Elektronica Assignment 2

Anoop K

April 2025

Task: Designing a digital filter using MATLAB's filterDesigner tool and applying it to filter a mixture of sine waves.

1 Designing the filter

I designed a low-pass filter using the MATLAB filter designer with the following parameters:

- Filter order: Minimum order
- Fs (sampling frequency): 48000 Hz
- Fpass (cut-off frequency of the low-pass filter): 9600 Hz

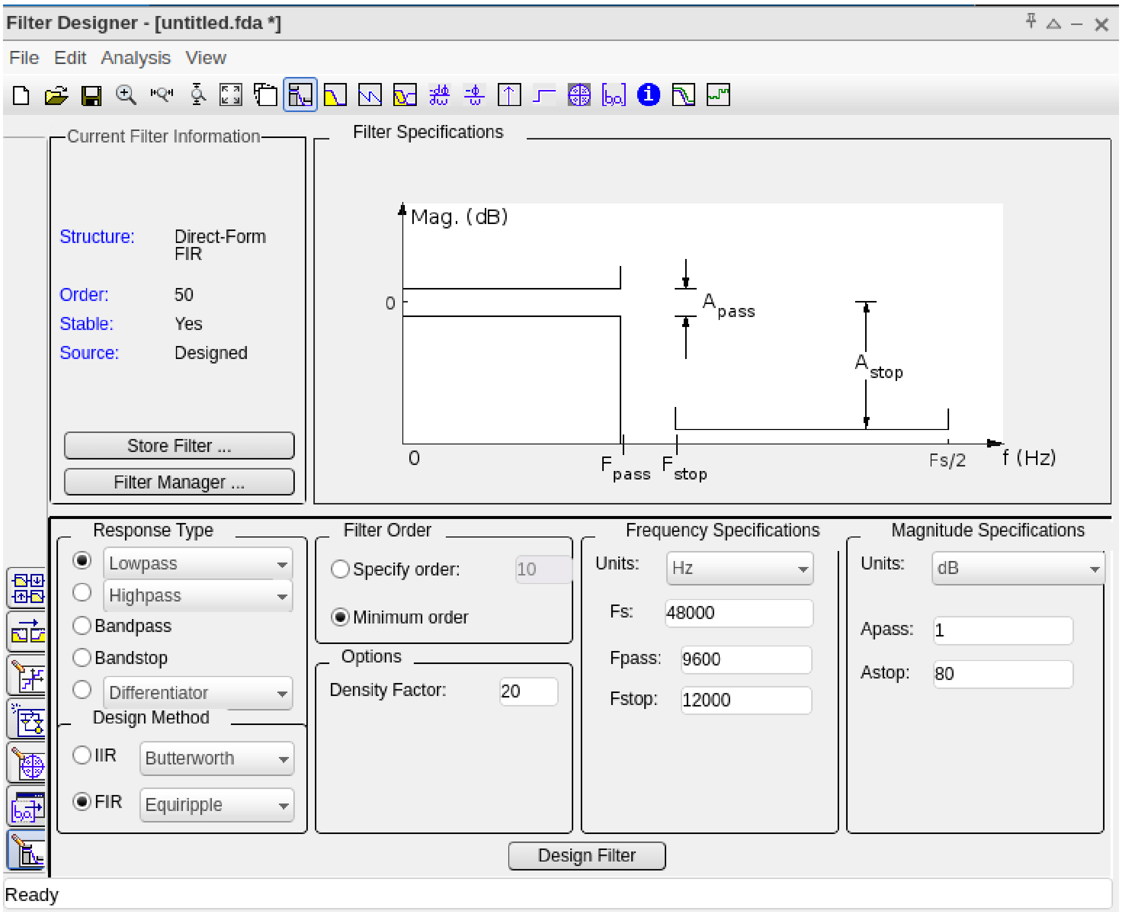

Figure 1: Screenshot of the parameters taken during filter design

After confirming the filter response plot, I exported the filter coefficients as .mat in file.

