Department of Electronic and Telecommunication Engineering

University of Moratuwa

BM4152 - Biosignal Processing

Assignment 1 - Digital filters

PUSHPAKUMARA H.M.R.M. 200488E

Date: September 24, 2024

Table of Content

1 Smoothing Filters

1.1 Moving average *MA(N)* filter

i. Load *ECG_template.mat*

ii. Plot the loaded signal with the adjusted time scale

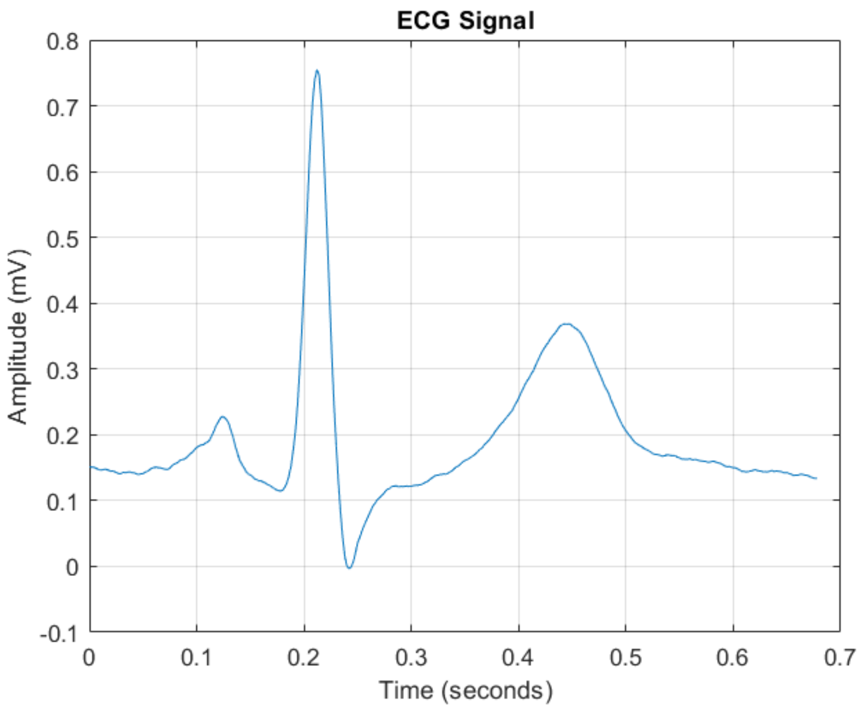

Figure 1: ECG template

iii. Add white Gaussian noise of 5 dB

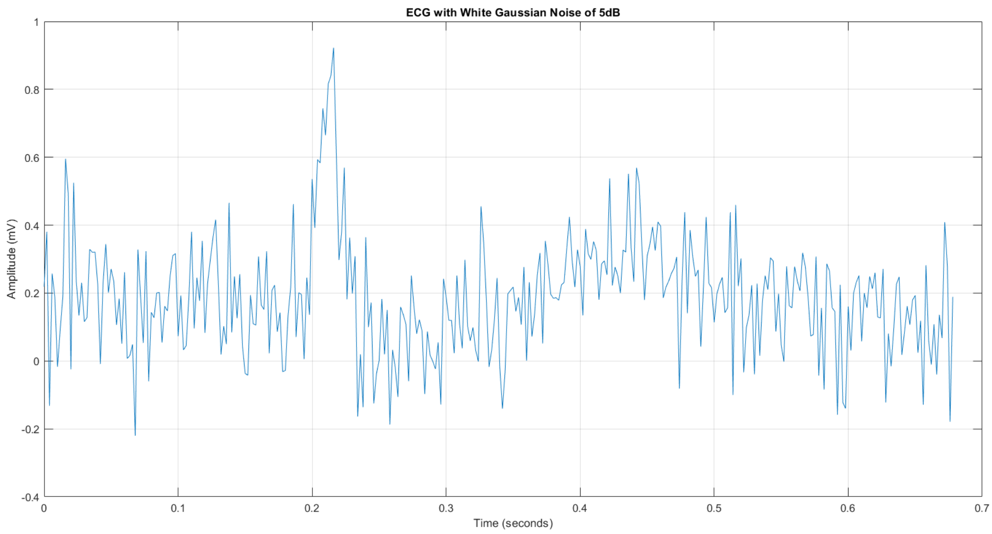

Figure 2: white Gaussian noise added ECG

iv. Plot the power spectral density (PSD) estimate of the nECG

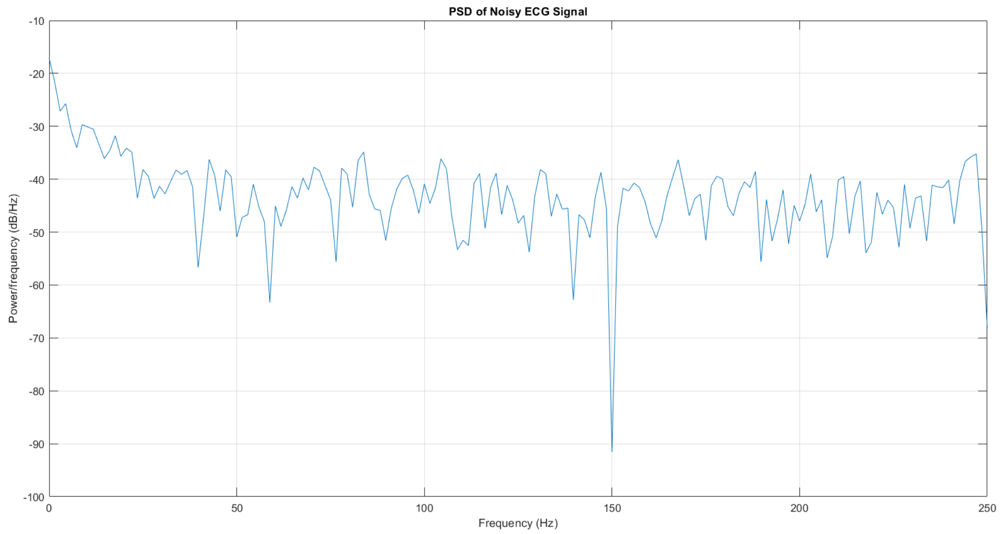

Figure 3: PSD of the white Gaussian noise added ECG

1.1.1 *MA(3)* filter implementation with a customised script

i. Write a MATLAB script for a MA(3) filter formulated in *Equation 1*
The written customized Matlab script for MA(3) has saved as MA_3.m

ii. Derive the group delay.

$$Group\ delay = \frac{N-1}{2} \tag{1}$$

Here, the order is 3. Therefore the group delay is 1.

iii. Plot the delay compensated filtered ECG Signal and noisy ECG on the same plot and compare

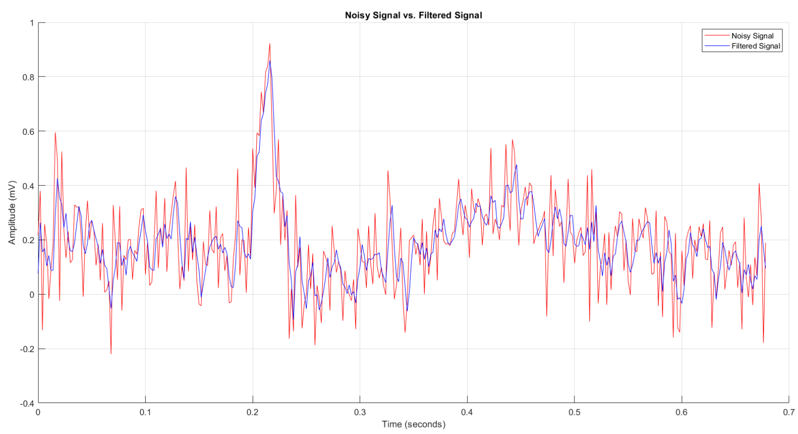

Figure 4: MA(3) filtered ECG Signal and noisy ECG

Here we can see, after applying the moving average filter, the signal-to-noise ratio has reduced by a considerable amount. But still, the signal contains a significant amount of noise and needs further filtering to extract the real ECG.

iv. Produce overlapping PSDs of filtered ECG Signal and noisy ECG signal and compare the filter effect

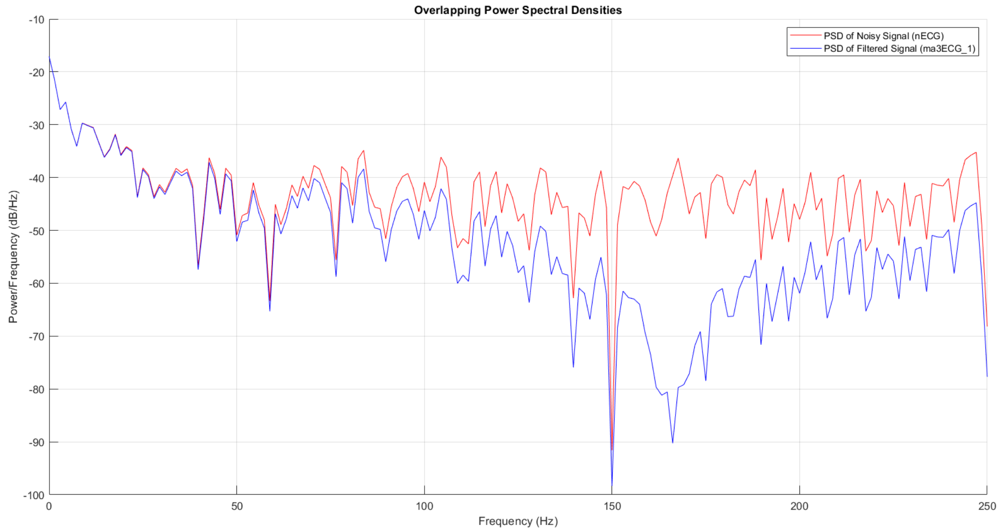

Figure 5: PSD of Noisy and Filtered ECG signals

When we compare the power spectral densities of the noisy ECG and the filtered ECG, we can see the reduction of the frequencies beyond 100Hz. This is acceptable since ECG doesn't have important details beyond 150Hz. And vice versa, it has preserved the frequencies that contained information.

Also, the signal-to-noise ratio has improved here.

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

i. Use the *filter(b,a,X)* command to filter the nECG signal with a *MA(3)* filter while compensating for the group delay. Name the filtered signal as *ma3ECG_2*.

ii. Plot *nECG*, *ECG_template* and *ma3ECG_2* on the same plot and compare.

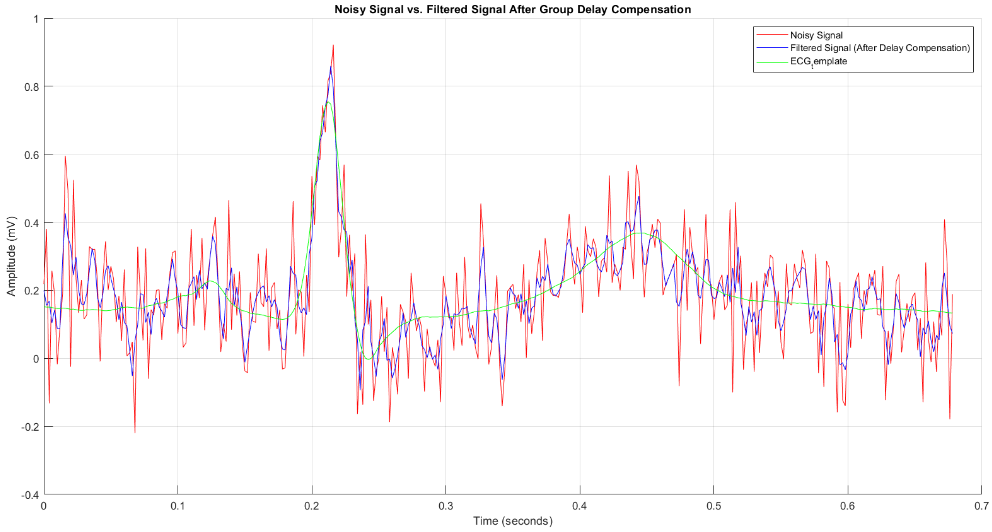

Figure 6: Noisy ECG Signal, ECG template and filtered ECG Signal using inbuilt Matlab MA filter

Here we can see, as above the signal-to-noise ratio has improved, but still, it contains lots of unwanted noise. After filtering, the noisy signal has become some sort of identifiable ECG shape, but not perfect or contain useful information. It needs further filtering.

iii. Use the *fvtool(b,a)* to inspect the magnitude response, phase response and the polezero plot of the MA(3) filter

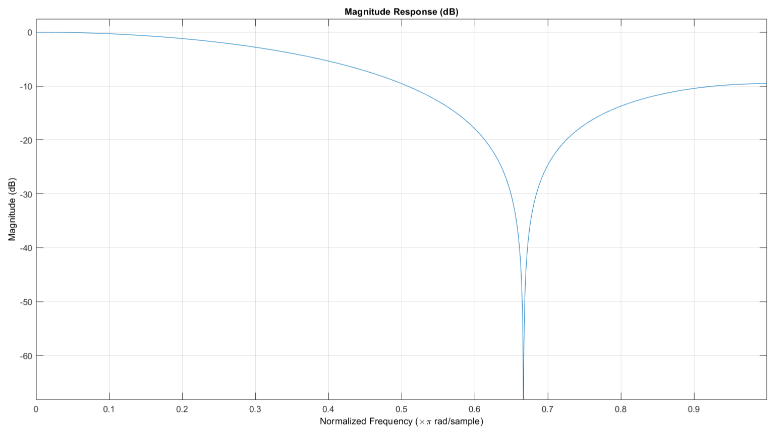

Figure 7: Magnitude response

Here we can see the magnitude response gives some sort of low pass filtering effect.

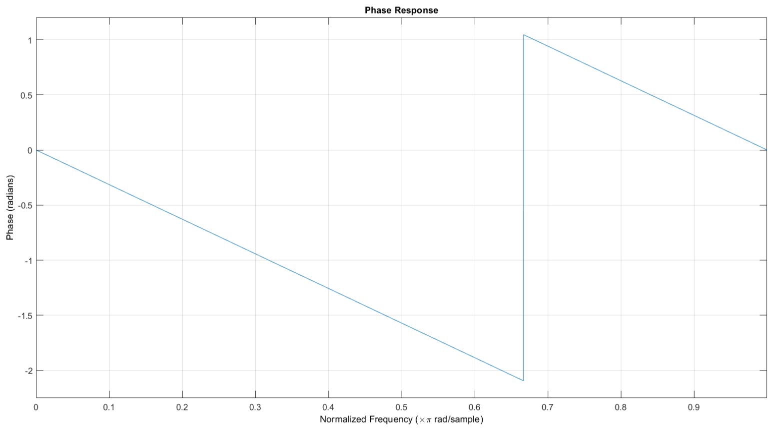

Figure 8: Phase response

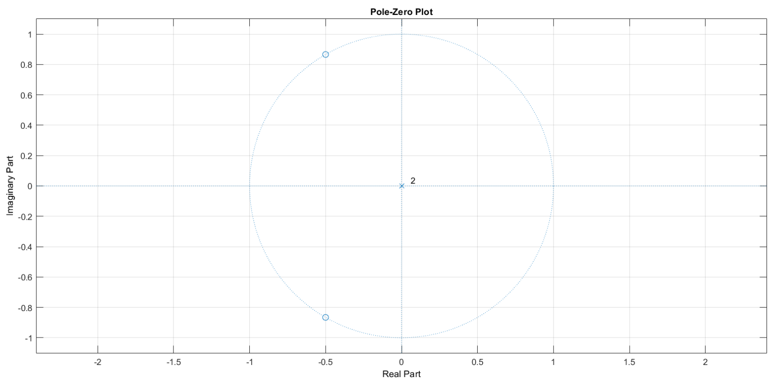

Figure 9: pole-zero plot

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

i. Identify the improvement of the MA(10) filter over the MA(3) filter using the *fvtool(b,a)*

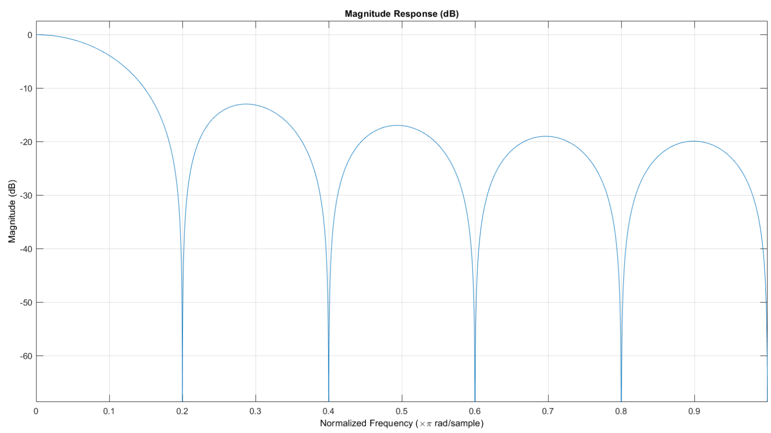

Figure 10: Magnitude response

Here we can see when compared with the magnitude response of the MA(3), this has a sharper edge and a comparatively low passband gain

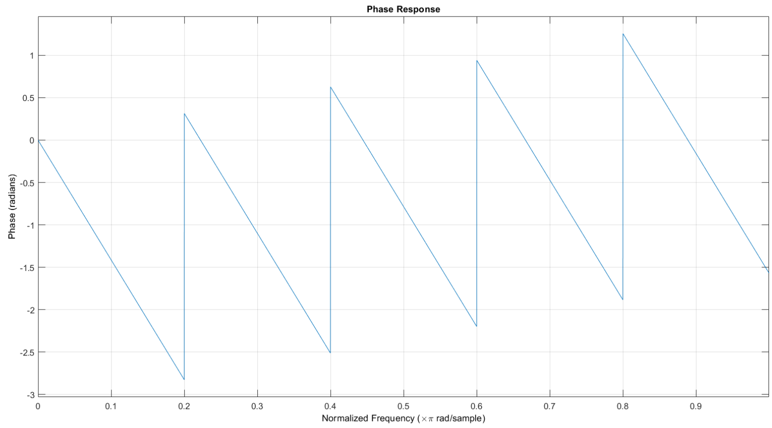

Figure 11: Phase response

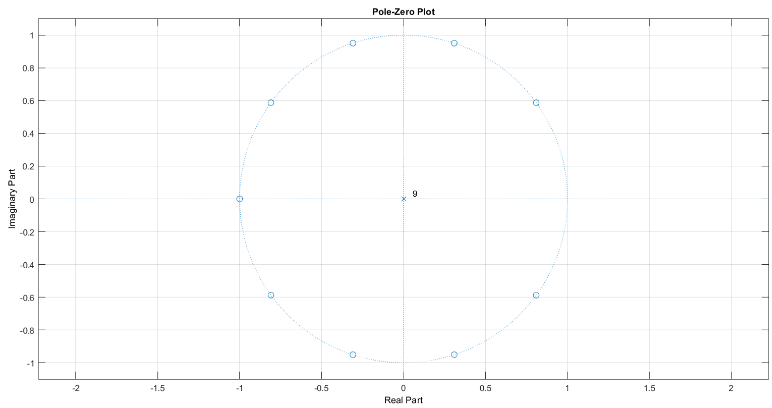

Figure 12: pole-zero plot

ii. Filter the nECG signal using the above *MA(10)* filter while compensating for the group delay.

