Ahsanullah University of Science and Technology

Department of Electrical and Electronic Engineering

3^{rd} Year 2^{nd} Semester

Open Ended Lab(OEL) Report

Course No. : EEE-3218

Course Name : Digital Signal Processing-I Lab

Submitted By,

Name : Md. Mamunar Rahoman Roton

Student ID : 20200105196

Section : D1

Submitted To,

Mr. Muhammad Jakaria Rahimi

Associate Professor

Ahsanullah University of Science and Technology

Mr. Md. Adnan Quaium

Assistant Professor

Ahsanullah University of Science and Technology

Title **:** Design and analysis of Digital Filter and its application to remove noise from ECG signal.

Objective **:** The objective of this study to design, analyze, and implement digital filters for the purpose of removing noise from Electrocardiogram (**ECG**) signals. This task aims to address the critical need for accurate ECG signal processing, which plays a pivotal role in diagnosing and monitoring cardiac conditions.

Software **:** MATLAB-2019a

- Functions :

Functions Used In the Filtering Process	
Functions	Significance
Y=fft(X,n)	The fft function in MATLAB calculates the Discrete Fourier Transform (DFT) of a sequence X.
Y=fftshift(X)	fftshift rearranges the frequency components of a signal's Discrete Fourier Transform (DFT) result
Value=poly(X)	To determine the roots form the pole and zero array.
[h,f]=freqz(a,b,N,Fs)	To analyze the frequency response of a filter
Y=filter(a,b,X)	For applying the filter effect to the input signal.
Impz=(a,b,n)	To analyze the impulse response of a filter
Load('xxx.mat')	To insert the ECG data to workspace in MATLAB
Y=ceil(X)	The ceil(X) function in MATLAB rounds numbers up to the nearest greater or equal integer
W=Kaiser(L,N)	The Kaiser function in MATLAB generates a Kaiser window, a tool used to improve the accuracy of frequency analysis.

- Flow diagram :
    - PART-1:

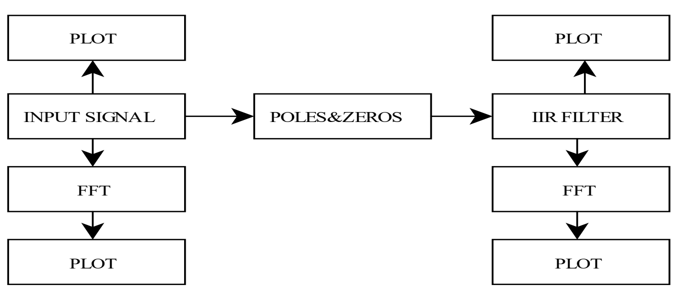

Figure 1.1 : Flow diagram for Part-1.

- PART-2 :

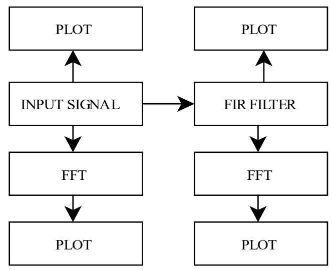

Figure 1.2 : Flow diagram for Part-2.

Procedure :

1. Filter Specification : From the input signal, to illuminate a frequency component for minimizing noise effect on a signal it is necessary to pass the given signal into a filter. There are four types of filter such as :
    - Low Pass Filter.
    - High Pass Filter.
    - Band Pass Filter.
    - Band Stop Filter or Notch Filter.

    To minimize a specific frequency component the best choice is Band Stop Filter or Notch Filter.

2. Filter Design Method :
    - IIR Filter : For better stability and efficient filtering IIR filter could be a better option for noise or unwanted frequency cancelation because it has Low Latency (minimum delay) and more flexible for designing than others.
    - FIR Filter : FIR filters are inherently stable, regardless of the filter coefficient and design and this filter can achieve exact linear phase response which means that all frequency components of the input signal are delayed by the same amount. It has predictable delay and no feedback also less sensitivity to coefficient quantization.

3. Evaluate Filter Parameters : For a specific reason there are specified filter for reducing unwanted frequency component form the given signa, that's why we have to determine the specific filter for the specific purpose and also we have to calculate the filter parameters.

