1.2.1\_butterworth

Wp = 2000/6000;

Ws = 1500/6000;

[n,w] = buttord(Wp,Ws,0.5,60);

format short e;

[b,a]=butter(n,w,'high');

figure(1);

freqz(b,a,100,12000);

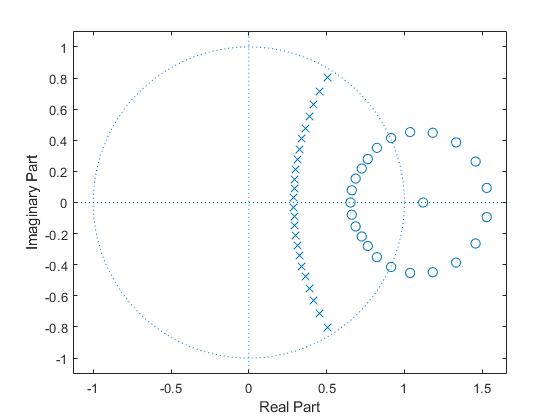
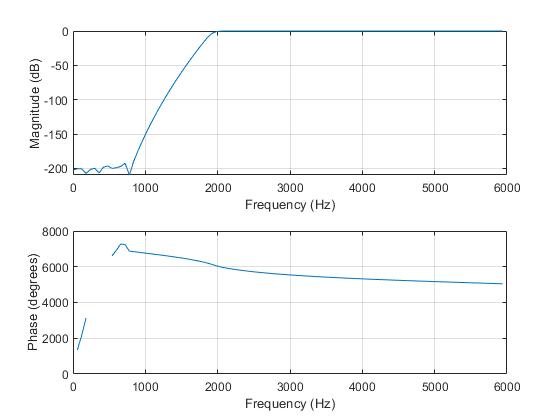
figure(2);

zplane(b,a);

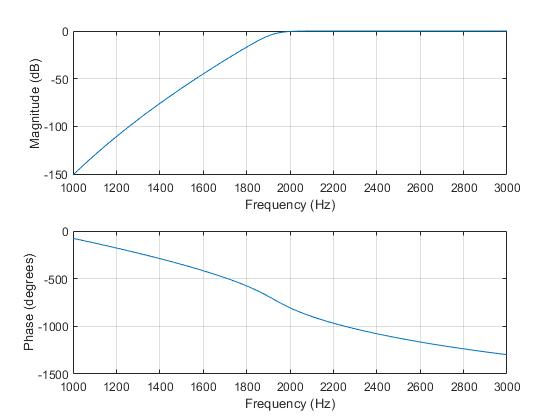
figure(3);

f=1000:2:3000;

freqz(b,a,f,12000);



1.2.1\_butterworth\_zplot



1.2.1\_butterworth\_transistin\_band

1.2.1\_Chebyshev Type I

Wp = 2000/3000;

Ws = 1500/3000;

Rp = 0.5;

Rs = 60;

[n,w] = cheb1ord(Wp,Ws,Rp,Rs);

format short e;

[b,a]=cheby1(n,Rp,w);

figure(1);

freqz(b,a,100,12000);

figure(2);

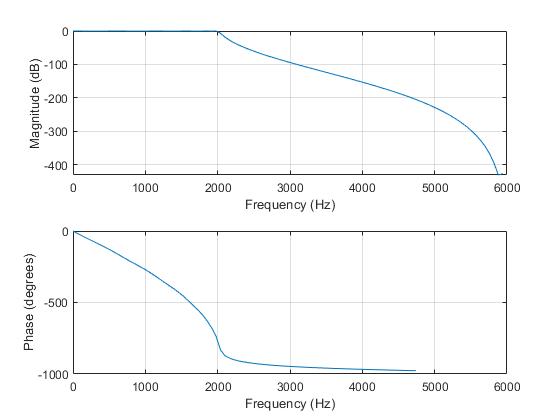
zplane(b,a);

figure(3);

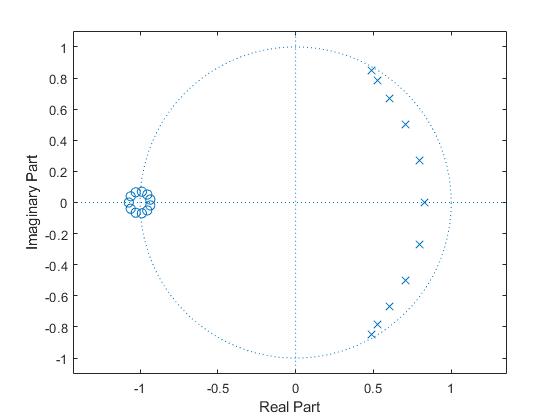
f=4000:2:6000;

freqz(b,a,f,12000);

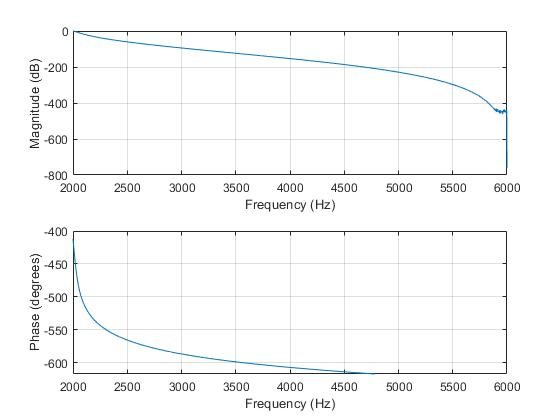
1.2.1\_Chebyshev Type I



1.2.1\_Chebyshev Type I\_zplot



1.2.1\_Chebyshev Type I\_transistion\_band



1.2.1\_ elliptic

Wp = 2000/3000;

Ws = 1500/3000;

Rp = 0.5;

Rs = 60;

[n,w] = ellipord(Wp,Ws,Rp,Rs);

format short e;

[b,a]=ellip(n,Rp,Rs,Wp);

figure(1);

freqz(b,a,100,12000);

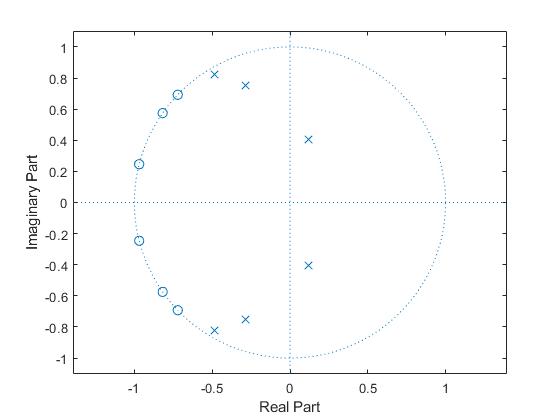
figure(2);

zplane(b,a);

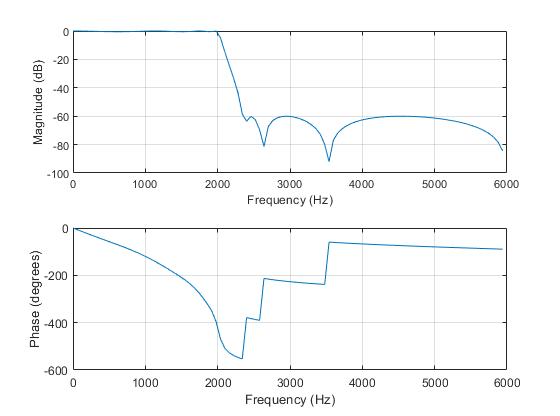
figure(3);

f=4000:2:6000;

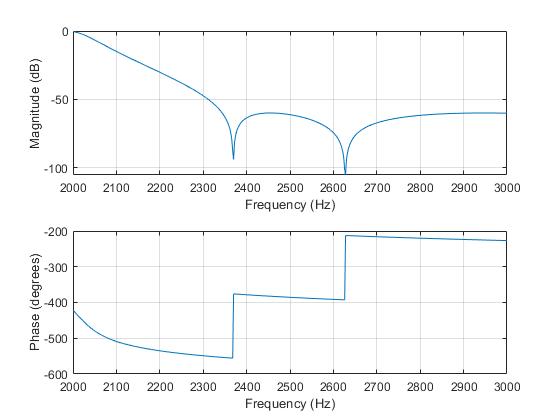
freqz(b,a,f,12000);



1.2.1\_ elliptic\_zplot



1.2.1\_ elliptic\_transition\_band



As we can see from those three kinds of filter,

The Butterworth filter is the flattest filter and its passband response offers the steepest roll-off without inducing a passband ripple. So Butterworth filter produce the most linear phase within the passband. However, elliptic filter produce the least linear phase within the passband.

Elliptic has the closest to the unit circle.

1.2.2

Wp =[2500 3500]/6000;

Ws = [2000 4000]/6000;

[n,w] = buttord(Wp,Ws,0.5,60);

format short e;

[b,a]=butter(n,w);

figure(1);

freqz(b,a,100,12000);

title(sprintf('n = %d Butterworth Lowpass Filter',n))

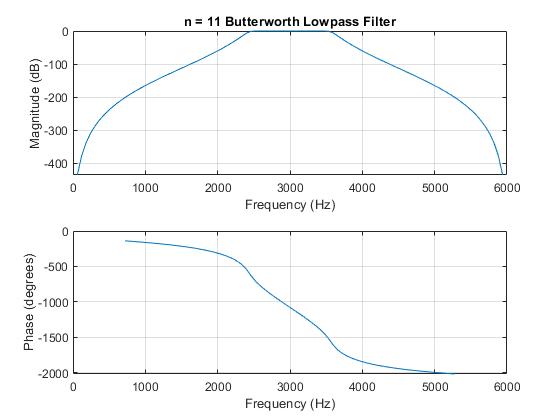
figure(2);

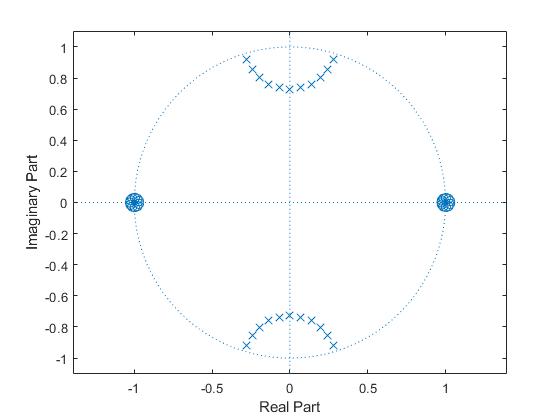
zplane(b,a);

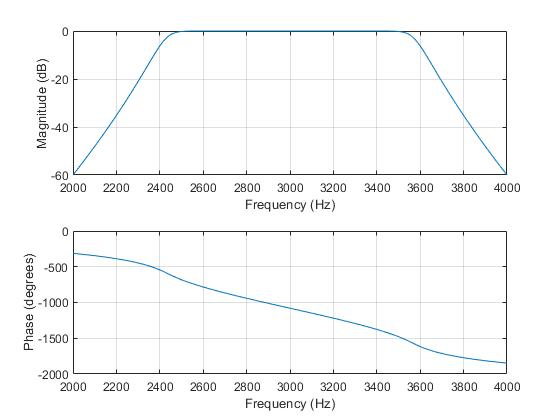
figure(3);

f=2000:2:4000;

freqz(b,a,f,12000);







1.2.3

Original graph

As we can see from the graph, I guess we can use a low-pass Butterworth filter to filter out the noise, which is

clearvars;

close all;

Fs = 16000;

Fn=Fs/2;

load('sh27.mat');

figure(1);

freqz(sh27);

Wp = [150 5800]/Fn;

Ws = [50 6100]/Fn;

Rp=1;

Rs=30;

[n,w] = cheb2ord(Wp,Ws,Rp,Rs);

format short e;

[b,a]=cheby2(n,Rs,Ws);

[SOS,G] = tf2sos(b,a);

figure(2);

freqz(SOS, 4096, Fs)

y=filtfilt(SOS,G,sh27);

t = 1:length(sh27);

figure(3);

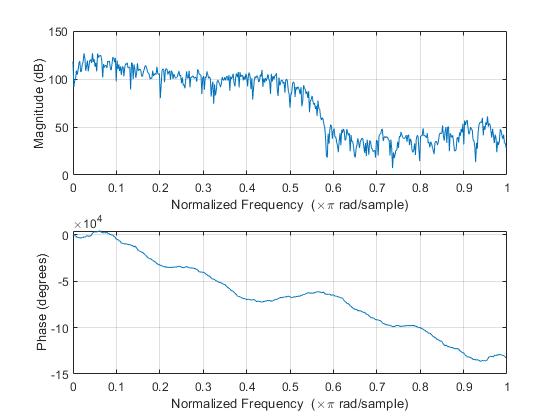
plot(t,sh27,'--',t,y,'-')

legend('Original Data','Filtered Data')

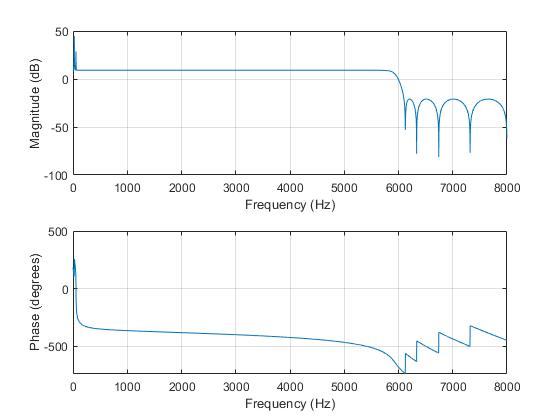
soundsc(y);

I found a way to include this signal into the filter, but I don’t know if it is the right way to include this sh27 into the filter. I still plot it. But I can’t hear a sentence in the voice,

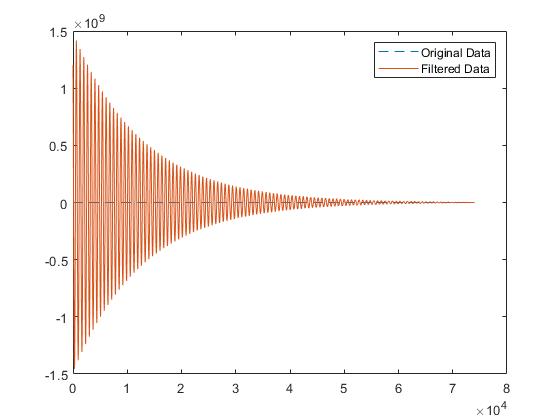
Original



Filter



Filtered signal



2.1.1

Because it require at least 60 db, so we can must use blackman window as FIR filter design

To calculate M = 9.79pi/0.1pi = 97.9 = 98 order， However, the order is not satisfying the requirement, so I try different order until 250, that can meet the requirement of -60 dB or more at and below 1,500 Hz

a=1;

figure(1);

b = fir1(250,0.32,'high');

g=0:1:6000;

freqz(b,a,g,12000)

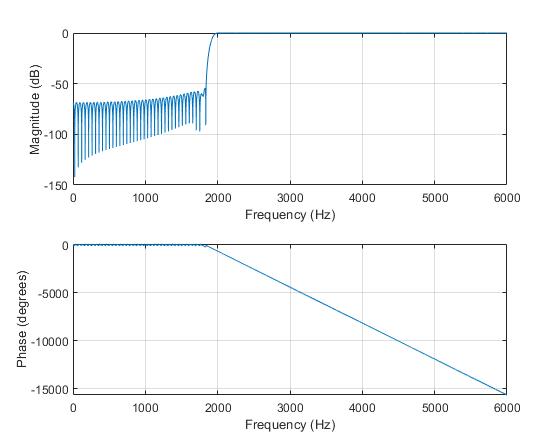
figure(2);

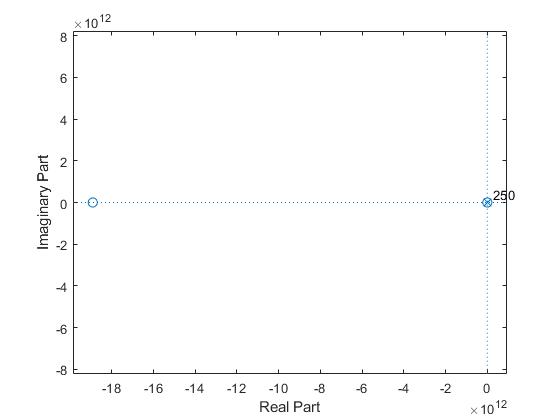
zplane(b,a);

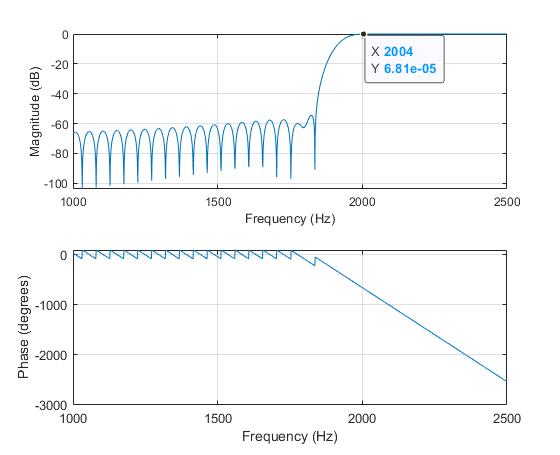
figure(3);

f=1000:1:2500;

freqz(b,a,f,12000);







2.1.2

a=1;

figure(1);

b = fir1(98,[4/14 8/14]);

g=0:1:7000;

freqz(b,a,g,14000)

figure(2);

zplane(b,a);

figure(3);

f=1000:1:5000;

freqz(b,a,f,14000);

