Lab 3 Part 1: FIR Filter Design

Haohan Zhu

1. Filter Design Exercise

1.1 Design a minimum order lowpass FIR filter

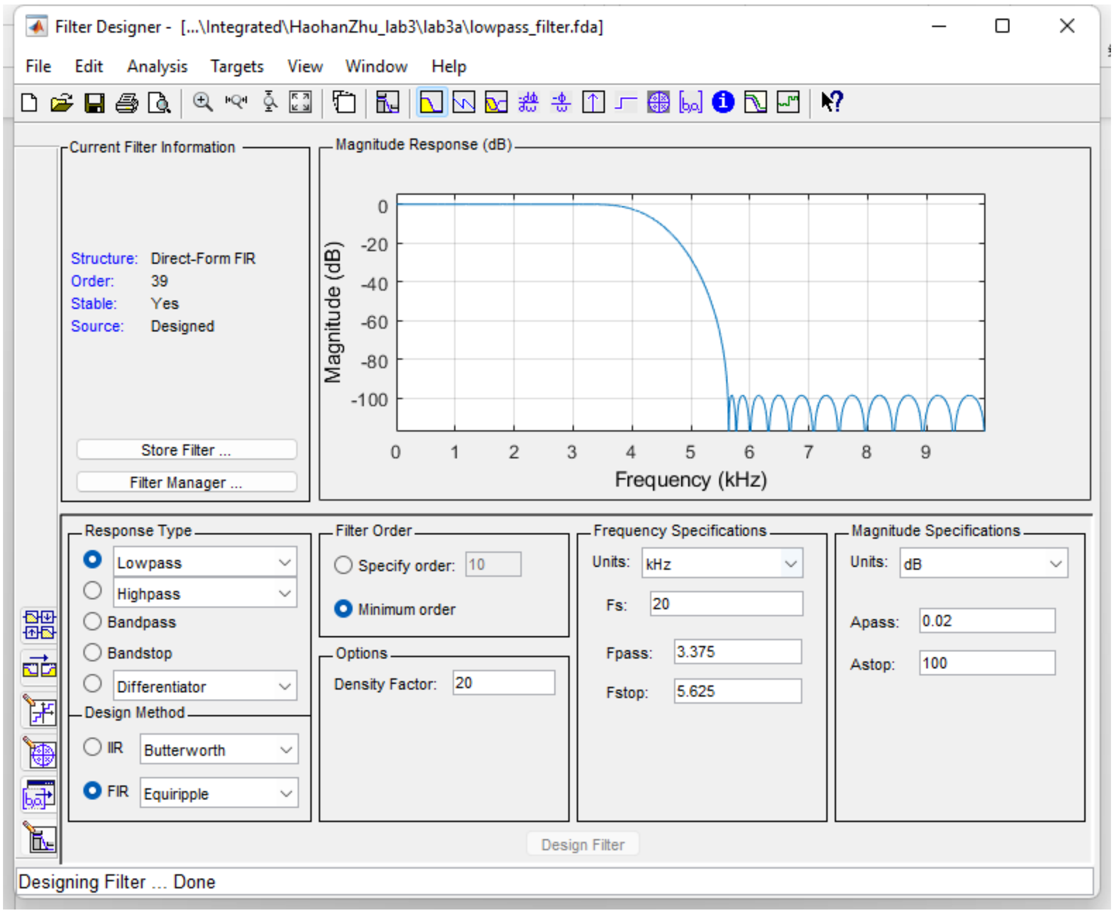


Figure 1. low pass filter design

1.2 Lab Report Questions

1.2.1 What is the order of the filter you designed? What does this mean?

The rate at which the filter response falls into the transition band is determined by the order of the filter. The higher the order of the filter, the faster the roll-off rate. The order of the filter is given as an integer value and is derived from the filter's transfer function. The order of the filter also indicates the minimum number of reactance components required for the filter. For example, a third-order filter requires at least three reactance components: one capacitor and two inductors, two capacitors and one inductor, or in the case of an active filter, three capacitors. Higher order can process more data.

