



**Department of Electronic &
Telecommunication Engineering**

University of Moratuwa
Sri Lanka

**EN4152 Assignment 1
Digital Filters**

210321X Kumarasinghe R D

27/08/2025

Contents

1 Smoothing Filters	2
1.1 Moving Average MA(N) Filter	2
1.1.1 Preliminaries	2
1.1.2 MA(3) filter implementation with a customised script	3
1.1.3 MA(3) filter implementation with the MATLAB built-in function	5
1.1.4 MA(10) filter implementation with the MATLAB built-in function	5
1.1.5 Optimum MA(N) filter order	8
1.2 Savitzky–Golay (SG) Filter	9
1.2.1 Application of SG(N,L)	9
1.2.2 Optimum SG(N,L) filter parameters	9
2 Ensemble Averaging	12
2.1 Signal with multiple measurements	12
2.1.1 Preliminaries	12
2.1.2 Improvement of the SNR	13
2.2 Signal with repetitive patterns	14
2.3 Segmenting ECG into separate epochs and ensemble averaging	15
3 Designing FIR filters using windows	18
3.1 Characteristics of window functions (use the fdatool)	18
3.2 FIR Filter Design and Application using the Kaiser Window	21
4 IIR filters	27
4.1 Realising IIR filters	27
4.2 Filtering methods using IIR filters	32

1 Smoothing Filters

1.1 Moving Average MA(N) Filter

The MA filter of length N is:

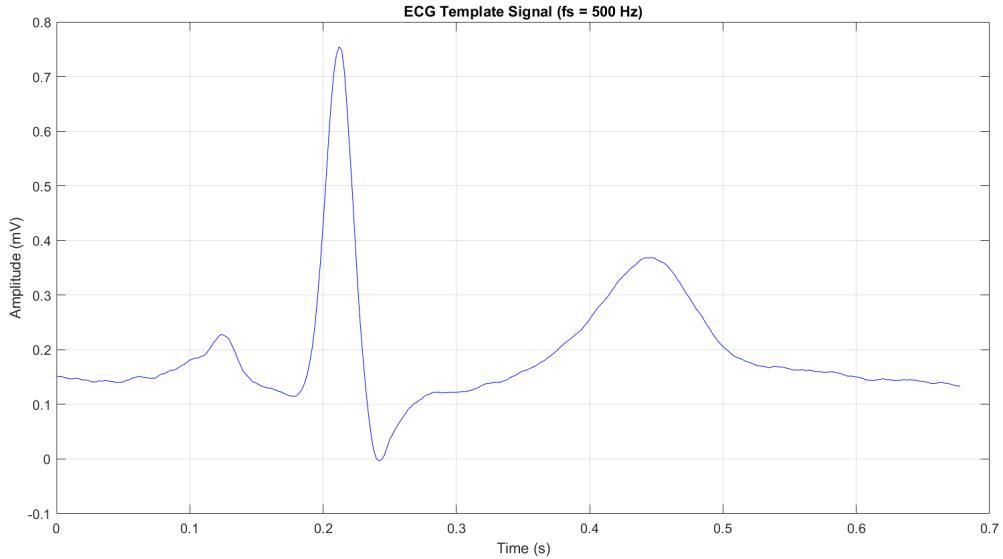
$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k], \quad H(e^{j\omega}) = \frac{1}{N} \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}}. \quad (1)$$

1.1.1 Preliminaries

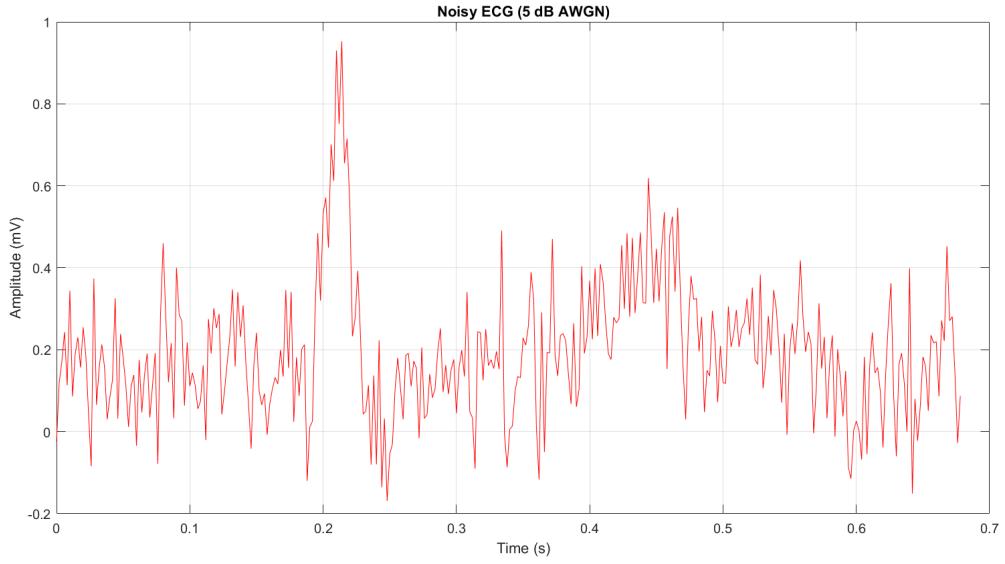
Observation of Typical ECG Characteristics (Before Noise Addition). Before applying noise or filtering, the clean ECG signal clearly demonstrates its fundamental components:

- The **P-wave**, a small and smooth upward deflection before the QRS complex, reflecting atrial depolarisation.
- The sharp and narrow **QRS complex**, the most dominant feature, corresponding to ventricular depolarisation. The R-peaks are the most easily identifiable fiducial points.
- The broader **T-wave**, following the QRS, associated with ventricular repolarisation.

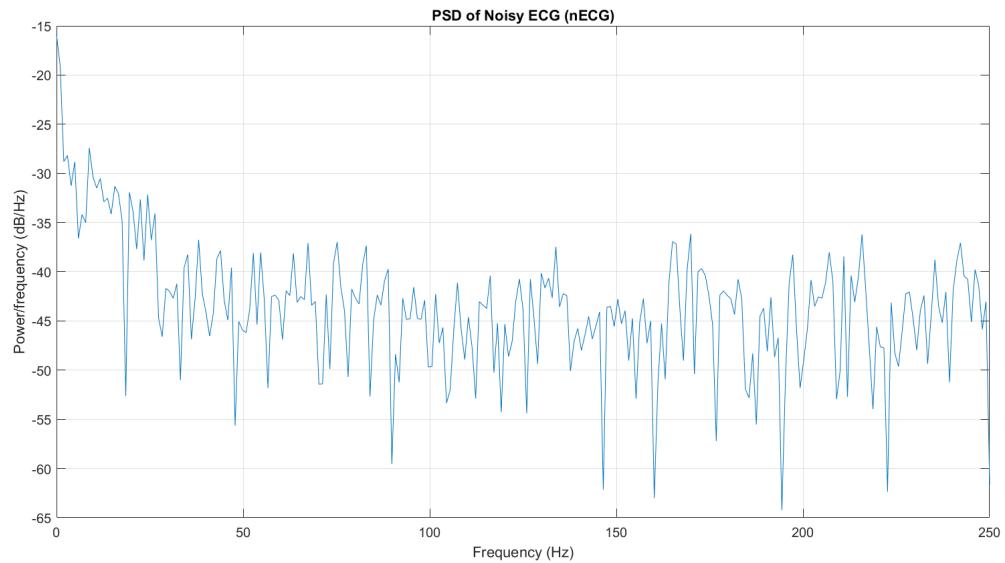
In subsequent sections, noise is added to simulate real-world acquisition conditions, and the task of the filters is to suppress the noise while preserving these morphological components as accurately as possible.



White Gaussian noise added signal.

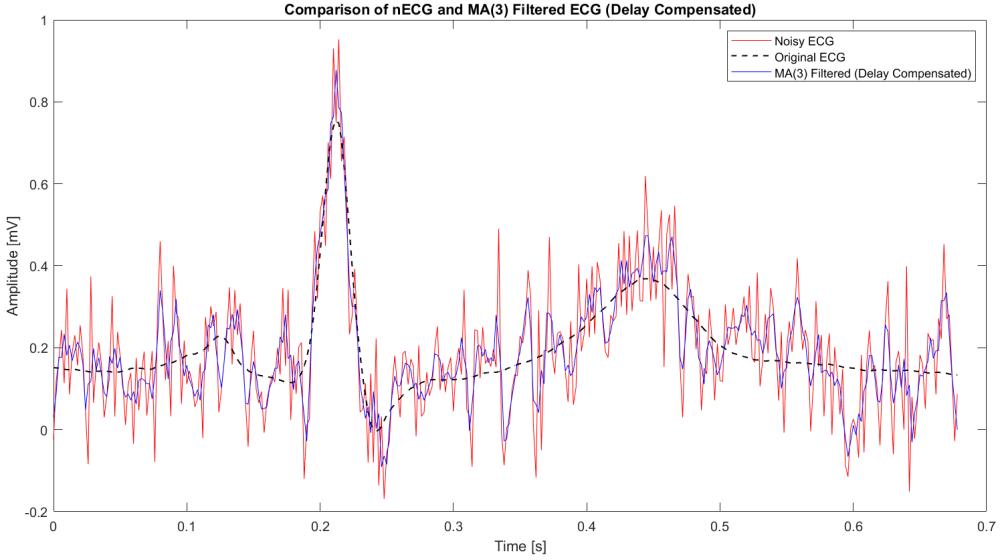


Power spectral density (PSD) estimate of the nECG.



1.1.2 MA(3) filter implementation with a customised script

Comparison of delay compensated `ma3ECG_1` and ‘nECG’



Observation. The signal has been smoothed, removing high frequency spikes to a greater extent. When both signals are plotted against time, several key differences become apparent:

- **Noise Reduction:** The `ma3ECG_1` signal is significantly smoother than the `nECG` signal. The high-frequency noise components, which cause rapid oscillations in `nECG`, are attenuated by the low-pass nature of the MA filter.
- **Delay Compensation:** The delay compensation process shifts the filtered signal by the filter's group delay to align it with the original. For a 3-point MA filter, the group delay is:

$$\tau_g = \frac{M - 1}{2} = \frac{3 - 1}{2} = 1 \text{ sample}$$

Therefore, the `ma3ECG_1` signal is aligned to start at the same time point as the '`nECG`' signal, making a direct comparison of their features possible. Without this compensation, the filtered signal would appear shifted forward in time relative to the original.

Comparison of PSDs of delay compensated `ma3ECG_1` and '`nECG`'

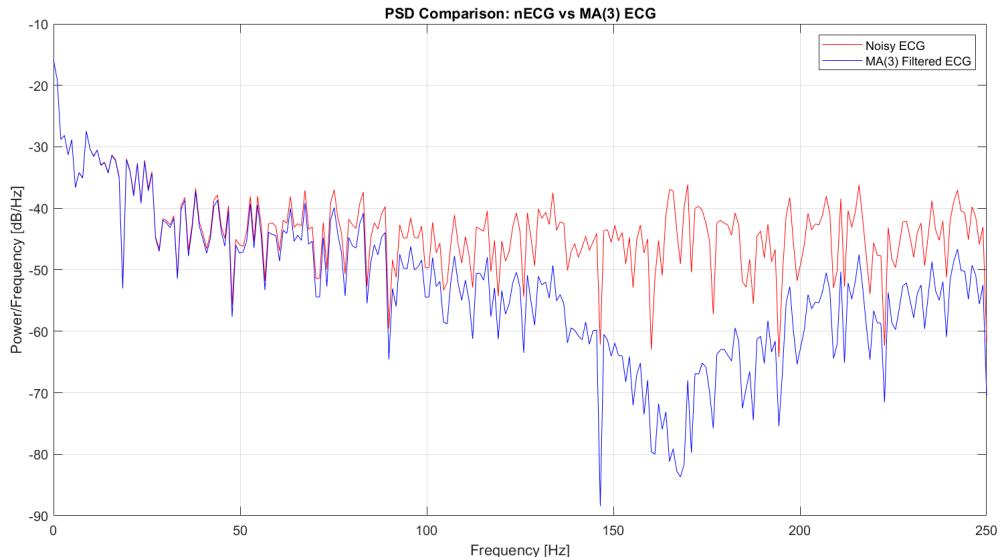


Table 1: Comparison of PSDs

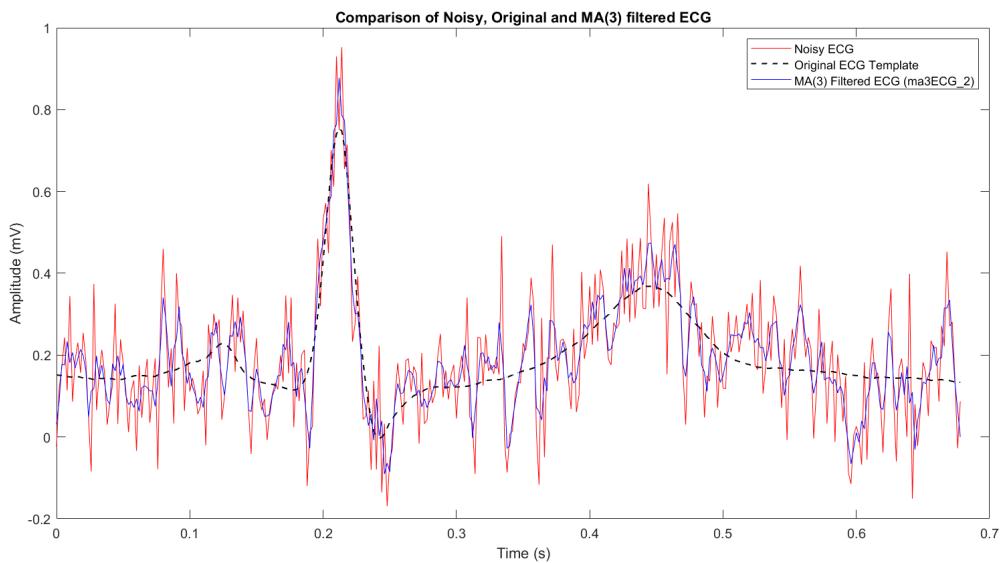
Characteristic	PSD of nECG	PSD of ma3ECG_1
High-Frequency Power	High	Significantly Attenuated
Total Signal Power	High (includes noise power)	Lower (noise power is reduced)

The table below summarizes the key differences in the PSDs of the two signals.

The PSD of the `ma3ECG_1` signal shows the effectiveness of the moving average filter in removing high-frequency noise, which is shown by the reduction in power at frequencies above the ECG's main bandwidth.

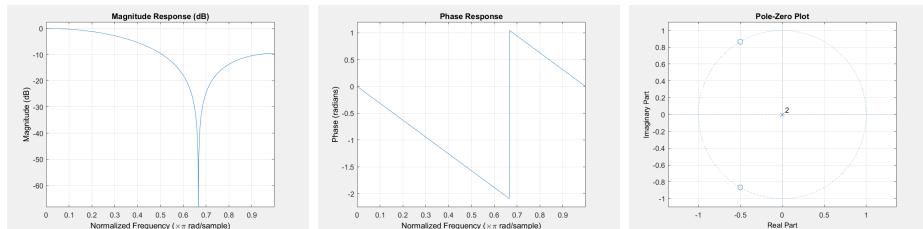
1.1.3 MA(3) filter implementation with the MATLAB built-in function

Comparison of nECG, ECG_template and ma3ECG_1



Observation. The `nECG` signal is corrupted by high-frequency noise, which shows as rapid fluctuations on the waveform. The `ma3ECG_2` signal is smoother, as the 3-point MA filter has effectively attenuated some of this high-frequency noise. The `ECG_template` is perfectly clean, with no noise artifacts.

Magnitude response, phase response and the polezero plot of the MA(3) filter



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

Improvement of the MA(10) filter over the MA(3) filter The Moving Average (MA) filter is a simple FIR filter that reduces noise by averaging a number of consecutive samples. The order of the filter (M) determines the degree of smoothing as well as the group delay introduced.

