EAST WEST UNIVERSITY

Department of Electrical & Electronic Engineering

EEE 309 – Digital Signal Processing

Project report

Project on Digital Filter Design

Course Instructor : Dr. Halima Begum

Group - 6

Section - 2

Date of Allocation : 27 May 2024

Date of Submission : 03 June 2024

Student Name : Sakib Hasan Mitul
Student ID : 2020-1-80-092

Table of Contents

Objective

The goal of this project is to create a digital filter designed to reduce power line interference in ECG data.This project focuses on developing a filter that can efficiently reduce this interference while maintaining the quality of the ECG signal. The filter design will prioritize low implementation cost and compatibility with fixed precision algorithms, making it suitable for use in devices .The project will ensure the filter is computationally efficient and can be implemented within hardware constraints, ensuring its practicality and scalability for real-world use.

Introduction

Electrocardiogram (ECG) signals are crucial for diagnosing cardiovascular diseases, providing valuable information about the heart's electrical functions. However, these signals often face various types of noise, such as power line interference, electrode contact noise, instrumentation noise, and electrosurgical noise. Among these, power line interference is particularly problematic, as it can significantly distort ECG data, leading to inaccurate interpretations. This project aims to tackle this issue by developing an effective filter to suppress power line interference, thereby enhancing the accuracy and reliability of ECG readings.This is especially important in clinical settings where precise signal quality is necessary for accurate diagnosis and treatment planning. Additionally, the project will focus on creating a filter that is not only effective but also efficient, capable of being implemented in various medical devices, including portable and wearable monitors. This versatility ensures that the benefits of enhanced ECG signal clarity can be widely accessible, contributing to better patient outcomes and advancing the field of biomedical signal processing.

Observation and analysis of given signal

If we plot our given corrupted ECG signal in time domain,

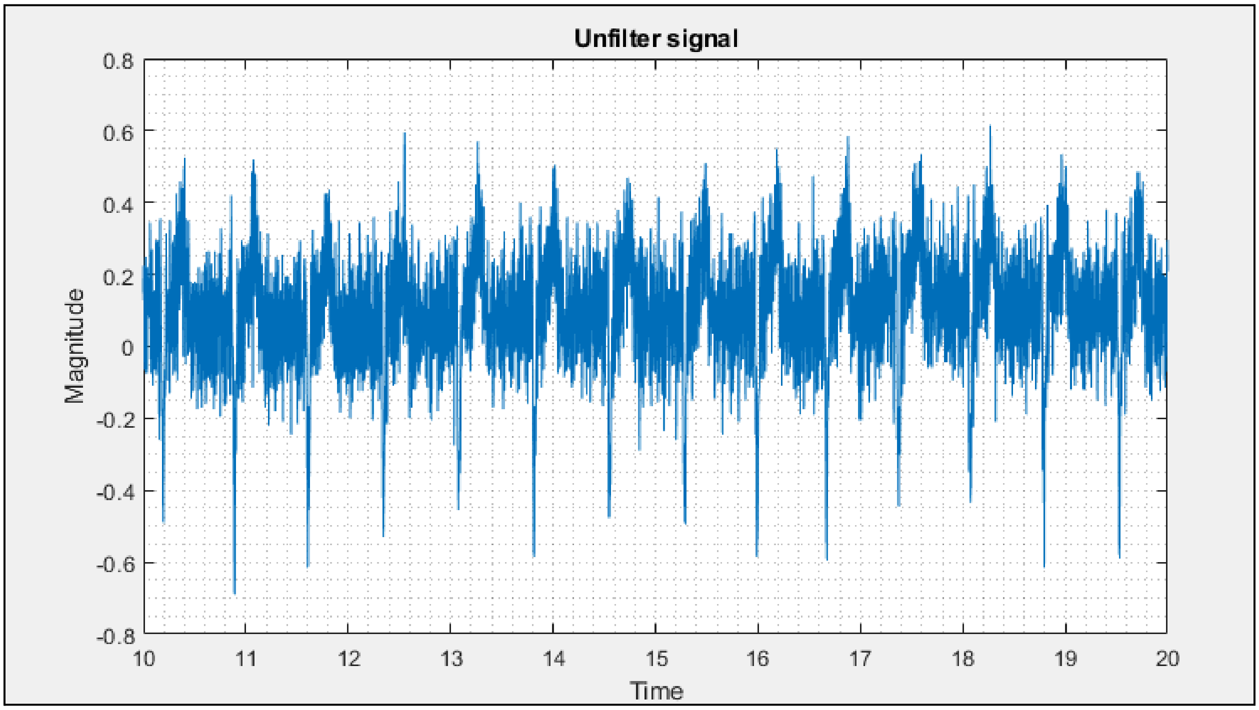

Figure 1: Noise Corrupted ECG signal in time domain

As we can see there is so much noise present in the signal. In the ideal case the signal has to be clearer.