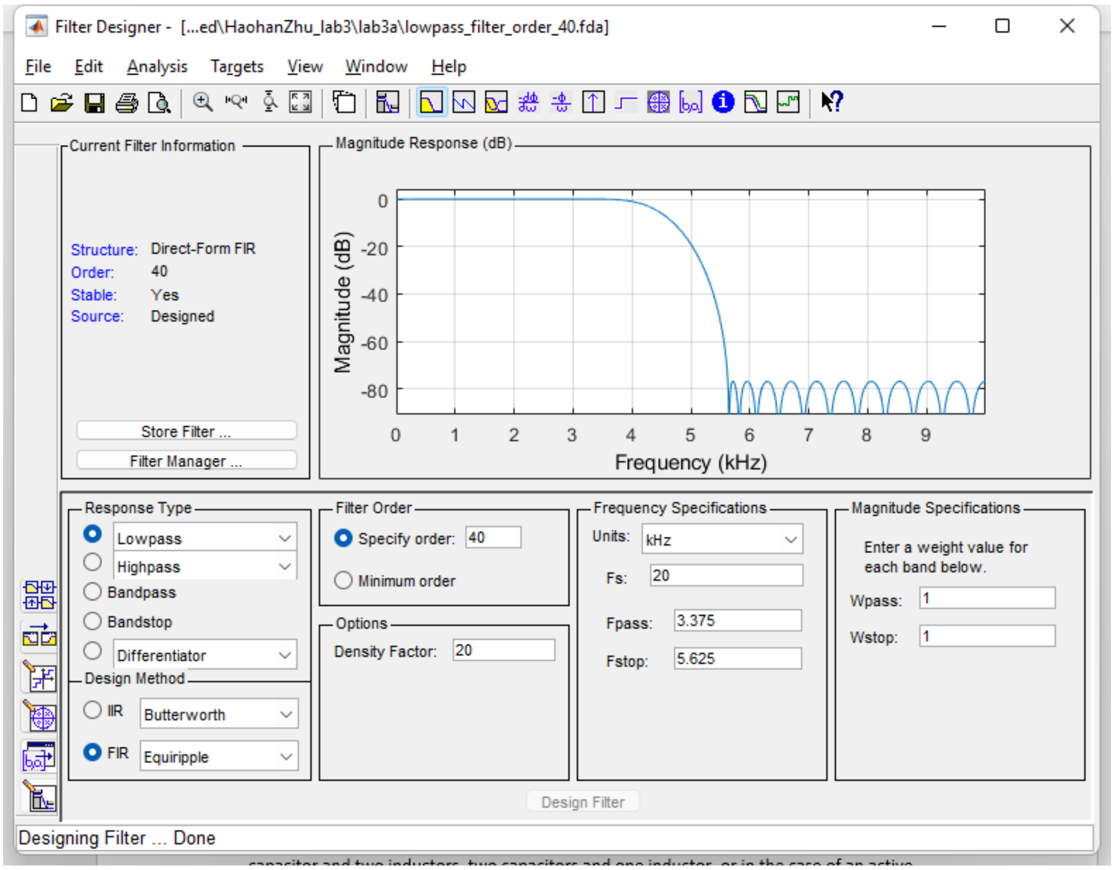


Figure 2. low pass filter with order 40

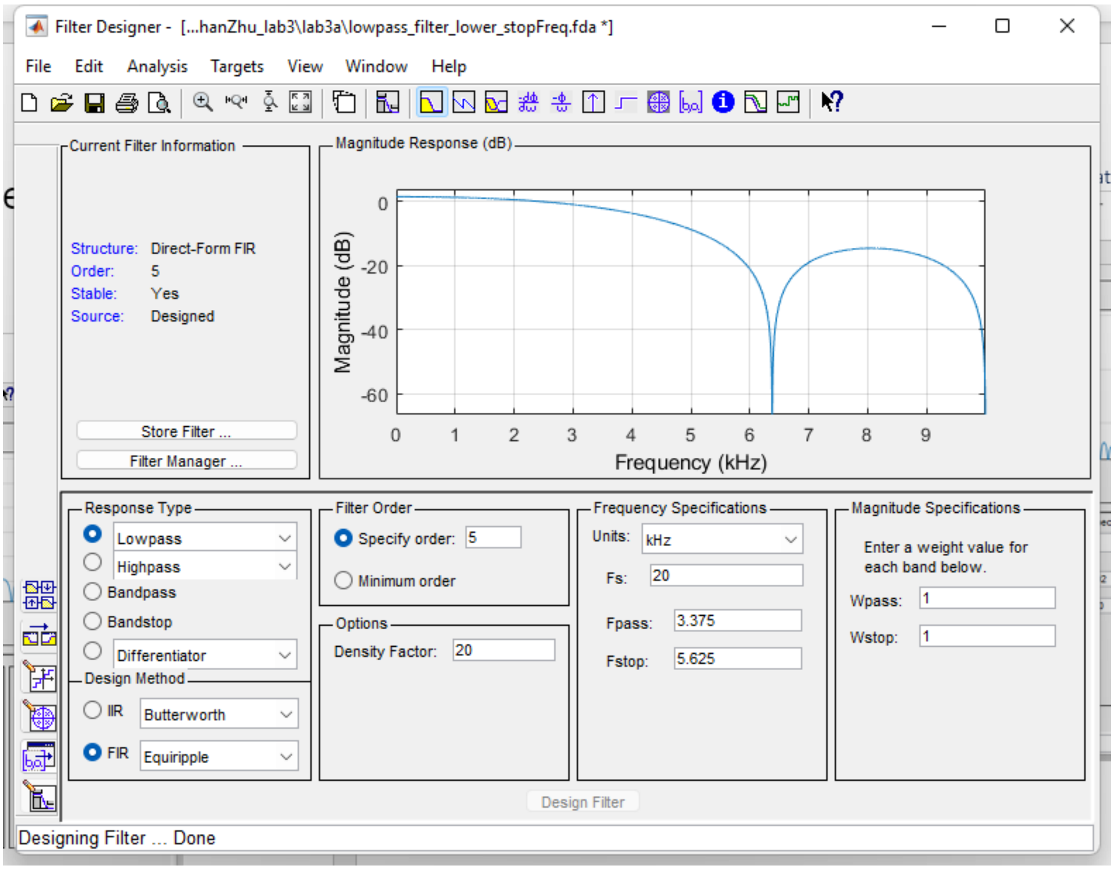


Figure 3. low pass filter with order 5

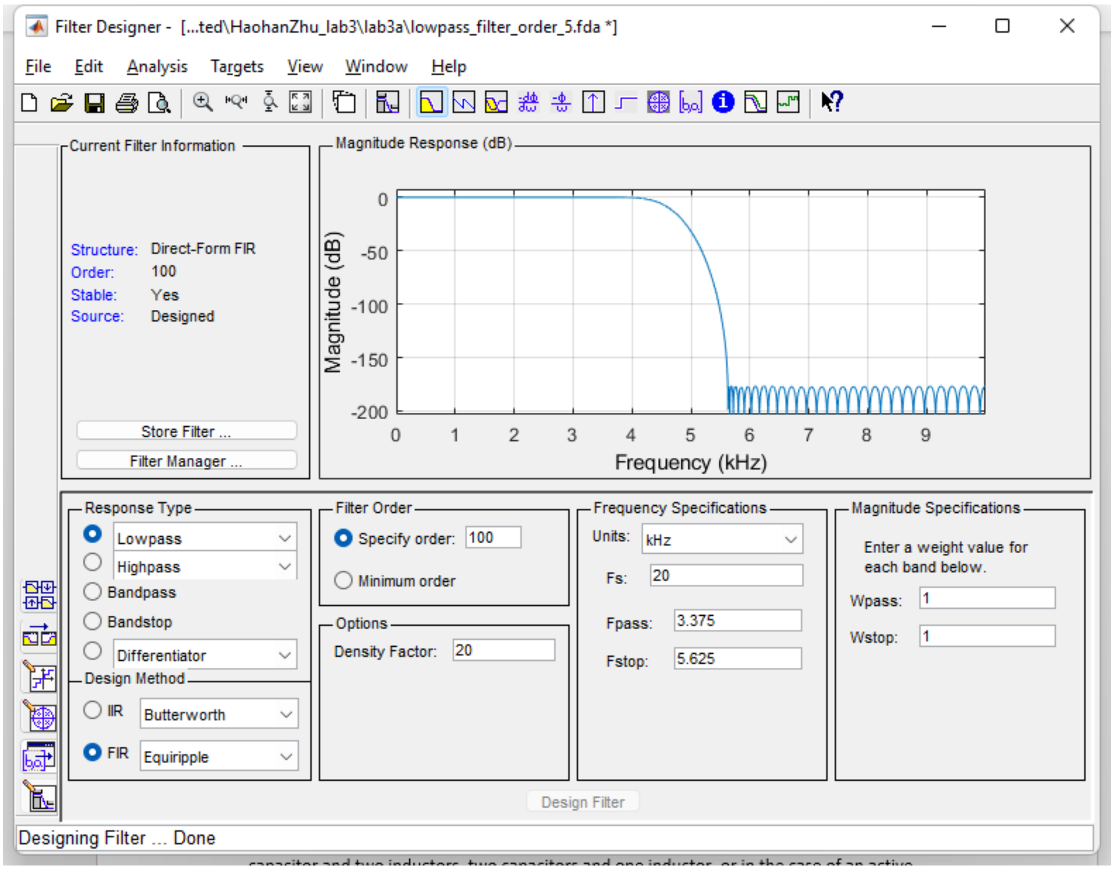


Figure 4. low pass filter with order 100

1.2.2 What are the effects of altering the passband and stopband attenuation?

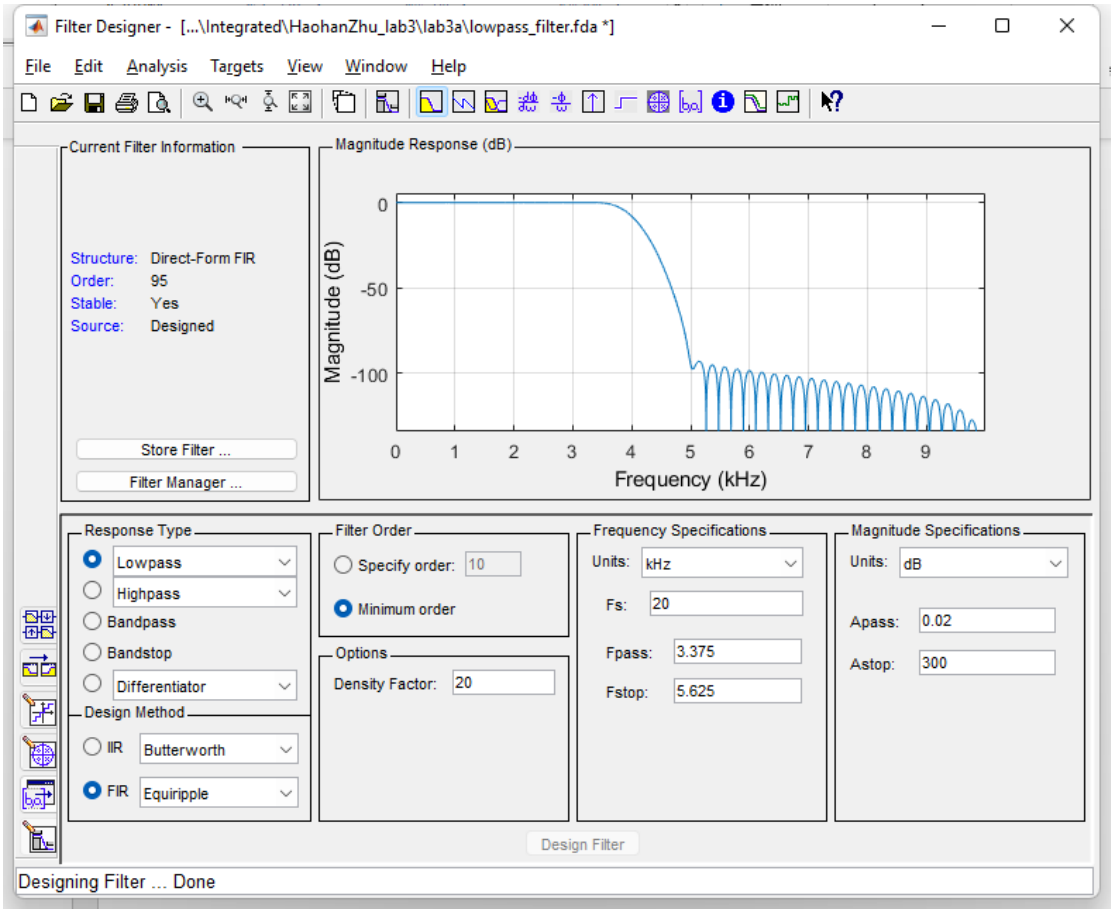


Figure 5. low pass filter with larger attenuation

