Independent University Bangladesh

Department of Electrical and Electronics Engineering

Project on Digital Filter Design

Name: Hemal Sharma

Id: 2221855

Cours Code: EEE 321

Couse: Digital Signal Processing Lab

Semester: Spring 2024

Date: 25/04/2024

Objective:

Design and apply digital filters to noisy electrocardiogram (ECG) signal, with the aim of removing unwanted noise.

Introduction:

Biomedical signals, such as electrocardiograms (ECGs), are crucial for diagnosing and monitoring various heart conditions. However, these signals are often corrupted by noise during acquisition. This project aims to design and apply digital filters in MATLAB to remove noise from a raw ECG signal, improving its quality for analysis. Digital filters, such as Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters, offer powerful tools for removing noise from biomedical signals while preserving essential diagnostic information.

FIR Filters:

FIR filters produce a system's finite impulse response, meaning their impulse response is of finite length. FIR filters exhibit a linear phase response, and their system function consists solely of b coefficients. Due to the absence of poles, the system remains inherently stable. Designing higher-order FIR filters may be necessary, and they operate as open-loop systems. FIR filters are versatile, applicable to both analog and digital signal filtering, albeit with higher computational complexity compared to some alternatives.

IIR Filters:

IIR filters stand for Infinite Impulse Response, its phase response is non-linear as these filters have an internal feedback system, meaning that the system has both b_m coefficients and a_k coefficient is present. IIR filters are used in systems that produce infinite outputs, so therefore, IIR filters are represented as closed loop systems since they are utilized in systems that generate endless outputs. Despite having a reduced computational cost, the system may become unstable if the poles are located on the right.

In the context of filtering noisy biomedical signals like ECG, the choice between IIR and FIR filters hinges on factors such as computational complexity, filter characteristics, and stability requirements. IIR filters typically offer steeper roll-off in frequency response with fewer parameters, making them computationally efficient but potentially prone to phase distortion and instability. FIR filters, on the other hand, can achieve linear phase response and stability at the expense of higher computational demands due to more tap weights. Ultimately, the selection between IIR and FIR filters depends on balancing these trade-offs to best suit the specific needs of the application, such as preserving signal timing and minimizing noise while considering available computational resources.

Choosing between FIR and IIR filters

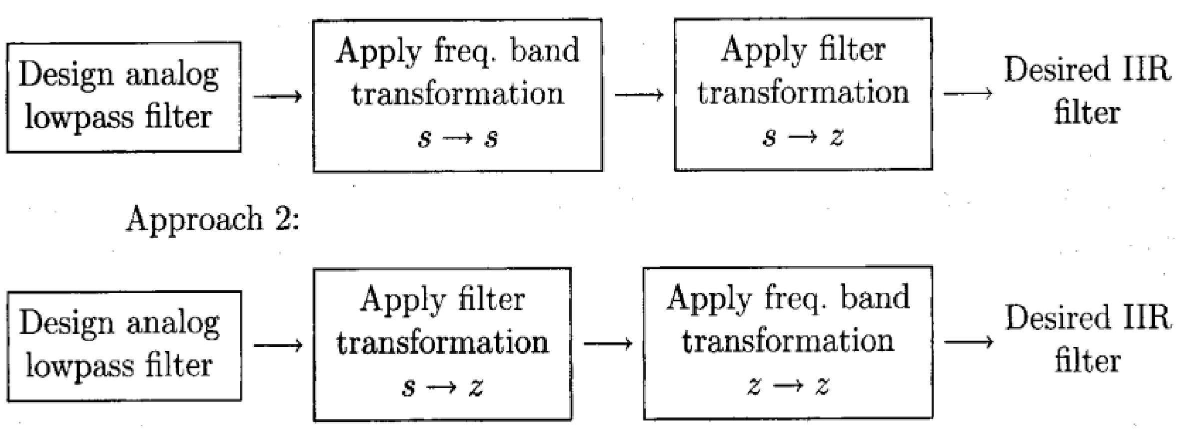
FIR filters	IIR filters
Can have exactly linear phase response, i.e. do not distort the phase of the signal.	The phase responses are nonlinear, especially at the band edges.
FIR filters realized non-recursively are always stable.	The stability of IIR filters cannot be guaranteed.
The effects of using limited number of bits to implement filters are less severe.	The effects like roundoff noise and coefficient quantization errors are more severe.
Require more coefficients for sharp cutoff filters.	Less coefficients and thus less processing time and storage.
No analog counterpart, but, easier to synthesize filters with arbitrary frequency responses.	Analog filters can be readily transformed into equivalent IIR digital filters.
Algebraically more difficult to synthesize if CAD support is not available.	Less difficult to synthesize.

