Electrical And Electronic Engineering

Course Code and Title:

EEE 309: Digital Signal Processing (DSP)

Semester and Year - Fall 2022

Section - 01

Project on Digital Filter Design

(Group - 01)

Students Name and ID :

S. M. Abu Nayem	2015-2-80-074
Shamsur Joha Shams	2018-3-80-032
Mubeen Abdullah	2019-1-80-072
Maisha Nawshin	2019-2-80-027
Kazi Sadia	2019-1-80-077

Name of the Lab Instructor :

Dr. Halima Begum

Assistant Professor

Submission Date: 14 January, 2023

Introduction:

ECG data is often corrupted with different types of noises, such as power line interference, electrode contact noise, instrumentation noise, electrosurgical noise, etc. The objective of this project is to design an appropriate digital filter that can suppress the noise in ECG data mainly due to power line interference. The filter must be designed to minimize implementation cost and consider implementation structure using a fixed precision algorithm.

Course Outcome:

By doing this project we can achieve the following course outcome (CO):
CO4: Design filters subject to different specification and constraints

Observation and analysis of given signal:

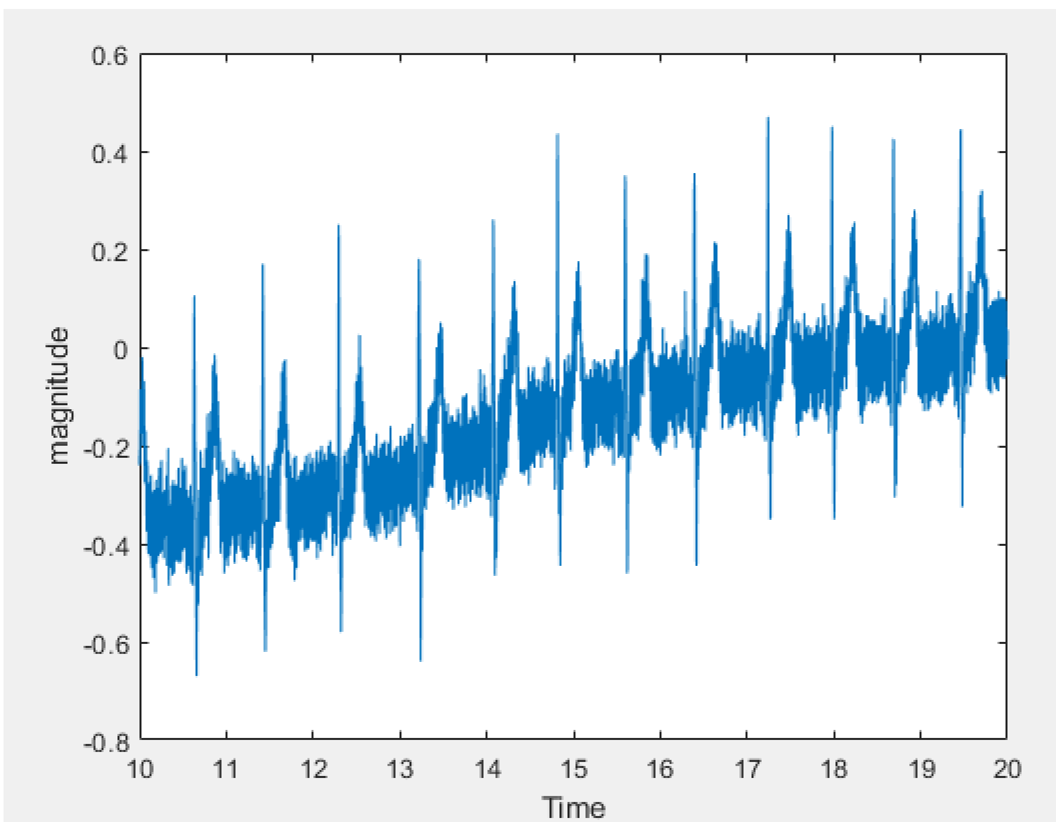

Figure-01 : Time domain signal of corrupted ECG data.