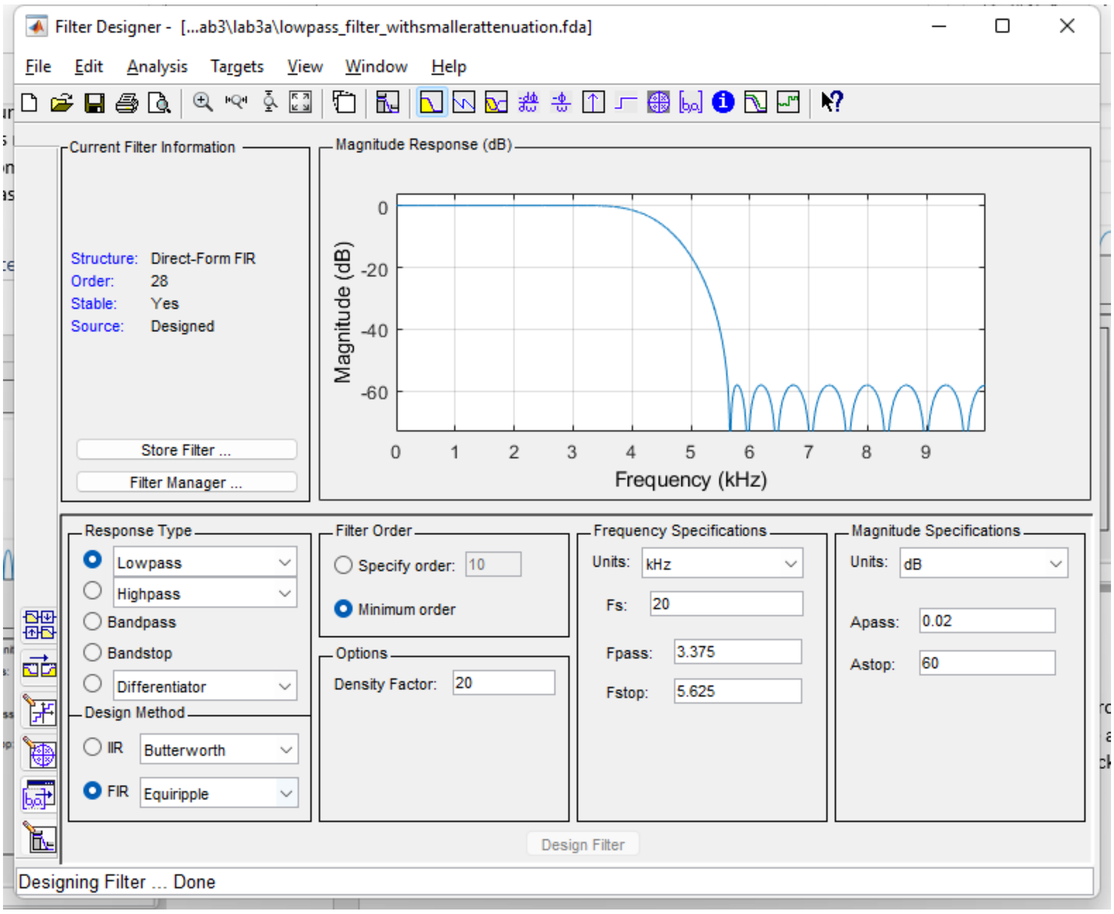


Figure 6. low pass filter with smaller attenuation

The passband ripple is the difference between the maximum and minimum amplitude of the passband in the frequency response of the filter. The passband ripple is related to the order of the filter, the higher the order the lower the ripple. In general, changing the attenuation will affect the amplitude of the filter and makes the frequency change.

