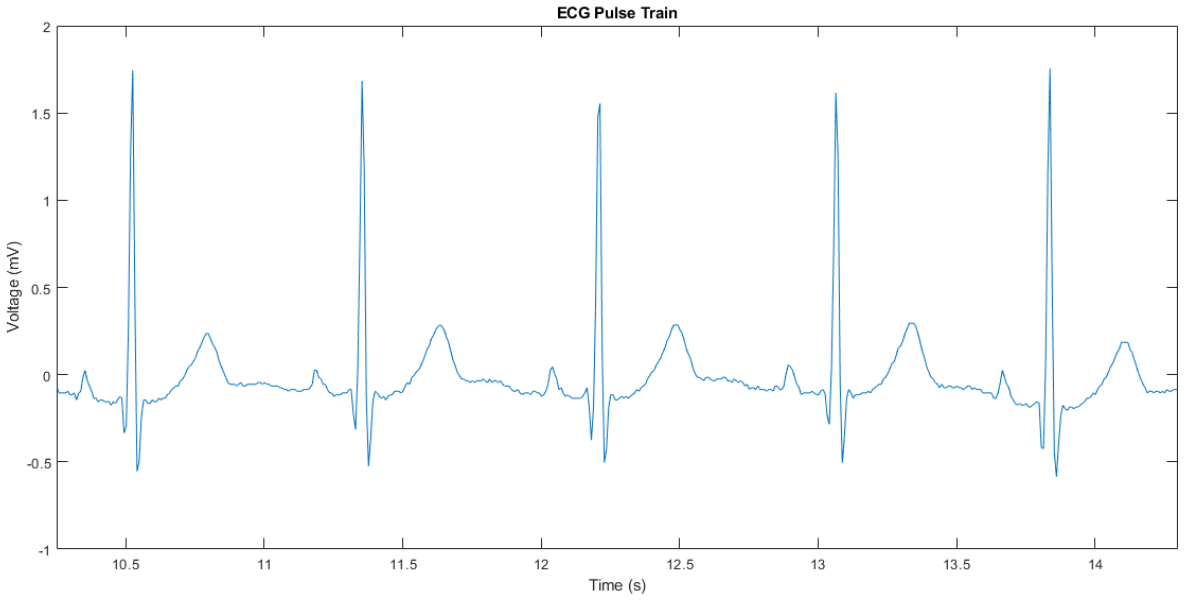

Figure 32: Few Pulses of the ECG Train

2.2.1.3 Extracting a Single PQRST Waveform

The MATLAB's *findpeaks* command was used to determine the peaks in the ECG pulse train. 79 peaks were detected and after manually inspecting a few waveforms, the following PQRST waveform (waveform corresponding to the 17th peak) was assigned to the ECG_Template.

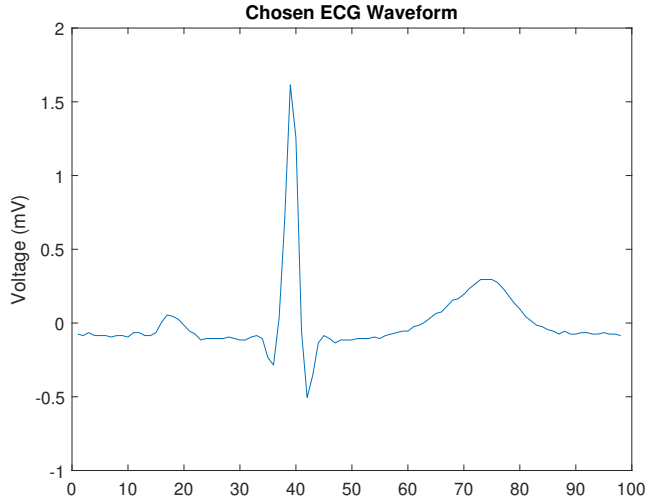

Figure 33: Chosen ECG Waveform

2.2.1.4 Adding AWGN of 5dB to ECG_rec.mat

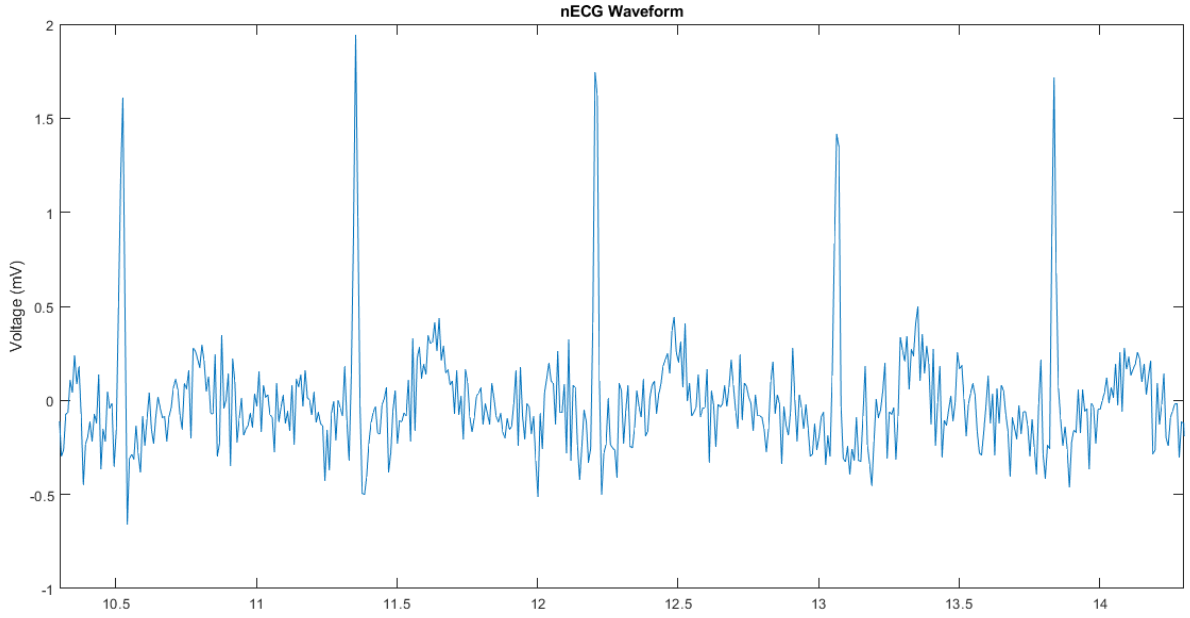

Figure 34: ECG_rec Signal after the Addition of AWGN of 5dB

Fig. 34 represents only a portion of the nECG. We can clearly see the impact of noise addition on the ECG signal by comparing Fig. 32 (noise-free) with Fig.34 (noisy).

2.2.2 Segmenting ECG into Separate Epochs and Ensemble Averaging

2.2.2.1 Calculating the Normalized Cross-correlation between ECG_Template and nECG

$$\Theta_{xy} = \frac{\sum_{n=1}^{N}\left([x[n]-\bar{x}][y[n-k]-\bar{y}]\right)}{\sqrt{\sum_{n=1}^{N}\left(x[n]-\bar{x}\right)^2 \sum_{n=1}^{N}\left(y[n-k]-\bar{y}\right)^2}}$$

$$\begin{aligned} x[n] &: nECG \\ y[n] &: ECG\_Template \\ \bar{x}[n] &: Mean\ of\ nECG \\ \bar{y}[n] &: Mean\ of\ ECG\_Template \\ k &: lag \\ N &: Length\ of\ the\ Template \end{aligned}$$

2.2.2.2 Plotting the Normalized Cross-correlation Values

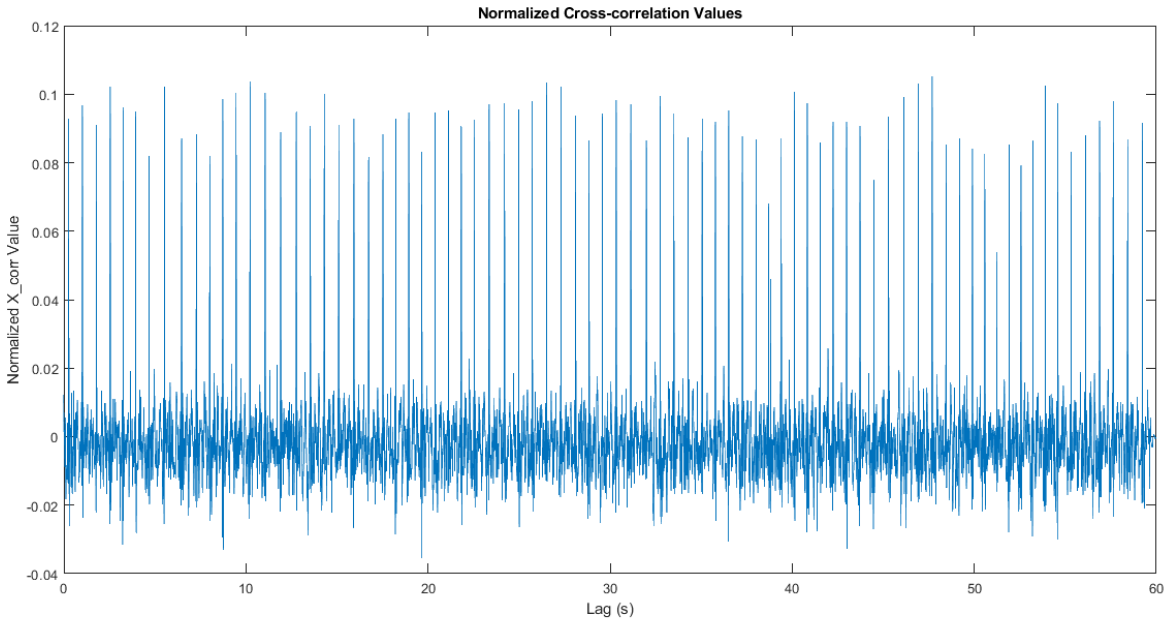

Figure 35: Normalized Cross-correlation Values

From -60 to 0 lag (s), the X_corr is 0. Therefore, that region is trimmed in the above plot.

2.2.2.3 Segmenting ECG Pulses by Defining a Threshold and Storing Pulses

To determine the starting point of an ECG pulse, a threshold of 0.08 was used as the optimum threshold after manually trying out other arbitrary thresholds. Moreover, once the starting points are thresholded, the list was further filtered so that the minimum gap between two consecutive starting points is greater than 10.

2.2.2.4 Calculating and Plotting the Improvement in SNR

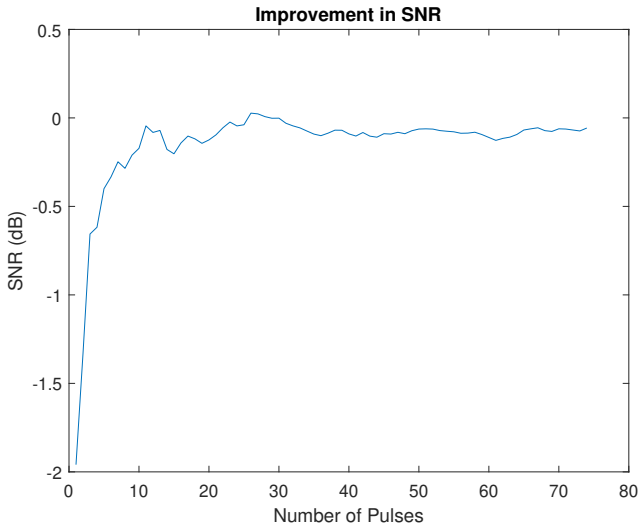

Figure 36: Improvement in SNR

2.2.2.5 Plotting and Comparing an nECG Pulse and Two Ensemble Averaged Pulses

As the noisy ECG pulse, the noisy pulse corresponding to the ECG_template was used for convenience. The two arbitrary epoch values chosen for the ensemble averaging are 30 and 60. We can clearly observe from the plot below, that the ensemble-averaged signals have managed to suppress the noise to a significant extent while preserving the ECG template's shape and key features.

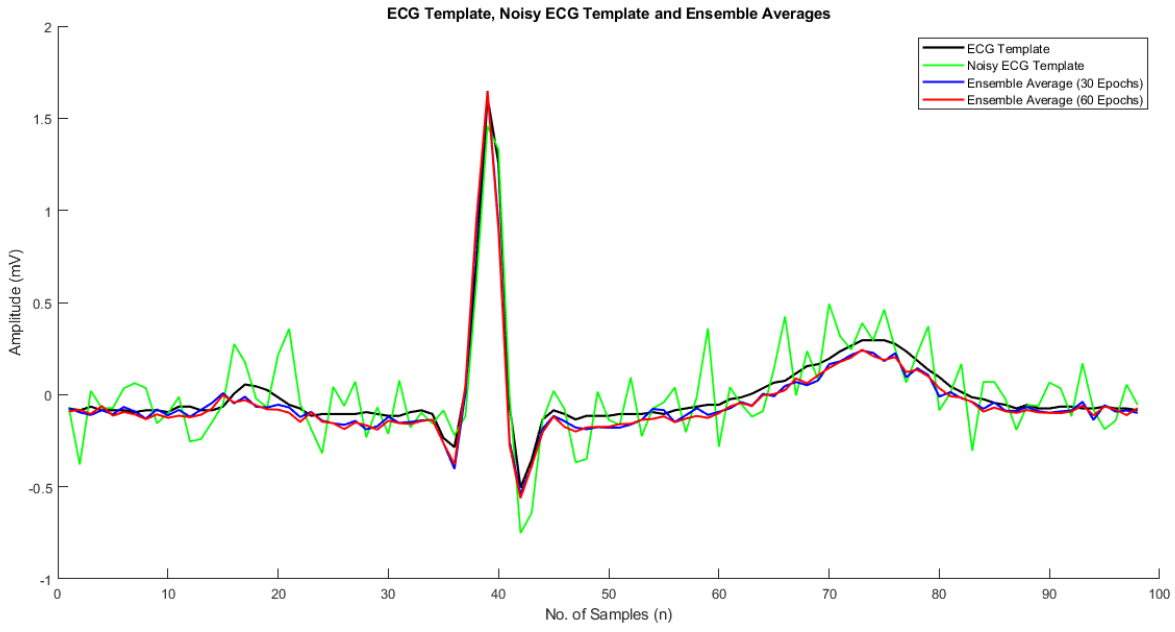

Figure 37: Comparison between an nECG Pulse and Two Ensemble Averaged Pulses

2.2.2.6 Justification

Claim to be Justified: *'It is a better method to use points of maximum correlation with noisy ECG pulse train rather than merely detecting the R-wave to segment the ECG pulse train into separate epochs'*

Justification:

Via cross-correlation, we can detect individual ECG pulses more accurately compared to peak detection as the peak detection approach is susceptible to errors owing to its over-reliance on the assumption that the time window is fixed. However, cross-correlation is more flexible against varying time windows. Moreover, the subtle variations in the P wave, QRS complex and T wave would also directly impact the performance of the peak detection approach (R-wave detection). Nevertheless, as shown in the following plot, we can see that the points of maximum correlation and starting points determined via peak detection are nearly aligned.

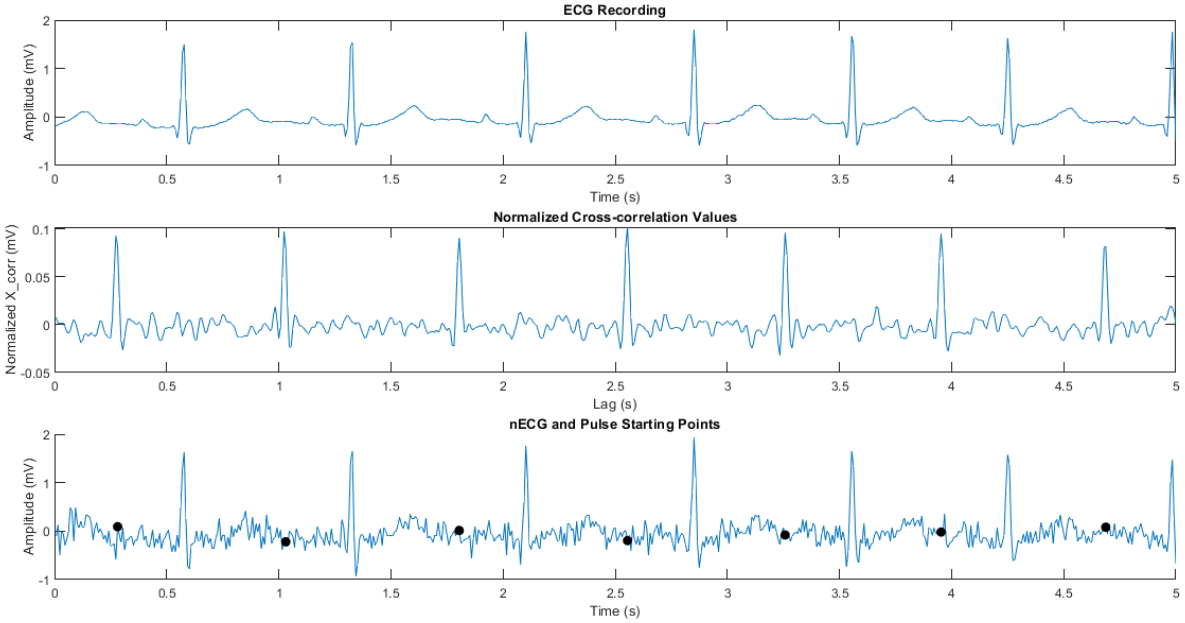

Figure 38: Comparison between Points of Maximum Correlation and Starting Points Detected via Peak Detection

3 FIR Derivative Filters

3.1 FIR Derivative Filter Properties

3.1.1 Observing First Order and Central Difference Derivative Filters

First order filter:

$$y[n] = \frac{1}{T}\left(x[n] - x[n-1]\right) \quad \xrightarrow{\mathcal{L}} \quad Y(Z) = \frac{1}{T}\left(X(z) - X(z)z^{-1}\right) \rightarrow H(z) \quad = \frac{1}{T}\left(1 - z^{-1}\right)$$

Three-point central difference derivative filter:

$$y[n] = \frac{1}{2T}\left(x[n] - x[n-2]\right) \quad \xrightarrow{\mathcal{L}} \quad Y(Z) = \frac{1}{2T}\left(X(z) - X(z)z^{-2}\right) \rightarrow H(z) \quad = \frac{1}{2T}\left(1 - z^{-2}\right)$$