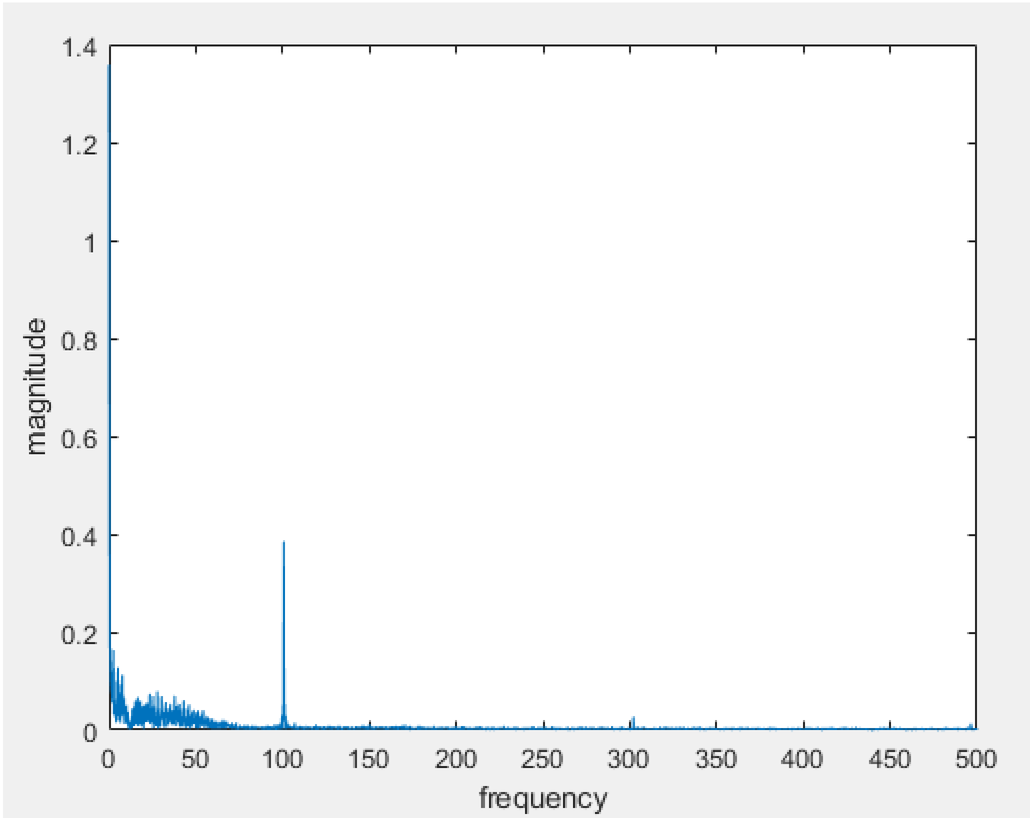

Figure-02 : Frequency domain signal of corrupted ECG data.

We have got a spike at frequency = 100 Hz
Here, we can see that noise in the ECG data value gradually decreases after 70Hz and after 90Hz amplitude is nearly zero. So, we can say that, on the given signal, ECG information lies between 0-70Hz. So, if we pass 0 to 70 Hz frequency components then we will get a noise-free appropriate ECG signal.

Filter selection:

We have to design a filter that is able to pass frequencies from 0-70Hz and stops frequencies
greater than that so that interference gets suppressed.
We must employ either a FIR or an IIR low pass filter to do this. We've gone with a **FIR low pass filter** in this case

Reason Behind Choosing FIR Filter:

FIR filter is a linear phase system filter where IIR filter implementation requires less memory so it is cost efficient but IIR filter shifts the phase angle. FIR filters require higher memory but it maintains linear phase response which will be used in the medical field where accurate information is very much important. So, we will use **FIR Low-pass filter** because of its linear phase response.

Filter Design:

Let,

Sampling Frequency, Fs = 1000 Hz

Passband edge frequency, Fpass = 70Hz

Stopband edge frequency, F Stop = 100Hz

Transition width, ΔF = 30Hz

So, Cut-Off Frequency, Fc = $\frac{70 + 100}{2}$ = 85 Hz

Passband ripple, δp = 0.01

Stopband ripple, δs = 0.001

Stop band attenuation,A = $-20\log10(\delta)$

= $-20\log10(0.001)$

= 60dB

According to the stopband attenuation, Blackman and Kaiser Window satisfy the stopband attenuation criteria.

Calculation for Blackman Window:

As we have, transition width, $\Delta F = 30Hz$

Sampling Frequency, $Fs = 1000$ Hz

So, $\Delta f = \frac{30}{1000} = 0.03$

We know for Blackman window transition width (normalized) = $\frac{5.5}{M+1}$

$\frac{5.5}{M+1} = \Delta f = 0.03$

$M+1=183.33 \approx 184$

So, M=183

So, Filter length M+1=184

So, Filter order M=183

Calculation for Kaiser Window:

From our selection Parameter, Passband Ripple $\delta p=0.01$

Stopband Ripple, $\delta s=0.001$

The Kaiser filter must be designed to meet the smaller of the two ripple constraints:

$\delta = \min \{\delta p, \delta s\}$

As our stopband ripple is less than the pass band ripple so, $\delta = \delta s = 0.001$

Now we know, $A = -20\log10\ (\delta)$

$= -20\log10(0.001)$

$= 60dB$

The value of Kaiser Parameter, β =0.1102(A-8.7)

=0.1102(60-8.7)

=5.65326

Window length for Kaiser Window, $M+1 \geq [\ 1+ \frac{A-8}{2.285 * \Delta\omega}\ ]$

Here, $\Delta\omega$ = Transition width, ($\Delta f * 2\pi$)

$$M+1 \geq [\ 1+ \frac{60-8}{2.285 * 2 * pi * 0.03}\ ]$$

=122.72 ≈ 123

So, filter length, M+1=123

Filter order, M =122

Selection of Window(Kaiser Window) :

From the calculation above, we can see that the Filter length and Filter order of the Kaiser Window is much smaller than the Blackman window which is an advantage for us. The window length mainly affects the transition band. So, keeping the window length constant, we can adjust the passband and stopband ripples. Other windows cannot be used as they do not fulfill all the requirements here.