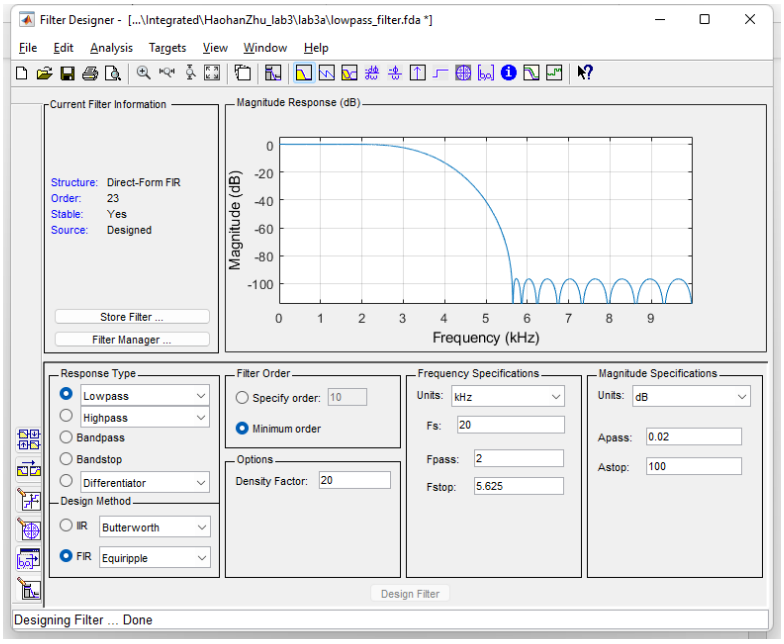


Figure 7. low pass filter design with Passband frequency = 2kHz

Stopband attenuation: in the passband, there is part of the signal through, part of the signal blocked, and the blocked part cannot be all blocked, only part of the attenuation, part of the stay, the minimum attenuation describes the ability to block the blocked signal, the greater the attenuation, the better the ability.