iii. Name the filtered signal *ma10ECG*

iv. Plot *nECG*, *ECG_template*, *ma3ECG_2* and *ma10ECG* on the same plot to compare the improvement

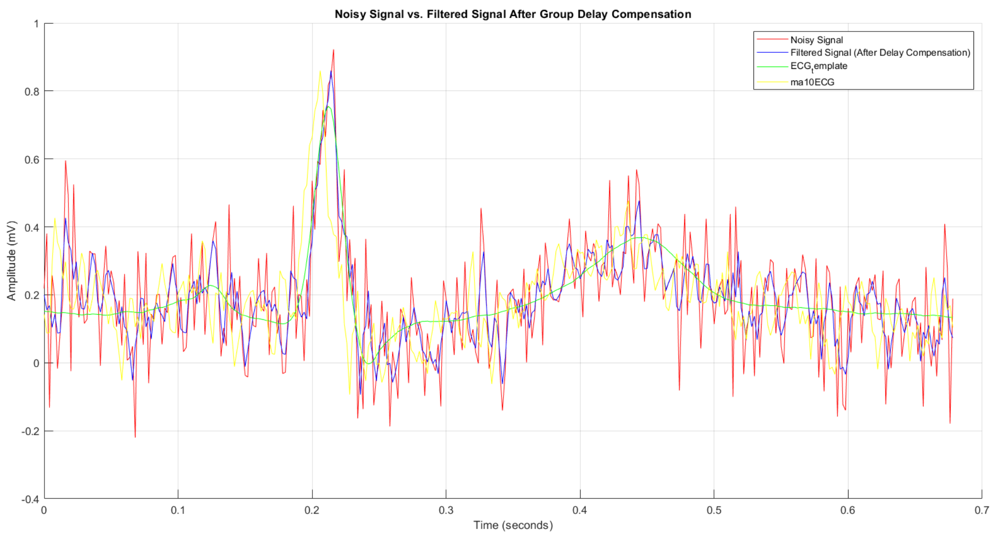

Figure 13: nECG, ECG template, ma3ECG and ma10ECG

When comparing the plots of the filtered signals, we observe that the signal filtered with a moving average filter of window size 10 (MA(10)) exhibits a lower signal-to-noise ratio (SNR) compared to the signal filtered with a moving average of window size 3 (MA(3)). Although the MA(10) filter provides more smoothing, it is not sufficient to fully eliminate the noise present in the signal.

Even after filtering with the larger MA(10) window, the signal still contains a considerable amount of residual noise. This remaining noise could interfere with the extraction of key information from the signal, especially in scenarios where high precision is required, such as ECG. To obtain a cleaner signal for accurate interpretation, more effective filtering methods are needed.

1.1.4 Optimum *MA(N)* filter order

i. Write a MATLAB script to calculate the mean-squared-error (MSE) between the noise-free signal and the filtered signal with the *MA(N)* filter
The written customized Matlab script for calculate the mean-squared-error (MSE) has saved as MSE_MA_filter.m

ii. Hence test for a range of N values and determine the optimum filter order which gives the minimum MSE by plotting MSE vs. N

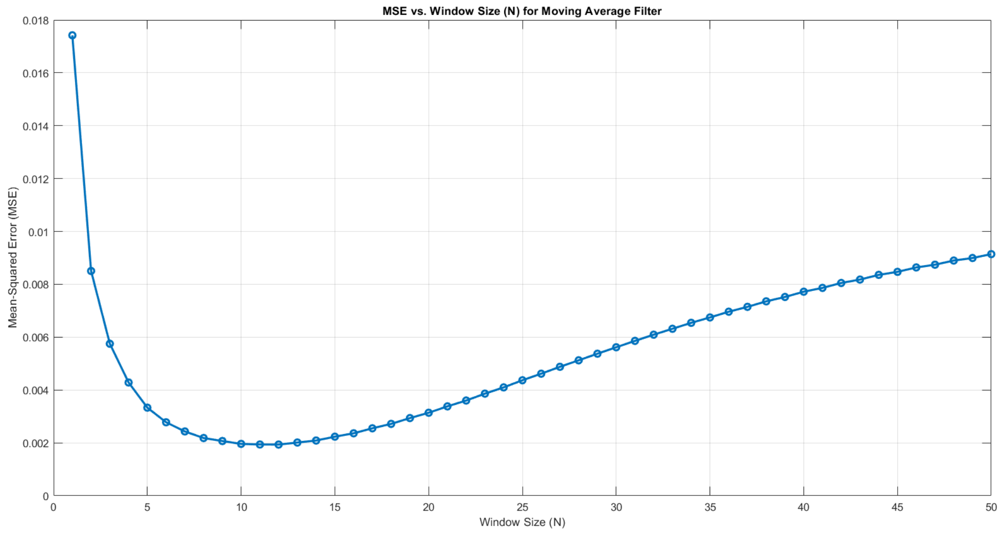

Figure 14: MSE values vs filter order

Optimum filter order (N) = 12

iii. Suggest the reasons for large MSE values at low and high filter orders

- **Low filter order:** While this can help reduce noise, it might not be sufficient for signals with high noise levels. The filter might not capture enough data points to effectively smooth out the noise. Therefore, it results large MSE value.
- **High filter order:** A larger filter order can provide more significant noise reduction. However, it can also lead to excessive smoothing, blurring important signal details. This can also result in large MSE values.

1.2 Savitzky-Golay SG(N,L) filter

1.2.1 Application of SG(N,L)

i. Apply a *SG(3,11)* filter on the *nECG* signal while compensating the group delay. Name the filtered signal as *sg310ECG*.Use the command *sgolayfilt(x,N,L)*

ii. Plot nECG, *ECG_template* and *sg310ECG* on the same plot and compare

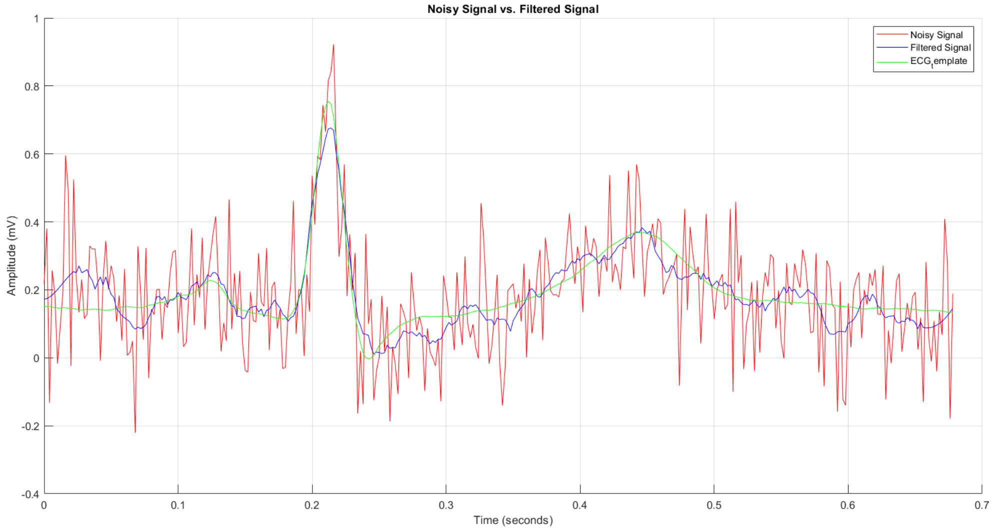

Figure 15: nECG, ECG_template and the filtered ECG signal (sg310ECG)

Here we can see, the signal filtered with a Savitzky-Golay filter (SG(3,11)) shows a significant improvement in the signal-to-noise ratio (SNR) compared to the signal filtered with the moving average filter (MA(10)). The Savitzky-Golay filter has not only improved the SNR but also smoothed the signal more effectively, preserving important features of the waveform.

However, despite this improvement, there is still some residual noise present in the filtered signal. For more accurate analysis or feature extraction, this remaining noise must be further reduced.

1.2.2 Optimum SG(N,L) filter parameters

i. Calculate the MSE values for a range of parameters of the *SG(N,L)* filter and determine the optimum filter parameters which gives the minimum MSE.

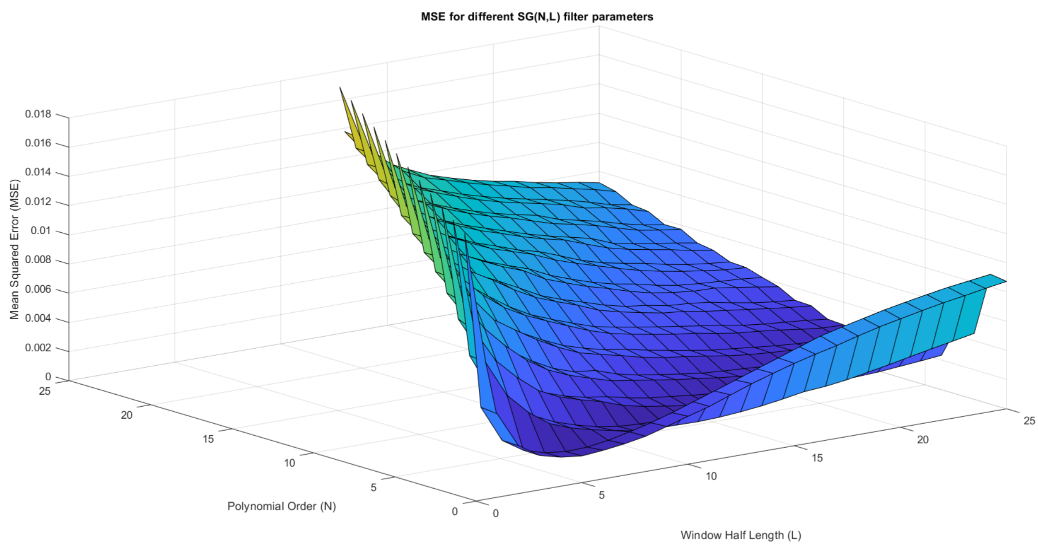

Figure 16: Plot of MSEs with N and L

The optimum polynomial parameters

- ***N = 7***
- ***L = 17 (Length = 2L+1 = 35)***

with minimum MSE of 0.0017074

ii. Plot *ECG_template*, *sg310ECG* and the signal obtained from *optimum SG(N,L)* filter on the same plot to compare the improvement

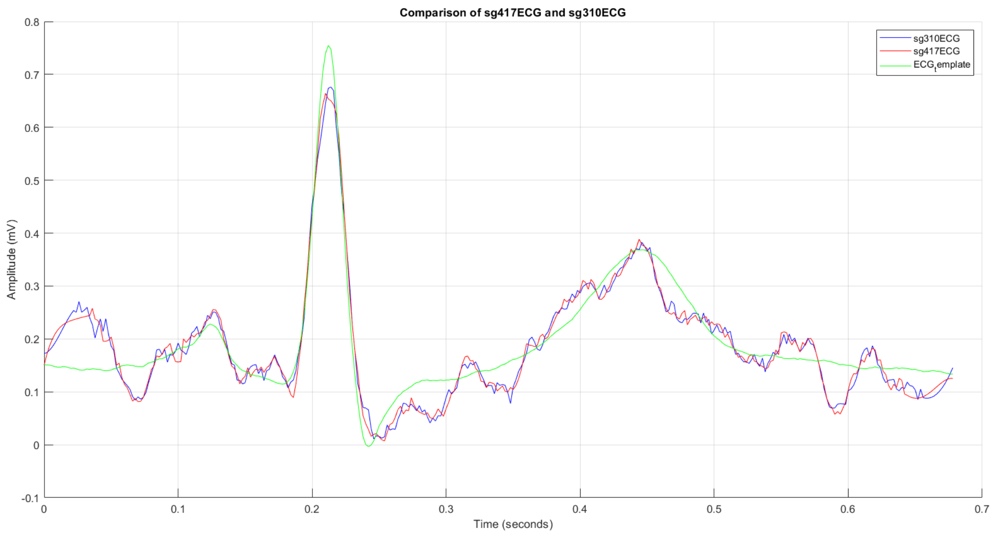

Figure 17: ECG_template, sg310ECG and the signal obtained from optimum SG(N,L) filter

When comparing the Savitzky-Golay filter with optimum parameters (N, L) to the SG(3,11) filter, we can observe that the optimum filter has provided a more significant smoothing of the signal and has reduced a larger portion of the high-frequency noise. The result is a cleaner signal with enhanced features when compared to the SG(3,11) filter.

However, despite the improvement, there is still some low-frequency noise present in the filtered signal. Also, this noise has been smoothed along with the signal by the optimum filter, which might mask certain subtle but important features in the signal.

iii. Compare signal features and computational complexity of the optimum filtered signals derived from *MA(N)* and *SG(N,L)* filters

- Optimum Savitzky-Golay filter
  - N = 7
  - L = 17

  The SG filter is computationally more demanding compared to the MA filter. It involves fitting a polynomial of order L (where $L = 17$) to each window of size N, requiring polynomial regression. This involves more complex arithmetic operations, such as solving systems of equations, leading to a time complexity of $\mathcal{O}(N \cdot L)$.

- Optimum Moving Average filter
  - N = 12

The MA filter has a relatively low computational complexity. For each sample in the signal, it requires N additions (where $N = 12$) and 1 division, leading to a time complexity of $\mathcal{O}(N)$ for each sample. Thus, it's efficient, especially for real-time applications.

Feature	Optimum MA filter (N = 12)	Optimum SG filter (N = 7, L = 17)
Smoothing	Moderate smoothing, may blur details	Significant smoothing, preserves details
Signal Shape Preservation	Can distort sharp features	Excellent preservation of sharp features
Noise Reduction	Reduces high-frequency noise	Reduces both high- and low-frequency noise
Phase Response	Causes phase shift (signal lag)	Minimal phase distortion (linear phase)
Computational Complexity	$\mathcal{O}(N)$ (low complexity)	$\mathcal{O}(N)$ (higher complexity)
Suitability for Real-Time	Suitable for real-time systems	Higher overhead, less suited for real-time without optimization

Table 1: Comparison of Signal Features and Computational Complexity Between Optimum Moving Average (MA) and Savitzky-Golay (SG) Filters

2 Ensemble averaging

2.1 Signal with multiple measurements

2.1.1 Preliminaries

i. Clear the workspace and the command window

ii. Load *ABR_rec.mat* to MATLAB workspace

iii. Plot the train of stimuli and ABRs on a single plot and observe

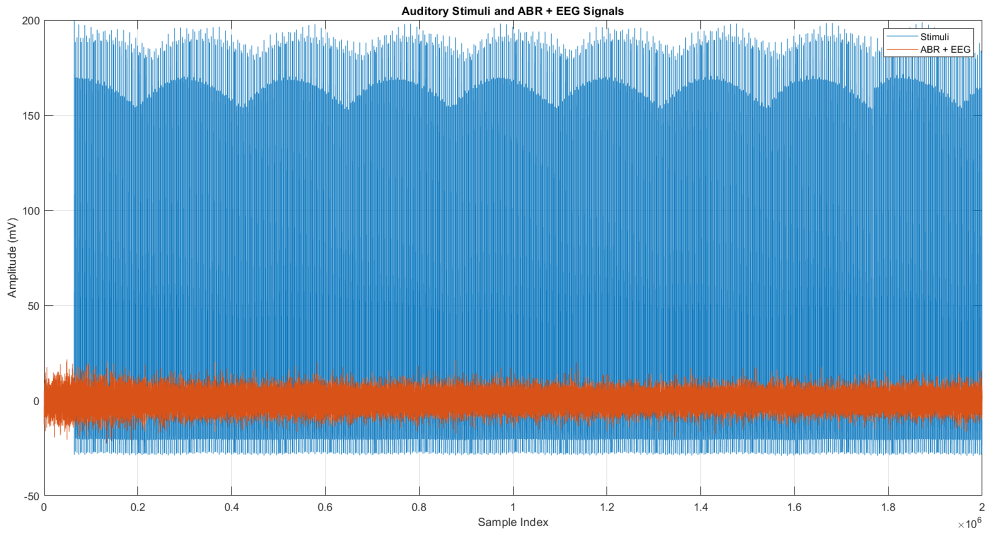

Figure 18: Train of stimuli and ABRs

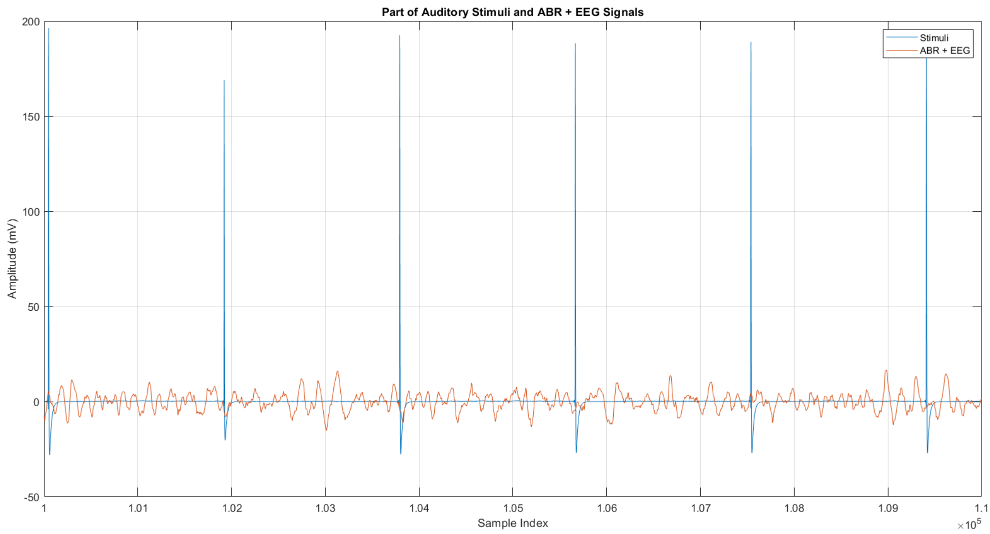

Figure 19: part of the train of stimuli and ABRs

iv. Determine a voltage threshold to automatically detect the likely stimuli occurrences

v. Extract actual stimulus points

vi. Window ABR epochs according to the extracted stimulus points. Consider 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)*