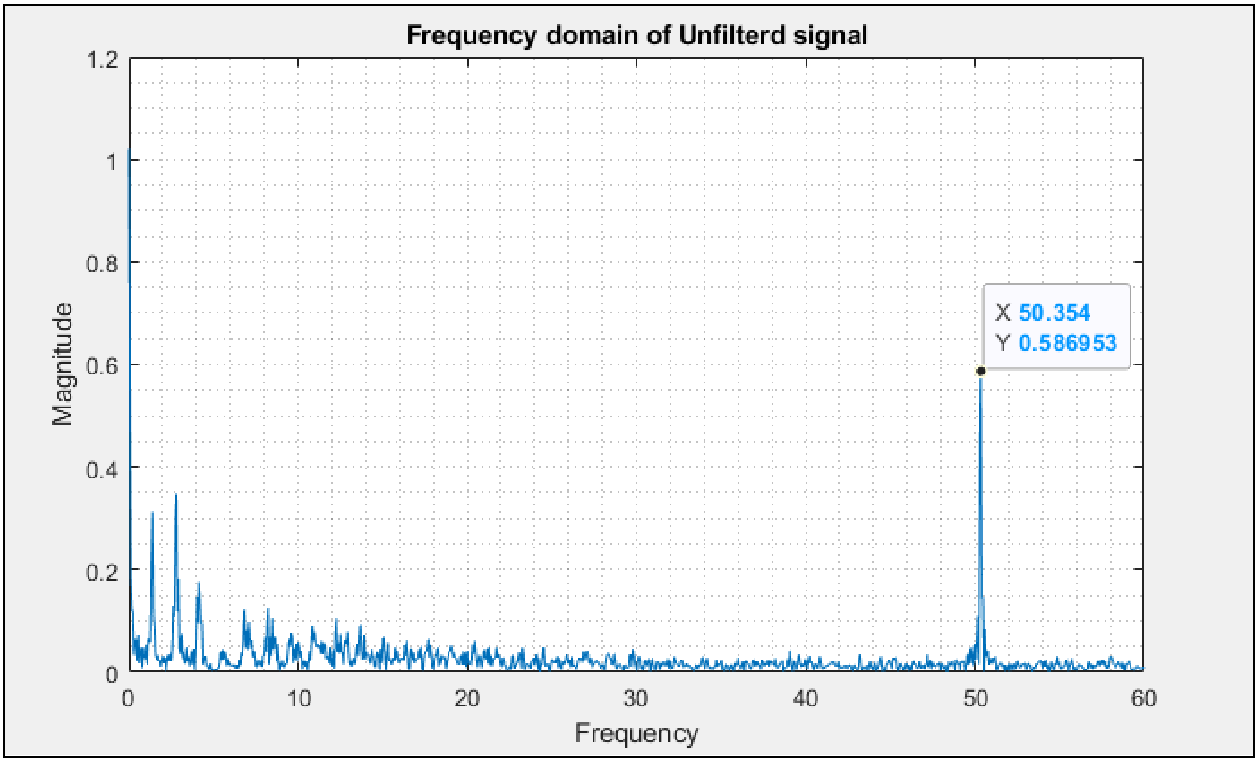

Figure 2: Frequency component analysis of corrupted ECG Signal

If we look at Figure 2, the ECG signal starts to fade around 35 Hz and is almost gone by 40 Hz. This tells us that the important information in the ECG is mostly between 0 and 40 Hz. So, if we only let frequencies from 0 to 40 Hz through, we should end up with a cleaner and more accurate ECG signal .

Filter selection

We want to make a filter that lets through only frequencies between 0 and 40 Hz and blocks anything higher, including the 50 Hz power line interference. To achieve this, we've decided to use a FIR low-pass filter.

Calculation for designing the filter

For designing a low pass filter the parameters are given below

From figure,

Passband edge frequency, F_p = 40 Hz

Stopband edge frequency, F_s = 48 Hz

Transition width, $\Delta F = (F_s - F_p)$ Hz

= 8 Hz

From ECG data sampling period, $T_s= 0.002$ s
Sampling frequency, $F_s=(1 / T_s)$
$=(1 / 0.002)$
$=500$ Hz
Cutoff frequency, $F_c= F_p+(\Delta F / 2)$
$=40+(8 / 2)$
$=44$ Hz
Let's consider
Passband ripple, $\delta p =0.01$
Stopband ripple, $\delta s = 0.001$

We know,
Stopband attenuation, $A = - 20\log10 (\delta)$
$\therefore$ $A= -20\log10 (0.001)$
$= 60$ dB

Since, stopband attenuation is 60 dB we can select Blackman and Kaiser Window which satisfies the stopband attenuation criteria.

Calculation of Blackman Window
Transition width, $\Delta F =8$ Hz
Sampling Frequency, $F_s = 500$ Hz
$\therefore$ Normalized transition width, $\Delta f= (\Delta F / F_s)$
$= (8 / 500)$
$= 0.016$
For, Blackman Window
$\Delta f= (5.5/M+1)$
$\Rightarrow M+1= (5.5 / \Delta f)$
$\Rightarrow M+1= (5.5 / 0.016)$
$\Rightarrow M+1=343.74 \cong 344$
So, the length of the Blackman Window is 344 and the order is, M=344-1=343

Calculation of Kaiser Window

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.

Now, δ will be the smallest value between the δp and δs.

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

We know,

$A = -20\log10\ (\delta s)$

$= -20\log10\ (0.001)$

$= 60$ dB

$\therefore$ The value of $\beta = 0.1102(A-8.7)$

$= 0.1102(60-8.7)$

$\therefore\ \beta = 5.65$

We know,

The length of, $M+1 \geq [\ 1+(A-8)\ /(2.285^*\ \Delta\omega)\ ]$

Here, $\Delta\omega = 2\pi^*\ \Delta f$

$= 2\pi^*0.016$

$= 0.1005$ radian/sample

$\therefore\ M+1 \geq [1+\ (A-8)\ /\ (2.285^*\ \Delta\omega)$

$\Rightarrow M+1 = [1+\ (60-8)\ /\ (2.285^*\ 0.1005)]$

$= [1+52\ /0.229]$

$= 231.44 \cong 232$

So, the length of the Kaiser Window is 232 and the order is, M=232-1=231

Since the order of the Blackman Window is higher than that of the Kaiser Window, we will choose the Kaiser Window.

Selection of Window and Reason

We choose the Kaiser Window based on the order of the window. The Kaiser Window has a lower order than the Blackman Window, so we select the Kaiser Window. The Blackman Window is much longer than the Kaiser Window, which would affect the filter's performance and characteristics. Therefore, we opt for the Kaiser Window to ensure better overall filter performance.