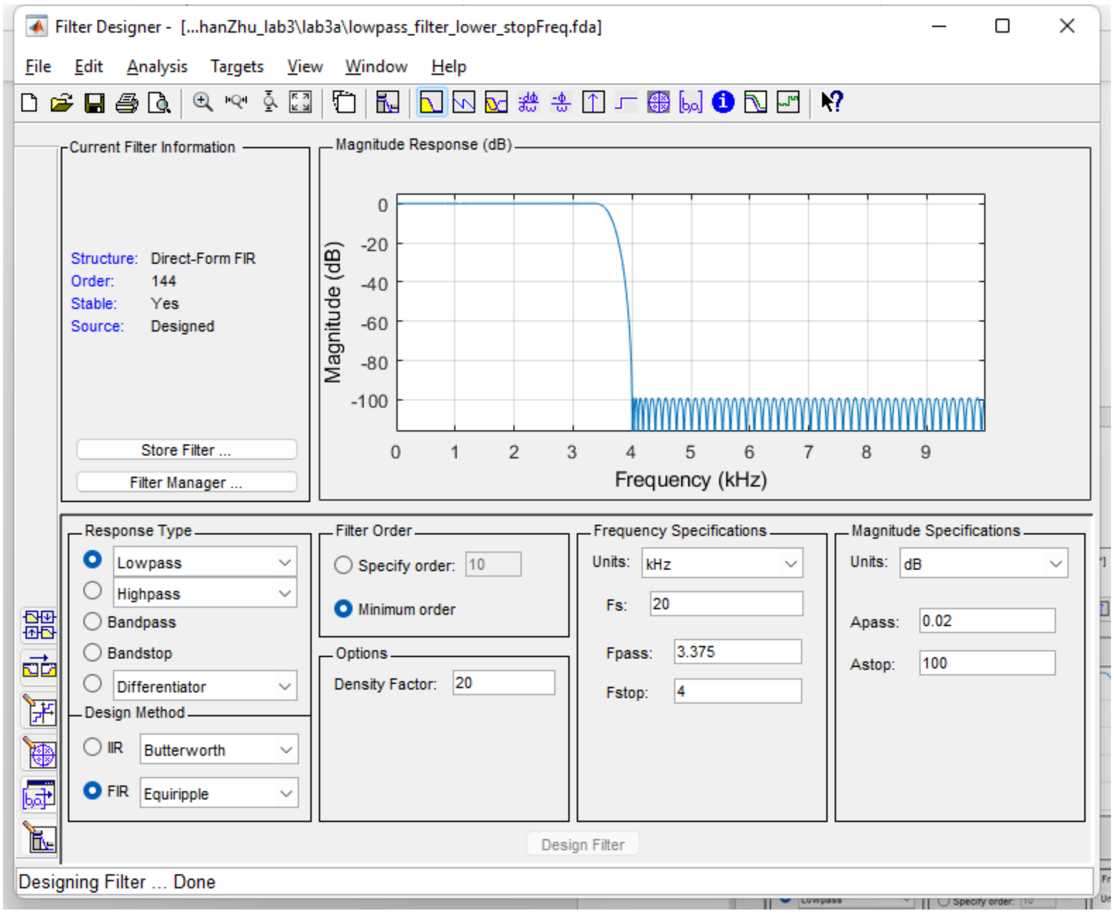


Figure 8. low pass filter design with stopband frequency = 4kHz

2. Filtering a Noisy Audio File

2.1 Identify the Noise

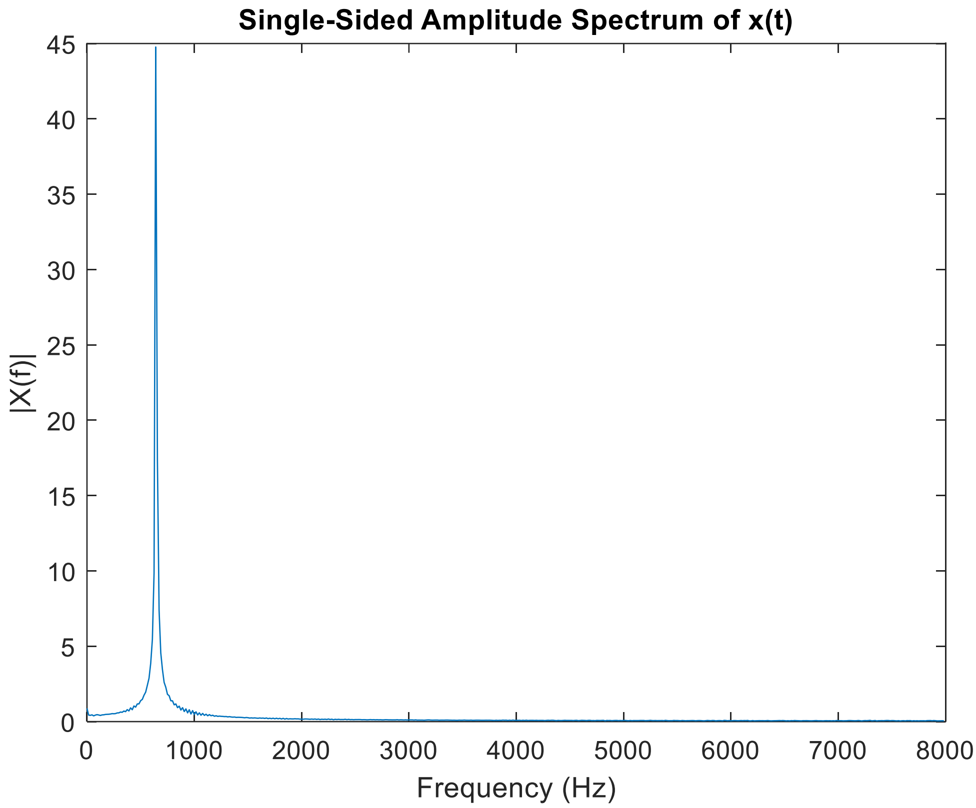


Figure 8. Noise

2.1.1 Code

```
%=====================================================
% fine frequency part work area begin
%=====================================================
% I is index which point to the maximum position
[I] = argmax(fvec, fresp);
f_max = max(fresp);
% fstep is interval
% fvec are the fft samples duration
frequency_I = fstep * I;

disp(f_max);
disp(I)
disp(frequency_I)

% F_max = fresp(I)
% fmax = max(fresp)
% [fmax, I] = max(fresp);
%=====================================================
% fine frequency part work area over
%=====================================================


%=====================================================
% fine frequency part work area begin
%=====================================================
```

```
function [index] = argmax(fvec, fresp)
    % initial index
    index = 0;
    % find maximum in the fresquency
    fmax = max(fresp);
    % when detection meet the maximum
    % there is the index of noise
    for i = 1 : length(fvec)
        f_index = fresp(i);
        if f_index == fmax
            index = i;
        end
    end
end
%======================================================
% fine frequency part work area begin