Filter implementation in MATLAB:

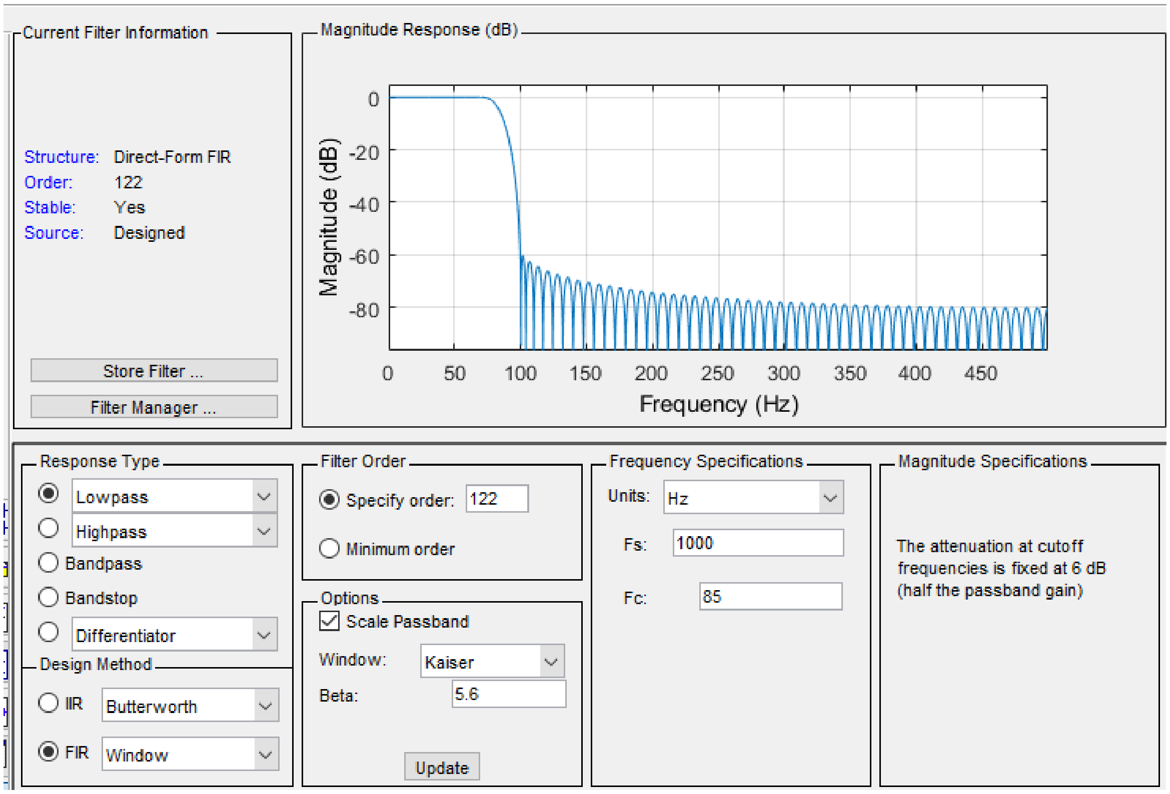

Figure-03 : Low-pass filter parameters and magnitude response. (In dB)

