**1.Analysis of signal**

We choose the ***projsignal0.mat****,* from the table we get the .

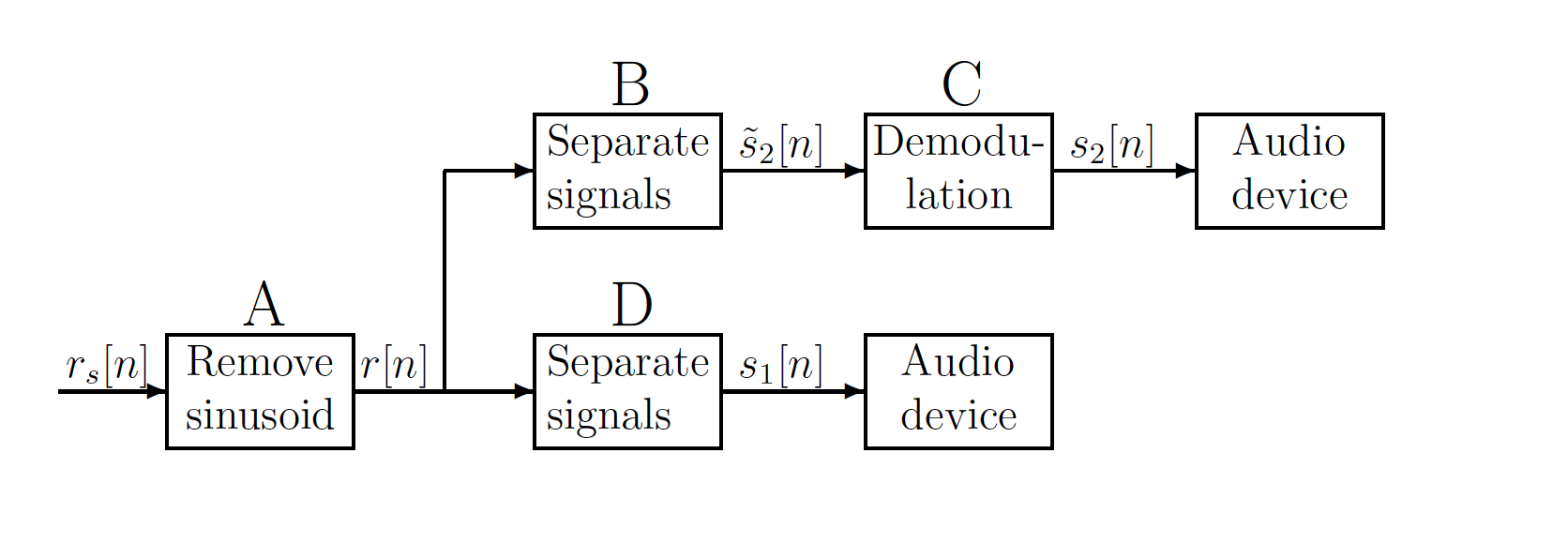
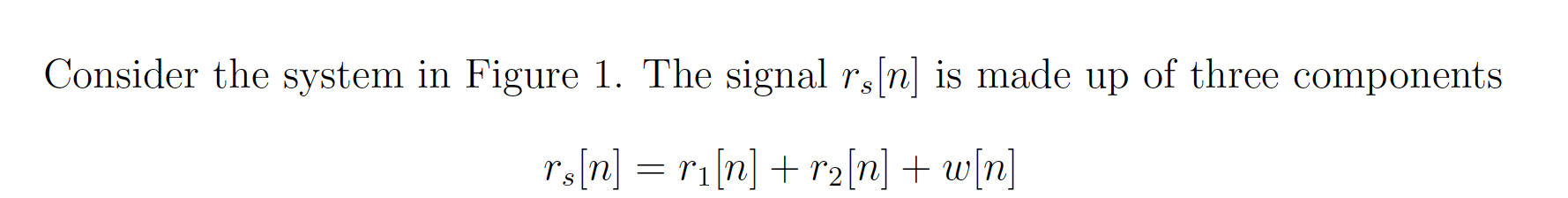


Figure1: Signal removal and separation



and are two parts of the signal, is the sinusoidal disturbance signal. has a bandwidth of 4096Hz and is a DSB-SC modulated signal. The carrier frequency is =12.288kHz, the bandwidth of the original signal is 4096Hz.

The function of Block A is to remove the sinusoidal disturbance signal. The Block D is a lowpass filter to get the low frequency part of the original signal. And the Block B is a highpass filter to get the high frequency part of the original signal. Block C has two functions, the first is to demodulate the output signal of Block B, and then use a lowpass filter to get the low frequency part of the signal.

After loading the ***projsignal0.mat,*** it can easily be observed that the notch frequency of the disturbance is 0.3π, which is almost 4915Hz in time domain.