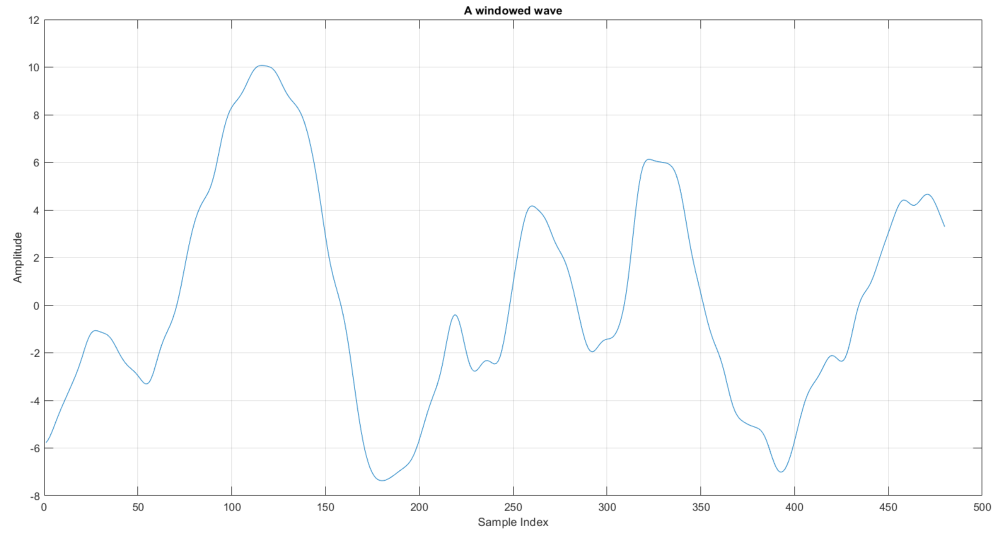

Figure 20: A windowed wave

vii. Calculate the ensemble average of all the extracted epochs

viii. Plot the ensemble averaged ABR waveform

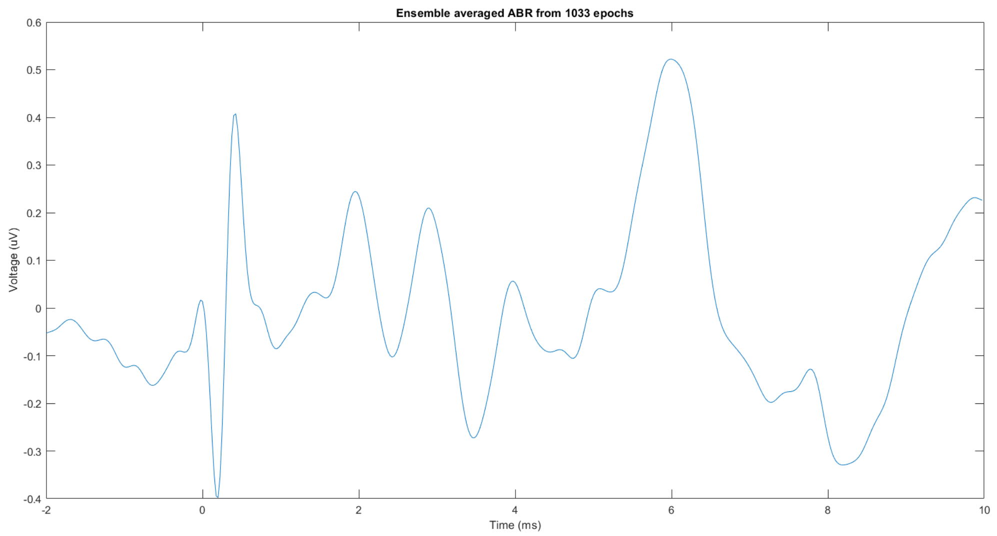

Figure 21: ensemble - averaged ABR waveform

2.1.2 Improvement of the SNR

i. Using the ensemble averaged ABR *(ensmbl_avg)* as the template, write a MATLAB script to calculate an array containing progressive MSEs

The written Matlab script for calculate an array containing progressive MSEs has been saved as *calculate_progressive_MSEs.m*

ii. Plot a graph of MSEk against k and describe the behaviour with reference to the theoretically derived improvement of the SNR

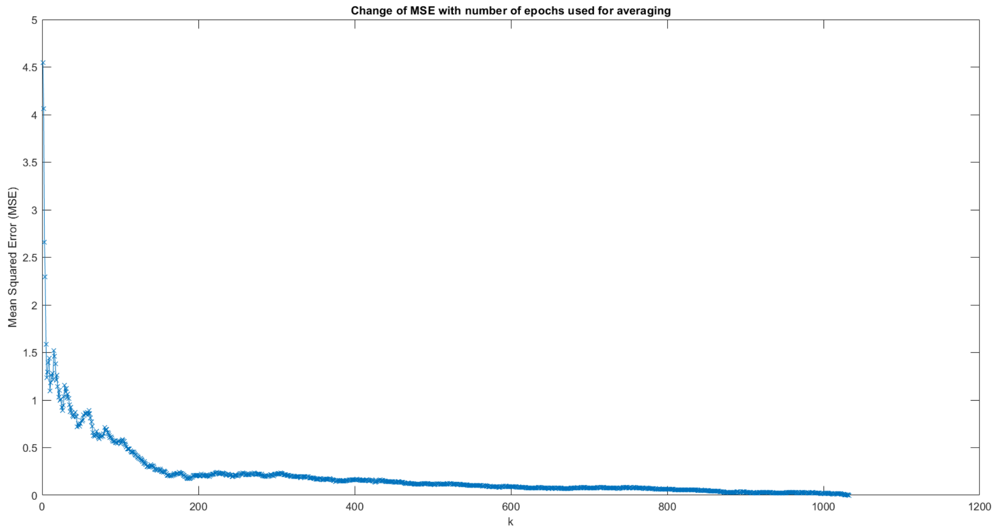

Figure 22: MSE vs number of waves (k)

The relationship between the number of averages k and the Mean Squared Error (MSE) in auditory brainstem responses (ABR) shows a clear trend: as k increases, the MSE consistently decreases. This happens because averaging multiple measurements reduces the noise in the signal.

Theoretically, when noise with a mean of zero and variance σ^2 is added to a signal, averaging k measurements decreases the variance of the noise by a factor of k. This leads to an improvement in the Signal-to-Noise Ratio (SNR) of the signal. Specifically, the SNR improves as:

$$\text{SNR} \propto \frac{1}{\sqrt{k}}$$

This means that as k increases, the noise becomes smaller, resulting in lower MSE values. The graph of MSE_k against k shows this downward trend, reflecting how effective averaging is at making the signal clearer. Overall, the decrease in MSE matches the expectation that more averages give a clearer signal, thus improving the SNR.

2.2 Signal with repetitive patterns

2.2.1 Viewing the signal and addition of Gaussian white noise

i. Load *ECG_rec.mat* to MATLAB workspace

ii. Plot the data and observe amplitudes and the waveforms.

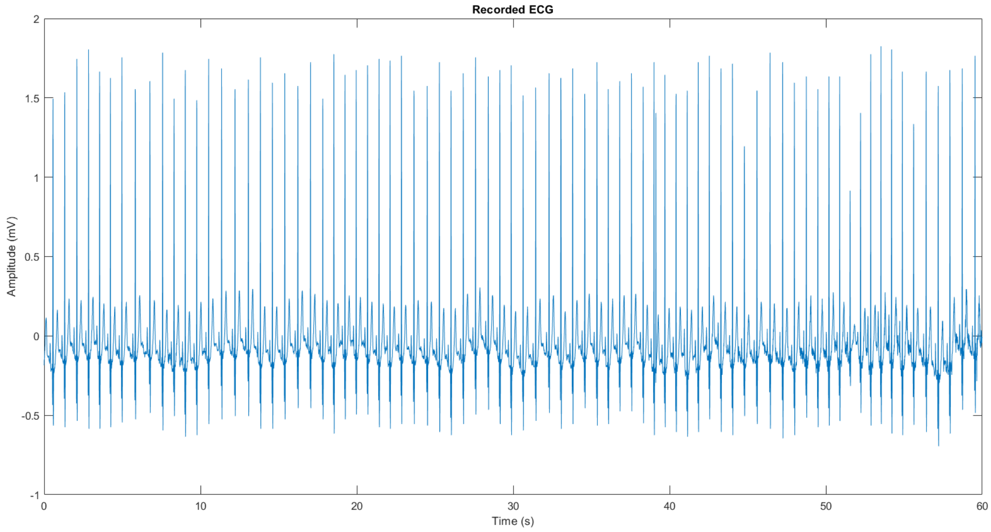

Figure 23: ECG data

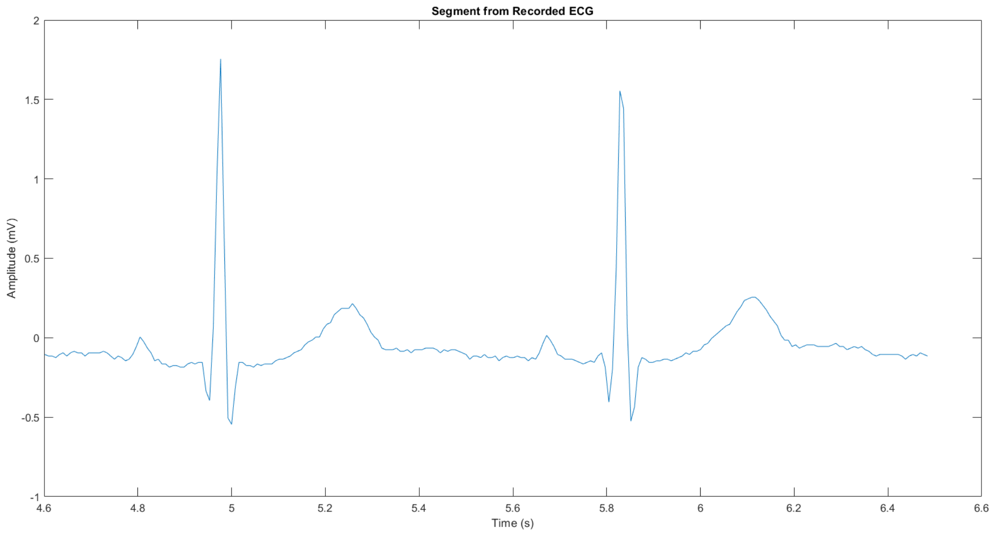

Figure 24: Segment from the ECG data

iii. Extract a single PQRST waveform and define it as the *ECG_template*

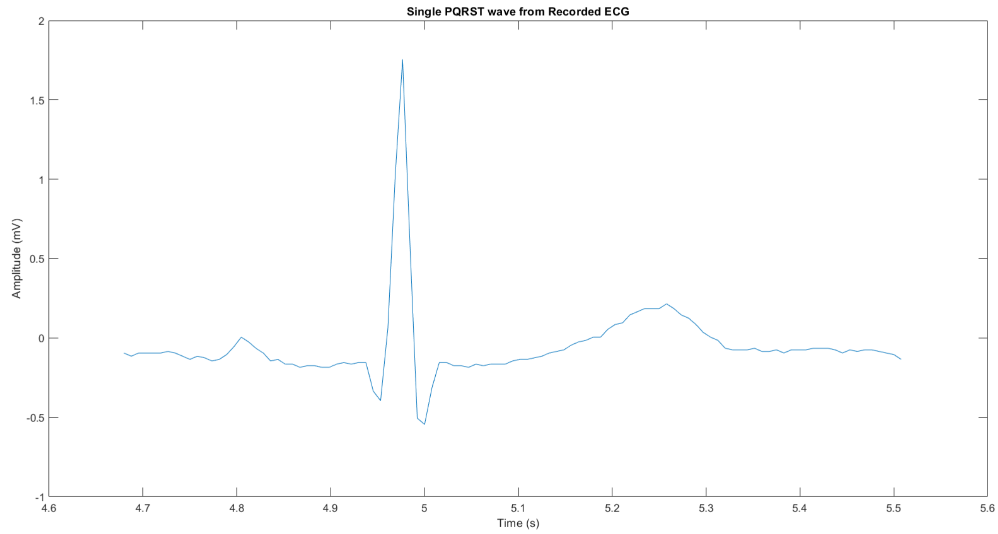

Figure 25: Extracted single PQRST waveform

iv. Add Gaussian white noise of *5 dB* to *ECG_rec* and name it as *nECG*

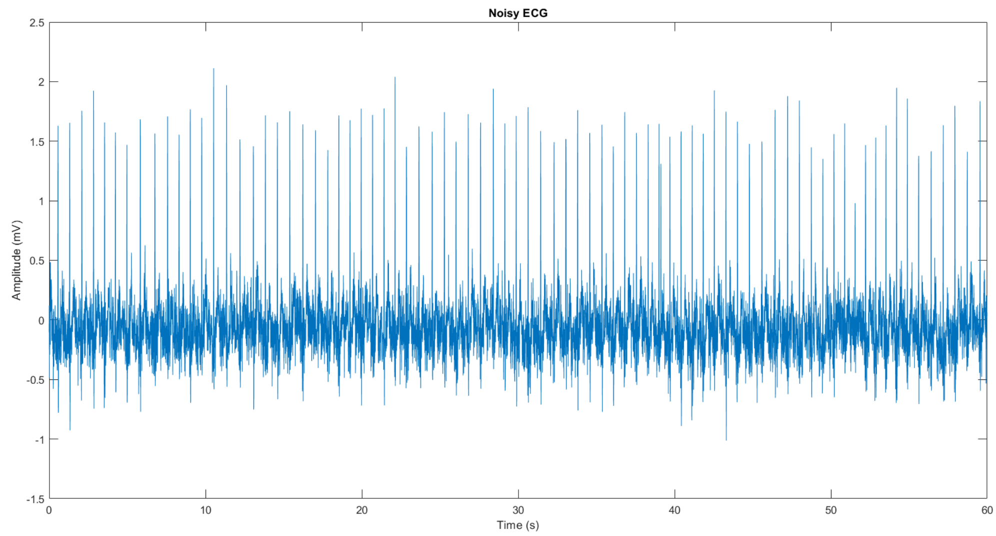

Figure 26: 5 dB white Gaussian noise added ECG wave

2.2.2 Segmenting ECG into separate epochs and ensemble averaging

i. Calculate the normalised cross-correlation between the *ECG_template* and the *nECG*

ii. Plot the normalized cross-correlation values against the adjusted lag axis converted to the time axis

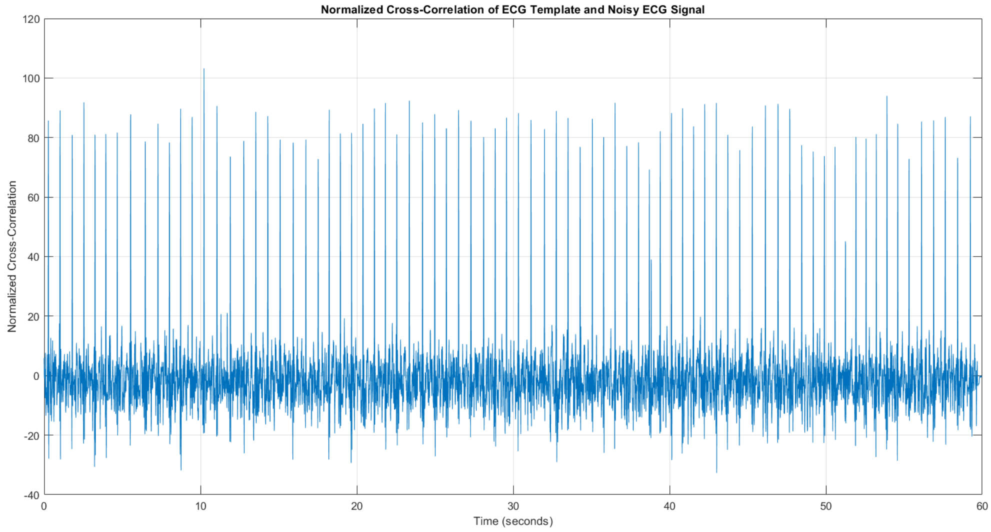

Figure 27: Normalized cross-correlation values plotted against the time-adjusted lag axis

iii. Segment ECG pulses by defining a threshold and store in a separate matrix

iv. Calculate and plot the improvement in SNR as the number of ECG pulses included in the ensemble average is increased

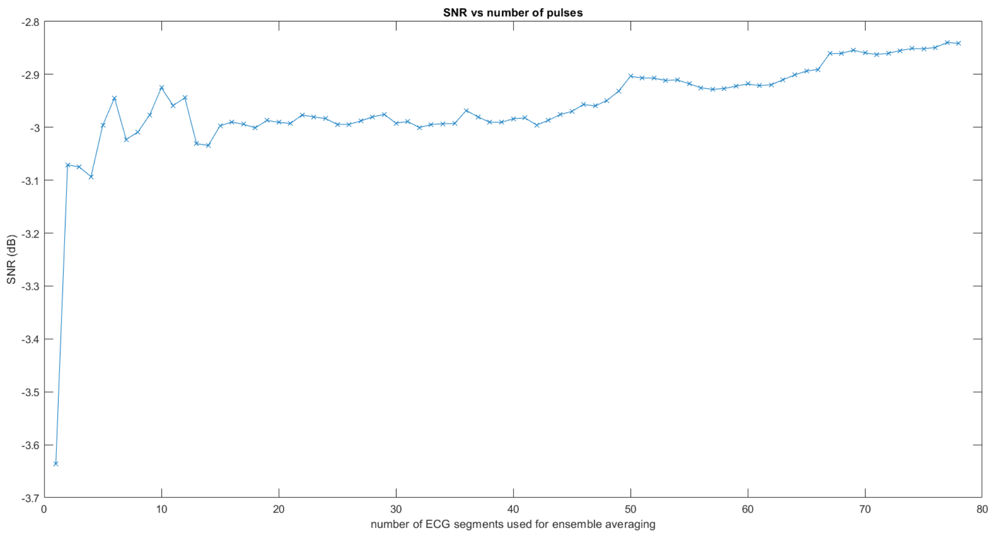

Figure 28: Improvement in SNR with number of ECG pulses

v. Plot and compare (in the one graph), a selected noisy ECG pulse and two arbitrarily selected ensemble averaged ECG pulses