4. Analyze and Visualize the Filter : After filtering the given signal is it really effective for this signal we have to optimize by analyzing and visualizing the frequency spectrum and phase spectrum of the applied filter.

5. Modify Filter Design : Sometimes our designed filter passes some unwanted frequency component after applying the desired filter. At this moment we need to tune the filter for the better result.

6. Performance and Stability Test : After completing all step for designing a filter it is necessary to analyze that the filter is stable and efficient or not.

PART-1

Result and Discussion :

- Data : For my student ID 20200105196, LMN=196 , the frequency components for the given signal.

  ```
  f1=20*(L+1)=20*(1+1)=40 Hz
  f2=13*(M+2)=12*(9+2)=13*11=143 Hz
  f3=18*(N+3)=18*(6+3)=162 Hz
  f4=10*(L+M-N+4)=10*(1+9-6+4)=10*8=80 Hz
  ```

  For even student ID, have to remove the f2 frequency component by filtering the given signal.

MATLAB CODE:

```
clc;
clear all;
close all;
%LMN=196
f1=40;
f2=143;
f3=162;
f4=80;

fs=5000; %samplig Frequency>=2*f
ts=1/fs;
t0=0.3;
t=-t0/2:ts:t0/2;
% Input Signal
signal=sin(2*pi*f1*t)+sin(2*pi*f2*t)+sin(2*pi*f3*t)+sin(2*pi*f4*t);

N=256;
df=fs/N;
% Fast Fourier Transform
X=fft(signal,N);
f=-fs/2:df:fs/2-df+df/2*mod(N,2);

figure(1)
subplot 311
plot(t,signal,'black')
xlabel('Time(s)')
ylabel('Amplitude(Volt)')
title('Given Signal')
```

```matlab
subplot 312
stem(f,abs(fftshift(X)),'black')
xlabel('Frequency(Hz)')
ylabel('Magnitude Response')
title('Magnitude Spectrum')
subplot 313
stem(f,angle(fftshift(X))*180/pi,'black')
xlabel('Frequency(Hz)')
ylabel('Phase Response')
title('Phase Spectrum')

bw=10;
w=2*pi*f2/fs;
r=1-(bw/fs)*pi;

zeros=[exp(1j*w) exp(-1j*w)];
poles=[r*exp(1j*w) r*exp(-1j*w)];

a=poly(zeros);
b=poly(poles);
% Frequency Response
[h,f]=freqz(a,b,512,fs);
figure(2)
subplot 311
plot(f,abs(h)/max(abs(h)),'black')
xlabel('Frequency(Hz)')
ylabel('Magnitude Response')
title('Frequency Response of designed Filter')
subplot 312
plot(f,angle(h)*180/pi,'black')
xlabel('Frequency(Hz)')
ylabel('Phase Response')
title('Frequency Response of designed Filter')
subplot 313
impz(a,b,100);
% Applying Filter
N1=256;
df1=fs/N1;
f1=-fs/2:df1:fs/2-df1+df1/2*mod(N1,2);
filtered_signal = filter(a,b,signal);
Y = fft(filtered_signal,N1);
figure(3)
subplot 311
plot(t, filtered_signal,'black')
xlabel('Time(s)')
ylabel('Amplitude(Volt)')
title('Filtered Signal')
subplot 312
stem(f1,abs(fftshift(Y)),'black')
xlabel('Frequency(Hz)')
ylabel('Magnitude Response')
title('Magnitude Spectrum')
```

```
subplot 313
stem(f1,angle(fftshift(Y))*180/pi,'black')
xlabel('Frequency(Hz)')
ylabel('Phase Response')
title('Magnitude Spectrum')
```

- Data Analysis : The required figures and graphs are attached below to compare the condition of the given signal before filtering and after filtering. Also attached the frequency and impulse response of the filter and signal is attached with this documentation. For this purpose we have used the notch filter to reject a specific frequency component with a very small bandwidth.