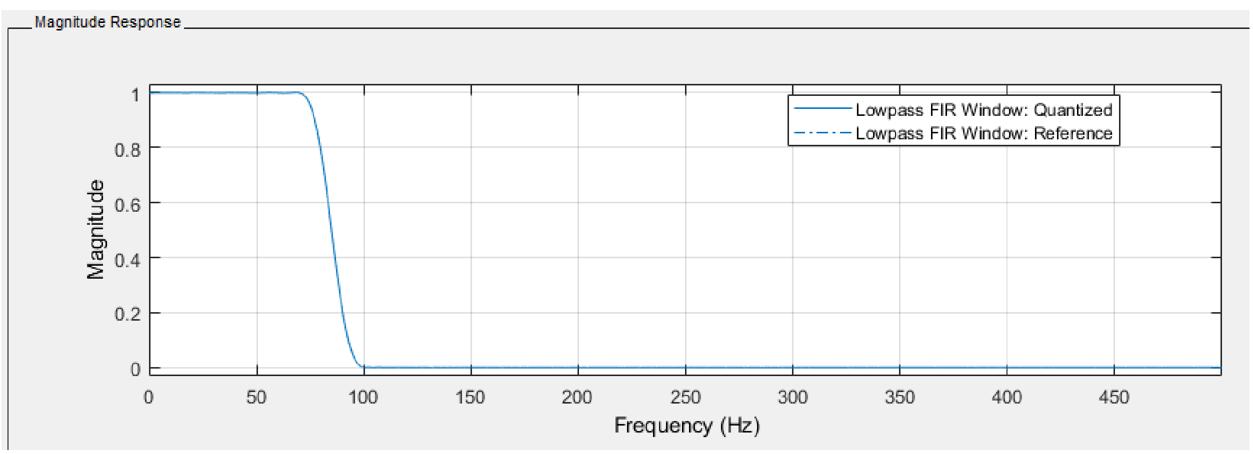

Figure-04 : Low-pass filter parameters and magnitude response.

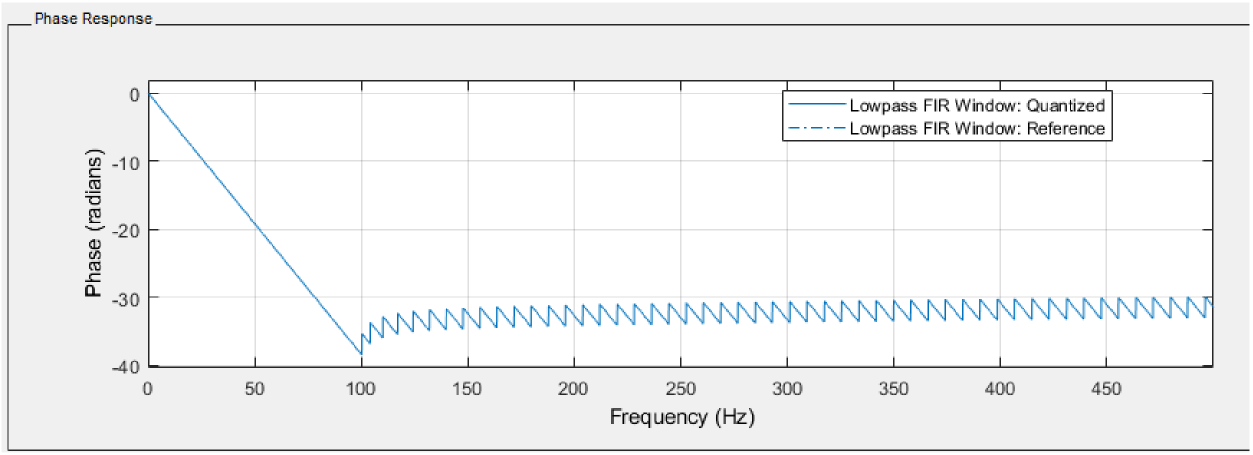

Figure-05: Phase response of our low pass filter.

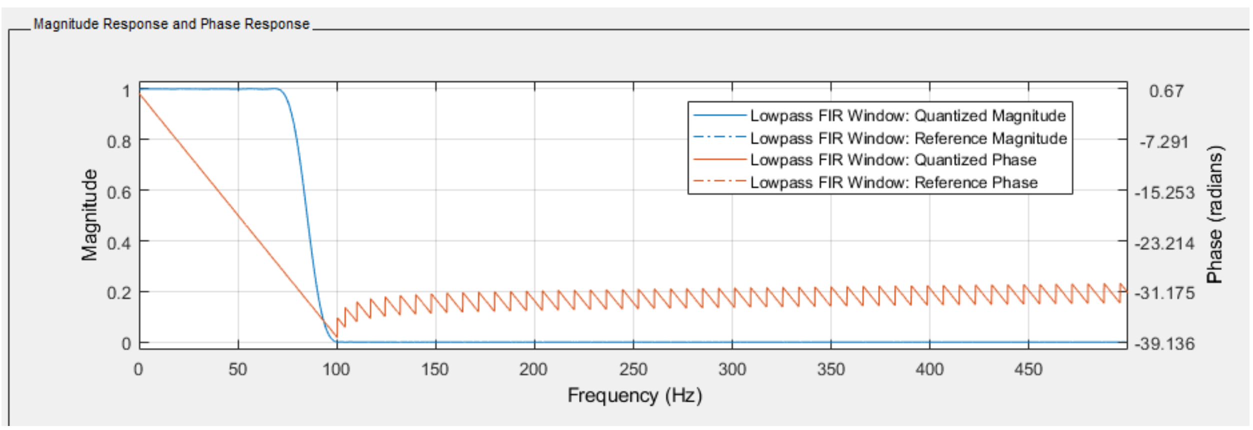

Figure-06: Magnitude and Phase response of our low pass filter.

