6

EXPERIMENT

FIR Filter Design

AIM

Part I: FIR Low Pass and High Pass Filters

(a) To design a Low Pass FIR filter using Hamming window and to plot its frequency response

(b) To design a High Pass FIR filter using Hamming window and to plot its frequency response

(c) To demonstrate the operation of the designed filters by filtering a composite frequency signal and plotting the filtered output

Part II: FIR Band Pass and Band Stop Filters

(a) To design a Band Pass FIR filter using Hamming window and to plot its frequency response

(b) To design a Band Stop FIR filter using Hamming window and to plot its frequency response

THEORY

Ideal Filter Impulse Responses

An Ideal Low Pass Filter with generalized linear phase is given by the frequency response

$$H_{lp}(e^{j\omega}) = \begin{cases} e^{-j\omega M/2} & ;|\omega| < \omega_c \\ 0 & ;\omega_c < |\omega| \leq \pi \end{cases}$$

Corresponding ideal impulse response is,

$$h_{lp}[n] = \frac{sin(\omega_c(n - M/2))}{\pi(n - M/2)}; \quad -\infty < n < \infty$$

An ideal High Pass filter with generalized linear phase is given by,

$$H_{hp}(e^{j\omega}) = \begin{cases} 0 & ;|\omega| < \omega_c \\ e^{-j\omega M/2} & ;\omega_c < |\omega| \leq \pi \end{cases}$$

$$h_{hp}[n] = \frac{sin(\pi(n - M/2))}{\pi(n - M/2)} - \frac{sin(\omega_c(n - M/2))}{\pi(n - M/2)}; \quad -\infty < n < \infty$$

where ω_c is the cut-off frequency.
An ideal Band Pass Filter with generalized linear phase is given by,

$$H_{bp}(e^{j\omega}) = \begin{cases} e^{-j\omega M/2} & ;\omega_{c1} < |\omega| < \omega_{c2} \\ 0 & ;0 < |\omega| < \omega_{c1} \\ 0 & ;\omega_{c2} < |\omega| \leq \pi \end{cases}$$

Corresponding ideal impulse response is,

$$h_{bp}[n] = \frac{sin(\omega_{c2}(n - M/2))}{\pi(n - M/2)} - \frac{sin(\omega_{c1}(n - M/2))}{\pi(n - M/2)}; \quad -\infty < n < \infty$$

An ideal Band Stop Filter with generalized linear phase is given by,

$$H_{bs}(e^{j\omega}) = \begin{cases} 0 & ;\omega_{c1} < |\omega| < \omega_{c2} \\ e^{-j\omega M/2} & ;0 < |\omega| < \omega_{c1} \\ e^{-j\omega M/2} & ;\omega_{c2} < |\omega| \leq \pi \end{cases}$$

Corresponding ideal impulse response is,

$$h_{bs}[n] = \frac{sin(\pi(n - M/2))}{\pi(n - M/2)} + \frac{sin(\omega_{c1}(n - M/2))}{\pi(n - M/2)} - \frac{sin(\omega_{c2}(n - M/2))}{\pi(n - M/2)}; \quad -\infty < n < \infty$$

where ω_{c1} and ω_{c2} are the two cut-off frequencies.

Window Method of FIR Design

The basic idea behind the window method for FIR filter design is to truncate the impulse response of an ideal frequency-selective filter (which always has a non-causal, infinite duration impulse response) to obtain a linear phase and causal FIR filter.
Truncation of the ideal response can be achieved by defining a new system with impulse reponse, $h[n]$ defined as the product of the ideal system's impulse response, $h_d[n]$ and a finite duration "window" sequence $w[n]$ as,

$$h[n] = h_d[n]w[n]$$

Consider the simple truncation shown below,

$$h[n] = \begin{cases} h_d[n] & ;0 \le n \le M \\ 0 & ;otherwise \end{cases}$$

Here, the window we have used is called as the rectangular window and is defined as,

$$w[n] = \begin{cases} 1 & ;0 \le n \le M \\ 0 & ;otherwise \end{cases}$$

Here M is the order of the filter (corresponding impulse response length will be $M+1$).
Due to the disadvantages associated with rectangular window we normally choose other "tapered" windows. Depending on how we define $w[n]$, we obtain different window designs.