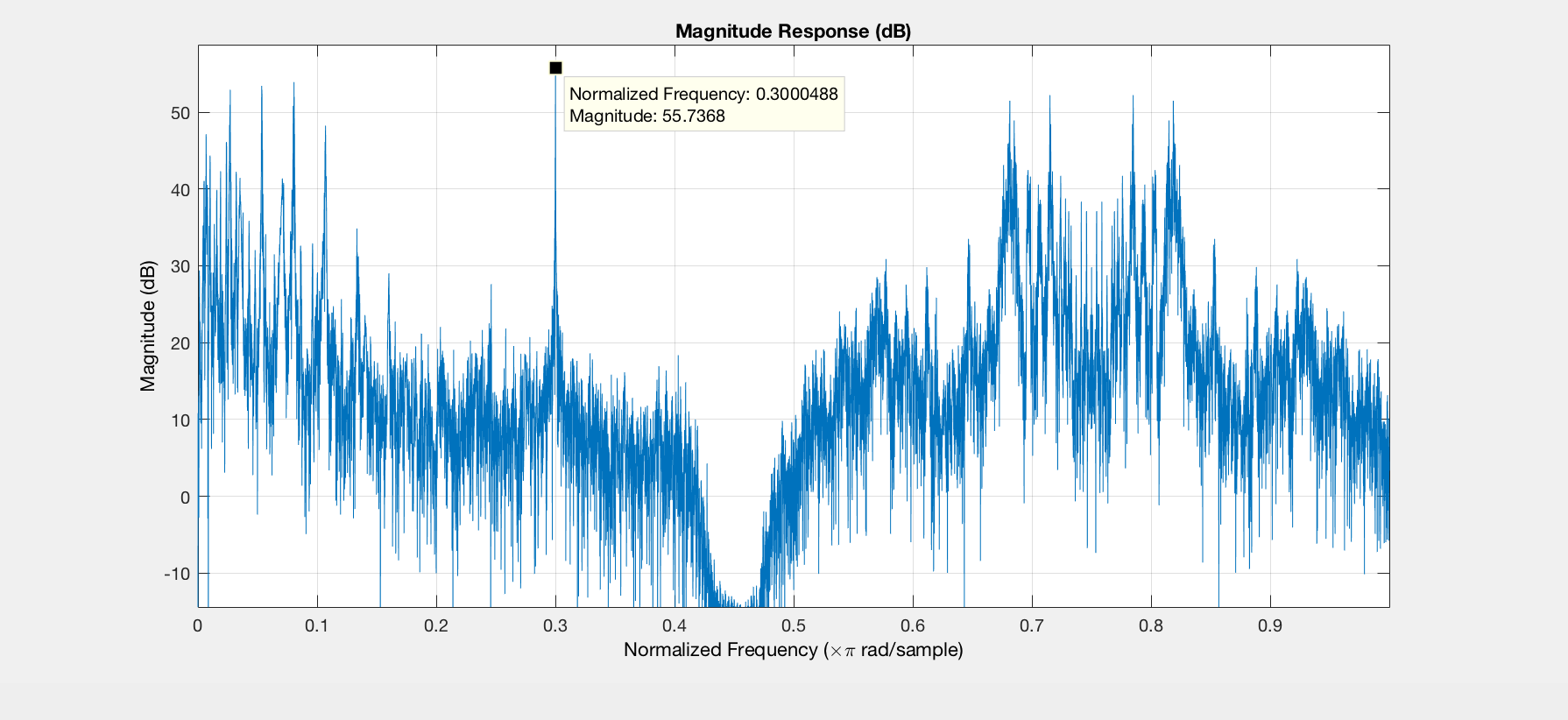


Figure2: the Magnitude Response of the original signal

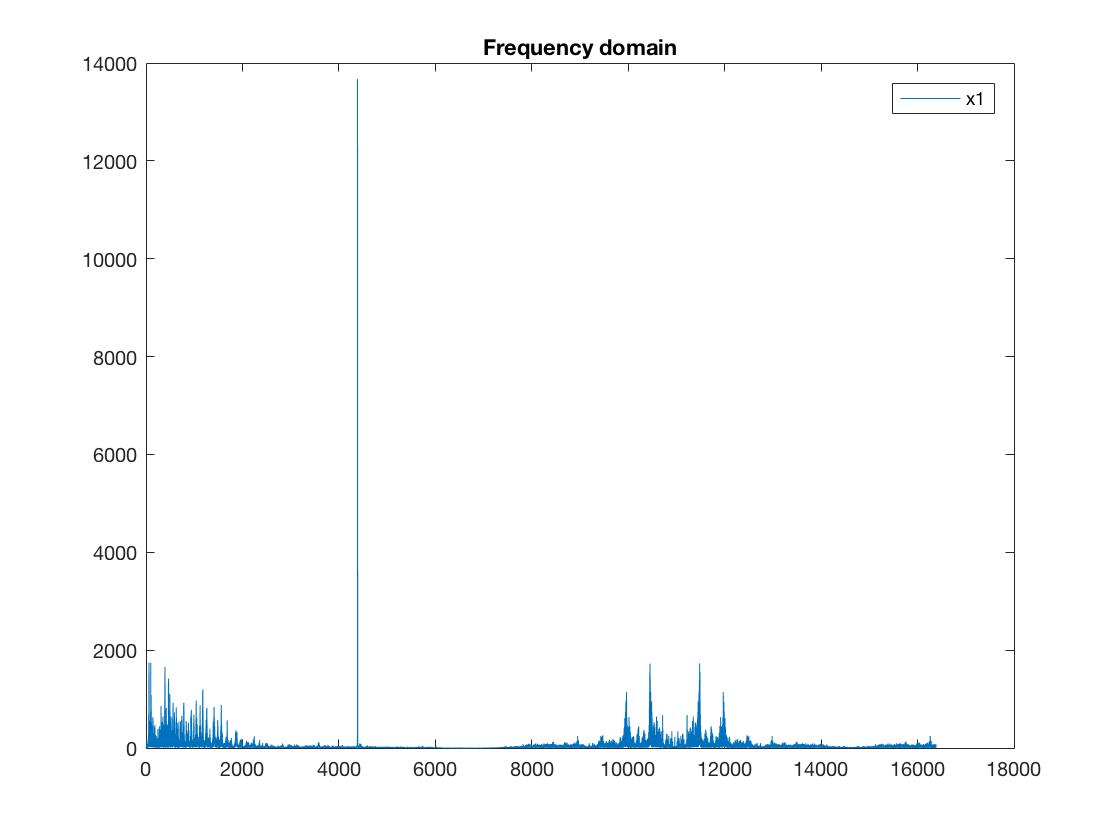


Figure 3: the original signal in frequency domain

Assuming signal is the DSB-SC modulated signal of m(t). Then we get:

Hence, the bandwidth of is , since .

So we get the bandwidth is [8192Hz, 16348Hz].

继续分析这个demodulation的过程

**2.FIR Filters Design**

2.1 Block A

In this block, we need a FIR notch filter, using the function of the Kaiser window, we can get a FIR notch filter.

Passband: [0 4815Hz] and [5015Hz ];

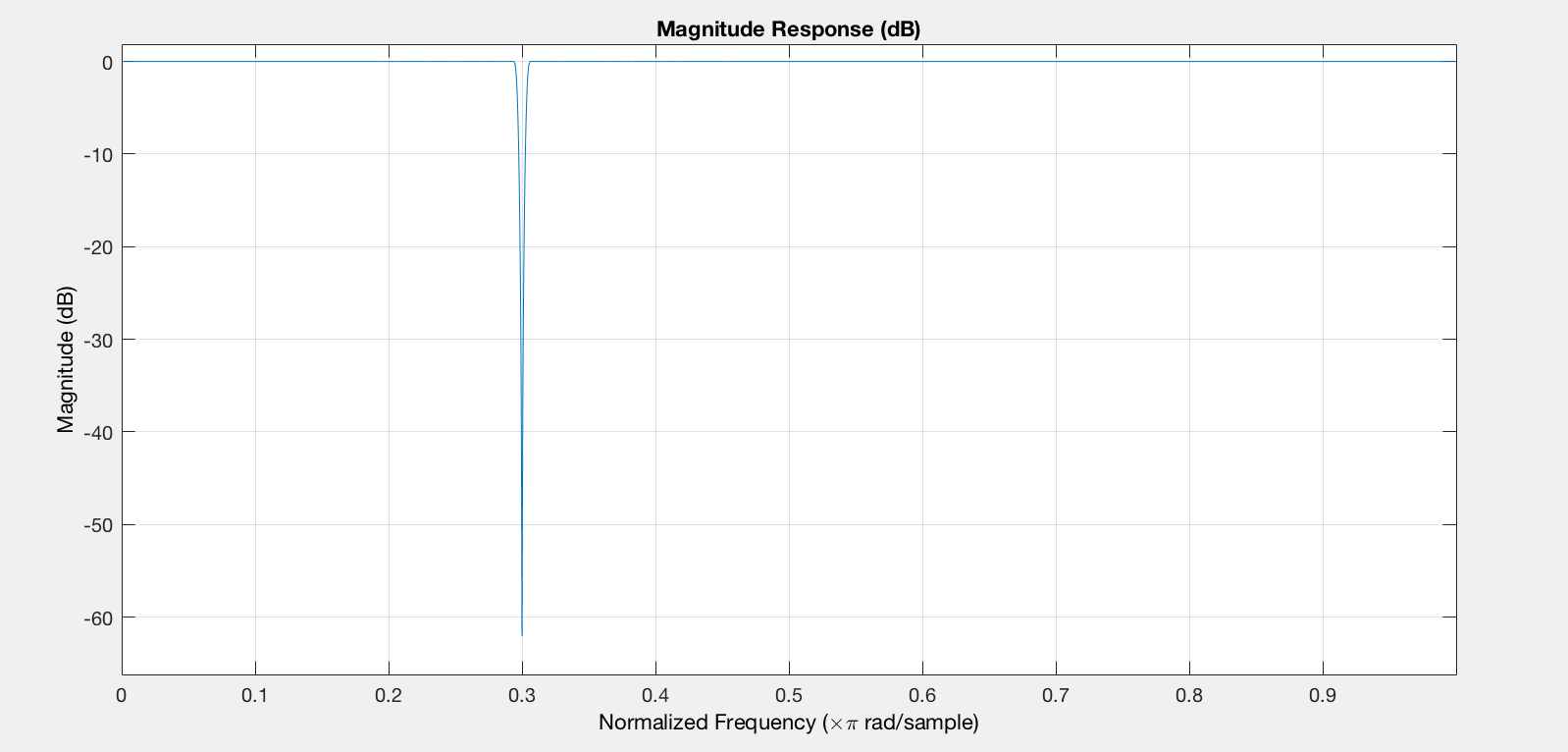
Stopband: [4914Hz 4916Hz];

Passband ripple: 0.001(for further thought about requirements of ripple)

Stopband ripple: 0.001

Using the function of *kaiserord()* to calculate the order*,*

the lowest order .