Figure 1 Comparison between FIR & IIR Filters

In this project, I will be using IIR Filter as it requires lower order or less computational complexity and no separate window design.

Methodology:

There are two approaches of basic IIR filter design technique:



For this approach, I will be using the latter method. It allows for the design of an analog low-pass filter, followed by filter transformation from the s-domain to the z-domain before applying frequency band transformations, offering several benefits. Firstly, by directly transforming the analog filter prototype into the digital domain, it eliminates the need for intermediate analog-to-digital conversion steps, simplifying the overall design process. Secondly, this approach provides greater flexibility in manipulating the filter characteristics in the digital domain, allowing for precise control over parameters such as cutoff frequency, passband ripple, and stopband attenuation.

Filter Specifications:

When designing filter relative specifications are required, where Rp is the pass band ripple factor and As is the stop band attenuation in dB.

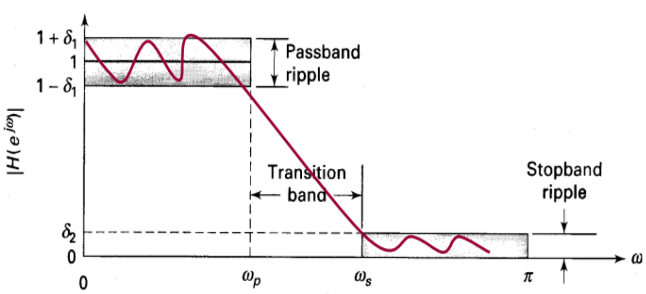


Figure 1 Absolution Specification

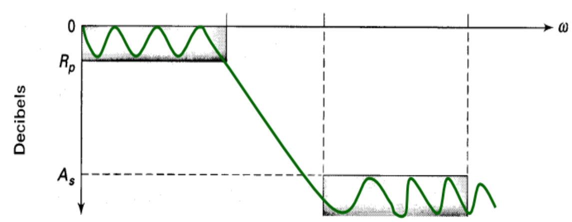


Figure 2 Relative Specification

Where δ_1 = Pass band tolerance
δ_2 = Stop band tolerance

Pass band ripple, $R_p = -20\mathrm{Log}_{10}\frac{1-\delta_1}{1+\delta_1}$

Stopband attenuation, As = $-20\mathrm{Log}_{10}\frac{\delta_2}{1+\delta_1}$

Filter Type:

There are three widely used prototype analog filters:

1. Butterworth filters
2. Chebyshev filters
    - Chebyshev type-I
    - Chebychev type-II
3. Elliptic filters.

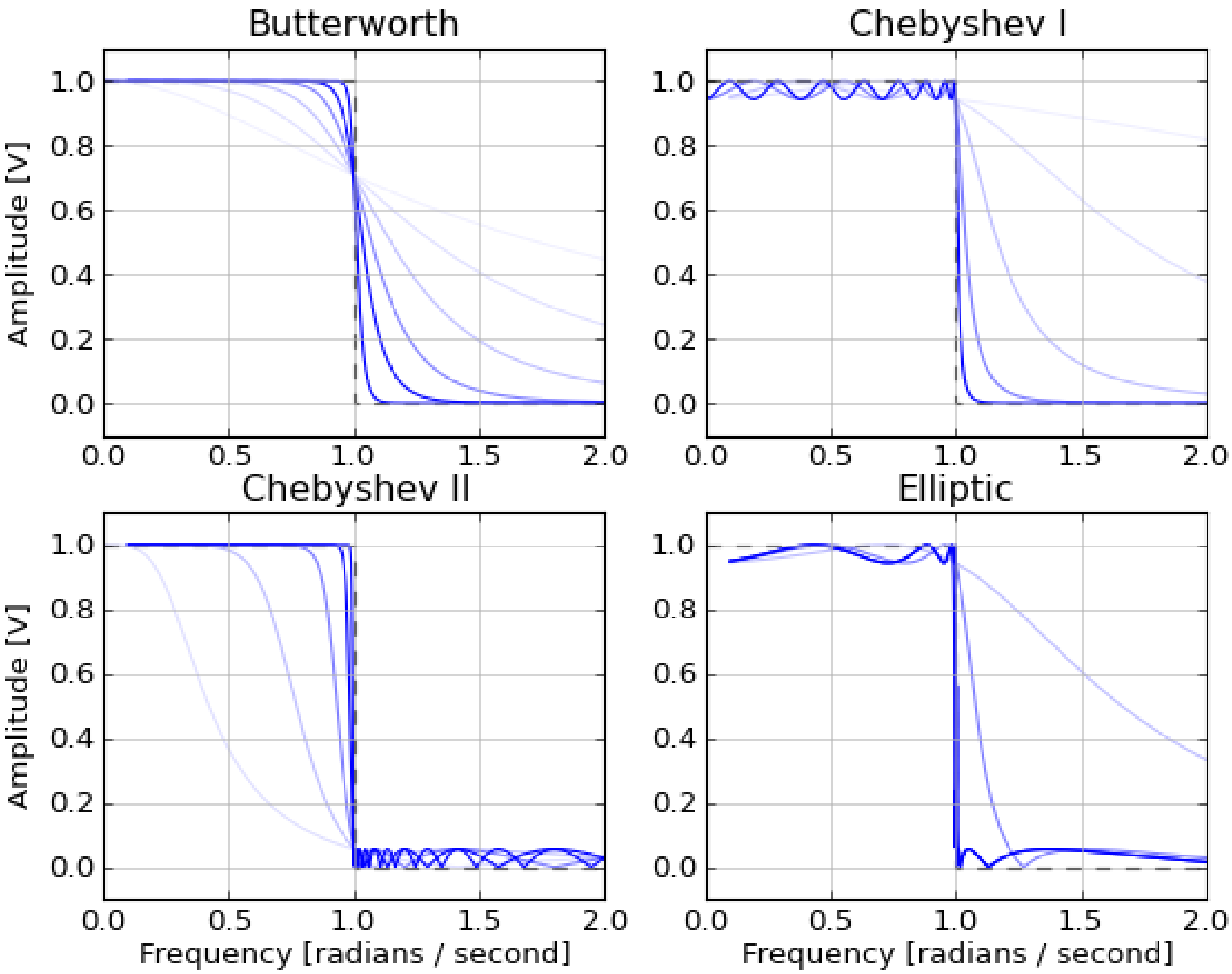


Figure 2 Frequency response of IIR Filters

Butterworth Filter: Characterized by a maximally flat passband with no ripple, providing a smooth frequency response ideal for applications requiring uniform attenuation across the passband.

Chebyshev Type I Filter: Achieves steeper roll-off in the stopband compared to Butterworth filters but introduces ripple in the passband, suitable for applications where sharp transition between passband and stopband is important.

Chebyshev Type II Filter: Provides sharp roll-off in the stopband with ripple in the stopband instead of the passband, beneficial in applications prioritizing minimal distortion in the passband.