Hamming window

This is a raised cosine window optimized to minimize the maximum (nearest) side lobe.
The Hamming window of length $M+1$ is given by,

$$w[n] = \begin{cases} 0.54 - 0.46cos\frac{2\pi n}{M} & ;0 \le n \le M \\ 0 & ;otherwise \end{cases}$$

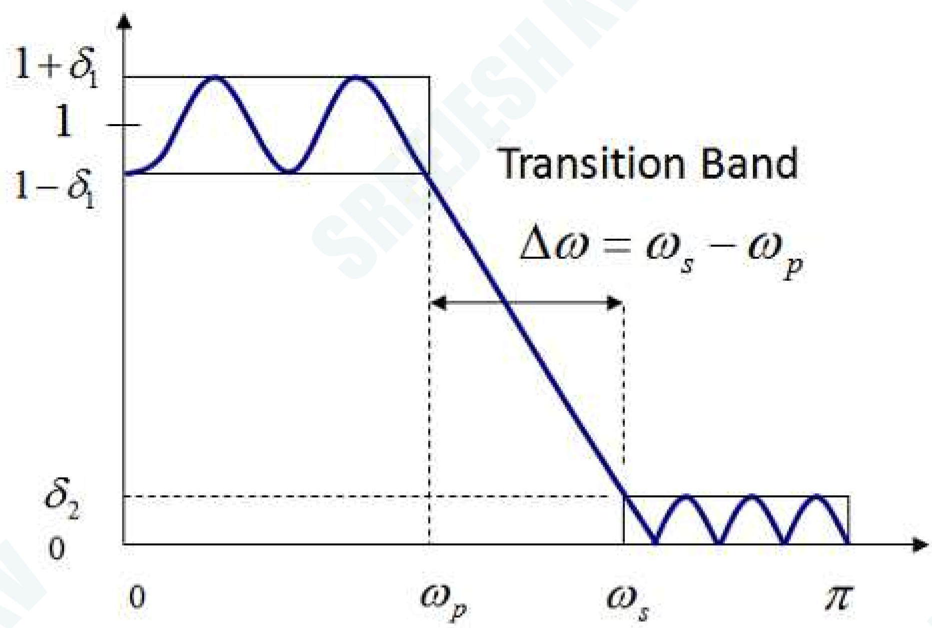
The approximate transition width of main lobe is $8\pi/(M+1)$.
The minimum stop band attenuation is 53dB.

Kaiser's formula to find order

Kaiser's formula gives an approximate estimate of the order of a linear phase FIR filter with the given cut off frequencies and ripple specifications.
The formula for minimum order (M) is,

$$M \cong \frac{-10log_{10}(\delta_1\delta_2) - 13}{14.6(\Delta\omega/2\pi)}$$

The parameters in the equation are shown below.



MATLAB FUNCTIONS USED

fir1

Window-based FIR filter design

b = fir1(n,Wn) uses a Hamming window to design an nth-order lowpass, bandpass, or multiband FIR filter with linear phase. The filter type depends on the number of elements of Wn

b = fir1(n,Wn,ftype) designs a lowpass, highpass, bandpass, bandstop, or multiband filter, depending on the value of ftype and the number of elements of Wn.

Wn represents Frequency constraints, specified as a scalar, a two-element vector, or a multi-element vector.

All elements of Wn must be strictly greater than 0 and strictly smaller than 1, where 1 corresponds to the Nyquist frequency: 0 < Wn < 1. The Nyquist frequency is half the sample rate or π rad/sample.

If Wn is a scalar, then fir1 designs a lowpass or highpass filter with cutoff frequency Wn. The cutoff frequency is the frequency at which the normalized gain of the filter is -6 dB.
If Wn is the two-element vector [w1 w2], where w1 < w2, then fir1 designs a bandpass or bandstop filter with lower cutoff frequency w1 and higher cutoff frequency w2.

ftype represents the Filter type - It can be 'low' , 'bandpass' , 'high' , 'stop', 'DC-0' or 'DC-1'

hamming

Hamming window

w = hamming(L) returns an L-point symmetric Hamming window.

