

```

% PROGRAM FOR THE DESIGN OF FIR LPF, HPF, BPF & BSF USING HANNING
WINDOW
%Enter the Pass Band Ripple: .05
%Enter the Stop Band Ripple: .04
%Enter the Pass Band Frequency: 1500
%Enter the Stop Band Frequency: 2000
%Enter the Sampling Frequency: 8000

%We want to design a Discrete Time Low Pass Filter for a voice
signal. The specifications are:
%Passband fp=1500 Hz, with 0.05dB ripple;
%Stopband fs=2000 Hz, with 50dB attenuation and with 0.04dB ripple;
%Sampling Frequency f=8000 Hz.
%Recall the mapping from analog to digital frequency normalised
passband frequency is given by  $w_p = 2*f_p/f$ , and normalised stopband
frequency is given by  $w_s = 2*f_s/f$  with f as the sampling frequency.

clc; close all; clear all;
%rp = input('Enter the Pass Band Ripple: .05 ');
%rs = input('Enter the Stop Band Ripple: .04 ');
%fp = input('Enter the Pass Band Frequency: 1500 ');
%fs = input('Enter the Stop Band Frequency: 2000 ');
%Fs = input('Enter the Sampling Frequency: 2e3,4e3,8e3 ');
rp = .05;
rs = .04;
fp = 2e3;
fs = 25e3;
Fs = 800e3;

Ts=1/Fs;          %set sampling time
dt=0:Ts:5e-3-Ts; %Lets say 5 ms

f1=1e3; %say 1 kHz
f2=20e3; %say 20 kHz
f3=30e3; %say 30 kHz

%Noisy Signal
y=5*sin(2*pi*f1*dt)+5*sin(2*pi*f2*dt)+10*sin(2*pi*f3*dt);
subplot(5,2,1) ;
plot(dt,y);
title(' Noisy Signal= y(t)');

%convert the noisy signal to frequency domain
nfft=length(y);
nfft2=2.^nextpow2(nfft); %select line 15&16 press F9

fy=fft(y,nfft2);          %FFT of y signal fy is complex: both mag
and phase, but we want only mag so->

```

```
%subplot(5,5,2) ;
%stem(abs(fy)); %FFT is mirrored.Because both the positive
and negative frequency sinusoids are 90 degrees out of phase and
have the same magnitude, they will both respond to real signals in
the same way. Or rather, the magnitude of their response will be the
same, but the correlation phase will be different.
%title(' ***** Mirrored FFT signal: fy *****');
```

```
%So we will go for only left half of the mag of the fft signal
fy=fy(1:nfft2/2);
xfft=Fs.*(0:nfft2/2-1)/nfft2; %X-axis of the fft sinal
```

```
%subplot(5,2,2) ;
%plot(xfft,abs(fy));
%title(' ***** Left side FFT signal: mag fy *****');
```

```
subplot(5,2,2) ;
plot(xfft,abs(fy/max(fy))); %Normalising Y-axis (magnitude) to 1
title('Y(w)=Left side FFT signal: mag fy Normalised');
```

```
wp = 2 * fp /Fs;
ws = 2 * fs/Fs;
num = - 20 * log( sqrt(rp*rs))- 13;
den = 14.6 * (fs-fp)/Fs;
n = ceil (num/den) ;
%n=30000;
n1 = n+1;
if(rem(n,2)~=0)
n1 = n;
n = n-1;
end
win = hanning(n1);
disp('Filter order n= ');n
```

```
% LOW PASS FILTER
b1 = fir1(n,wp,win);
[h,w] = freqz(b1,1,256);
%[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.
%Ref: https://in.mathworks.com/help/signal/ref/freqz.html
```

```
fh_low=fft(b1,nfft2); %Transform impulse response from time domain
to frequency domain
fh_low=fh_low(1:nfft2/2); %Only LHS of the +ve frequency (X-axis/2)
```

```
m = 20 * log(abs(h));
```

```
subplot(5,2,3) ;
```

```

%plot(xfft,abs(fh_low));
plot(xfft,abs(fh_low/max(fh_low))); %Normalising Y-axis (magnitude)
to 1
title('H(w)=Impulse response in Frquency Domain LP');

%plot (w/pi,m) ;
title(' ***** HANNING WINDOW *****');
%ylabel('Gain indb----->');
%xlabel(' (a) Normalised Frequency----->');

%con=conv(y,b);

mul=fh_low.*fy; %Multiply Impulse response in Freq domain*FFT of y
signal

subplot(5,2,4) ;
%plot(con);
plot(xfft,abs(mul/max(mul)));
title(' ***** Filtered signal: Low pass *****');

% HIGH PASS FILTER
b2 = fir1(n,wp,'high',win);
[h,w] = freqz(b2,1,256);

fh_high=fft(b2,nfft2); %Transform impulse response from time domain
to frequency domain
fh_high=fh_high(1:nfft2/2); %Only LHS of the +ve frequency (X-
axis/2)

m = 20*log(abs(h));
subplot(5,2,5);

plot(xfft,abs(fh_high/max(fh_high))); %Normalising Y-axis
(magnitude) to 1
title('H(w)=Impulse response in Frquency Domain HP');

%plot(w/pi,m) ;
%ylabel('Gain in db----->');
%xlabel(' (b) Normalised Frequency----->');

%con=conv(y,b);

mul=fh_high.*fy; %Multiply Impulse response in Freq domain*FFT of y
signal

subplot(5,2,6) ;
%plot(con);
plot(xfft,abs(mul/max(mul)));
title(' ***** Filtered signal: High Pass *****');

```

```

% BAND PASS FILTER
wn = [wp ws];
b3 = fir1(n,wn,win);
[h,w] = freqz(b3,1,256);

fh_bp=fft(b3,nfft2); %Transform impulse response from time domain to
frequency domain
fh_bp=fh_bp(1:nfft2/2); %Only LHS of the +ve frequency (X-axis/2)

m = 20*log(abs(h));
subplot(5,2,7) ;

plot(xfft,abs(fh_bp/max(fh_bp))); %Normalising Y-axis (magnitude) to
1
title('H(w)=Impulse response in Frquency Domain BP');

%plot (w/pi, m) ;
%ylabel('Gain in db----->');
%xlabel(' (c) Normalised Frequency----->');

%con=conv(y,b);

mul=fh_bp.*fy; %Multiply Impulse response in Freq domain*FFT of y
signal

subplot(5,2,8) ;
%plot(con);
plot(xfft,abs(mul/max(mul)));
title(' ***** Filtered signal: Band pass *****');

% BAND STOP FILTER
b4 = fir1 (n, wn, 'stop' , win) ;
[h,w] = freqz(b4,1,256);

fh_bs=fft(b4,nfft2); %Transform impulse response from time domain to
frequency domain
fh_bs=fh_bs(1:nfft2/2); %Only LHS of the +ve frequency (X-axis/2)

m = 20*log(abs(h));
subplot(5,2,9) ;

plot(xfft,abs(fh_bs/max(fh_bs))); %Normalising Y-axis (magnitude) to
1
title('H(w)=Impulse response in Frquency Domain BS');

%plot (w/pi, m) ;
%ylabel('Gain in db----->');
%xlabel(' (d) Normalised Frequency----->');

```

```
%con=conv(y,b);

mul=fh_bs.*fy; %Multiply Impulse response in Freq domain*FFT of y
signal

subplot(5,2,10) ;
%plot(con);
plot(xfft,abs(mul/max(mul)));
title(' ***** Filtered signal: Band Stop *****');
```