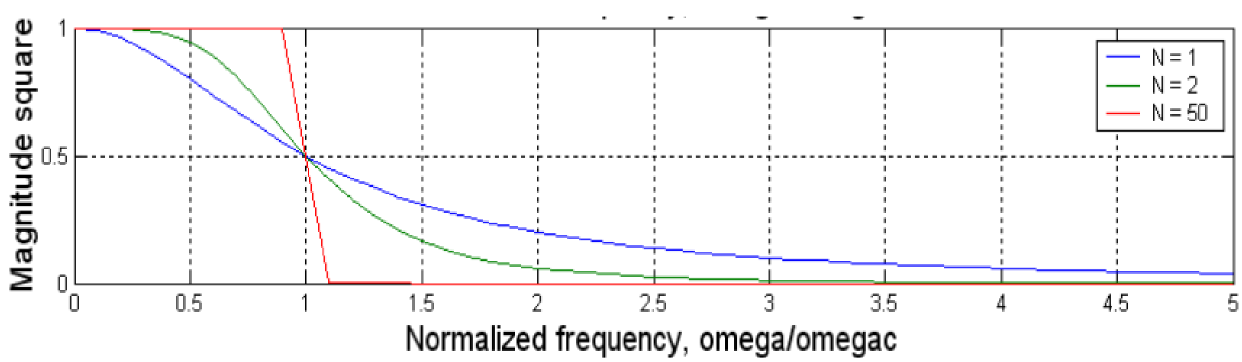
Elliptic Filter: Offers the steepest roll-off among IIR filters for a given order by allowing ripple in both passband and stopband, advantageous in applications requiring very sharp transition bands and high stopband attenuation.

Filter Type	Pass-band	Stop-band	Roll-off
Butterworth	Flat, No Ripple	Gradual	Moderate
Chebyshev Type I	Ripple	Sharp	Steep
Chebyshev Type II	Flat	Ripple	Steep
Elliptic	Ripple	Ripple	Steepest

For this project, I will be using the Butterworth filter as it provides a balance among various tradeoffs. With a flat passband characterized by minimal ripple, the Butterworth filter ensures that our desired ECG component, which typically resides in the lower frequency range, can be efficiently preserved. This makes it an optimal selection for our application. Additionally, its inherent design offers a smooth transition from the passband to the stopband, contributing to the effective suppression of unwanted higher frequency noise or interference.

The Butterworth Low Pass Filter:

A Butterworth low pass filter is usually represented by its magnitude-square response shown below, where Ωc is as the order is increased.



For the filter response in s-domain, the poles of the filter needs to be determined using either of the equations (depending on the filter order, N).

$$\boxed{p_k = \Omega_c e^{jk\pi/N}, \qquad k = 0,1,\ldots,2N-1} \quad \text{for odd } N$$

$$\boxed{p_k = \Omega_c e^{j\left(\frac{\pi}{2N}+\frac{k\pi}{N}\right)}, \qquad k = 0,1,\ldots,2N-1} \quad \text{for even } N$$

Simulation:

At first, I have plotted the time domain and frequency domain spectrum of the provided Raw and Clean ECG Signal. From the frequency domain plot of Raw ECG signal, one can observe there is a sharp monotone peak at 50 Hz.