freqz

Frequency response of digital filter

[h,w] = freqz(b,a,n) returns the n-point frequency response vector h and the corresponding angular frequency vector w for the digital filter with transfer function coefficients stored in b and a.
freqz(__) with no output arguments plots the frequency

```
response of the filter.
[h,f]=freqz(__,n,fs) returns the frequency response vector
h and the corresponding physical frequency vector f for a
digital filter designed to filter signals sampled at a rate
fs.
```

filter

1-D digital filter

```
y = filter(b,a,x) filters the input data x using a rational
transfer function defined by the numerator and denominator
coefficients b and a.

If a(1) is not equal to 1, then filter normalizes the filter
coefficients by a(1). Therefore, a(1) must be nonzero.
```

Note:

For highpass and bandstop configurations, `fir1` always uses an even filter order(n). The order must be even because odd-order symmetric FIR filters must have zero gain at the Nyquist frequency. If you specify an odd n for a highpass or bandstop filter, then `fir1` increments n by 1.In this case, the length of the filter coefficients will be odd($n+1$) and hence the length of the window selected should also be $n+1$

ALGORITHM

Part 1: FIR Low Pass and High Pass Filters

Step 1. Start

Step 2. Input the pass band ripple rp in dB, stop band ripple rs in dB, pass band and stop band edge frequencies fp and fs in Hz and sampling frequency $fsamp$ in Hz

Step 3. Compute the digital band edge frequencies and normalize them (=digital frequencies/π)

Step 4. Find the minimum order of the filter required to satisfy the given specifications by using Kaiser's formula as, $M = ceil((-20*log10(sqrt(dp*ds))-13)/(14.6*(ws-wp)/2))$

Step 5. If M is odd, increment M to ensure that the filter is a symmetrical linear phase type-1 filter

Step 6. Find normalized cut off frequency using, $wn = (wp + ws)/2$

Step 7. Find the $N = M + 1$ long symmetric hamming window using the function `hamming`

Step 8. Design the low pass FIR filter from the hamming window and the normalized cut off frequency wn using the `fir1` function

Step 9. Plot the magnitude and phase response using MATLAB function `freqz`

Step 10. Repeat steps 7 and 8 by setting the window type option to 'high' in `fir1` function to obtain the high pass FIR filter for the same specifications and to plot its magnitude and phase response.

Step 11. Create a composite signal as the sum of two sinusoids: one having a frequency less than fp and the other having a frequency higher than fs. Filter the composite signal using the designed lowpass and highpass filters and observe & plot the filters' input and output in time domain

Step 12. Stop

Part 2: FIR Band Pass and Band Stop Filters

Step 1. Start

Step 2. Define the band edge frequencies in Hz, Order of the filter M and sampling frequency $fsamp$

Step 3. Compute the digital band edge frequencies and normalize them (=digital frequencies/π)

Step 4. Find the $M+1$ point symmetric hamming window using the function hamming()

Step 5. Design the M-th order band pass FIR filter using the hamming window and the normalized band edge frequencies using the `fir1` function.

Step 6. Plot the magnitude and phase response using MATLAB function `freqz`

Step 7. Repeat steps 5 and 6 by setting the window type option to 'stop' in `fir1` function to obtain the bandstop FIR filter coefficients for the same specifications and to plot its magnitude and phase response.

Step 8. Stop

PROGRAM

Part 1: FIR Low Pass and High Pass Filters

```
%Title: Program to
% 1) Design a Low Pass FIR filter using Hamming window and to ...
    plot its frequency response
% 2) Design a High Pass FIR filter using Hamming window and to ...
    plot its frequency response
% 3) Demonstrate the designed filters by filtering a composite ...
    frequency
%signal and plotting the filtered output

%Author: Sreejesh K V, Dept. of ECE, GCEK
%Date: 02/10/2022