Filter implementation in MATLAB

By giving input of calculated data in filter designer cost and considering implementation using fixed precision algorithm

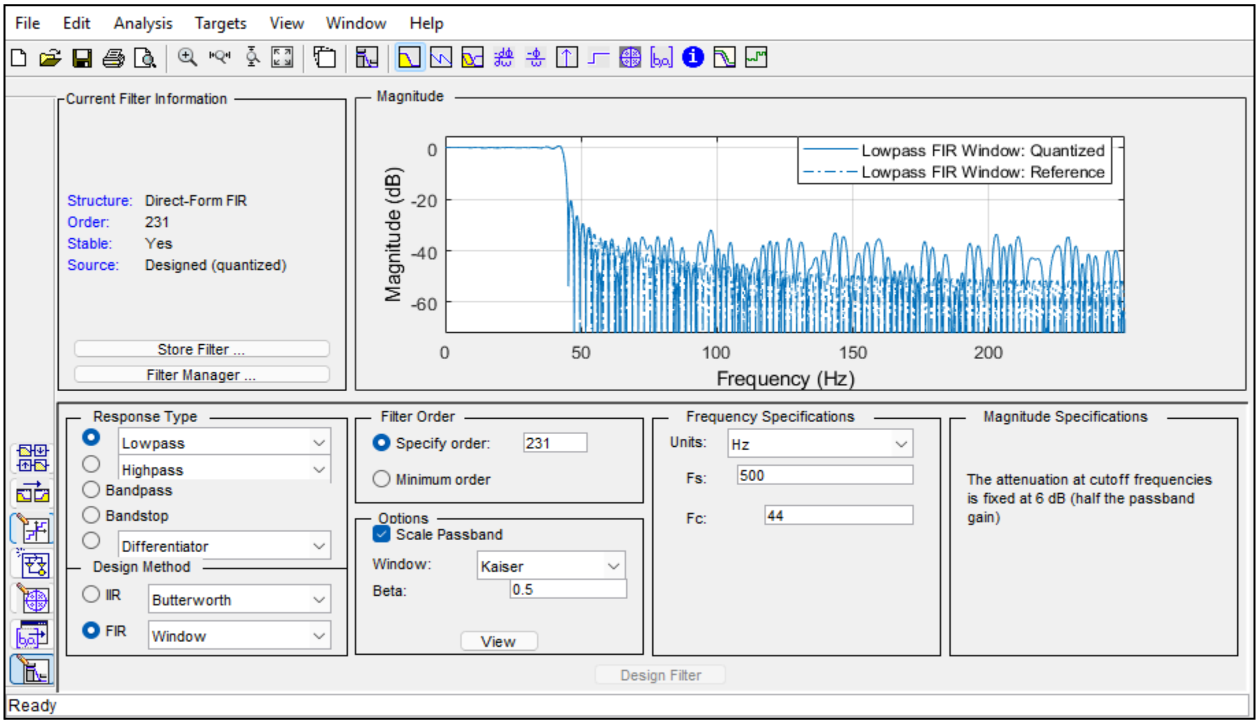

Figure-03: Magnitude Response (in dB) of Designed Filter

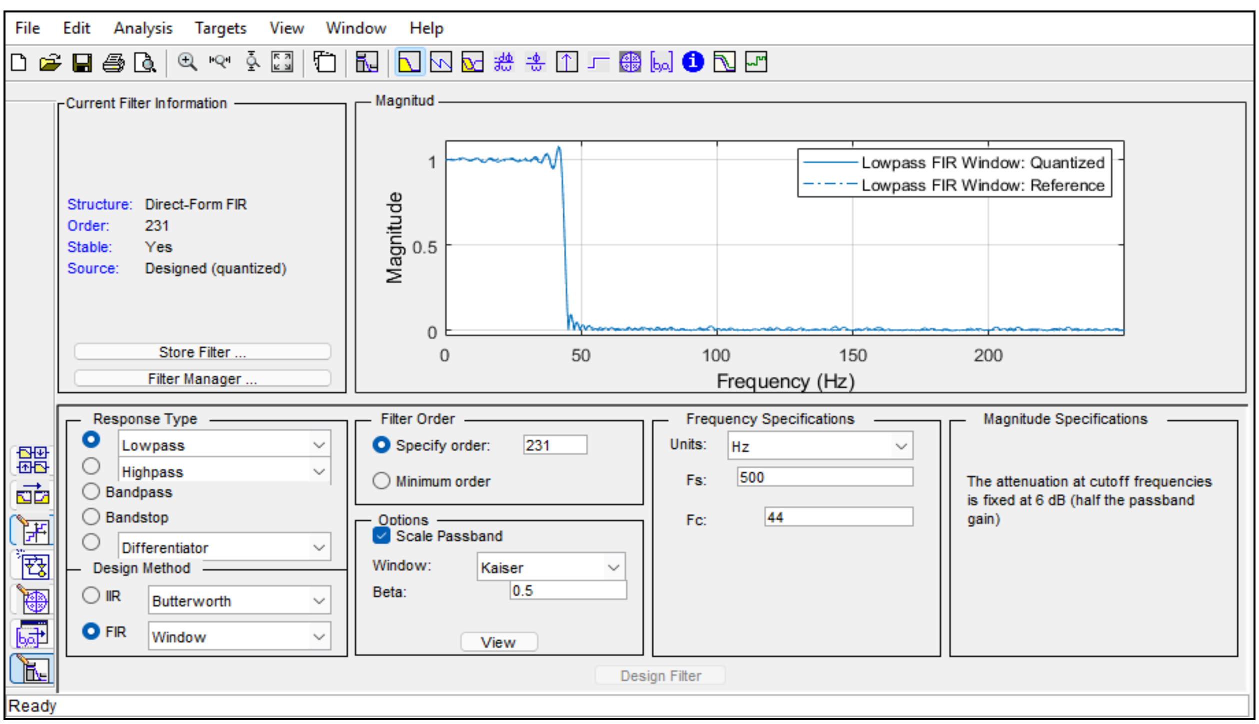

Figure-04: Magnitude Response of Designed Filter

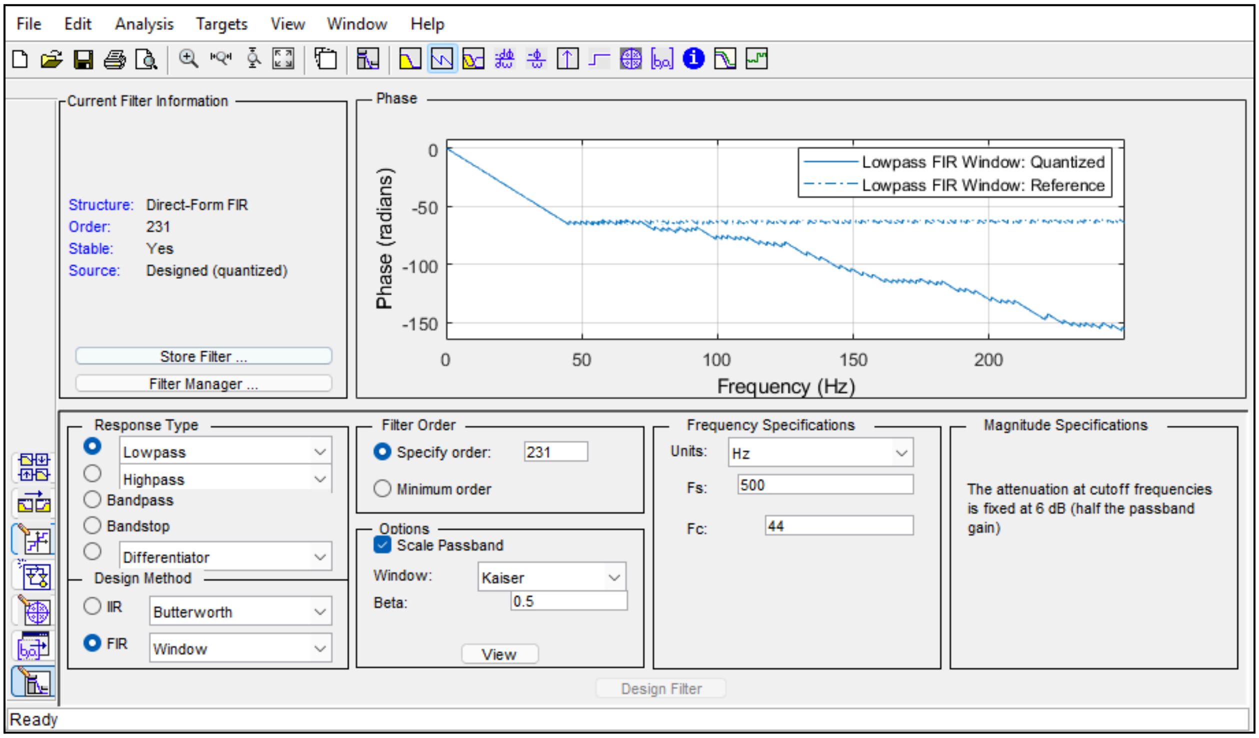

Figure-5: Phase Response of Designed Filter