%======================================================
```

2.2 Design FIR

Based on the previously generated spectrum about the noise, I wanted to get clean audio after the noise. So I used the high pass filter to filter the audio signal. At the beginning I design the filter, I just noticed the main lobe, and the result always had noise. Then I recalled the DSP knowledge and found the side lobe also had many corrupted signal. That is way that the Passband is bigger than the maximum noise position (in my case the max noise is in 656.2500Hz).

2.2.1 Code

```
function Hd = filtershow
    %FILTERSHOW Returns a discrete-time filter object.

    % MATLAB Code
    % Generated by MATLAB(R) 9.11 and DSP System Toolbox 9.13.
    % Generated on: 02-Nov-2022 20:34:49

    % Equiripple Highpass filter designed using the FIRPM function.

    % All frequency values are in Hz.
    Fs = 16000;  % Sampling Frequency

    Fstop = 100;               % Stopband Frequency
    Fpass = 2500;            % Passband Frequency
    Dstop = 0.0001;          % Stopband Attenuation
    Dpass = 0.057501127785;  % Passband Ripple
    dens  = 20;              % Density Factor

    % Calculate the order from the parameters using FIRPMORD.
    [N, Fo, Ao, W] = firpmord([Fstop, Fpass]/(Fs/2), [0 1], [Dstop,
Dpass]);

    % Calculate the coefficients using the FIRPM function.
    b  = firpm(N, Fo, Ao, W, {dens});
    Hd = dfilt.dffir(b);

    % [EOF]
End
```