```

```
clc;
clear;
close all;

fsamp=2000;% Sampling frequency in Hz
fp=200;%passband edge frequency in Hz
fs=400;%stopband edge frequency in Hz
wp=2*fp/fsamp;%normalized digital passband edge frequency(=dig ...
    frequency/pi)
ws=2*fs/fsamp;%normalized digital stopband edge frequency
rp=0.1;% peak-to-peak passband ripple in dB
rs=35;%minimum stopband attenuation in dB

% ---   Calculating filter order M=N-1   --- %
deltap=(10^(rp/20)-1)/(10^(rp/20)+1);%converting from dB
deltas=1/(10^(rs/20));%stopband ripple
M=((-20*log10(sqrt(deltap*deltas)))-13)/(14.6*(ws-wp)/2);%M=estimated ...
    min. filer order
M=ceil(M);%round up to next highest integer if M has a fractional ...
    part

% Making the filter order Even for linear phase ...
    operation(symmetrical filter coefficients)
if rem(M,2) ~= 0
M=M+1;
end;

N=M+1;%N=length of filter coefficients - Will be an odd value at ...
    this point

%-- Design of Lowpass filter --%
wn=(wp+ws)/2;%normalized digital cutoff frequency for the filter
ham=hamming(N);%N length hamming window samples; N=M+1 will be ...
    Odd Here
b=fir1(M,wn,ham);%designs an M th order Lowpass fir filter with ...
    the given cutoff frequency & window

%-- Design of Highpass filter --%
bh=fir1(M,wn,'high',ham);%designs an Mth order Highpass fir ...
    filter with the given cutoff frequency & window

%-- Displaying the hamming window samples
figure(1);
stem(0:N-1,ham);
title(['Hamming window samples of Length =' num2str(N)]);
xlabel('n');
ylabel('Amplitude');

%--Plotting the frequency response of the low pass filter
figure(2);
freqz(b,1,256,fsamp);
```

```
title('Response of the Lowpass filter');

%--Plotting the frequency response of the high pass filter
figure(3);
freqz(bh,1,256,fsamp);
title('Response of the Highpass filter');

% -- Demonstration of Filtering operation for both filters --%
f1=100;
f2=500;
T=1/f1;
t=0:1/fsamp:3*T;
s1=sin(2*pi*f1*t);%samples of 100Hz sine wave
s2=sin(2*pi*f2*t);%samples of 500Hz sine wave
s=s1+s2;%the composite signal

%---Filtering Process--%
yl=filter(b,1,s);%low pass filtering
yh=filter(bh,1,s);%highpass filtering

%---Plotting the c.t.signals--%
figure;
subplot(2,2,1);
plot(t,s1,'r','LineWidth',2);
hold on
plot(t,s2);
legend('s1','s2');
title(['Signals s1 ( f = ' num2str(f1) ' Hz) & s2 ( f = ' ...
    num2str(f2) ' Hz)  ']);
xlabel('time in seconds');
ylabel('Amplitude');

subplot(2,2,2);
plot(t,s);
title('Composite signal=s1+s2');
xlabel('time in seconds');
ylabel('Amplitude');

subplot(2,2,3);
plot(t,yl);
title('LowPass Filtered Output');
xlabel('time in seconds');
ylabel('Amplitude');

subplot(2,2,4);
plot(t,yh);
title('HighPass Filtered Output');
xlabel('time in seconds');
ylabel('Amplitude');
```

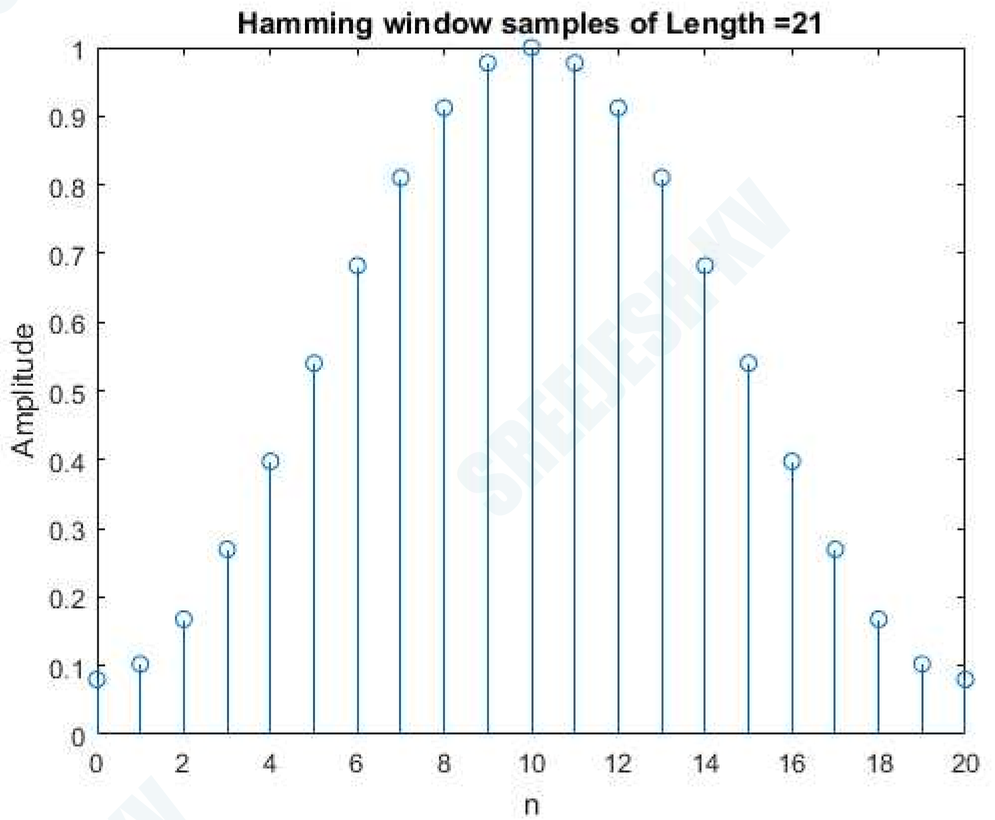
Part 2: FIR Band Pass and Band Stop Filters