3.1.1.1 First Order Filter

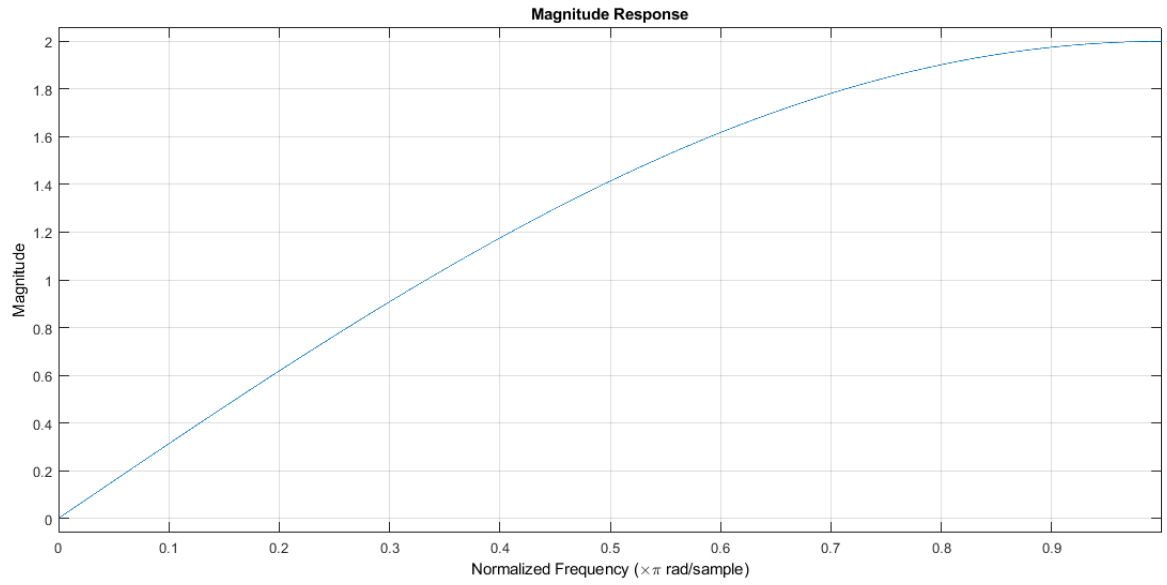

Figure 39: Magnitude Response (Linear) of the First Order Filter

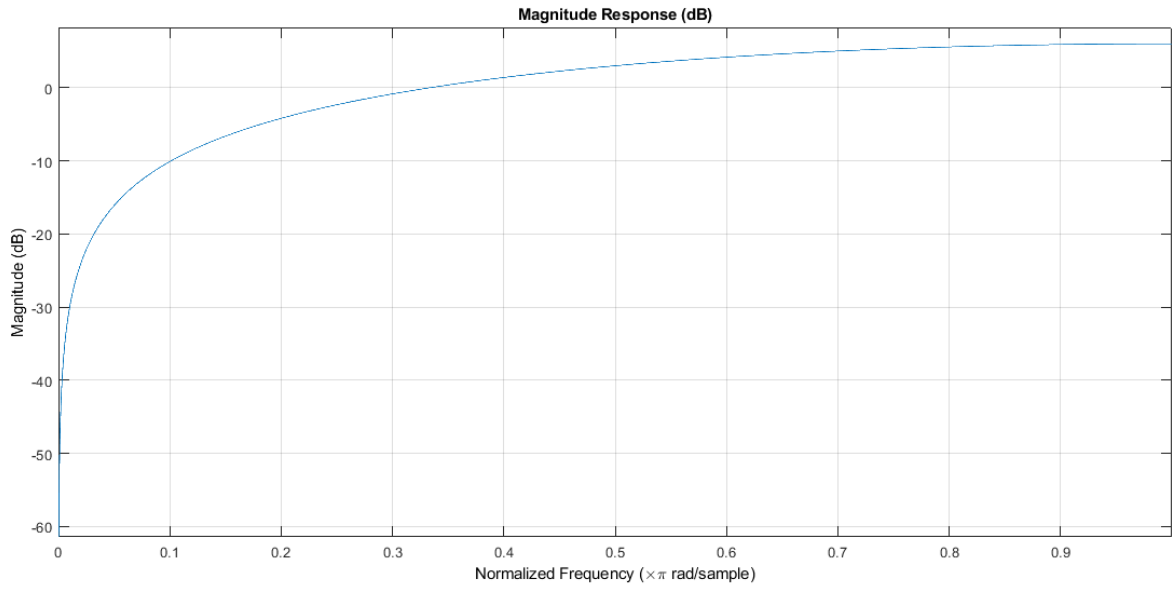

Figure 40: Magnitude Response (Log) of the First Order Filter

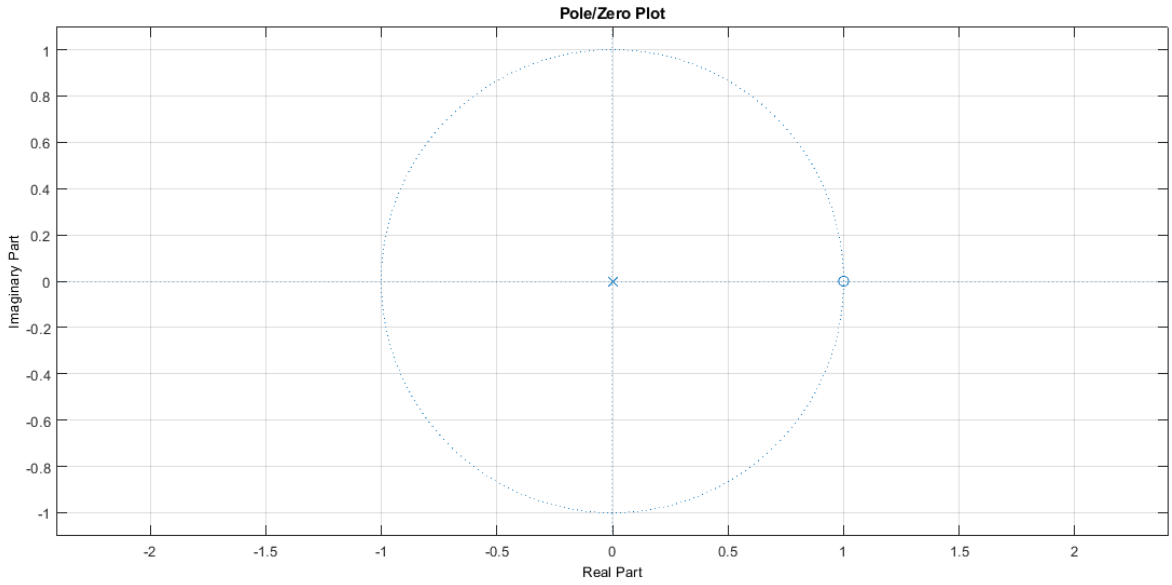

Figure 41: Pole/ Zero Plot of the First Order Filter

3.1.1.2 Three Point Central Difference Derivative Filter

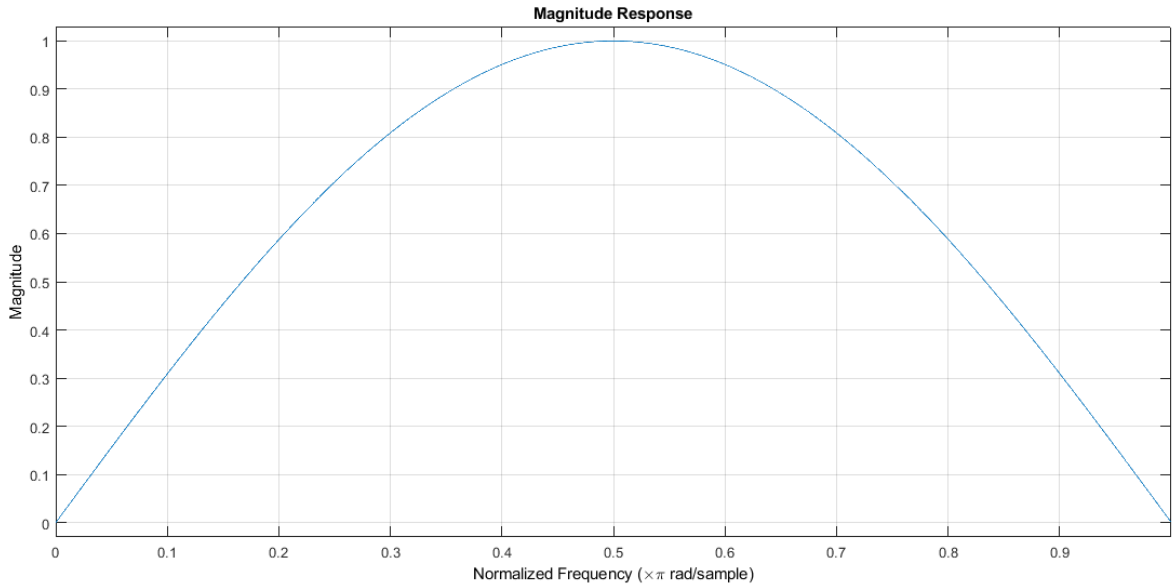

Figure 42: Magnitude Response (Linear) of the Three Point Central Difference Derivative Filter

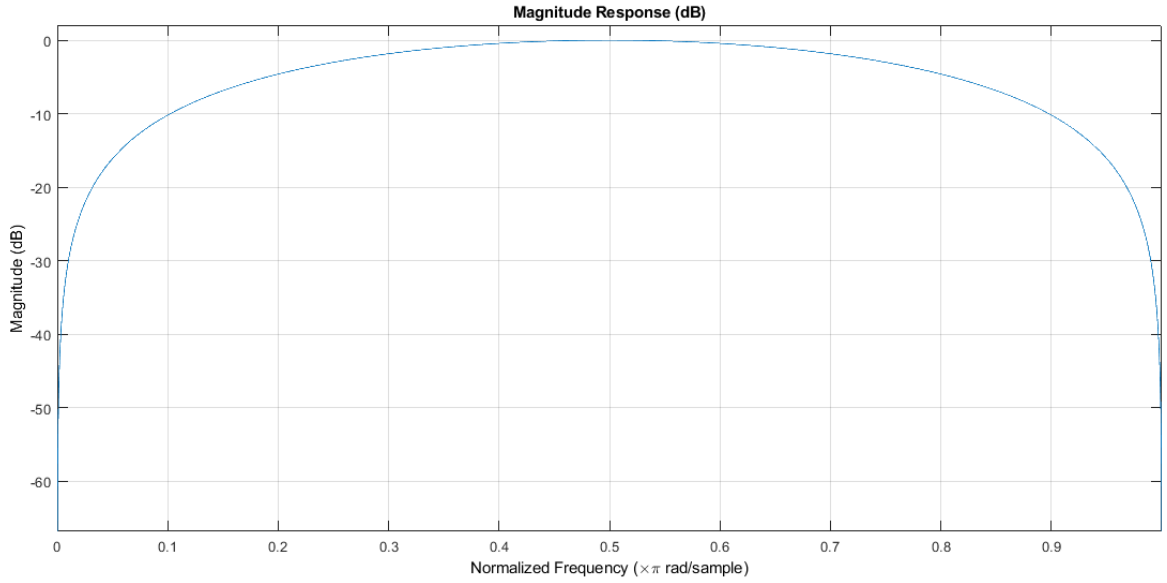

Figure 43: Magnitude Response (Log) of the Three Point Central Difference Derivative Filter

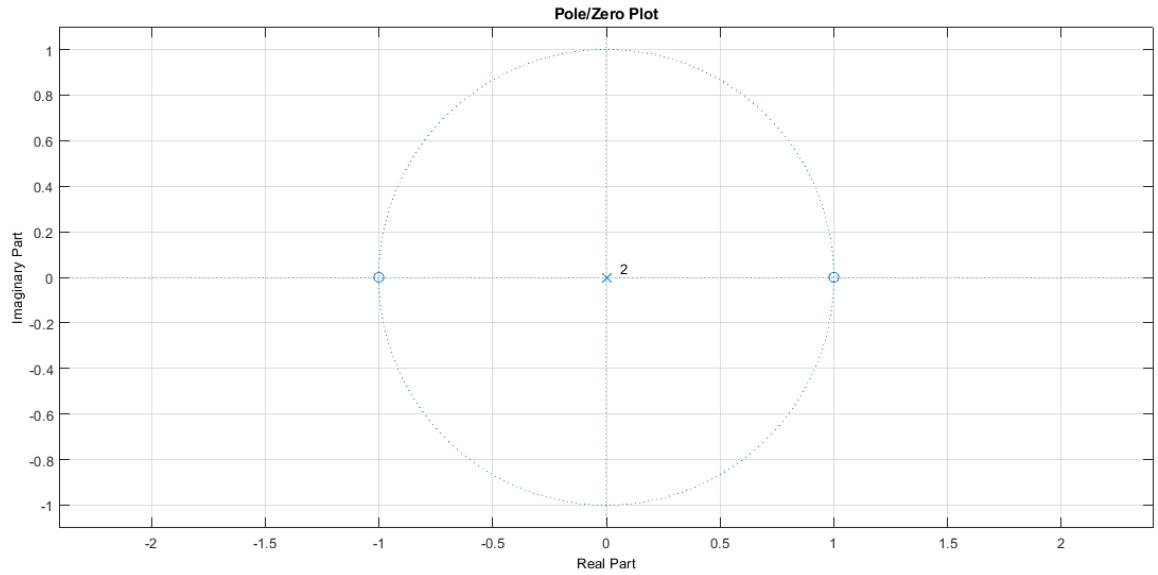

Figure 44: Pole/ Zero Plot of the Three Point Central Difference Derivative Filter

3.1.2 Multiplying Factors to Maximize Gains

The derived transfer functions for the derivative filters are influenced by the T value (sampling period). Moreover, in some cases, the gain from $H(z)$ could exceed 1. This can be observed in Fig. 39 and Fig. 40. When the gain is greater than 1, it could result in distortions.

To negate the dependency on the sampling period and to restrict the maximum gain to unity, we could multiply the transfer function $H(z)$ by a gain G.

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

For the first order filter,

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

For the three point difference derivative filter,

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

The magnitude responses are replotted after adjusting the coefficients to ensure that the maximum gain is unity. The phase response is not changed. They will be the same as before. The magnitude responses with adjusted coefficient are given below.

3.1.2.1 First Order Filter

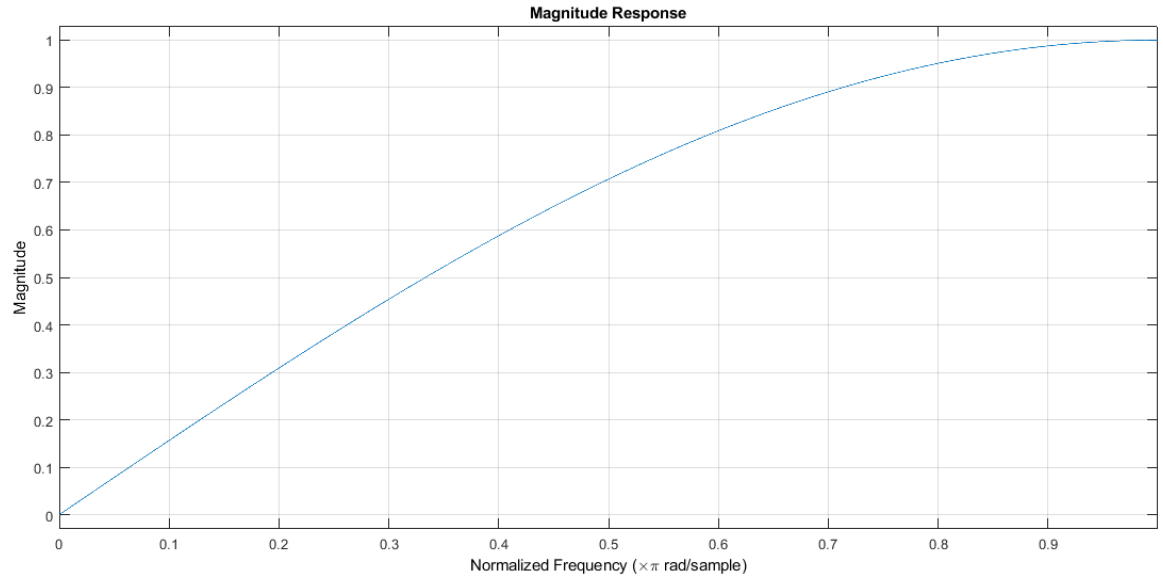

Figure 45: Magnitude Response (Linear) of the First Order Filter (Unity Gain)

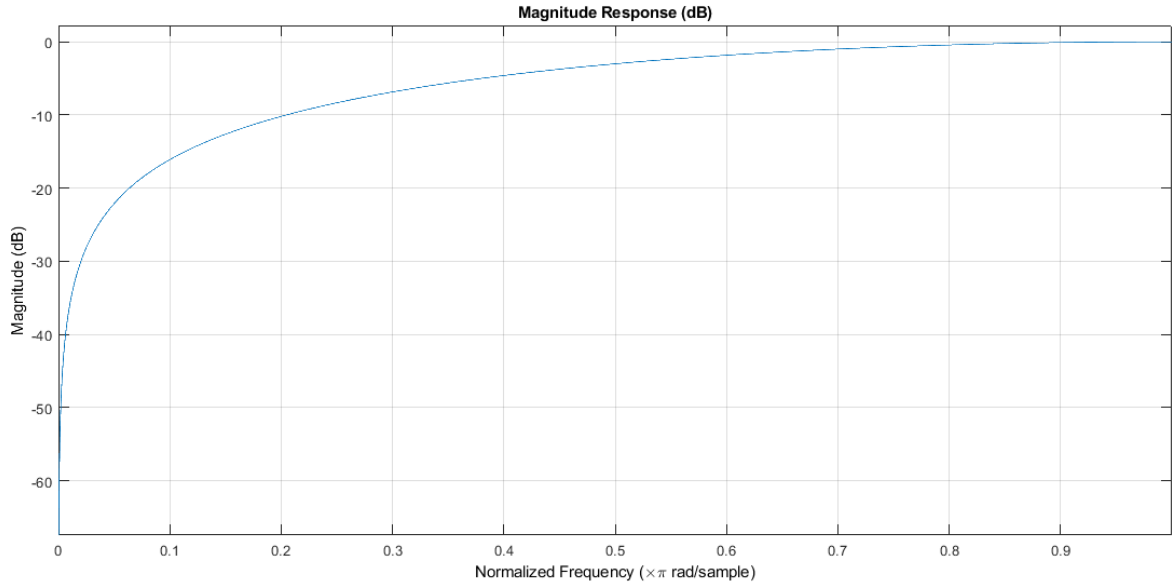

Figure 46: Magnitude Response (Log) of the First Order Filter (Unity Gain)

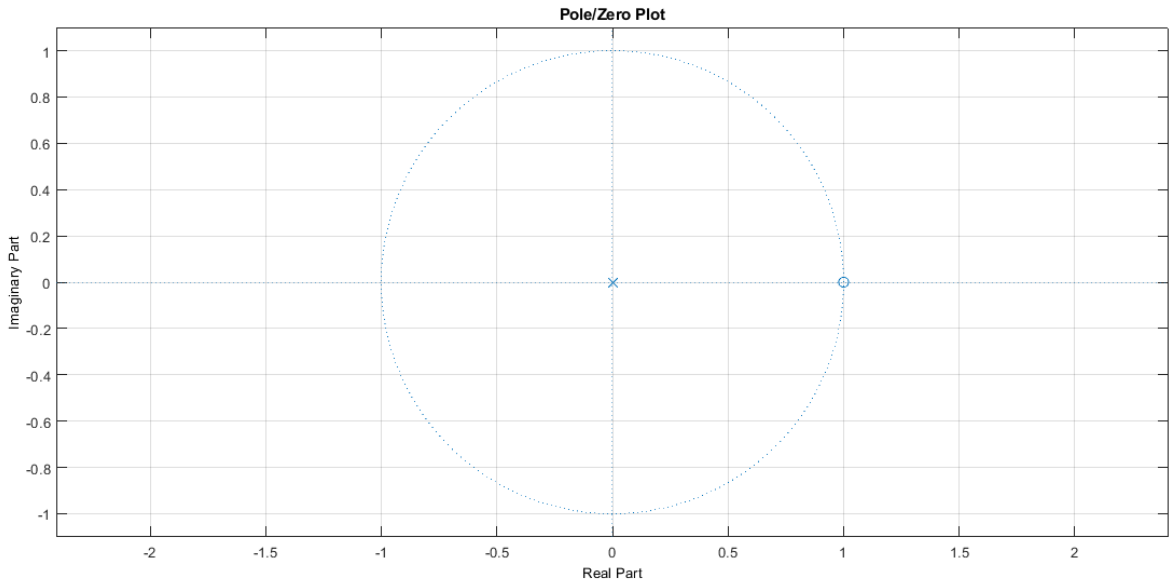

Figure 47: Pole/ Zero Plot of the First Order Filter (Unity Gain)

3.1.2.2 Three Point Central Difference Derivative Filter

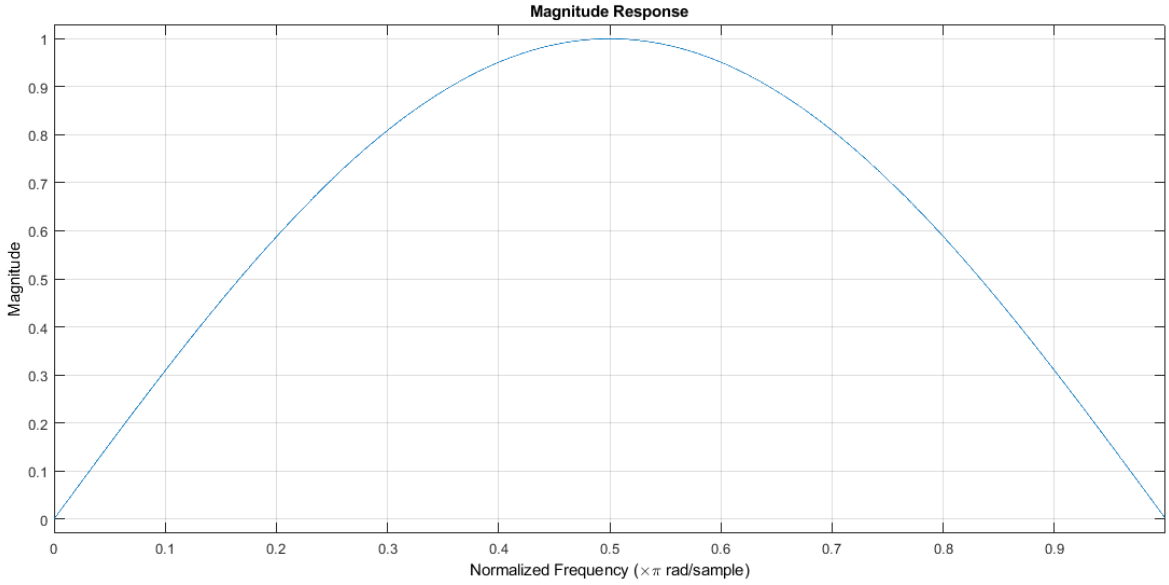

Figure 48: Magnitude Response (Linear) of the Three Point Central Difference Derivative Filter (Unity Gain)

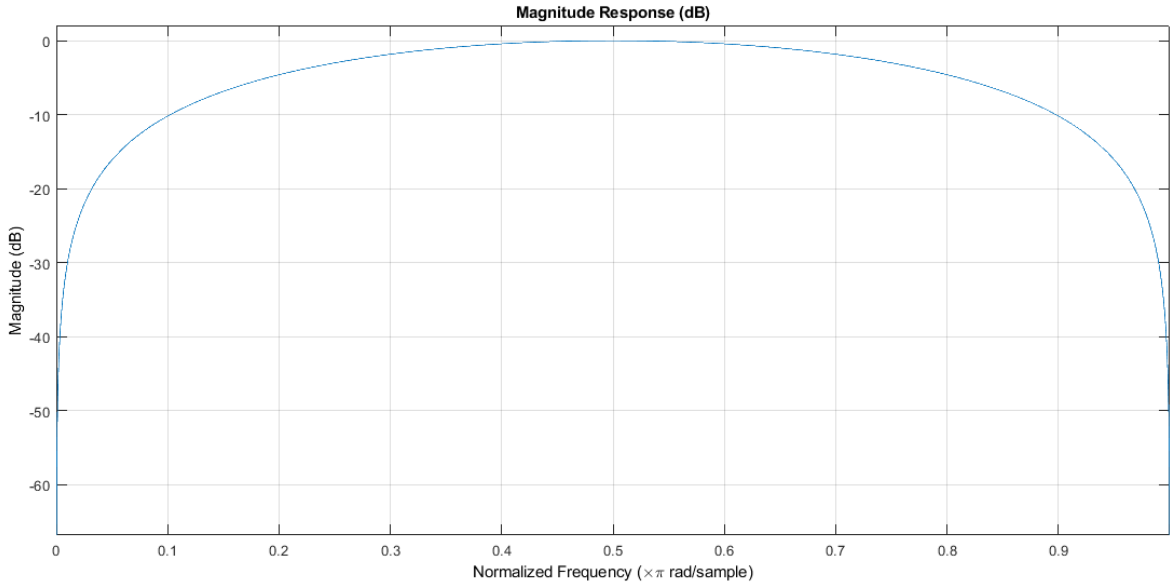

Figure 49: Magnitude Response (Log) of the Three Point Central Difference Derivative Filter (Unity Gain)

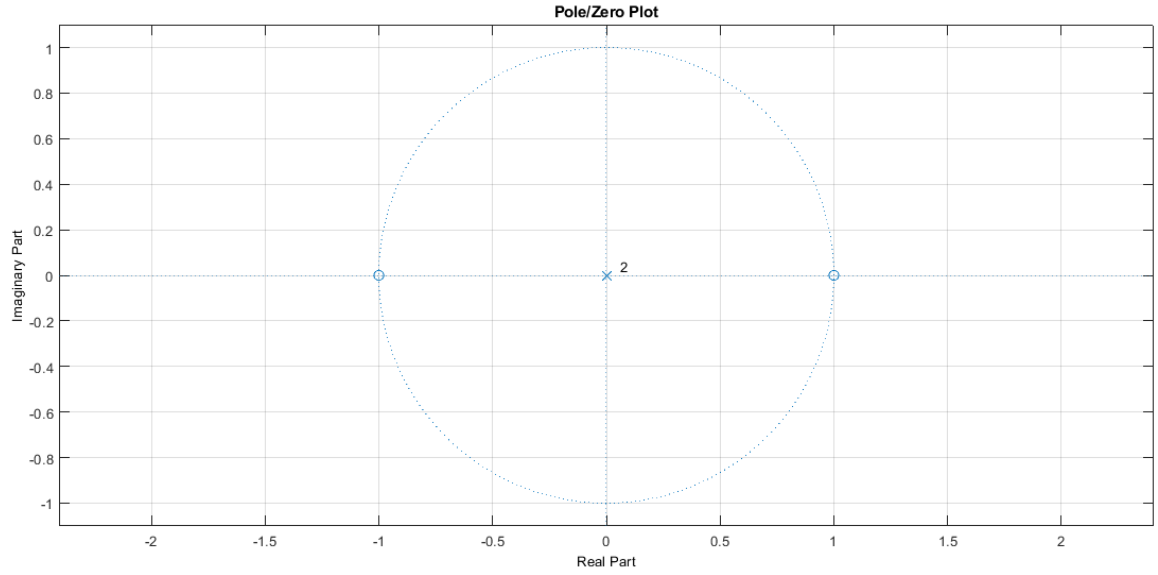

Figure 50: Pole/ Zero Plot of the Three Point Central Difference Derivative Filter (Unity Gain)

3.2 FIR Derivative Filter Application

3.2.1 Clearing the Workspace and Loading the ECG_rec.mat

The workspace variables were cleared using the 'clear all' command and the command window was cleaned using the 'clc' command. 'Load' command of MATLAB was used for loading ECG_rec.mat file.

3.2.2 Adding Noise Components

The following noise components were added to the recording.

- AWGN: 10dB
- Muscle Artifacts: $\text{EMG(t)} = 2\sin\left(\frac{2\pi t}{4}\right) + 3\sin\left(\frac{2\pi t}{2} + \frac{\pi}{4}\right)$

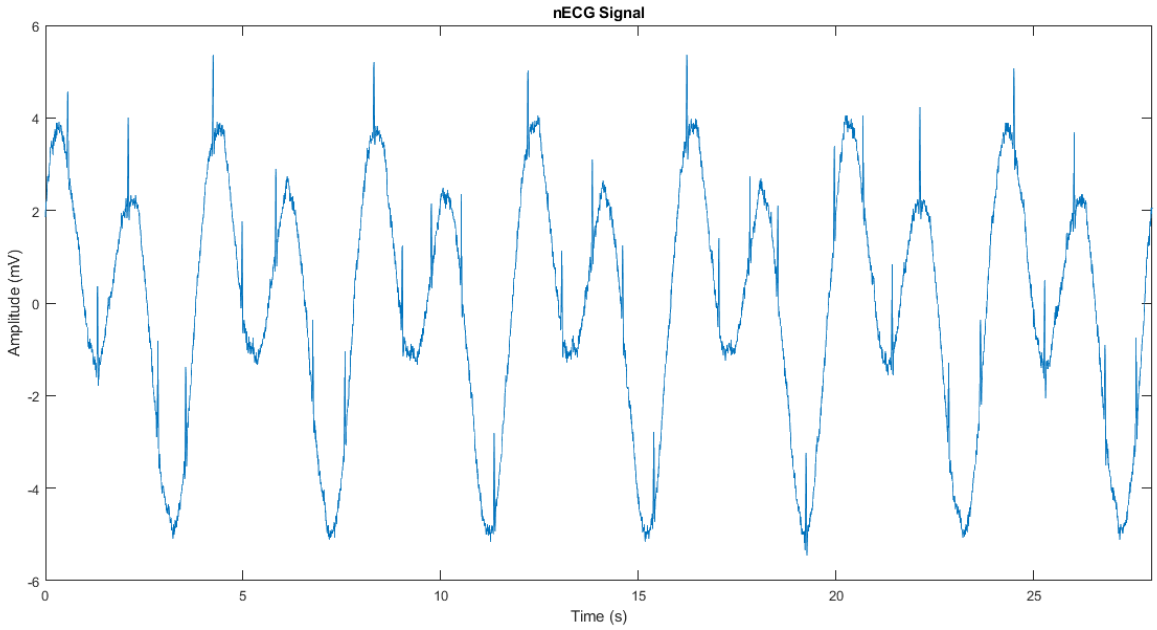

Figure 51: nECG Signal

To get a better insight into what AWGN and muscle artifacts have done to the original ECG signal, let's consider the following plot.

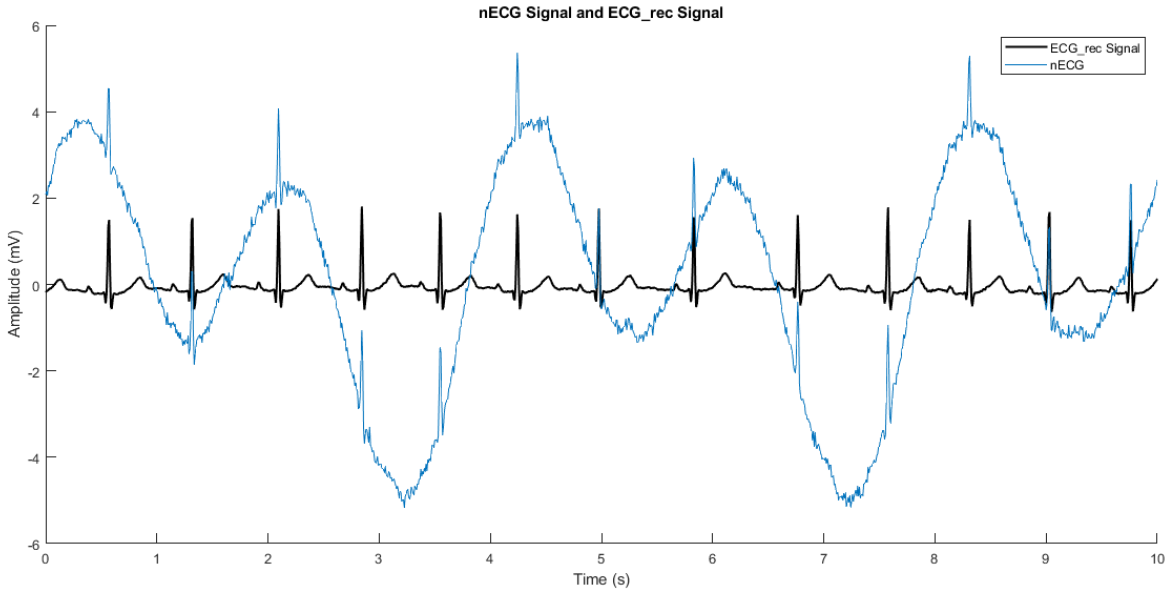

Figure 52: Comparison between nECG Signal and ECG_rec Signal

3.2.3 Applying Derivative Filters to nECG

For the first order filter, a multiplying factor of T/2 was used and for the three-point difference derivative filter, a factor of T was used to make maximum gain 1.

The filtering is implemented in the code.

3.2.4 Plotting the Filtered Signals

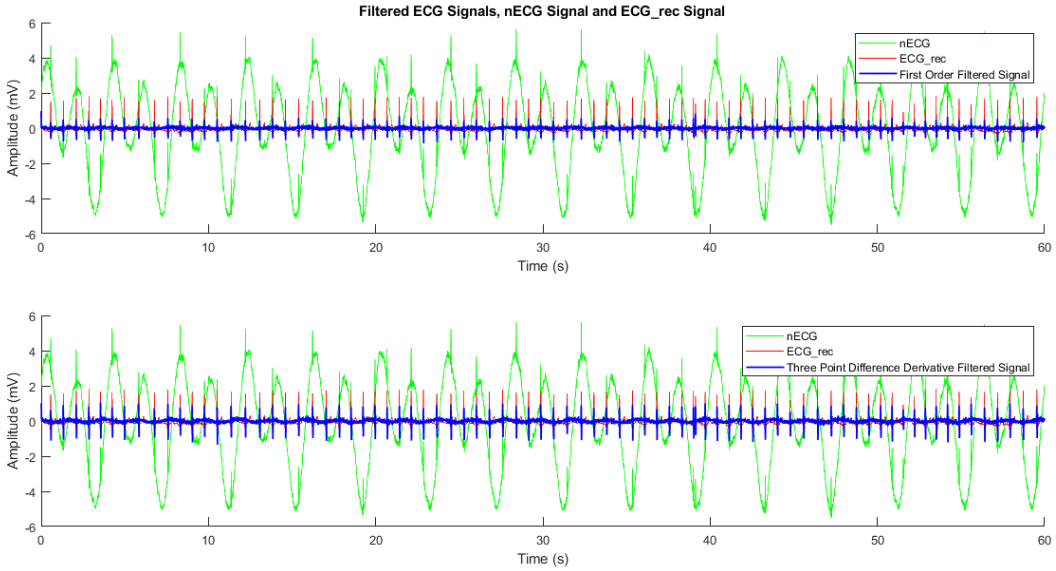

Figure 53: Comparison between Filtered ECG Signals, nECG Signal and ECG_rec Signal

Let's zoom in notice the effects of filtering.

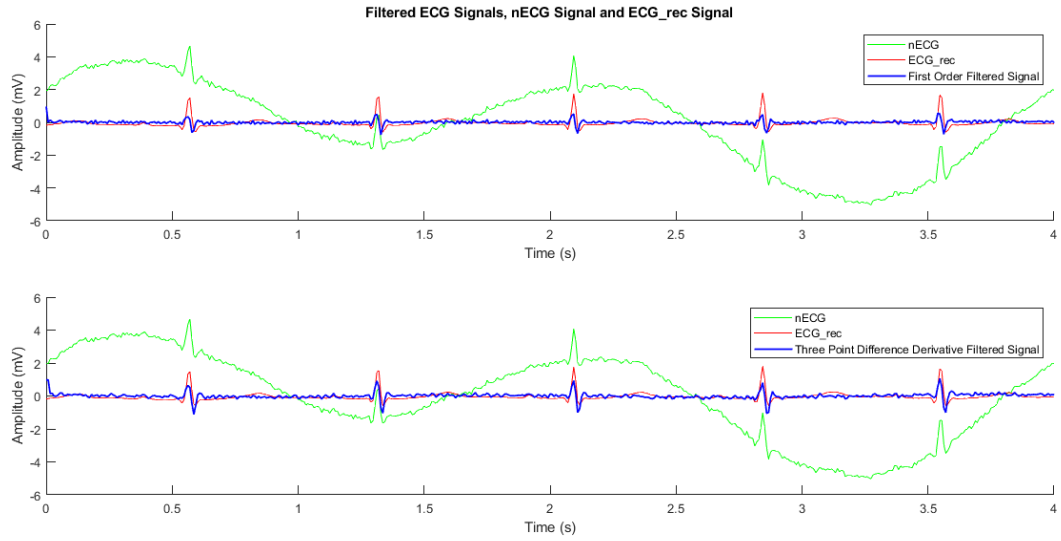

Figure 54: Comparison between Filtered ECG Signals, nECG Signal and ECG_rec Signal (Zoomed-in)

Let's plot the original ECG signal and the filtered nECG signals on the same plot.

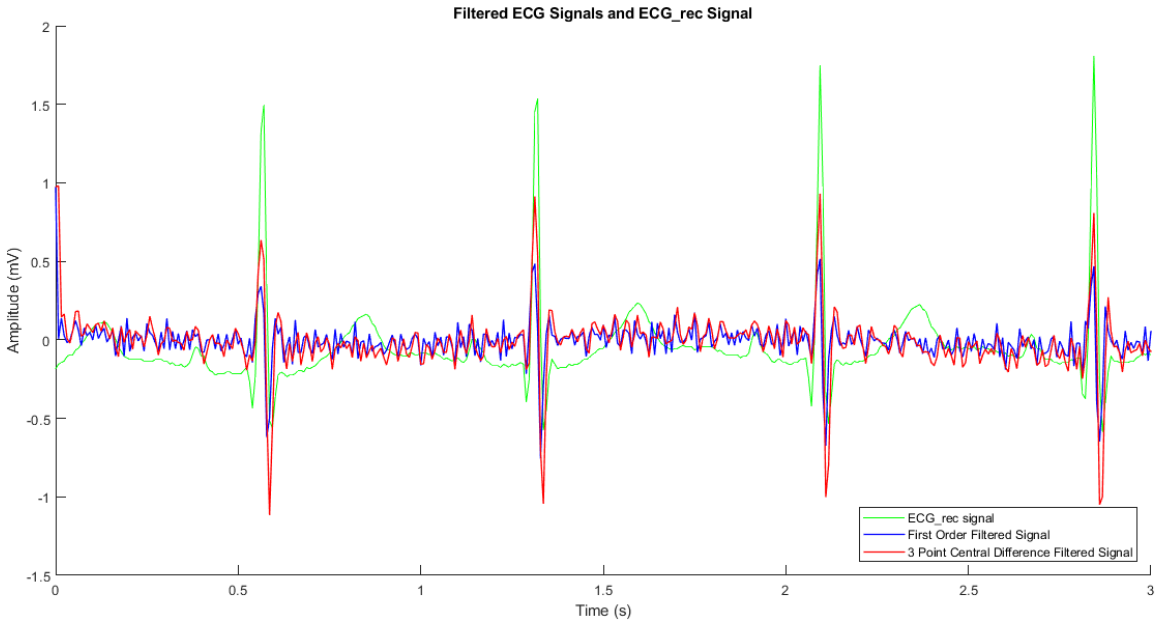

Figure 55: Comparison between Noise Free ECG Signal and the Filtered ECG Signals(Zoomed-in)

From Fig.55, it is evident that both filters have managed to get rid of the noisy components while preserving the trace of the initial noise-less ECG signal. However, both filtered signals exhibit a jitter throughout. Moreover, those filters have failed to grasp the subtle variation of amplitudes at the peaks and troughs of the QRS complexes.

4 Designing FIR Filters using Windows

4.1 Characteristics of Window Functions

4.1.1 Effect of the Length of the Window Function

The desired ideal low pass filter is,

$$H_d(\omega) = \begin{cases} 1 & |\omega| \leq \omega_c \\ 0 & \omega_c < |\omega| < \pi \end{cases}$$

That is, the impulse response of an ideal LPF would be infinite in duration.

$$h_d(n) = \begin{cases} \sin\left(\frac{\omega_c n}{\pi n}\right) & n \neq 0 \\ \frac{\omega_c}{\pi} & n = 0 \end{cases}$$

Since it's not possible to have infinite durations in real world applications, a truncated version with n data points where n = M + 1 with M being the order of the filter, is selected. Accordingly, the higher the order of the rectangular window, the closer will the filter's behaviour to the ideal filter behaviour. Thus, the filtering effect would be much sharper when the order of the rectangular window is high enough. The following plot includes the impulse response generated for rectangular windows of order (M) 5, 50, 100 with 0.4π being the normalized cutoff frequency.

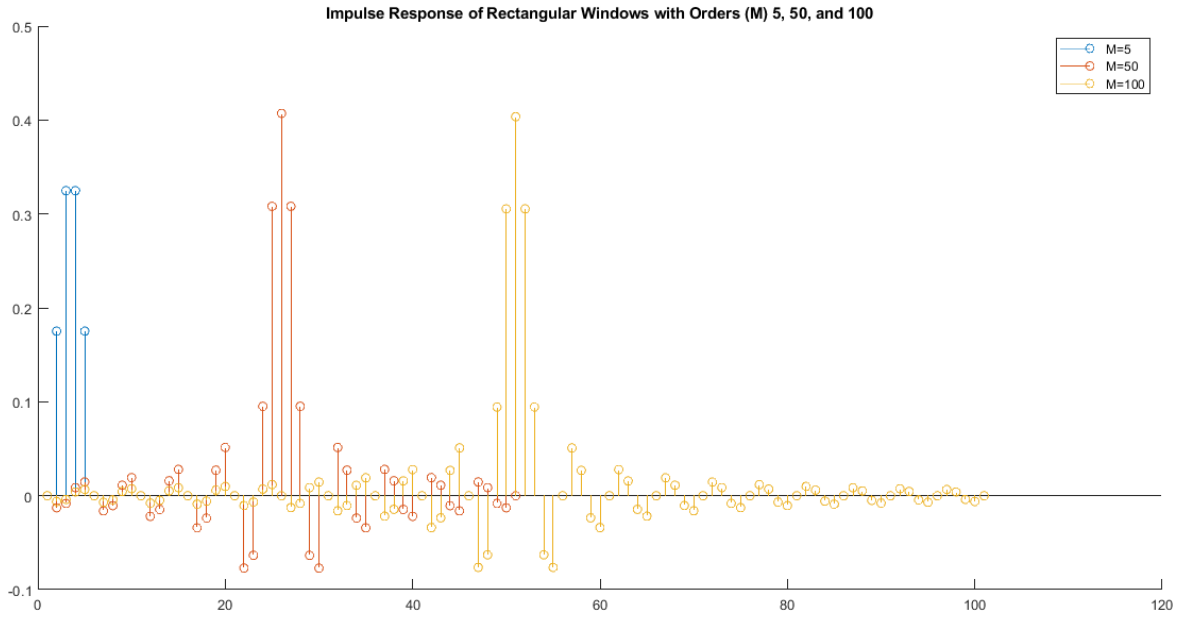

Figure 56: Impulse Response for the Rectangular Windows of Orders 5, 50, and 100

4.1.2 Magnitude Response and Phase Response of the Rectangular Windows

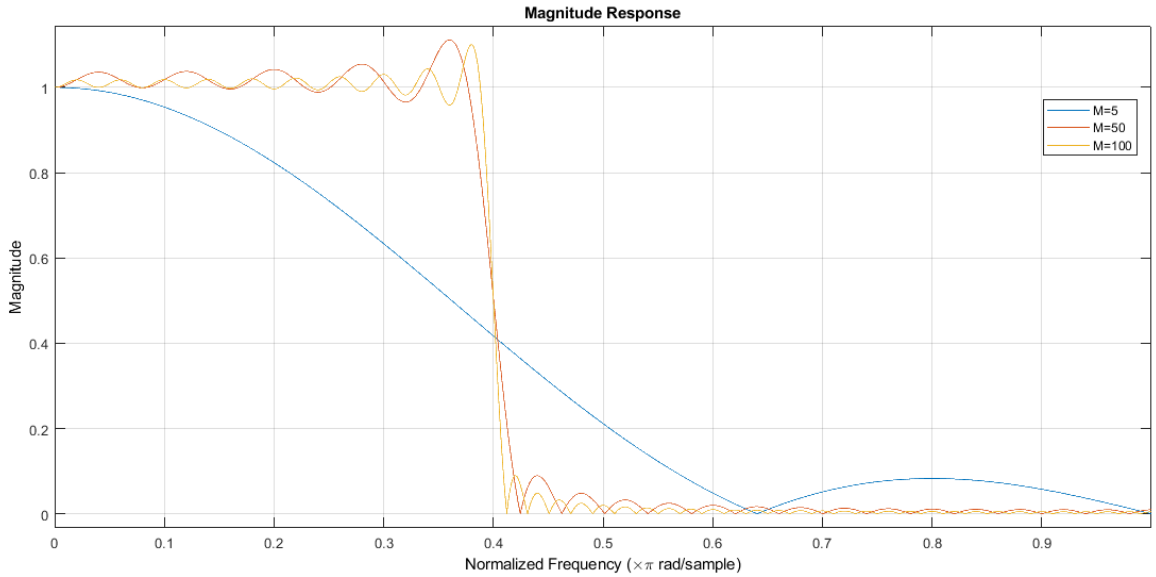

Figure 57: Magnitude Response (Linear) of the Rectangular Windows of Order 5, 50, and 100

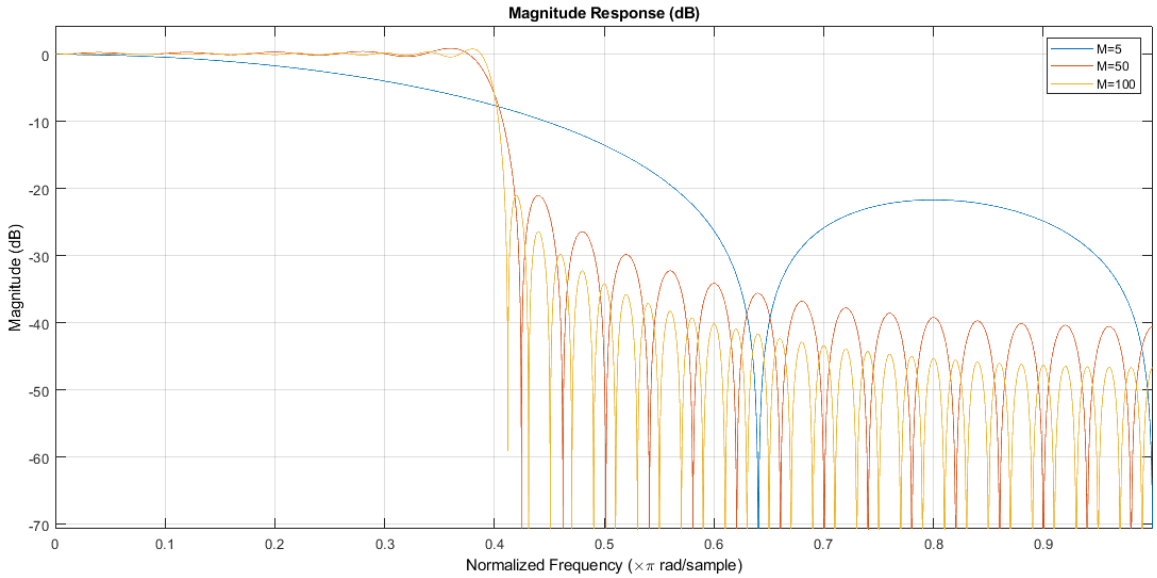

Figure 58: Magnitude Response (Log) of the Rectangular Windows of Order 5, 50, and 100

Based on the observations from Fig.57, we can draw out the following relations between the magnitude response and the order of the window. That is, with increasing order,

- The transition from passband to stopband becomes sharper
- No. of ripples in the passband and stopband increases
- The amplitudes of the ripples decrease except for the last ripple in the passband and the first ripple in the stopband
- Attenuation of the frequencies beyond the cut-off increases

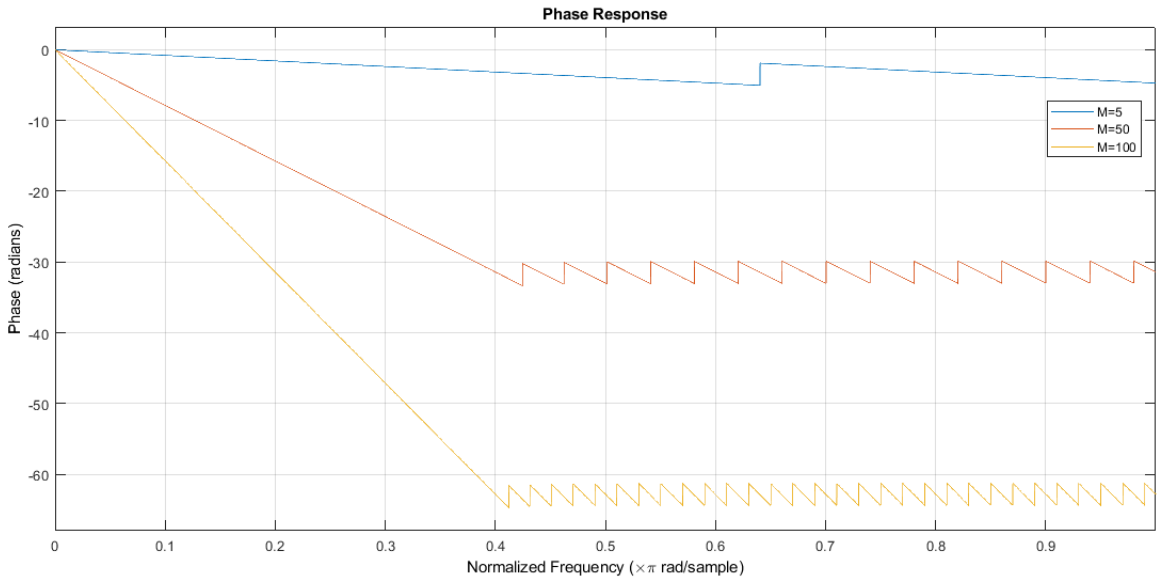

Figure 59: Phase Response of the Rectangular Windows of Order 5, 50, and 100

4.1.3 Comparative Characteristics of Window Functions: Rectangular, Hanning, Hamming, Blackman

Let's consider the following rectangular functions.

$$\text{Rectangular } w(n) = \begin{cases} 1 & |n| \leq \frac{M}{2} \\ 0 & otherwise \end{cases}$$

$$\text{Hanning } w(n) = \begin{cases} 0.5 - 0.5\cos\left(\frac{2\pi n}{M}\right) & 0 \leq n \leq M \\ 0 & otherwise \end{cases}$$

$$\text{Hamming } w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{M}\right) & 0 \leq n \leq M \\ 0 & otherwise \end{cases}$$

$$\text{Blackman } w(n) = \begin{cases} 0.42 - 0.5\cos\left(\frac{2\pi n}{M}\right) + 0.08\cos\left(\frac{4\pi n}{M}\right) & 0 \leq n \leq M \\ 0 & otherwise \end{cases}$$

Regardless of how high the order of the rectangular window is, such filters would always result in discontinuities in the time domain.

To eliminate discontinuities in time domain, various functions have been introduced to be used as the window function.

In the subsequent plots, the following aspects of these windows are discussed.

- Morphology of the windows
- Magnitude response in the linear domain
- Magnitude response in the log domain
- Phase response

4.1.3.1 Morphology of the Windows

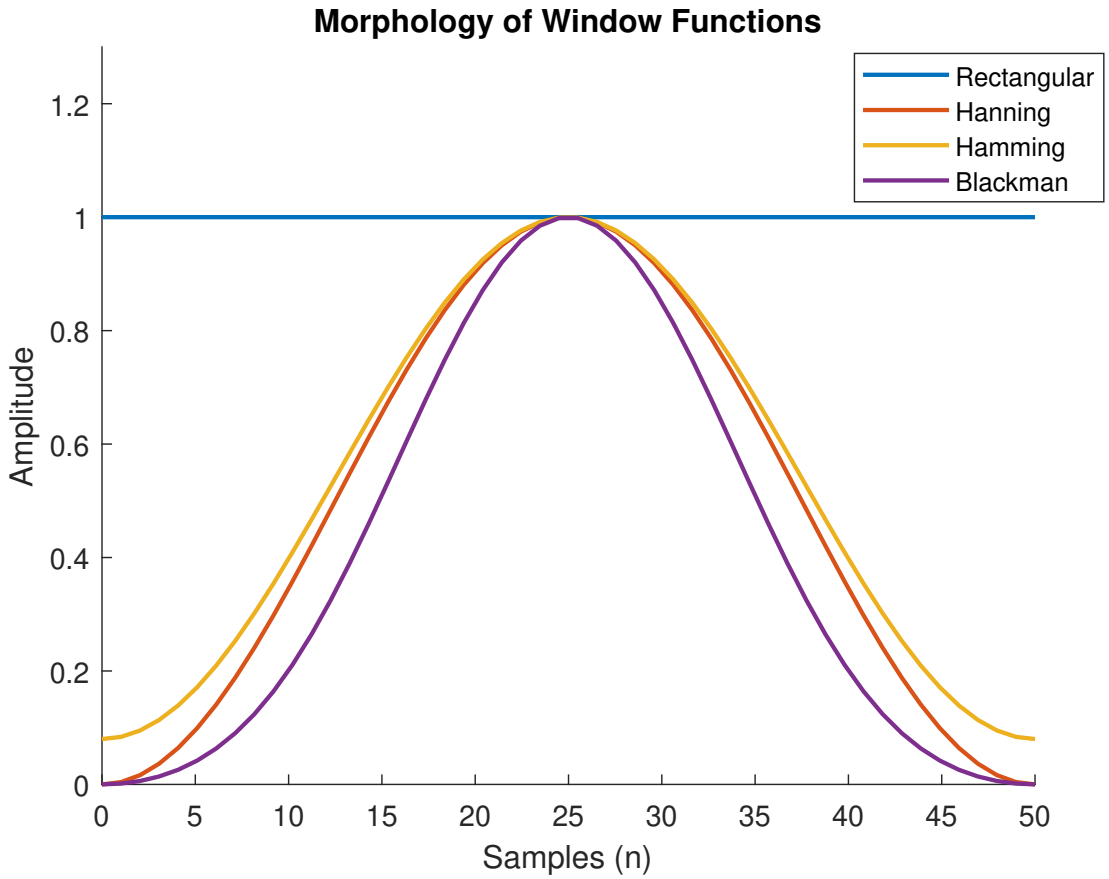

Figure 60: Morphology of the Window Functions

At the extremities of the rectangular window, there are sharp transitions which cause the filter to have discontinuities in the time domain. Comparatively, other filer windows eliminate this limitation by smoothly transitioning out of the window.

4.1.3.2 Magnitude Response of the Windows

We can observe from Fig.62 that the transition from the passband to the stopband is really smooth in Hanning, Hamming and Blackman windows. Moreover, the passband and stopband ripples are minimal in those responses (in contrast to that of the rectangular window). The transitions in the Hanning, Hamming and Blackman windows are not as sharp as the rectangular window, that is, those windows result in smaller ripples in stopband and passband at the expense of a wider transition band. Nevertheless, the half-power cut off frequency of all the windows remains the same.

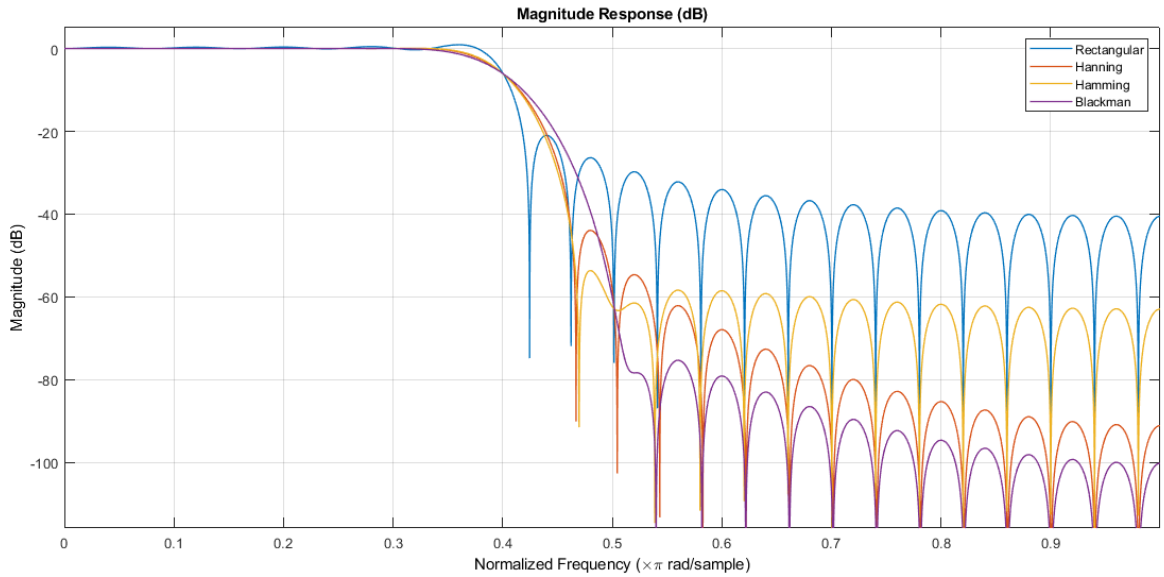

Figure 61: Magnitude Response (Linear) of the Window Functions

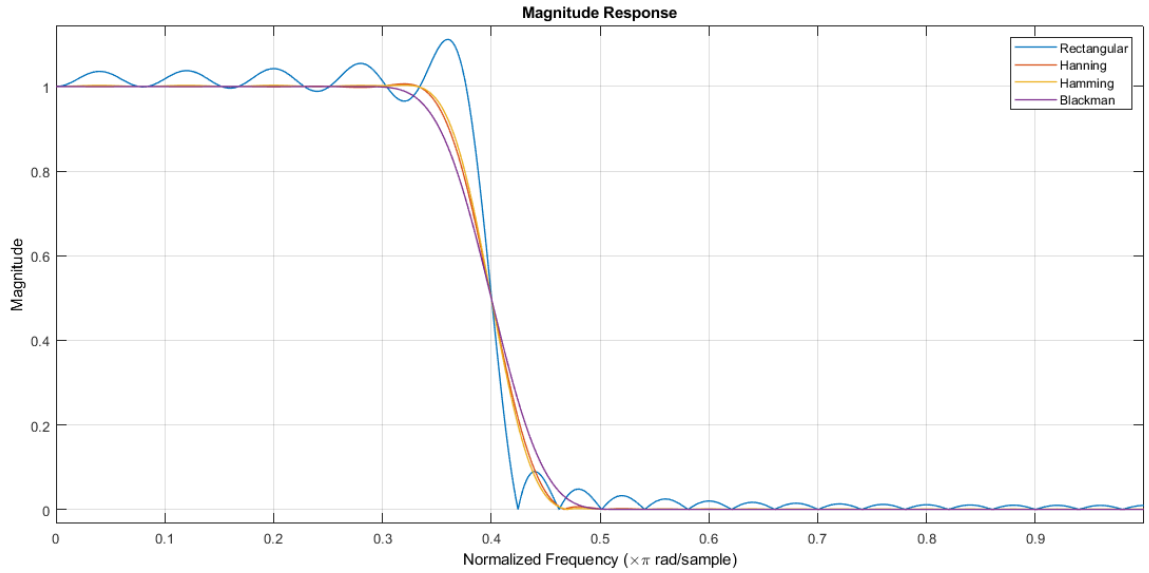

Figure 62: Magnitude Response (Log) of the Window Functions

4.1.3.3 Phase Response of the Windows

Phase responses of the window functions are summarized below in Fig.63.