Figure(1) :

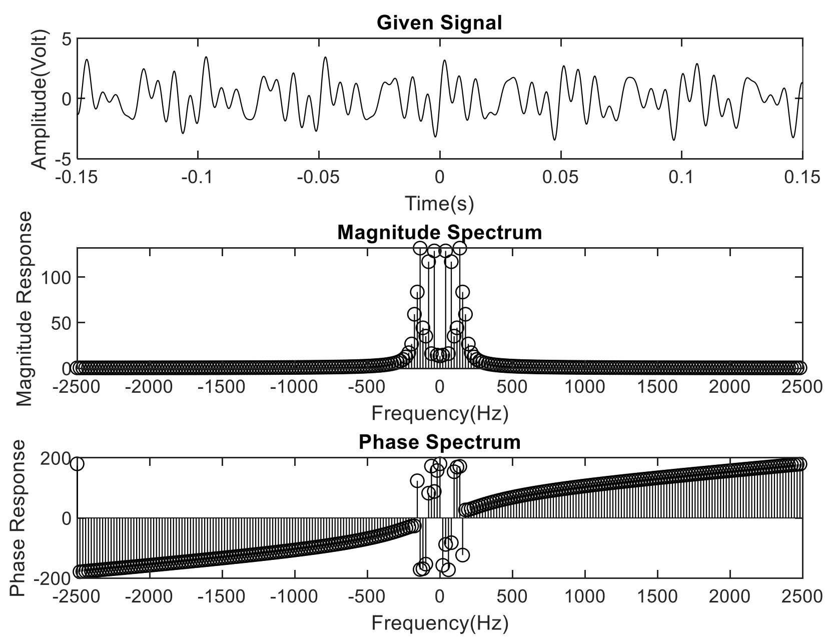

Figure 1.3 : Time domain and Frequency domain plots of generated signal

Figure(2) :

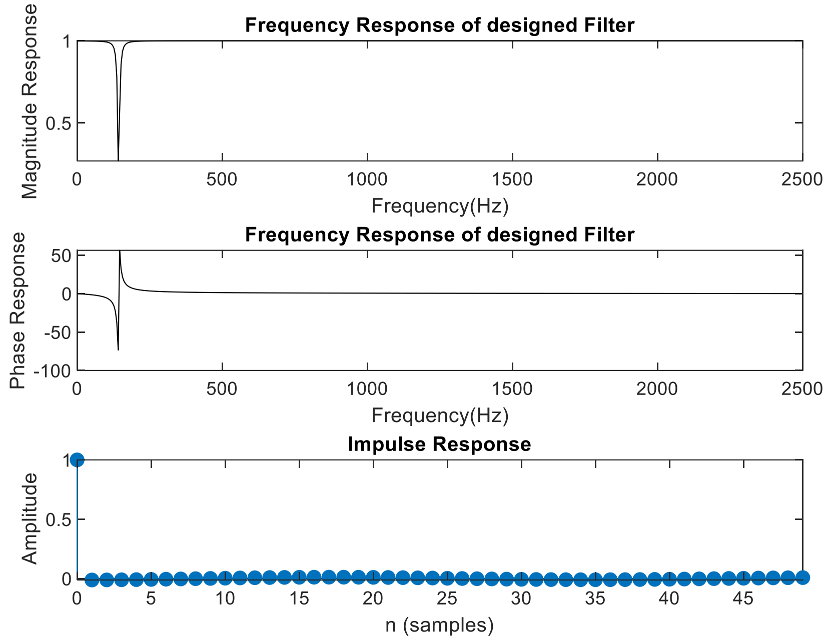

Figure 1.4 : Impulse response and frequency response of the designed IIR filter

Figure(3) :