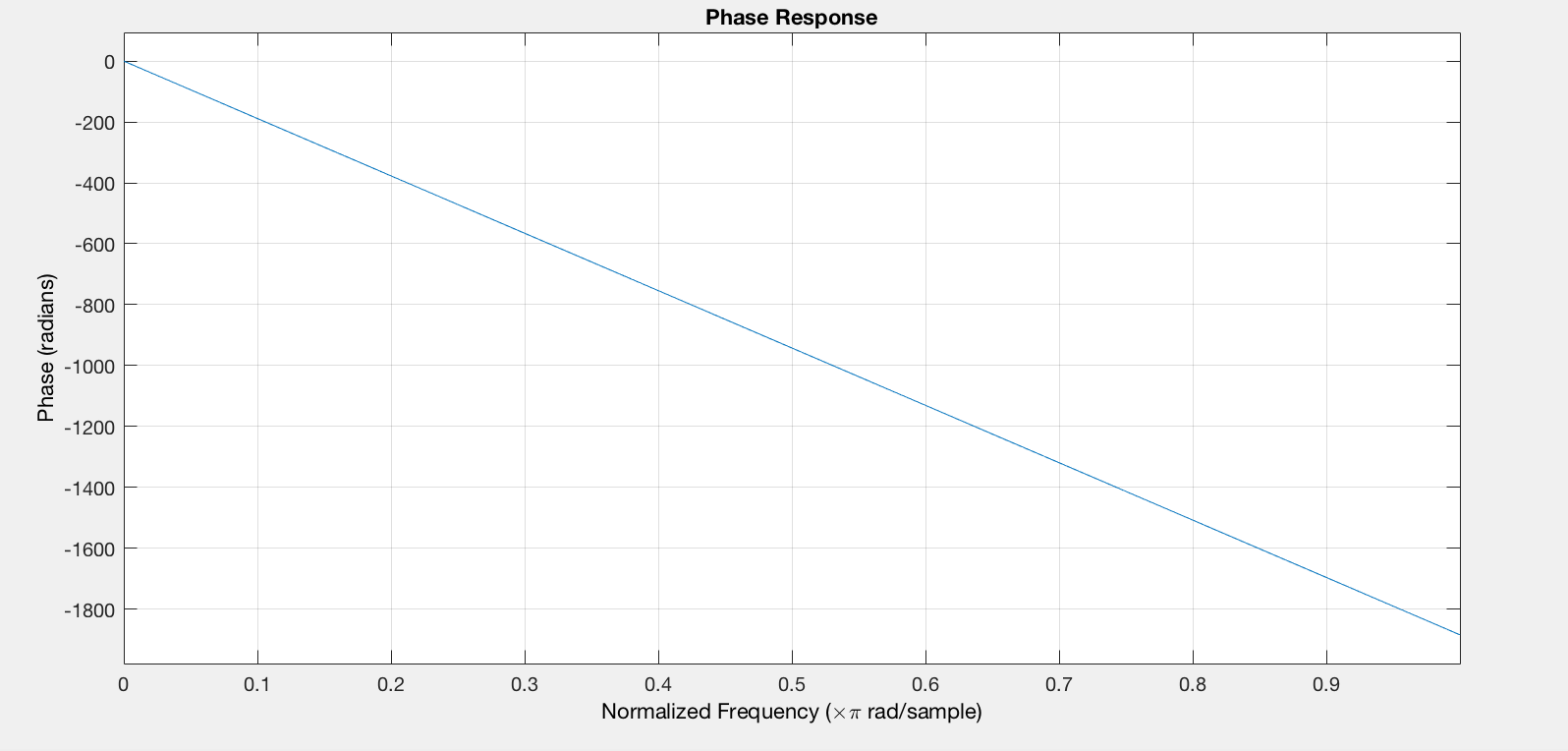


Figure 4: The magnitude and phase response of the notch filter

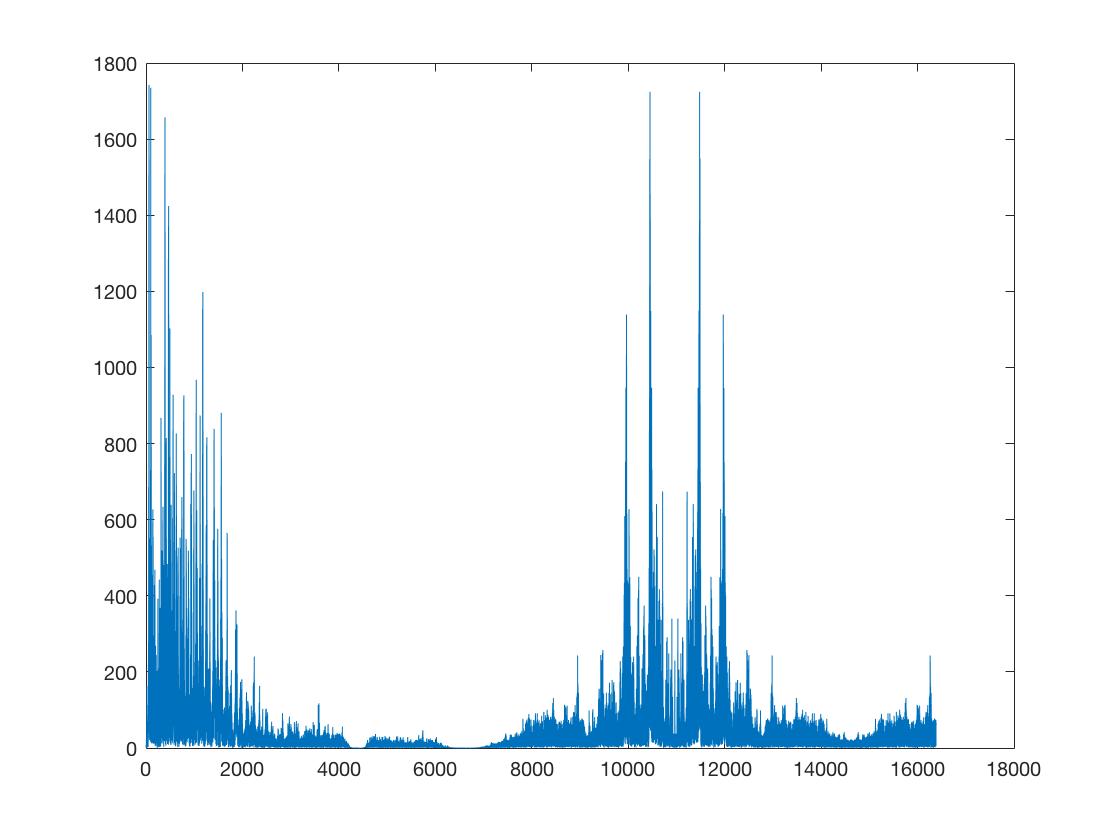


Figure 5: The output of the notch filter

Use the function *islinphase()* in Matlab to check if the filter has the linear phase. The result is logic, hence the filter satisfies the requirement of linear phase.

2.2 Block D

Using the Kaiser Window to design a lowpass filter, the requirements are below:

Passband: [0 4096Hz];

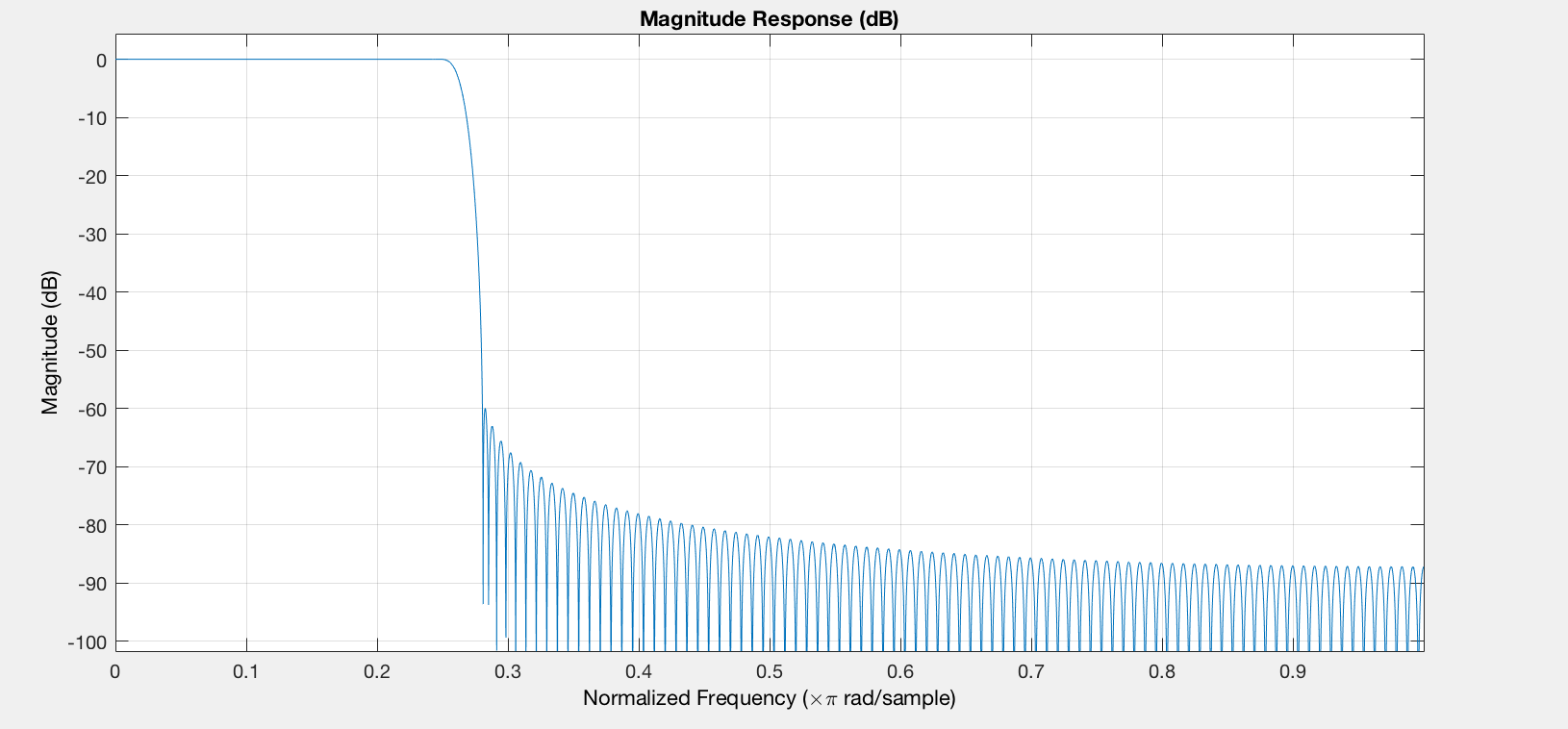
Stopband: [4596Hz ];

Passband ripple: 0.001

Stopband ripple: 0.001

Using the the function of *kaiserord()* to calculate the order,

the lowest order is .