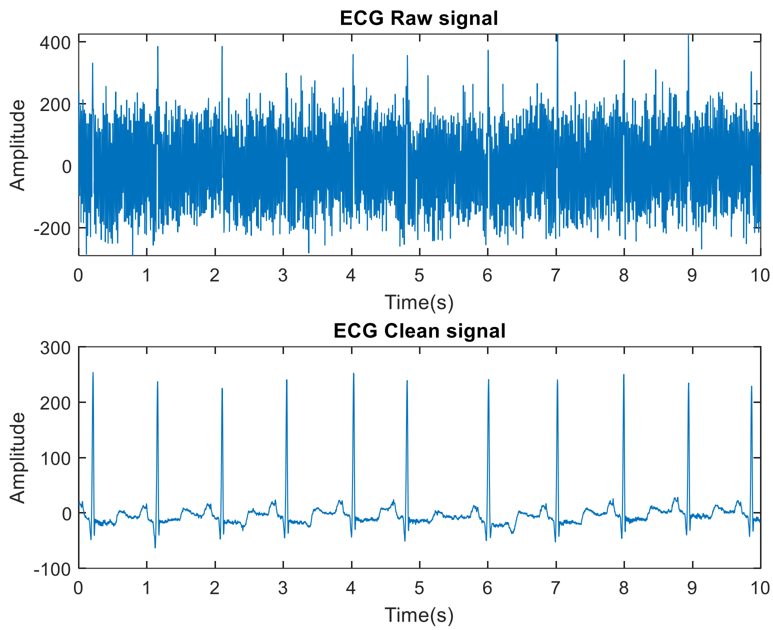


Figure 3 Time Domain Plot of Raw and Clean ECG Signal

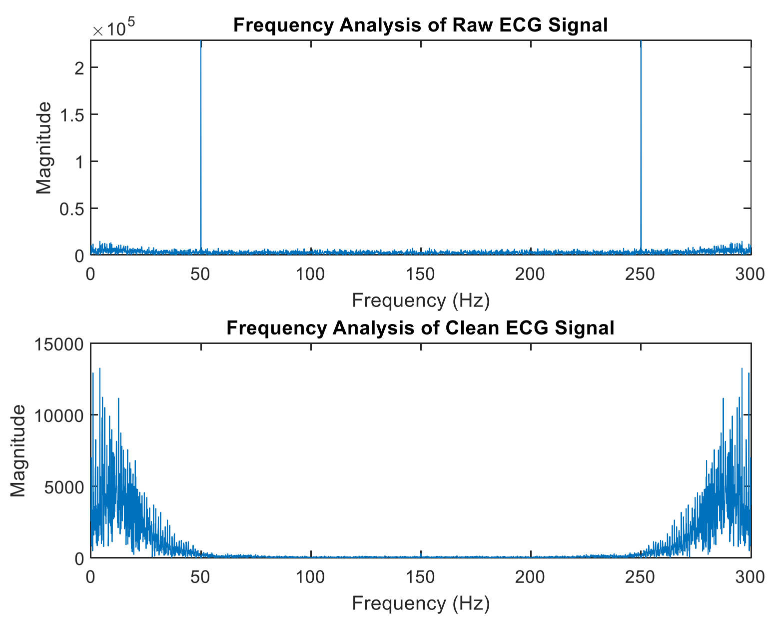


Figure 4 Frequency Domain Plot of Raw and Clean ECG Signal

To suppress the 50Hz noise in the Raw ECG signal, I utilized a Butterworth band-stop filter with cutoff frequencies set at 49 Hz and 51 Hz, and the filter order was chosen to be 3. Subsequently, I designed a Butterworth Low Pass Filter as the desired ECG signal's bandwidth lies below 100Hz. After analyzing the frequency plot of the clean ECG signal, I determined a cutoff frequency of 60Hz, as the intensity significantly decreases beyond this point. The filter order was also set to 3, based on an evaluation of performance parameters such as correlation coefficient, root mean square error, magnitude squared coherence, and SNR. Since altering the filter order didn't result in significant changes in these metrics, the priority was to choose a lower order to reduce complexity.

Filter Order vs Performance Parameters:

1 Filter_Order	2 Mean_Squared_Coherence	3 Correlation_Coefficient	4 RMSE_Before_Filtering	5 RMSE_After_Filtering	6 SNR_dB
1	0.2432	0.7668	118.8418	28.0221	-20.8219
2	0.2479	0.7587	118.8418	29.7910	-20.9175
3	0.2483	0.7510	118.8418	30.6235	-20.9522
4	0.2502	0.7464	118.8418	31.0855	-20.9684
5	0.2524	0.7434	118.8418	31.3761	-20.9775
6	0.2549	0.7413	118.8418	31.5752	-20.9832
7	0.2564	0.7398	118.8418	31.7202	-20.9873
8	0.2576	0.7386	118.8418	31.8305	-20.9904
9	0.2582	0.7377	118.8418	31.9166	-20.9929
10	0.2583	0.7370	118.8418	31.9857	-20.9950
11	0.2581	0.7364	118.8418	32.0424	-20.9969
12	0.2581	0.7359	118.8418	32.0895	-20.9984
13	0.2581	0.7355	118.8418	32.1296	-20.9998
14	0.2583	0.7351	118.8418	32.1638	-21.0010
15	0.2585	0.7348	118.8418	32.1937	-21.0021
16	0.2586	0.7346	118.8418	32.2199	-21.0031
17	0.2587	0.7343	118.8418	32.2429	-21.0040
18	0.2587	0.7341	118.8418	32.2634	-21.0048
19	0.2587	0.7339	118.8418	32.2816	-21.0056
20	0.2587	0.7337	118.8418	32.2980	-21.0062