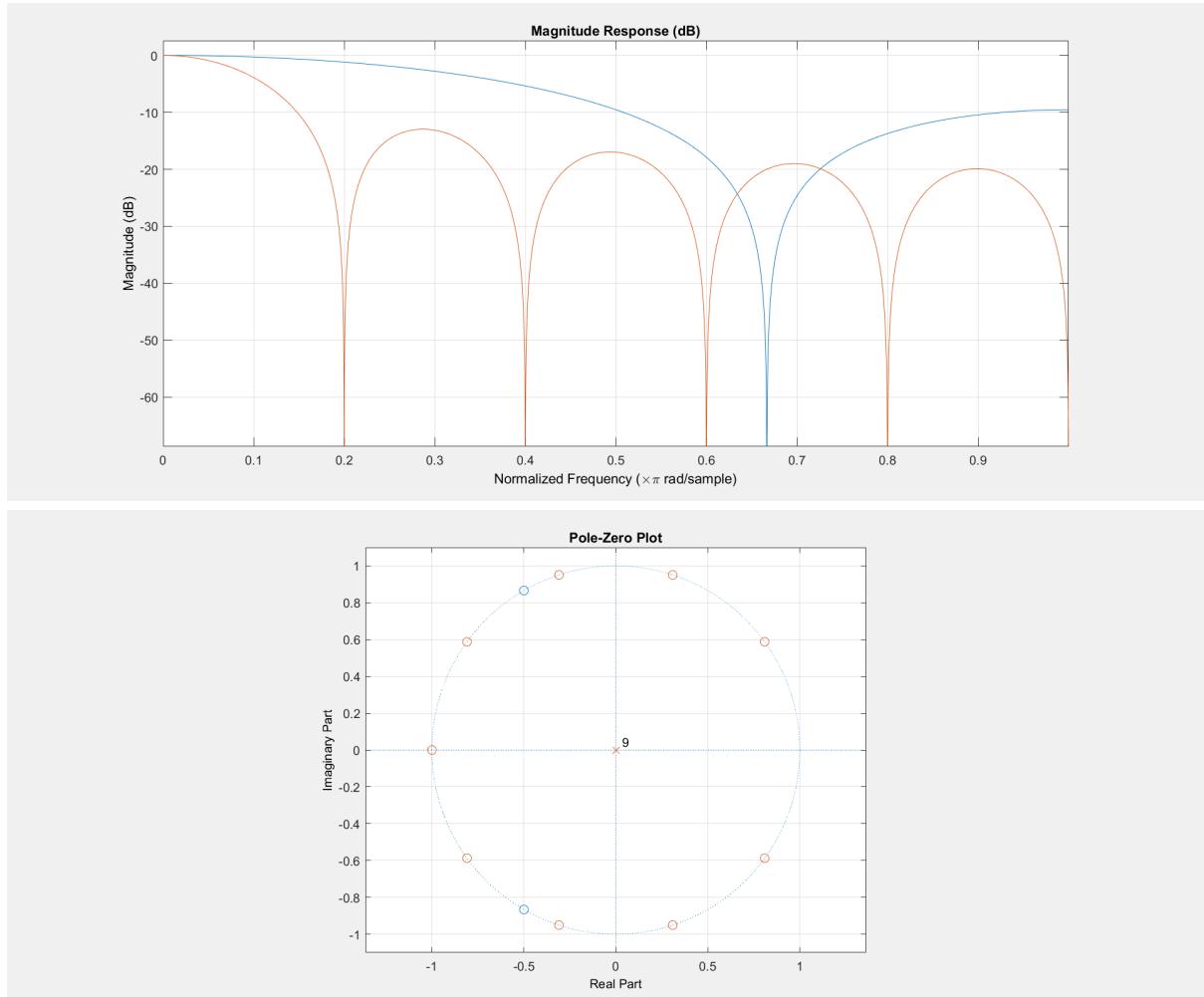
- **MA(3) Filter:**

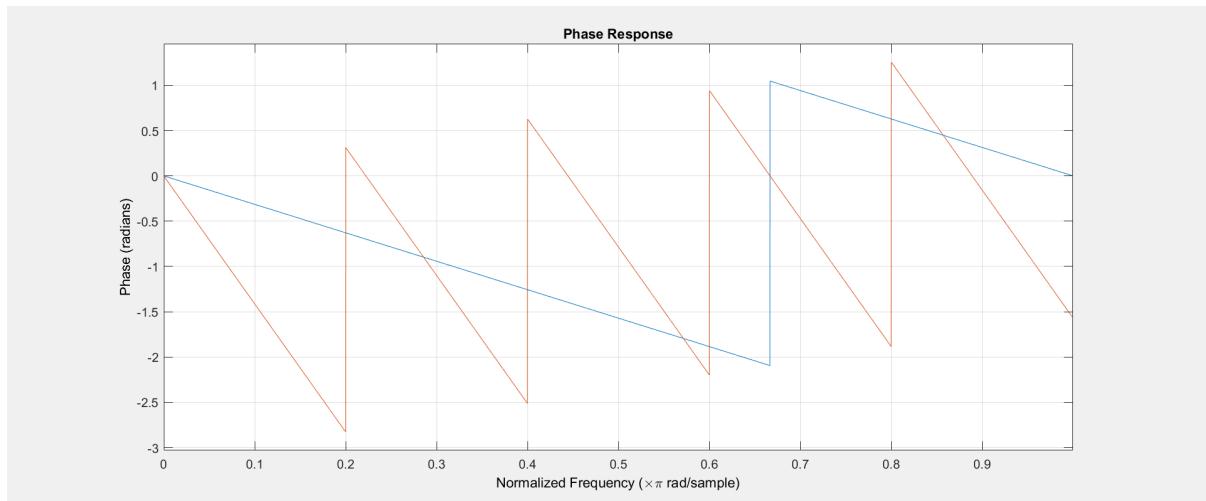
- Takes the average of 3 consecutive samples.
- Provides modest noise reduction while preserving most of the ECG waveform details.
- Group delay is $(M - 1)/2 = 1$ sample, which is negligible.
- However, high-frequency noise suppression is limited.

- **MA(10) Filter:**

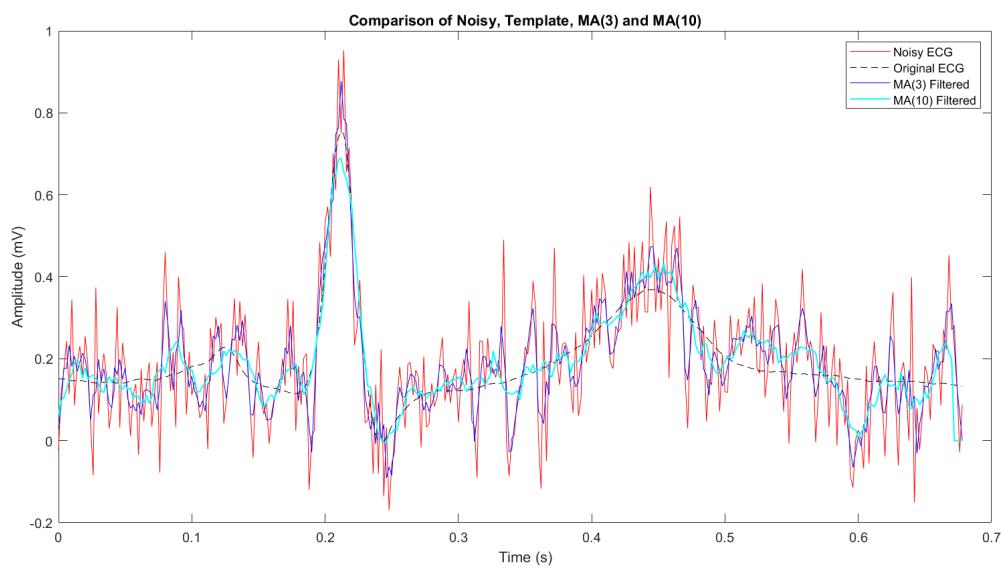
- Averages over 10 consecutive samples.
- Provides significantly better smoothing of noise compared to MA(3).
- Group delay is $(M - 1)/2 = 4.5$ samples, which introduces noticeable time shift.
- Risk of attenuating higher-frequency components of the ECG, such as sharp QRS complexes.

Conclusion: The MA(10) filter achieves greater noise suppression than the MA(3) filter, but at the cost of increased delay and potential distortion of fast-changing ECG features.





Comparison of nECG, ECG_template, ma3ECG_1 and ma10ECG

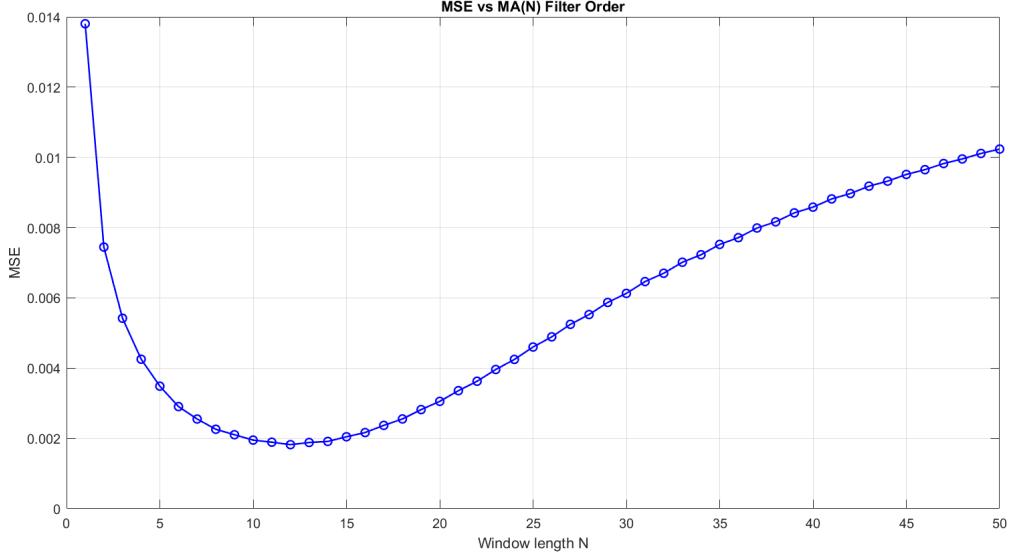


From the figure it is observed that:

- The MA(3) filter smooths the noisy ECG while preserving most of the QRS complex sharpness. However, some residual noise is still visible.
- The MA(10) filter achieves stronger noise suppression compared to MA(3), but it introduces more distortion to the fast-changing components of the ECG, such as slight broadening of the QRS complex.

1.1.5 Optimum MA(N) filter order

Determine the optimum filter order which gives the minimum MSE



By varying the filter order N and plotting MSE versus N , it was observed that:

- For very small N (e.g., $N = 2-3$), the filter does not sufficiently smooth the noise, resulting in high MSE values.
- As N increases, the MSE decreases, reaching a minimum at an intermediate filter order. This order corresponds to the best trade-off between noise suppression and preservation of the ECG waveform morphology.
- Beyond this optimum range, increasing N further causes the MSE to rise again, since the filter begins to excessively smooth the ECG and distort clinically important features such as the sharp QRS complex.

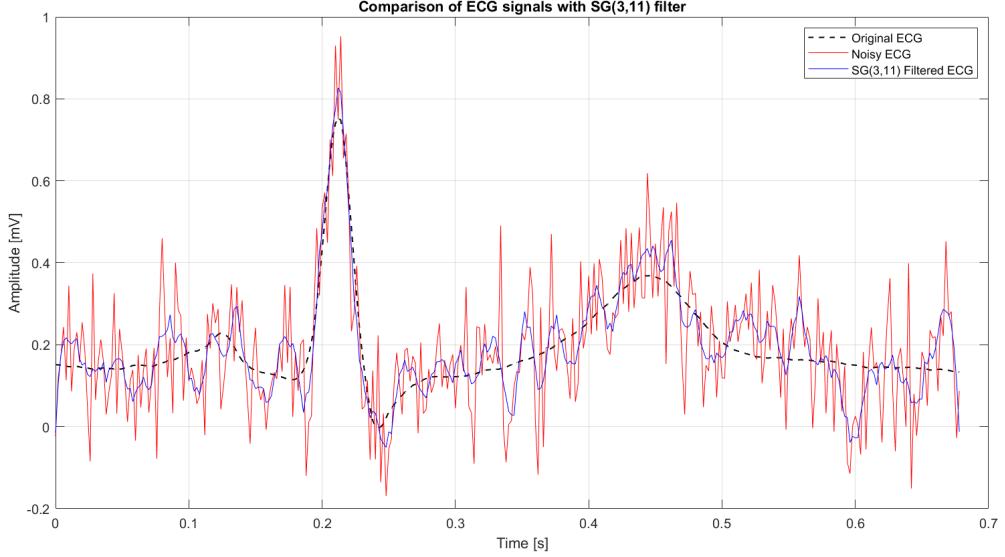
The optimum order $N = 12$, where the MSE reached its minimum.

Reason/s for large MSE values at low and high filter orders

- At **low filter orders**, the averaging window is too small to suppress the random noise effectively. Hence, the filtered signal still contains a significant noise component, resulting in a large deviation from the noise-free reference signal.
- At **high filter orders**, the long averaging window introduces excessive smoothing. While noise is reduced, the sharp transitions in the ECG, such as the QRS complex, become distorted and broadened. This distortion increases the deviation from the reference waveform, thereby increasing the MSE.

1.2 Savitzky–Golay (SG) Filter

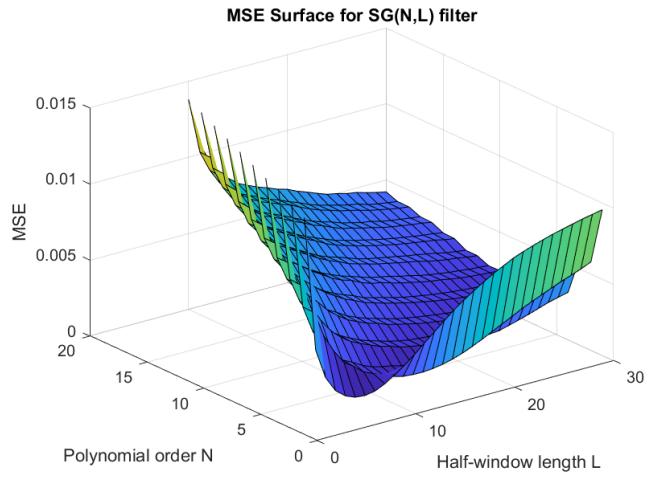
1.2.1 Application of SG(N,L)



The Savitzky–Golay (SG) filter of polynomial order $N = 3$ and frame length $L = 11$ was applied to the noisy ECG signal ($nECG$) using the `sgolayfilt` command. The filtered signal, denoted as $sg310ECG$, was delay compensated and plotted together with the $nECG$ and the noise-free $ECG_template$. The SG(3,11) filter successfully reduced noise while preserving the sharp features of the QRS complex better than the equivalent Moving Average filter.