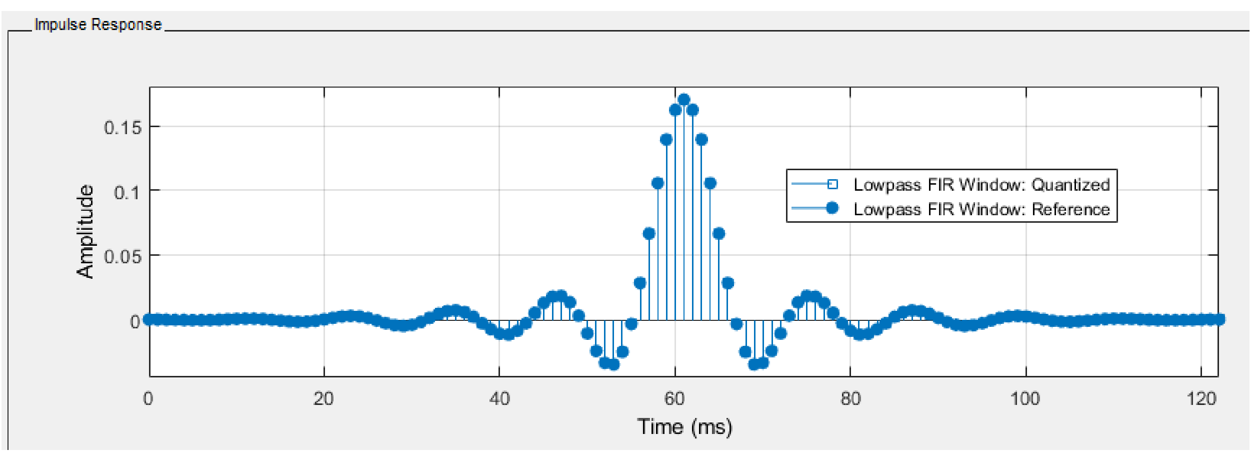

Figure-07 : Impulse response of our low pass filter

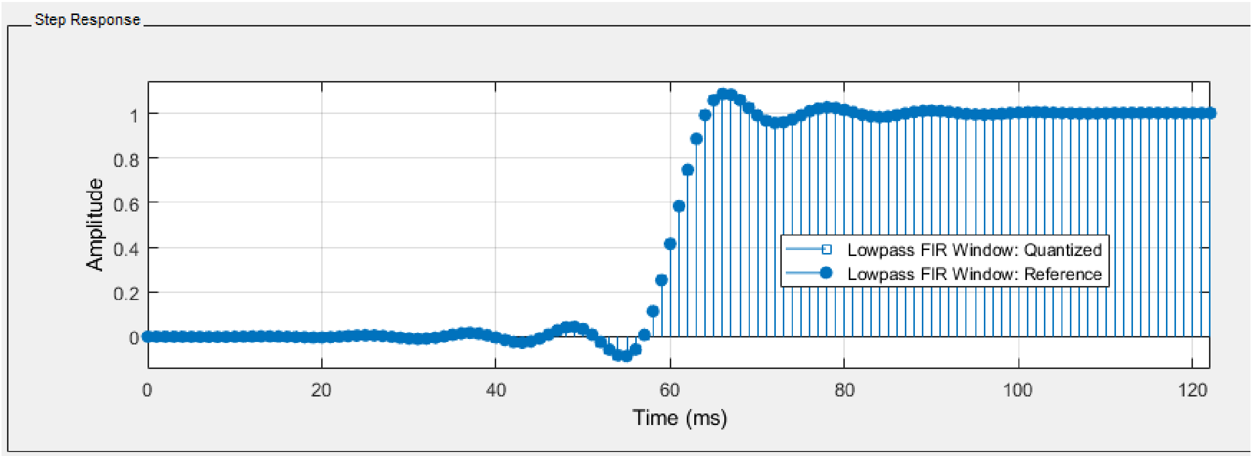

Figure-08 : Step response of our low pass filter.