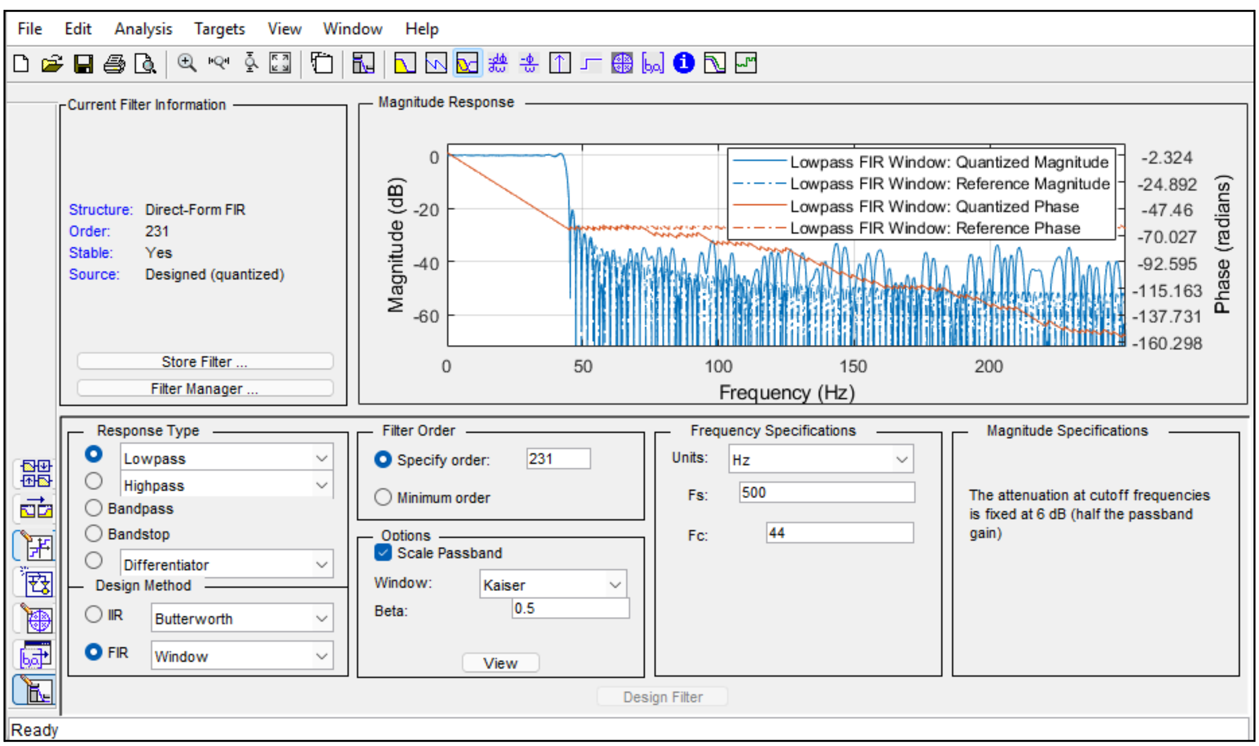

Figure-06: Magnitude & Phase Response of Designed Filter

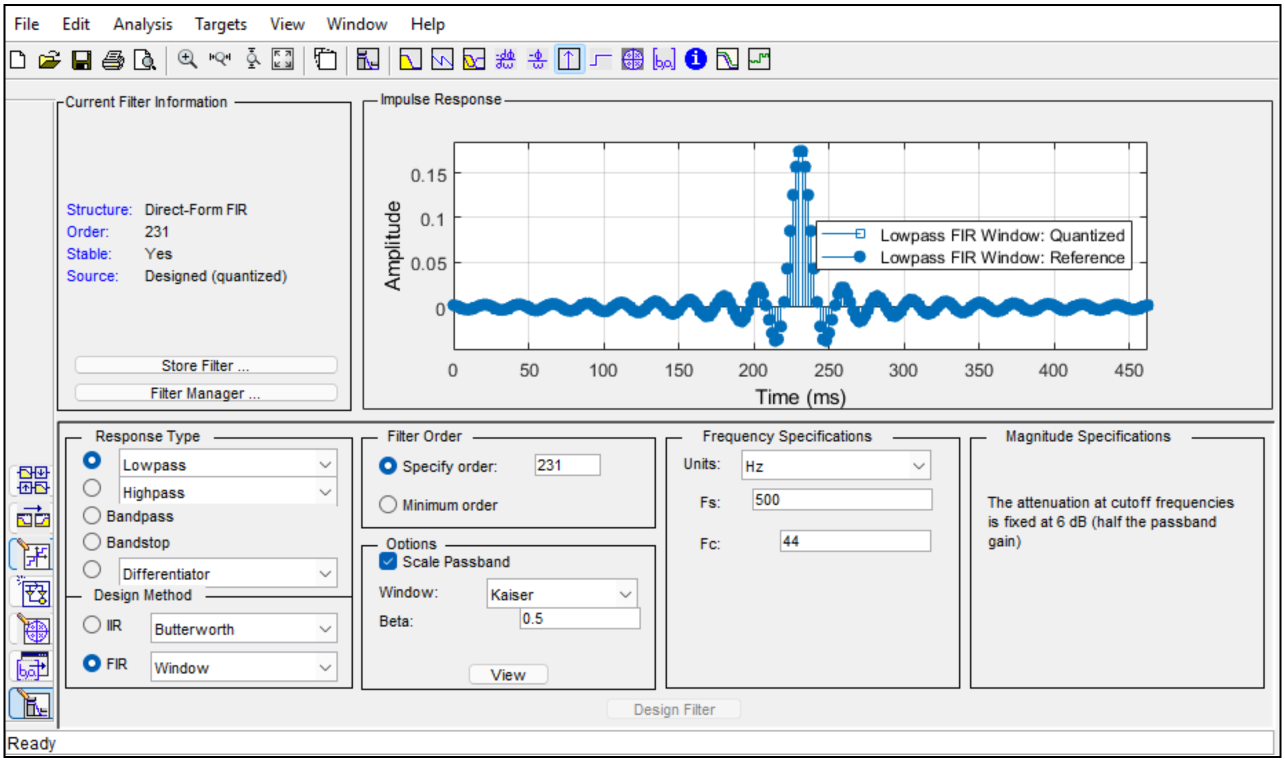

Figure-07: Impulse Response of Designed Filter

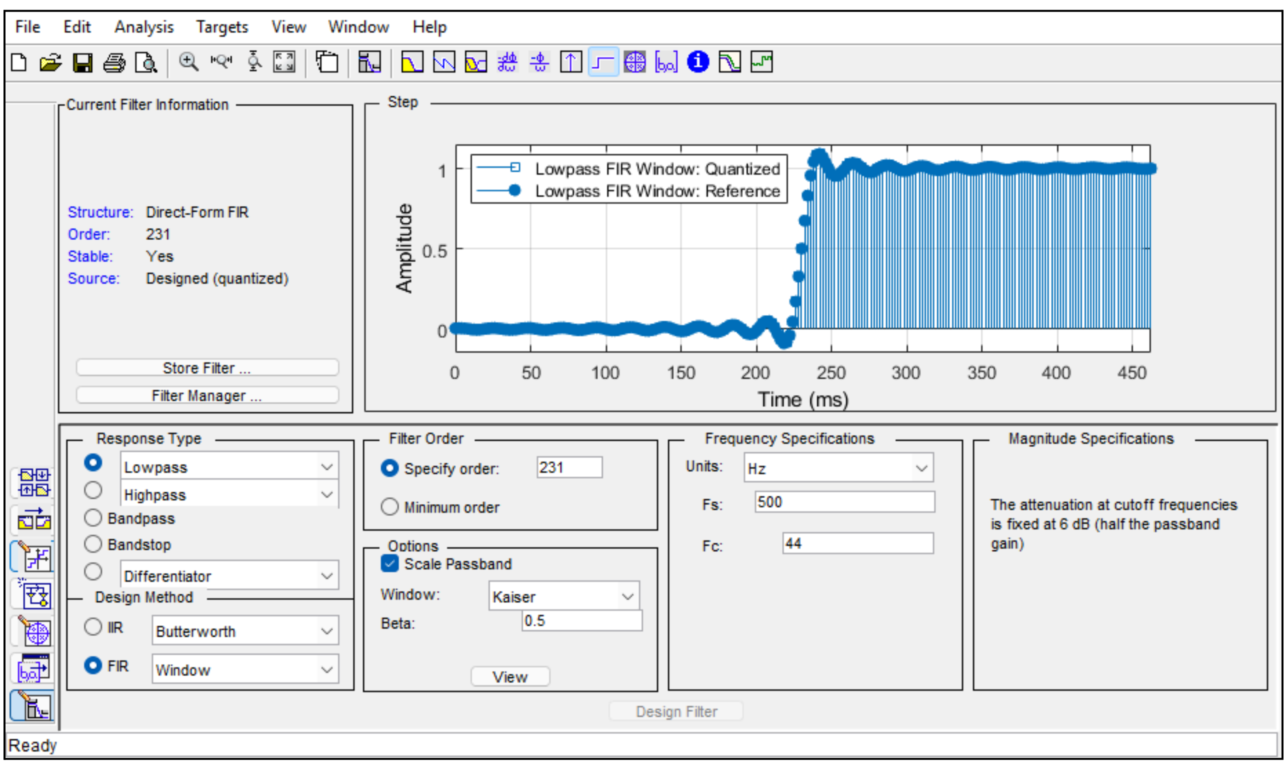

Figure-08: Step Response of Designed Filter

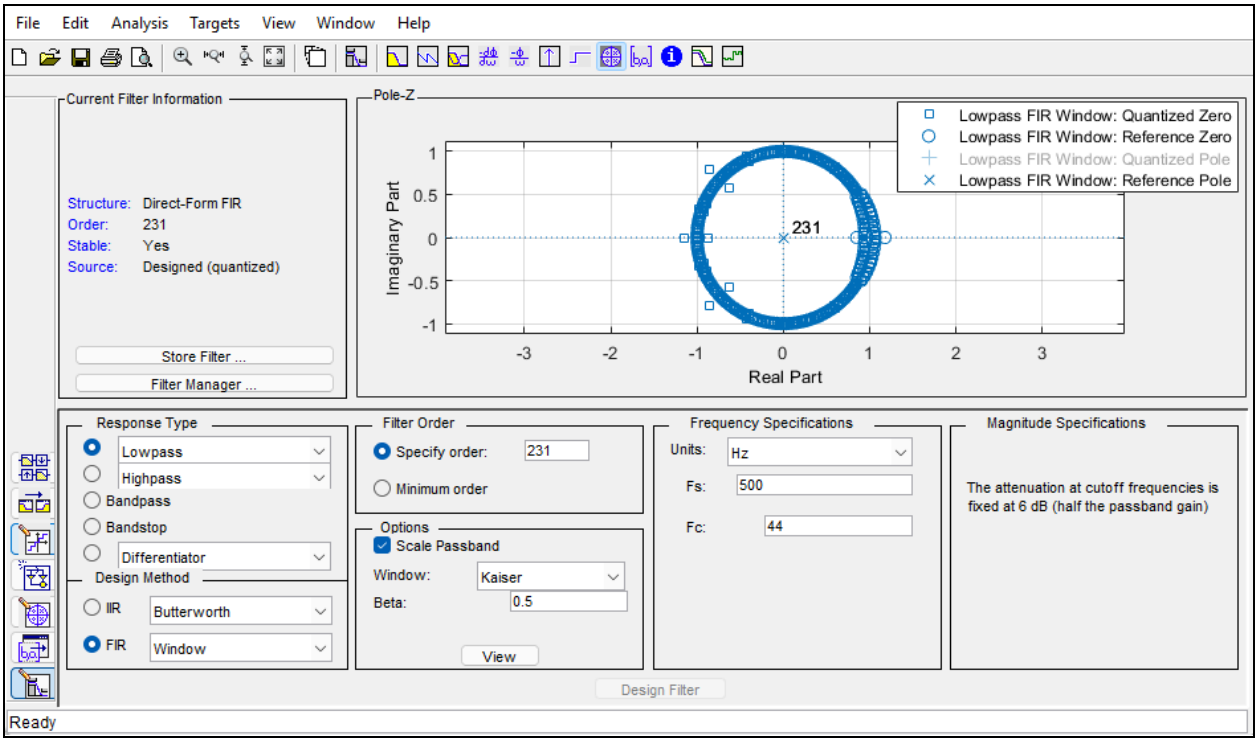

Figure-09: Poles & Zeros of Designed Filter

Designed Filter Specification

```
Discrete-Time FIR Filter (real)
-------------------------------
Filter Structure    : Direct-Form FIR
Filter Length       : 232
Stable              : Yes
Linear Phase        : Yes (Type 2)
Arithmetic          : fixed
Numerator           : s8,9 -> [-2.500000e-01 2.500000e-01)
Input               : s16,15 -> [-1 1)
Filter Internals    : Full Precision
  Output            : s27,24 -> [-4 4)  (auto determined)
  Product           : s23,24 -> [-2.500000e-01 2.500000e-01)  (auto determined)
  Accumulator       : s27,24 -> [-4 4)  (auto determined)
  Round Mode        : No rounding
  Overflow Mode     : No overflow

Implementation Cost
Number of Multipliers            : 232
Number of Adders                 : 231
Number of States                 : 231
Multiplications per Input Sample : 232
Additions per Input Sample       : 231
```