1.2.2 Optimum SG(N,L) filter parameters

Optimal SG(N,L) filter

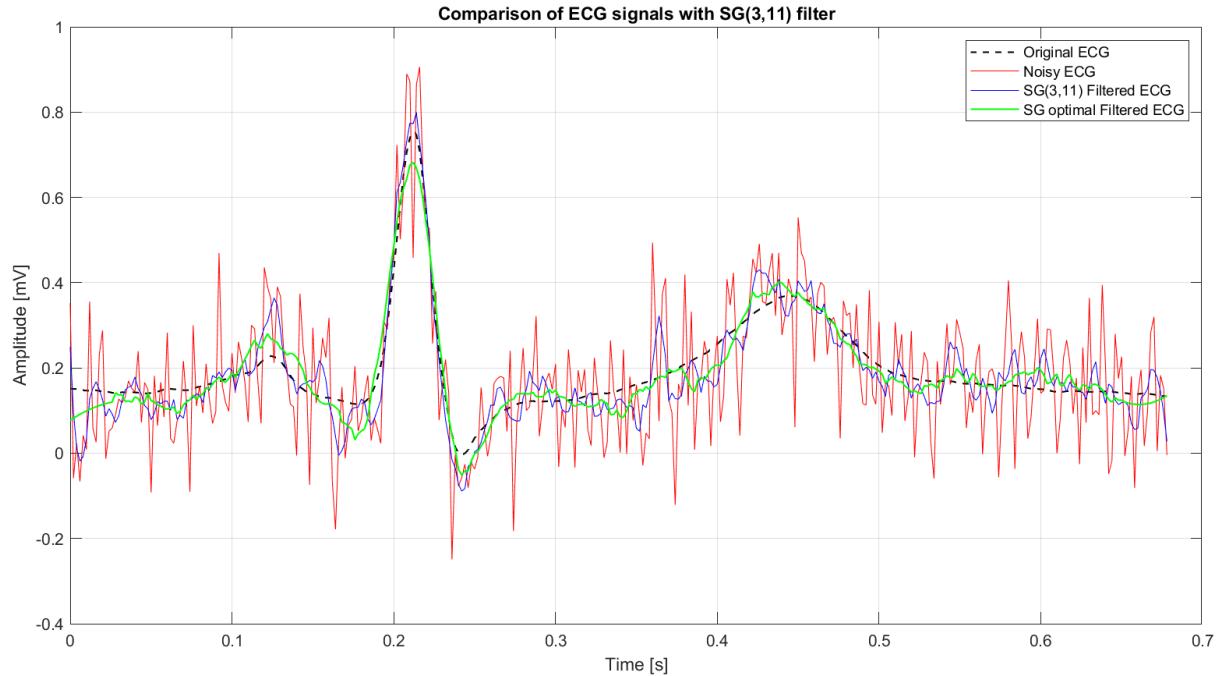


The mean squared error (MSE) was computed for a range of filter parameters (N, L) to determine the optimum filter configuration. A surface plot of MSE versus (N, L) was generated using `surf` or `pcolor`, and the minimum MSE point was identified. The optimum filter parameters which gives the minimum MSE are for $N = 2$ and $L = 13$. This optimum filter provided the best trade-off between noise reduction and waveform fidelity.

The ECG_template, the output of SG(3,11), and the output of the optimum SG(2,13) filter were plotted on the same figure for comparison. The optimum filter demonstrated superior preservation of the ECG morphology while further reducing the noise compared to SG(3,11).

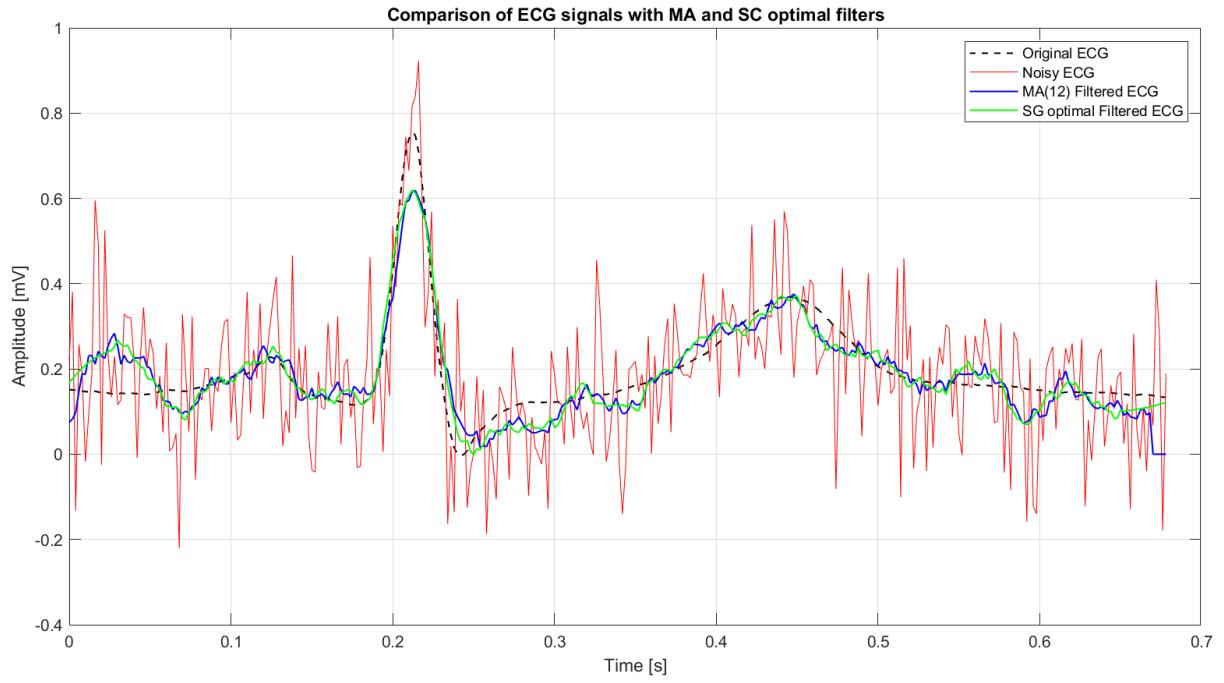
Comparison of delay compensated sg310ECG, the signal obtained from optimum SG(N,L) and ECG_template

Comparison of delay compensated sg310ECG, the signal obtained from optimum SG(N,L) and ECG_template



The ECG_template, the output of SG(3,11), and the output of the optimum SG(N,L) filter were plotted on the same figure for comparison. The optimum filter demonstrated superior preservation of the ECG morphology while further reducing the noise compared to SG(3,11)

Comparison of delay compensated the signal obtained from optimum SG(N,L) and MA(N) and ECG_template



The optimum SG(N, L) filtered signals were compared with the previously obtained optimum MA(N) filtered signals. The observations are as follows:

- **Signal Features:**

- The SG filter, due to its polynomial fitting approach, preserves higher-order signal features such as the sharp slopes of the QRS complex and the curvature of the P and T waves more effectively than the MA filter.
- The MA filter tends to oversmooth the signal, which may distort clinically relevant features, particularly at higher orders.

- **Noise Suppression:**

- Both filters reduce random high-frequency noise.
- The SG filter achieves a better balance by reducing noise while maintaining the diagnostic morphology of the ECG.

- **Computational Complexity:**

- The MA filter is computationally simpler, involving only averaging operations over N samples.
- The SG filter requires polynomial fitting computations for each frame, which increases computational cost, but this is manageable in modern DSP applications.

2 Ensemble Averaging

2.1 Signal with multiple measurements

2.1.1 Preliminaries

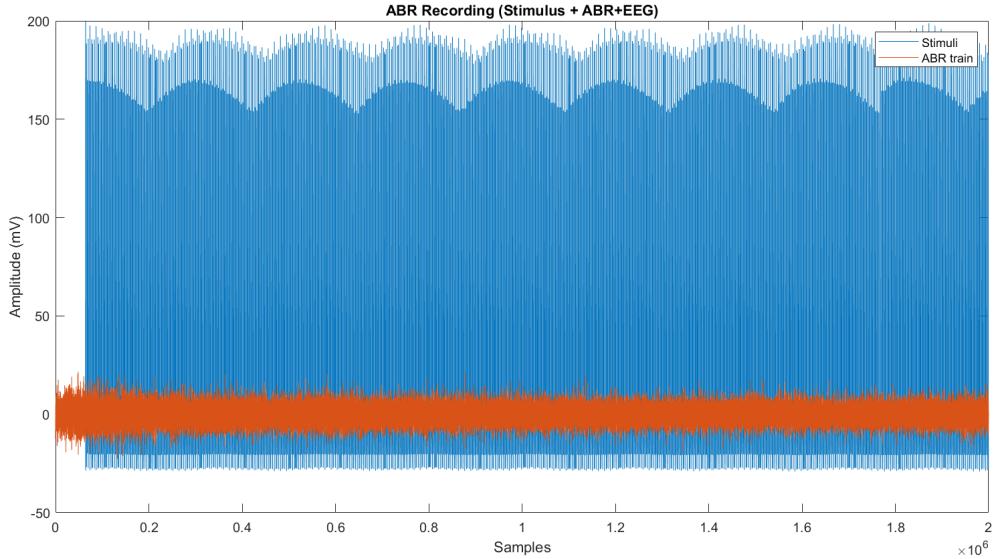
The auditory brainstem response (ABR) is a small-amplitude evoked potential (signal range in μV) that is typically buried under ongoing EEG noise (range in mV). To extract the ABR from noise, multiple recordings of the response are acquired and ensemble averaging is applied.

The recording `ABR_rec.mat` was used, containing over 1000 ABR trials. Stimuli were detected by applying a voltage threshold to the stimulus channel, and corresponding ABR epochs were extracted within a 12 ms window (-2 ms to $+10$ ms around the stimulus onset). The ensemble average of all epochs was then computed as

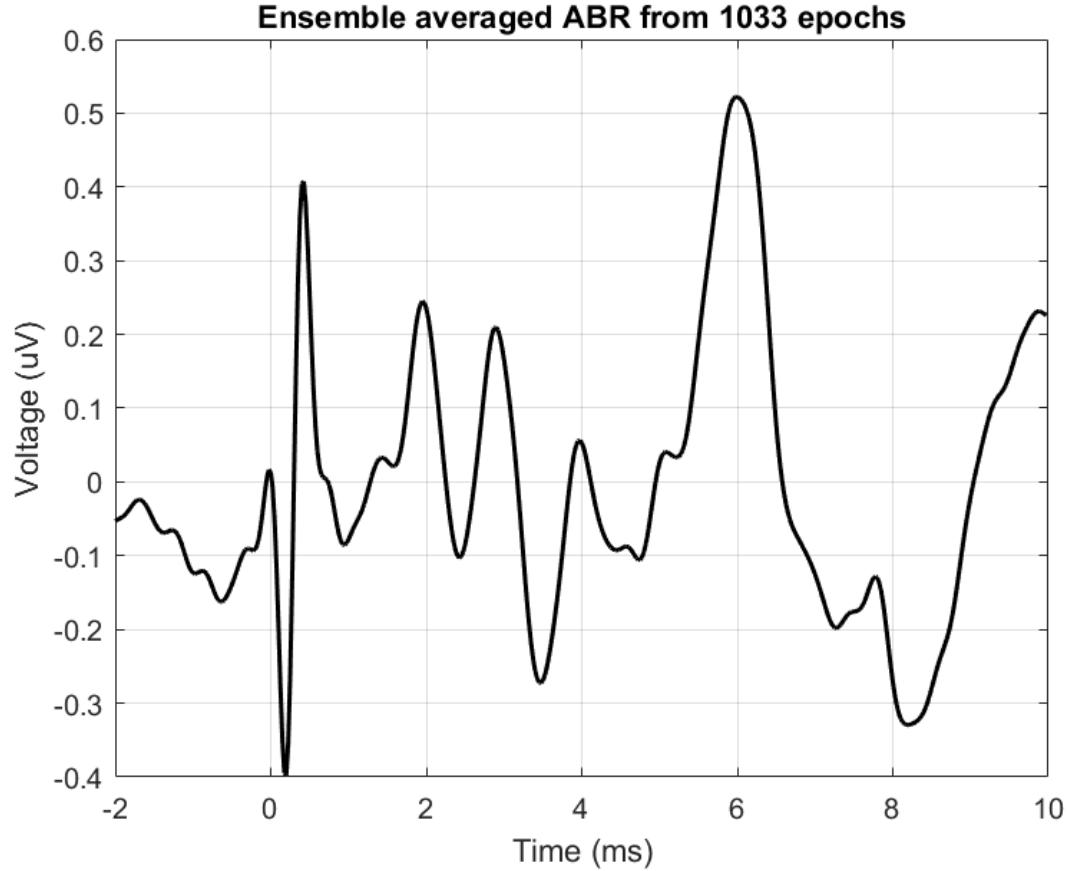
$$\hat{y}(n) = \frac{1}{M} \sum_{k=1}^M x_k(n),$$

where $x_k(n)$ is the k -th ABR epoch and M is the total number of epochs. This averaging suppresses the uncorrelated EEG noise while preserving the stimulus-locked ABR waveform. The ensemble averaged ABR waveform clearly exhibited identifiable peaks corresponding to the characteristic ABR components.

The train of stimuli and ABRs



Ensemble averaged ABR waveform



2.1.2 Improvement of the SNR

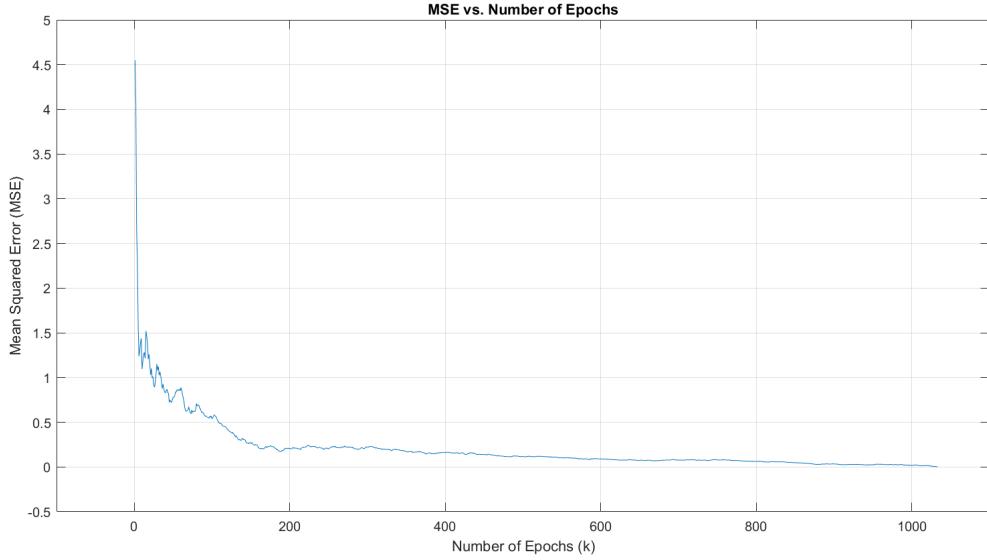
The improvement in SNR through ensemble averaging was quantified using the progressive mean squared error (MSE), calculated as:

$$MSE_k = \sqrt{\frac{1}{N} \sum_{n=1}^N (\hat{y}(n) - \tilde{y}_k(n))^2}, \quad k = 1, 2, \dots, M$$

where:

- $\hat{y}(n)$ is the template (ensemble average using all epochs),
- N is the number of samples per epoch,
- M is the total number of epochs,
- $\tilde{y}_k(n)$ is the progressive average over the first k epochs:

$$\tilde{y}_k(n) = \frac{1}{k} \sum_{i=1}^k x_i(n).$$



A plot of MSE_k against k demonstrates that the error decreases rapidly as the number of epochs increases, and gradually approaches a minimum value for large k . This behaviour agrees with the theoretical expectation that ensemble averaging improves the SNR approximately by a factor of \sqrt{M} .

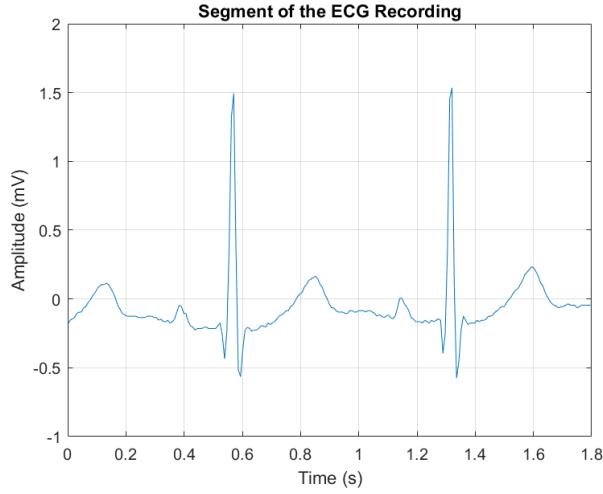
At small k (few epochs), the MSE is large because the noise dominates and the progressive average poorly represents the true template. As k increases, random EEG noise tends to cancel out, while the time-locked ABR remains, reducing the MSE. At very large k , the error converges slowly, since the SNR improvement follows the law of diminishing returns ($\propto 1/\sqrt{M}$). Thus, ensemble averaging provides a simple yet powerful approach for extracting low-amplitude evoked responses from high-amplitude EEG noise.