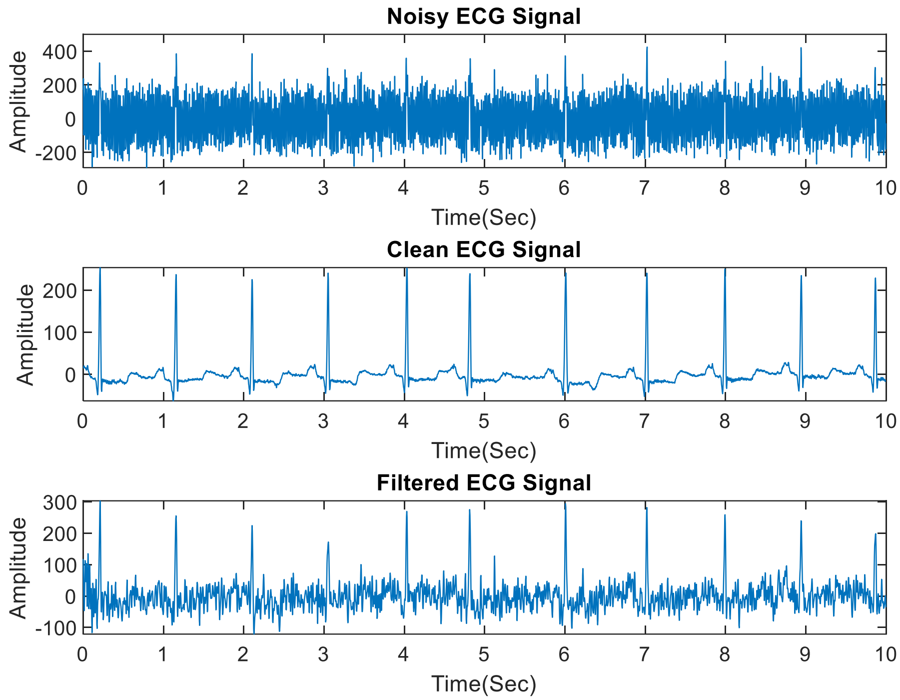


Figure 5 Time domain Plot

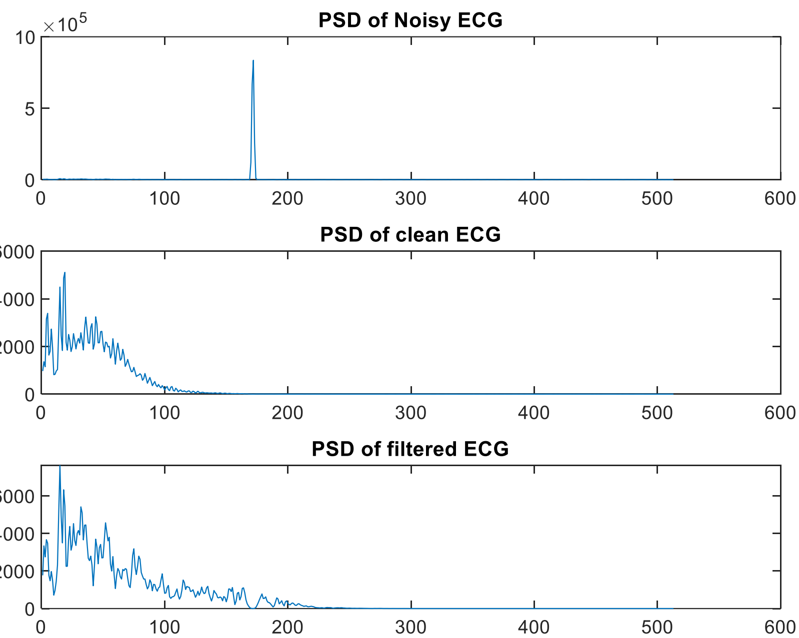


Figure 6 Power Spectral Density Plot

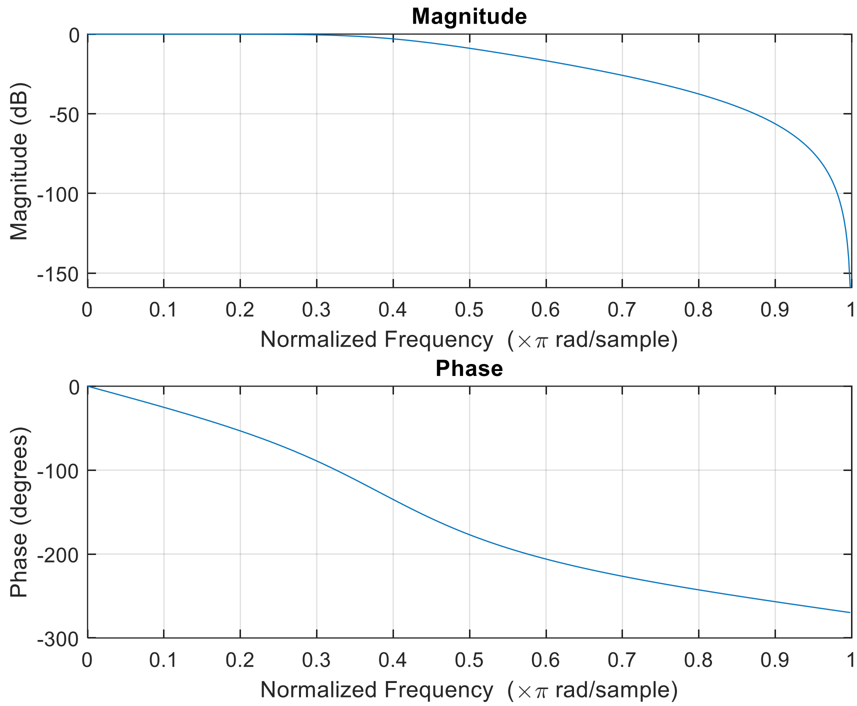


Figure 7 Magnitude & Phase response.