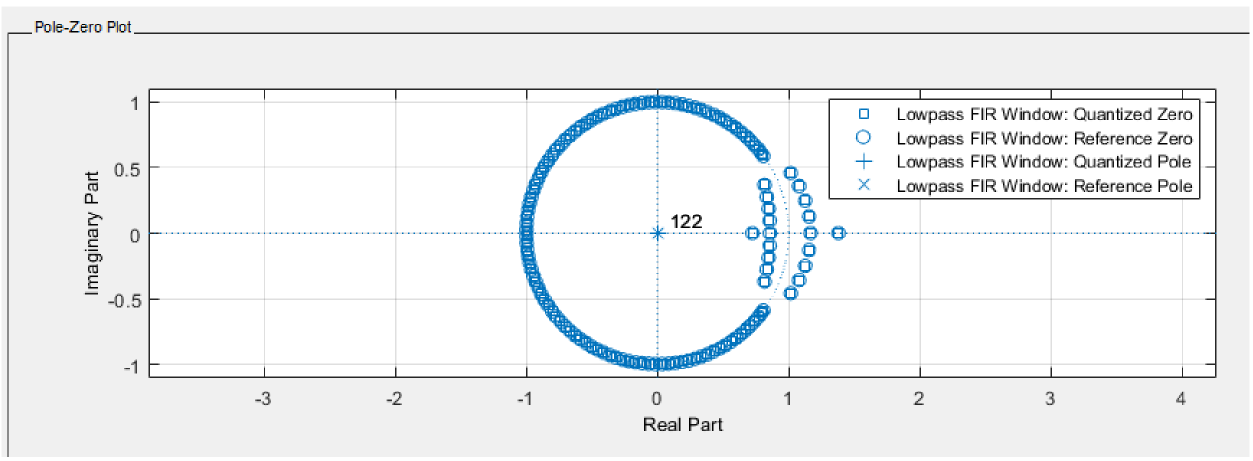

Figure-09: Poles And zeroes of our designed low pass filter

```
Discrete-Time FIR Filter (real)
-------------------------------
Filter Structure  : Direct-Form FIR
Filter Length     : 123
Stable            : Yes
Linear Phase      : Yes (Type 1)
Arithmetic        : single

Implementation Cost
Number of Multipliers            : 123
Number of Adders                 : 122
Number of States                 : 122
Multiplications per Input Sample : 123
Additions per Input Sample       : 122
```

Figure-10 : Filter information of our low pass filter

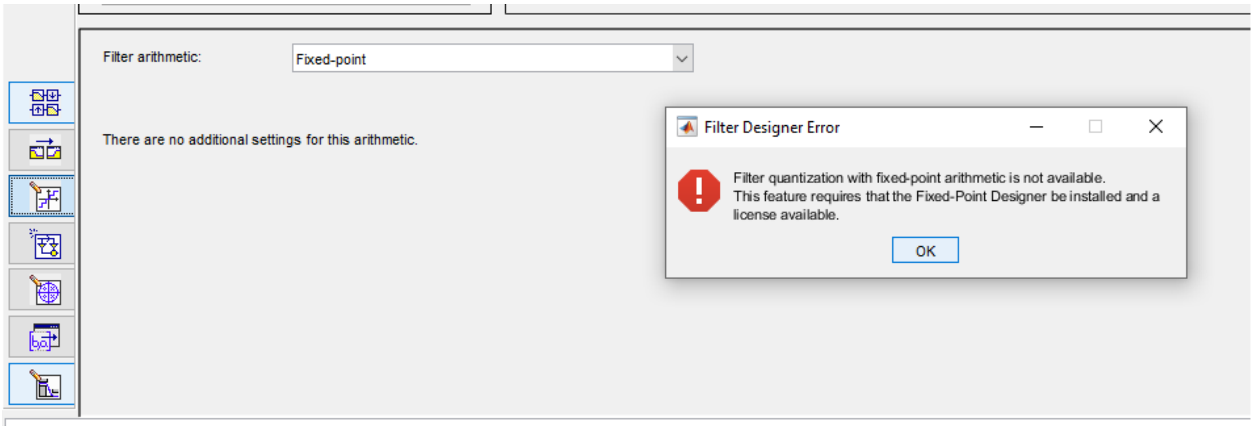

Figure-11 : Error Message for fixed point designer

We also get one error message while setting *fixed-point precision* that can decrease numerical accuracy later. So, now that reason we work on *double-precision floating point*.

<u>Filter Performance Analysis:</u>

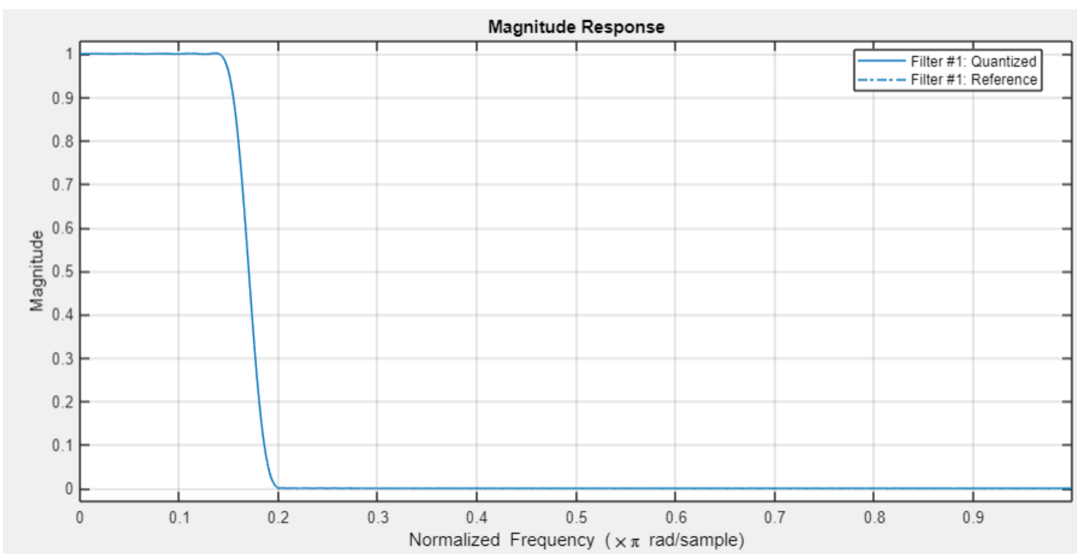

Figure-12 : Magnitude Response of Given Filtered ECG signal.