2 Generating a signal consisting of 10 sine waves of different frequencies and taking their FFT

Selecting frequencies 1 Hz, 10 Hz, 100 Hz, 500 Hz, 1000 Hz, 2500 Hz, 5900 Hz, 7400 Hz, 10000 Hz, 15000 Hz.

Source code:

```
Fs = 48*10^3; % Sampling frequency
t = 0:1/Fs:1; % Time vector

function output= analyse_t(t, x)
    plot(t, abs(x));
    xlim([0, 0.003]);
    grid on;
end

function output= analyse_f(Fs,X)
    N = length(X);
    f = (0:N/2-1)*(Fs/N);
    plot(f, abs(X(1:floor(N/2))));
    grid on;
end

% Defining 10 sinusoidal signals of different frequencies
x = sin(2*pi*t) + sin(2*pi*10*t) + sin(2*pi*10^2*t) + sin(2*pi*500*t)
    + sin(2*pi*10^3*t) + sin(2*pi*2500*t) + sin(2*pi*7400*t) + sin(2*pi*5900*t)
    + sin(2*pi*10^4*t) + sin(2*pi*1.5*10^4*t);

%Computing FFT
X = fft(x);

analyse_t(t, x);
analyse_f(Fs, X);
```

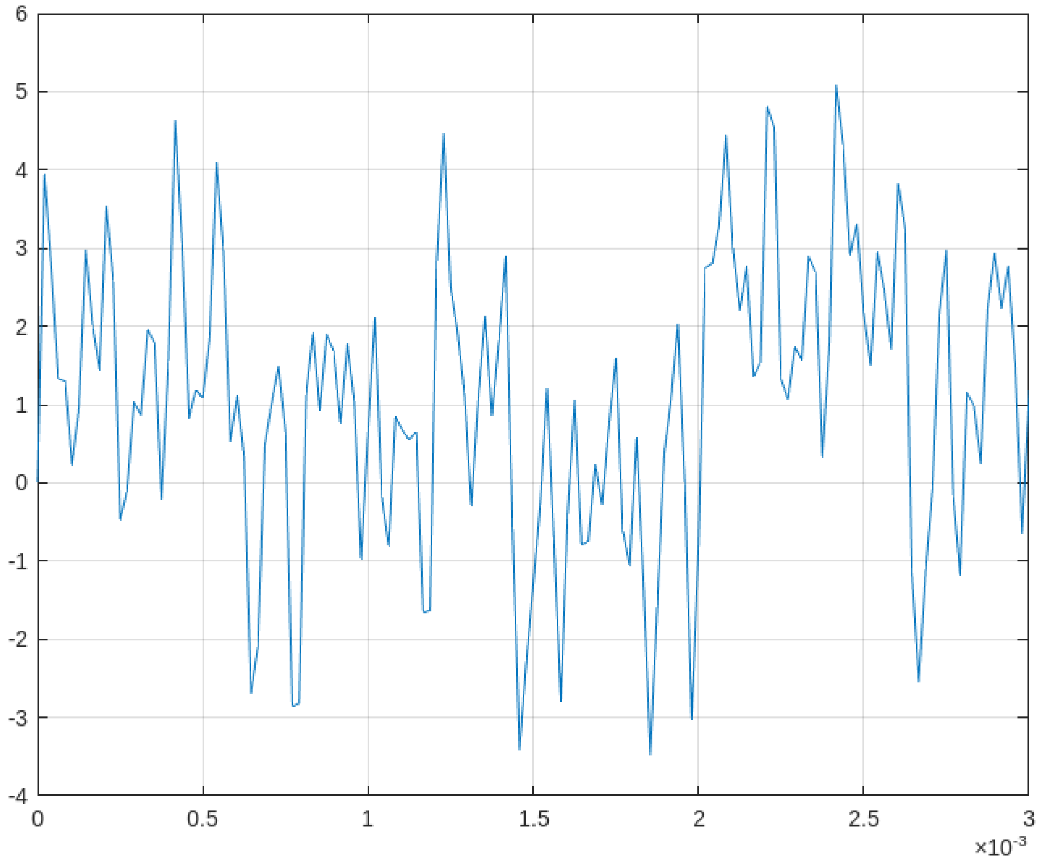

Figure 2: Signal $x(t)$ in $[0, 0.003]$

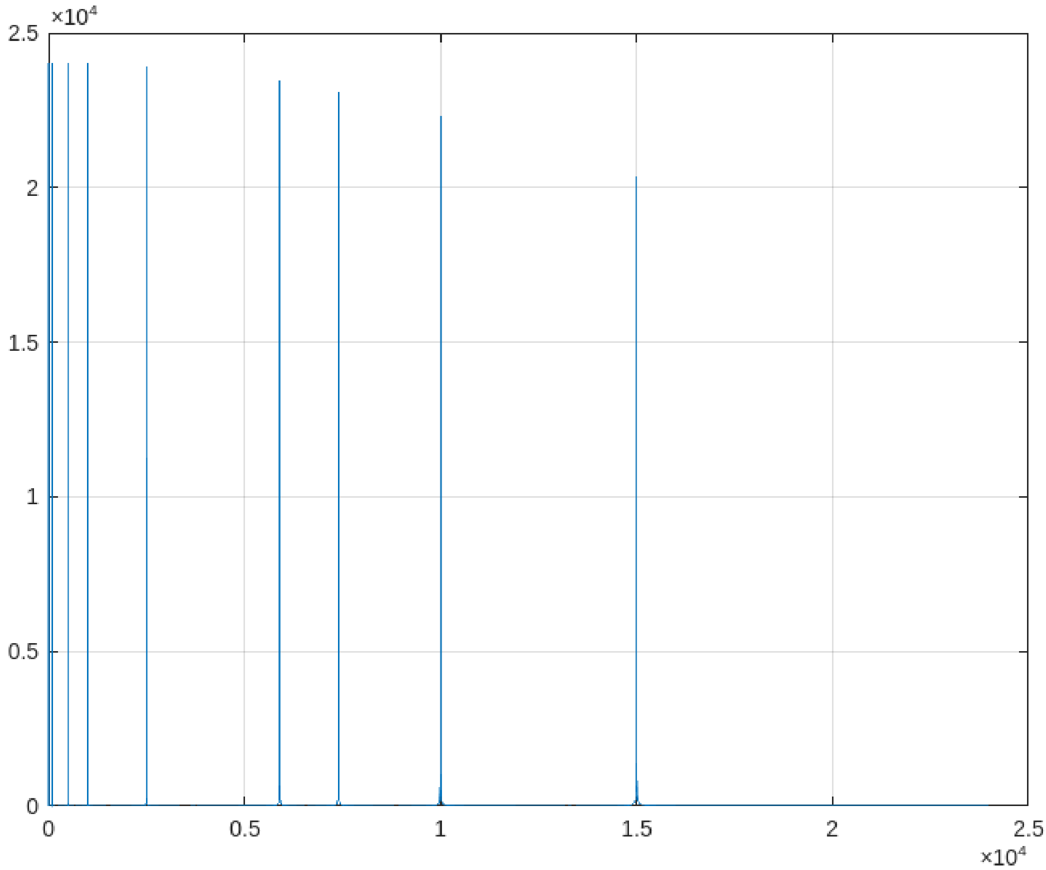

Figure 3: FFT of signal $x(t)$

Note: I am plotting the signal $x(t)$ from 0 to 0.003 for clear curves

3 Applying the filter to the signal and taking their FFT

Source code:

```
%Loading the filter
load('filter.mat');

%Filtering the generated signal using filter coefficients
y = filter(Num, 1, x);

%Computing the FFT of the filtered signal.
Y = fft(y);

analyse_t(t, y);
analyse_f(Fs, Y);
```