3. Quantise the FIR Filter

3.1 Compare the magnitude response for full precision and different levels of coefficient quantisation.

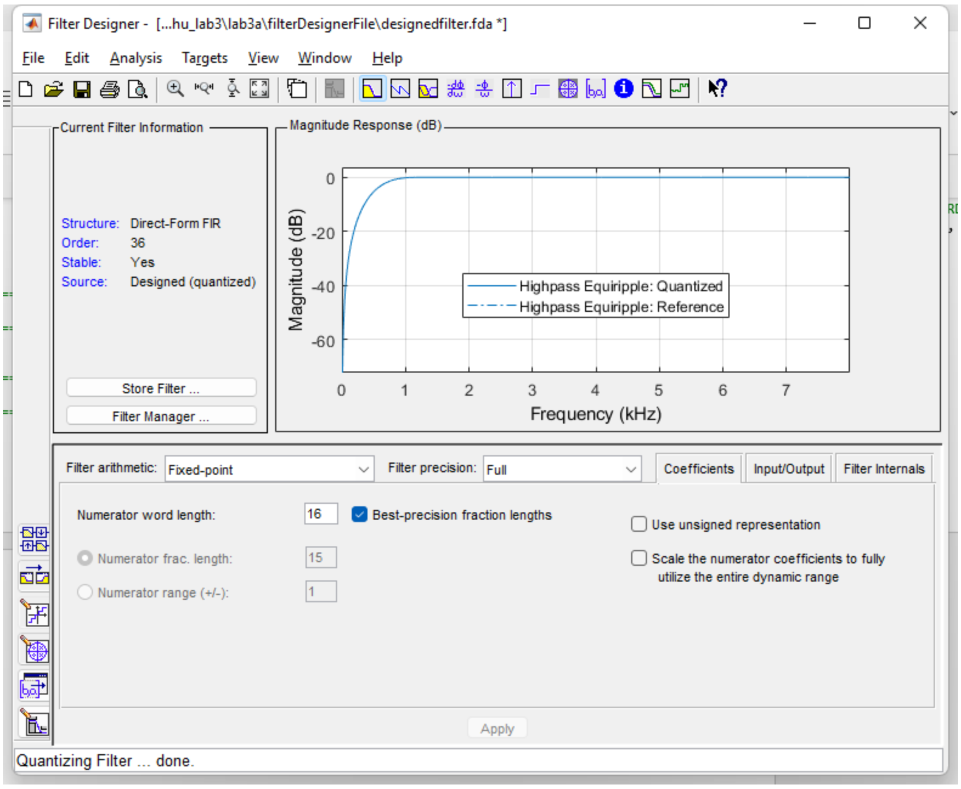


Figure 9. Appropriately quantised

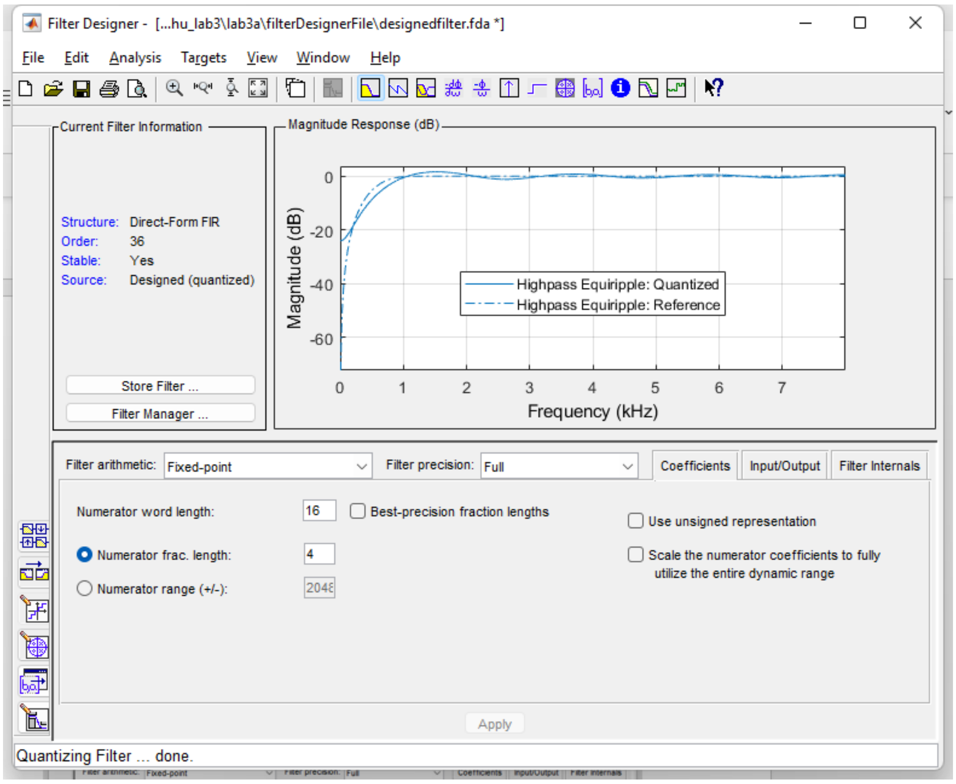