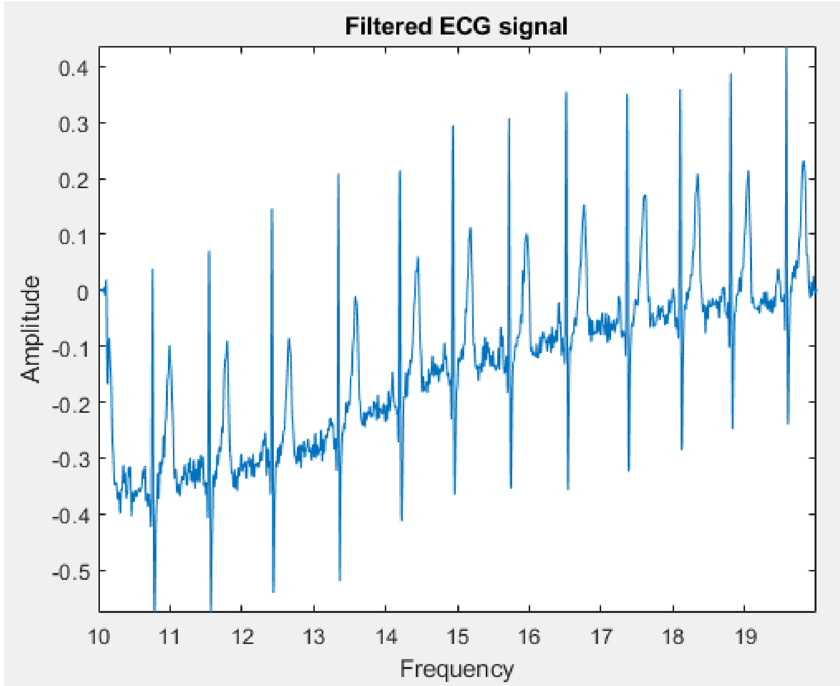

Figure-13 : Filtered ECG signal using our low pass filter

<u>Observation:</u>

Now we can observe from **Figure-12** that, the noises are removed and finally we get a clear signal from the corrupted ECG signal.

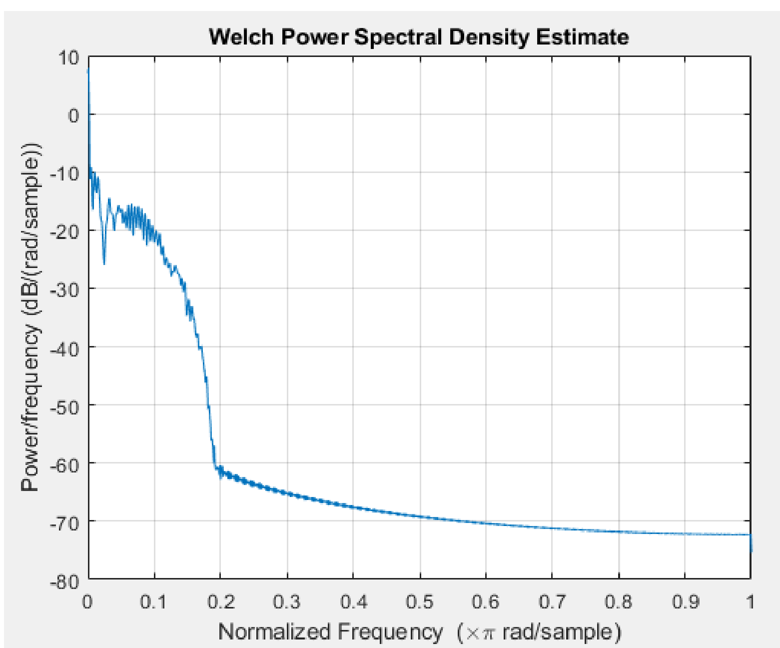

Figure-14 : Power Spectral Density Estimate spectrum of Given Filtered ECG signal.

Comment:

We can see the graph is comparatively smoother and the noises are suppressed. We have seen in the frequency domain that we got a spike at 100 Hz in the corrupted signal. So here we try to filter out that noise.