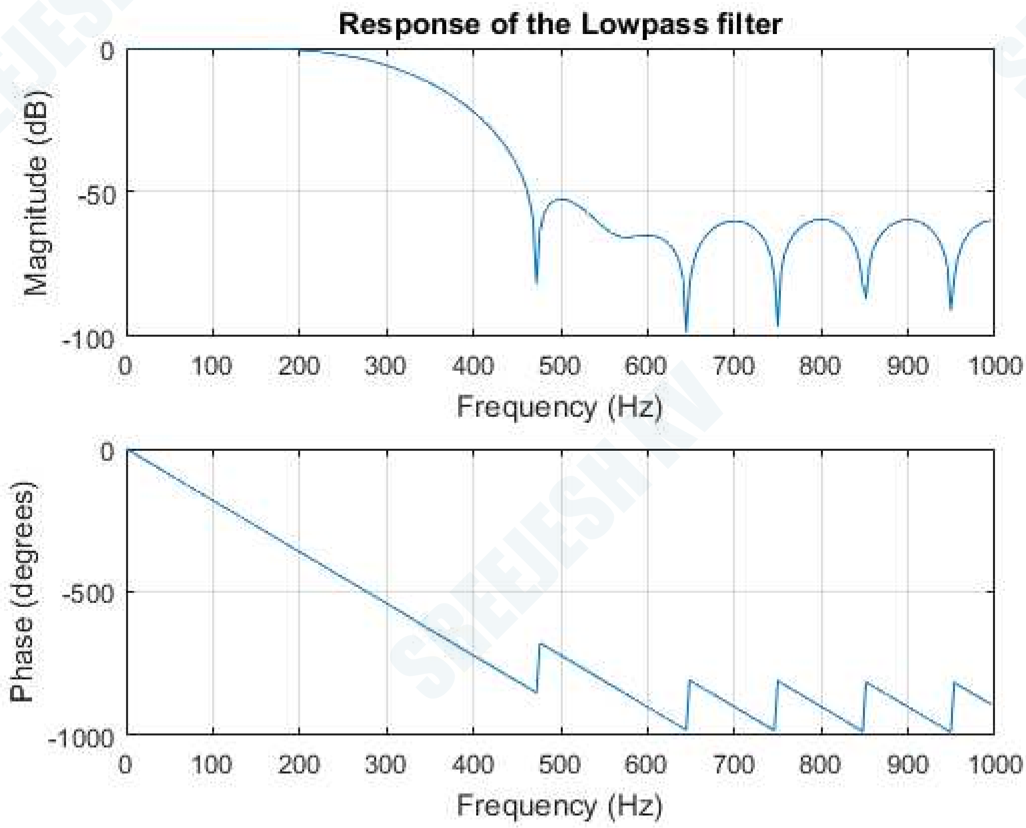
```
%Title: Program to
% 1) Design a Band Pass FIR filter using Hamming window and to ...
    plot its frequency response
% 2) Design a Band Stop FIR filter using Hamming window and to ...
    plot its frequency response

%Author: Sreejesh K V, Dept. of ECE, GCEK
%Date: 02/10/2022

clc;
close all;
clear;

fp=[400 600];%band edge frequencies in Hz
M=60;%order of the filter
fsamp=2000;% Sampling frequency in Hz
wp=(2/fsamp).*fp;% Normalizing the frequencies
y=hamming(M+1);%length of window; M+1 will be Odd Here

% ---Designing the Bandpass filter--- %
bpass=fir1(M,wp);%Calculation of filter coefficients

% ---Designing the bandstop filter with the same cutoff ...
    frequencies--- %
bstop=fir1(M,wp,'stop',y);%Calculation of filter coefficients

%--Plotting the BP filter response
figure,
freqz(bpass,1,500,fsamp);
title('Magnitude and Phase response of Band Pass Filter');


%--Plotting the BS filter response
figure,
freqz(bstop,1,500,fsamp);
title('Magnitude and Phase response of Band Stop Filter');
```