2.2 Signal with repetitive patterns

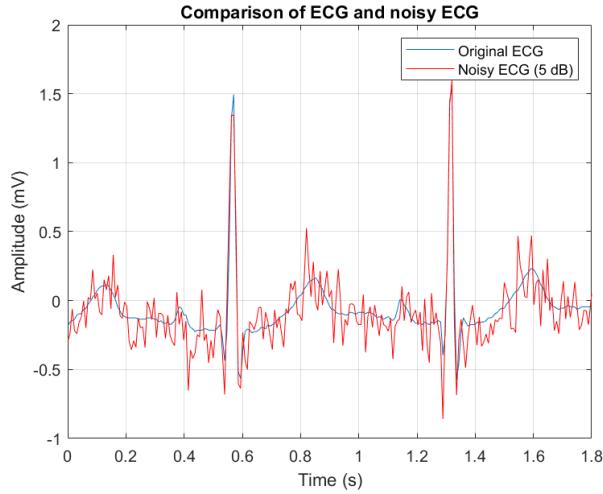
In this section, an almost periodic recording of an ECG pulse train (ECG_rec.mat) was analysed to demonstrate ensemble averaging for noise reduction. The acquisition parameters were: recorded from Lead II, sampling frequency $f_s = 128$ Hz, and amplitude range in mV.

Viewing the signal and adding Gaussian noise

The recorded ECG pulse train was first visualised in the time domain. A single PQRST waveform was extracted and defined as the *ECG_template*.

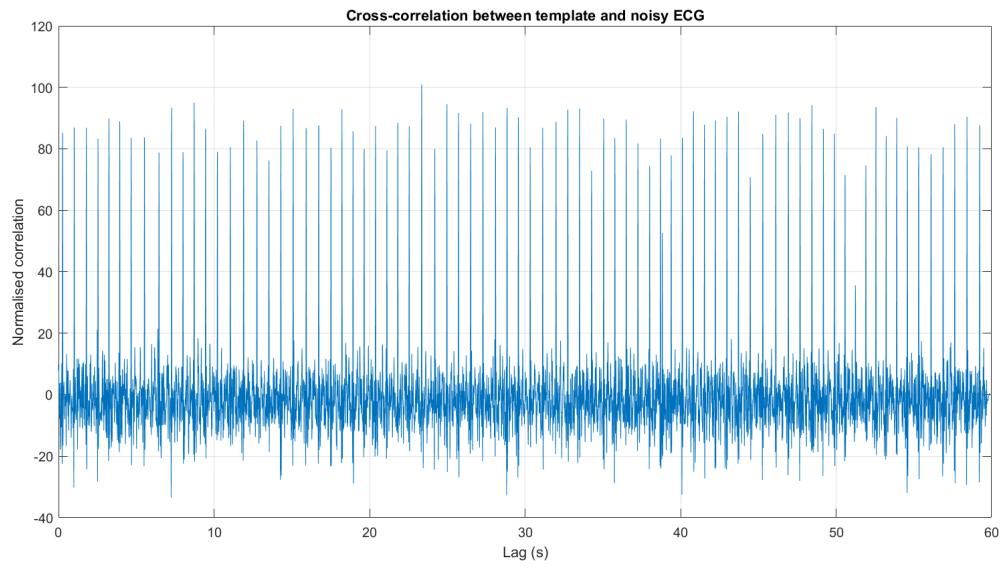


To simulate realistic conditions, Gaussian white noise with a signal-to-noise ratio (SNR) of 5 dB was added to the ECG signal using MATLAB's `awgn` function. The resulting noisy signal was denoted as $nECG$.

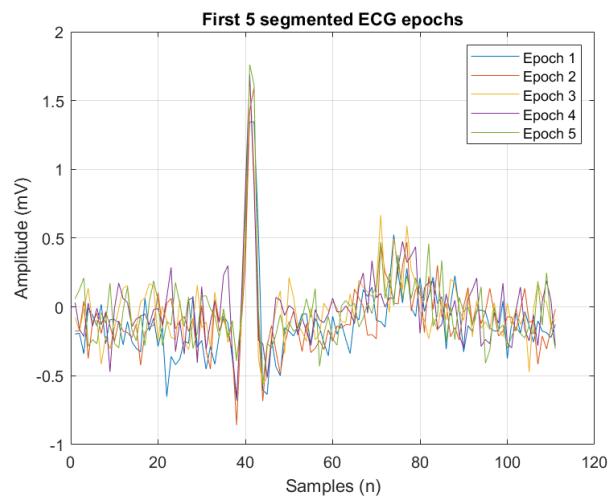


2.3 Segmenting ECG into separate epochs and ensemble averaging

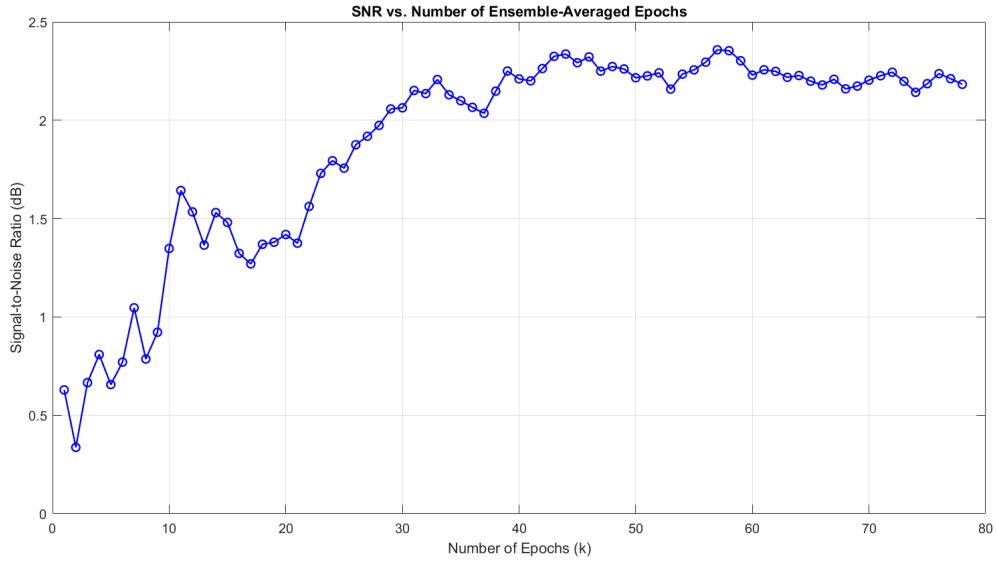
Normalised cross-correlation values against the adjusted lag axis converted to time axis



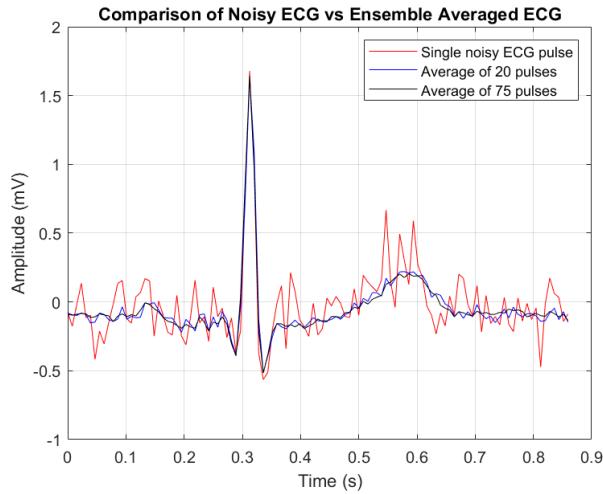
Segment ECG pulses



The improvement in SNR as the number of ECG is increased



Noisy ECG pulse and two arbitrary selected ensemble averaged ECG pulses



A single noisy ECG beat was compared with ensemble averaged pulses obtained from 20 and 80 epochs. The ensemble averaged waveforms exhibited clearly defined PQRST complexes with reduced baseline noise, whereas the noisy ECG showed significant distortion.

Discussion It is evident that segmenting ECG beats using points of maximum correlation yields more accurate epochs than merely detecting R-peaks. While R-peak detection aligns only the QRS complex, cross-correlation ensures alignment of the entire PQRST morphology, thus minimising temporal distortions in ensemble averaging. This justifies the claim that maximum correlation is a better method for segmenting ECG pulse trains under noisy conditions.

3 Designing FIR filters using windows

3.1 Characteristics of window functions (use the fdatool)

FIR filters can be designed using the window method, where the ideal impulse response is truncated and multiplied by a window function. The choice of window function and its length M directly affect the frequency response of the filter, including the transition width, stopband attenuation, and side-lobe behaviour.

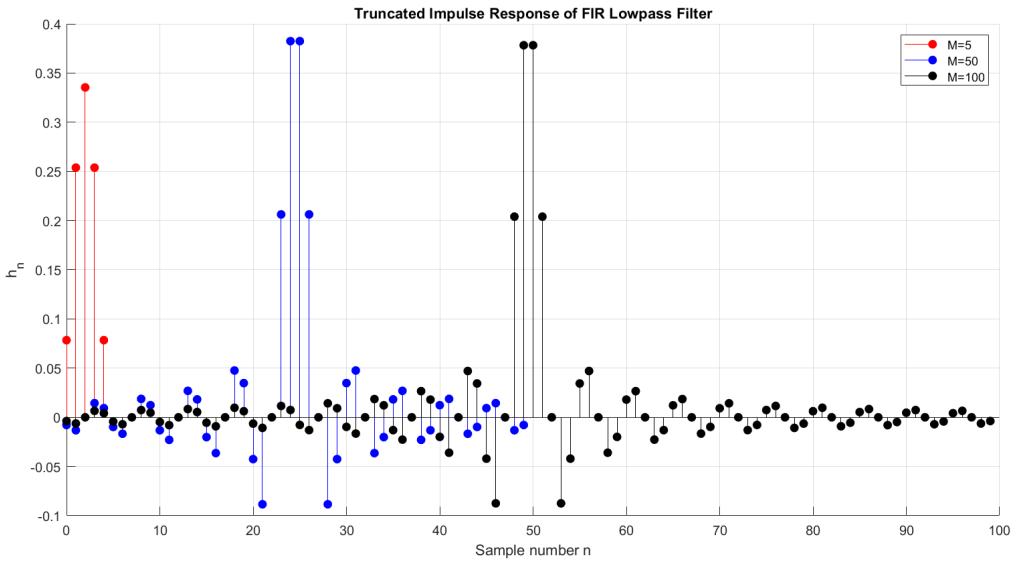
(i) Effect of M for the Rectangular Window

The rectangular window is defined as:

$$w(n) = \begin{cases} 1, & 0 \leq n \leq M - 1 \\ 0, & \text{otherwise.} \end{cases}$$

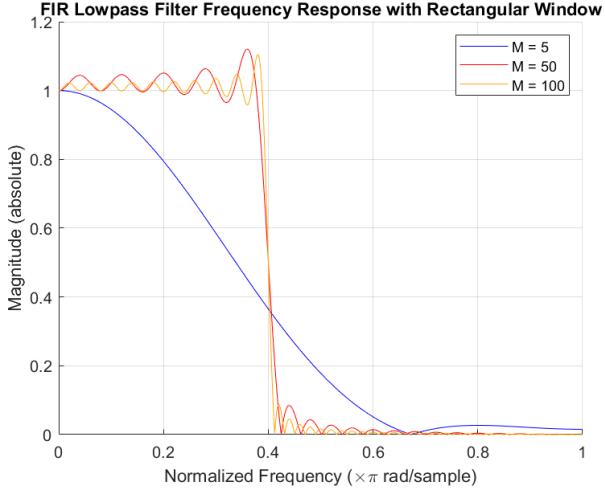
Figure shows the impulse responses of rectangular windows with $M = 5$, $M = 50$, and $M = 100$. It can be observed that:

- As the window length becomes larger, the filter's impulse response gets closer to a perfect sinc function. A perfect sinc function in the time domain corresponds to an ideal rectangular frequency response (an ideal filter).
- For small M (e.g., $M = 5$), the impulse response is short, leading to a very wide transition band.
- As M increases ($M = 50$ and $M = 100$), the window length increases, resulting in a narrower main lobe and improved frequency selectivity.



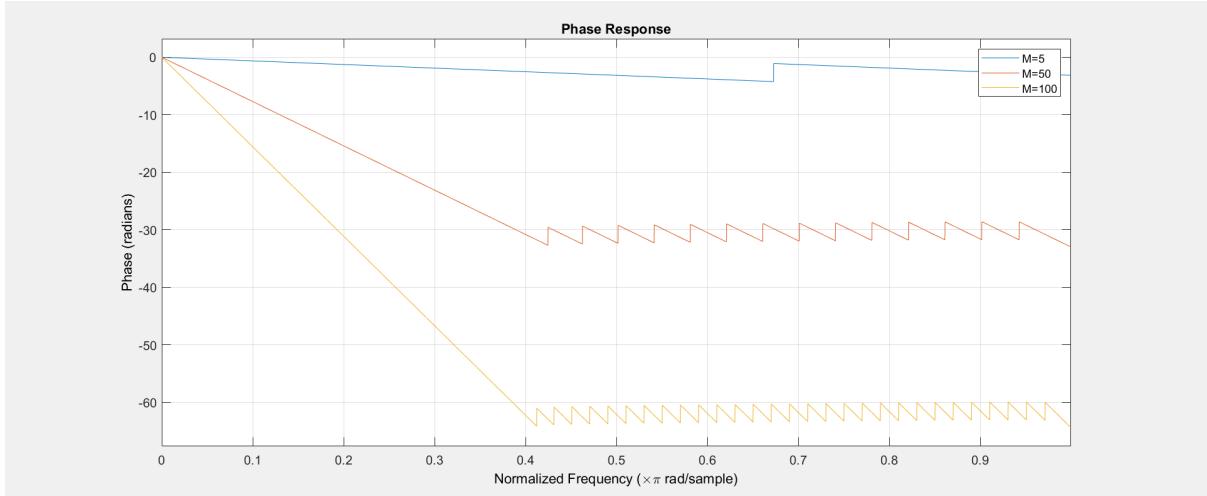
(ii) Rectangular Window Low-pass Filter

A low-pass filter with a cutoff frequency $\omega_c = 0.4\pi$ was designed using rectangular windows of length $M = 5, 50, 100$. The magnitude and phase responses are shown in Figures.



- For $M = 5$, the transition band is very wide, and the stopband attenuation is poor.
- For $M = 50$, the transition width decreases, producing a better approximation of the ideal low-pass filter but large ripples are visible in both pass and stop band.
- For $M = 100$, the transition width is further reduced, but large ripples are visible due to high side-lobe levels.

Phase Response: Figure shows the phase responses of the rectangular-windowed low-pass FIR filters for $M = 5, 50, 100$.



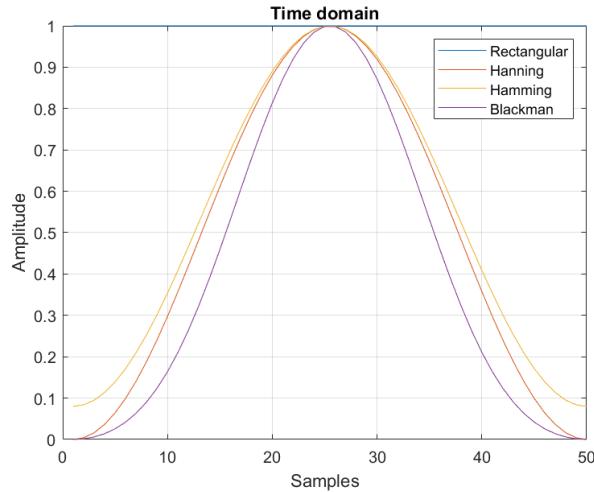
- For small M (e.g., $M = 5$), the filter length is short, resulting in a nearly flat but less accurate phase response.
- As M increases ($M = 50$ and $M = 100$), the phase shift increases linearly with frequency, indicating a longer group delay (equal to $\frac{M-1}{2}$ samples).
- The periodic jumps (phase wrapping) observed are due to MATLAB's representation of phase in the $(-\pi, \pi]$ range.