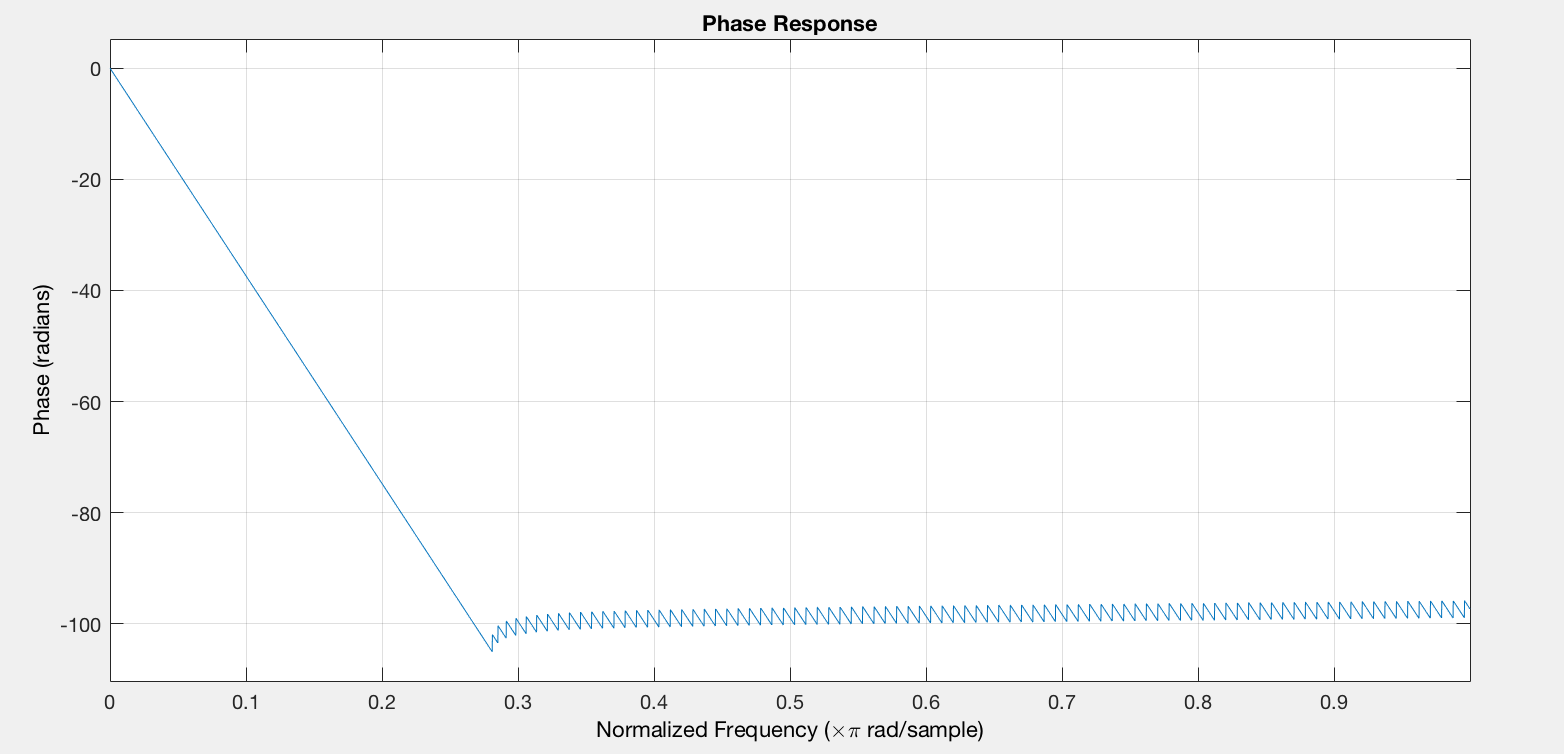
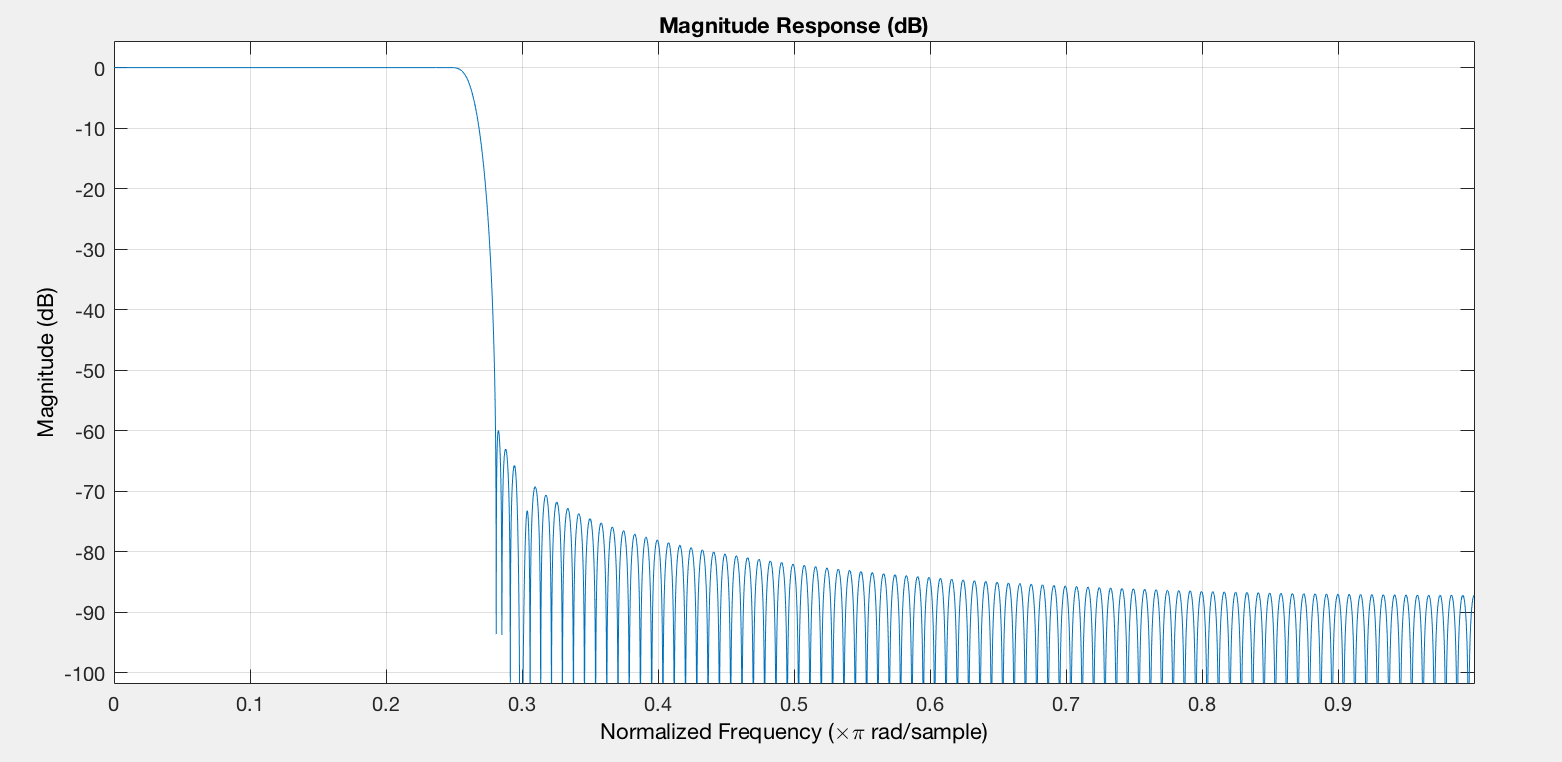