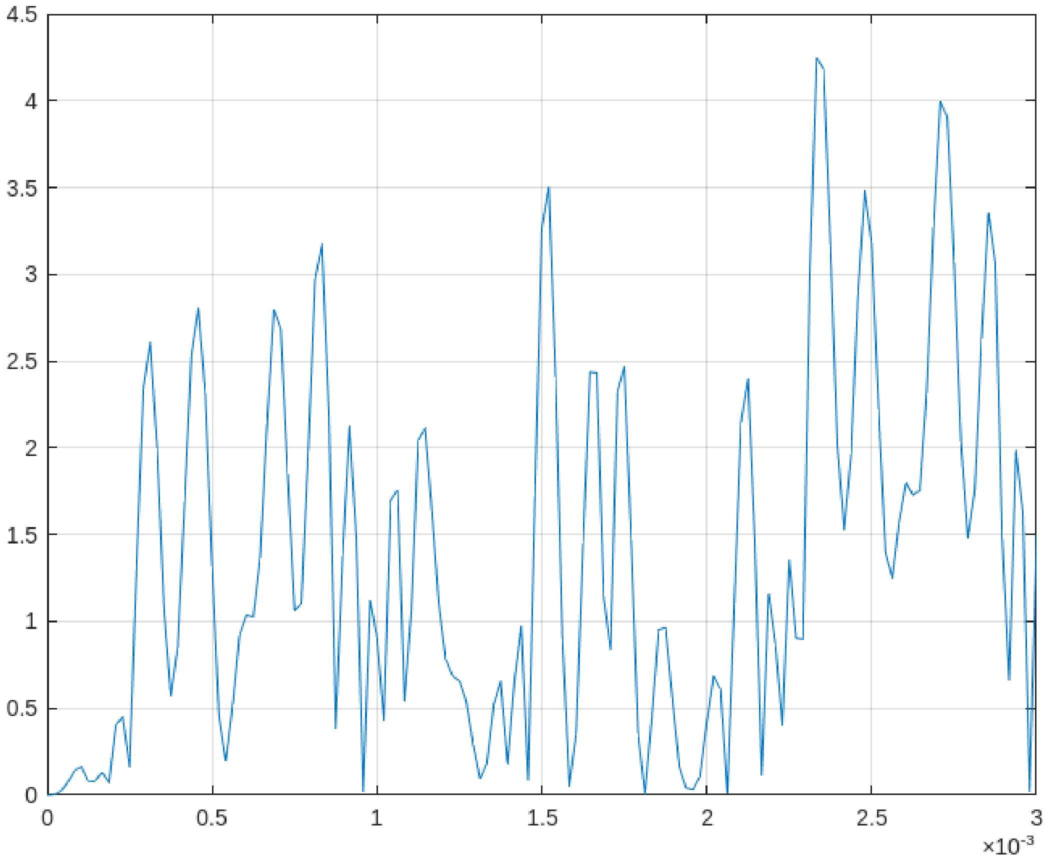

Figure 4: Filtered signal of $x(t)$ in $[0, 0.003]$

Clearly there is some change in the filtered signal y(t). From this we can't clearly conclude that the frequency greater or equal to 10000 Hz is removed or not.