Figure 10. Under-quantised

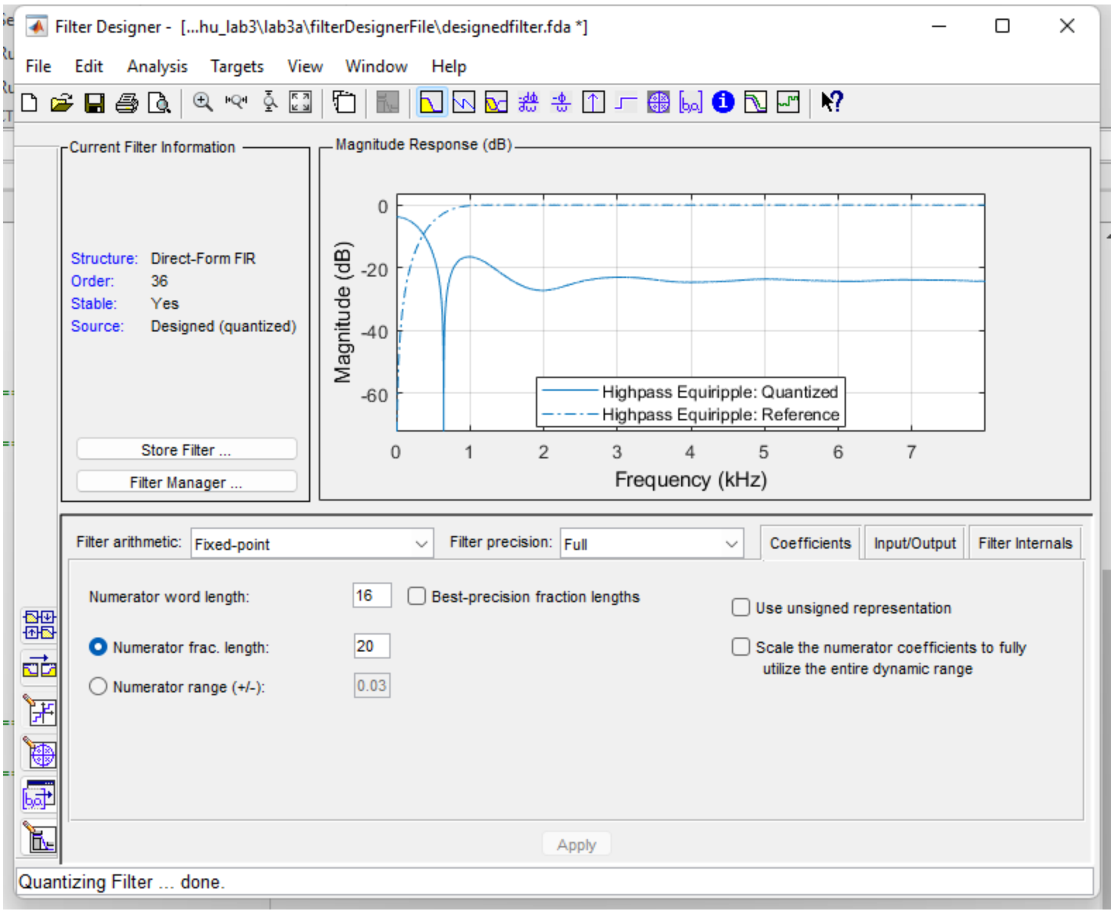
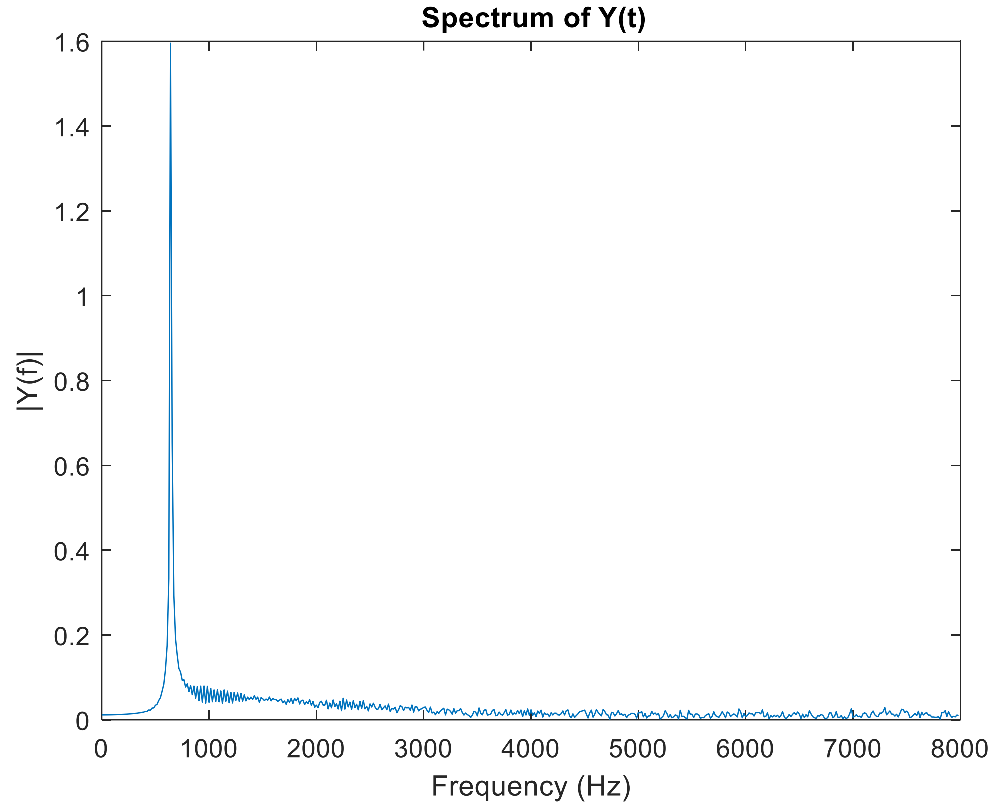
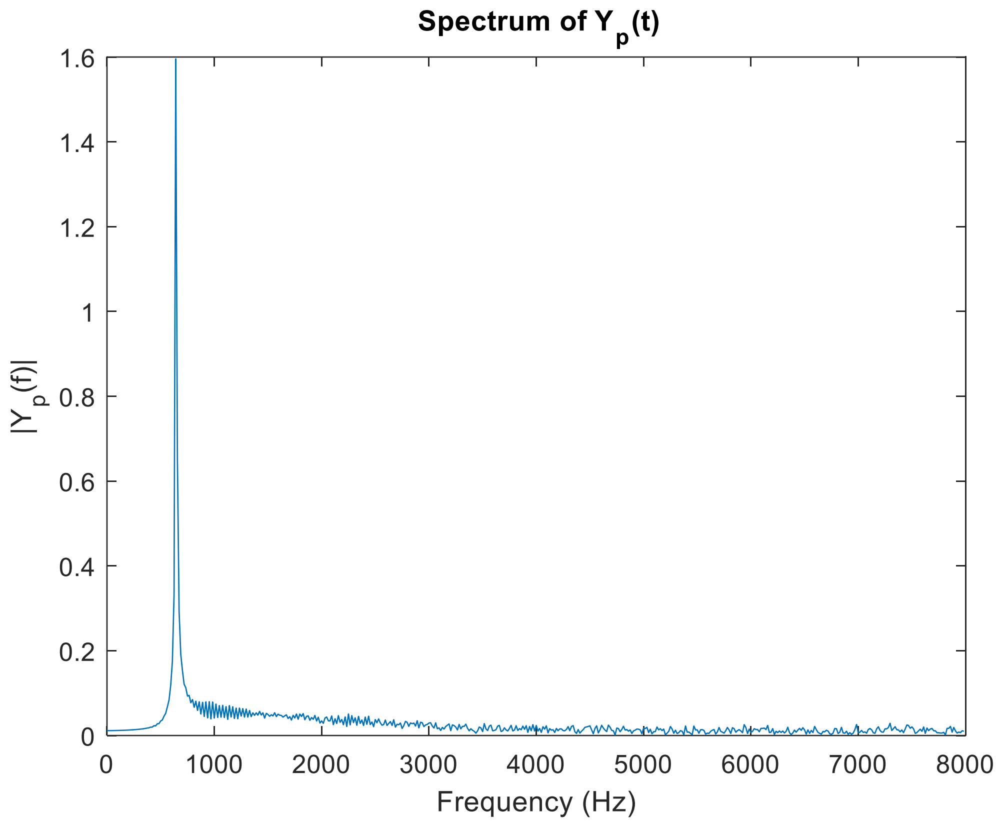


Figure 11. Over-quantised

4. Result



Spectrum of Y(t)
|Y(f)|
Frequency (Hz)



Spectrum of Y_p(t)
|Y_p(f)|
Frequency (Hz)