(iii) Comparison of Different Window Functions ($M = 50$)

To reduce the discontinuities at the edges of the rectangular window, smoother windows such as Hanning, Hamming, and Blackman are used. Their characteristics for $M = 50$ are summarized below.

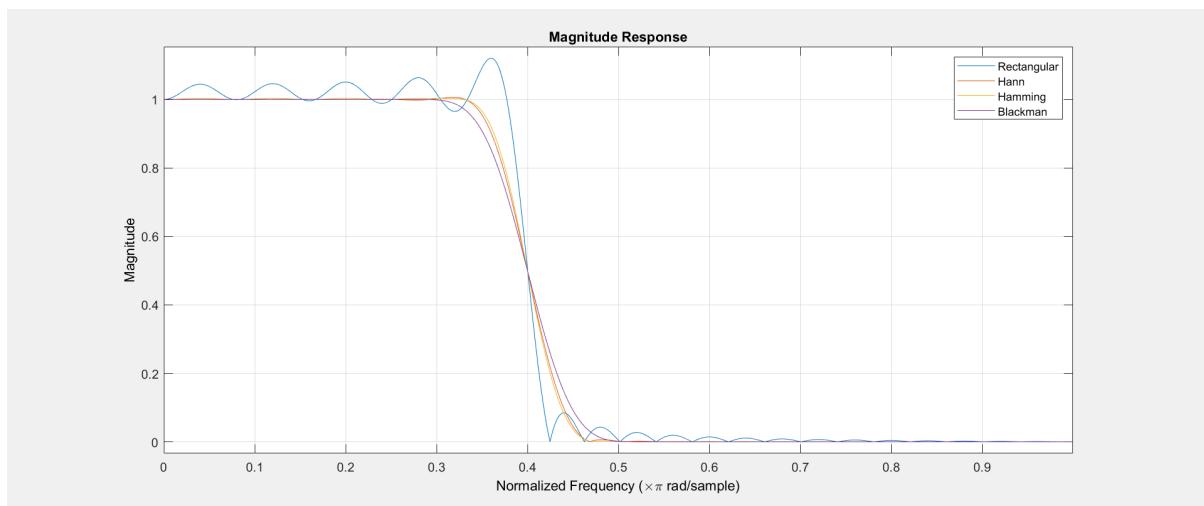
Morphology of the windows:

- The rectangular window is constant with abrupt edges.
- The Hanning and Hamming windows taper smoothly to zero at both ends.
- The Blackman window provides the smoothest tapering.



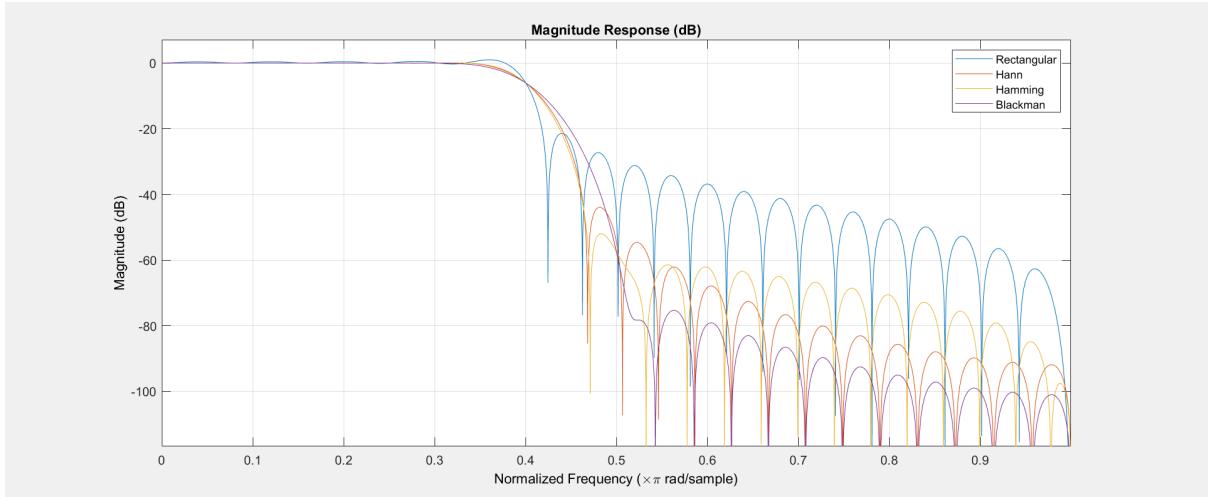
Magnitude response (linear scale):

- The rectangular window has the narrowest main lobe but very high side-lobes.
- Hanning and Hamming windows widen the main lobe slightly but reduce side-lobe levels.
- The Blackman window offers the best side-lobe attenuation at the cost of a much wider main lobe.

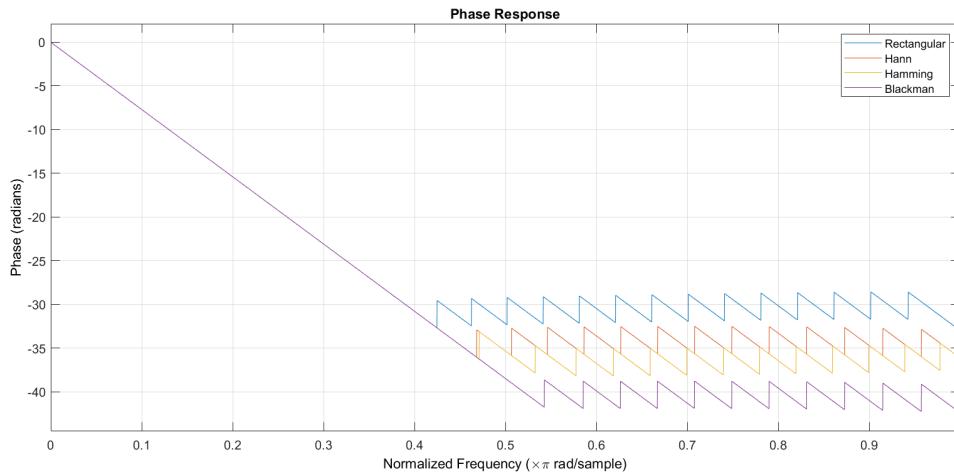


Magnitude response (logarithmic scale):

- Blackman window: best performance.



Phase response: All four window functions yield approximately linear phase responses, consistent with their symmetric FIR impulse responses.



3.2 FIR Filter Design and Application using the Kaiser Window

The Kaiser window is a flexible window function that can approximate a wide variety of windows by tuning the shaping parameter β and the filter length M . In this section, a low-pass and a high-pass filter using the Kaiser window, along with a FIR comb filter, are designed and applied to denoise the ECG signal.

(i) Time-domain and frequency-domain representation

The noisy ECG signal is first visualized in the time domain, and its frequency content is analyzed using the periodogram method. Following figure shows time-domain waveform .

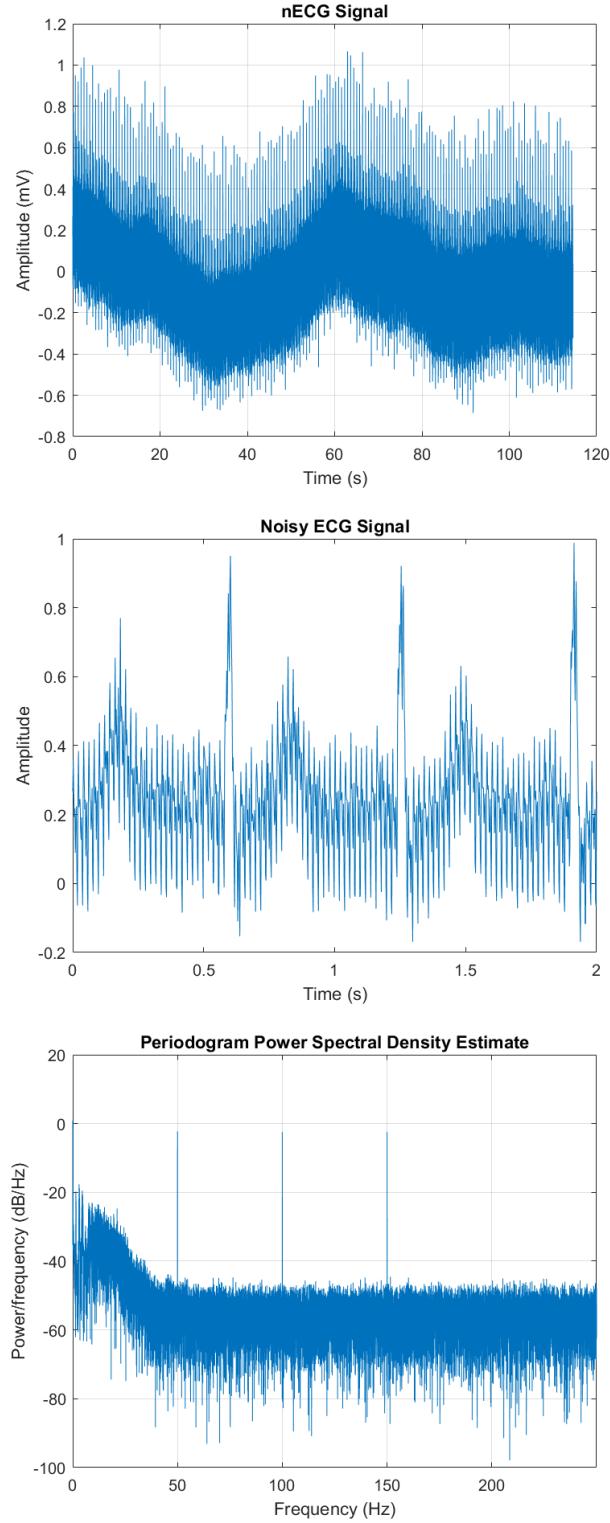


Figure shows the PSD estimate of the noisy ECG. It can be seen that the ECG spectral components mainly lie in the bandwidth of 0.05–60 Hz, while distinct peaks at 50 Hz and its harmonics correspond to powerline interference. Additionally, higher-frequency components are due to additive white Gaussian noise.

(ii) Filter parameter choices

I choose the following pass/stop edges and ripple (in Hz):

	Highpass	Lowpass	Comb
f_{pass}	0.2	60	—
f_{stop}	0.7	100	—
δ	10^{-3}	10^{-3}	—
$f_{\text{stop},1}$	—	—	50
$f_{\text{stop},2}$	—	—	100
$f_{\text{stop},3}$	—	—	150

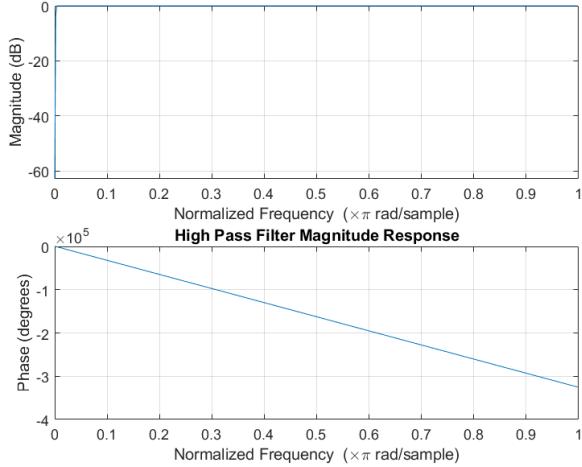
Here, the HPF transition band is $0.2 \rightarrow 0.7$ Hz (to suppress baseline wander), the LPF transition band is $60 \rightarrow 100$ Hz (to suppress high-frequency noise), and comb filter at 50Hz, 100Hz and 150Hz to remove powerline noise.

(iii) Kaiser window parameters

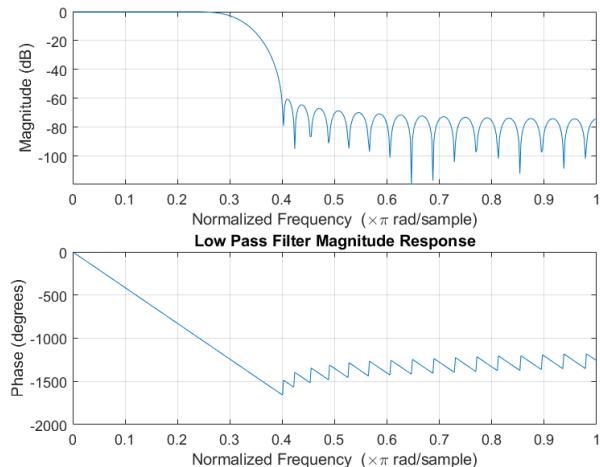
- The optimum filter parameters for HPF are $M = 3626$ and $\beta = 5.653260$.
- The optimum filter parameters for LPF are $M = 46$ and $\beta = 5.653260$.