Figure-10: Specification of Designed Filter

Filter performance

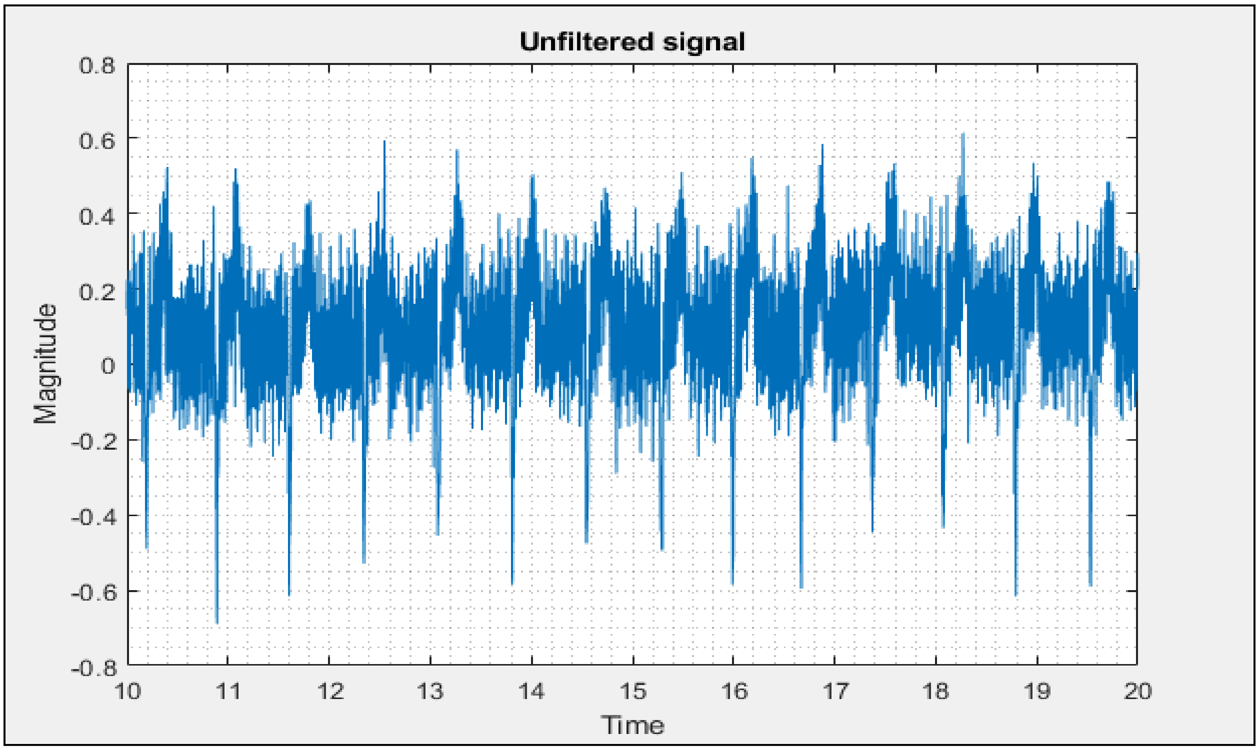

Figure-11: Noise corrupted ECG Signal in Time Domain

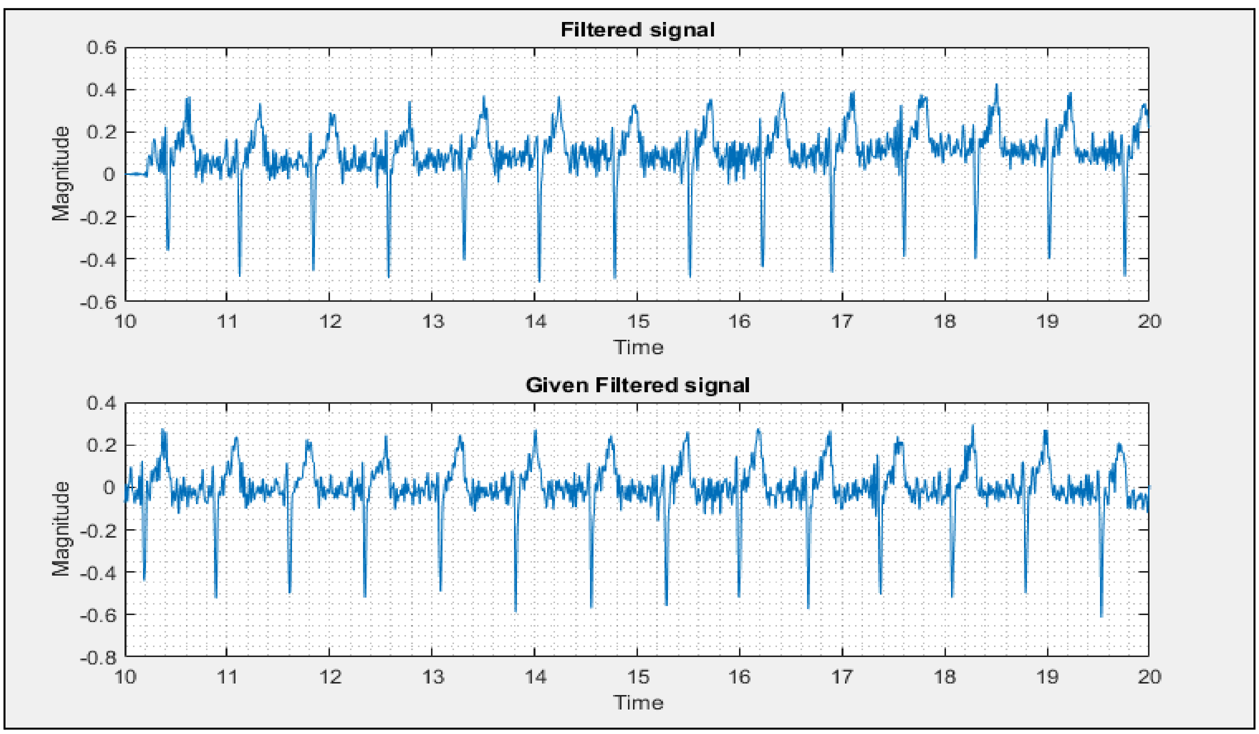

Figure-12: Filtered ECG Signal by designed filter and the result given by the instructor in Time Domain.

