ECE/EEE Digital Signal Processing
Lab session -6
FIR filter design using Windowing method

FIR filters can be designed using windowing method, frequency sampling method, or optimal (computer-aided) method.

One can use built-in functions or apps for the FIR filter design. For example, fir1 is the function for window based method, fir2 is the function for frequency sampling method, firpm is the function for Parks-McClellan optimal equiripple FIR filter design.

Various other functions available in signal processing toolbox are as follows:

a) **firls** Linear-phase FIR filter design using least-squares error minimization.
b) **firgr** Generalized Remez FIR filter design.
c) **filter** Filter data with digital filter
d) **designfilt** Design a digital filter
e) **digitalFilter** Digital filter class
f) **fvtool** Filter Visualization Tool (FVTool)
g) **filter** One-dimensional digital filter
h) **filterDesigner** Filter Designer – GUI for design, import and analyze filters
i) **windowDesigner** Window Designer –GUI for designing and analyzing window

Below is the description of fir1 function available in Signal processing toolbox -

>> help fir1

fir1 FIR filter design using the window method. B = fir1(N,Wn) designs an N'th order lowpass FIR digital filter and returns the filter coefficients in length N+1 vector B. The cut-off frequency Wn must be between 0 < Wn < 1.0, with 1.0 corresponding to half the sample rate. The filter B is real and has linear phase. The normalized gain of the filter at Wn is -6 dB.

B = fir1(N,Wn,'high') designs an N'th order highpass filter. You can also use B = fir1(N,Wn,'low') to design a lowpass filter.

If Wn is a two-element vector, Wn = [W1 W2], fir1 returns an order N bandpass filter with passband W1 < W < W2. You can also specify B = fir1(N,Wn,'bandpass'). If Wn = [W1 W2], B = fir1(N,Wn,'stop') will design a bandstop filter.

If Wn is a multi-element vector, Wn = [W1 W2 W3 W4 W5 ... WN], fir1 returns an order N multiband filter with bands, 0 < W < W1, W1 < W < W2, ..., WN < W < 1.

B = fir1(N,Wn,'DC-1') makes the first band a passband.

B = fir1(N,Wn,'DC-0') makes the first band a stopband.

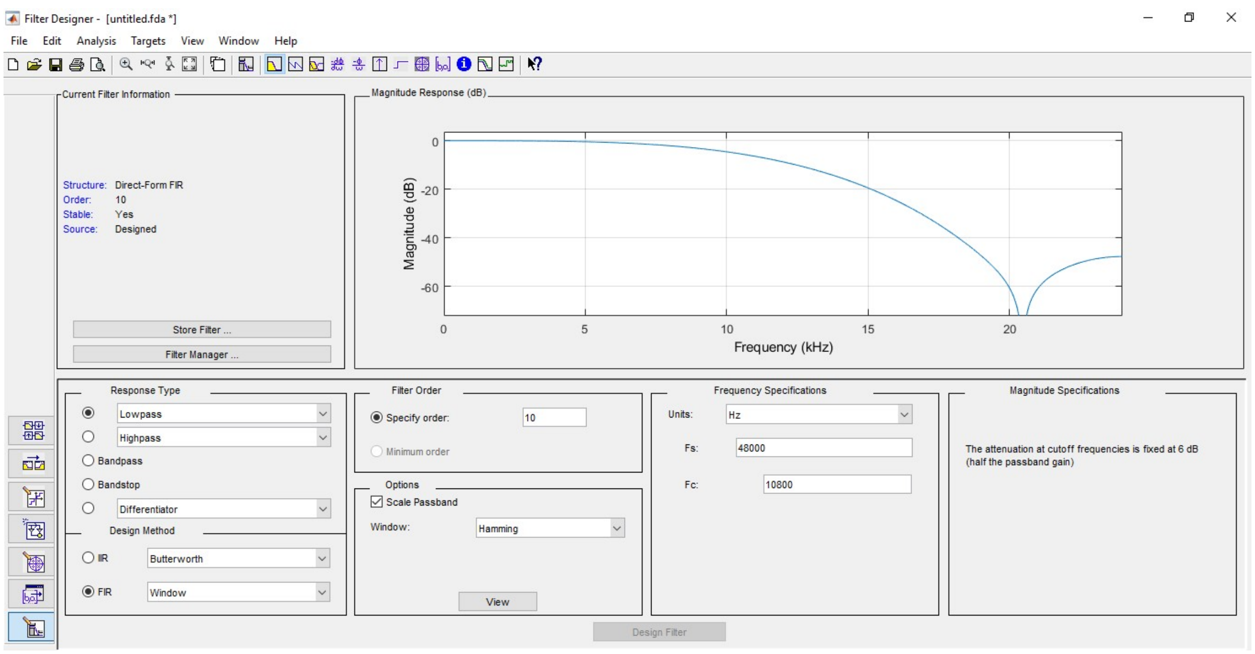
B = fir1(N,Wn,WIN) designs an N-th order FIR filter using the N+1 length vector WIN to window the impulse response. If empty or omitted, fir1 uses a Hamming window of length N+1.