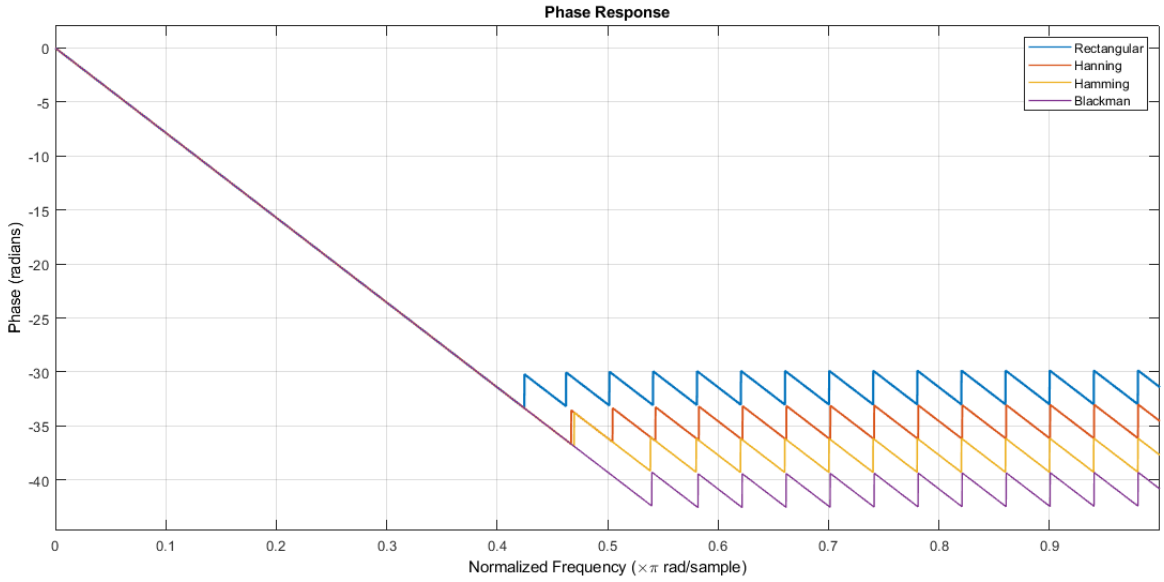

Figure 63: Phase Response of the Window Functions

4.2 FIR Filter Design and Application using the Kaiser Window

The Kaiser window is a generic function which can approximate a variety of windows by varying the shaping parameter β and the window length M. A low pass filter and a high pass filer will be implemented using Kaiser windows and FIR Comb filters. A noisy ECG signal with the following properties would be used.

Name of the recording	ECG_with_noise.mat
Recorded lead	II
Sampling frequency	500Hz
Amplitude range	mV

Table 4: Properties of the Noisy ECG Signal

4.2.1 Plotting the Time Domain Signal and Power Spectral Density

For a better observation, let's zoom in on the Fig.64 to obtain Fig. 65. As we can see, the noise seems to be overpoweing the actual signal as we can't distinguish the key features of a noise-free ECG, in this noisy ECG signal.

In Fig. 66, the PSD estimates of a noise-free ECG signal is provided along with that of the noisy ECG. It is evident that the noisy ECG signal carries a plethora of high frequency components with large amplitudes. Moreover, notice the spikes at 50Hz, 100Hz and 150Hz of the PSD of the noisy ECG which could potentially be corresponding to the powerline noise (i.e. 50Hz).

On the other hand, a normal ECG signal has most of its power concentrated in the low frequency region with high frequencies having relatively lower amplitudes.

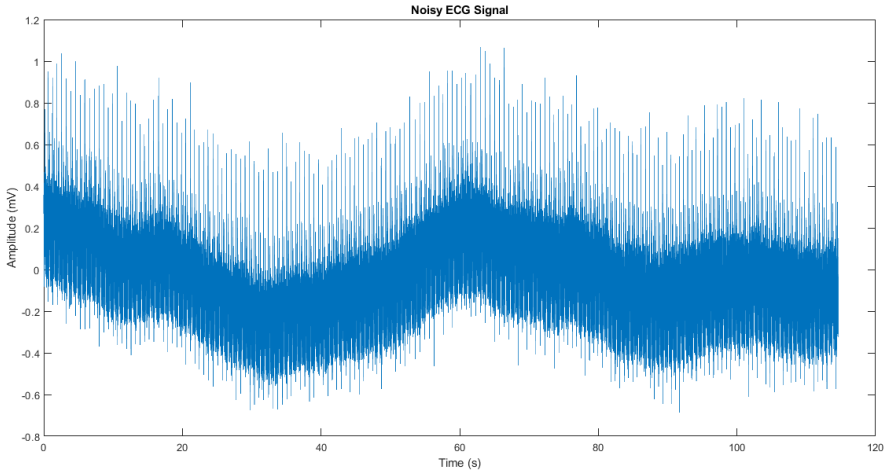

Figure 64: Noisy ECG Signal

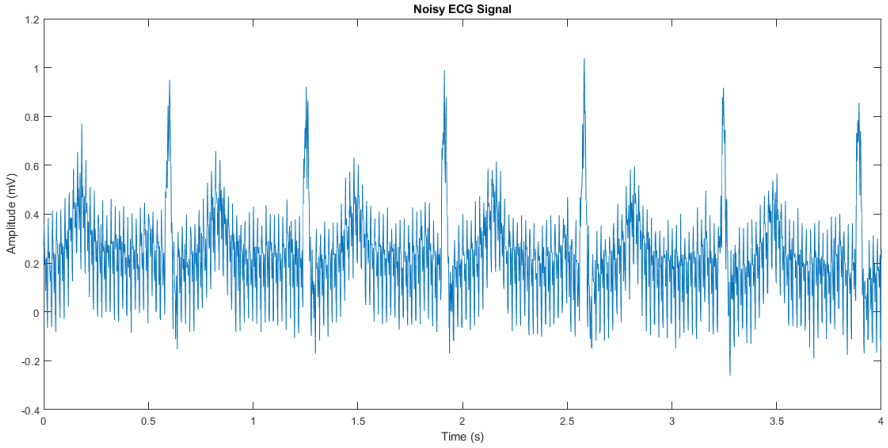

Figure 65: Noisy ECG Signal (Zoomed-in)

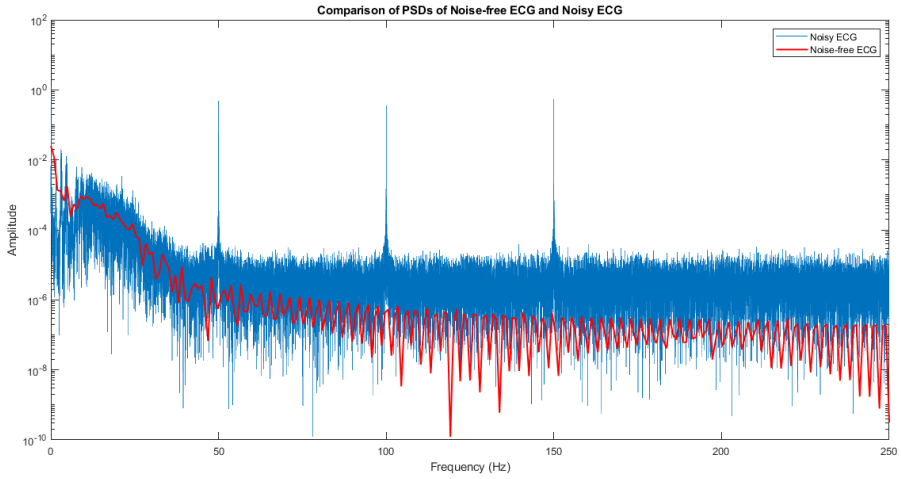

Figure 66: PSD Estimate of the Noisy ECG Signal and Noise-free ECG Signal

4.2.2 Deciding the Parameters of the Filters to be used

The following filters will be used to suppress the noise components.

- Low pass filter: Eliminate high frequency noise components
- High pass filter: Eliminate low frequency noise components
- Comb filters: Eliminate spikes at 50Hz,100Hz and 150Hz

It was given the high pass cut off frequency is 5Hz and the low pass cut off frequency is 125Hz.

	High Pass	Low Pass
f_{pass}	8	122
f_c	5	125
f_{stop}	2	128
δ	0.001	

Table 5: Properties of the High Pass and Low Pass Filter

Comb Filter	
f_{stop_1}	50
f_{stop_2}	100
f_{stop_3}	150

Table 6: Properties of the Comb Filters

A sufficiently wide transition band needs to be there and a transition width of 6Hz was allocated to the LPF and the HPF. Moreover, the peak approximation error (δ) is assumed to be 0.001.

In the PSD estimate, we observed spikes at 50Hz, 100Hz and 150Hz. Accordingly, comb filters will be applied at those frequencies to suppress their impact.

4.2.3 Calculating β and M Values for the LPF and HPF

The following parameters are required to be defined to use the Kaiser Window.

- β: Shape parameter
- M: Order of the filter

$$
\begin{aligned}
A &= -20log_{10}\delta \\
&= -20log_{10}0.001 \\
A &= 60 \\
\Delta\omega &= 2\pi\frac{|f_{stop} - f_{pass}|}{f_s} \\
&= \frac{12\pi}{500} \\
\Delta\omega &= 0.024\pi \qquad (0.0754)
\end{aligned}
$$

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

$$\beta = 0.1102(60 - 8.7)$$

$$\beta = 5.65326$$

$$M = \left\lceil \frac{A-8}{2.285\Delta\omega} \right\rceil = \left\lceil \frac{60-8}{2.285 \times 0.024\pi} \right\rceil$$

$$M = \lceil 301.8256 \rceil$$

$$M = 302$$

The calculated M value was 301.8256. However, we are allowed to make ±2 deviation to the calculated M value. For convenience, M needs to be even so that the window length is odd and symmetric. Therefore, let's take M = 302 to be the order of the filter.

The calculated filter parameters for the HPF and LPF are summarized in the following table.