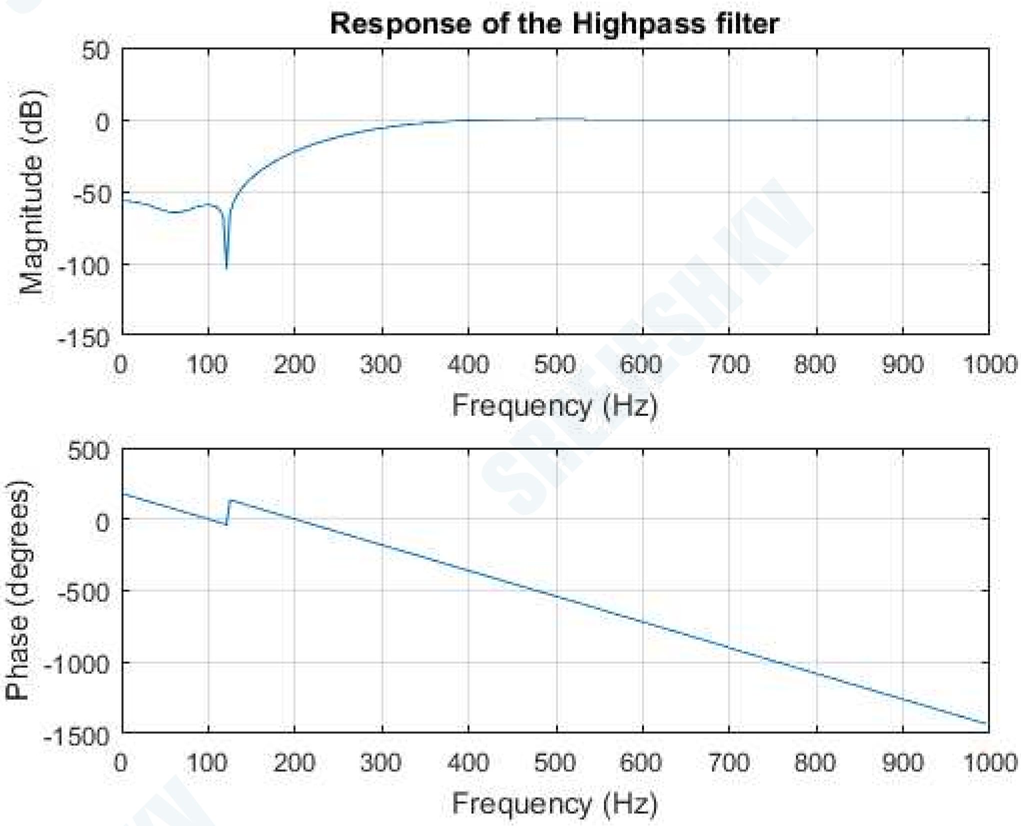
OUTPUT & OBSERVATIONS

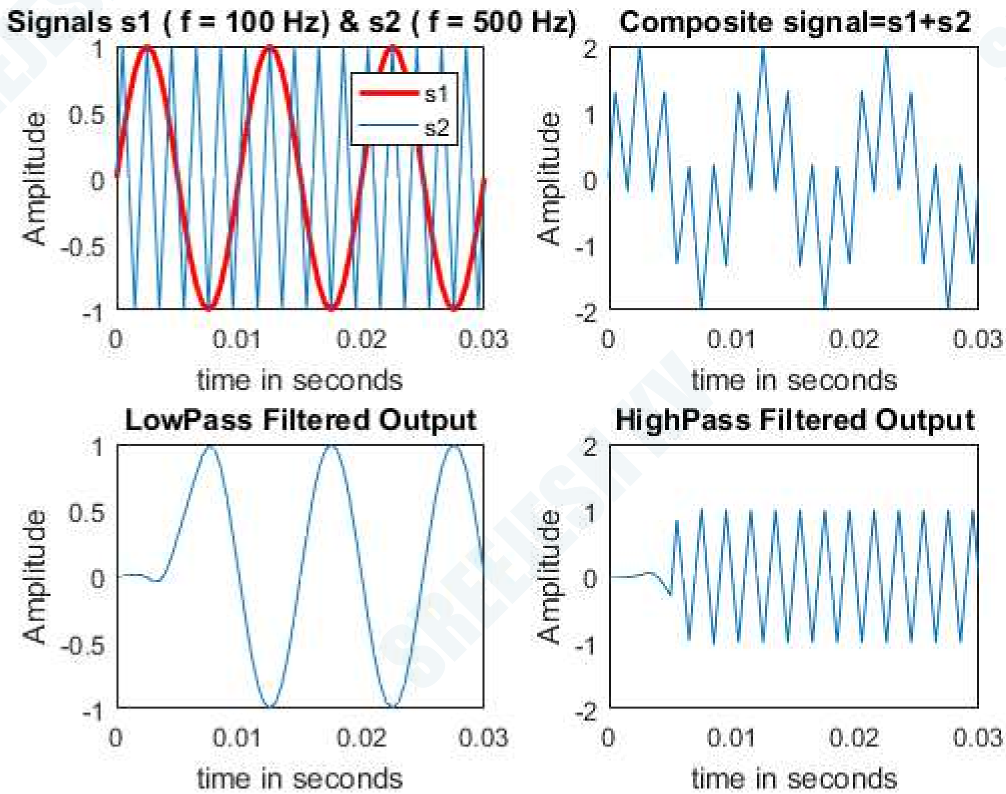
Part 1: FIR Low Pass and High Pass Filters

Figure Window Output:


Hamming window samples of Length =21
Amplitude
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2
4
6
8
10
12
14
16
18
20
n