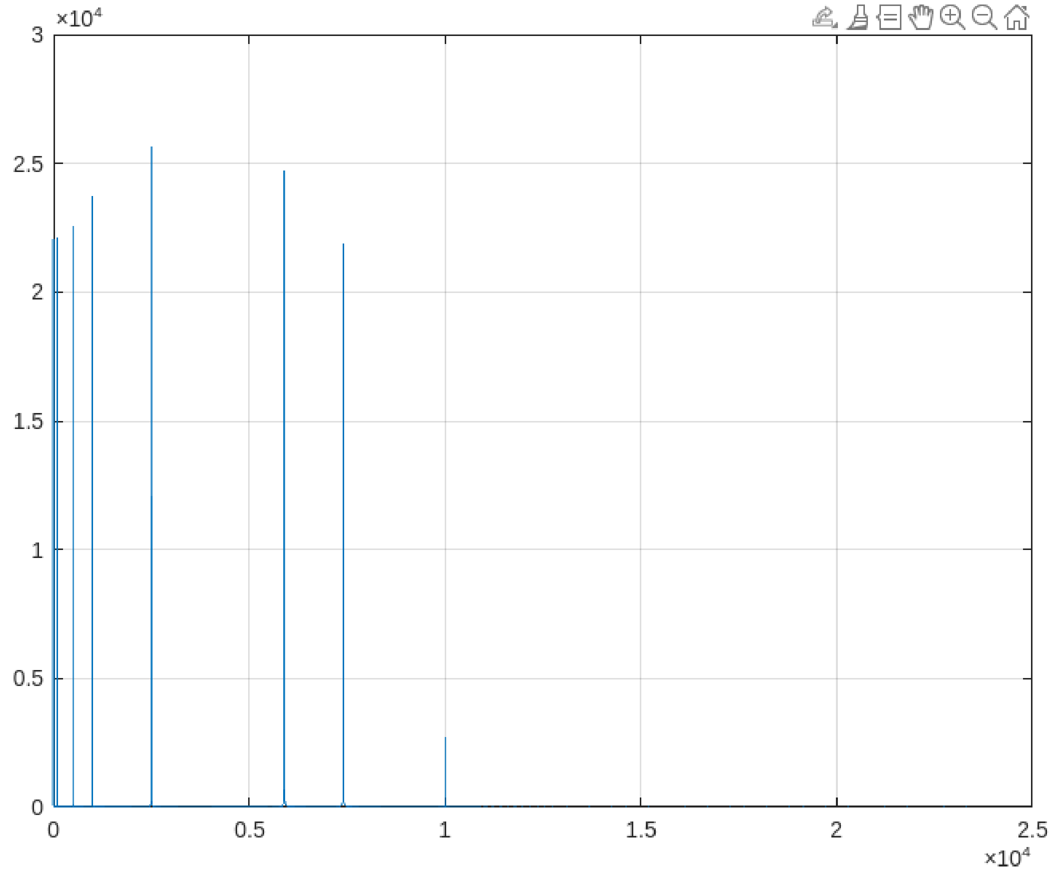

Figure 5: FFT of filtered signal of $x(t)$

We can see that the sine component of frequency 15000 Hz is successfully removed from the signal $x(t)$. From this we can say that the filter we designed is working correctly.