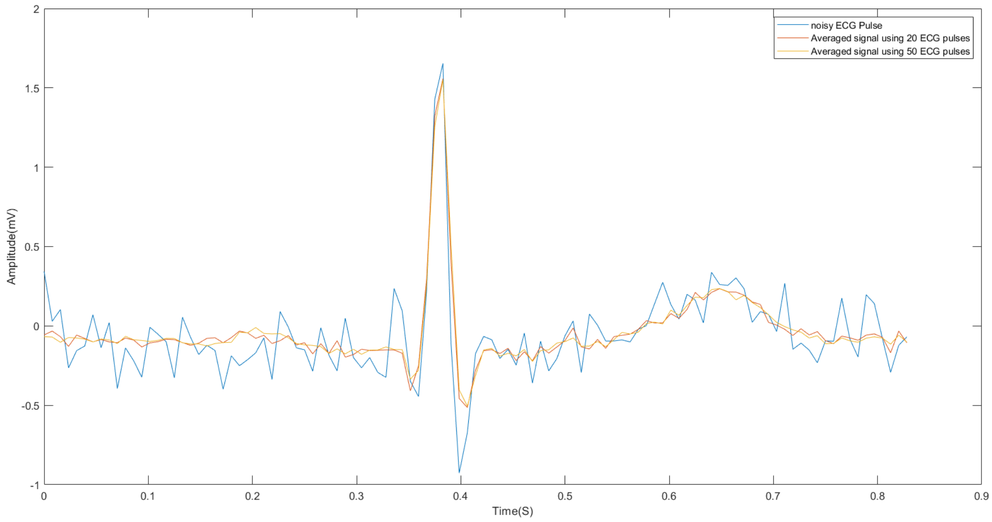

Figure 29: Averaged ECG wave with 20 epochs and 50 epochs

In the comparison graph, three ECG signals are plotted: one selected noisy *ECG pulse*, an *ensemble-averaged ECG pulse created from 20 pulses*, and another *ensemble-averaged ECG pulse created from 50 pulses*. The noisy ECG pulse displays significant fluctuations and noise, making it difficult to identify key features and other morphological characteristics, while the presence of high-frequency noise distorts the signal and reduces its clarity. The ensemble-averaged ECG pulse from 20 pulses shows some improvement over the noisy ECG pulse, with somewhat clearer main features, but it still retains some noise and noticeable artifacts. In contrast, the ensemble-averaged ECG pulse from 50 pulses demonstrates a marked improvement in accuracy and clarity, with significantly reduced noise and well-defined key features of the ECG waveform. Overall, the observations confirm that increasing the number of ECG pulses included in the ensemble average enhances the signal quality, with the averaged signal from 50 pulses exhibiting the least noise and the most accurate representation of the underlying ECG waveform.

vi. Suggest a justification to the claim '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'

Using points of maximum correlation to segment the ECG pulse train is better than just detecting the R-wave because it handles noise and changes in the ECG signals more effectively. R-wave detection can sometimes miss beats or incorrectly identify them, especially if there is noise or artifacts in the signal. In contrast, maximizing correlation looks at the whole shape of the ECG waveform, making it easier to find the right segments even when the R-wave is not clear. This approach improves accuracy by capturing important changes in the ECG signal, leading to fewer mistakes and more reliable extraction of segments for further analysis.

3 Designing FIR filters using windows

3.1 Characteristics of window functions (use the fdatool)

i. The rectangular window function is the simplest. Explain the effect of the length of this window function *(M)* using overlaying plots of the impulse response for *M = 5, 50, 100*

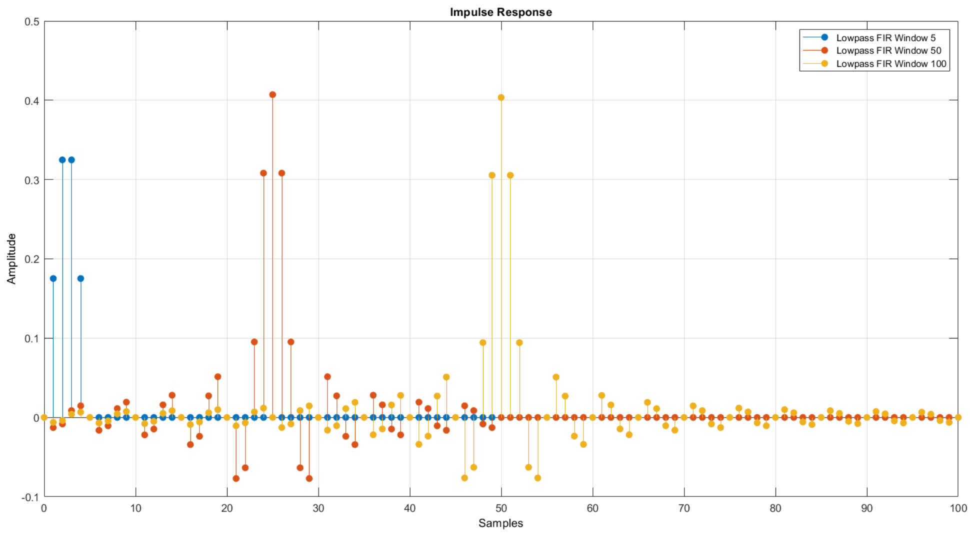

Figure 30: plots of the impulse response for M = 5, 50, 100

The length of the rectangular window function M has a significant impact on the impulse response and, consequently, on the filter's performance. As M increases, the impulse response approaches the ideal $sinc$ function, resulting in improved frequency selectivity, reduced spectral leakage, and a more accurate representation of the desired filter characteristics.

ii. For a lowpass filter having a cut-off frequency at $\omega_c = 0.4\pi$ (normalized frequency), we will plot the comparative magnitude response with a linear magnitude scale and the comparative phase response plot for filter lengths *M = 5, 50, 100*

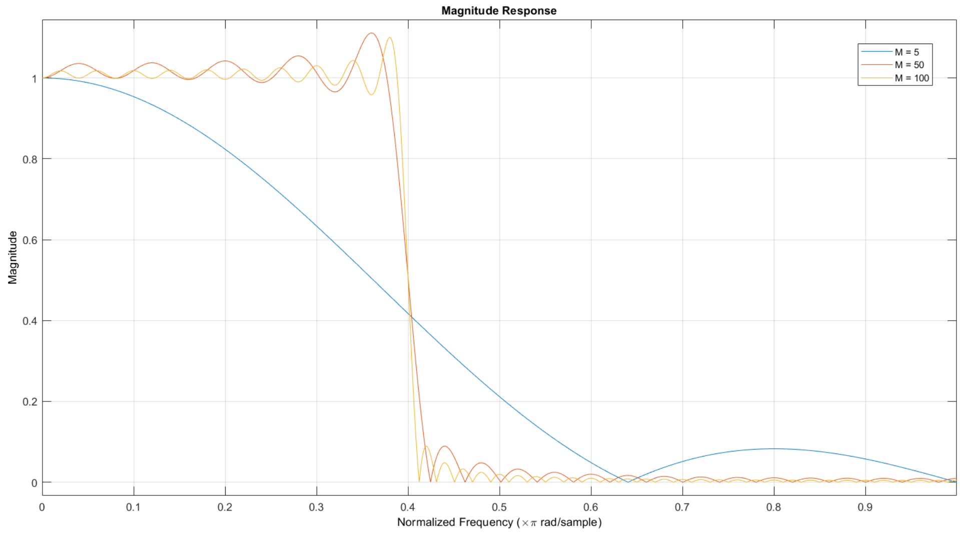

Figure 31: plots of magnitude responses for M = 5, 50, 100

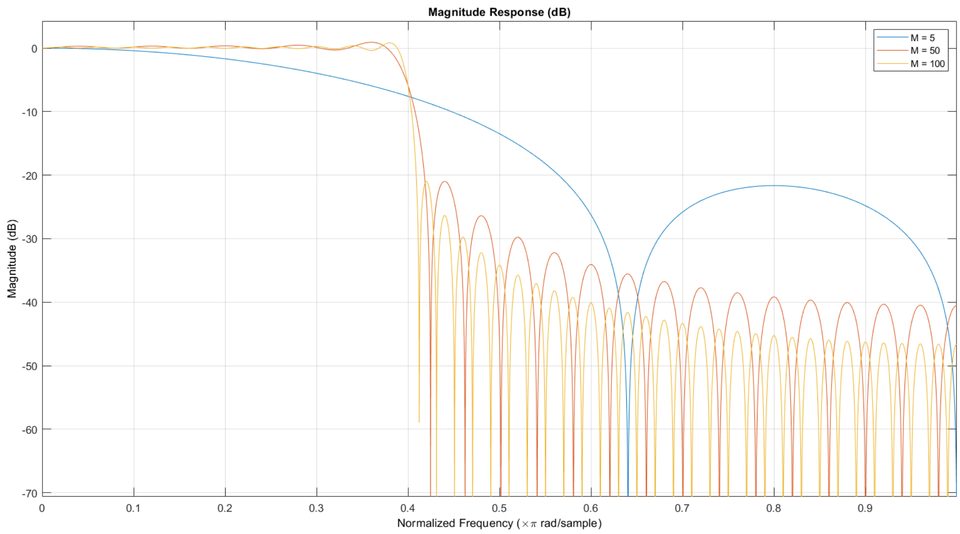

Figure 32: plots of magnitude responses (in dB) for M = 5, 50, 100

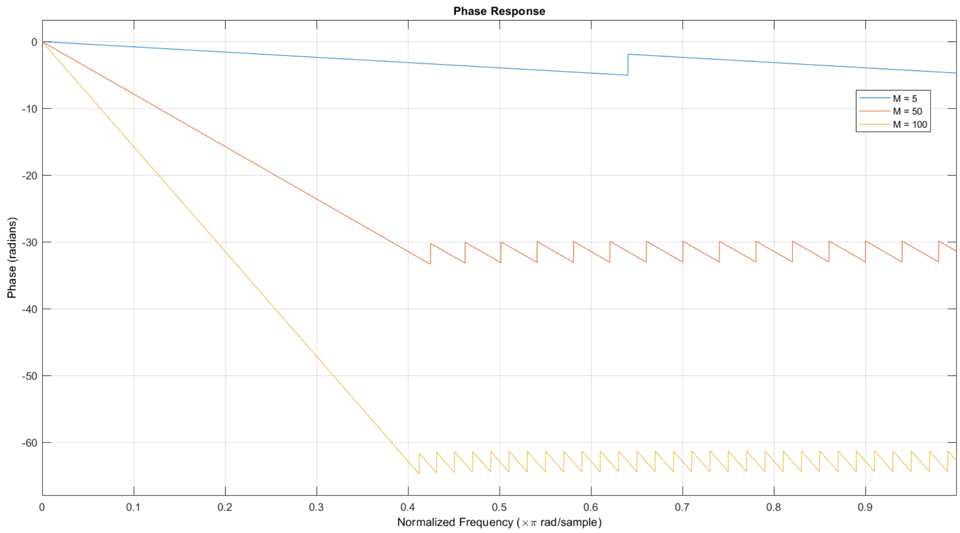

Figure 33: plots of phase responses for M = 5, 50, 100

iii. To improve the magnitude response of a lowpass filter, smoother windows have been designed by removing the discontinuities present in the rectangular window. Plot and explain the comparative characteristics of different window functions: Rectangular, Hanning, Hamming, and Blackman, for a window length $M = 50$.

- Morphology of the window

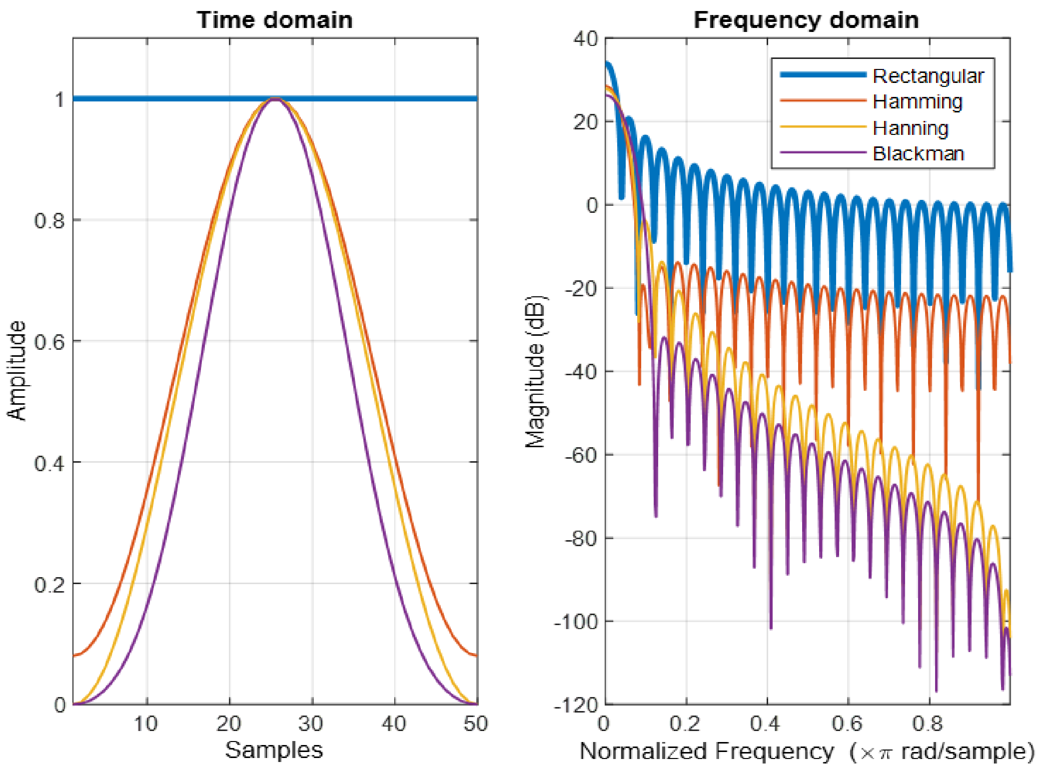

Figure 34: morpology of the 5 windows

The rectangular window has a constant height throughout its length, which causes sudden changes at the edges. In contrast, other window functions like Hanning and Blackman have smooth, bell-shaped curves that start and end at zero. This smoothness helps to reduce sudden changes and minimizes spectral leakage in the frequency domain, leading to better frequency resolution and filter performance.

- Magnitude response with a linear magnitude scale

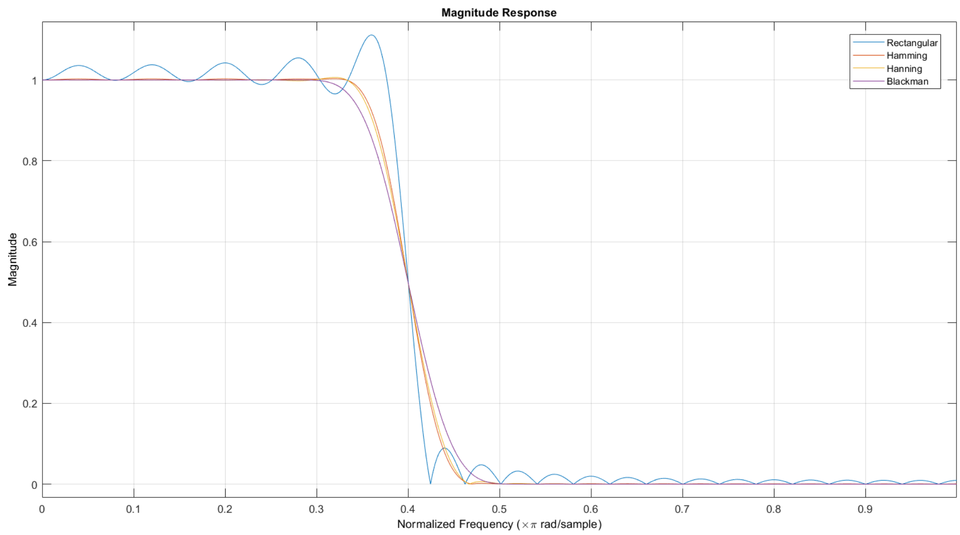

Figure 35: magnitude responses of the 5 windows in the linear scale

In alternative window functions, reduced ripples are observed in the passband, but this comes at the cost of a wider transition band. Compared to Hanning and Hamming windows, the Blackman window has a slightly wider transition band for the same length M. However, the half-power cutoff frequency remains the same for all three windows.

- Magnitude response with a logarithmic magnitude scale

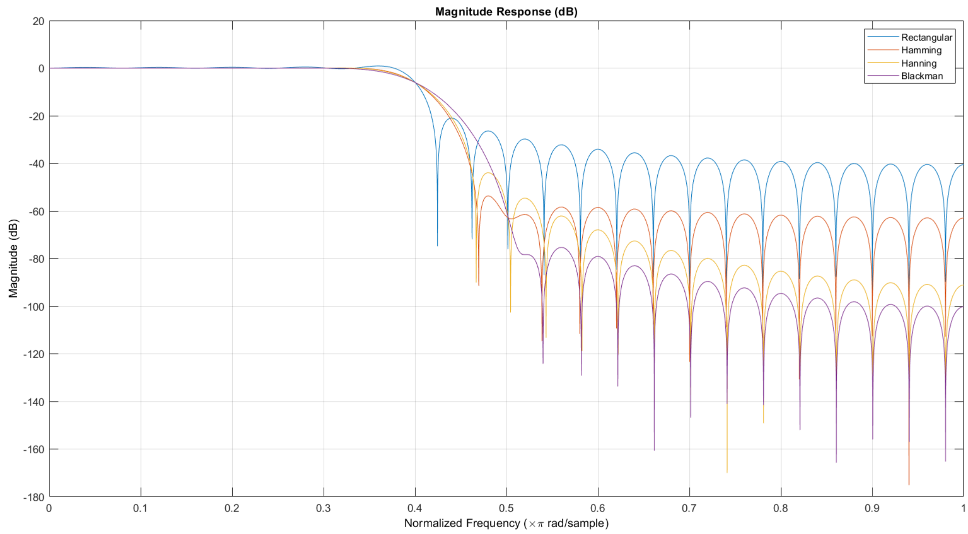

Figure 36: magnitude responses of the 5 windows in a logarithmic scale

- Phase response

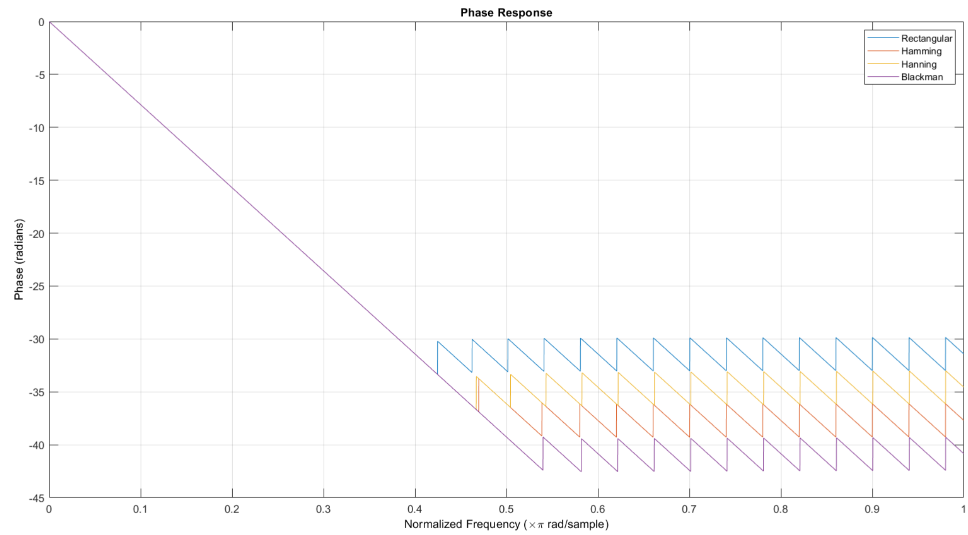

Figure 37: Phase responses of the 5 windows

All these window functions are symmetric, which results in a linear phase response in the passband region.

3.2 FIR Filter design and application using the Kaiser window

i. Plot the time domain signal and the power spectral density (PSD) estimate (or in other words frequency spectrum) of the signal

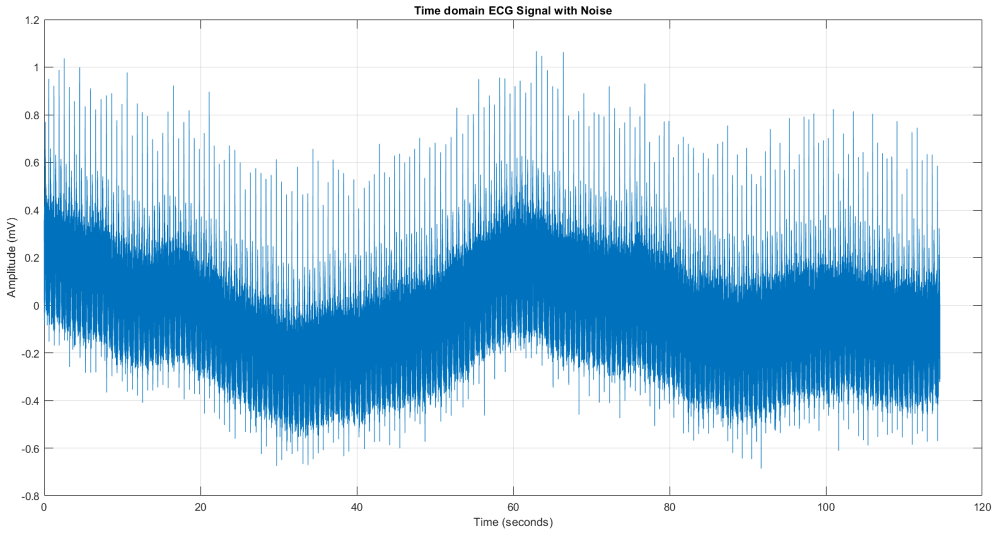

Figure 38: Time domain ECG signal

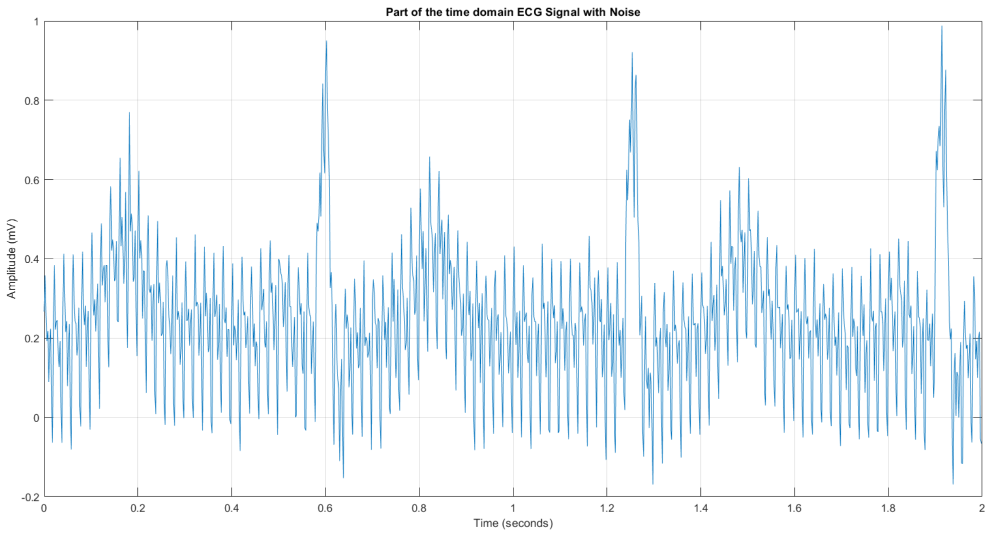

Figure 39: Part of the time domain ECG signal

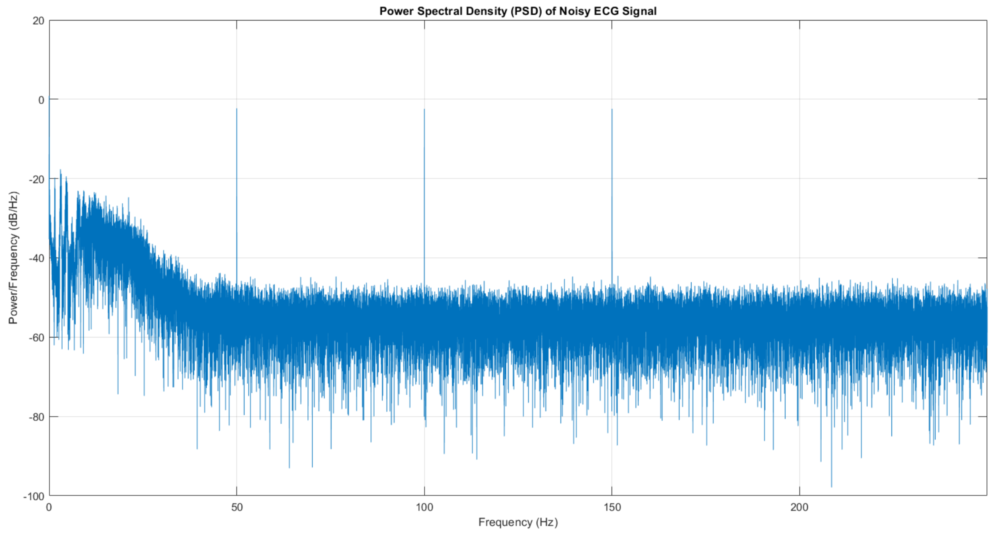

Figure 40: Frequency spectrum of the ECG signal

ii. Decide on parameters for the following filters that need to be applied on the above ECG

	Highpass	Lowpass
$f$pass (Hz)	1.5	125
$f$stop (Hz)	0.05	150
δ	0.01dB	0.01dB

	Stopband
$f$stop1 (Hz)	50
$f$stop2 (Hz)	100
$f$stop3 (Hz)	150

Justification for parameter selection
The main components of an ECG signal, such as the P-wave, QRS complex, and T-wave, typically lie within the range of 0.05 Hz to 150 Hz. The majority of significant signal energy is concentrated below 50 Hz as we can see in the ECG frequency spectrum.

Common noise sources affecting ECG signals are;

- **Powerline interference:** At 50 Hz or 60 Hz, depending on the region.
- **Baseline wander:** Caused by respiration and body movements, often below 0.5 Hz.
- **Electromyographic (EMG) Noise:** Higher frequency noise, usually in the range of 20 Hz to 150 Hz.
- **Motion artifacts:** Hard contaminations caused by electrode motions away from the skin.

Highpass Filter Design

- $f_{pass} = 1.5$ Hz: Captures essential low-frequency components, ensuring the P-wave and T-wave are preserved while blocking higher-frequency noise.
- $f_{stop} = 0.05$ Hz: Effectively removes baseline wander and slow drift caused by respiration or movement, stabilizing the ECG baseline.

Lowpass Filter Design

- $f_{pass} = 125\,\text{Hz}$: Preserves high-frequency components of the ECG, particularly the QRS complex, which typically lies above this frequency.
- $f_{stop} = 150\,\text{Hz}$: Further suppresses noise from muscle activity and electronic sources that could distort the ECG signal.

Ripple Justification

- Passband and Stopband Ripple = 0.01 dB: Ensures signal integrity by maintaining consistent peaks of the QRS complex and effectively suppressing noise without introducing distortion.

iii. Calculate the relevant β and M values for the highpass and lowpass filters

- **Highpass filter**
  - $M = 916$
  - $\beta = 4.090904$
- **Lowpass filter**
  - $M = 45$
  - $\beta = 4.090904$

iv. Visualise the windows (highpass and lowpass), magnitude response and the phase response of above filters

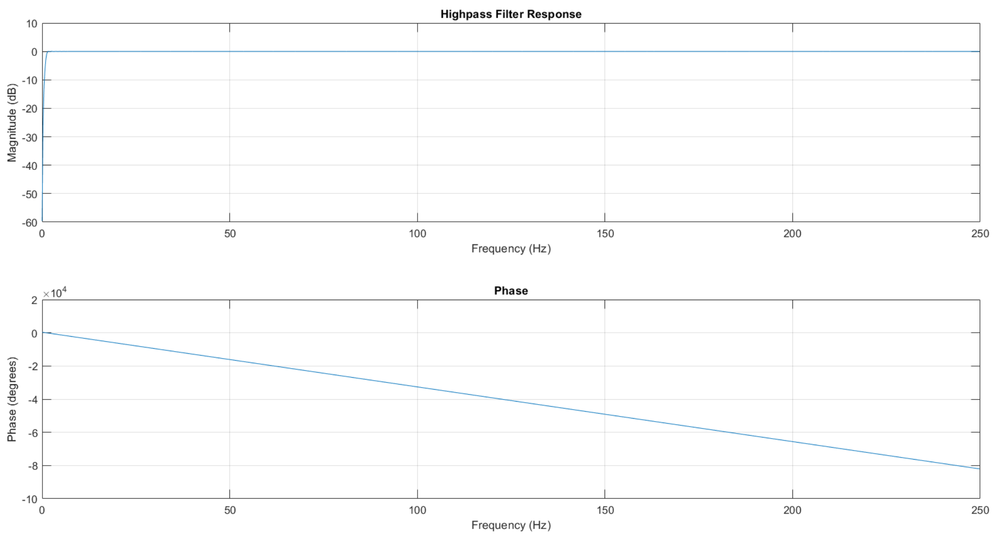

Figure 41: Highpass filter magnitude and frequency responses

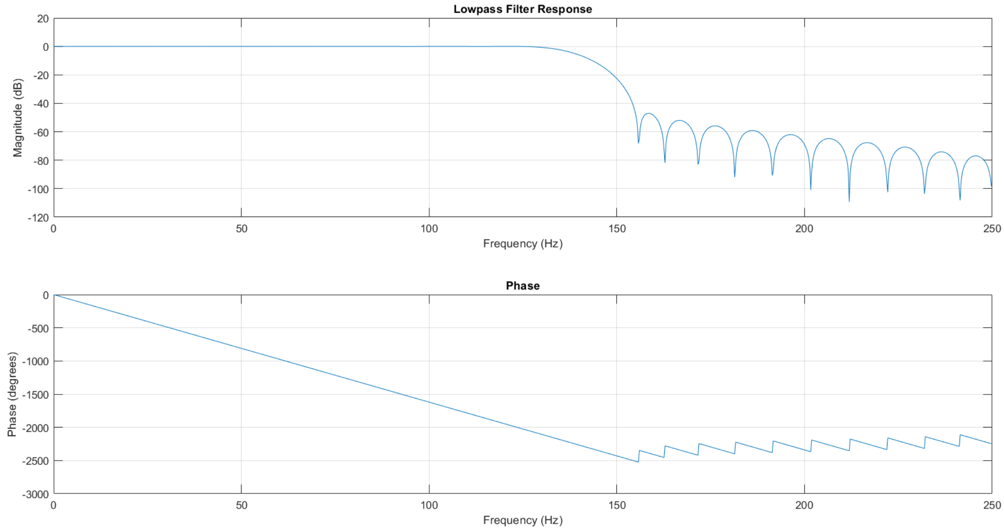

Figure 42: Lowpass filter magnitude and frequency responses

v. Apply these filters one by one to the *ECG_with_noise.mat* signal with necessary compensations for the group delay. Plot the effects in the time domain

- **Effect of the Highpass filter**

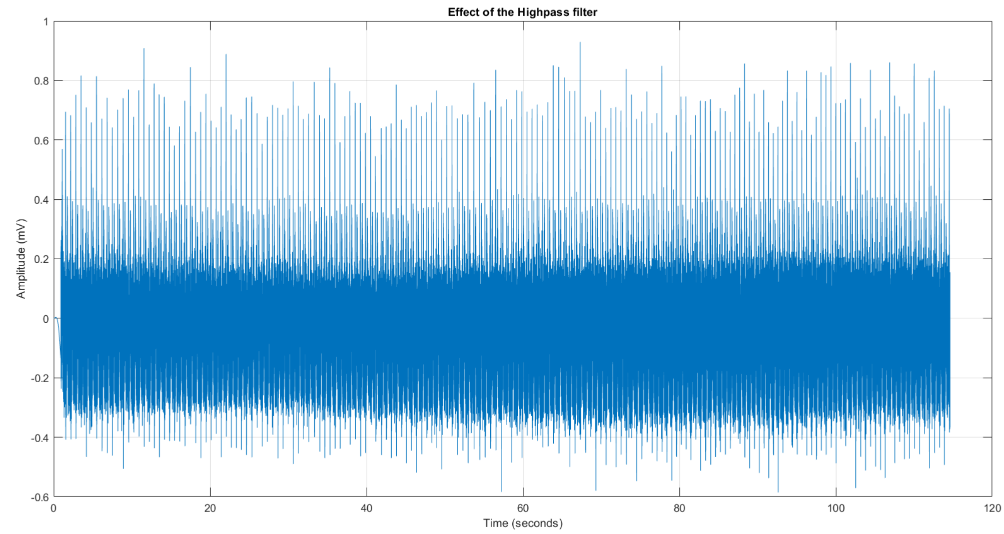

Figure 43: Effect of the Highpass filter on the ECG signal

Here we can see the baseline wander has removed after highpass filtering.