Figure 5: The magnitude and phase response of lowpass filter



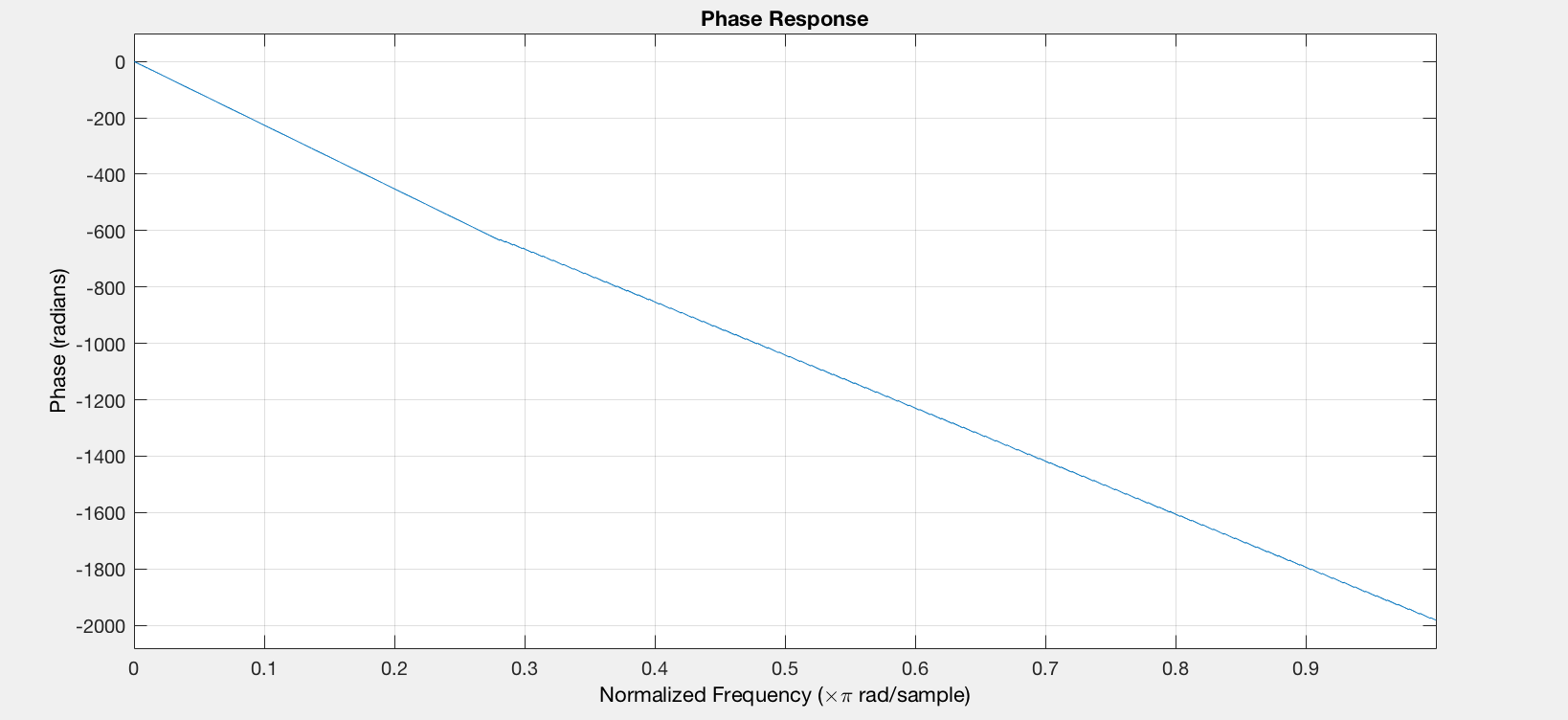


Figure 6: The magnitude and phase response of

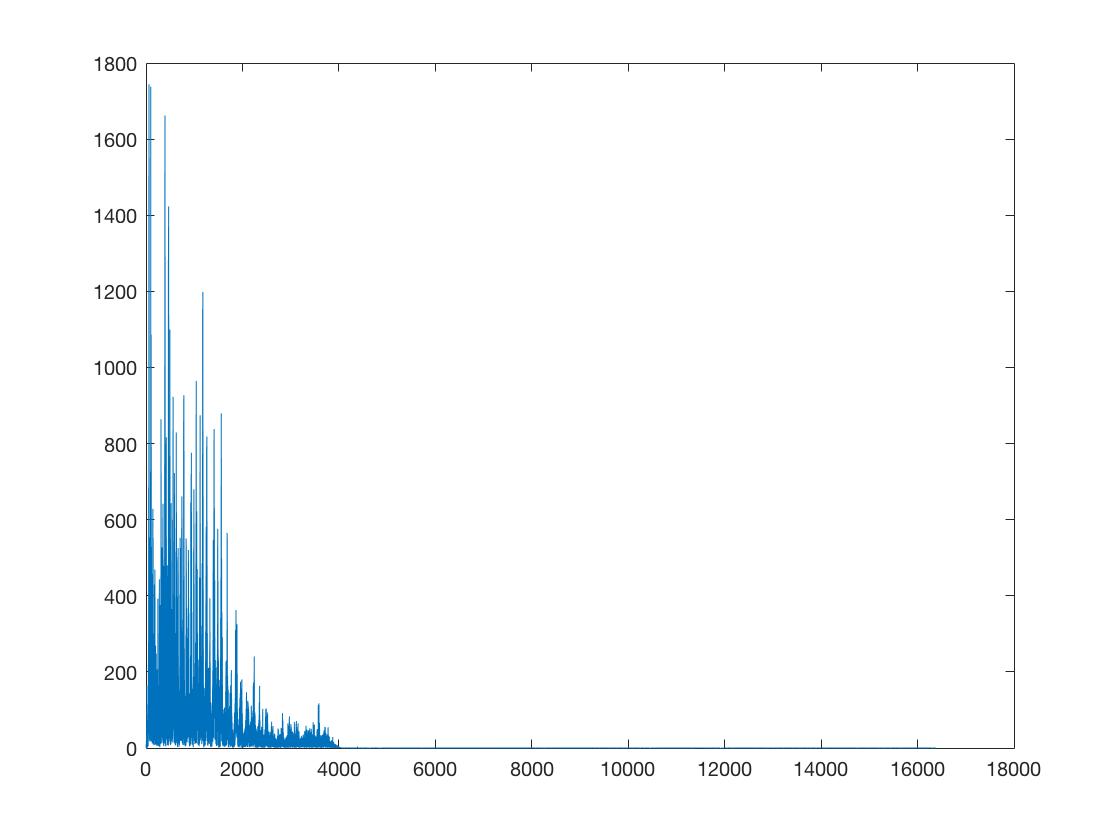


Figure 7: The output of Block D

Using the function *islinphase()* in Matlab to check if the filter has the linear phase. The result is logic, hence the filter satisfies the requirement of linear phase.

2.3 Block B

Using the Kaiser window to design a highpass filter.

Passband: [12288Hz ];

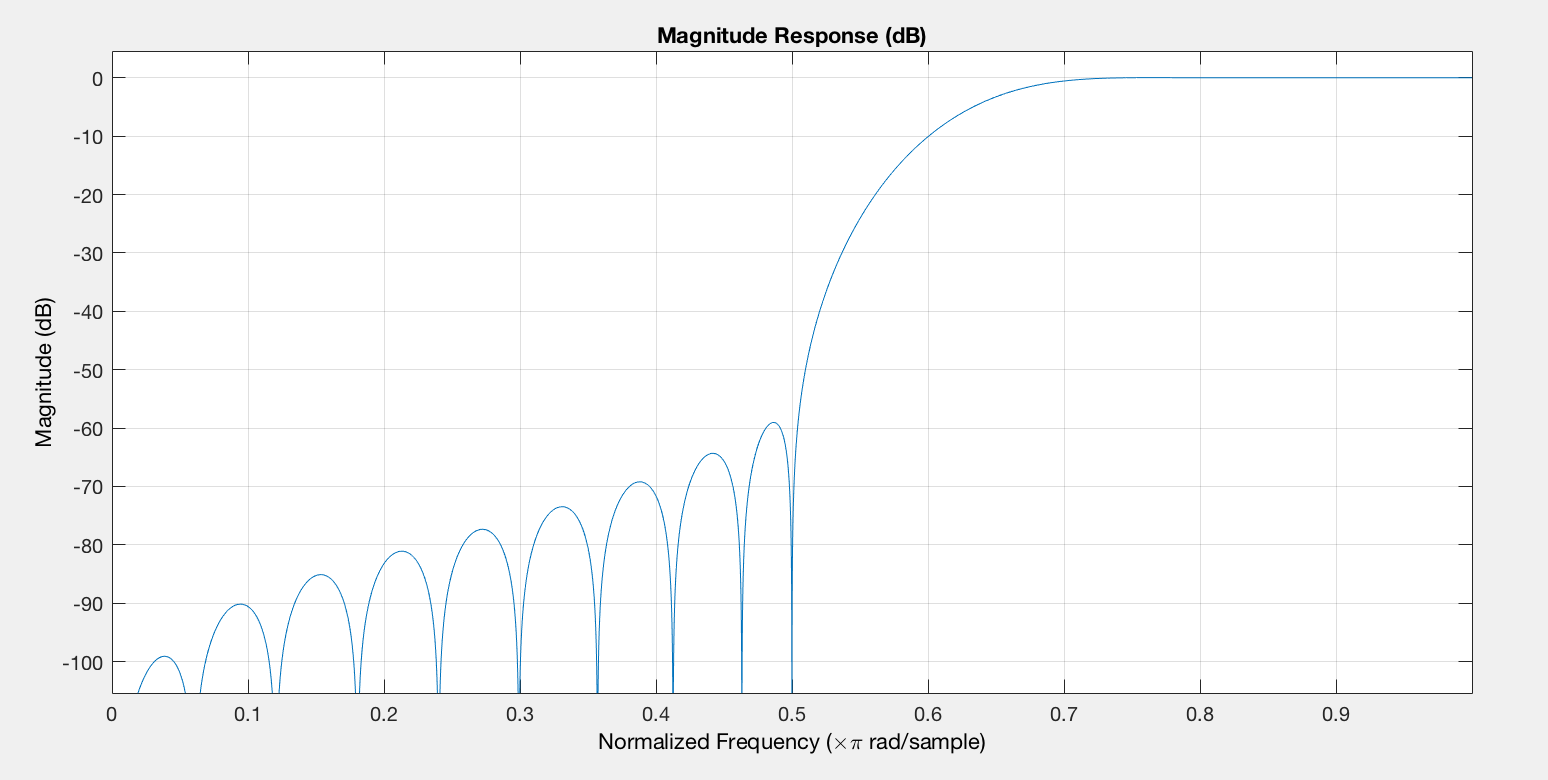
Stopband:[0 8192Hz];

Passband ripple: 0.001;

Stopband ripple: 0.001;

Using the the function of *kaiserord()* to calculate the order,

the lowest order is .



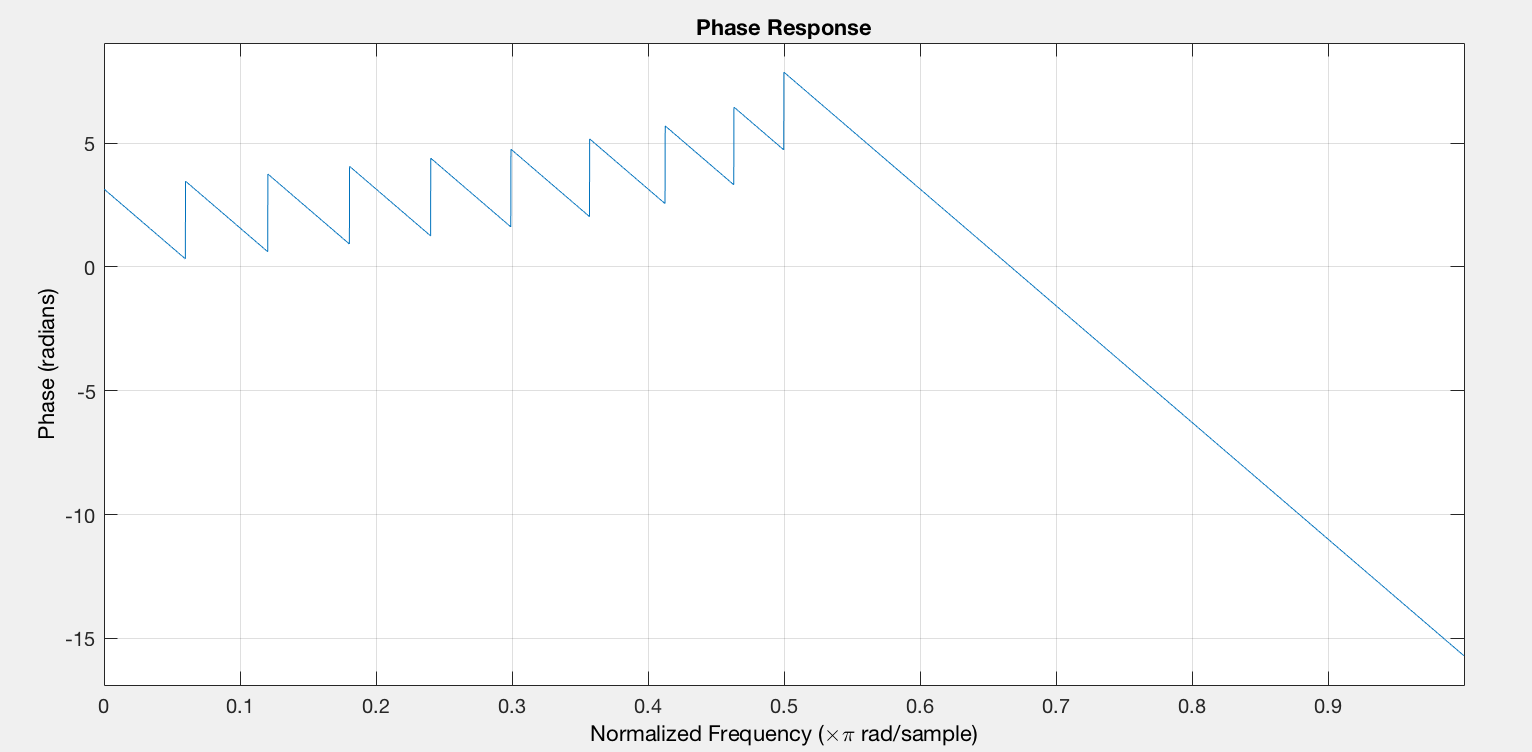


Figure 8: the magnitude and phase response of highpass filter

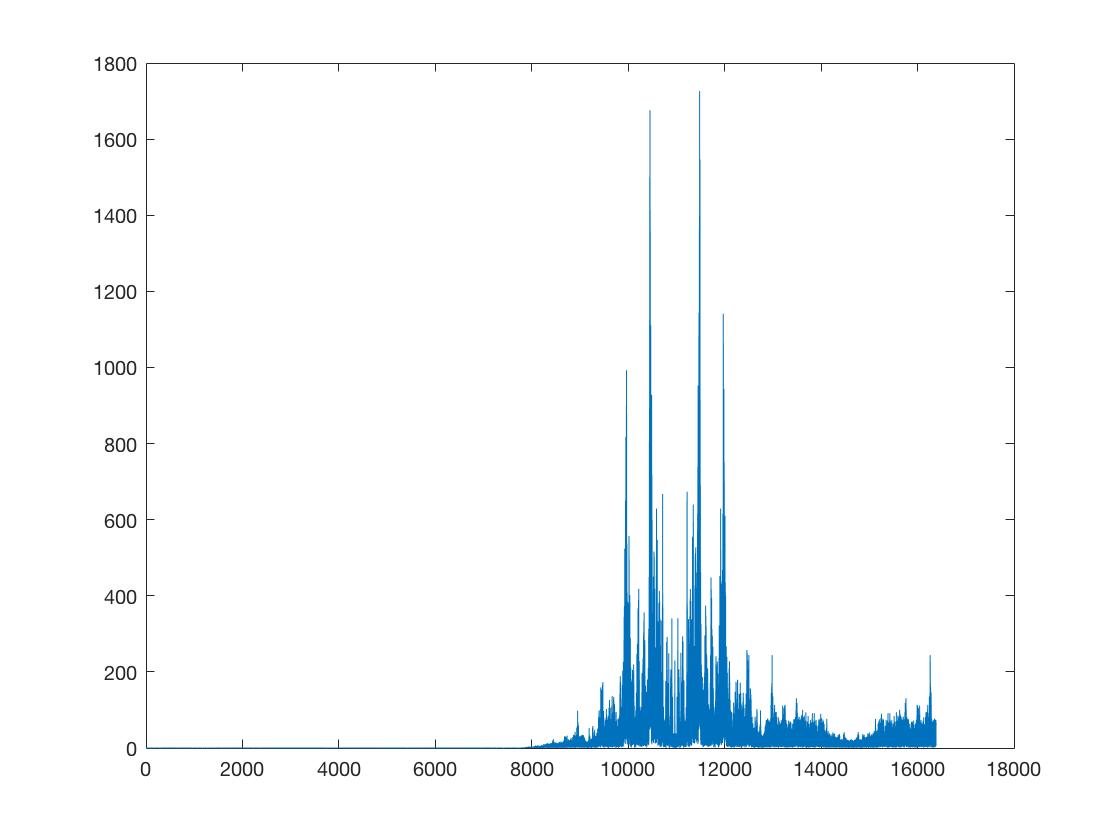


Figure 9: the output of Block B

Using the function *islinphase()* in Matlab to check if the filter has the linear phase. The result is logic, hence the filter satisfies the requirement of linear phase.

2.4 Block C

Demodulation(求phase shift的过程)

Using the Kaiser Window to design a lowpass filter

Passband: [0 4096Hz];

Stopband: [4596Hz ];

Passband ripple: 0.001

Stopband ripple: 0.001

Using the the function of *kaiserord()* to calculate the order,

the lowest order is .

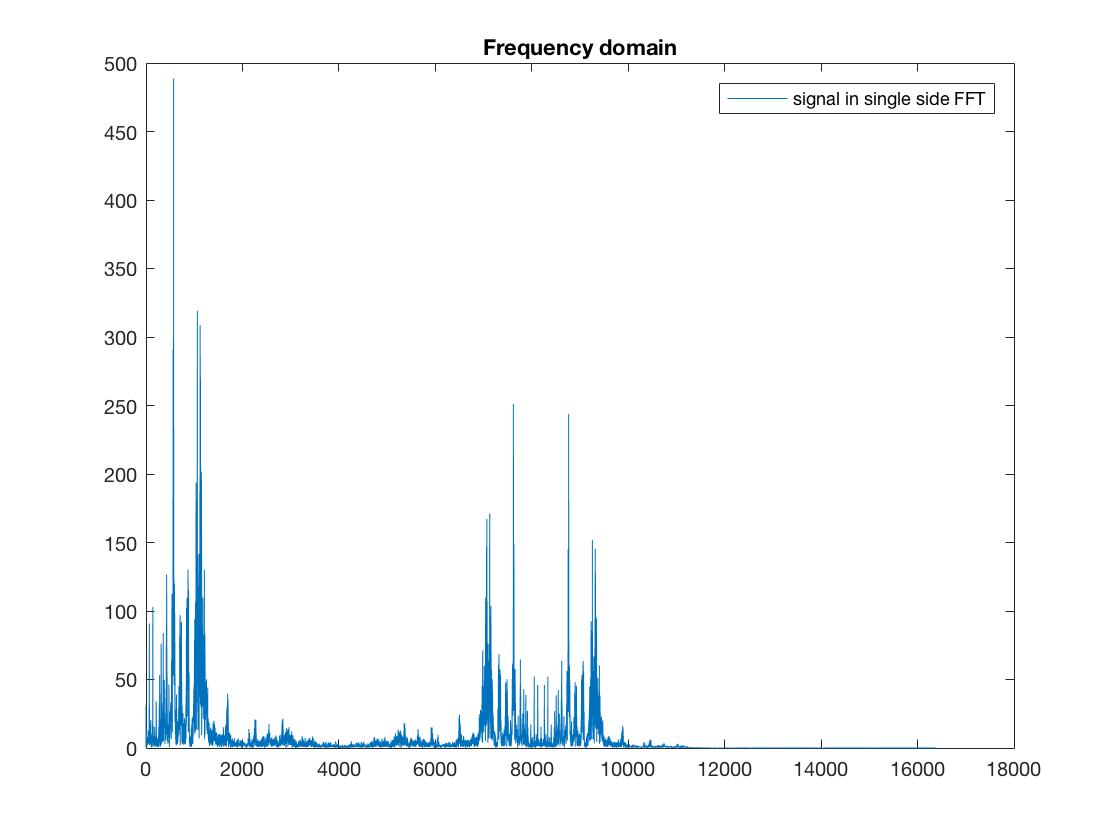


Figure 10: after demodulation

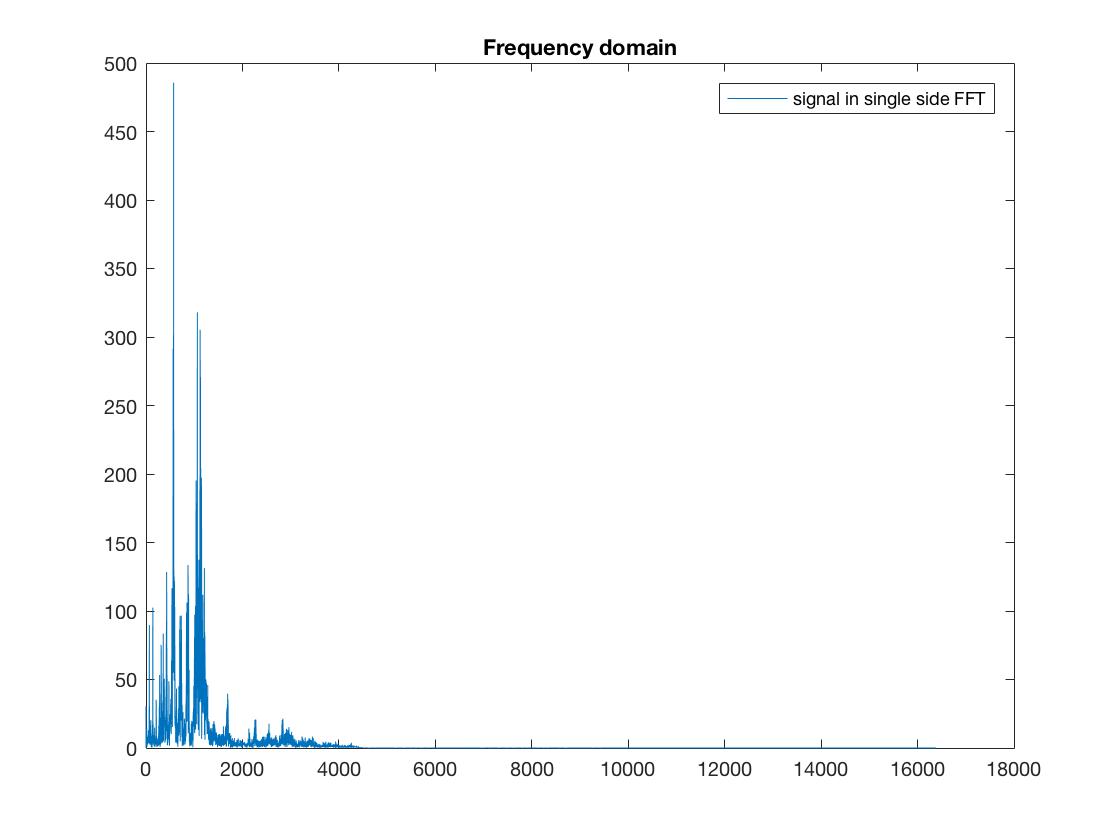


Figure 11: the output of demodulated signal after lowpass filter

Using the function *islinphase()* in Matlab to check if the filter has the linear phase. The result is logic, hence the lowpass filter in Block C satisfies the requirement of linear phase.

**3. FIR Filters Optimization**

3.1 Total Group Delay

If a filter has linear phase, its frequency response can be written as

where N is the order of the filter and is a real function of .

So the Group Delay can be calculated by

The total group delay for cascade

The total group delay for the cascade of the filters in blocks A, B and C

3.2 Minimise the Total Group Delay

By increasing the transition region and making full use of ripple, we optimize the filters with new specifications.

Block A

Passband: [0 4015Hz] and [5815Hz ];

Stopband: [4915.4Hz 4915.6Hz];

Passband ripple: 0.001

Stopband ripple: 0.001

Now the order of the notch filter is 134.

Block D and Block C

Passband: [0 4096Hz];

Stopband: [5415Hz ];

Passband ripple: 0.01

Stopband ripple: 0.01

Now the order of the lowpass filter is 60.

From the magnitude response we can know the maximum ripple of is

A = 1.0074976

A-1=0.0074976<0.01

Block B

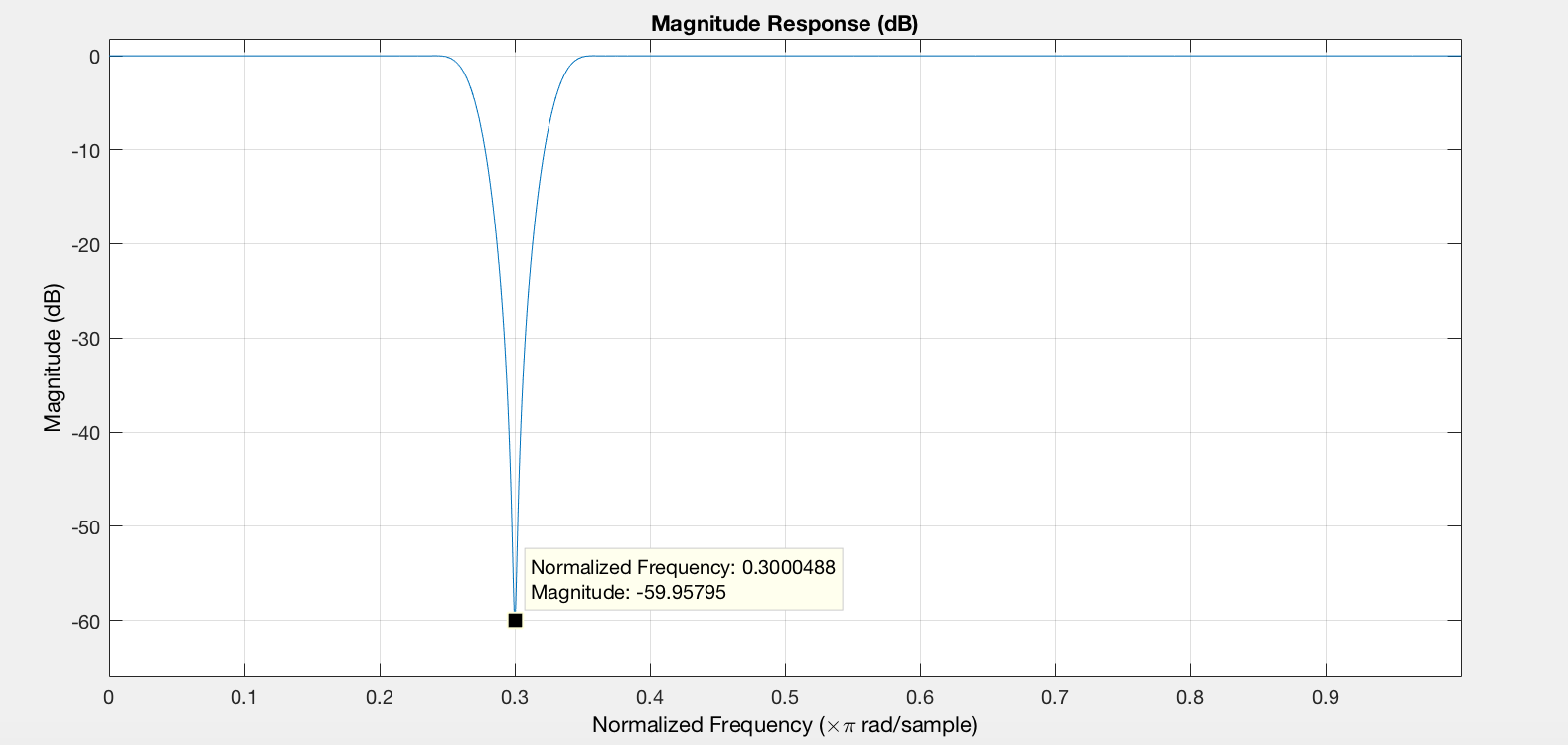
Passband: [8192Hz ];

Stopband:[0 4096Hz];

Passband ripple: 0.001;

Stopband ripple: 0.001;

Now the order of the highpass filter is 30.