4 Comparsion between both normal signal and filtered ones and also their FFTs

Source code:

```
analyse_t(t, x);
hold on;
analyse_t(t, y);
hold off;
```

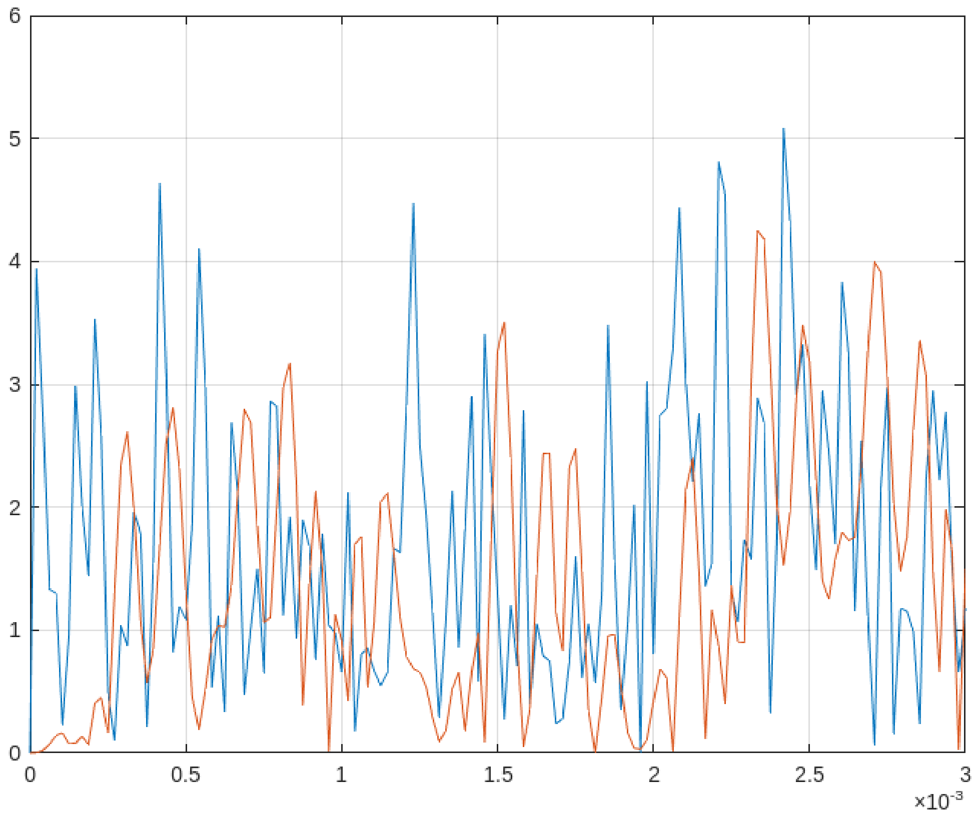

Figure 6: Blue curve=normal signal, Orange curve=filtered signal

Source code:

```
analyse_f(Fs, X);
hold on;
analyse_f(Fs, Y);
hold off;
```

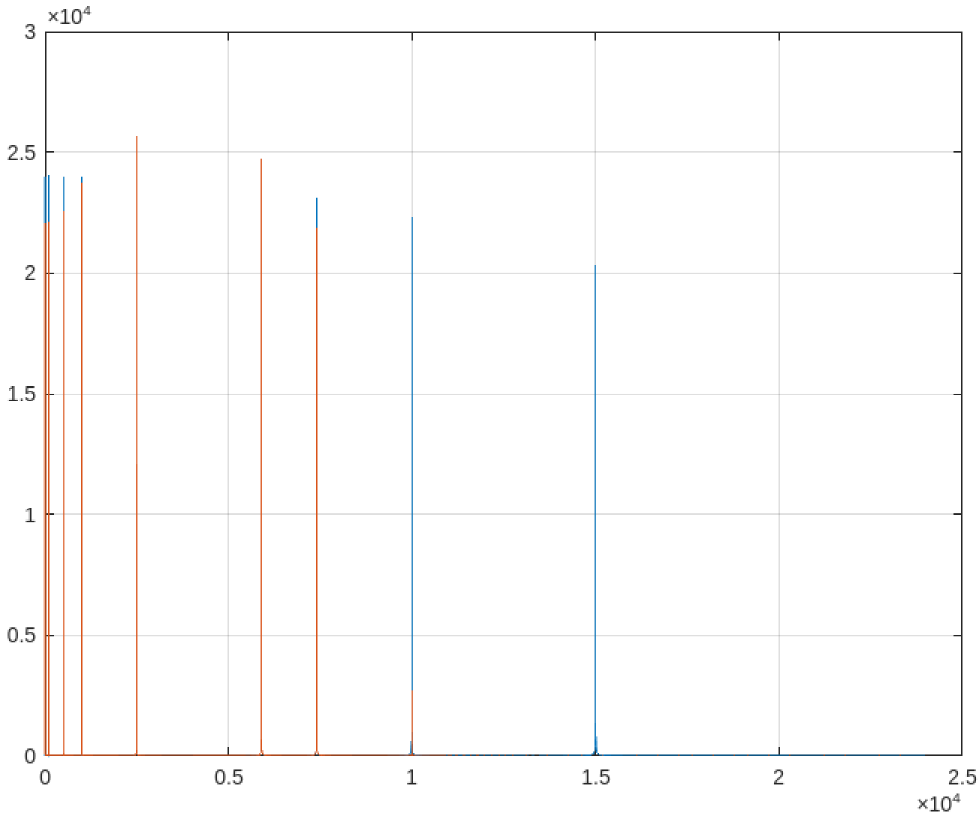

Figure 7: Blue curve=FFT of normal signal, Orange curve=FFT of filtered signal