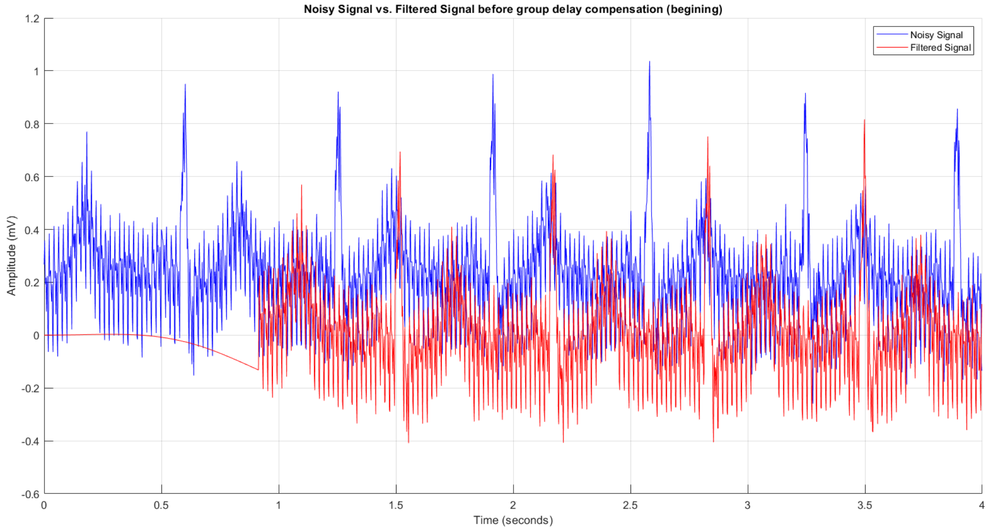

Figure 44: Effect of the Highpass filter on the ECG signal (Group delay)

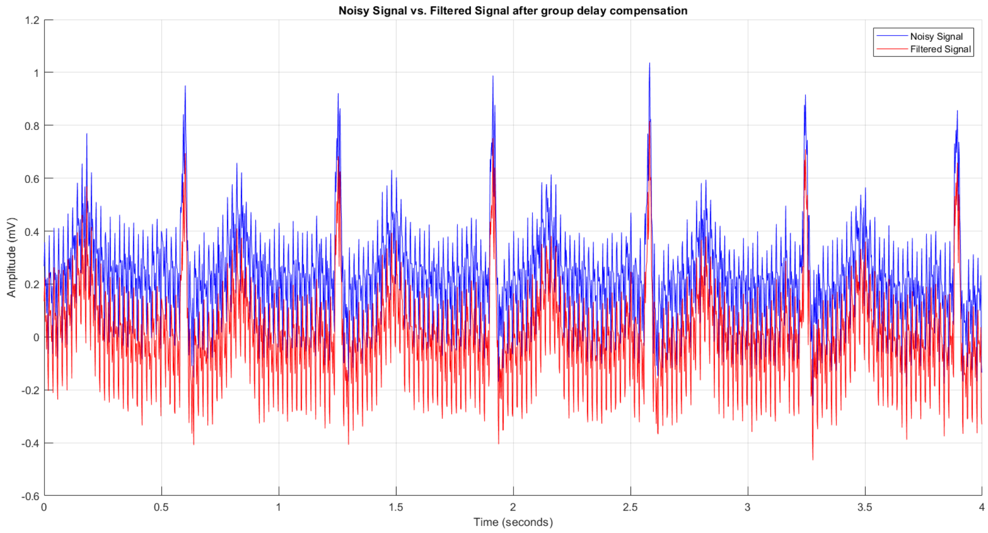

Figure 45: After compensate the group delay

- **Effect of the Lowpass filter**

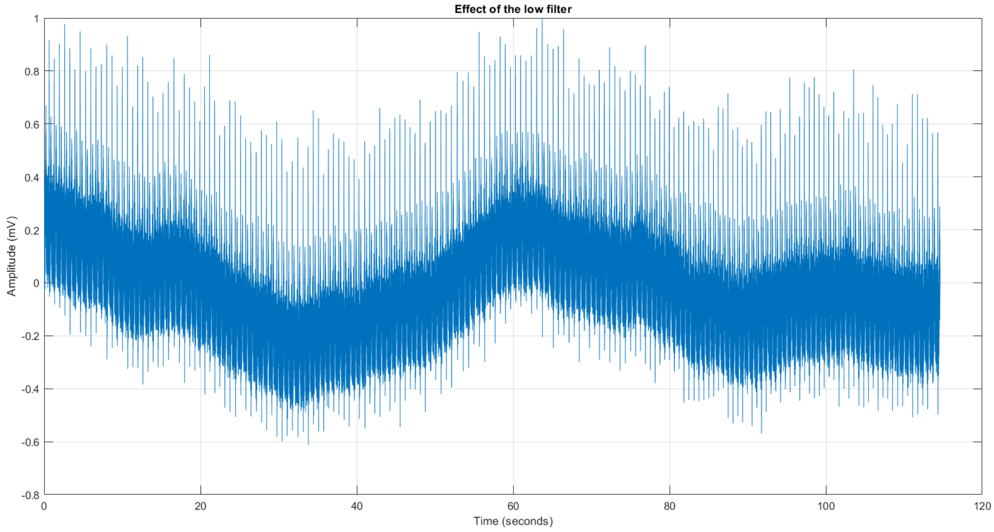

Figure 46: Effect of the Lowpass filter on the ECG signal

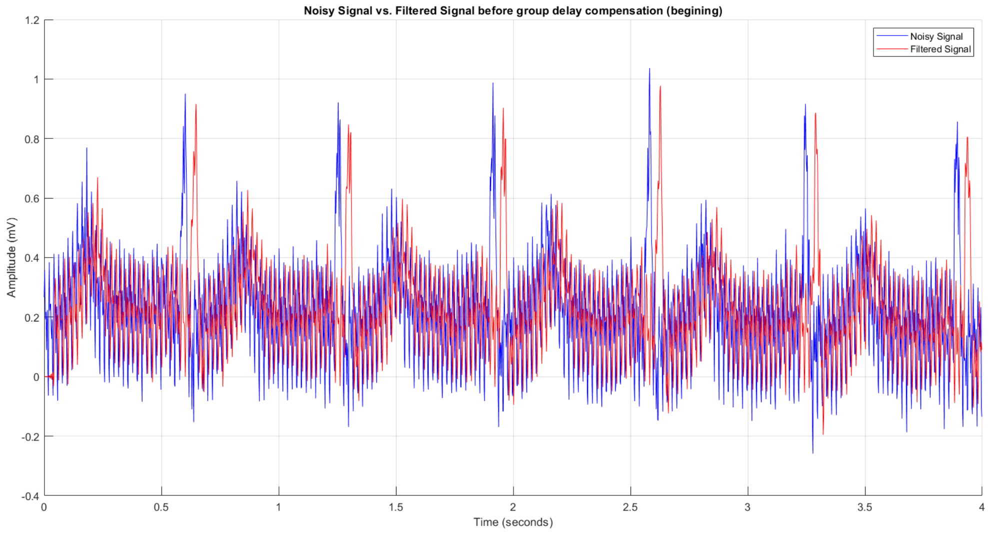

Figure 47: Effect of the Lowpass filter on the ECG signal (Group delay)

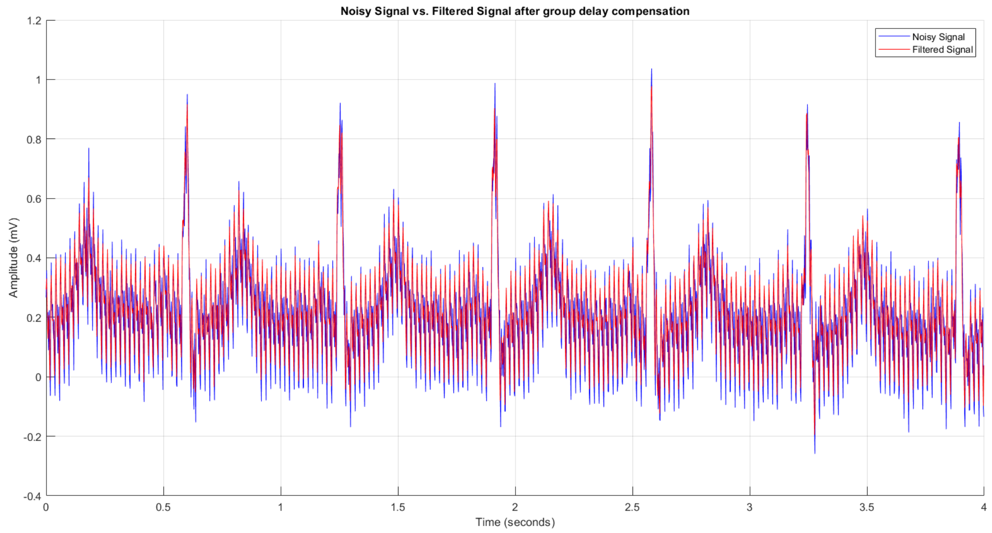

Figure 48: After compensate the group delay

- **Effect of the Comb filter**

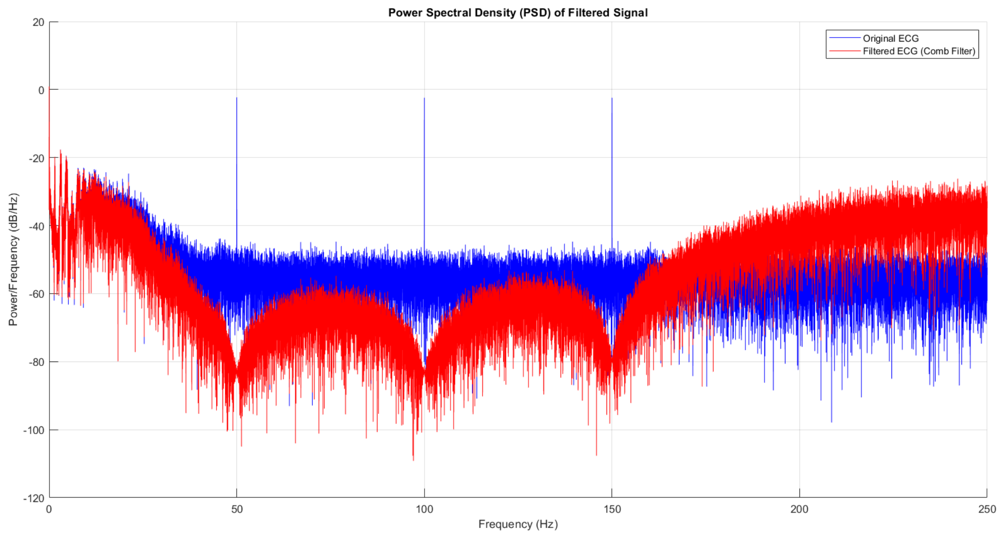

Figure 49: PSD of the Noisy ECG signal and filtered ECG signal using comb filter

Effect of after applying all 3 filters

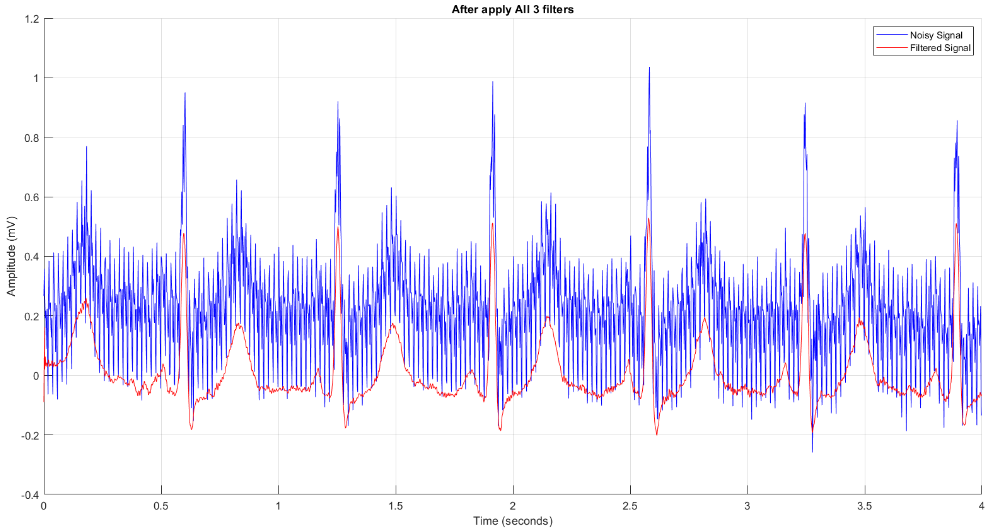

Figure 50: The filtered ECG signal by using all 3 filters and the Noisy ECG signal

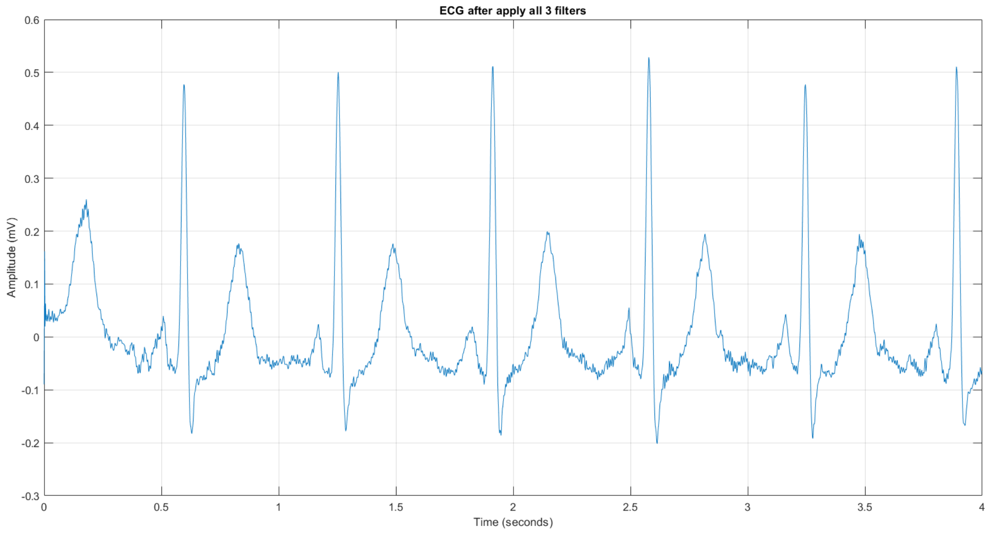

Figure 51: The filtered ECG signal by using all 3 filters

Here, we observe that some high-frequency noise remains present in the filtered ECG waveform. This noise likely resides within the 100 Hz to 150 Hz range. Since the cut-off frequency of our low-pass filter is set at 150 Hz, it has not effectively removed these noise components.

vi. Plot the magnitude response of the combined three filters and the PSD of the final filtered ECG

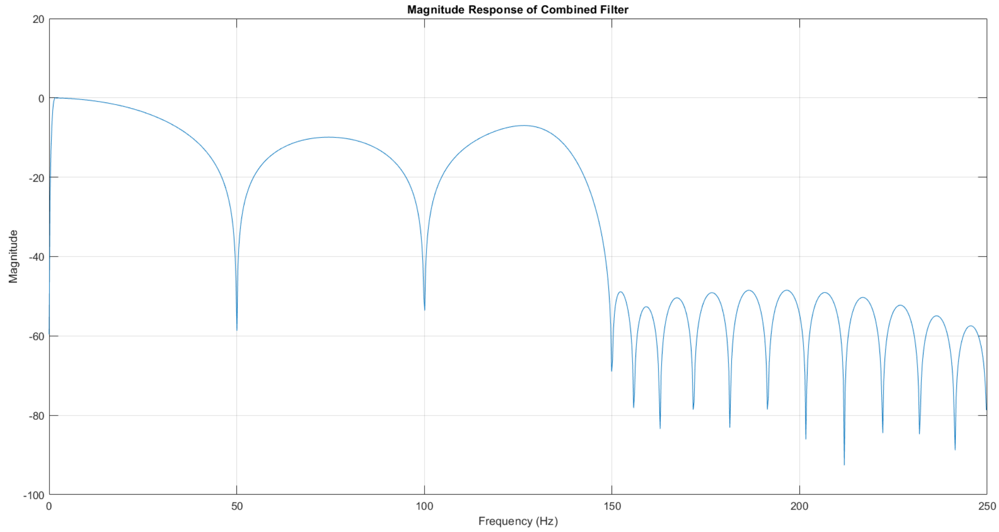

Figure 52: Magnitude response of the combined 3 filters

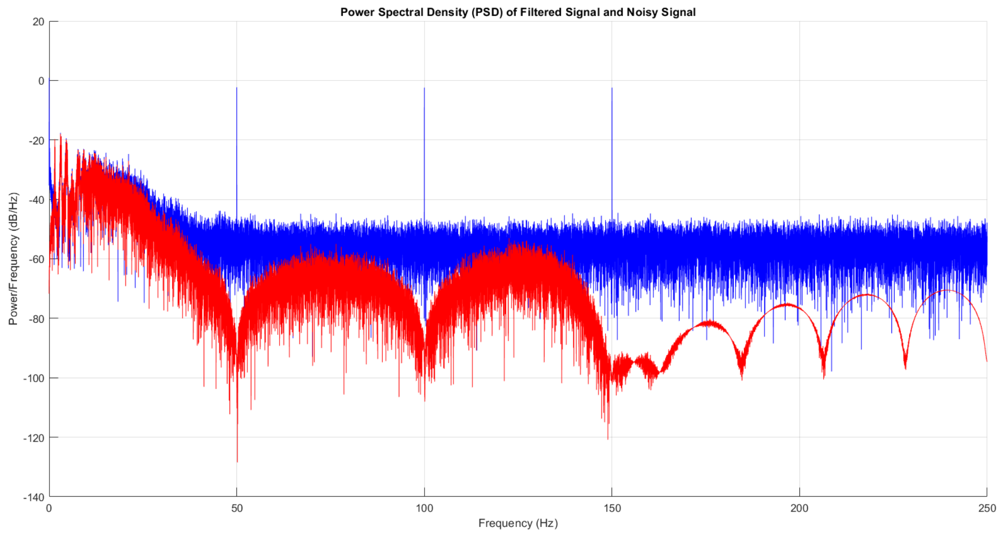

Figure 53: PSD of the Filtered signal and Noisy Signal

As we explained earlier, the power spectral density function clearly shows how the remaining high-frequency components behave.