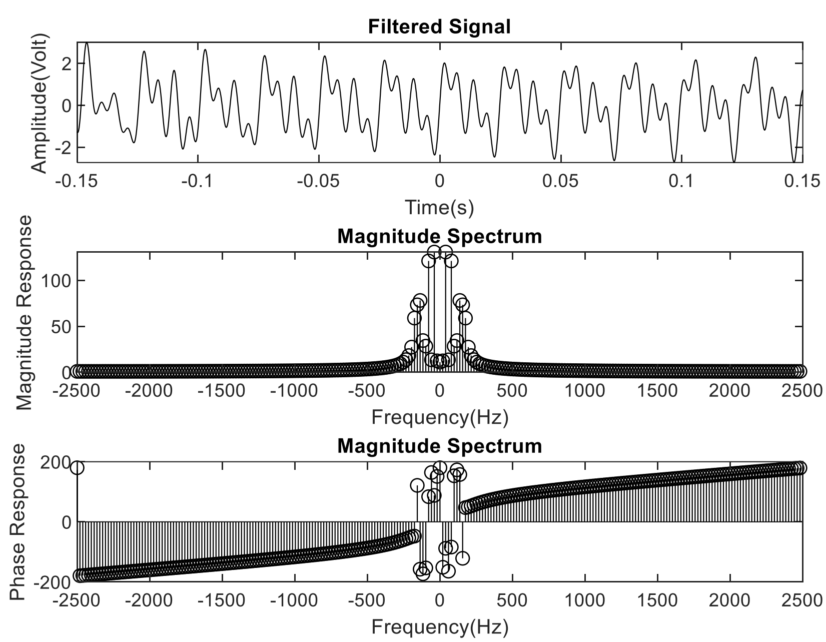

Figure 1.4 : Time domain and Frequency domain plots of the filtered signal

Discussion : This report aims to provide a comprehensive overview of Infinite Impulse Response (IIR) filters and their application in signal processing. IIR filters are widely utilized for various signal enhancement and noise reduction tasks due to their efficient computational nature.

PART-2

Result and Discussion :

- Data : For my student ID 20200105196, LMN=196 , The Electrocardiogram (ECG) data from the given ECGData.mat data sheet for me :

  Data-1 : Row Number=(L+N)=1+6=7
  Data-2 : Row Number=5*(M+N)=5*(9+6)=75

  The human ECG spectrum covers the frequency range 0.05 Hz - 100 Hz, with the main part of the QRS complex [QRS complex represents ventricular depolarization] being in the range 0.5 Hz - 45 Hz. Most of the time the ECG signal is overlaid by the mains frequency of 50 Hz/60 Hz. Low-pass filters are used to reduce high-frequency interference such as muscle tremors. These artefacts lie within the range from 5 Hz to 450 Hz. That's why we have used a low-pass filter to reduce noise or unwanted frequency component from the ECG signal. For this purpose we have used Kaiser Window Low-Pass FIR filter.

  Reference : https://academy.theortusgroup.com/en-gb/ecg-filtering-that-can-help-save-lives?fbclid=IwAR2OYYliDVPZZTGxpAuhlxgXNtYqfGGlcJUZSX4b5ZDCHPfOBoTli4AVTjE#:~:text=For%20diagnostic%20interpretation%2C%20the%20ECG,pass%20filter%20with%20150%20Hz

  The Kaiser Window Parameters :

	Kaiser Window Filter		
Transition Width (Normalized)	Stopband Attenuation(dB)	Main Lobe relative to side lobe(dB)	Passband Ripple(dB)
$\frac{2.55}{M}$	78	58	0.5

Reference : https://en.wikipedia.org/wiki/Kaiser_window

The designed filter Parameters are :

- No. of Points for Analysis,Fs=1000
- Passband frequency,Fp=40 Hz
- Stopband frequency,Fst=100 Hz
- Passband ripple,Apass=0.5 dB
- Stopband attenuation,Astop=60 dB
- Width of the transition band, transition_width=20 Hz

MATLAB CODE:

```
clc;
clear all;
close all;

% Load ECG signal (replace with your own ECG data)
load('ECGData.mat');
% For 20200105196;LMN=196
% L+N=1+6=7
% 5*(M+N)=5*(9+6)=75

ecg_signal_1=val(7,:);
ecg_signal_2=val(75,:);

% Define ECG Signal Parameter
fs = 500;  % Sampling frequency (in Hz)
N=512;
X1=fft(ecg_signal_1,N);
X2=fft(ecg_signal_2,N);
df=fs/N;
f=-fs/2:df:fs/2-df+df/2*mod(N,2);
t=0:1/fs:20;
t=t(1:length(t)-1);

figure(1)
subplot 311
plot(t,ecg_signal_1,'black')
xlabel('Time(s)')
ylabel('Amplitude(Volt)')
title('ECG Signal-I')
subplot 312
plot(f,abs(fftshift(X1)),'black')
xlabel('Frequency(Hz)')
ylabel('Magnitude Response')
title('Frequency Spectrum')
subplot 313
plot(f,angle(fftshift(X1))*180/pi,'black')
xlabel('Frequency(Hz)')
ylabel('Phase Response')
title('Phase Spectrum')
figure(2)
subplot 311
plot(t,ecg_signal_2,'black')
xlabel('Time(s)')
ylabel('Amplitude(Volt)')
title('ECG Signal-II')
subplot 312
plot(f,abs(fftshift(X2)),'black')
xlabel('Frequency(Hz)')
ylabel('Magnitude Response')
title('Frequency Spectrum')
```

```
subplot 313
plot(f,angle(fftshift(X2))*180/pi,'black')
xlabel('Frequency(Hz)')
ylabel('Phase Response')
title('Phase Spectrum')

% Define Filter Parameters
Fs = 1000;           % No. of Points for analysis
Fp = 40;             % Passband frequency (Hz)
Fst = 100;           % Stopband frequency (Hz)
Apass = 0.5;         % Passband ripple (dB)
Astop = 60;          % Stopband attenuation (dB)
transition_width = 20; % Width of the transition band (Hz)
N = ceil(2.55 / (transition_width / Fs)); % Filter order

% Calculate normalized frequencies
wp = 2 * pi * Fp / Fs;
ws = 2 * pi * Fst / Fs;


% Generate Kaiser window
kaiser_window = kaiser(N + 1);

% Calculate the ideal impulse response of the low-pass filter
n=[1:length(kaiser_window)-1];
hd = (ws/pi) * sinc(ws*(n - N/2)/pi);

% Apply the window to the ideal impulse response
h = hd.* kaiser_window(1:length(kaiser_window)-1)';

% Apply the filter kernel to the ECG signal using convolution
filtered_ecg_1 = conv(ecg_signal_1, h, 'same');
filtered_ecg_2 = conv(ecg_signal_2, h, 'same');

% Define Frequency Response
[h1,f]=freqz(h,1,1024,Fs);

figure(3)
subplot 311
plot(f(1:length(h1)),20*log10(abs(h1)),'black')
xlabel('Frequency(Hz)')
ylabel('Magnitude Response(dB)')
title('Filter Frequency Response')
subplot 312
plot(f(1:length(h1)),unwrap(angle(h1)),'black');
xlabel('Frequency(Hz)')
ylabel('Phase Respone')
title('Filter Phase Response')
subplot 313
impz(h,1)

% Apply the FIR filter
```

```
N1=1000;
df1=Fs/N1;
f=-Fs/2:df1:Fs/2-df1+df1/2*mod(N1,2);
filtered_ecg_1 = filter(h,1,ecg_signal_1);
filtered_ecg_2 = filter(h,1,ecg_signal_2);

Y1=fft(filtered_ecg_1,N1);
Y2=fft(filtered_ecg_2,N1);

figure(4)
subplot 311
plot(t,filtered_ecg_1,'black')
xlabel('Time(s)')
ylabel('Amplitude(Volt)')
title('Filtered ECG Signal-I')
subplot 312
plot(f,abs(fftshift(Y1)),'black')
xlabel('Frequency(Hz)')
ylabel('Magnitude Response')
title('Frequency Spectrum')
subplot 313
plot(f,angle(fftshift(Y1))*180/pi,'black')
xlabel('Frequency(Hz)')
ylabel('Phase Response')
title('Frequency Spectrum')
figure(5)
subplot 311
plot(t,filtered_ecg_2,'black')
xlabel('Time(s)')
ylabel('Amplitude(Volt)')
title('Filtered ECG Signal-II')
subplot 312
plot(f,abs(fftshift(Y2)),'black')
xlabel('Frequency(Hz)')
ylabel('Magnitude Response')
title('Frequency Spectrum')
subplot 313
plot(f,angle(fftshift(Y2))*180/pi,'black')
xlabel('Frequency(Hz)')
ylabel('Phase Response')
title('Frequency Spectrum')
```

- Data Analysis : The required figures and graphs are attached below to compare the condition of the given signal before filtering and after filtering. Also attached the frequency and impulse response of the filter and signal is attached with this documentation.

Figure(1) :

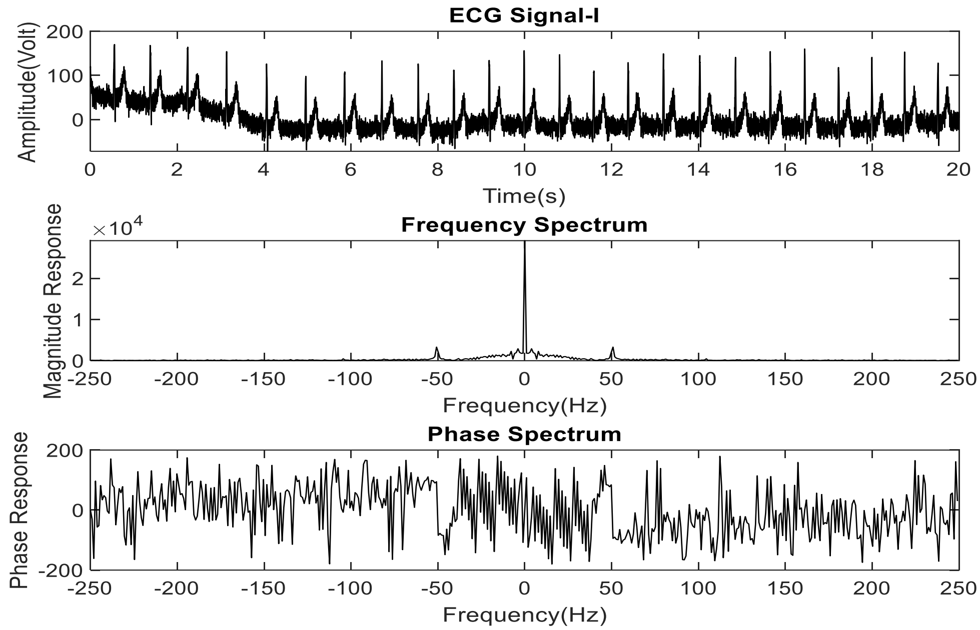

Figure 1.5 : Time and Frequency domain plot of ECG Signal-1

Figure(2) :

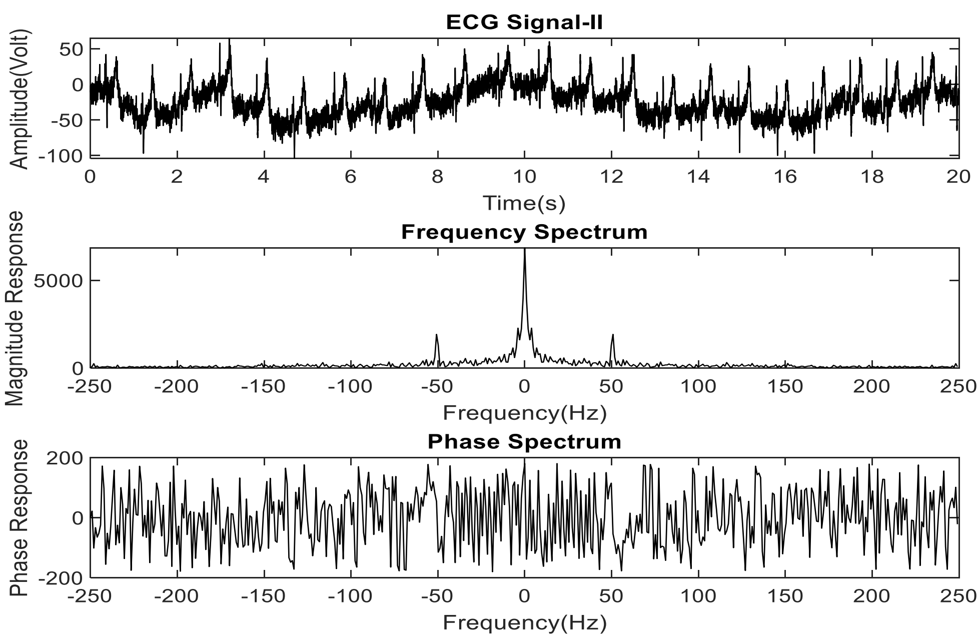

Figure 1.6 : Time and Frequency domain plot of ECG Signal-2

Figure(3) :

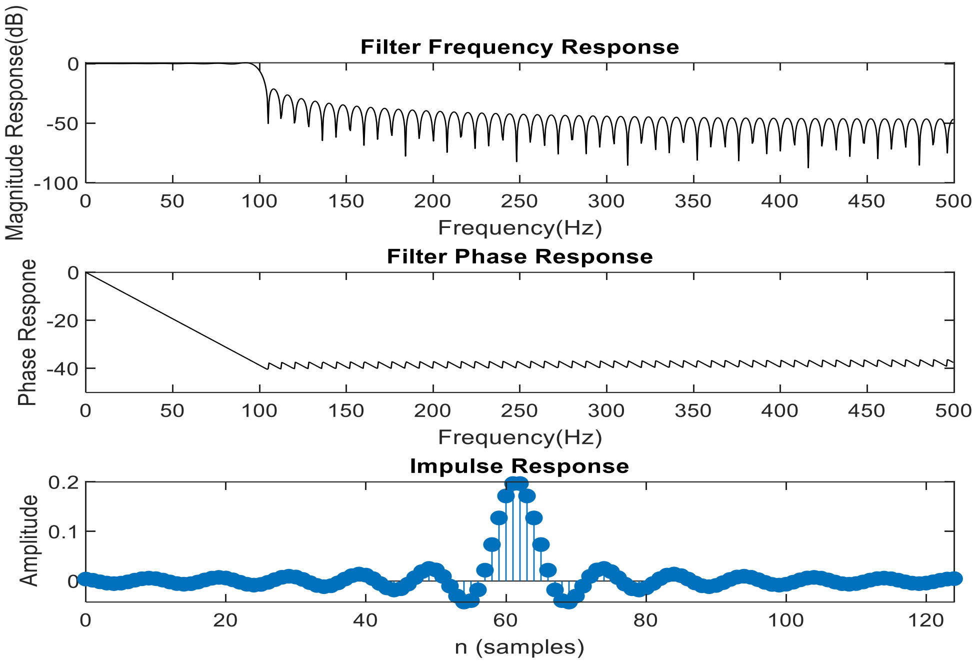

Figure 1.7: Frequency response and Impulse response of Kaiser Window FIR Filter

Figure(4) :

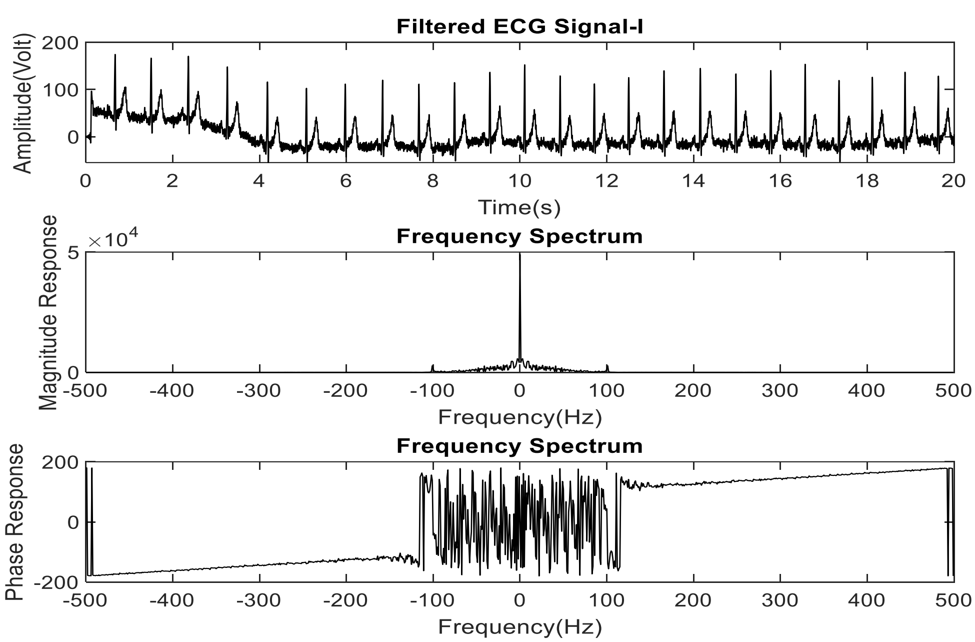

Figure 1.8: Time and Frequency domain plot of Filtered ECG Signal-1

Figure(5)   :

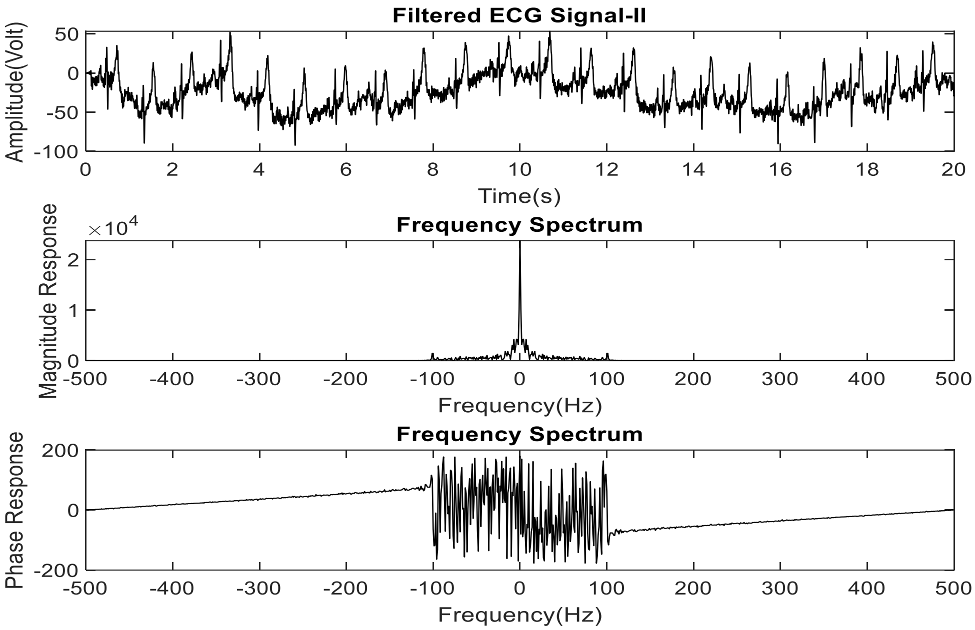

Figure 1.9 : Time and Frequency domain plot of Filtered ECG Signal-2

Discussion   : Kaiser window FIR filters are an essential tool in the realm of signal processing, offering a means to smooth abrupt transitions and enhance signal quality by employing the Kaiser window function. While their ability to soften signal edges is advantageous, it's crucial to weigh the trade-offs, particularly the potential for signal broadening and distortion. By understanding their characteristics and judiciously applying them Kaiser window FIR filters to achieve cleaner and aesthetically improved signals in various domains.

Conclusion   : In the world of signal processing, both Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) filters play significant roles, each with its own strengths and weaknesses. Through this exploration, we have gained insights into the characteristics, design considerations, and applications of these two types of filters. IIR Filter With their recursive nature, IIR filters excel in achieving efficient filtering solutions. Their ability to provide steeper roll-offs with fewer coefficients makes them ideal for real-time applications such as audio processing and communication systems. However, care must be taken in their design to ensure stability, as high-order IIR filters can be prone to instability if not well-configured. While IIR filters are efficient tools, they may introduce phase distortions and require attention to transient response settling times. On the other hand, FIR filters are known for their linear phase characteristics, making them suitable for applications where preserving the phase relationship between different frequency components is crucial. Their non-recursive structure offers stable responses and control over the filter's impulse response. Although FIR filters require more coefficients for similar performance compared to IIR filters, they offer precision and can be well-suited for scenarios where detail preservation and customization of the frequency response are paramount.