For filters with a gain other than zero at Fs/2, e.g., highpass and bandstop filters, N must be even. Otherwise, N will be incremented by one. In this case the window length should be specified as N+2.

Similarly a filter designer tool can also be used to design the filter-

FilterDesigner tool :- Filter Designer is a Graphical User Interface (GUI) that allows you to design or import, and analyze digital FIR and IIR filters.

Type filterDesigner at the command prompt of MATLAB. The GUI window opens. Provide appropriate options for filter design –such as Type of filter, FIR, Window function, Order of filter, IIR, Magnitude specs etc. and then click on "Design Filter" tab. Observe the filter response. Also verify the filter's information, Magnitude and phase response, impulse response, filter co-efficients and so on.



For example – Design a **FIR Bandpass filter using Window method** with following specifications:

- Lower stopband = 0-500 Hz
- Passband = 10100 – 13900 Hz
- Upper Stopband = 23500 -24000 Hz
- Stopband attenuation = 40 dB
- Sampling Frequency = 48000 Hz.

From the theory of FIR windowing method, we know that this filter can be designed using hann (hanning) window of length 17.

Hence, one of the following methods can be used.

<u>Method 1</u>

b = fir1(16,[0.2208 0.7792],hann(17))

Which results in impulse response -

b =

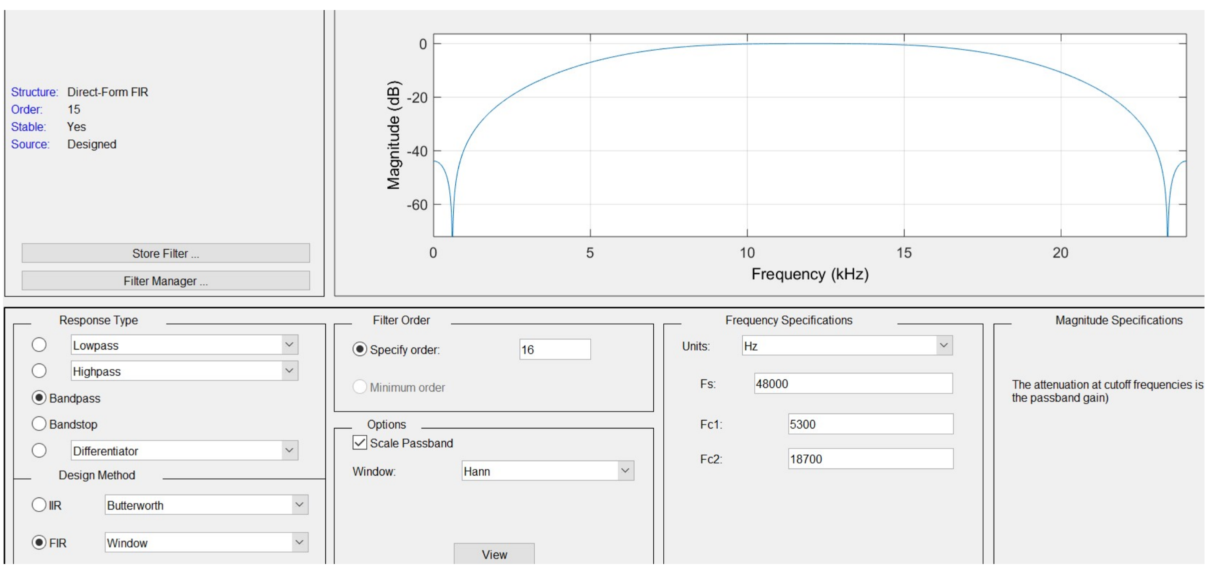
Columns 1 through 11

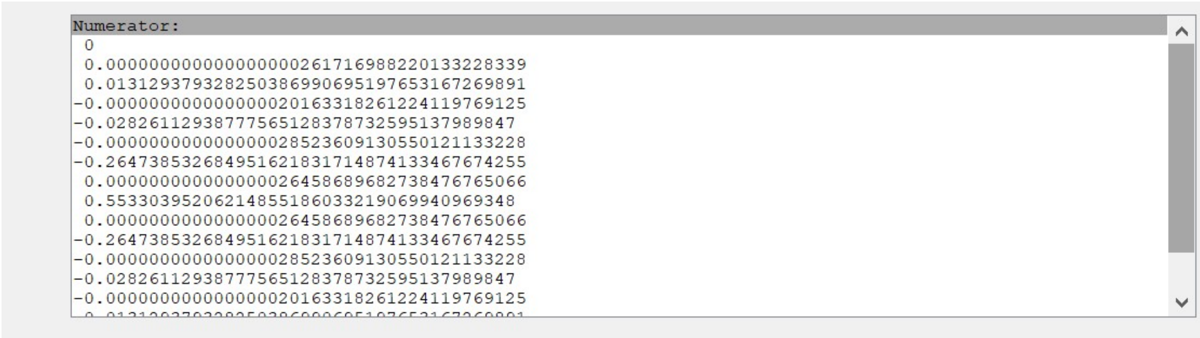
0 0.0000 0.0131 -0.0000 -0.0283 0 -0.2647 0.0000 0.5534 0.0000 -0.2647

Columns 12 through 17

0 -0.0283 -0.0000 0.0131 0.0000 0

<u>Method 2</u>





```
Numerator:
 0
 0.00000000000000000261716988220133228339
 0.013129379328250386990695197653167269891
-0.0000000000000000020163318261224119769125
-0.02826112938777565128378732595137989847
-0.0000000000000000028523609130550121133228
-0.264738532684951621831714874133467674255
 0.0000000000000000026458689682738476765066
 0.55330395206214855186033219069940969348
 0.0000000000000000026458689682738476765066
-0.264738532684951621831714874133467674255
-0.0000000000000000028523609130550121133228
-0.02826112938777565128378732595137989847
-0.0000000000000000020163318261224119769125
```

From the filterDesigner tool, filter coefficients can be exported to MATLAB using File –Export option. Provide a suitable name to the co-efficient array, which can be used as filter impulse response.

Once the impulse response is obtained by any of the method, filtering can be done using “filter” function.