4 IIR filters

4.1 Realising IIR filters

i. Obtain filter coefficients of a Butterworth lowpass filter with the same cut-off frequency and of the same order that was used to implement the FIR lowpass filter in the previous section. Use the MATLAB command *[b,a]=butter(n,Wn)*

- Filter order = 45
- cut-off frequency = 140Hz

ii. Visualise the magnitude response, phase response and the group delay

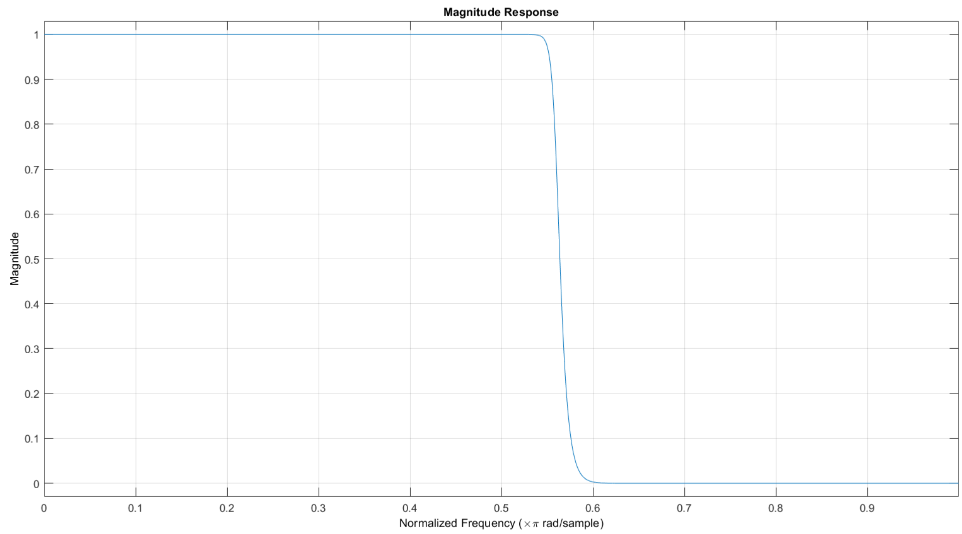

Figure 54: Magnitude response of the IIR lowpass filter

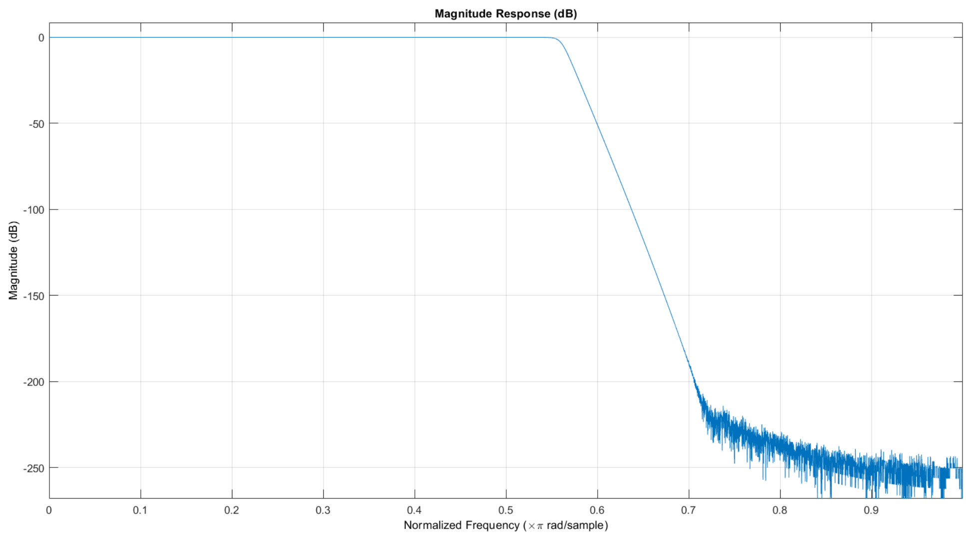

Figure 55: Magnitude response of the IIR lowpass filter (in dB)

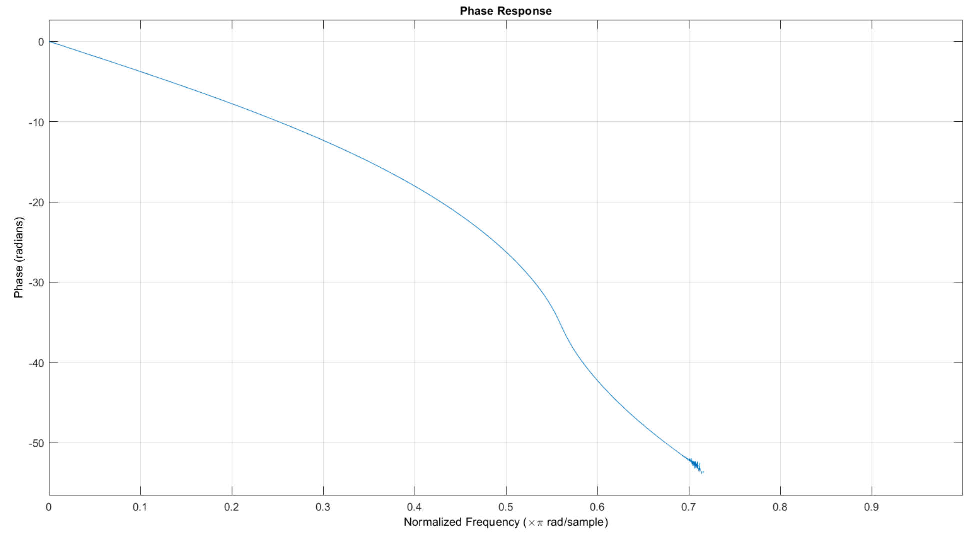

Figure 56: Phase response of the IIR lowpass filter

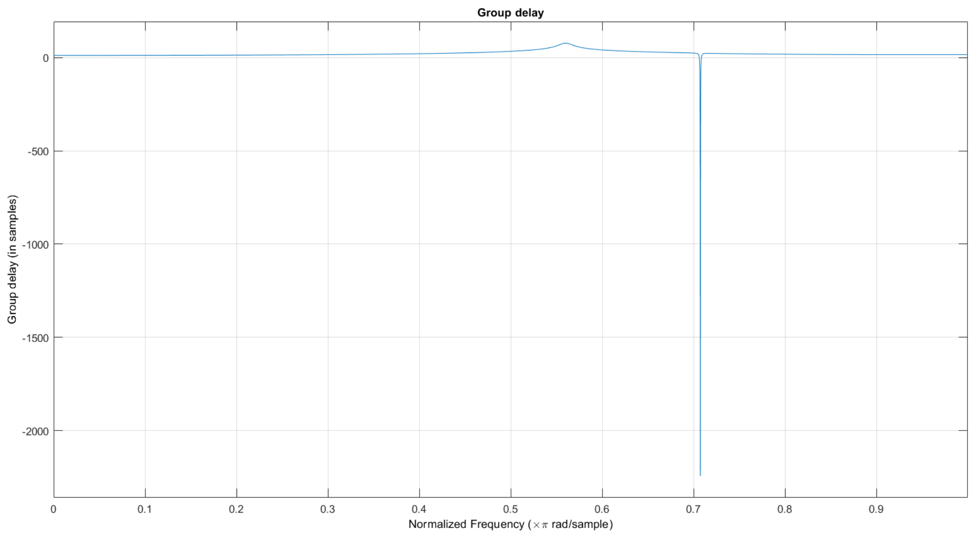

Figure 57: Group delay of the IIR lowpass filter (in dB)

iii. Obtain the filter coefficients of the highpass and the comb filter and visualize them

- **Highpass filter**
  - Filter order = 6
  - cut-off frequency = 1Hz

  Note: In the previous section, the highpass filter order was 916. But here, we can't go up to that much of higher order. Therefore, I took the filter order as 6.

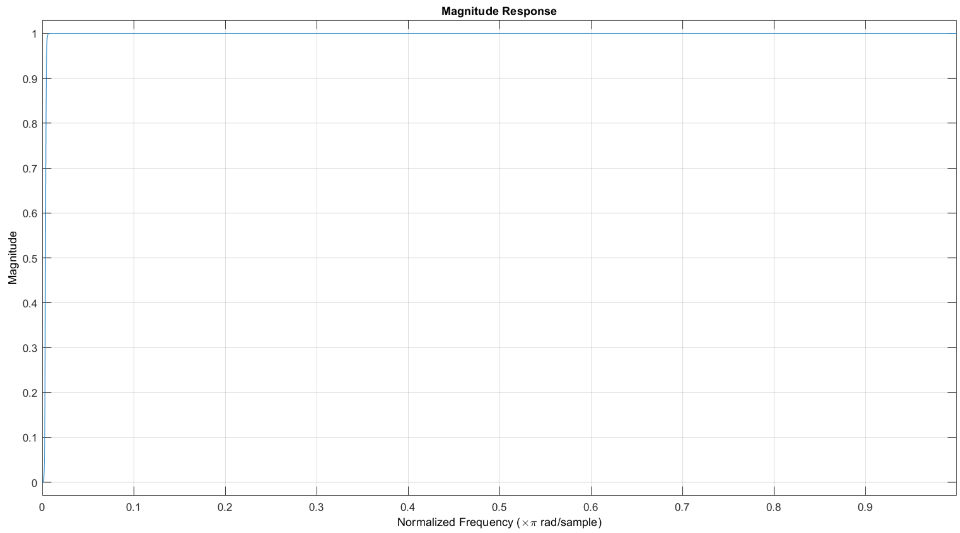

Figure 58: Magnitude response of the IIR highpass filter

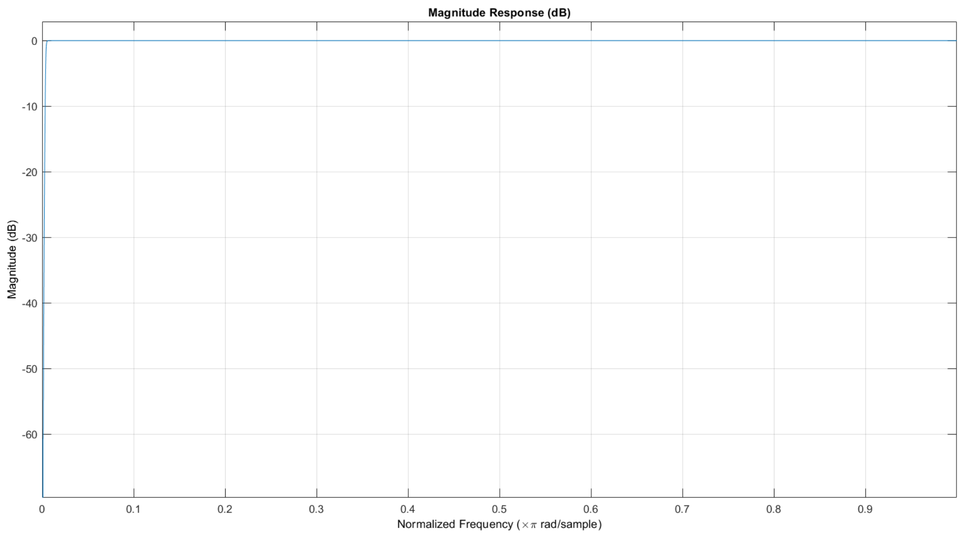

Figure 59: Magnitude response of the IIR highpass filter (in dB)

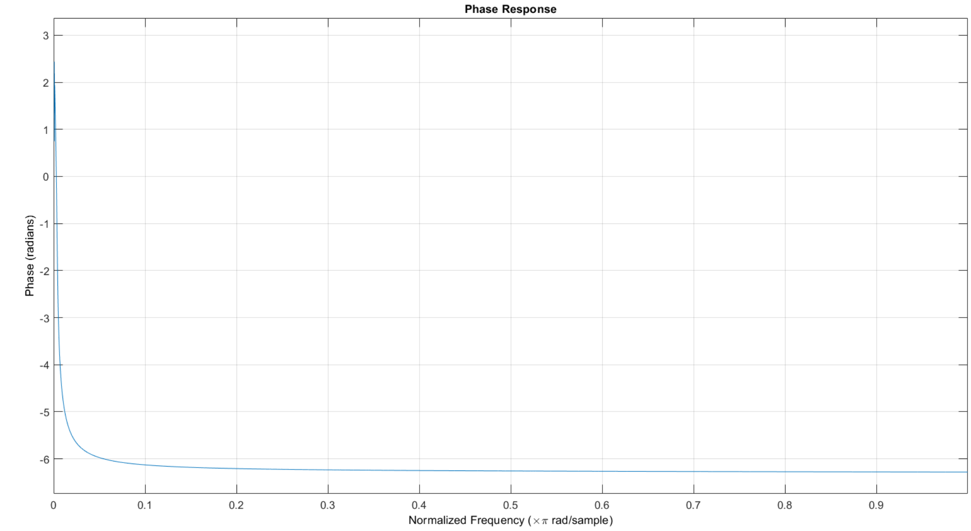

Figure 60: Phase response of the IIR highpass filter

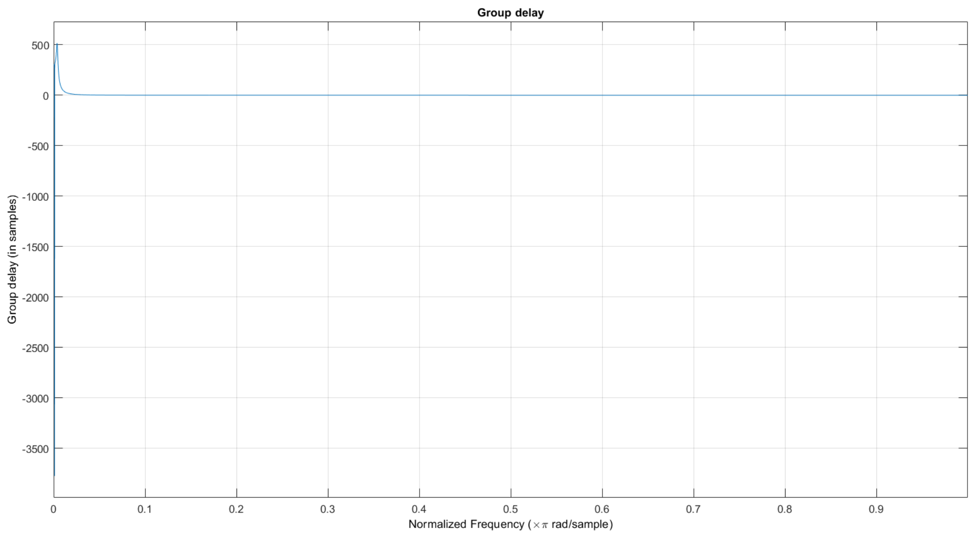

Figure 61: Group delay of the IIR highpass filter (in dB)

- **Comb filter**
  - cut-off frequencies = 50Hz, 100Hz, 150Hz

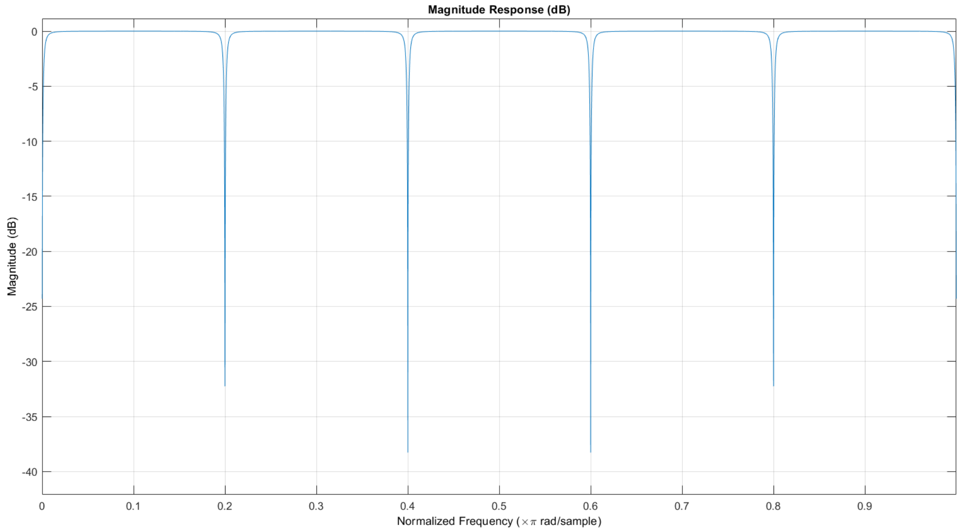

Figure 62: Magnitude response of the IIR comb filter

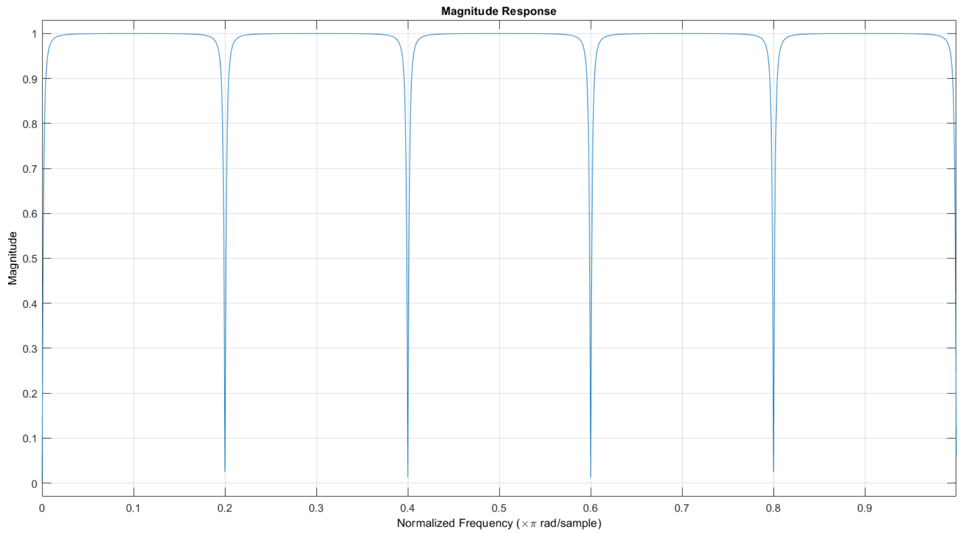

Figure 63: Magnitude response of the IIR comb filter (in dB)

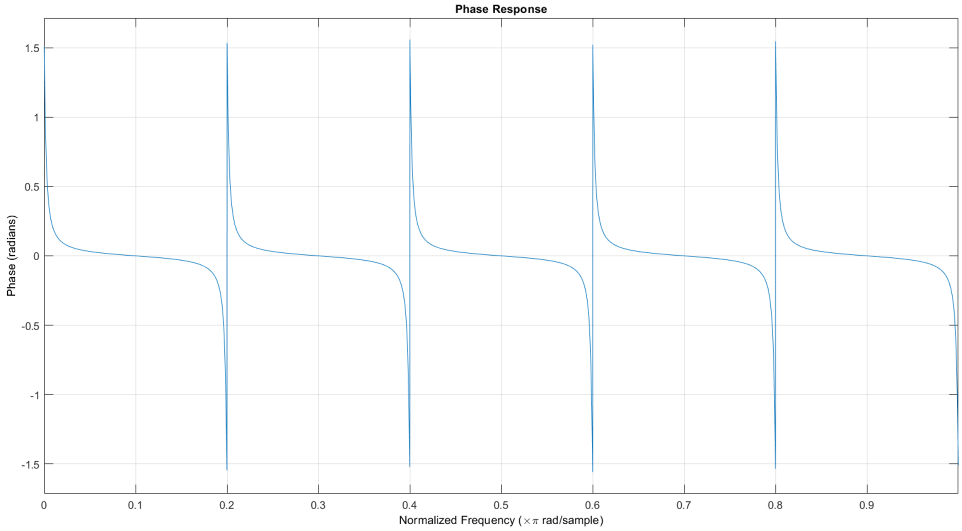

Figure 64: Phase response of the IIR comb filter

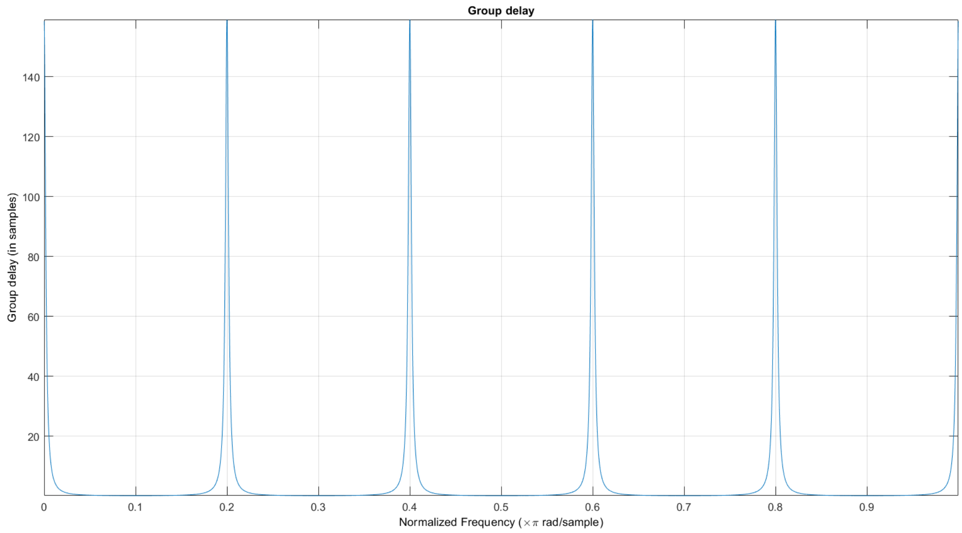

Figure 65: Group delay of the IIR comb filter (in dB)

iv. Plot the magnitude response of the combined three filters and compare it with the corresponding FIR filter plot