A	60
$\Delta\omega$	0.024π
β	5.65326
M	302

Table 7: Calculated Filter Parameters for LPF and HPF

4.2.4 Visualizing the Magnitude Response and Phase Response of the Windows

4.2.4.1 Magnitude Response and Phase Response of the Kaiser LPF and HPF

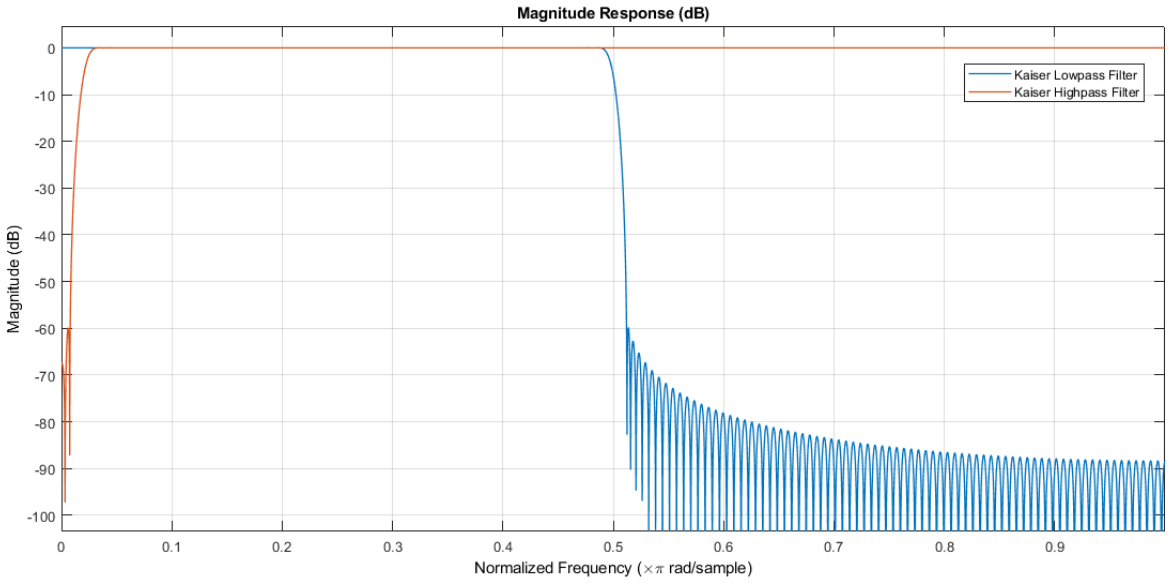

Figure 67: Magnitude Response (Log) of the Kaiser LPF and HPF

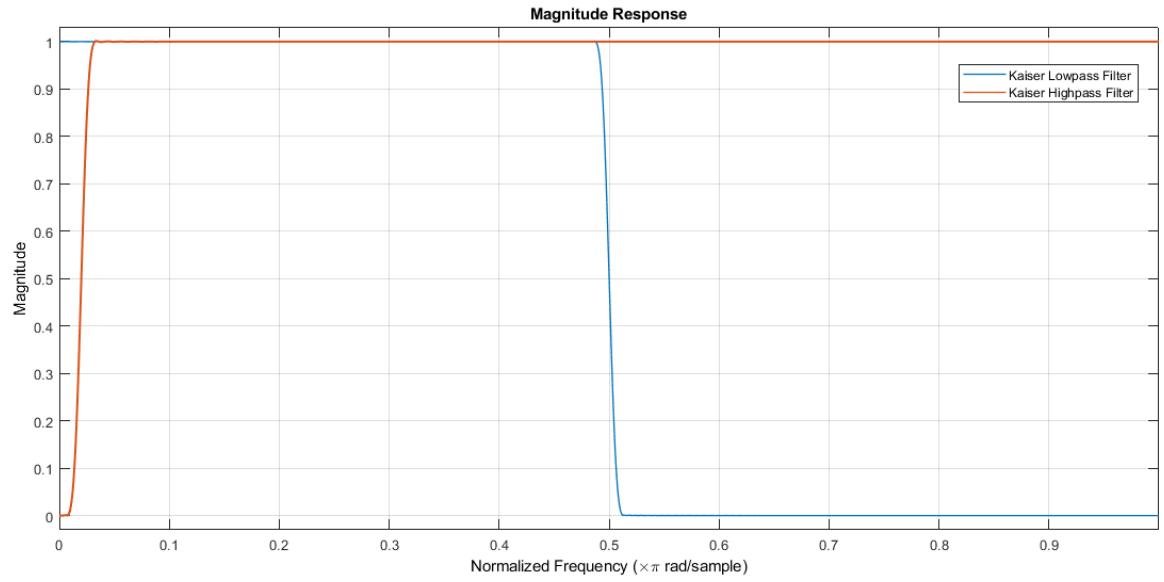

Figure 68: Magnitude Response (Linear) of the Kaiser LPF and HPF

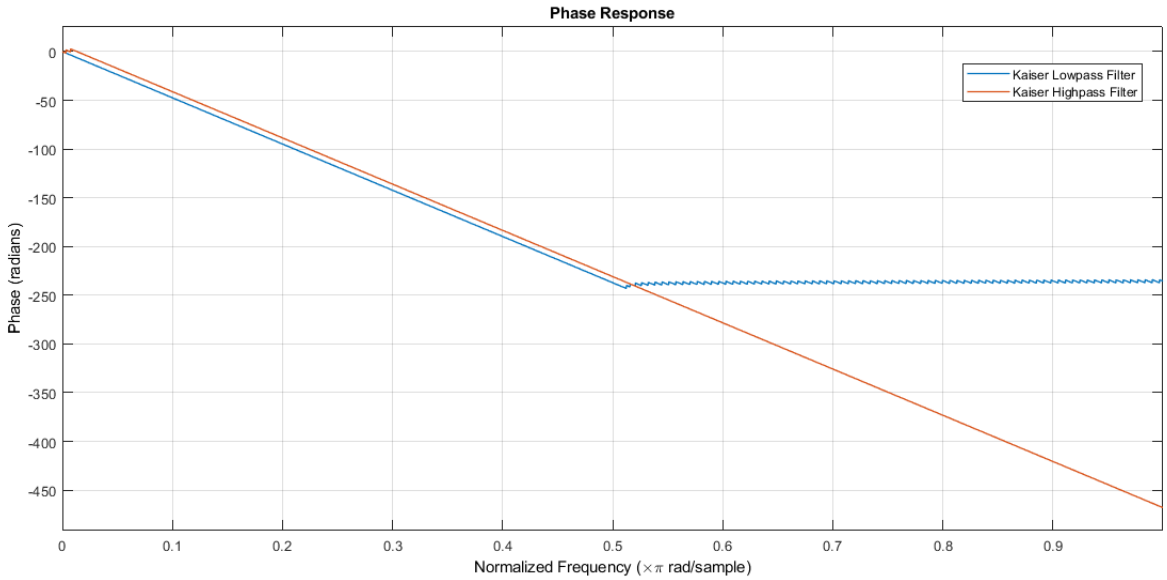

Figure 69: Phase Response of the Kaiser LPF and HPF

4.2.4.2 Magnitude Response and Phase Response of the Comb Filter

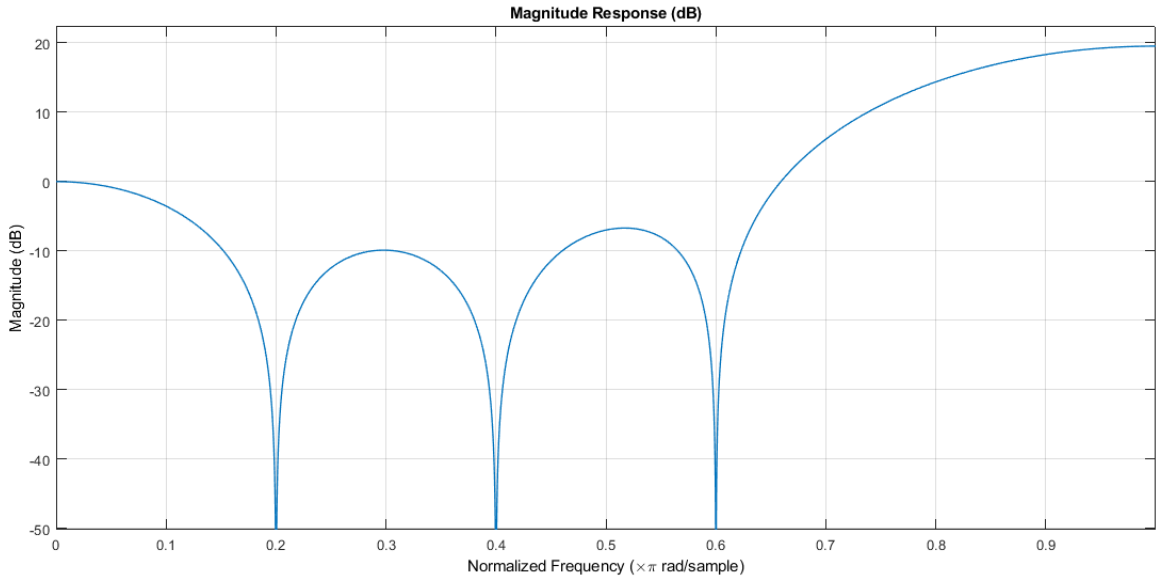

Figure 70: Magnitude Response (Log) of the Comb Filter

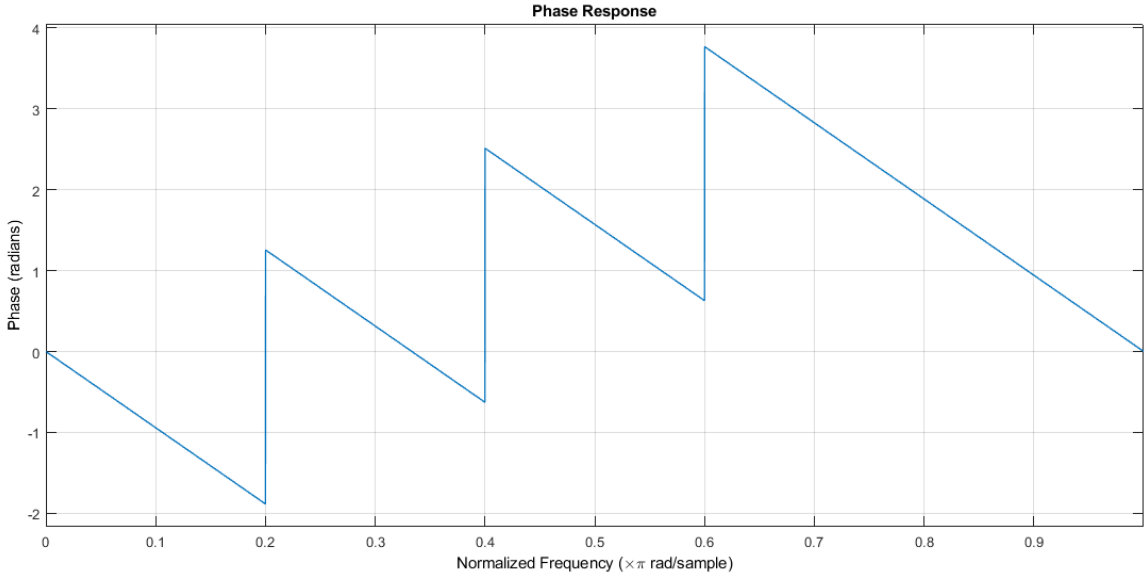

Figure 71: Phase Response of the Comb Filter

4.2.5 Filtering the Noisy ECG Signal using the LPF, HPF and the Comb Filter

Let's first visualize the individual effect of each filter; LPF, HPF, and the Comb filter on the noisy ECG.

Fig. 72 - Fig.77 represent nECG signal filtered by the aforementioned filters individually. Fig.78 and Fig.79 represent the nECG signal filtered by the combined filter.

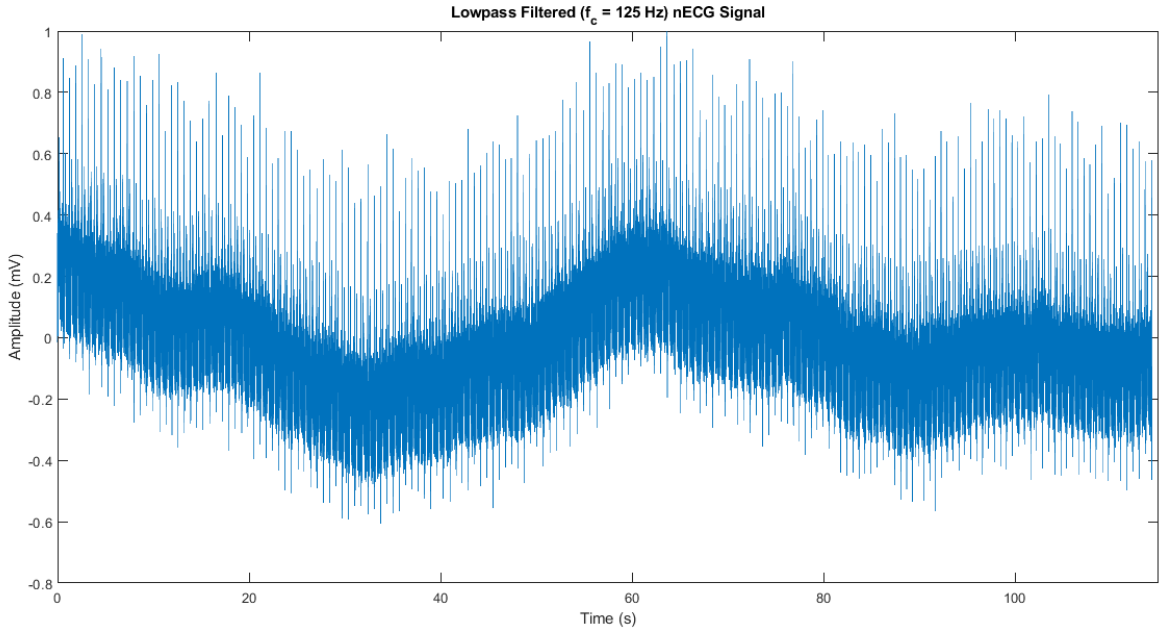

Figure 72: Low Pass Filtered nECG Signal

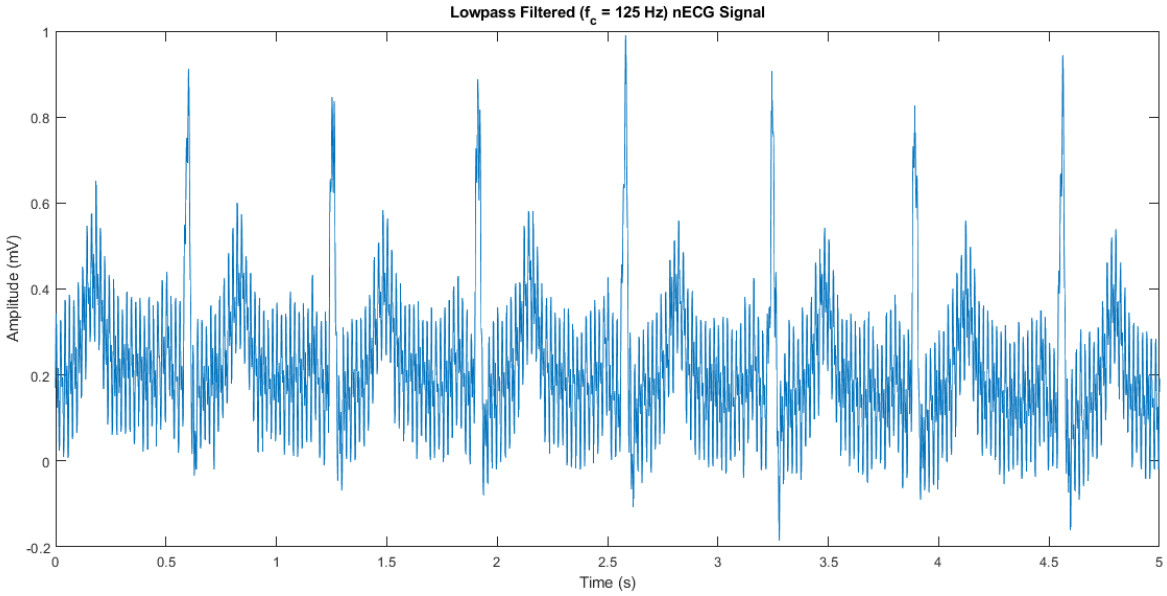

Figure 73: Low Pass Filtered nECG Signal (Zoomed in)

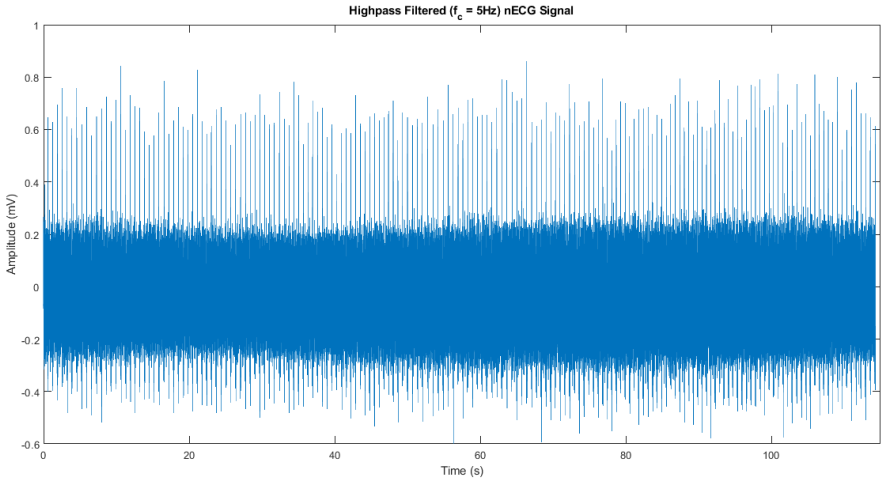

Figure 74: High Pass Filtered nECG Signal

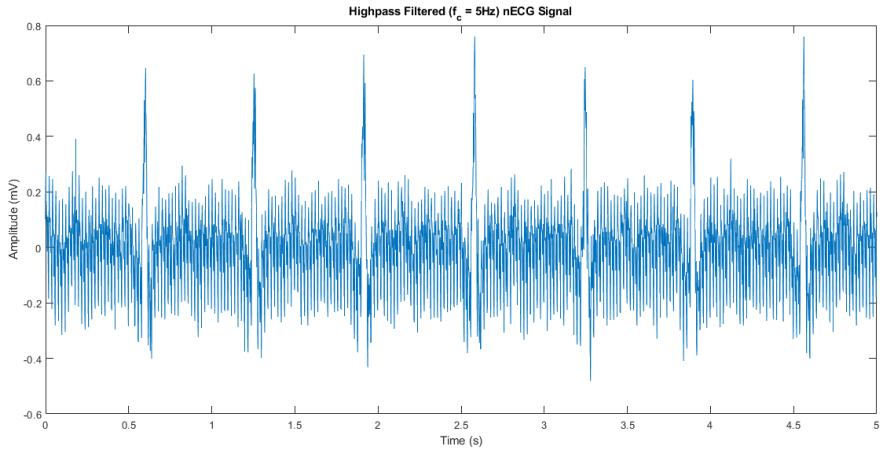

Figure 75: High Pass Filtered nECG Signal (Zoomed in)

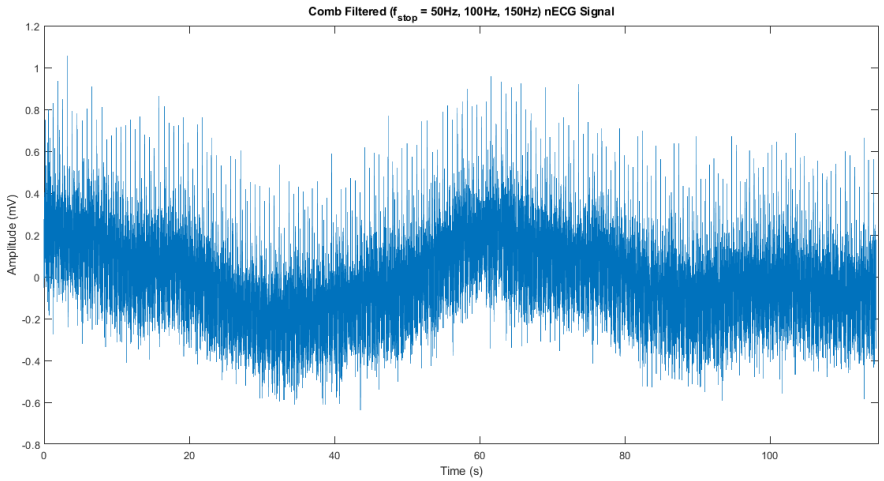

Figure 76: Comb Filtered nECG Signal

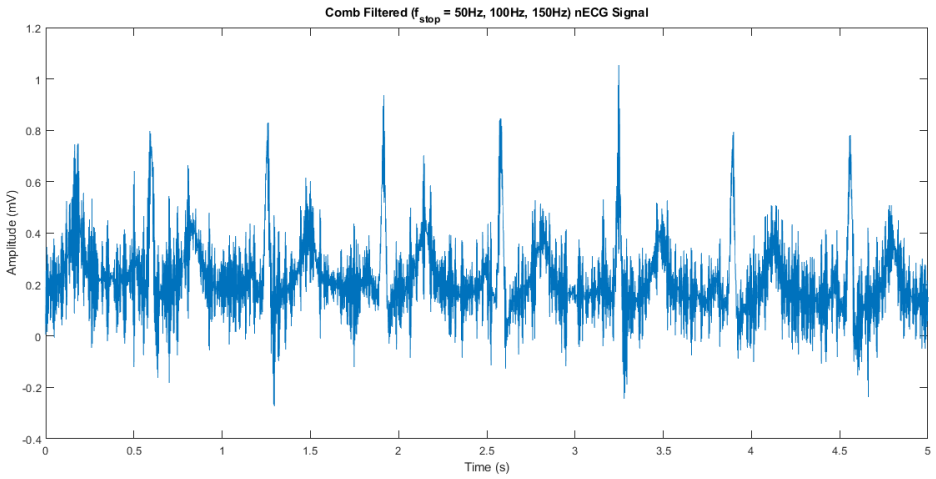

Figure 77: Comb Filtered nECG Signal (Zoomed in)

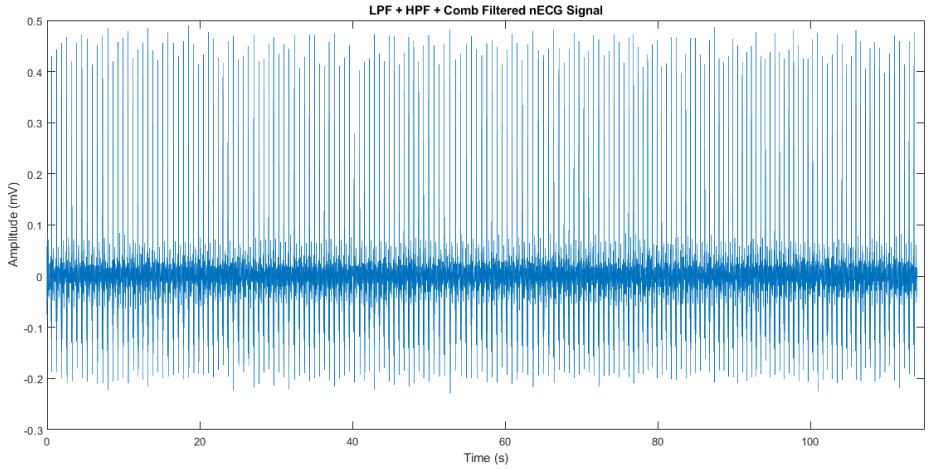

Figure 78: LPF + HPF + Comb Filtered nECG Signal

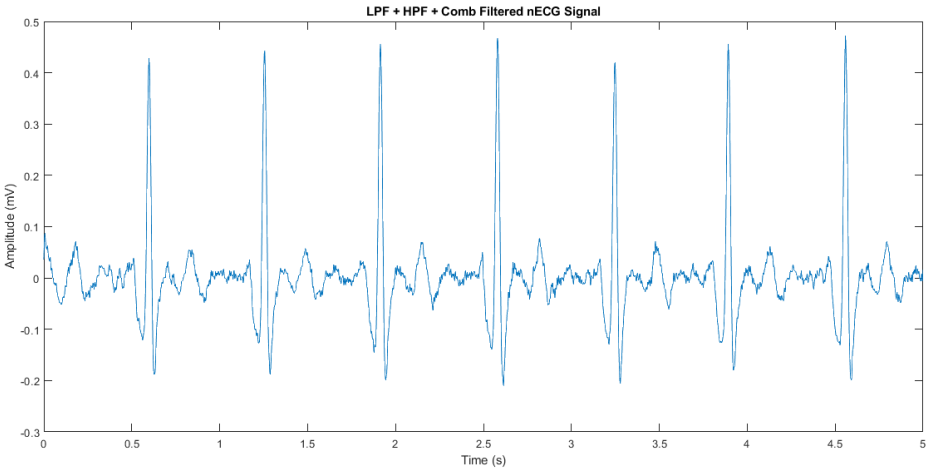

Figure 79: LPF + HPF + Combined Filtered nECG Signal (Zoomed in)

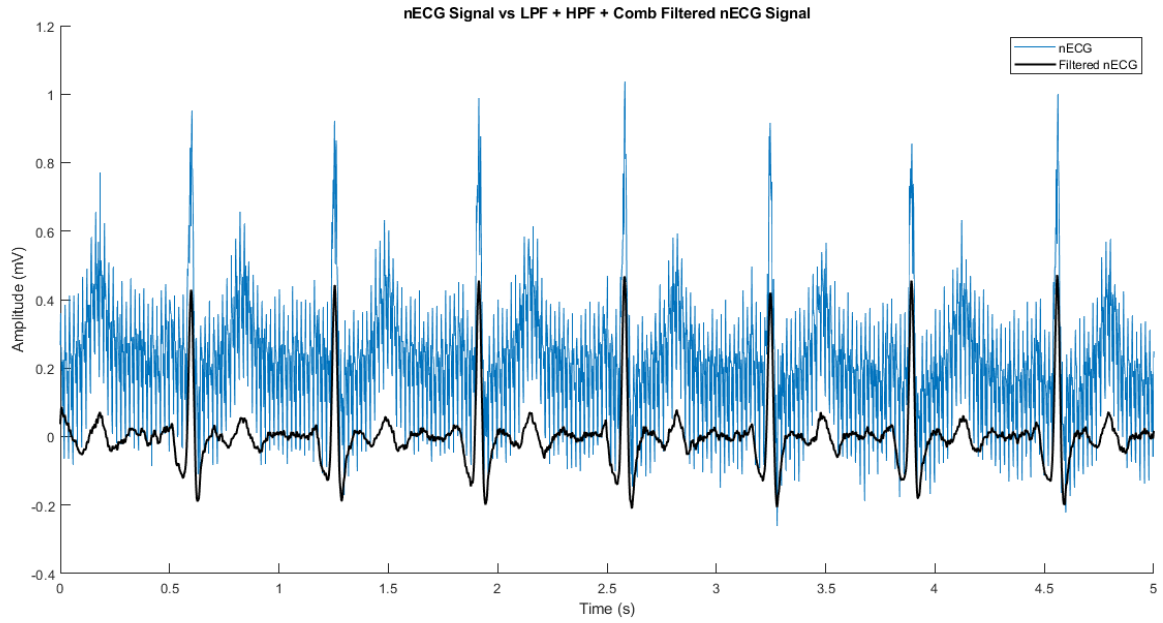

Figure 80: nECG Signal vs LPF + HPF + Combined Filtered nECG Signal (Zoomed in)

We can clearly observe from Fig.80 that the combined filters have managed to recover a significantly improved ECG signal from the noisy ECG signal. Further, we can see characteristic landmarks such as the P wave, QRS complex and the T wave in the recovered signal.