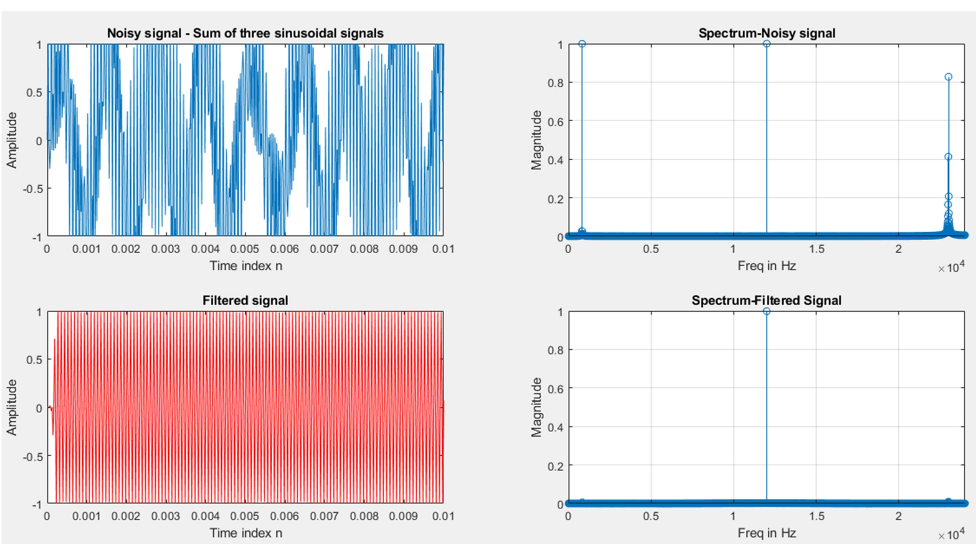
In this example, take three sinusoidal frequencies as defined below:

```
sine_1 = amplitude_1*sin(2*pi*freq_1.*time);
sine_2 = amplitude_2*sin(2*pi*freq_2.*time);
sine_3 = amplitude_3*sin(2*pi*freq_3.*time);
x_noisy = sine_1 + sine_2 + sine_3;
```

And filter the above signal using

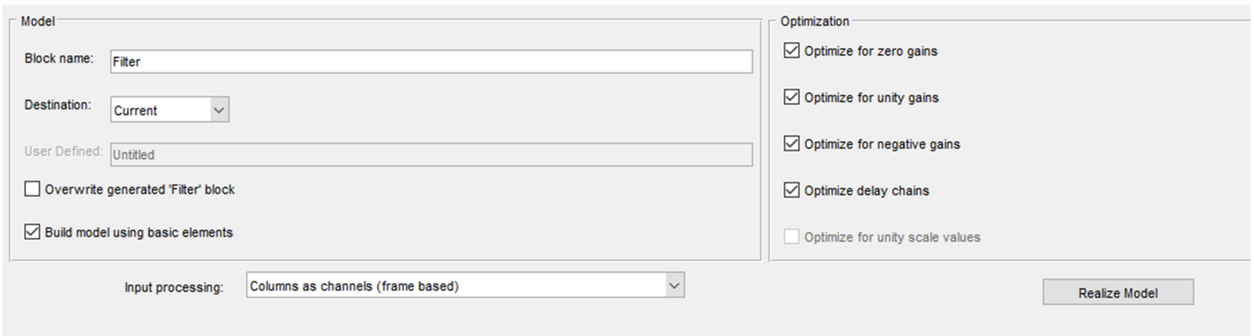
```
y = filter(h, 1, x_noisy);
```

It should result into filtered signal which can be visualized in frequency spectrum using FFT function –