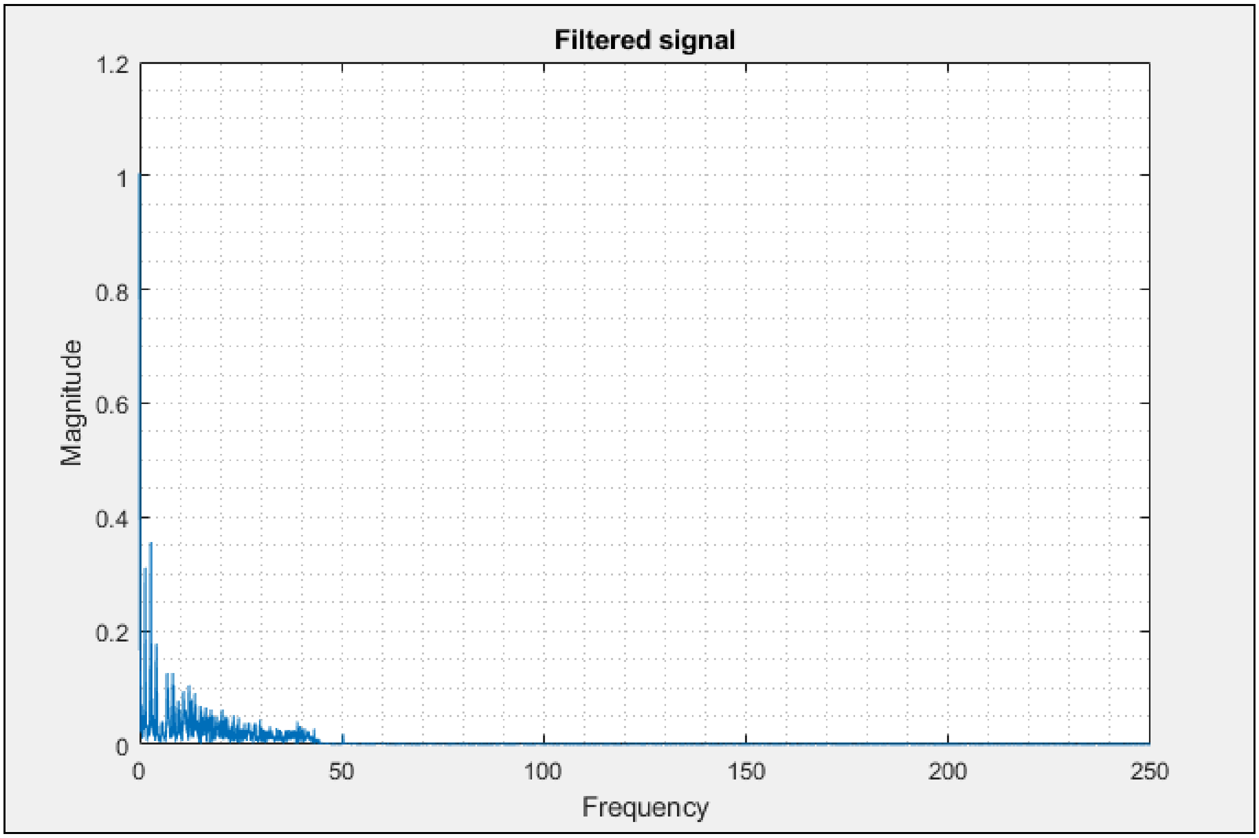

Figure-13: Frequency component analysis of Filtered ECG data

Improvement scopes

We have implemented this filter using a fixed-point algorithm instead of an infinite-precision method. As a result, the value of the filter coefficients was significantly reduced, which may lead to poor performance of the filter. In our custom filter creation, we focused on a particular time frame, resulting in a shorter filter. This not only reduced the need for memory, adders, multipliers, and computing time but also allowed for easier manipulation of data to alter the FIR filter specifications while preserving the fundamental concept of filter design.

Conclusion

This project was about building a special filter to clean up ECG readings. Power lines mess with these readings, but our filter gets rid of that noise without hurting the important ECG signal itself. The filter is cheap to make and works well with basic computers. It's also fast, so it can clean ECG data in real-time. This makes it perfect for putting in devices that doctors use to take ECGs.

Matlab code for filtering

```
clc
close all
% Load data from CSV file
ECG = load('Group_06.csv'); % load data
time = ECG(:,1); % time data
Ufilter = ECG(:,2); % unfiltered signal
% Plot the unfiltered signal
figure(1)
plot(time, Ufilter)
xlim([10, 20])
title('Unfiltered signal')
xlabel('Time')
ylabel('Magnitude')
grid minor
% Apply the exported filter (assuming it's named Hd)
Filter_signal = filter(Hd, Ufilter); % filtering data
% Compute frequency components of the filtered signal
fs = 500; % sampling frequency
frequency = Filter_signal / fs; % normalize by sampling frequency
frequency = double(frequency); % convert to double precision
L = length(frequency); % length of the signal
NFFT = 2^nextpow2(L); % next power of 2 for zero-padding
y = abs(fft(frequency, NFFT)); % compute FFT and take magnitude
frequency = fs / 2 * linspace(0, 1, NFFT / 2 + 1); % frequency axis
% Plot the filtered signal
figure(2)
subplot(2, 1, 1)
plot(time, Filter_signal)
xlim([10, 20])
title('Filtered signal')
xlabel('Time')
ylabel('Magnitude')
grid minor
% Plot the given filtered signal from the third column of the data
Actual = ECG(:,3);
subplot(2, 1, 2)
```

```
plot(time, Actual)
xlim([10, 20])
title('Given Filtered signal')
xlabel('Time')
ylabel('Magnitude')
grid minor
% Plot the frequency spectrum of the filtered signal
figure(3)
plot(frequency, y(1:length(frequency)));
xlabel('Frequency')
ylabel('Magnitude')
title('Filtered signal')
grid minor
```

Matlab code for data analysis

```
clc
close all
clear
ECG=load('Group_06.csv');
fs=500;
time=ECG(:,1);
signal=ECG(:,2);
frequency=signal/fs;
L=length(frequency);
NEFT=2^nextpow2(L);
y=abs(fft(frequency,NEFT));
frequency=fs/2*linspace(0,1,NEFT/2+1);
figure(1)
plot(time,signal)
title('Unfilter signal')
xlabel('Time')
ylabel('Magnitude')
grid minor
figure(2)
plot(frequency,y(1:length(frequency)));
xlim([0,60])
title('Frequency domain of Unfiltered signal')
xlabel('Frequency')
ylabel('Magnitude')
grid minor
```