Assignment: neural signal processing

Bm18mtech11005|pradeep kumar yadav

Generation of sine and cosine signal

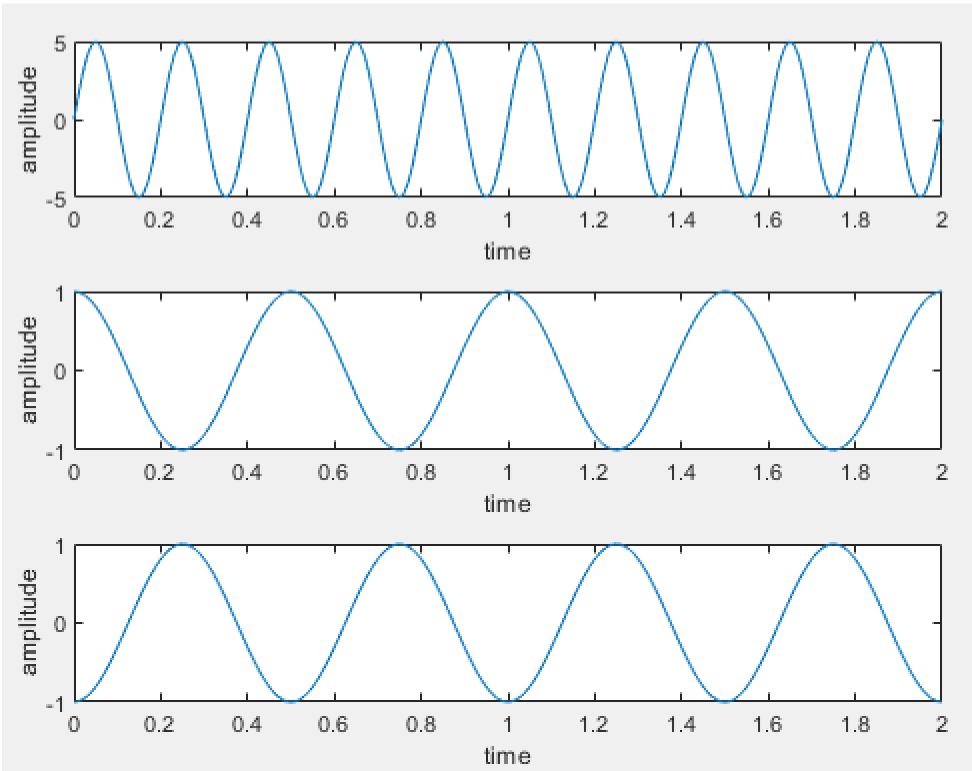
Matlab Code:

Fs=100; %sampling frequency

```
t=0:0.01:2; %time in sec(0.01=sampling time=1/fs),where fs is sampling
frequency
a= 5; %amplitude
f=5;  %frequency
y=a*sin(2*pi*f*t); %sinosoidal signal
subplot(3,1,1)


plot(t,y)

xlabel('time')
ylabel('amplitude')


t=0:0.01:2;
a= 1;
f=2;
y=a*cos(2*pi*f*t); %cosine signal
subplot(3,1,2)
plot(t,y)
xlabel('time')
ylabel('amplitude')
y=a*cos(2*pi*f*t-pi); %phase shift of pi
subplot(3,1,3);
plot(t,y)
xlabel('time')
ylabel('amplitude')
figure
spectrogram(y,'yaxis')
```

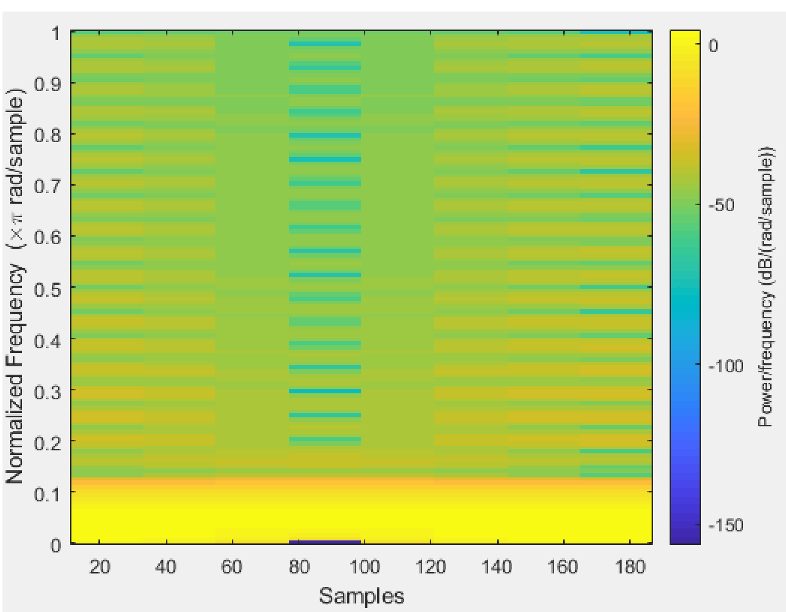
results:

magnitude and time plot for single frequency signal:



Spectrogram of cosine signal:

Single frequency component so single power distribution



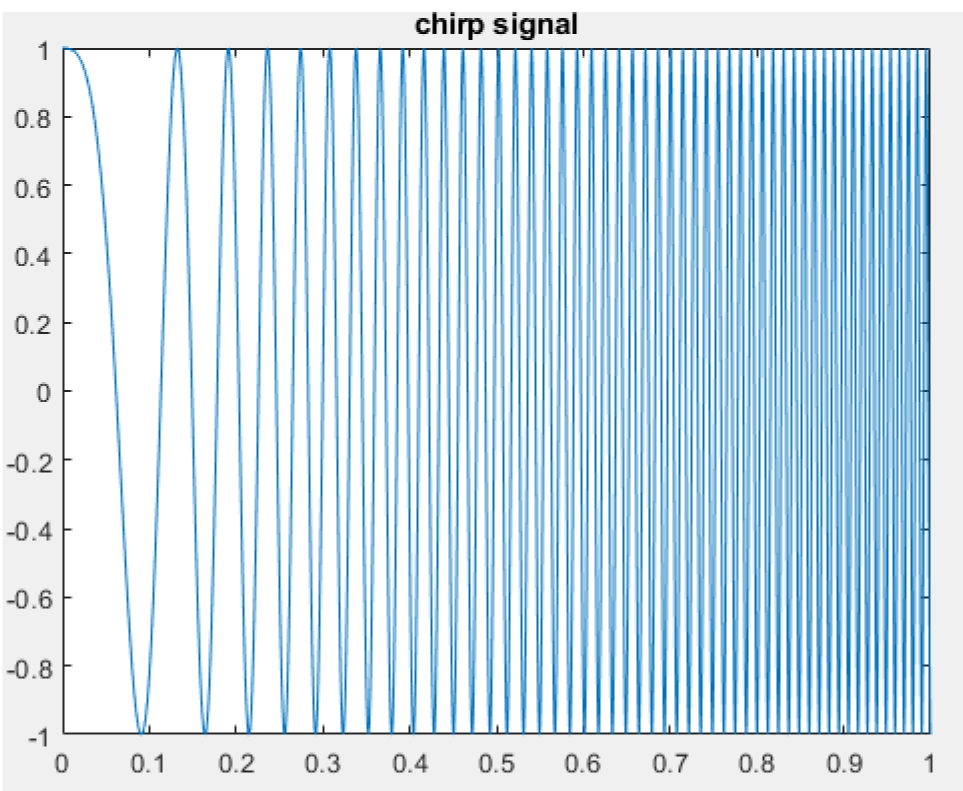
Chirp signal:

Matlab code:

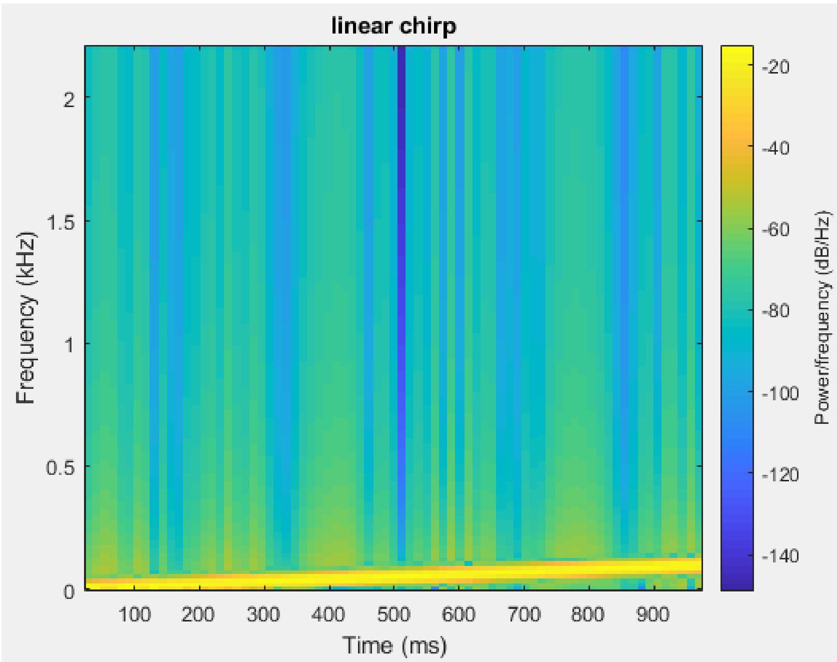
```
 Fs = 4410;                              % Sampling Frequency (Hz)
Tmax = 1                                % Duration (sec)
t = linspace(0, Tmax, Tmax*Fs);
f0 =1; %initial instantanious  frequency
f1 = 100;%final frequency
x = chirp(t,f0,t(end),f1,'linear');

figure(1)
plot(t,x)
title('chirp signal')
figure(2)
spectrogram(x,256,200,256,Fs,'yaxis')%windowsize=250 and 200=numberof overlap
title('linear chirp')
```

results:

linear chirp signal with increasing frequency



Spectrogram of linear chirp signal:

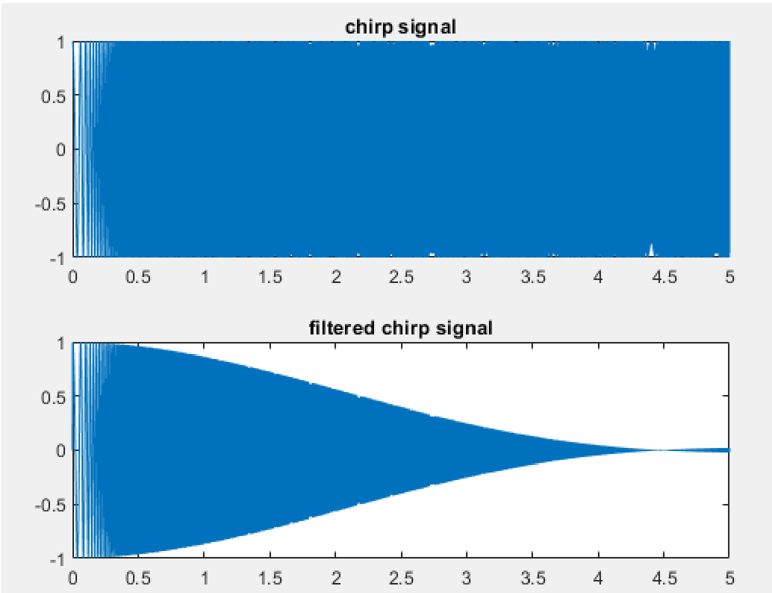


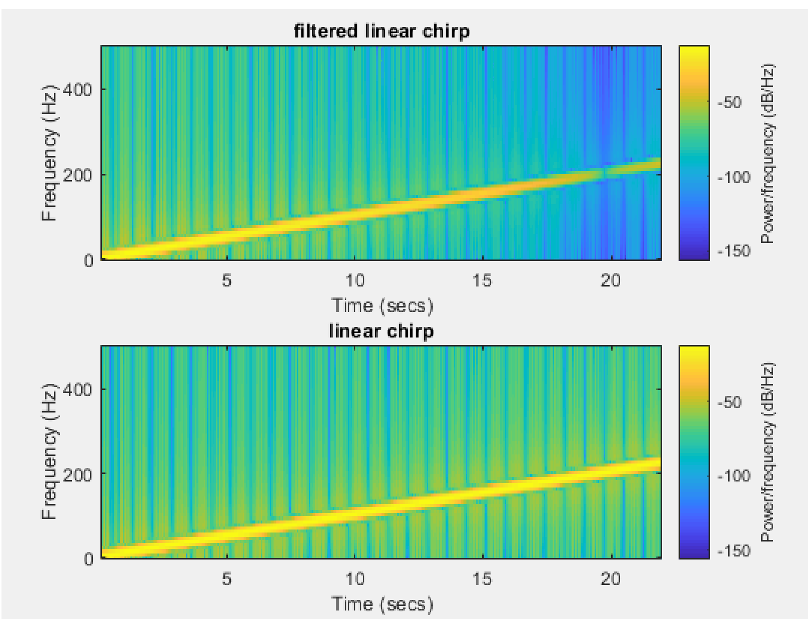
I have desigined a low pass hanning window filter in fdatool export that filter in matlab file and aplied on chirp signal.

```
Fs = 4410;                                      % Sampling Frequency (Hz)
Tmax = 5;                                        % Duration (sec)
t = linspace(0, Tmax, Tmax*Fs);
f0 =10; %initial instantanious  frequency
f1 = 1000;%final frequency
x = chirp(t,f0,t(end),f1,'linear');
sound(x,Fs)
windowsize=256;
filteredsignal= filter(lpf,x);
figure(1);
subplot(2,1,1)
plot(t,x)
title('chirp signal')
subplot(2,1,2)
plot(t,filteredsignal)
title('filtered chirp signal')
figure(2);
subplot(2,1,1)
spectrogram(filteredsignal,128,120,128,1e3,'yaxis')
title('filtered linear chirp')
subplot(2,1,2)
spectrogram(x,128,120,128,1e3,'yaxis')
```

```
title('linear chirp');
```

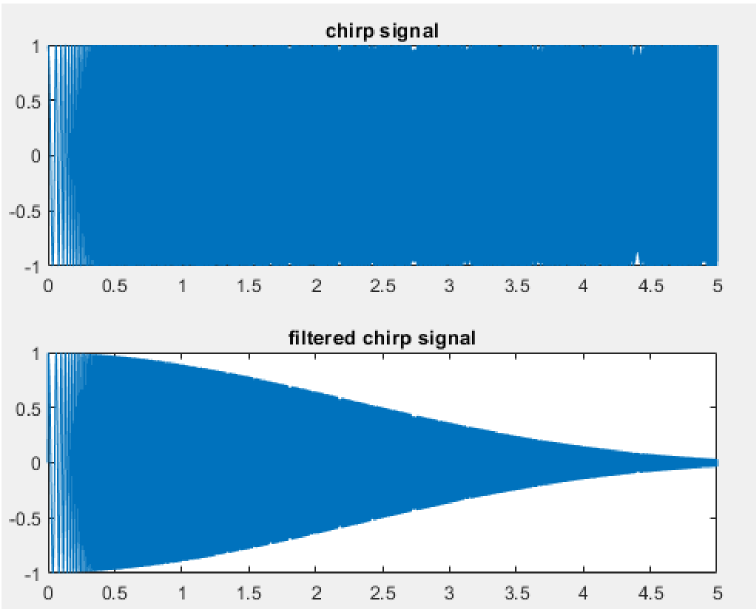
results

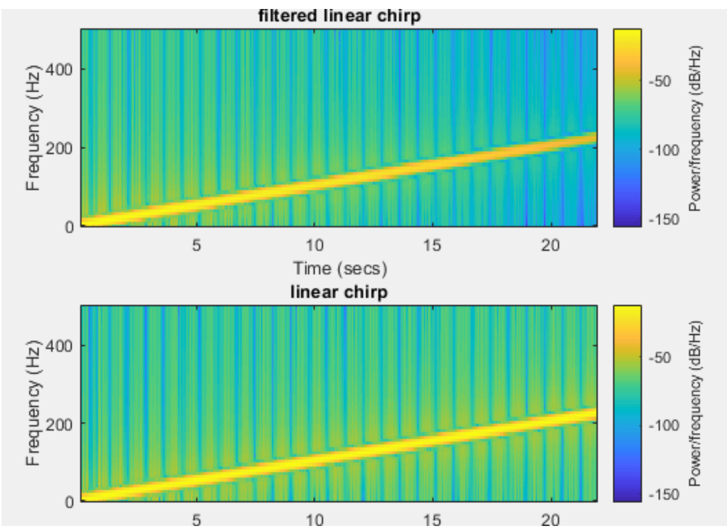




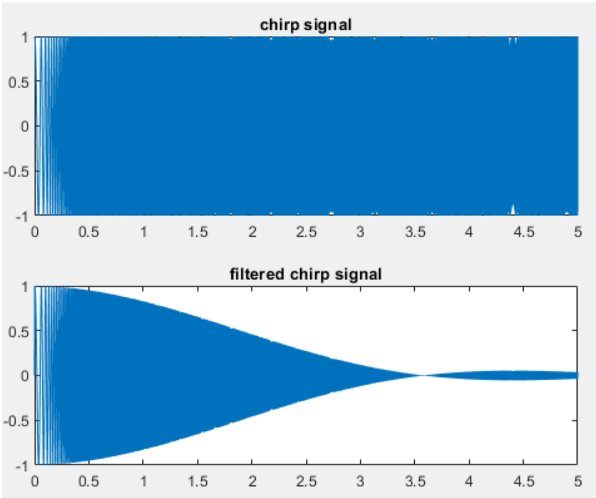
After filtering the noises reduced and frequency resolution improved.

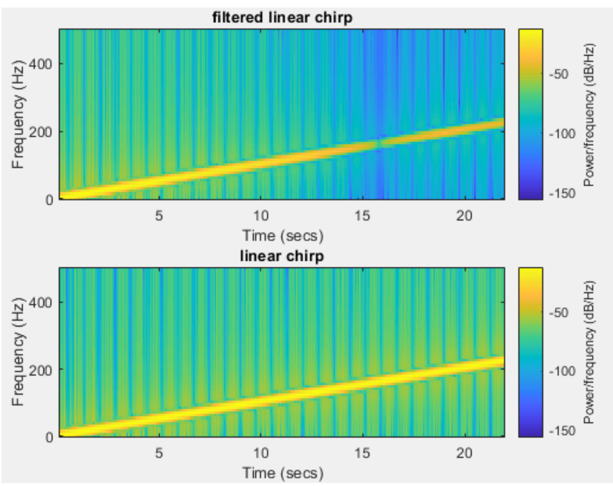
I have desigined a low pass hanning window filter in fdatool export that filter in matlab file and aplied on chirp signal.





- I have desigined a low pass rectangular window filter in fdatool export that filter in matlab file and aplied on chirp signal.





Conclution-

- The first side lobe of the Hamming is lower (i.e**. Hamming is better**) than the first side lobe of the Hanning,
- but the "distant" side lobes of the Hanning are lower than the **Hamming** (thus the Hanning is better in that regard)
- The Hamming window is preferred by many due to its relatively narrow main lobe width and good attenuation of the first few side lobes.

- The rectangular window has minimal side lobe attenuation, which is why it is a poor choiceand it has no taper.

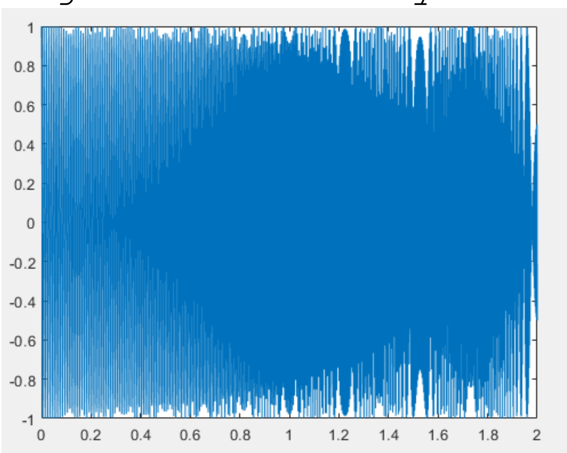
Quadratic chirp signal:

Matlab code:

```
fs = 1000;%sampling frequency
t = 0:1/fs:2;
x = chirp(t,100,1,200,'quadratic');
figure(1)
plot(t,x)
figure(2)
window=hamminh
spectrogram(x,256,120,256,fs,'yaxis')%n=128:size of window like
hamming,hanning,backman etc
%256=n  window size,120(nooverlap)
%[s,w,t] = spectrogram(x,window,noverlap,w)
%[s,f,t] = spectrogram(x,window,noverlap,f,fs)
title('Quadratic Chirp')
```

results:

quadratic chirp signal :
magnitude is on y axis and time is on x axis



Spectrum of quadratic chirp signal:

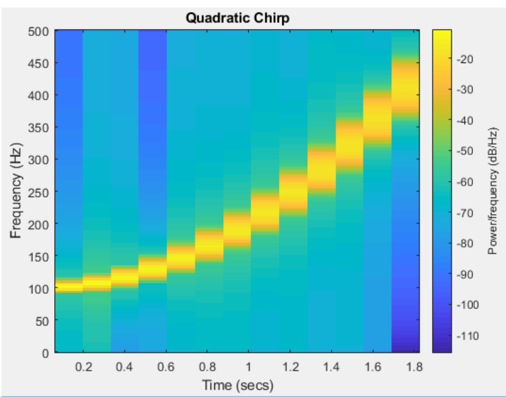
Frequency resolution is not good due to small sampling frequency:

Fs=1000 hz

NOTE:- sampling frequency nothing but nyquist frequency.

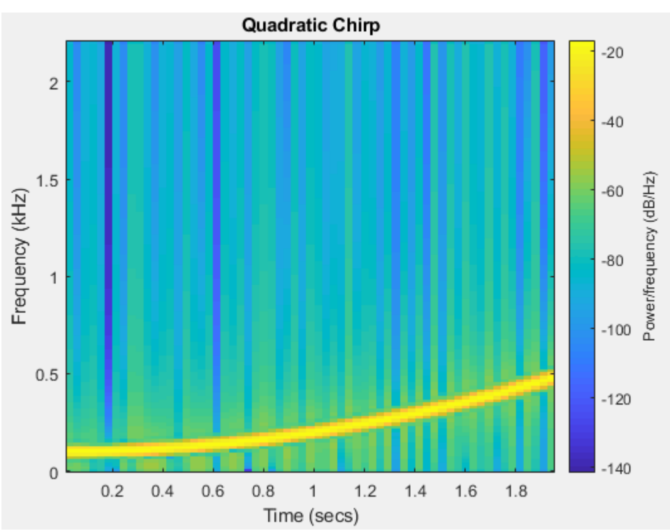
But fs >=2*fm where fm is maximum frequency of signal

To reconstruct the signal the above condition must be satisfied.



Now due to increasing sampling frequecncy the resolution is improved:

Fs=4410 hz



Note:-1- we can improve the frequency resolution with the help of three parameter(window size ,type of window and sampling frequency)

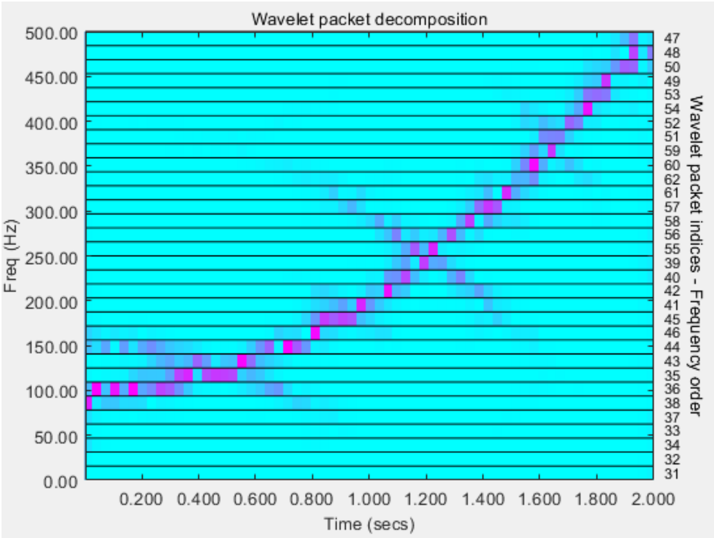
2:-resolution improves in frequency domain i.e. decreased in time domain.

Wavelet decompotion:

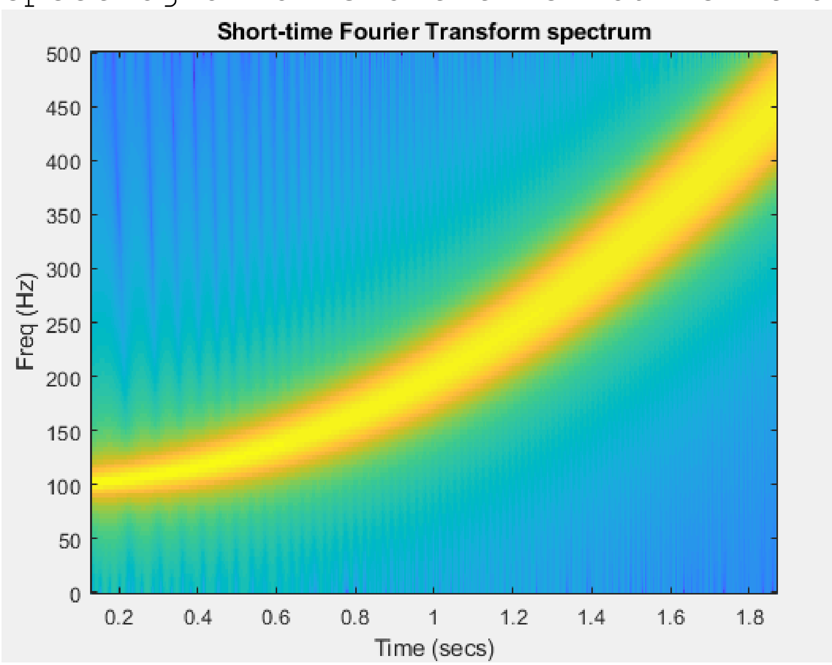
Matlab code:

```
fs = 1000; % sampling rate
t = 0:1/fs:2; % 2 secs at 1kHz sample rate
y = chirp(t,100,1,200,'quadratic');
plot(t,y)
level=5;

wpt = wpdec(y,level,'sym8');

[S,T,F] = wpspectrum(wpt,fs,'plot');
figure(2);
windowsize =256;
window = hann(windowsize);
nfft = windowsize;
noverlap = windowsize-1;
[S,F,T] = spectrogram(y,window,noverlap,nfft,fs);
imagesc(T,F,log10(abs(S)))
set(gca,'YDir','Normal')
xlabel('Time (secs)')
ylabel('Freq (Hz)')
title('Short-time Fourier Transform spectrum')
```

```
results:
wavelet decomposition in five level
```



Spectrogram of short time fourier transform

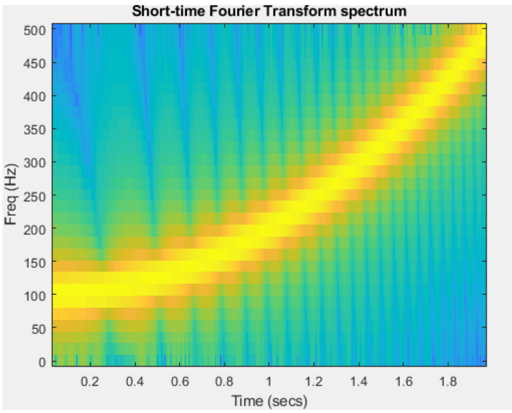


Note:-Frequency resolution is not that much good So to improve frequency resolution we have to change either type of window or size of window or sampling frequency.

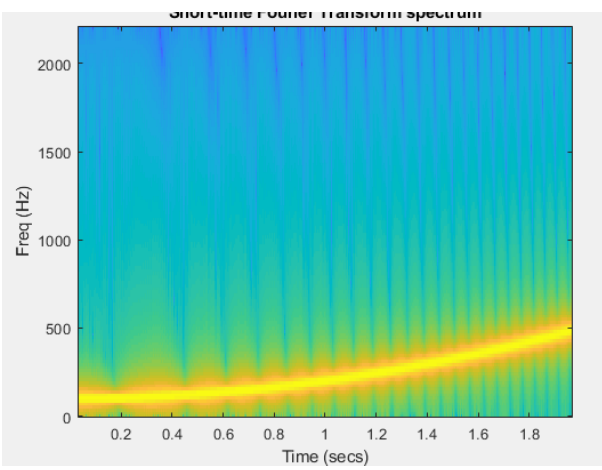
```
windowsize =56;
window = hann(windowsize);
nfft = windowsize;
noverlap = windowsize-1;
[S,F,T] = spectrogram(y,window,noverlap,nfft,fs);
imagesc(T,F,log10(abs(S)))
set(gca,'YDir','Normal')
```

```
xlabel('Time (secs)')
ylabel('Freq (Hz)')
title('Short-time Fourier Transform spectrum')
```

decreased window size



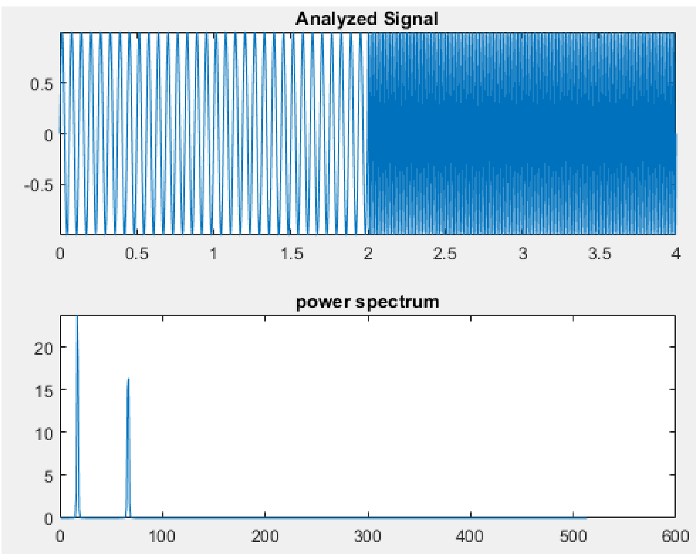
Increased sampling frequency

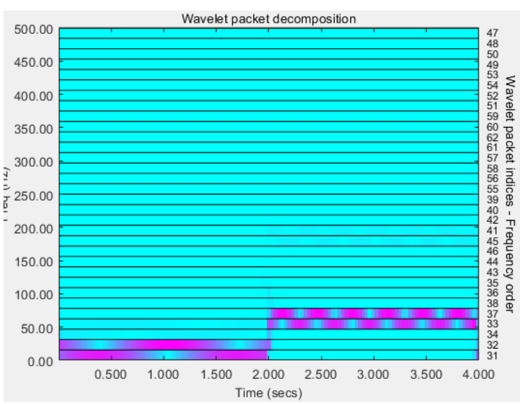


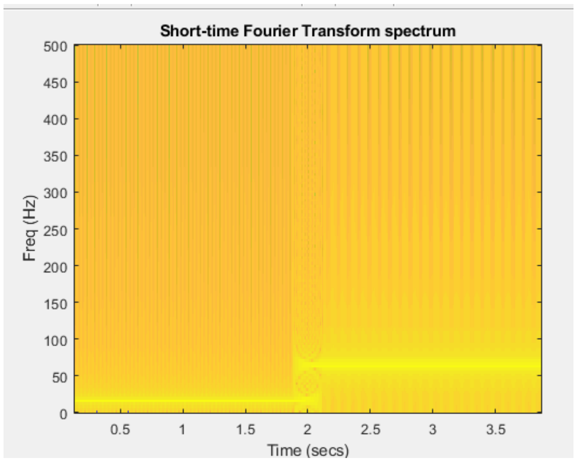

```
fs = 1000;
t = 0:1/fs:4;
y = sin(32*pi*t).*(t<2) + sin(128*pi*t).*(t>=2);
subplot(2,1,1)
plot(t,y);
axis tight
title('Analyzed Signal');
x=pwelch(y);
subplot(2,1,2)
plot(x)
title('power spectrum');
level =5 ;
wpt = wpdec(y,level,'sym6');%wavelet decomposition
figure(2);
[S,T,F] = wpspectrum(wpt,fs,'plot');
windowsize=256;
window = rectwin(windowsize);
```

```
nfft = windowsize;
noverlap = windowsize-1;
[S,F,T] = spectrogram(y,window,noverlap,nfft,fs);
figure(3);
imagesc(T,F,log10(abs(S)))%log10 value gives the magnitude response of
spectrum in db
set(gca,'YDir','Normal')
xlabel('Time (secs)')
ylabel('Freq (Hz)')
title('Short-time Fourier Transform spectrum')
```

results:







The above figure shows that the signal containing two frequency so we can clearly see that the two spectrum of stft is formed.

I have also applied pwelch function which gives two component of power spectrum and I have seen the two impulse it is clearly seems that signal have two frequency component.

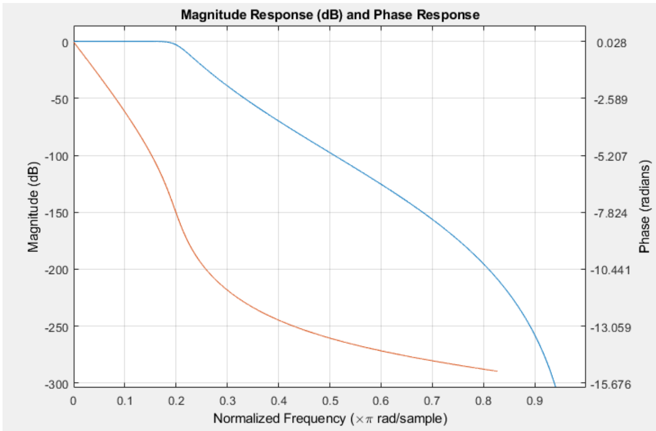
Butterworth filter:

Low pass bw filter:

Matlab code:

```
fc = 100;
fs = 1000;

[b,a] = butter(10,fc/(fs/2));%b and a are trasfer fuction coeficient
fvtool(b,a)
%dataIn = randn(1000,1);
%dataOut = filter(b,a,dataIn);
```

Results:



As we can see the -3db cut off frequency is 0.2 means 100 hz

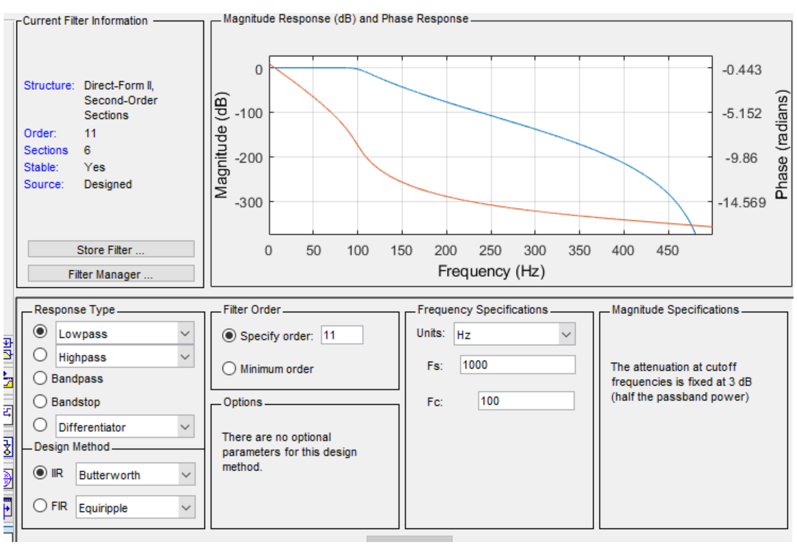
Wc=2*pi*fc=200*pi;

Normalized frequency is(w/fs) that is x axis

Where is w is frequency in radian ,clealy we can see the at 0.2 normalised frequency the -3d cut off.

Note- fvtool fuction gives the magnitude ,phase,impulse,unit impulse and estimated magnitude responses.

I have designed this low pass filter with fdatool.



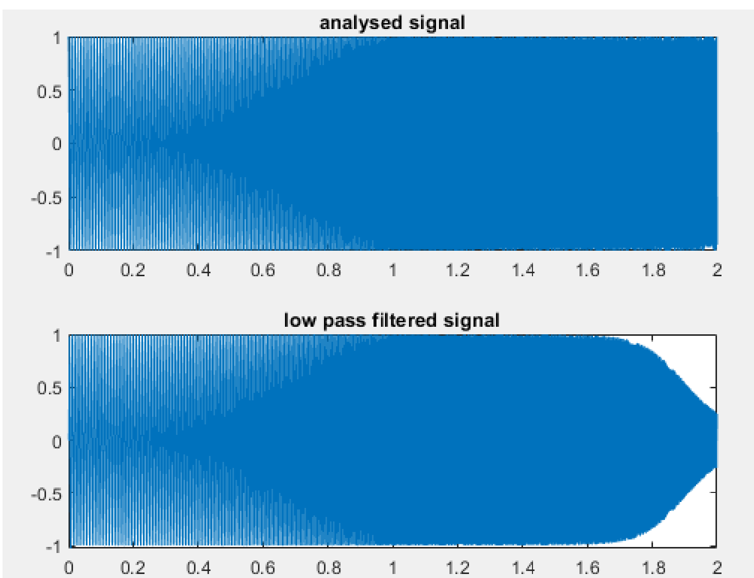
As order of filter increases the roll of facter reduced.

Roll of factor is nothing but excessive band width

I have applied the butterworth low pass filter on chirp signal.

```
fs = 4410;%sampling frequency
t = 0:1/fs:2;
x = chirp(t,100,1,200,'quadratic');
subplot(2,1,1)
plot(t,x)
title('analysed signal')
filteredsignal=filter(blpf,x);
subplot(2,1,2);
plot(t,filteredsignal);
title('low pass filtered signal')
```

results:



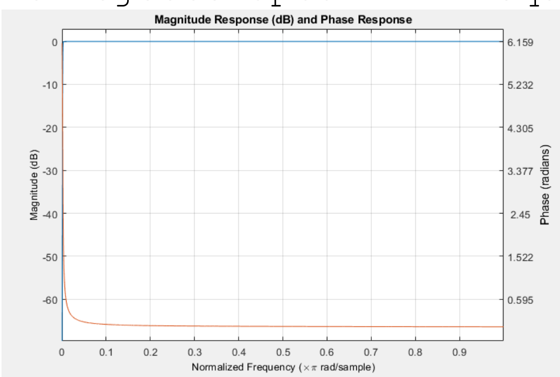
Butterworth high pass filter:

Matlab code:

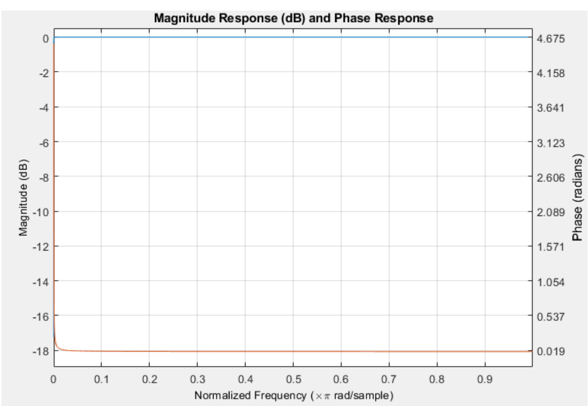
```
fc=1;%cut off frequency
fs=1000;
[z,p,k] = butter(4,fc/500,'high');%zeroes,pole and gain(k)
sos = zp2sos(z,p,k);
fvtool(sos,'Analysis','freq')
```

results

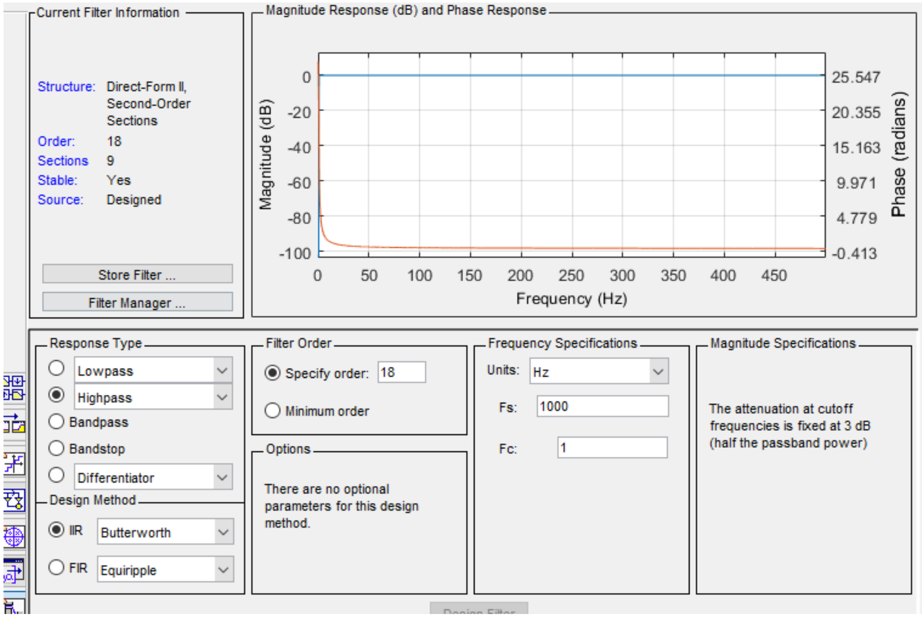
we can see the cut off frequency clearly at -3db
it rejects upto 1hz frequency.



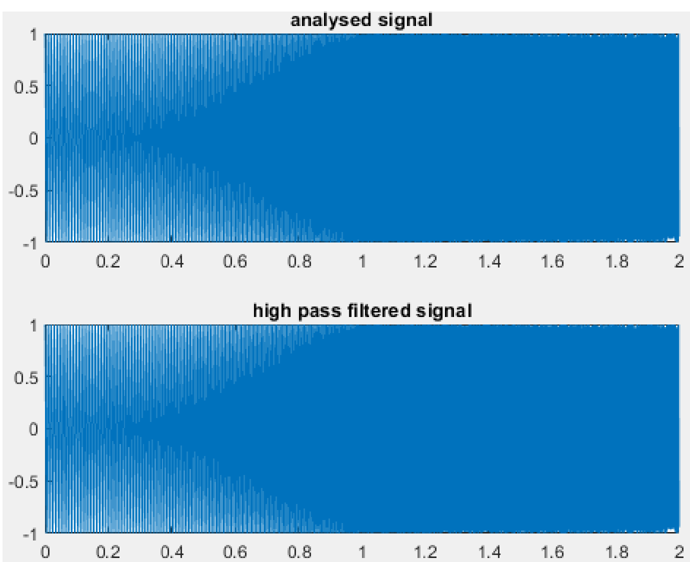
Response at 0.1hz cut off frequency.



I have designed this high pass filter with fdatool.




```
fs = 4410;%sampling frequency
t = 0:1/fs:2;
x = chirp(t,100,1,200,'quadratic');
subplot(2,1,1)
plot(t,x)
title('analysed signal')
filteredsignal=filter(bhpf,x);
subplot(2,1,2);
plot(t,filteredsignal);
title('high pass filtered signal')
```



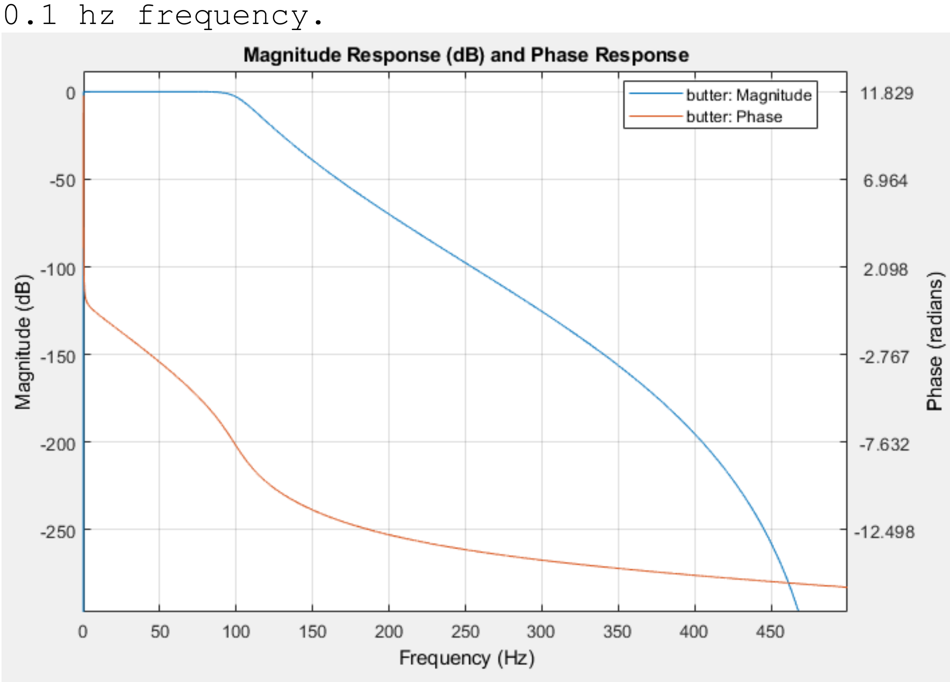
Butterworth band pass filter:

Matlab code:

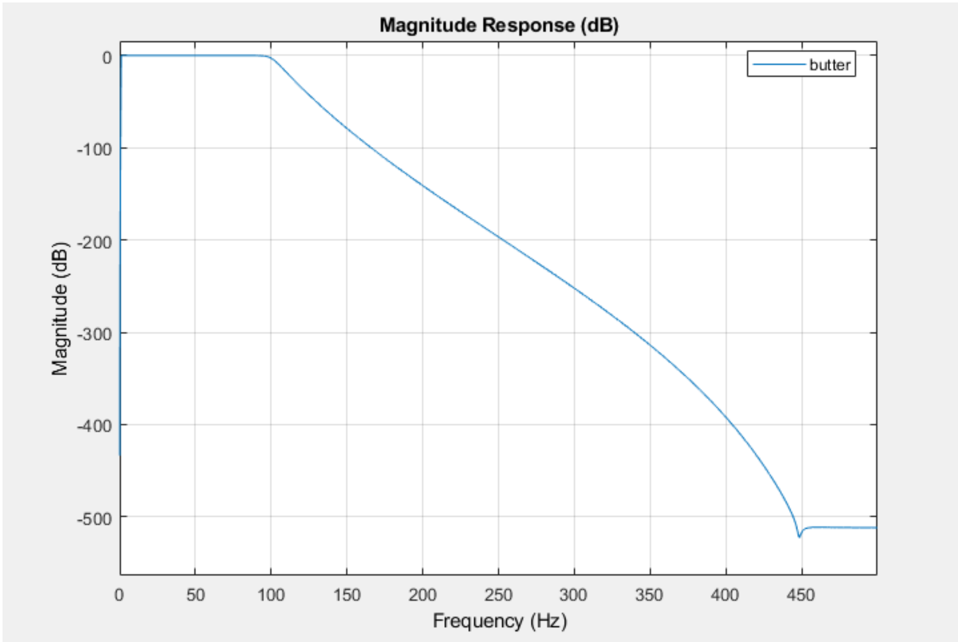
```
fc1=0.1,fc2=100;
fs=1000;

[A,B,C,D] = butter(10,[fc1 fc2]/500);
% d = designfilt('bandpassiir','FilterOrder',10, ...
%     'HalfPowerFrequency1',fc1,'HalfPowerFrequency2',fc2, ...
%     'SampleRate',1000);
sos = ss2sos(A,B,C,D);
fvt = fvtool(sos,'fs',1000);
legend(fvt,'butter')
```

results:

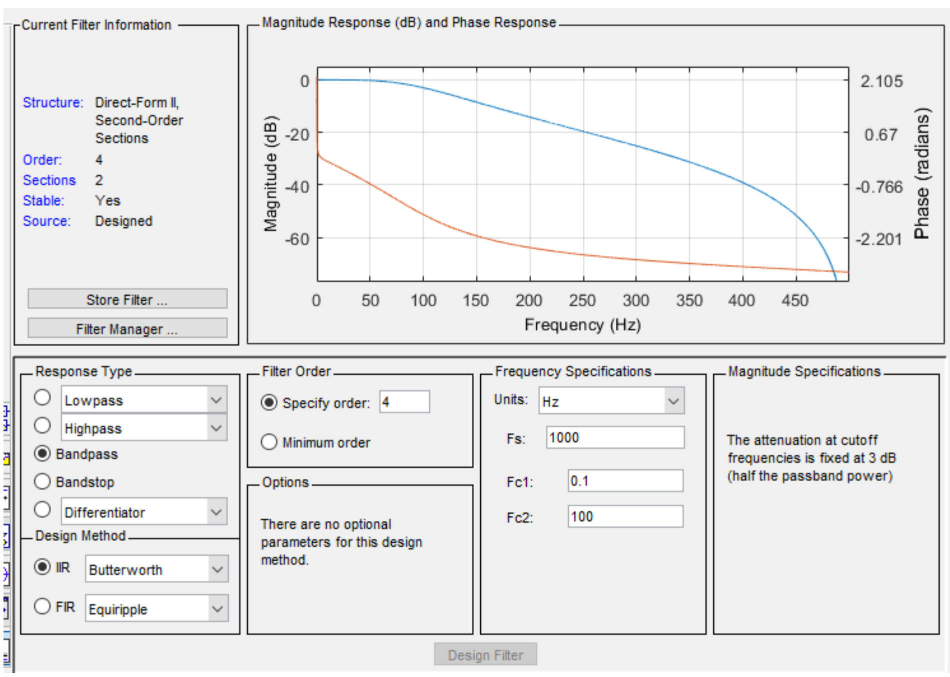
at cut off frequency is between 0.1 and 100 hz we can clearly see the magnitude and phase response.
It looks looks like a low pass filter but it rejecting the below 0.1 hz frequency.



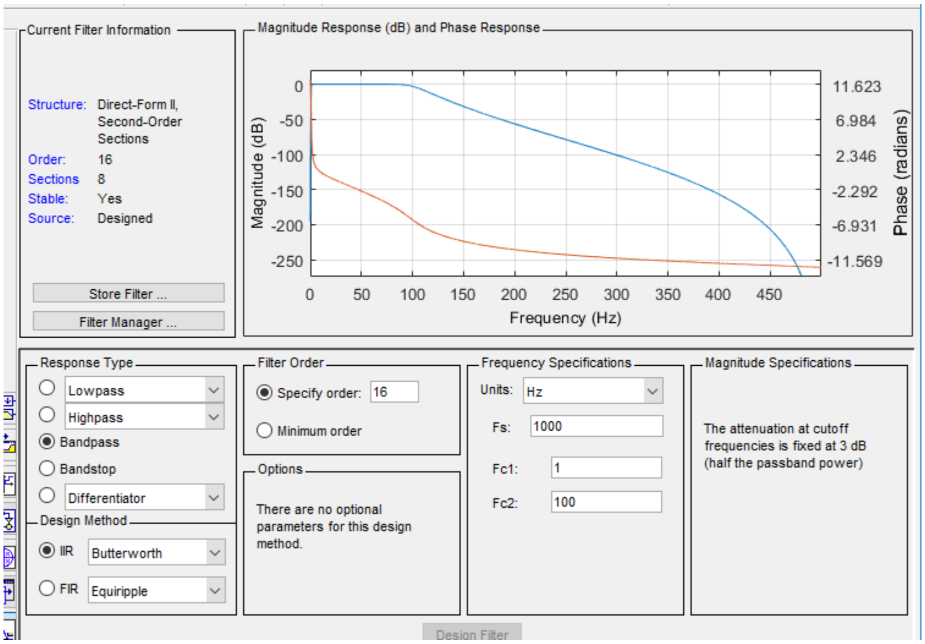
Cut off frequency is between 1 and 100hz and also I have increased the order of filter that reduce the roll of factor .it clearly differentiated in the both of the graph.

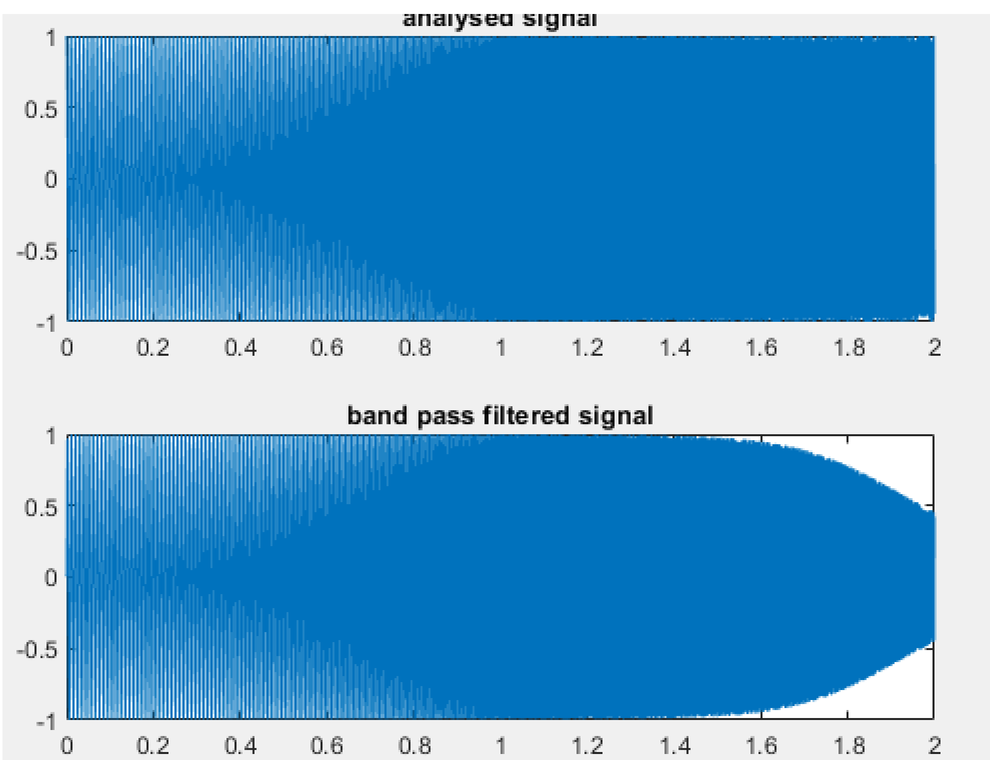


Designed with fdatool and at low order there are high roll of factor.



After increasing the order of filter it reduced the roll of facor





Butterworth band reject filter:

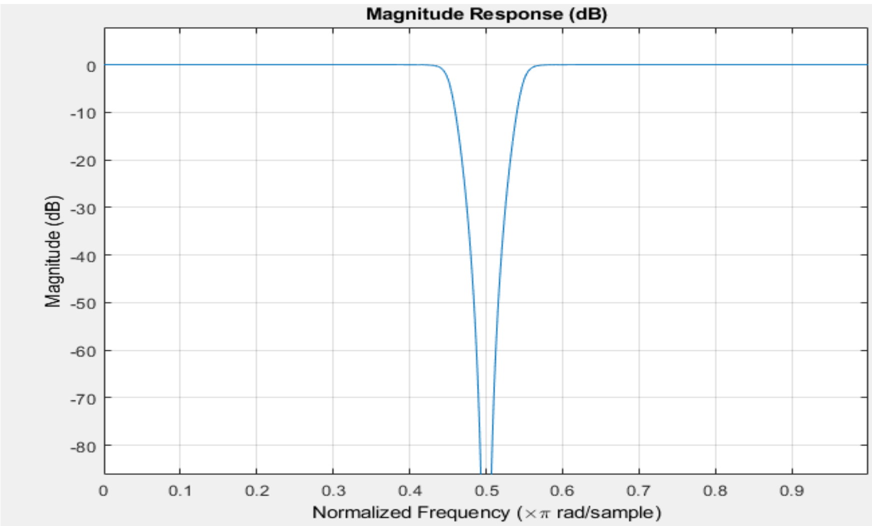
Matlab code:

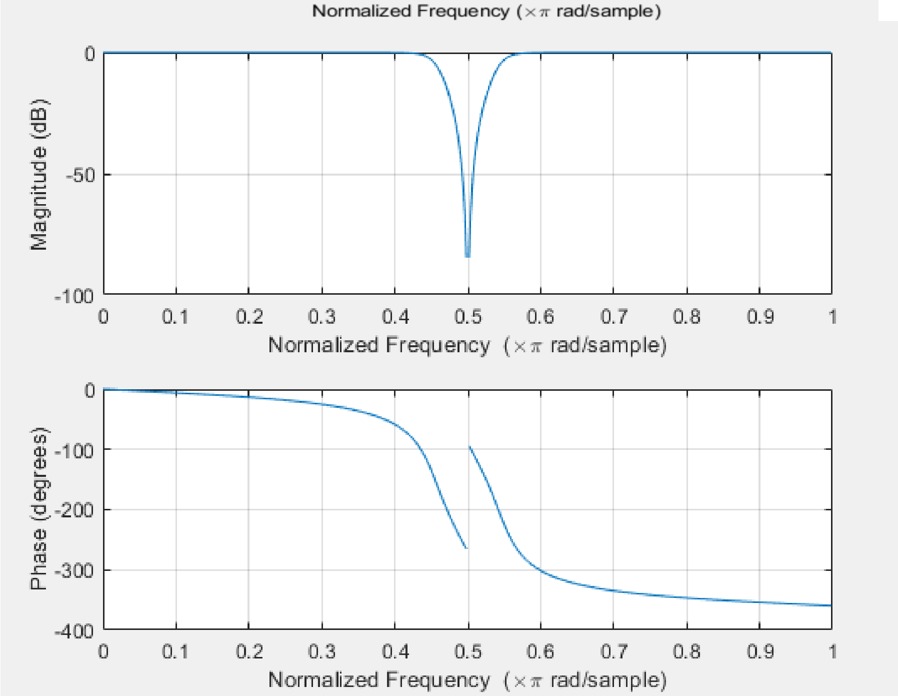
```
fc1=45,fc2=55;
```

```
fs=1000;
order=3;
[b,a] = butter(3,[0.45 0.55],'stop');
fvtool(b,a)
```

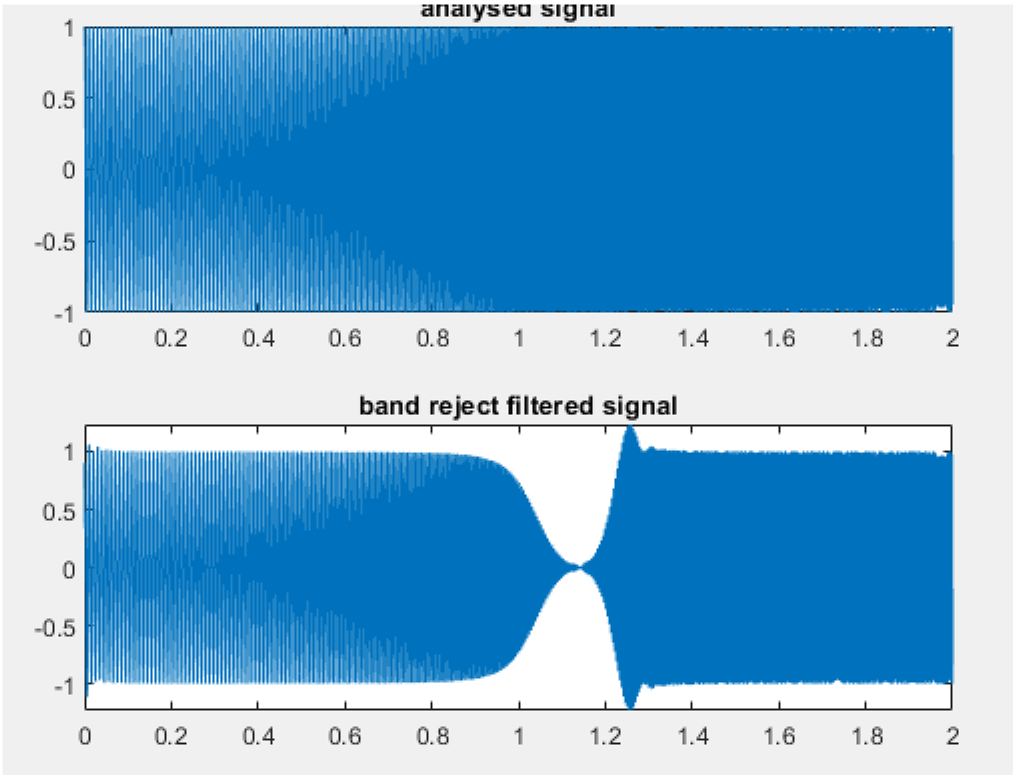
results:

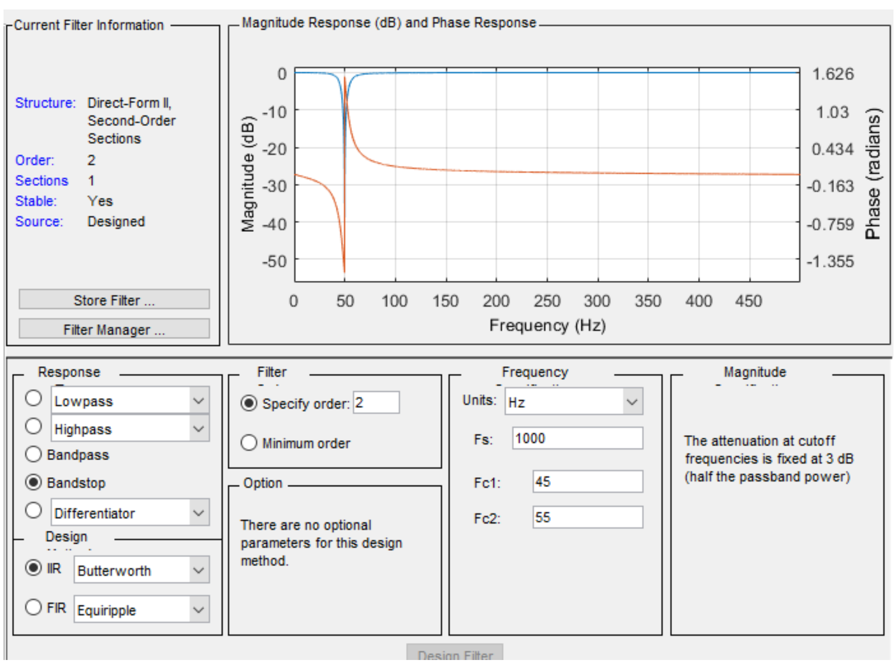
clearly we can see the stopping between 45 to 55hz.



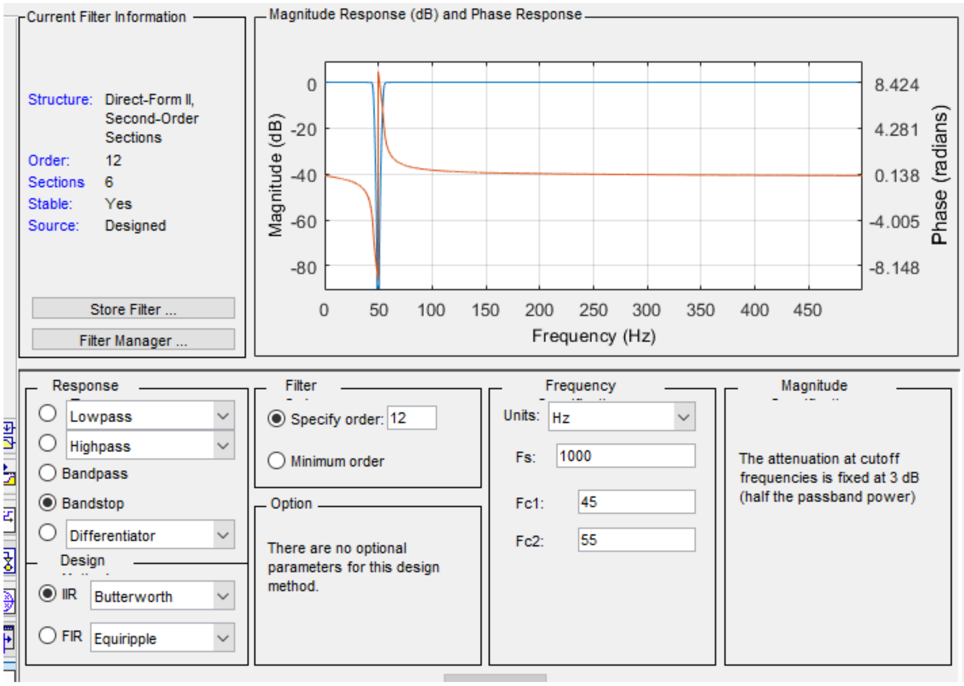


At second order band reject filter roll of factor is high





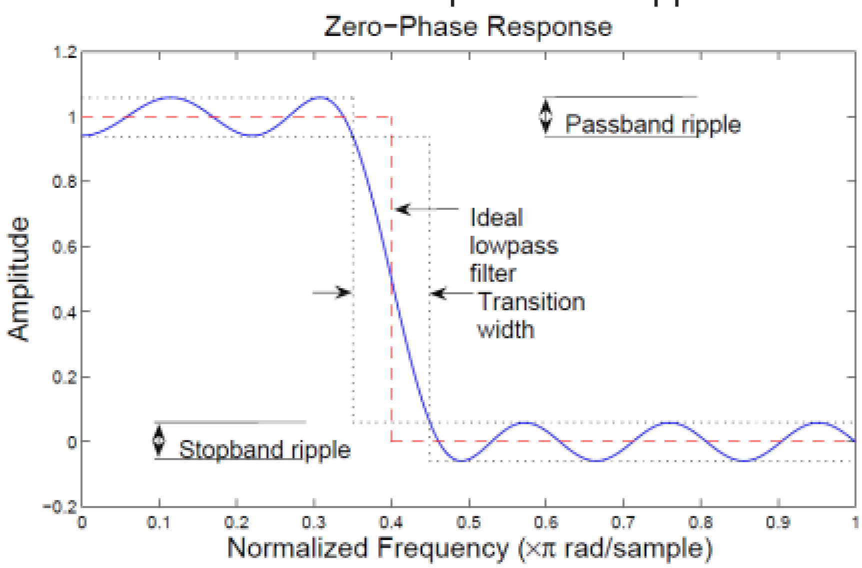
At order is increased roll of factor is decreased



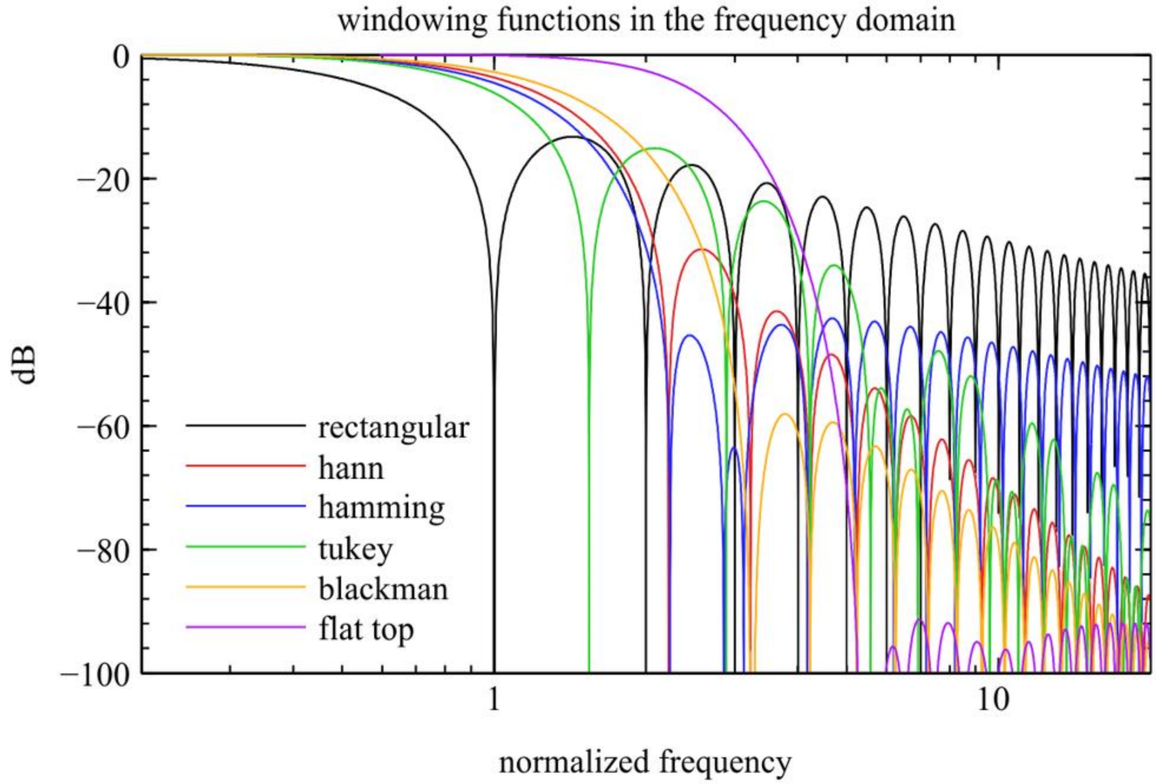
passband ripple is the amount of variation in the amplitude, within the designated passband of the filter, and stop band attenuation is the minimum attenuation level with the designated rejection band of the filter.

There is not be any ripple and attenuation in IIR filter.

But in FIR filter there is passband ripple and stop band attenuation.



FIR FILTERS-


windowing functions in the frequency domain
0
−20
−40
−60
−80
−100
dB
rectangular
hann
hamming
tukey
blackman
flat top
1
10
normalized frequency