4.2.6 Visualizing the Combined Filter and the PSD of the Filtered Signal

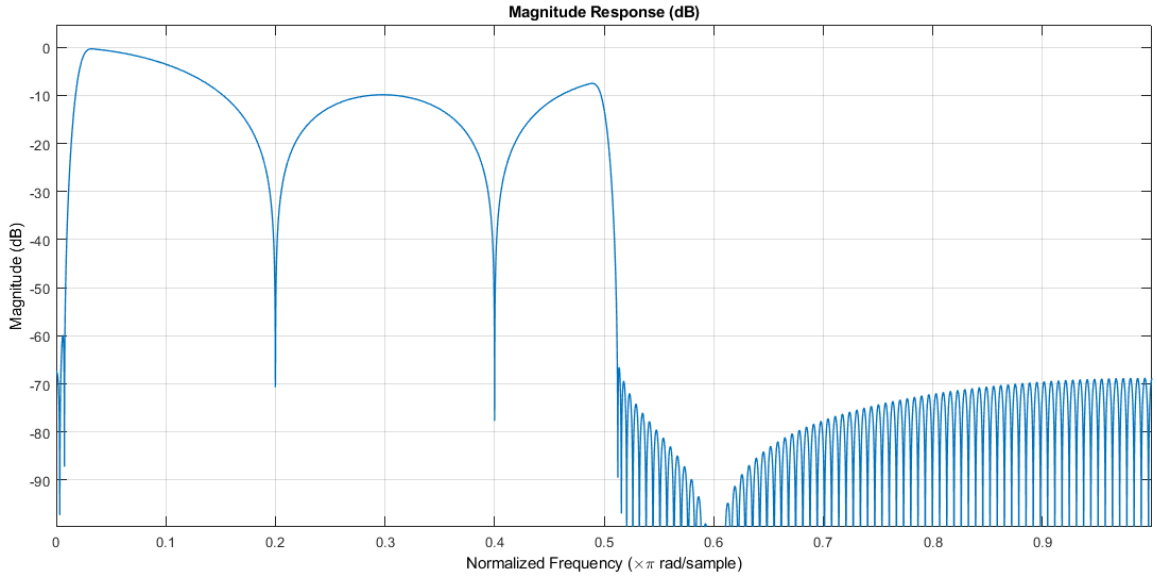

Figure 81: Magnitude Response of the Combined Filter

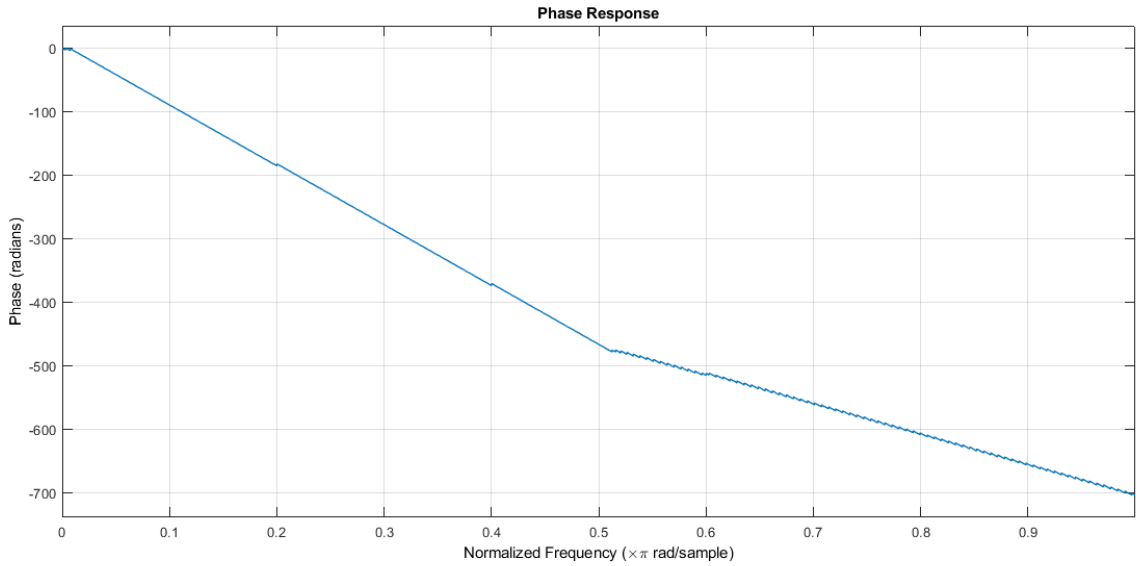

Figure 82: Phase Response of the Combined Filter

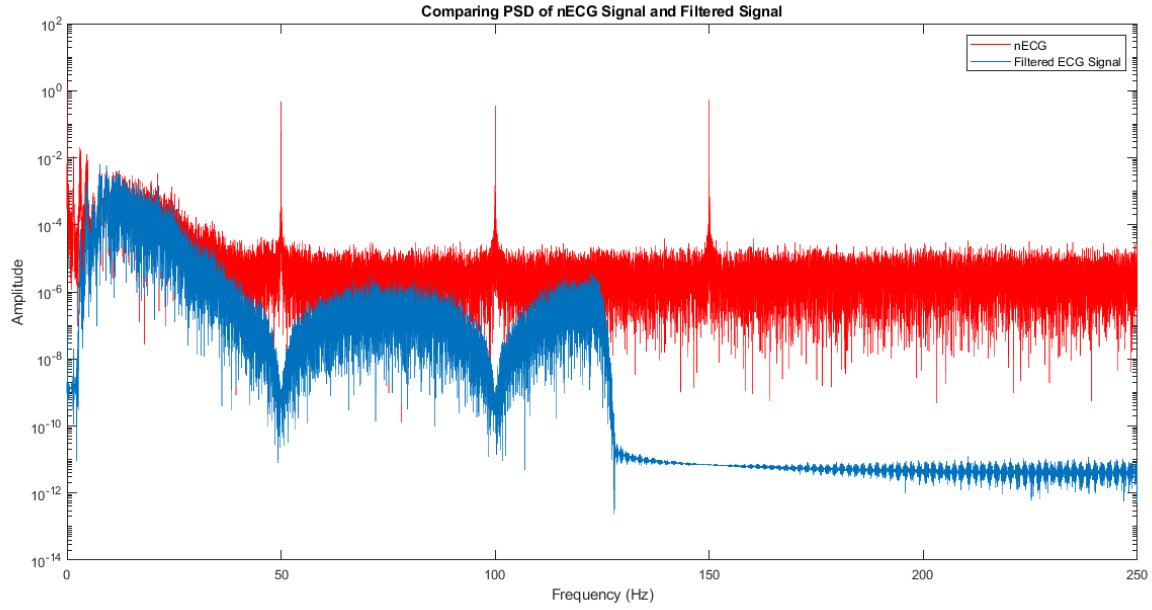

Figure 83: PSD Estimate of the nECG Signal and the Filtered Signal

We can observe from Fig.83 that the combined filter has suppressed frequencies beyond 125Hz significantly. Moreover, the Comb filter has managed to eliminate the spikes at the frequencies 50Hz, 100Hz, and 150Hz.

5 IIR Filters

5.1 Realizing IIR Filters

5.1.1 Obtaining Filter Coefficients of a Butterworth Low Pass Filter

Butterworth polynomials are a family of polynomials used in IIR filter design. Depending upon the order of the filter, we might get a unique polynomial for the relevant transfer function. The normalized Butterworth polynomials ($B_n(s)$) of order n are given by,

$$B_n(s) = \begin{cases} \prod_{k=1}^{\frac{n}{2}} \left[s^2 - 2s\cos\left(\frac{(2k+n-1)\pi}{2n}\right) + 1 \right] & n \ is \ even \\ (s+1) \prod_{k=1}^{\frac{(n-1)}{2}} \left[s^2 - 2s\cos\left(\frac{(2k+n-1)\pi}{2n}\right) + 1 \right] & n \ is \ odd \end{cases}$$

From the above equation, we can realize that as the order of the polynomial is really high, the transfer function may tend to be unstable. This could be observed in the subsequent section as we have designed a Butterworth low pass filter with order 302 and cut off frequency of 125Hz.

Filter coefficients were obtained using the command *[b,a] = butter(n, Wn).*

5.1.2 Visualizing the Butterworth Low Pass Filter

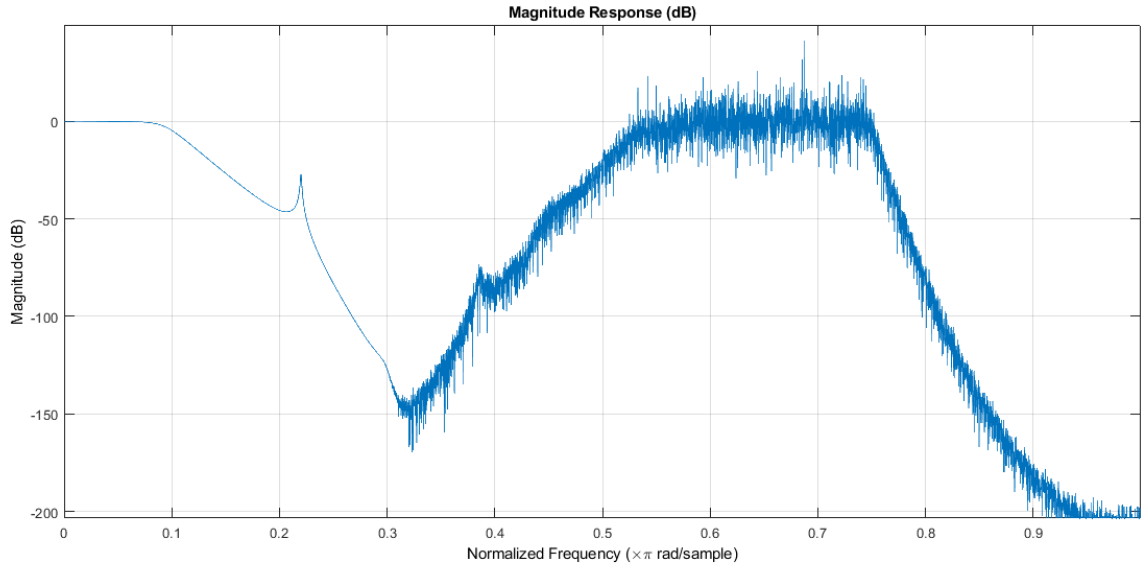

Figure 84: Magnitude Response of the Butterworth Low Pass Filter

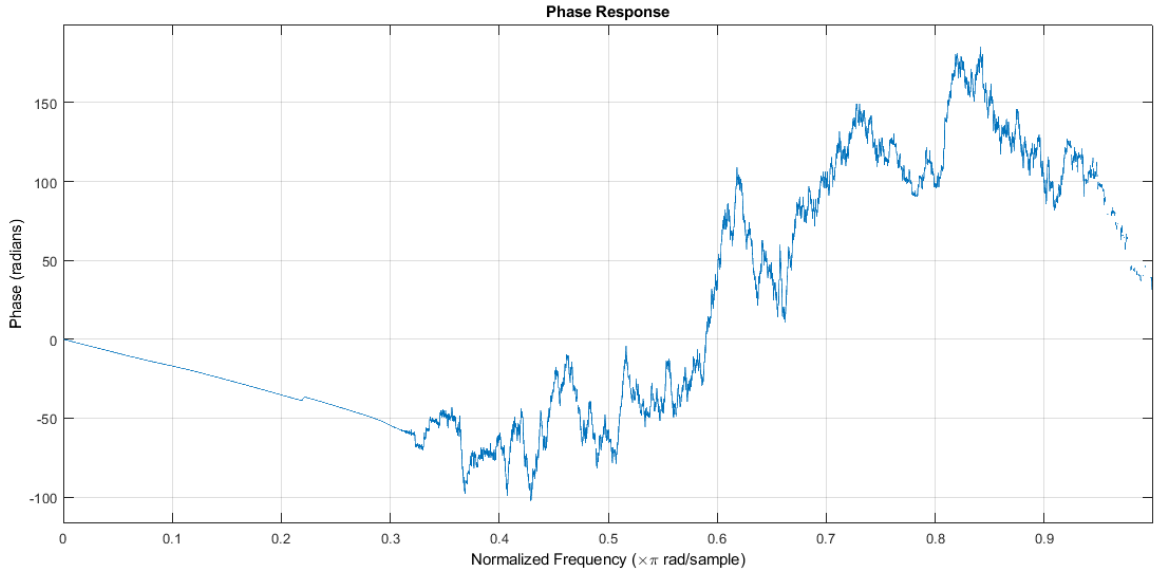

Figure 85: Phase Response of the Butterworth Low Pass Filter

Considering Fig.84 and Fig.85, it is noticeable that the filter is unstable when the order is extremely large. Moreover, notice the prominent distortions in the phase response of the filter.

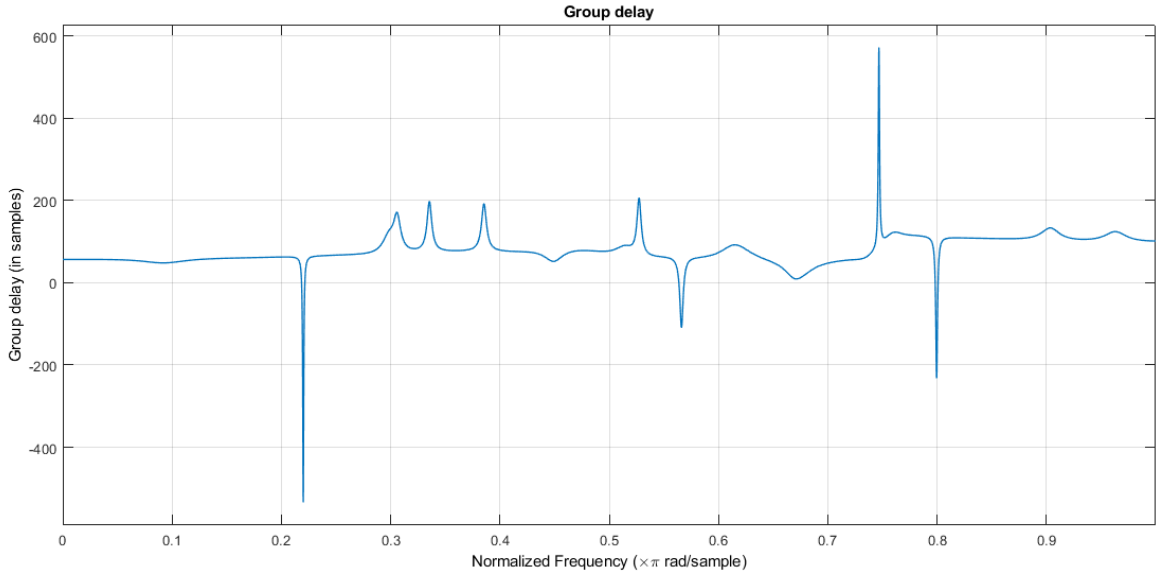

Figure 86: Group Delay of the Butterworth Low Pass Filter

Accordingly, there is no point of using an order of 302 in the Butterworth filters. Using MATLAB, several orders were experimented with and the maximum order that would result in stable magnitude and phase responses was chosen to implement the Butterworth filters in the question. The maximum order was selected to be 10. From 11 onwards, the responses are unstable.

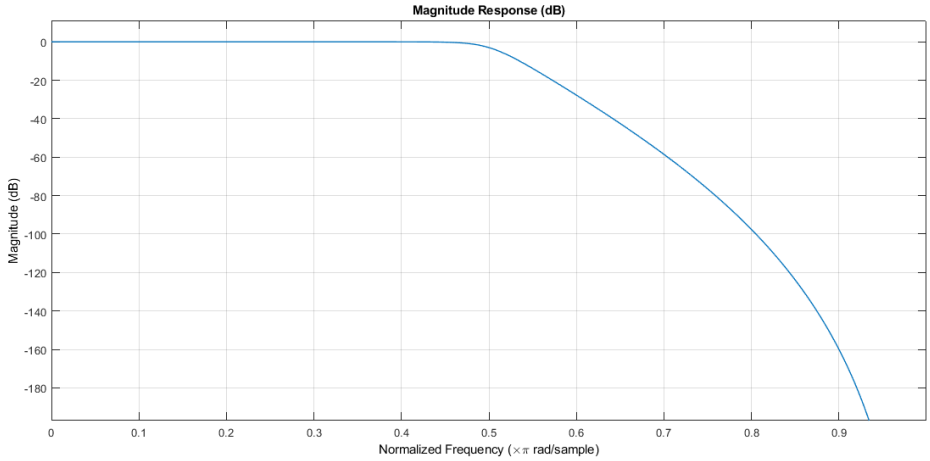

Figure 87: Magnitude Response of the Butterworth Low Pass Filter (Order 10)

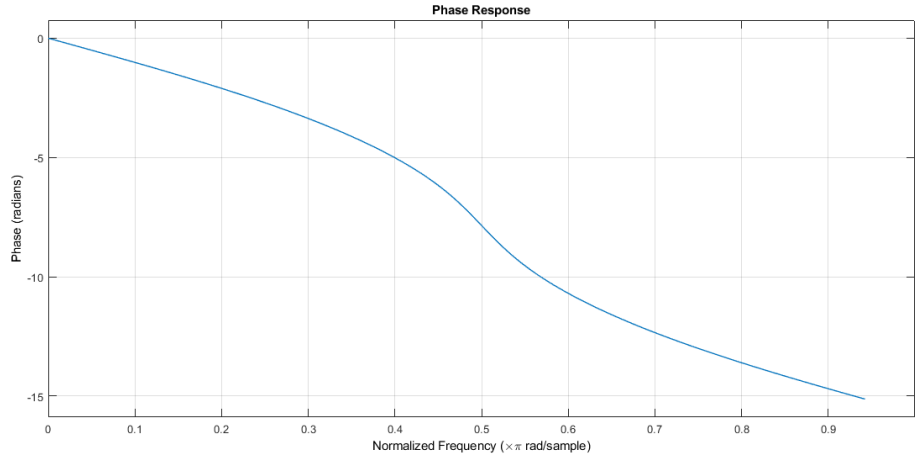

Figure 88: Phase Response of the Butterworth Low Pass Filter (Order 10)

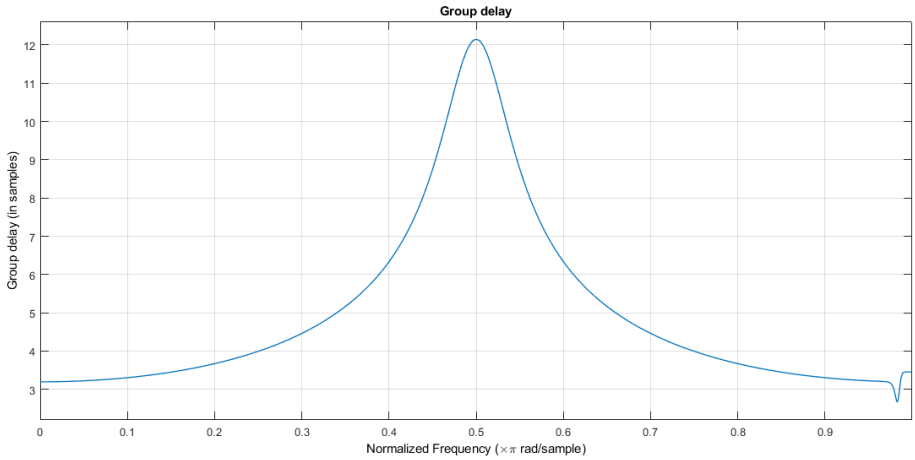

Figure 89: Group Delay of the Butterworth Low Pass Filter (Order 10)