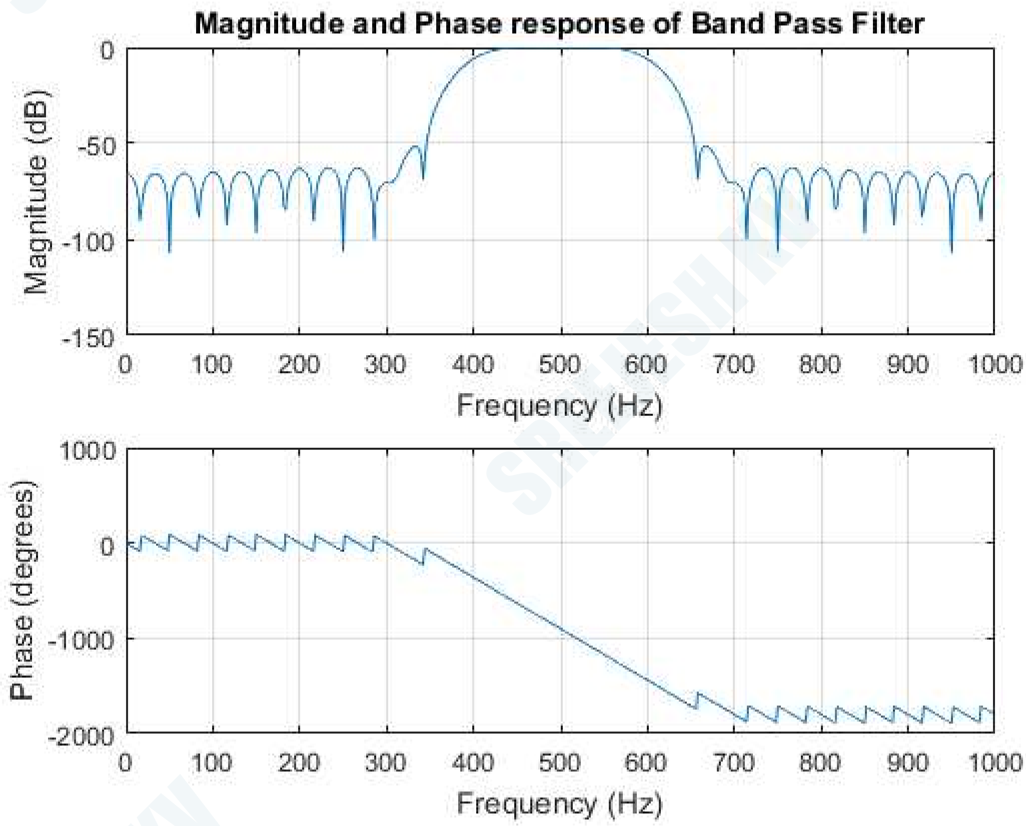

Response of the Lowpass filter
Magnitude (dB)
0
-50
-100
0
100
200
300
400
500
600
700
800
900
1000
Frequency (Hz)
Phase (degrees)
0
-500
-1000
0
100
200
300
400
500
600
700
800
900
1000
Frequency (Hz)

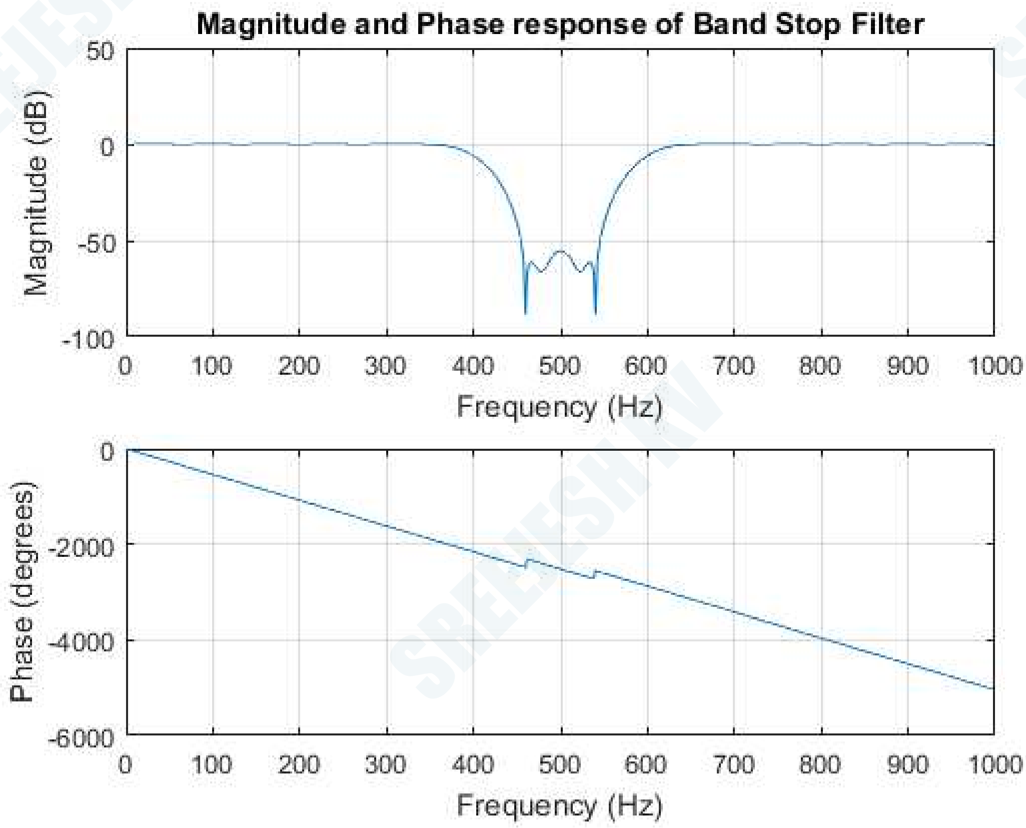




Part 2: FIR Band Pass and Band Stop Filters

Figure Window Output:





RESULTS

Part I: FIR Low Pass and High Pass Filters

(a) A Low Pass FIR filter was designed in MATLAB using Hamming window and its frequency response was plotted

(b) A High Pass FIR filter was designed in MATLAB using Hamming window and its frequency response was plotted

(c) The operation of the designed filters was demonstrated by filtering a composite frequency signal and plotting the filtered output in MATLAB

Part II: FIR Band Pass and Band Stop Filters

(a) A Band Pass FIR filter was designed in MATLAB using Hamming window and its frequency response was plotted

(b) A Band Stop FIR filter was designed in MATLAB using Hamming window and its frequency response was plotted