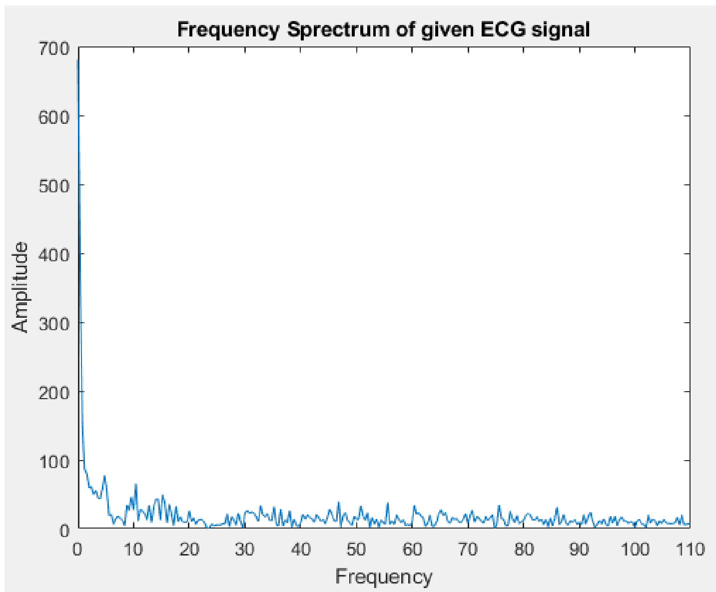

Figure-15 : Frequency spectrum of Given Filtered ECG signal

Comment:

Now here we tried to remove noise from the noise-corrupted ECG signal.
After this observation, we can say that we are finally able to filter our given corrupted ECG signal successfully.

Further Improvements:

We created our Low Pass Filter such that it only requires a small number of multipliers and adders, making it both cost-efficient and effective in filtering out undesired signals. Here, we have used a Kaiser window to design the filter but if we use a customized window defined by us as per our requirements then we may get a more cost-efficient and appropriate filter.

Limitations:

1. More memory is required for the FIR filter than for the IIR filter.
2. It is not cost-efficient.
3.The speed is slower than other filters because more numerical calculations are required.

Conclusion:

From this project, we learned how to design a digital filter to remove unwanted noise from a signal. With all this plot and calculation, we can conclude that we try to design the appropriate filter that fulfills all the given criteria. Keeping in mind the cost minimization, we can design any demandable type of FIR filter for any given criteria using the concept of this project. Overall, this digital filter design project will be really helpful in future projects.

Appendices:

Code for analyzing the corrupted (given) signal:

```
clc;
close all;
clear all;
M=load('C:\Users\User\Desktop\10th-Fall''22\309\Group_01.csv');
Fs=1000;
x1=M(:,1);
y1=M(:,2);
figure(1)
plot(x1,y1)
xlabel('Time')
ylabel('magnitude')
y2=y1/500;
L=length(y2);
NEFT=2^nextpow2(L);
y2_fft=abs(fft(y2,NEFT));
freq=Fs/2*linspace(0,1,NEFT/2+1);
figure(2)
plot(freq,y2_fft(1:length(freq)));
xlabel('frequency')
ylabel('magnitude')
```

Code for filter and analyze the signal:

```
clc
clear
close all
data = load('C:\Users\User\Desktop\10th-Fall''22\309\Group_01.csv'); %given ecg
signal
y= data(:,2);
t= data(:,1);
fvtool(lowpassfilter); %loading filter info.
filt= filter(lowpassfilter,y); %filtering the ecg signal
figure(2)
plot(t,filt)
xlim([1,10])
title ('Filtered ECG signal');
axis tight
xlabel ('Frequency');
```

```
ylabel ('Amplitude');
figure(3)
pwelch(filt) %power spectral density of filtered signal
L= length(filt);
magnitude = abs(fft(filt));
magnitude = magnitude(1:L/2);
freq = ((0:1/L:1-1/L)*2000)';
freq=freq(1:L/2);
figure(4)
plot(freq,magnitude)
xlim([0,110])
title ('Frequency Spectrum of given ECG signal');
xlabel ('Frequency');
ylabel ('Amplitude');
```

Function :

```
function Hd = lowpassfilter
%LOWPASSFILTER Returns a discrete-time filter object.
% MATLAB Code
% Generated by MATLAB(R) 9.12 and DSP System Toolbox 9.14.
% Generated on: 13-Jan-2023 21:13:32
% FIR Window Lowpass filter designed using the FIR1 function.
% All frequency values are in Hz.
Fs = 1000;  % Sampling Frequency
N    = 122;      % Order
Fc   = 85;       % Cutoff Frequency
flag = 'scale';  % Sampling Flag
Beta = 5.6;      % Window Parameter
% Create the window vector for the design algorithm.
win = kaiser(N+1, Beta);
% Calculate the coefficients using the FIR1 function.
b  = fir1(N, Fc/(Fs/2), 'low', win, flag);
Hd = dfilt.dffir(b);
set(Hd, 'Arithmetic', 'single');
% [EOF]
```