5.1.3 Butterworth High Pass Filter and Comb Filter

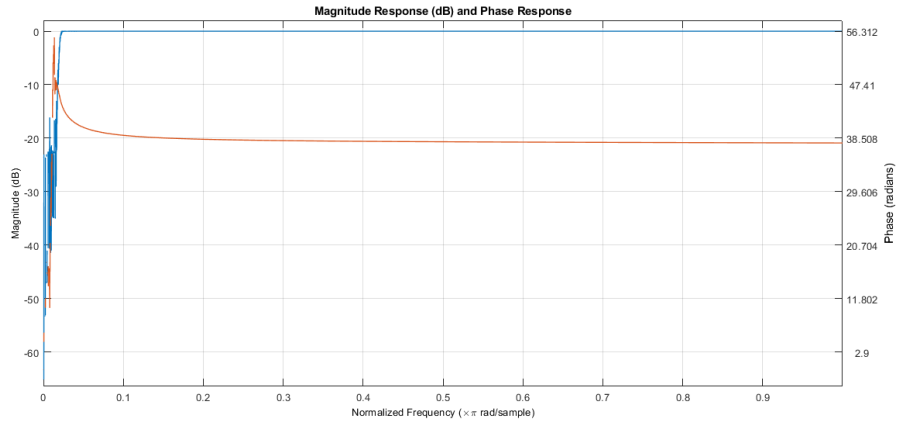

Figure 90: Magnitude Response and Phase Response of the Butterworth High Pass Filter (Order 10)

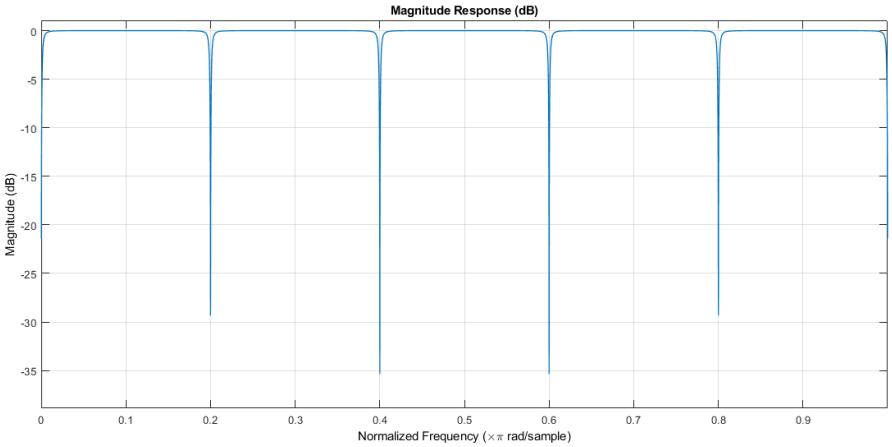

Figure 91: Magnitude Response of the IIR Comb Filter

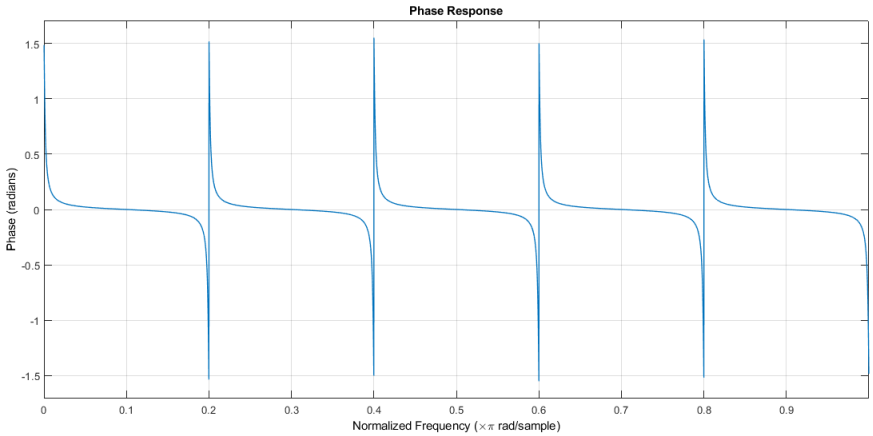

Figure 92: Phase Response of the IIR Comb Filter

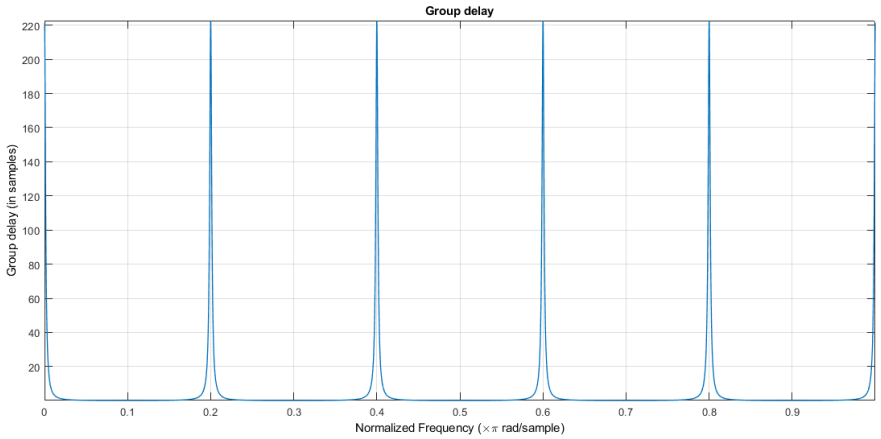

Figure 93: Group Delay of the IIR Comb Filter

5.1.4 Magnitude Response of the Combined Filter

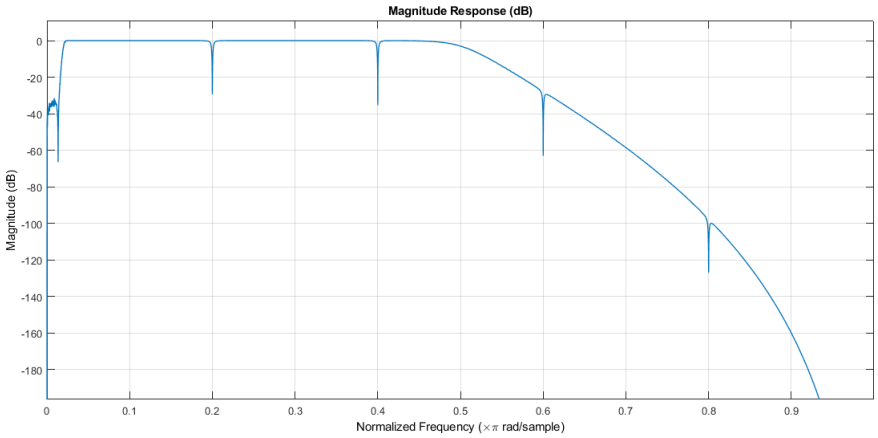

Figure 94: Magnitude Response of the IIR Combined Filter

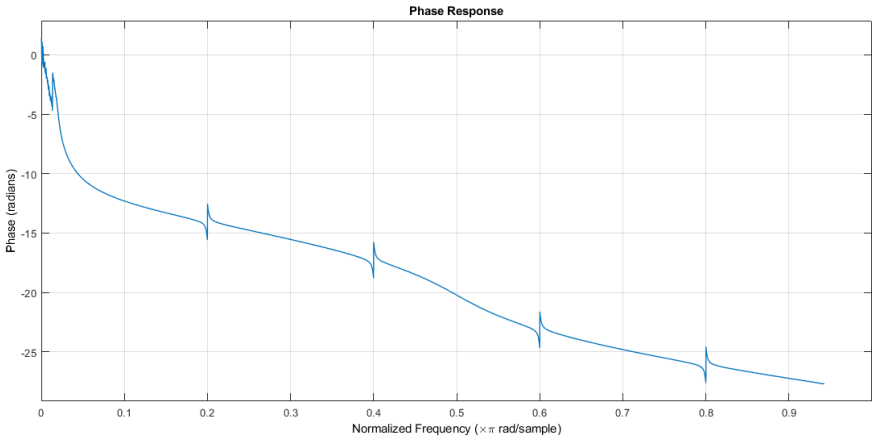

Figure 95: Phase Response of the IIR Combined Filter

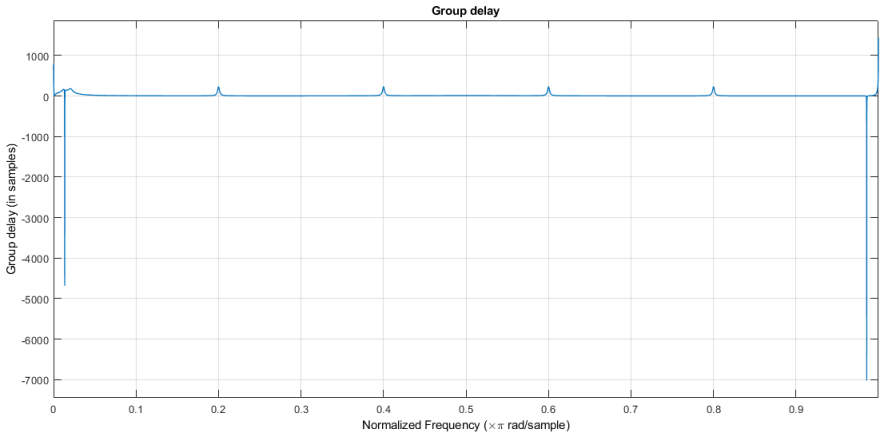

Figure 96: Group Delay of the IIR Combined Filter

5.1.5 Comparing the IIR Combined Filter with FIR Combined Filter

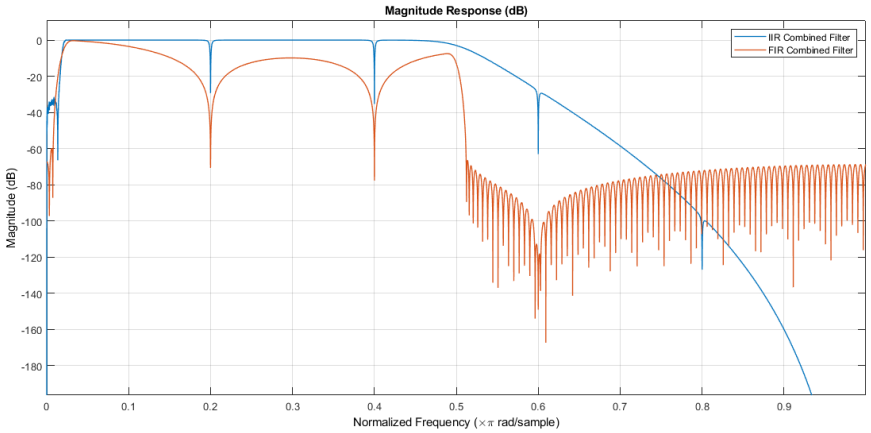

Figure 97: Magnitude Response of the IIR and FIR Combined Filters

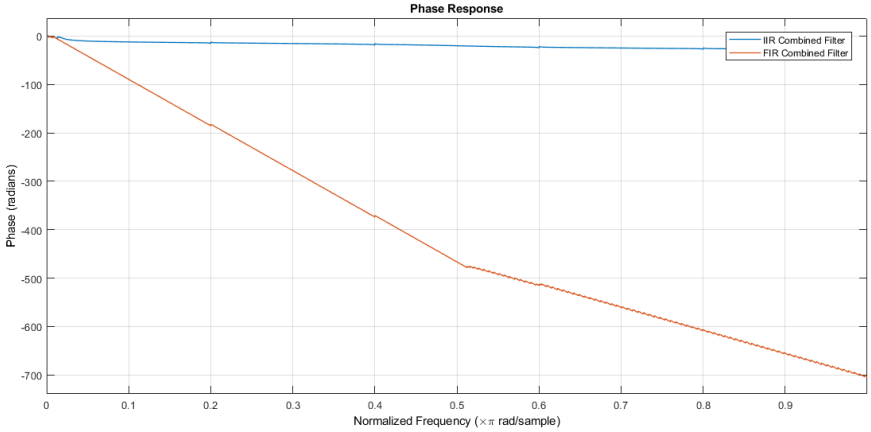

Figure 98: Phase Response of the IIR and FIR Combined Filters

Compared to the FIR filter, the IIR filter has a better magnitude response in the pass band with less attenuation. This behaviour is prominent specially around the 50Hz,100Hz, and 150Hz frequency components. The IIR comb filter incorporates a narrow bandwidth compared to the FIR filter. In addition, the stop band attenuation is more efficient in the IIR filter.

5.2 Filtering Methods using IIR Filters

5.2.1 Applying Forward Filtering

To apply forward filtering, the *filter(b,a,x)* command was applied on the 'ECG_with_noise.mat' signal with the IIR filters designed in the previous section. The implementation is provided in the code.

5.2.2 Applying Forward-Backward Filtering

To apply forward-backward filtering, the *filtfilt(b,a,x)* command was applied on the 'ECG_with_noise.mat' signal with the IIR filters designed in the previous section. The implementation is provided in the code.

5.2.3 Generating Time Domain Plots of the FIR Filtered ECG, IIR Forward Filtered ECG and IIR Forward-Backward Filtered ECG

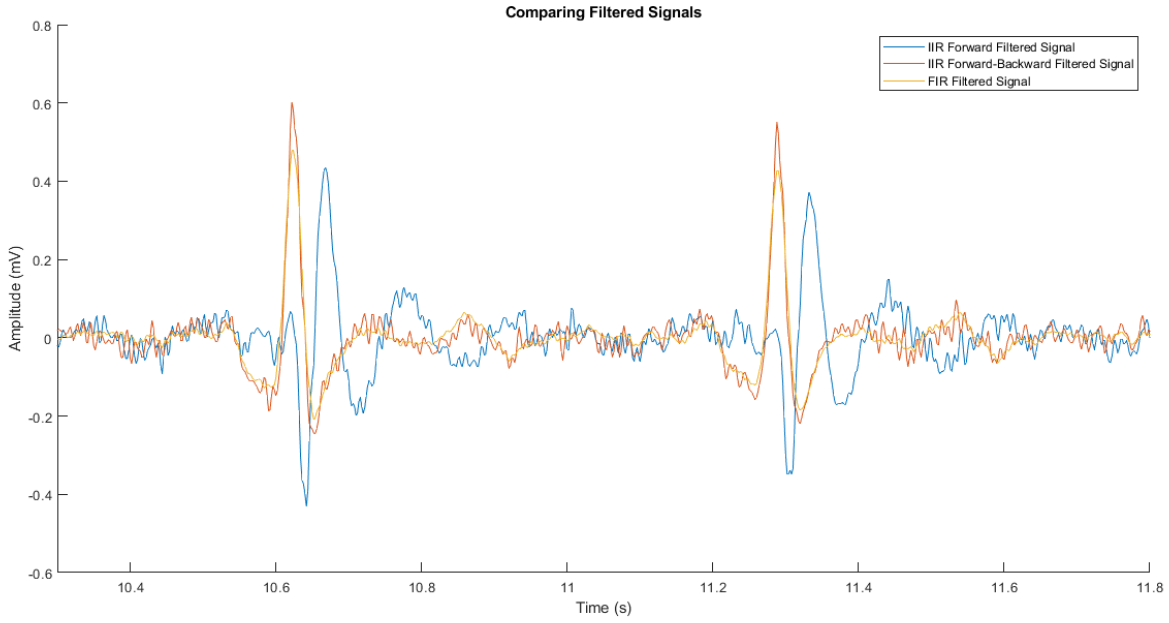

Figure 99: Time Domain Plots of the Filtered Signals

IIR filters typically correspond to better frequency responses. However, when it comes to phase response, their phase responses tend to be non linear. This limitation could be overcome in the forward backward filtering. In the above time domain plots, we can observe that the forward-backward filtered signal is better that the forward filtered signal. Moreover, the forward-backward filtered signal seems to be on par with the FIR filtered signal.

5.2.4 Generating PSD Estimates of the FIR Filtered ECG, IIR Forward Filtered ECG and IIR Forward-Backward Filtered ECG

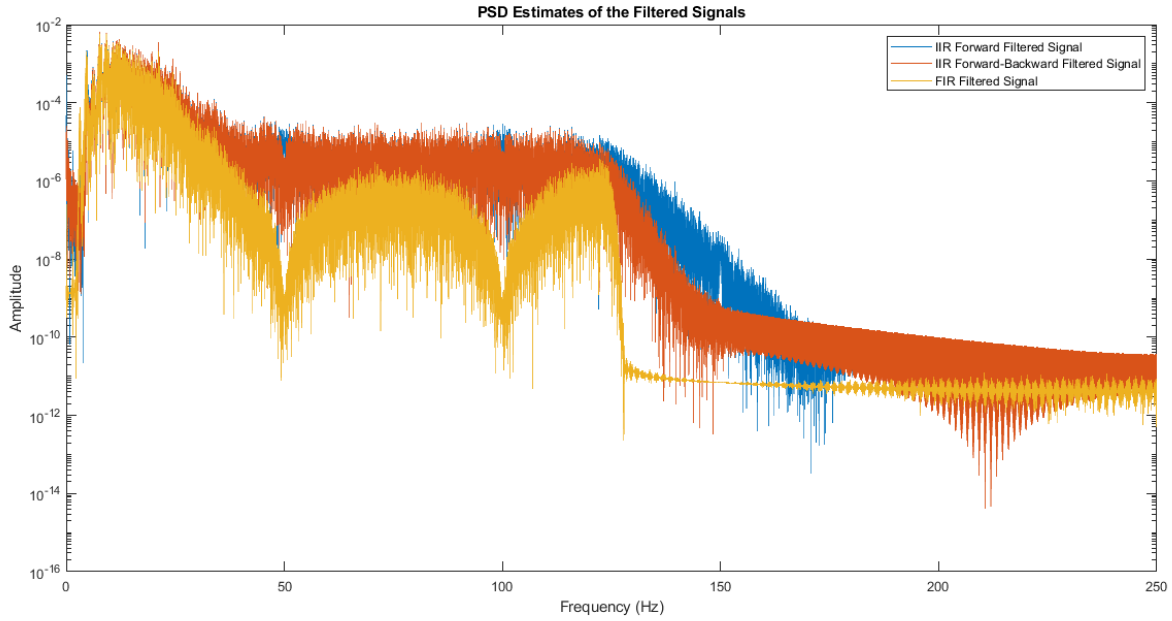

Figure 100: PSD Estimates of the Filtered Signals

Following observations could be made from Fig.100.

- High frequency (¿ 125Hz) suppression in the IIR forward filtered signal is the lowest compared to the other two filtered signals
- Up to 125Hz, the frequency components in the forward filtered signal and the forward-backward filtered signal seem to be identical
- FIR filter has done the best job when it comes to suppressing the spikes of the noisy ECG signal that were present at the frequencies; 50Hz, 100Hz, and 150Hz.