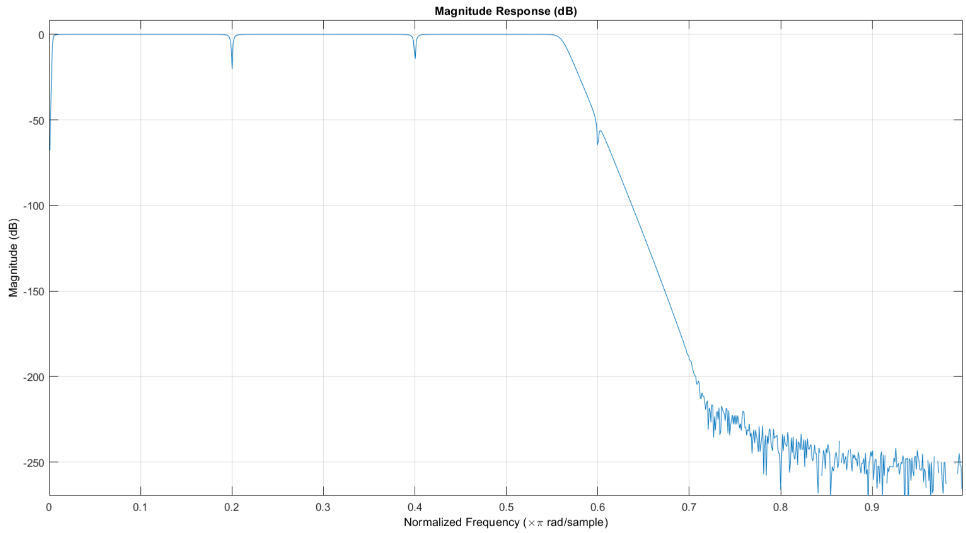

Figure 66: Magnitude response of the combined 3 IIR filters

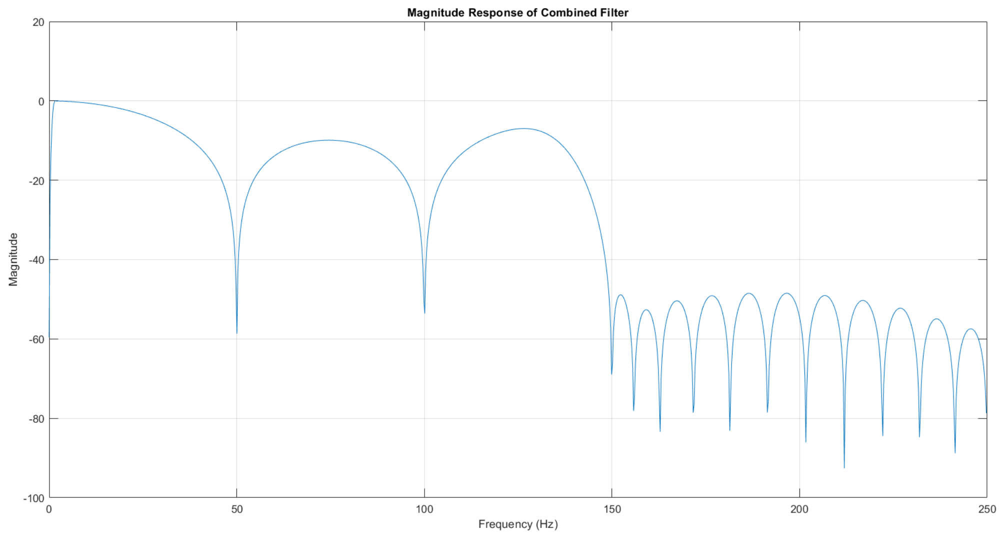

Figure 67: Magnitude response of the combined 3 FIR filters

The magnitude response of a Finite Impulse Response (FIR) filter exhibits large ripples, also known as lobes. This behavior occurs because the zeros of the FIR filter are positioned on the unit circle in the z-plane, while the poles are located at the origin. These zeros influence the filter's frequency response, creating pronounced variations in gain across different frequencies.

In contrast, the Infinite Impulse Response (IIR) filter has a different configuration. Its zeros are not constrained to the unit circle, which leads to a smoother magnitude response without the large ripples seen in FIR filters.

4.2 Filtering methods using IIR filters

i. To apply forward filtering, use the *filter(b,a,x)* command to filter the *ECG_with_noise.mat* signal with the IIR filters realised in the previous section

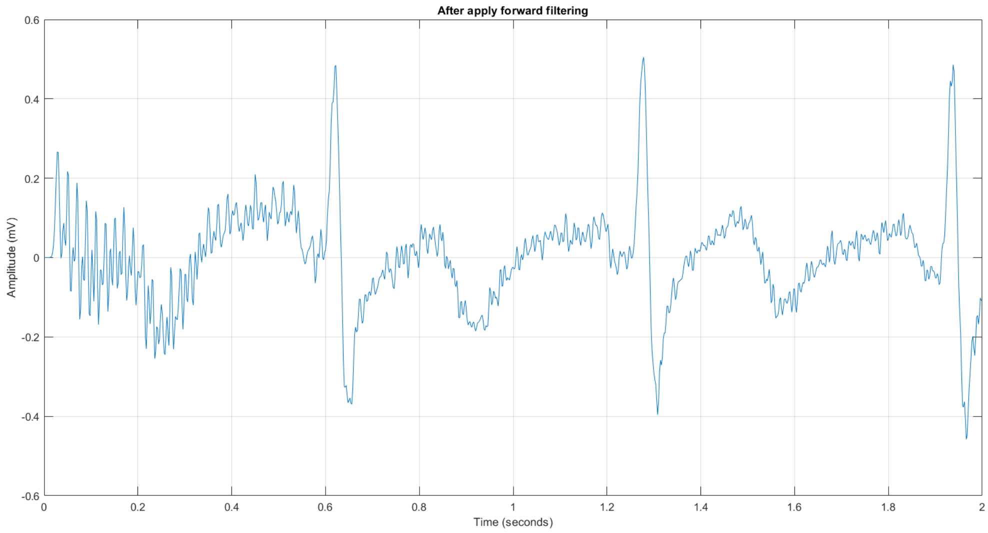

Figure 68: ECG signal after applying forward filtering

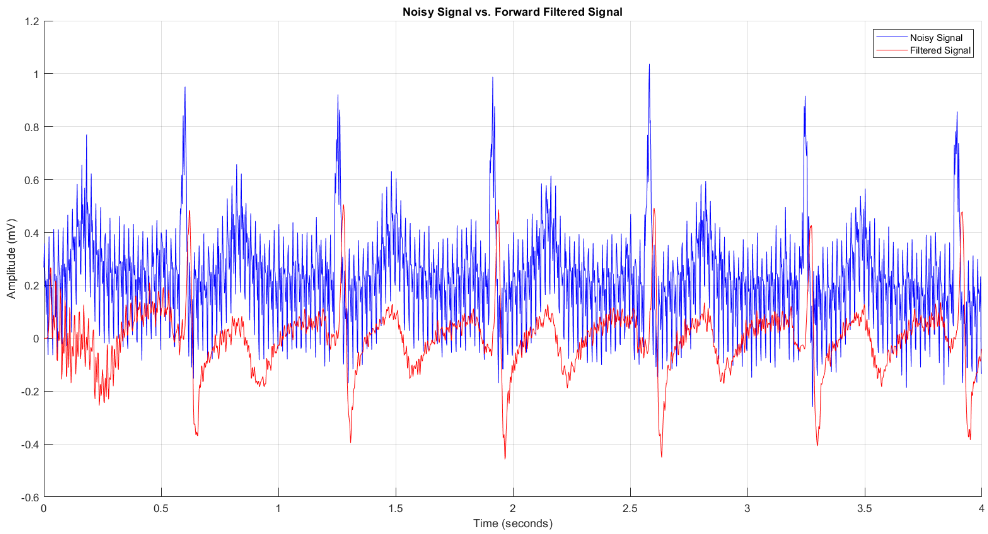

Figure 69: Noisy ECG signal and filtered ECG signal after applying forward filtering

ii. To apply forward-backward filtering use the *filtfilt(b,a,x)* command to filter the *ECG_with_noise.mat* signal with the IIR filters realized in the previous section.

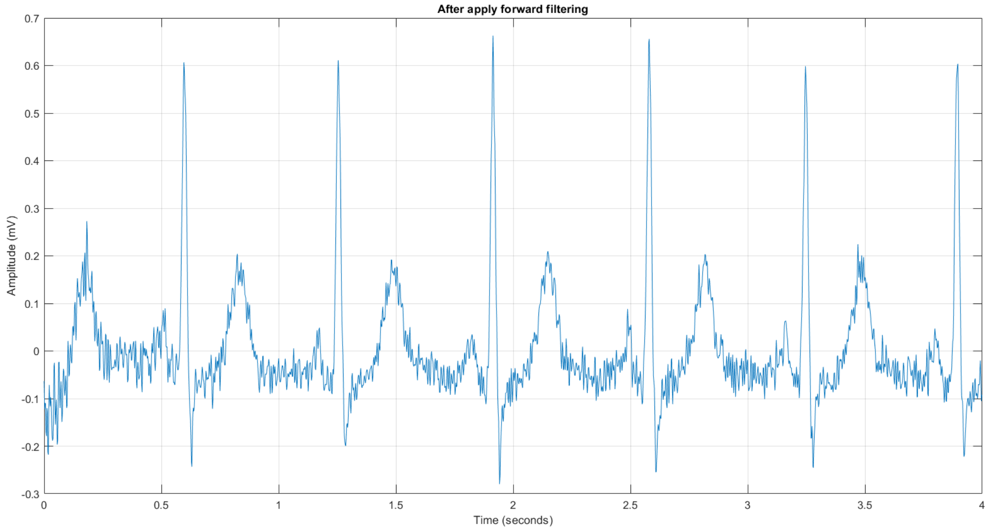

Figure 70: ECG signal after applying forward-backward filtering

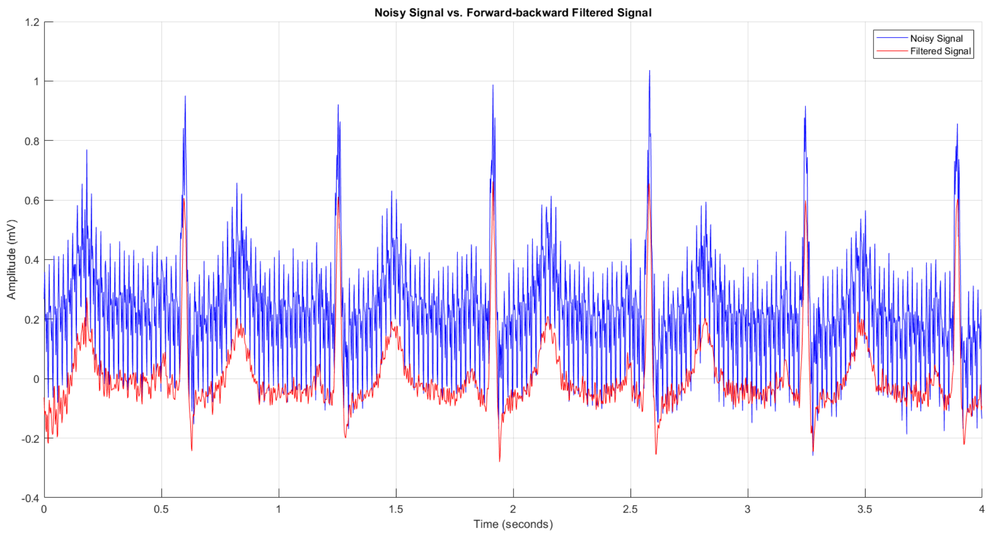

Figure 71: Noisy ECG signal and filtered ECG signal after applying forward-backward filtering

iii. Generate overlapping time domain plots of the FIR filtered ECG, IIR forward filtered ECG and IIR forward-backward filtered ECG. Compare the effect with respect to theoretical interpretations

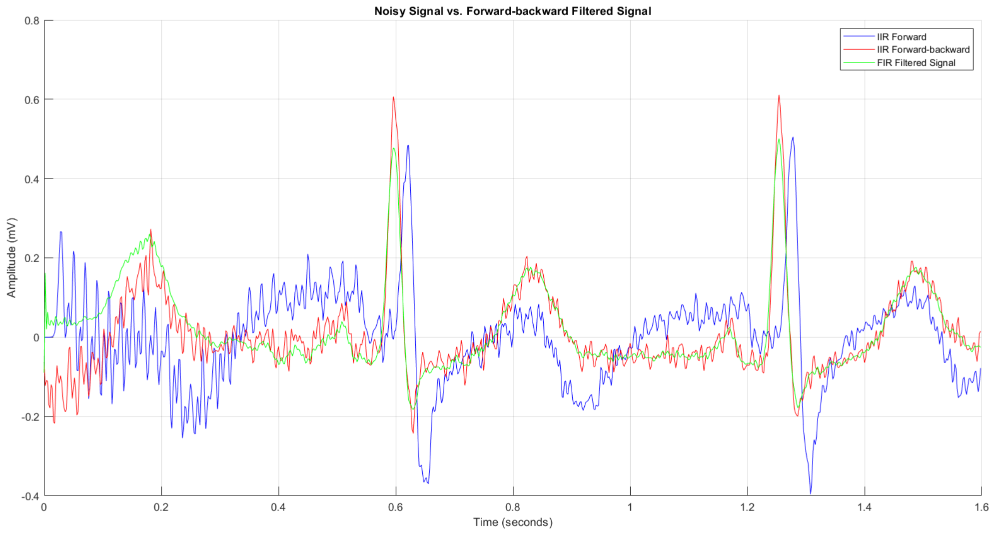

Figure 72: FIR filtered ECG, IIR forward filtered ECG and IIR forward-backward filtered ECG

Here mainly we can observe that the signal resulting from the forward IIR filtering has shifted to the right. This shift occurs because the filtering process introduces a delay to the signal, which varies across different frequencies. As a result, the shape of the forward IIR-filtered waveform has been changed.

After applying forward-backward IIR filtering, this group delay is compensated, and we can see the resulting signal. The forward-backward IIR-filtered ECG signal exhibits the proper characteristics of an ECG waveform, reflecting its typical features.

However, it is important to note that the FIR-filtered ECG waveform appears to be the cleanest, as it is mostly free of noise. In contrast, the IIR-filtered ECG signals still contain some high-frequency noise components when compared to the FIR-filtered waveform. This difference can be attributed to the design of the IIR low-pass filter, which has an order of 45. Such a relatively high order can lead to potential stability issues in the low-pass filter, resulting in the persistence of unwanted noise in the filtered signal.

iv. Generate overlapping plots of the PSDs of the FIR filtered ECG, IIR forward filtered ECG and IIR forward-backward filtered ECG and compare with respect to theoretical interpretations

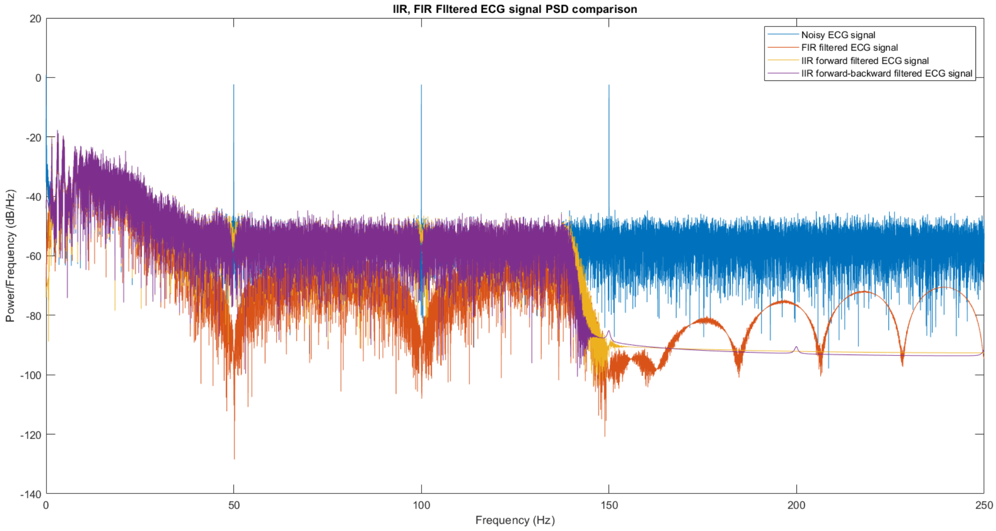

Figure 73: Overlapping plots of the PSDs of the FIR filtered ECG, IIR forward filtered ECG and IIR forward-backward filtered ECG

In the power spectral density plots, we can observe that both FIR and IIR filters have successfully suppressed unwanted frequencies. Notably, the forward-backward IIR filtering process demonstrates a much sharper transition compared to the other filtering methods.

And here we can observe that while FIR filters require very high orders to achieve a narrow transition band, IIR filters can achieve a similar effect with a lower order.