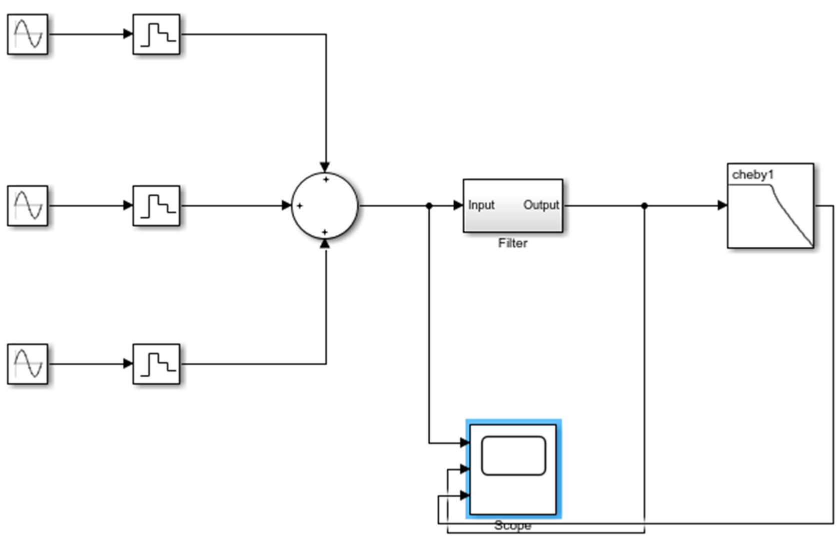


Demonstration in Simulink –

Filter designed by FilterDesigner can be exported as model in Simulink. For this, click on file-Export to Simulink model.



A Simulink model with label filter will be available in the Simulink .slx file. We will connect inputs and outputs as follows:



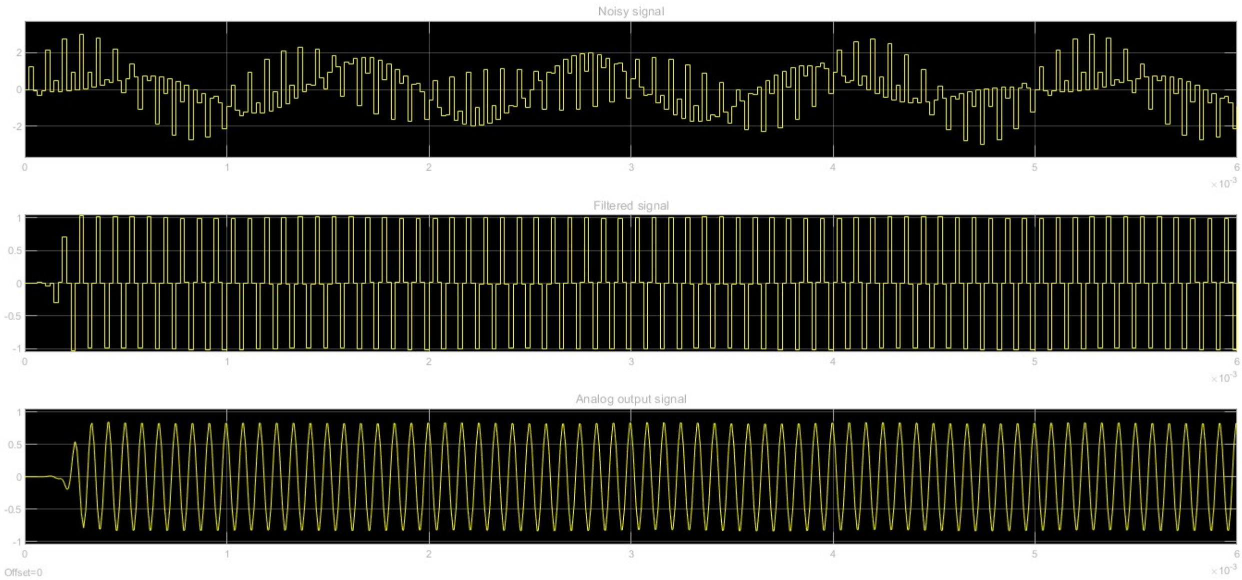
For adding the block shown here, click on "View-Library browser and type the name of the block in search field" Or you can choose the block from respective library.

For example, for sine wave block. You can search using "Sine wave", or search in library of sources.

Next add zero-order hold block (The Sine Wave source block is a continuous-time source. The Zero-Order Hold block is used to convert it to discrete-time and thus acts like an analog-to-digital converter. Make sure that the zero-order hold is set to the same sampling rate that your filters were designed for.)

Similarly, next blocks to add are – Sum, Scope, Analog filter. (Cutoff frequency of the analog filter should be half of the sampling frequency.)

Output of the scope should appear as follows:



Digital filter designer tool is also available in Simulink, and can be used directly. The model is as follows: