FIR filter design

This computer lab will investigate the FIR filter design and how it works (e.g. Demonstration of lowpass filtering). For input signal $x(n)$ we generate the mixing signal of adding two sine waves. The process $x(n)$ to be considered can be mathematically described as:

$$x(n) = A_1 \sin(2\pi f_1 n T_s) + A_2 \sin(2\pi f_2 n T_s)$$

where:

- $A_1 = A_2 = 1$ are the amplitudes, Number of samples $n = 0,\ldots, N-1$, $N = 2000$
- $f_1 = 10$ and $f_2 = 50$ are the frequencies in Hz,
- $T_s = 1/f_s$ is the sampling time in *s*, $f_s = 500$

Complete the following tasks:

1. Generate $x(n)$ in MATLAB. Make a plot of $x_1(n) = A_1 \sin(2\pi f_1 n T_s)$ and $x_2(n) = A_2 \sin(2\pi f_2 n T_s)$ as well as the mixing signal $x(n)$, Describe your observations.

   Open <u>Matlab</u>→<u>File</u>→<u>New</u>→<u>Script</u> Type the Follow script

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Demonstration of lowpass filtering
% x = sin(2*pi*f1*n*Ts) + sin(2*pi*f2*n*Ts)
% y = conv(h,x) where h = discrete sinc (lowpass filter coefficients)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, close all, clc

% Signal generation
N = 2000;
fs = 0.5e3; Ts = 1/fs;
n = 0:N-1;
f1 = 10;
f2 = 50;
s1 = sin(2*pi*f1*n*Ts);
s2 = sin(2*pi*f2*n*Ts);
x = s1 + s2 ;
t1 = 1;
```

```
t2 = 400;
figure,
subplot(521), plot(n(t1:t2),s1(t1:t2)), title('s_1(n)=sin(2*pi*f_1*n*Ts) ,
f_1=10Hz, f_s=500Hz')
subplot(523), plot(n(t1:t2),s2(t1:t2)), title('s_2(n)=sin(2*pi*f_2*n*Ts) ,
f_2=50Hz, f_s=500Hz')
subplot(525), plot(n(t1:t2),x(t1:t2)), title('x(n) = s_1(n) + s_2(n)')
```

You may want to save this and run the program
Save: **<u>File</u>→<u>Save as</u>→e.g. "James Bond.m"**

Run the programme: **<u>Copy the script</u>→<u>right click</u>→<u>Evaluate Selection</u>**

2. Generate Low pass filter $h_D[n] = 2F_c \cdot \frac{\sin(n\Omega_c)}{(n\Omega_c)}$ using IRT method. In this case, we only want retain the $x_1(n) = A_1 \sin(2\pi f_1 n T_s)$ from $x(n)$, How will you determine f_c (cut-off frequency), Assume the filter length is $2M+1=301$, the follow script may help you generate low pass filter

```
% Lowpass filtering
L = 301; %2M+1
fc = ??; % This requires you to determine
f_sampling = fs;
Fc = fc/f_sampling;
k = -(L-1)/2:(L-1)/2;
h = 2*Fc*sinc(2*k*Fc);%IRT method
```

%high pass filter
`This requires you to determine`

3. Once you determine the cut-off frequency and completed the design of the low pass filter, the next step is pass the input mixing signal $x(n)$ through the filter to see what output you get.

```
%Filtering
y = conv(h,x);% convolution with x(n) and h[m]
m = -(L-1)/2:length(y)-1-(L-1)/2;
subplot(527), stem(-(L-1)/2:(L-1)/2,h(1:L),'o'),
title('h(n)=2*Fc*sinc(2*n*Fc),f_c=30Hz,f_s=500Hz')
subplot(529), plot(m(t1:t2),y(t1:t2)), title('y(n) = h(n) * x(n)')
```

4. Above cases are all in time domain, let's look at their frequency domain and describe the plot.

```
% Frequency domain
NFFT = 2^ceil(log2(length(y)));% Number of FFT (should be 2^r)
S1 = fft(s1,NFFT);% FFT of signal x1(n)
S2 = fft(s2,NFFT); % FFT of signal x2(n)
X = fft(x,NFFT); % FFT of signal x(n)
H = fft(h,NFFT); % FFT of filter h(m)
Y = fft(y,NFFT); % FFT of filtered signal y(n)
f2=linspace(-fs/2,fs/2,NFFT);
subplot(522), plot(f2,fftshift(abs(S1))),xlabel('f [Hz]'), title('S_1(f)')
subplot(524), plot(f2,fftshift(abs(S2))),xlabel('f [Hz]'), title('S_2(f)')
subplot(526), plot(f2,fftshift(abs(X))),xlabel('f [Hz]'), title('X(f)')
subplot(528), plot(f2,fftshift(abs(H))),xlabel('f [Hz]'), title('H(f)')
subplot(5,2,10), plot(f2,fftshift(abs(Y))),xlabel('f [Hz]'), title('Y(f)')
```

5. Can we use windowed method to improve the IRT method (reduce ripples and etc). The follows show the different window function.

```
% Rectangular window(IRT method)
N = 31;
w = boxcar(N)';
W = fft([zeros(1,2*N) w zeros(1,2*N)]);
figure, subplot(211),plot(w), title('Rectangular window (time domain)')
subplot(212),plot(10*log10(abs(fftshift(W))))
title('Rectangular window (frequency domain)')
% Hamming window
w = hamming(N)';
W = fft([zeros(1,2*N) w zeros(1,2*N)]);
figure, subplot(211),plot(w), title('Hamming window (time domain)')
subplot(212),plot(10*log10(abs(fftshift(W))))
title('Hamming window (frequency domain)')
% Hanning window
w = hanning(N)';
W = fft([zeros(1,2*N) w zeros(1,2*N)]);
figure, subplot(211),plot(w), title('Hanning window (time domain)')
subplot(212),plot(10*log10(abs(fftshift(W))))
title('Hanning window (frequency domain)')
% Blackman  window
w = blackman(N)';
W = fft([zeros(1,2*N) w zeros(1,2*N)]);
figure, subplot(211),plot(w), title('Blackman window (time domain)')
subplot(212),plot(10*log10(abs(fftshift(W))))
title('Blackman window (frequency domain)')
```

6. Give the following specification which required from our customer, can we design the weindow based low pass filter use matlab codes (Hint: just expand your thinking and use above codes to design this one).

Specifications	**Value**
Passband edge frequency	25Hz
Stopband attenuation	> 50 dB
Transition width	5Hz
Sampling interval	500Hz