(iv) Window and filter responses

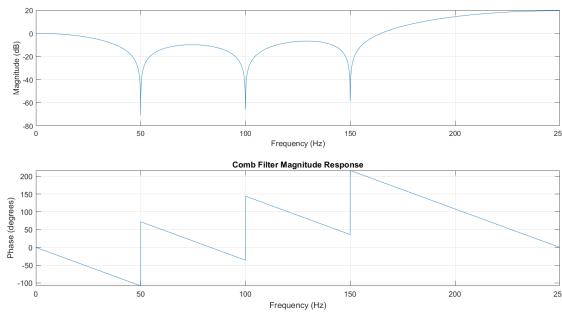
High Pass Filter



Low Pass Filter



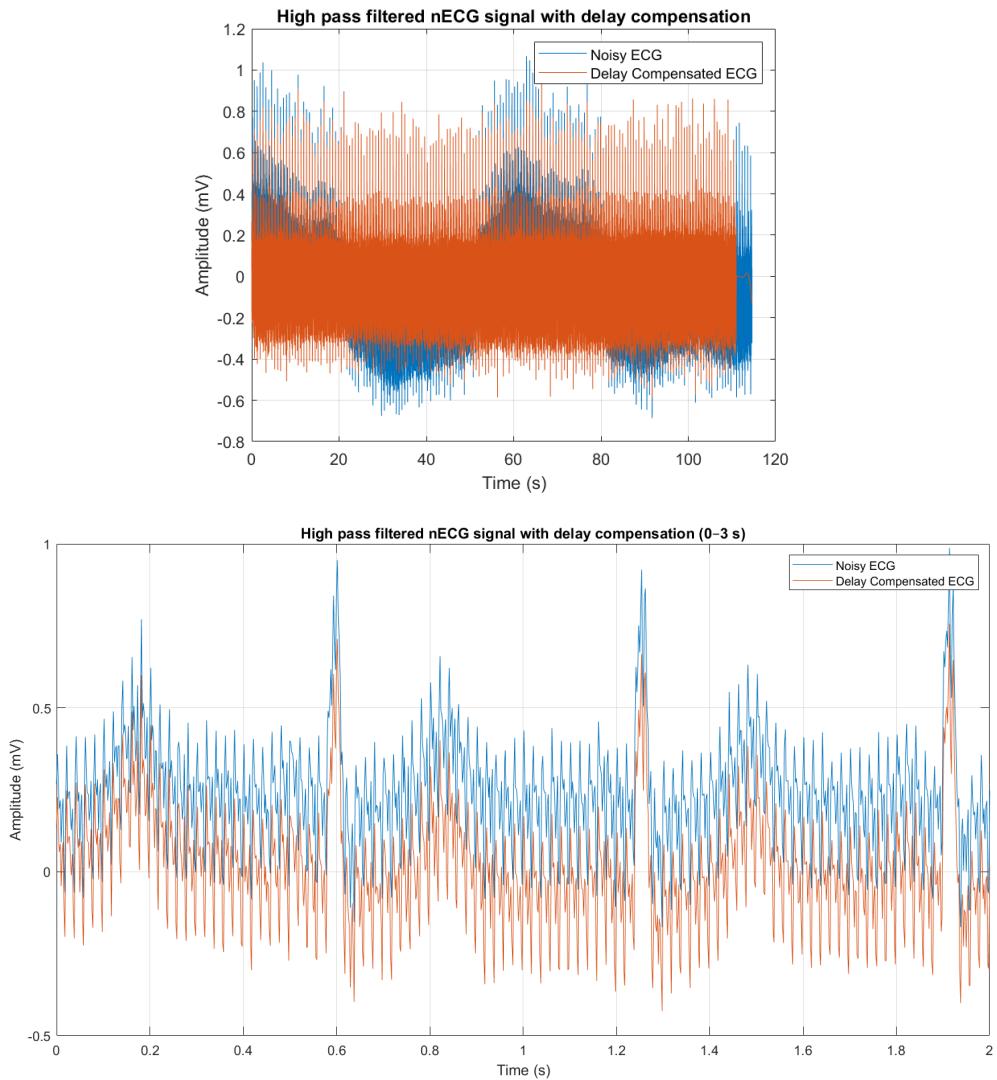
Comb Filter



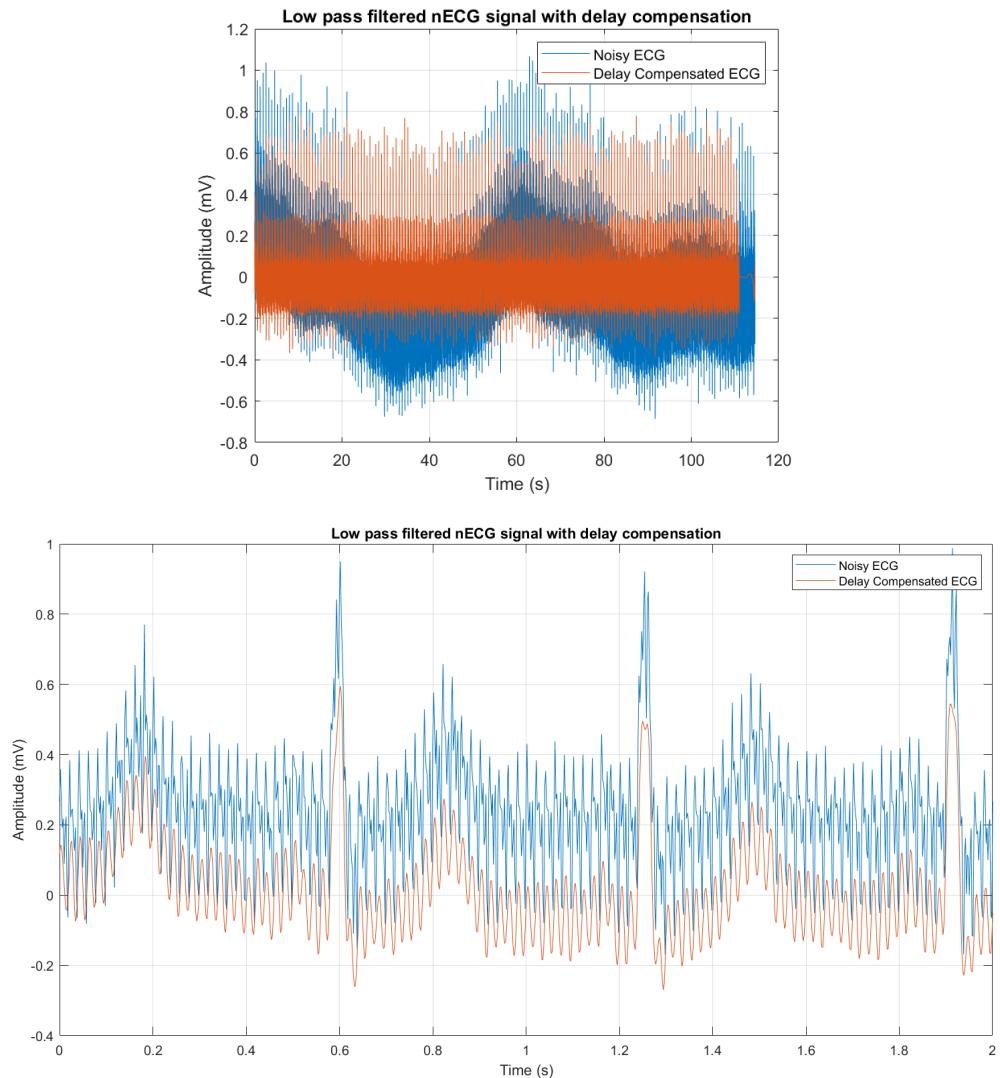
(v) Filtering the ECG signal

The filters (HPF, LPF, and comb) are applied sequentially to the noisy ECG. Group delay, equal to $(M - 1)/2$, is compensated in each case.

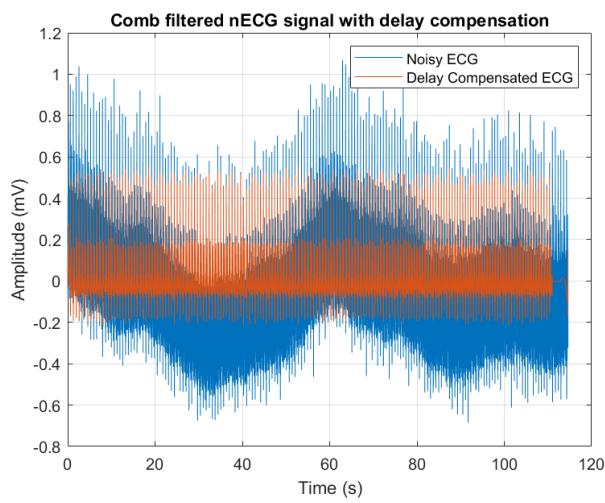
High Pass Filter applied signal

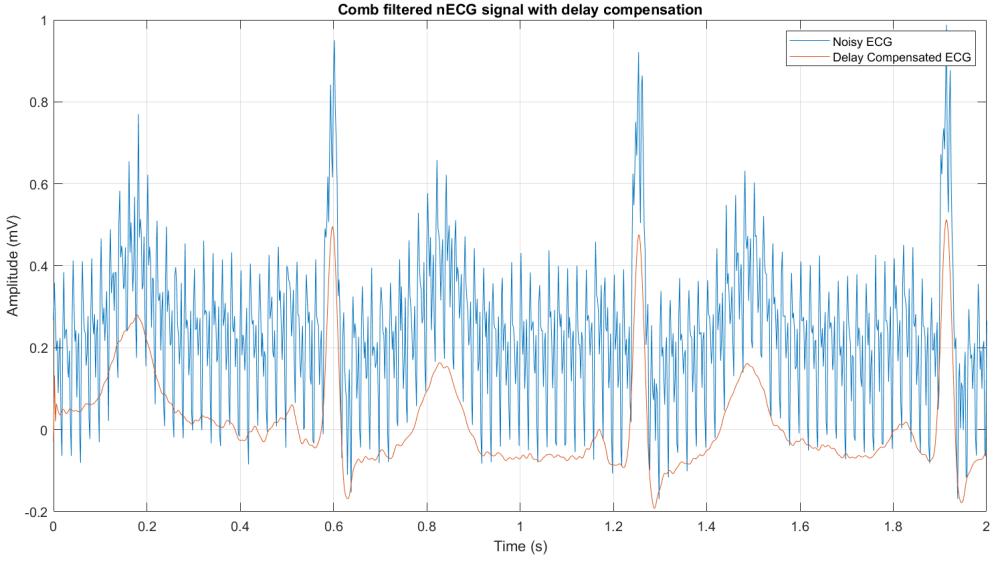


Low Pass Filter applied signal



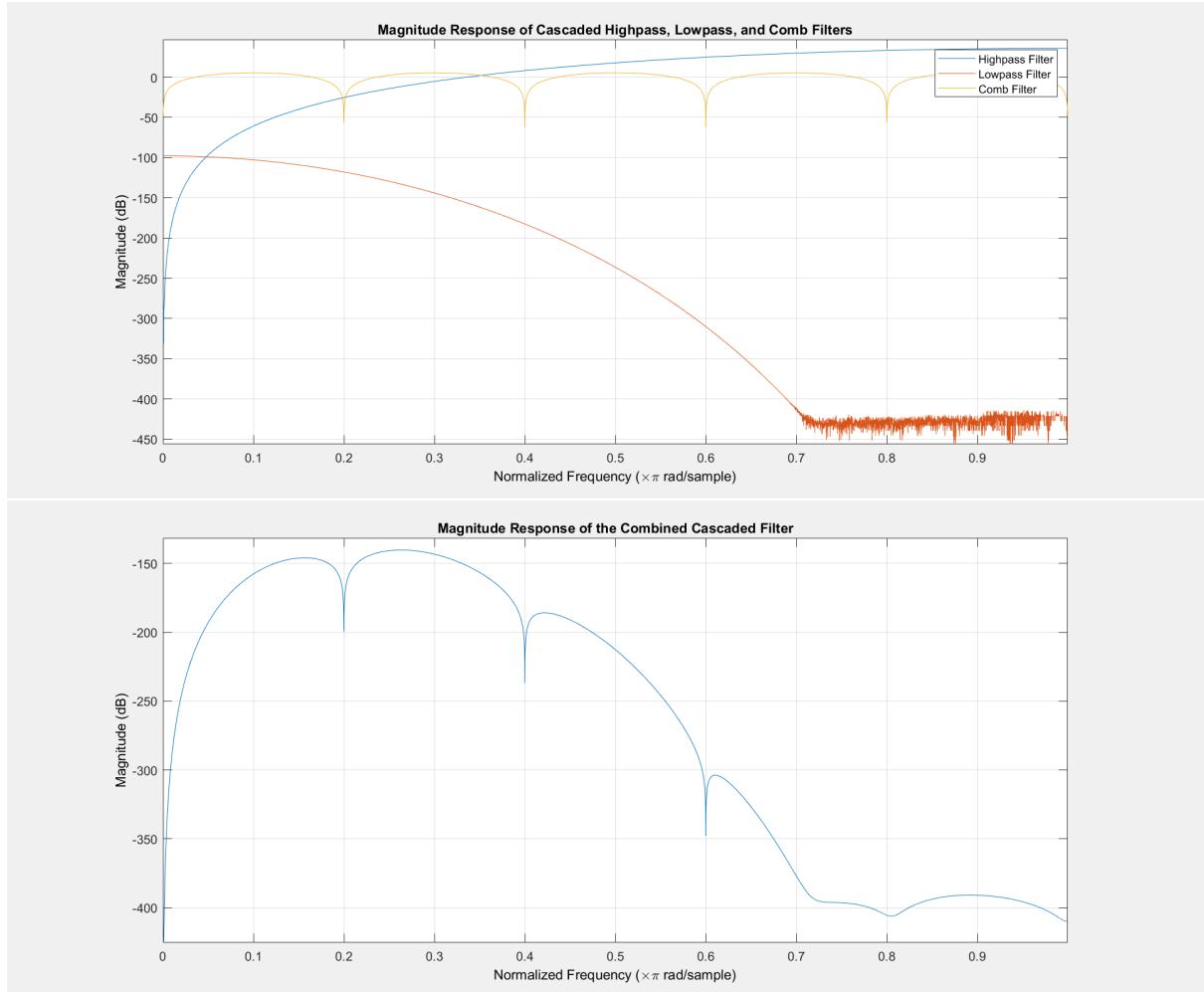
Comb Filter applied signal





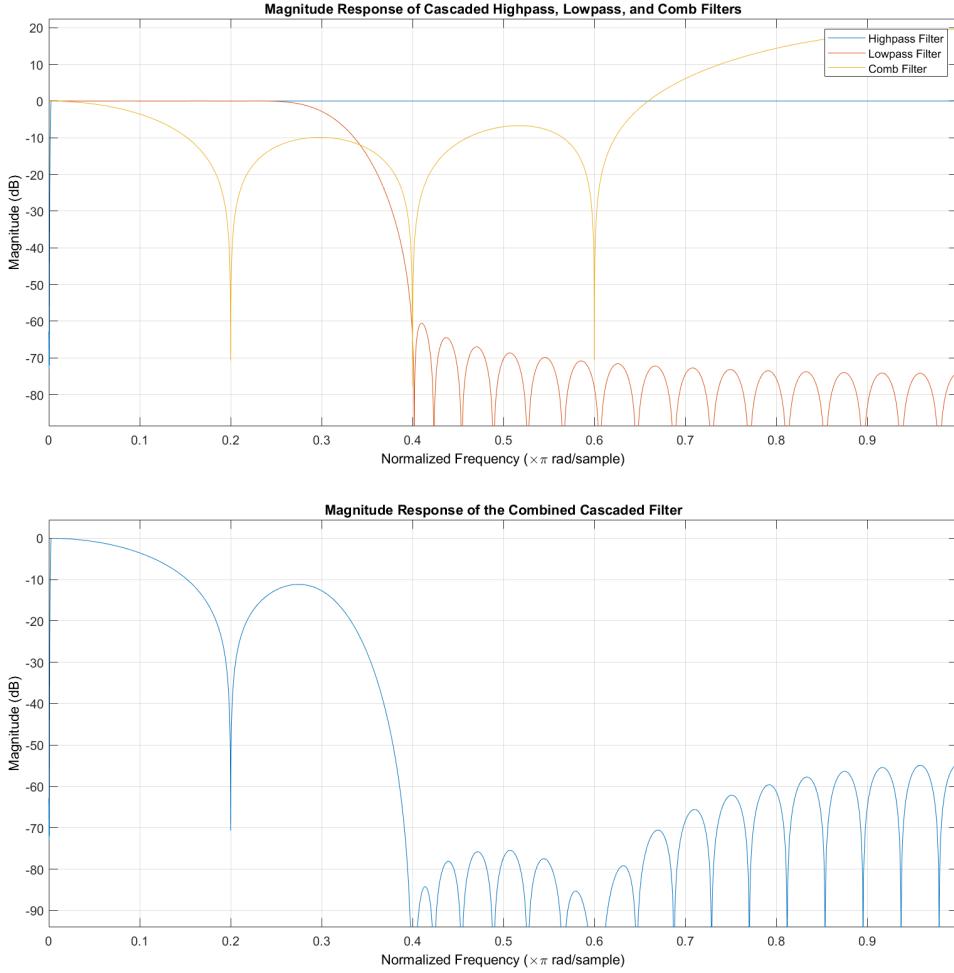
(vi) Overall filtering process

The magnitude response of the combined filters is plotted, and the PSD of the final filtered ECG is compared with the noisy ECG.



As shown in the figure below the noise components outside the ECG bandwidth and the pow-

erline interference peaks are effectively removed.



4 IIR filters

This section realises Butterworth IIR filters equivalent to the FIR filters designed earlier, and studies the impact of non-linear phase, forward vs. forward-backward filtering, and spectral effects.

4.1 Realising IIR filters

(i) Butterworth low-pass coefficients. To enable a fair comparison with the FIR low-pass (LPF) designed in Sec. 3, we use the same cutoff frequency f_c and choose an IIR order n to match the FIR design target.

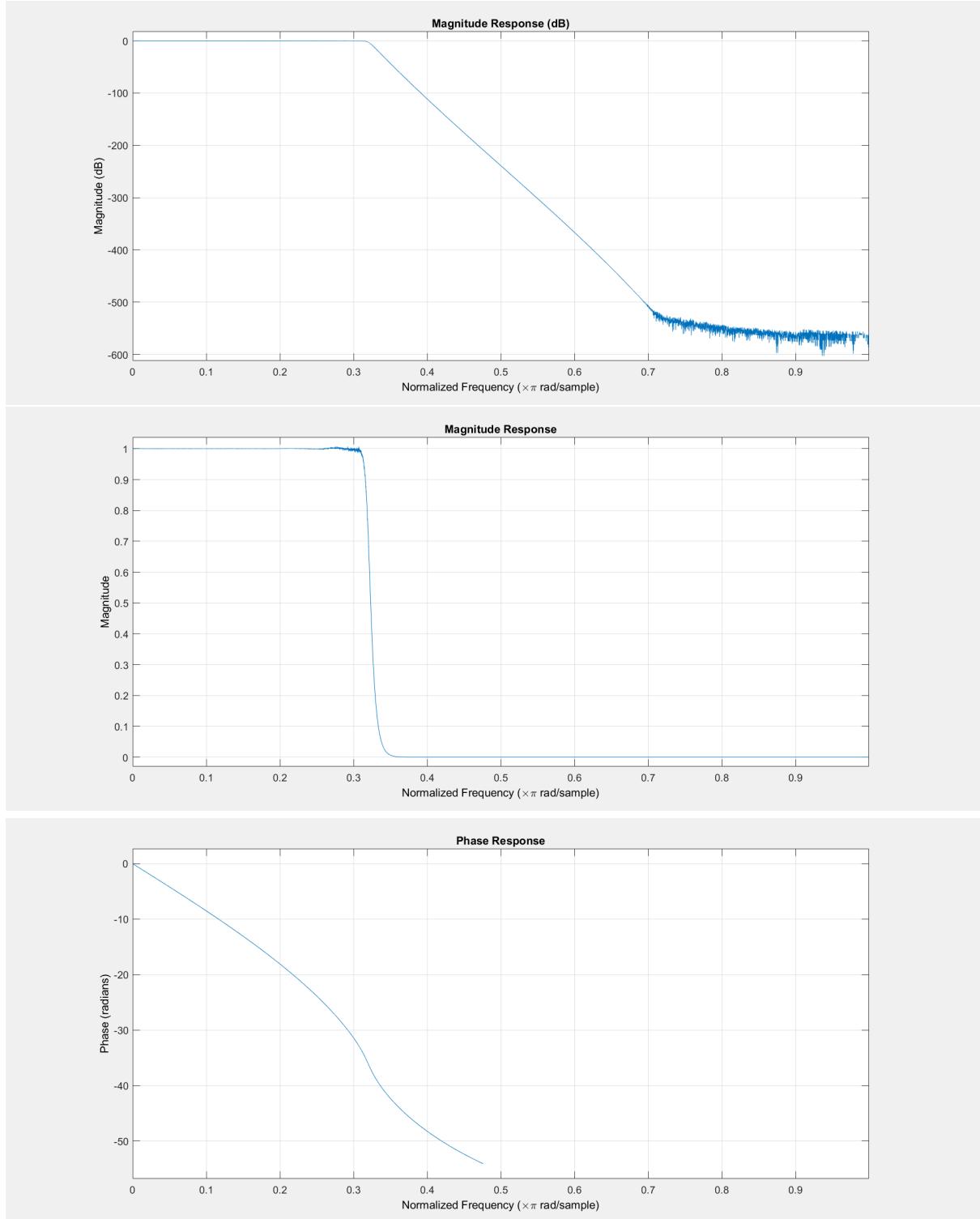
(ii) Magnitude, phase, and group delay of Low pass The IIR magnitude and phase responses and the group delay,

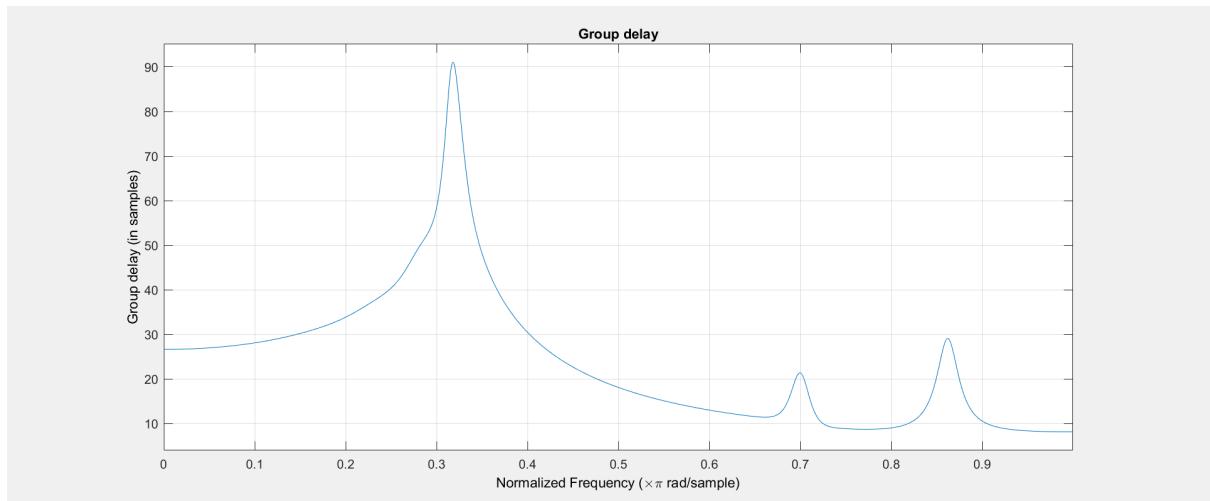
$$\tau_g(\omega) = -\frac{d}{d\omega} \arg\{H(e^{j\omega})\},$$

are visualised with `fvtool`. As expected for Butterworth filters:

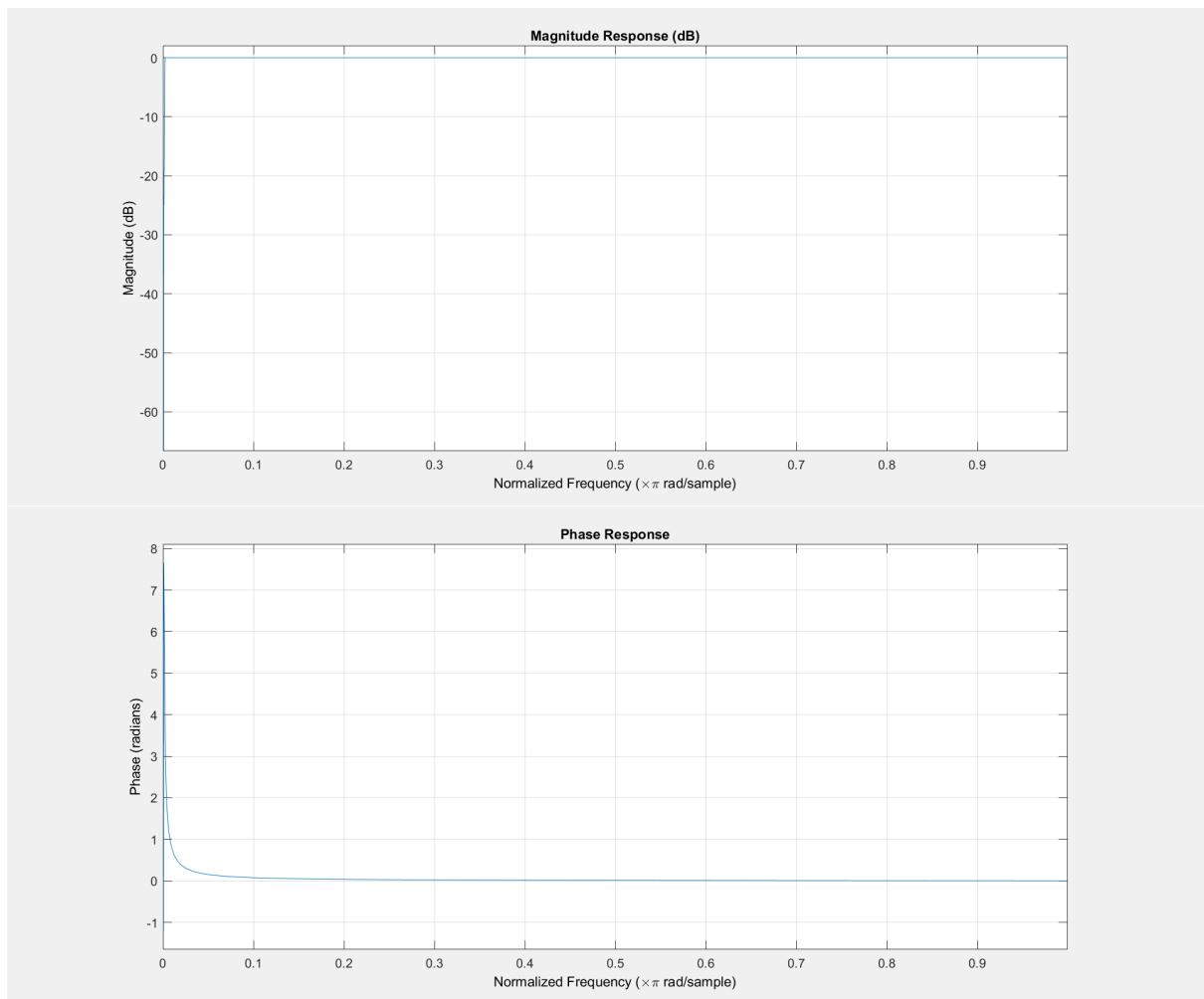
- the magnitude response is ripple-free in both passband and stopband,
- the phase is *non-linear* (frequency dependent), hence

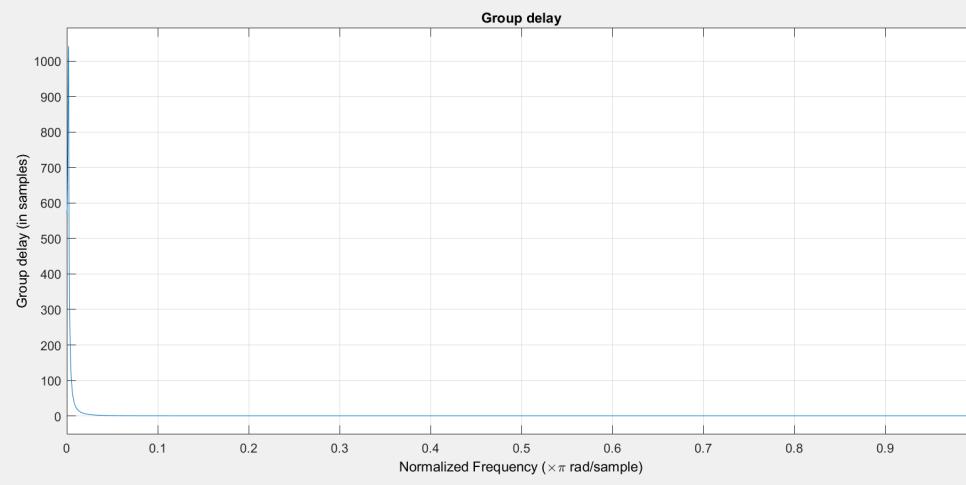
- the group delay is not constant (distortion possible in time-domain waveforms).





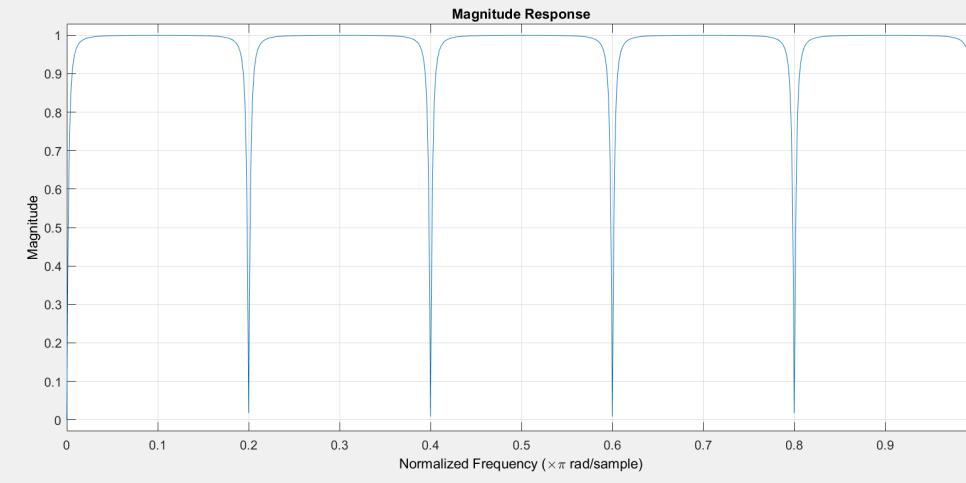
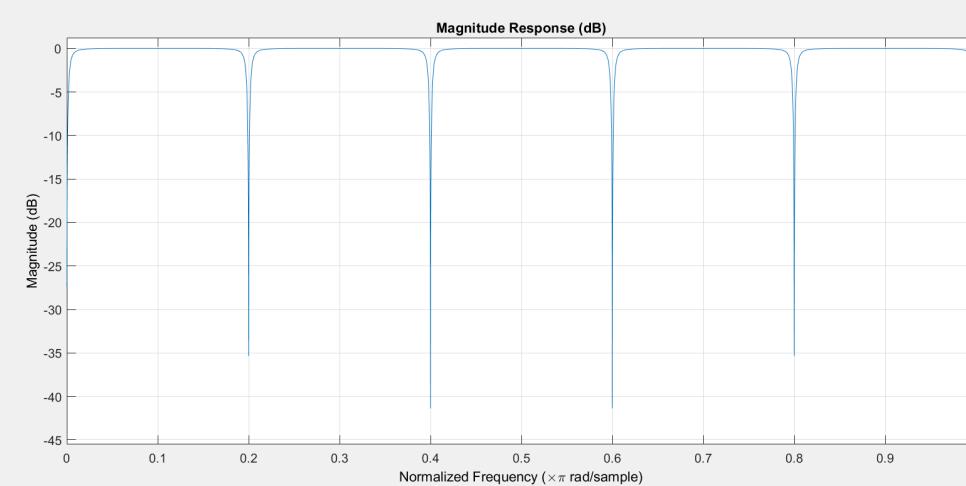
(iii) High-pass and comb filters. A Butterworth high-pass (HPF) with cutoff f_c^{HP} is designed via `butter(n,Wn, 'high')`.

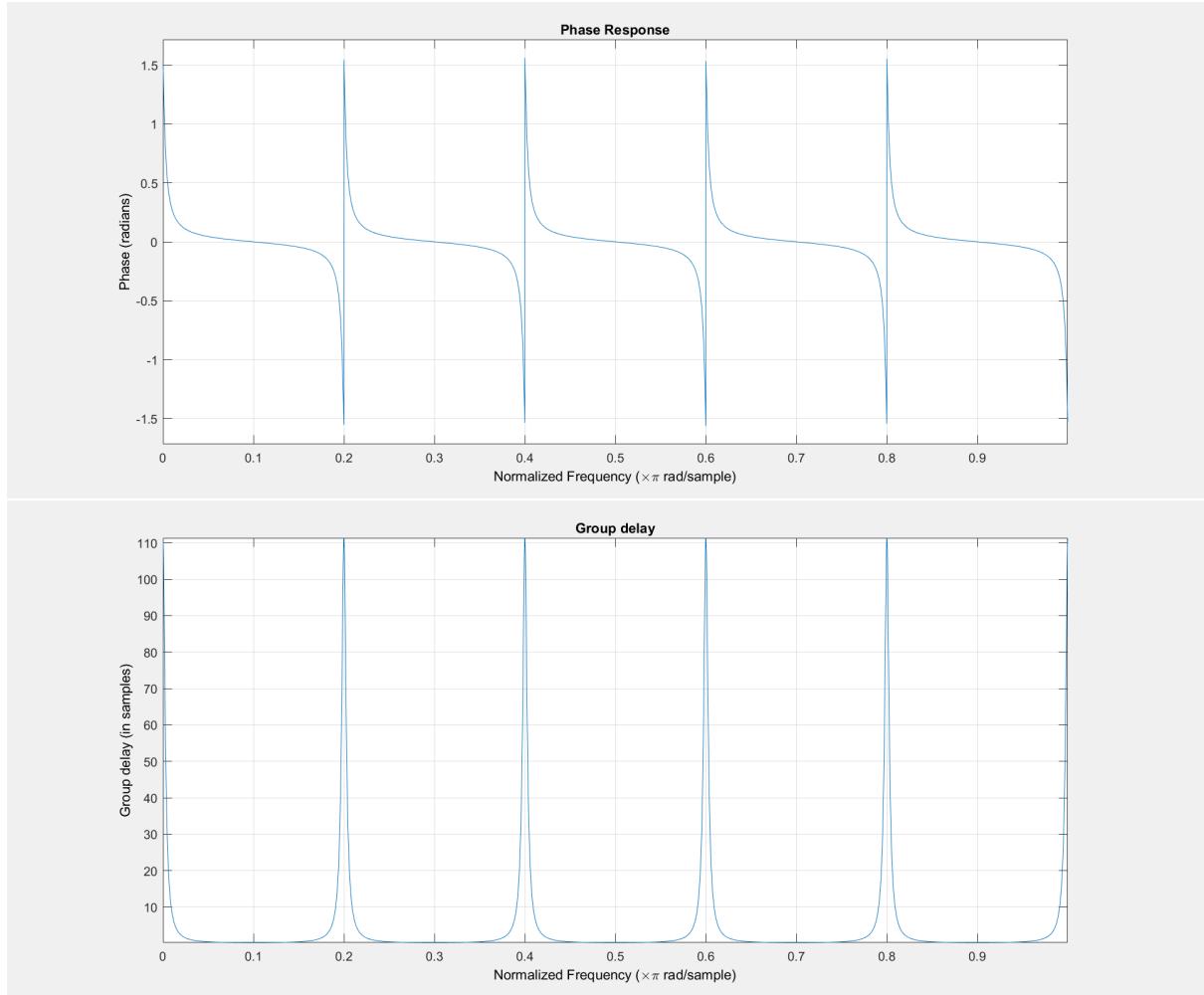




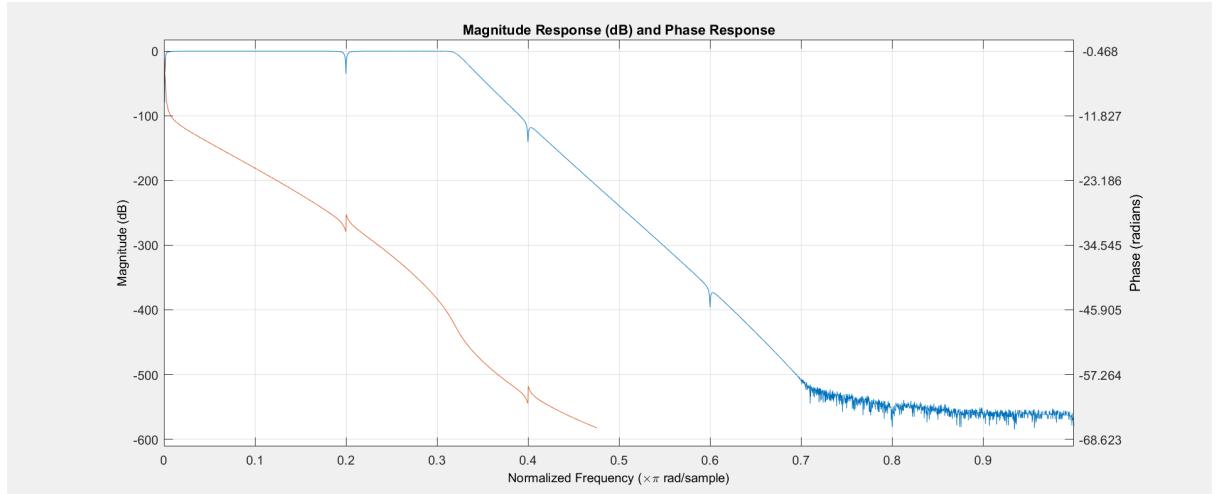
Periodic narrowband interference (e.g. 50 Hz mains and its harmonics) is suppressed by an IIR comb filter:

$$H_{\text{comb}}(z) = \frac{1}{1 - \alpha z^{-N}} \quad \text{or} \quad [b, a] = \text{iircmb}(N, \text{BW}),$$



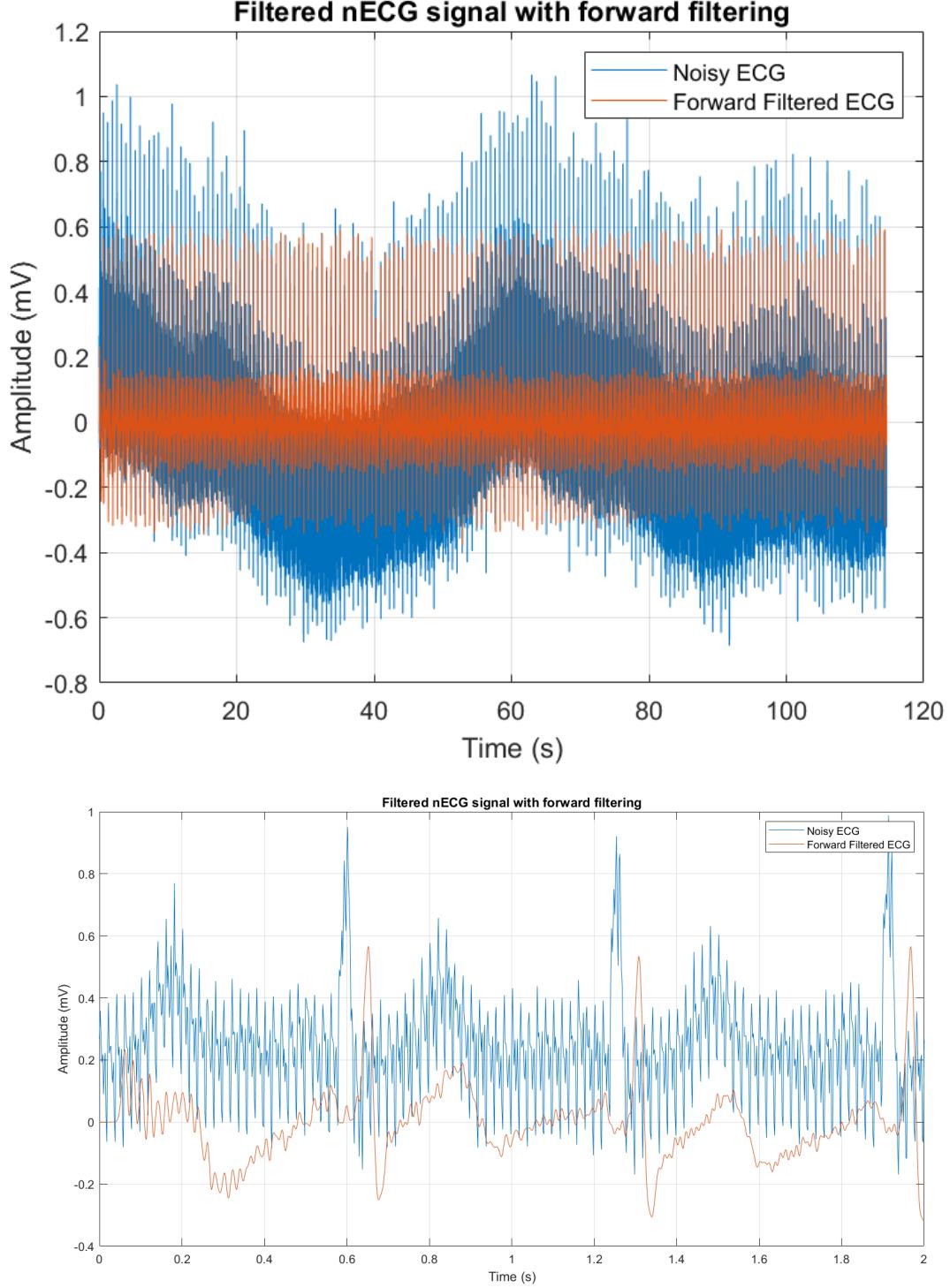


(iv) Combined response and comparison to FIR. The cascaded IIR response $H_{\text{tot}}(z) = H_{\text{HP}}(z) H_{\text{LP}}(z) H_{\text{comb}}(z)$ shows (i) a passband matching the ECG bandwidth, (ii) steep low- and high-frequency attenuation, and (iii) notches at the comb frequencies. Compared with the FIR cascade: the overall *magnitude* shaping can be made similar, but the IIR exhibits *non-linear phase* and frequency-dependent group delay, whereas the FIR (symmetric) is linear-phase.



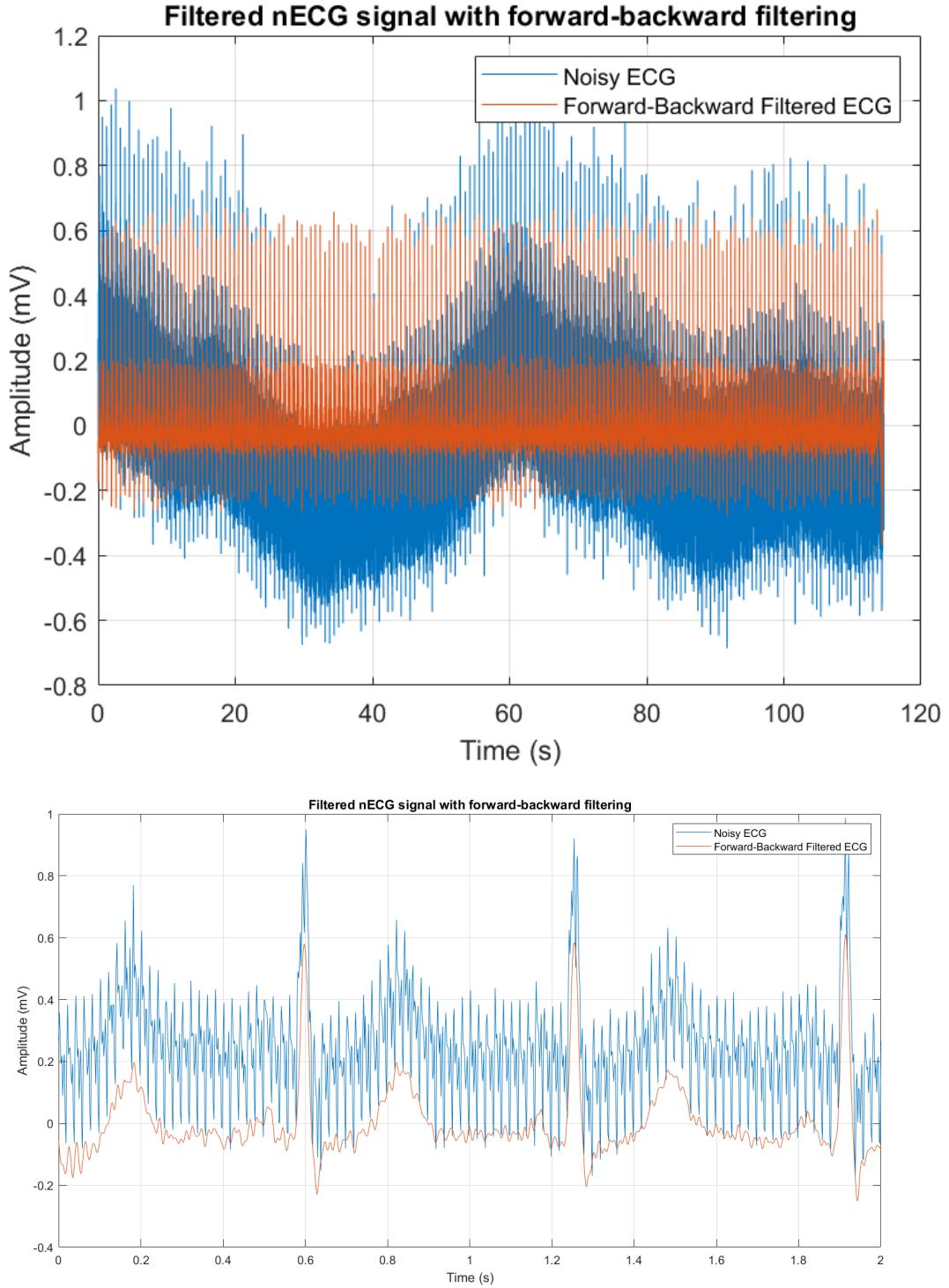
4.2 Filtering methods using IIR filters

(i) **Forward filtering** Applying `filter(b,a,x)` with the combined IIR passes the ECG through a causal realisation. The magnitude shaping matches $|H_{\text{tot}}(e^{j\omega})|$, but the non-linear phase introduces frequency-dependent delays (QRS widening and slight morphology distortion are possible).



(ii) **Forward–backward filtering.** Applying `filtfilt(b,a,x)` runs the filter forward and backward, yielding an *effective zero-phase* response with overall magnitude $|H_{\text{tot}}|^2$ and doubled order, while cancelling phase distortion. End-effects are mitigated by internal padding, but

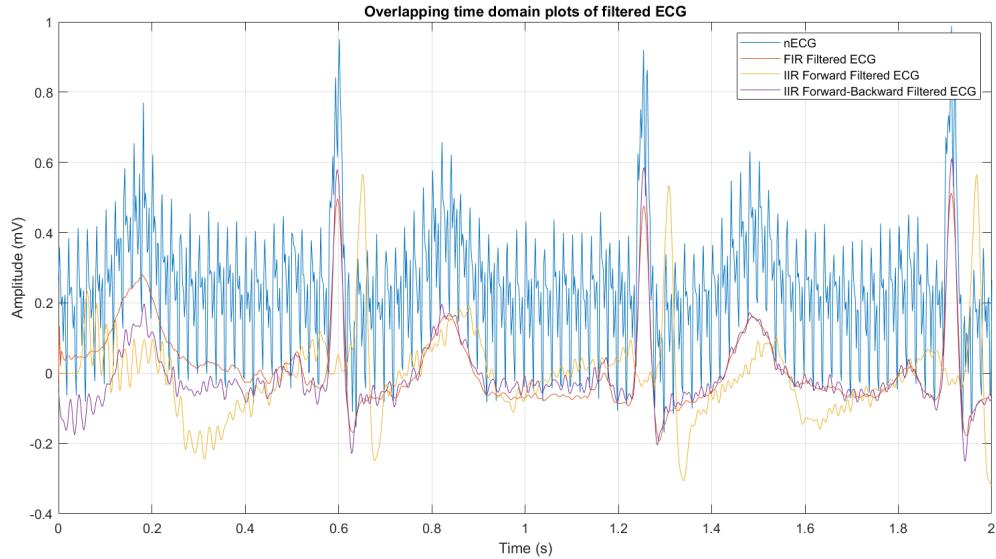
short records may still exhibit edge transients.



(iii) Time-domain comparison (zoom to two beats). Figure overlays: FIR (linear-phase) filtered ECG, IIR forward-filtered ECG, and IIR forward-backward ECG. Observations:

- **FIR (Kaiser/rectangular etc.)**: preserves beat morphology (linear phase), but requires higher order for a given transition width.
- **IIR forward**: good noise suppression; introduces lag and slight waveform asymmetry due to non-linear phase (group delay varies across P, QRS, T content).

- **IIR `filtfilt`:** morphology best preserved among IIR methods; no phase lag; sharper notch/roll-off perception since magnitude is effectively squared; minor endpoint artefacts possible.



(iv) PSD comparison. Figure shows overlapping periodogram PSDs of the three outputs. Key points:

- Both FIR and IIR filters successfully suppress the noise present in the ECG signal.
- IIR filters exhibit much sharper transition bands compared to FIR filters.
- When implementing comb filters, IIR filters provide significantly better noise suppression, which is clearly visible in the PSD comparison.
- FIR filters require very high orders to achieve narrow transition bands, whereas IIR filters can achieve the same with much lower filter orders.
- This demonstrates the efficiency of IIR filters in terms of order, although they suffer from non-linear phase characteristics unlike FIR filters.