Matlab Code:

```
close all;
clear all;
%% Load ECG data
load('ECG_Clean.mat')
load('ECG_Raw.mat')
Fs = 300;
dt = 1/Fs;
t = (0:dt:(length(ECG_Raw)*dt)-dt)';

%% Filter
% Define the notch filter frequencies
fc1 = 49; % Lower notch frequency
fc2 = 51; % Upper notch frequency

% Design Butterworth notch filter
[b_notch, a_notch] = butter(3, [fc1 fc2]/(Fs/2), 'stop');

% Apply Butterworth notch filter to remove 50 Hz noise
ECG_Raw_notch_removed = filtfilt(b_notch, a_notch, ECG_Raw);

% Design Butterworth low-pass filter
fc_lp = 60; % Cutoff frequency for low-pass filter
[b_lp, a_lp] = butter(3, fc_lp/(Fs/2), 'low');

% Apply Butterworth low-pass filter
filtered = filtfilt(b_lp, a_lp, ECG_Raw_notch_removed);
%% PSD
```

```
PSD_r= pwelch(ECG_Raw);
PSD_c= pwelch(ECG_Clean);
PSD= pwelch(filtered);

%% Plots
% Time Doamin Plot
figure(1)
subplot(3,1,1)
plot(t,ECG_Raw)
xlabel('Time(Sec)');
ylabel('Amplitude');
title('Noisy ECG Signal');
subplot(3,1,2)
plot(t,ECG_Clean)
xlabel('Time(Sec)');
ylabel('Amplitude');
title('Clean ECG Signal');
subplot(3,1,3)
plot(t,filtered);
xlabel('Time(Sec)');
ylabel('Amplitude');
title('Filtered ECG Signal');
% Low pass filter response in magnitude and phase
figure(2)
freqz(b_lp,a_lp)
% plot PSD
figure (3);
subplot(3,1,1); plot(PSD_r); title('PSD of Noisy ECG');
subplot(3,1,2); plot(PSD_c); title('PSD of clean ECG');
subplot(3,1,3); plot(PSD); title('PSD of filtered ECG');

%% Perfomance calculation
msc= mean(mscohere(filtered,ECG_Clean));
corr= det(corrcoef(filtered,ECG_Clean));
before= rmse(ECG_Clean,ECG_Raw);
after= rmse(ECG_Clean,filtered);
r = snr(ECG_Raw,filtered);
```

Discussion:

Upon analyzing the raw ECG signal, a notable high-pitch noise at 50Hz was observed. To suppress the noise, a Butterworth band-stop filter was applied. Subsequently, I designed a Butterworth Low Pass Filter to address the high-frequency noise, considering that the desired ECG signal's bandwidth lies below 100Hz as mentioned. After analyzing the frequency plot of the clean ECG signal, a cutoff frequency of 60Hz was determined, as the intensity significantly decreases beyond this point. The filter order was also set to 3, based on an evaluation of performance parameters such as correlation coefficient, root mean square error, magnitude squared coherence, and SNR. Since altering the filter order didn't result in significant changes in these metrics, the priority was to choose a lower order to reduce complexity.

Conclusion:

In conclusion, we were somewhat successful to filter the noise from the raw ECG signal. The PSD and the time domain plot of the filtered signal resembled to the clean ECG signal. We have also compared various performance parameter such as correlation coefficients, root mean square error (RMSE), magnitude squared coherence and Signal to noise ratio (SNR). Our correlation coefficient of 0.75 indicates a positive correlation, suggesting that the filtered signal closely resembles the clean ECG signal, while the low RMSE value indicates a smaller discrepancy between the signals, suggesting that the filtering process has effectively preserved the characteristics of the original signal.

References:

[1] John G. Proakis, Dimitris G. Manolakis, and Dimitris Manolakis, Digital Signal Processing, Prentice Hall, 2006

[2] Vinay K. Ingle, John G Proakis, Digital Signal Processing Using MATLAB, Prentice Hall

[3] https://en.wikipedia.org/wiki/Digital_filter

[4] https://en.wikipedia.org/wiki/Butterworth_filter

[5] https://www.mathworks.com/help/signal/ref/butter.html

[6] https://www.mathworks.com/help/signal/ug/power-spectral-density-estimates-using-fft.html

[7] https://circuitglobe.com/difference-between-fir-filter-and-iir-filter.html

Attributes of Complex Engineering Problems		**Addressing the complex engineering problems in the project**
WP1	Depth of knowledge required (WK3-WK5, WK8)	Comprehending the working principle of a Digital Filter, FIR Filter and IIR Filter
WP2	Range of conflicting requirements	It is conflicting to choose in between FIR and IIR Filter since both has its tradeoffs.
WP3	Depth of analysis required	Determine the order and cutoff frequencies of the digital filters through analysis of the given problem and the corrupted ECG Signal
WP4	Familiarity of issues	MATLAB functions such as pwelch,corrcoef, snr, mscohere