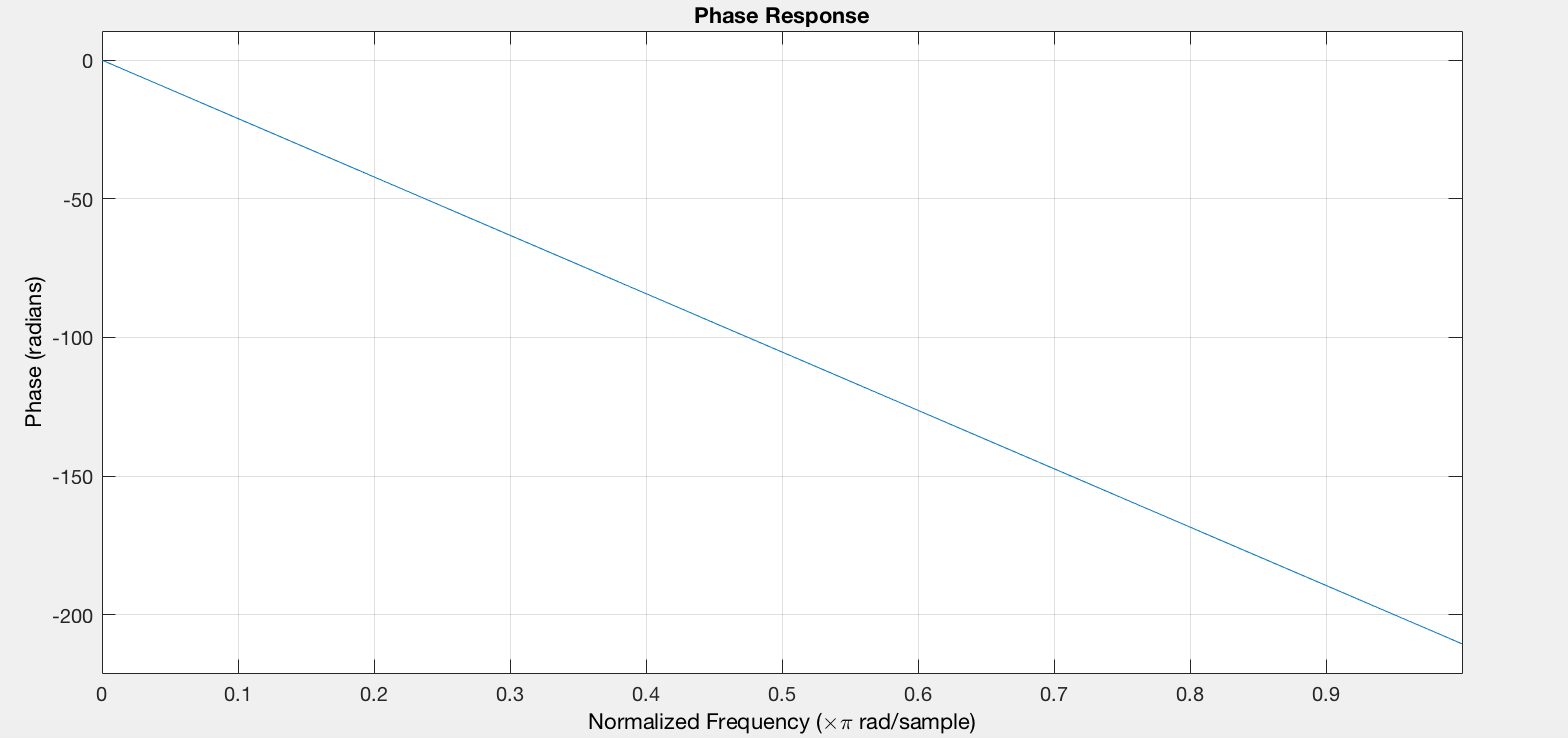
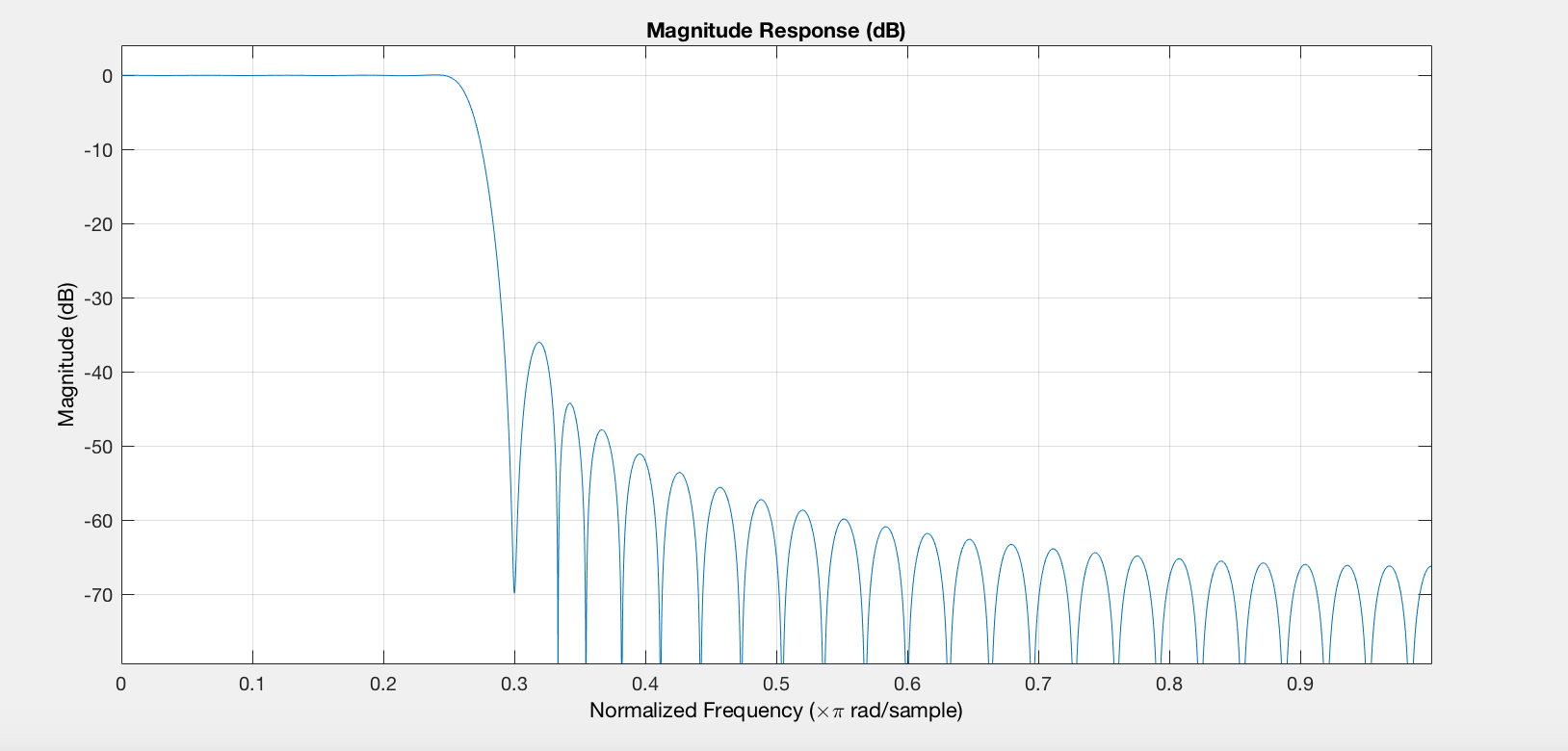
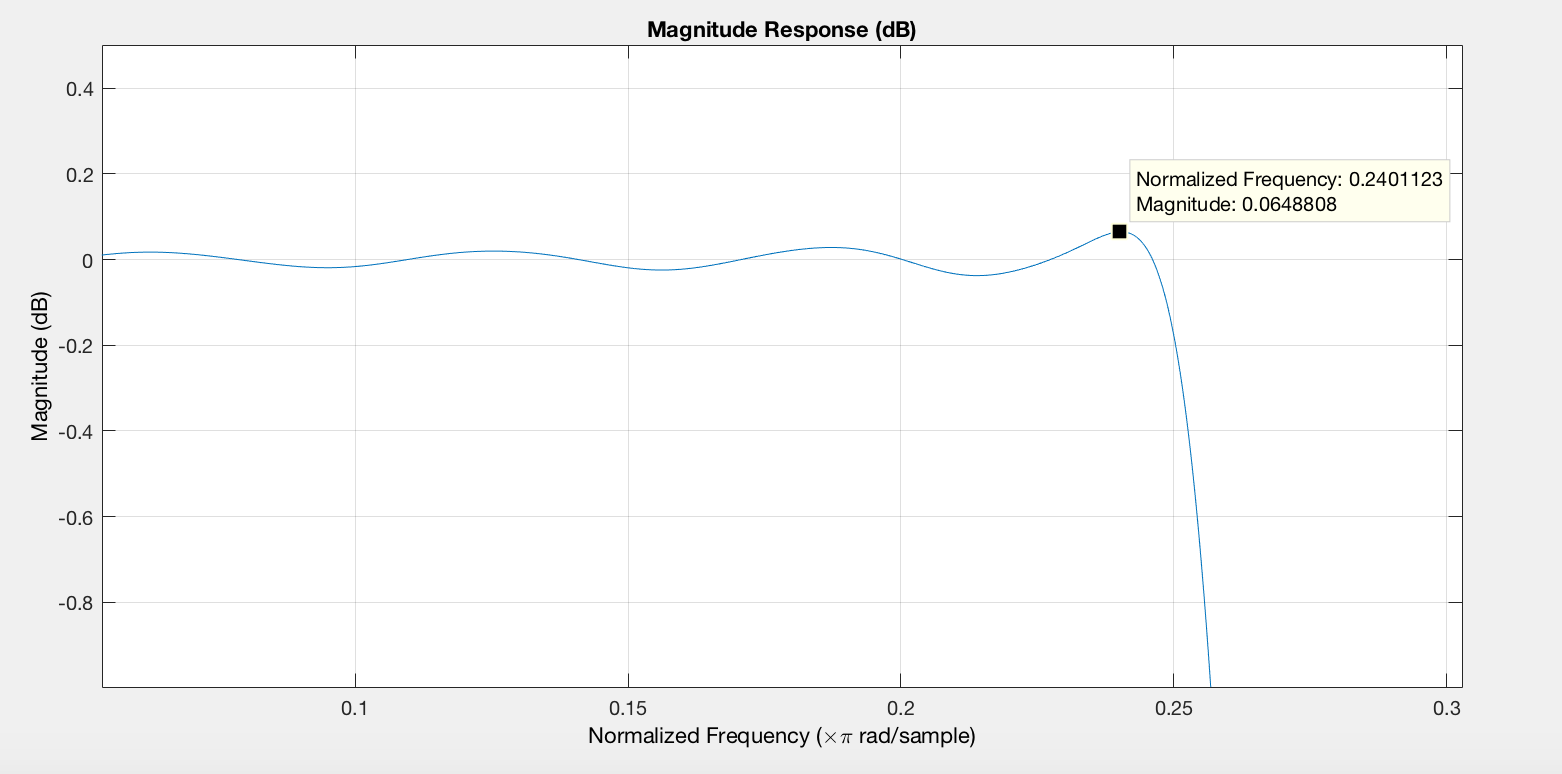


Figure 12: the new notch filter bode diagram





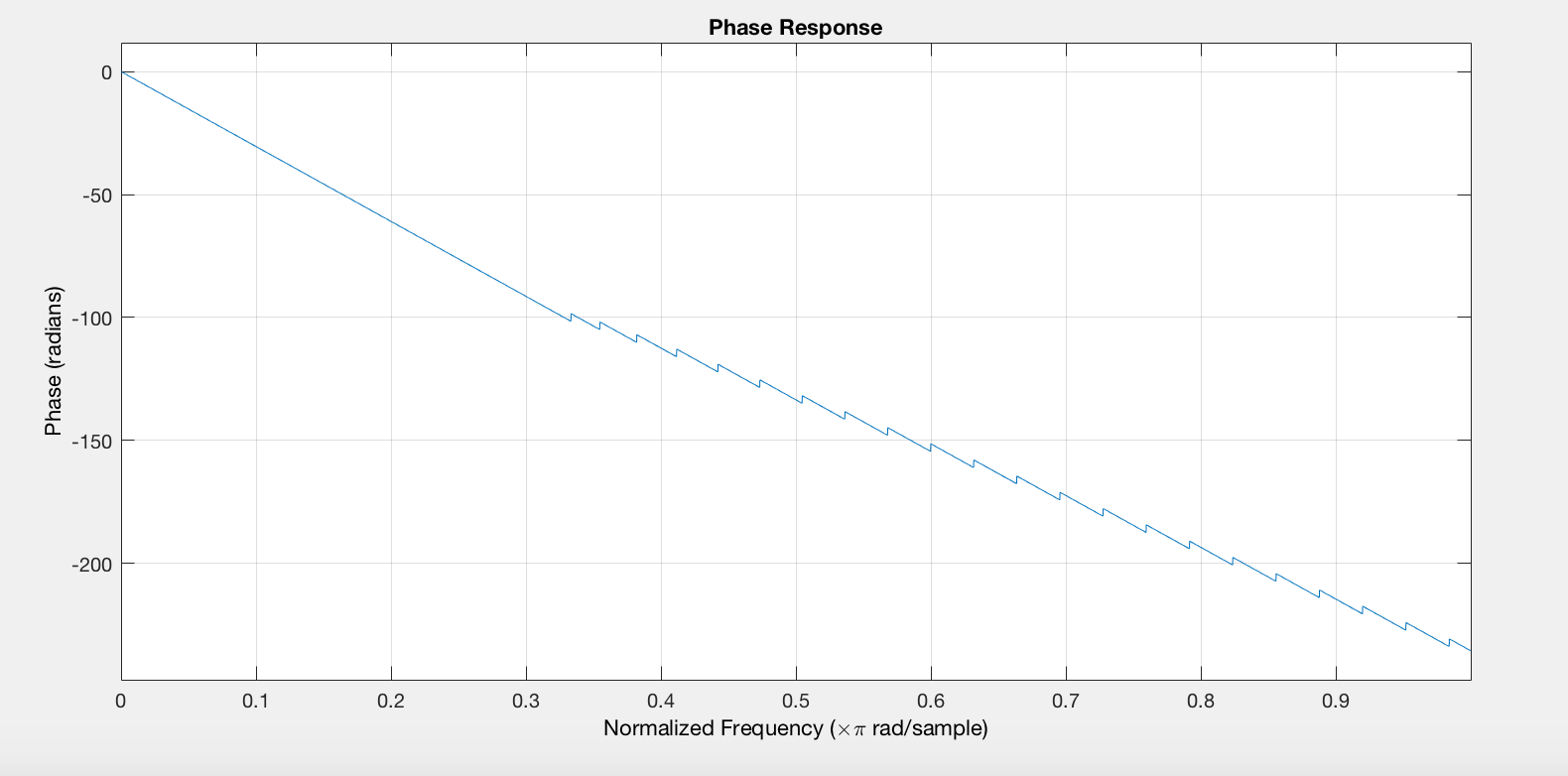
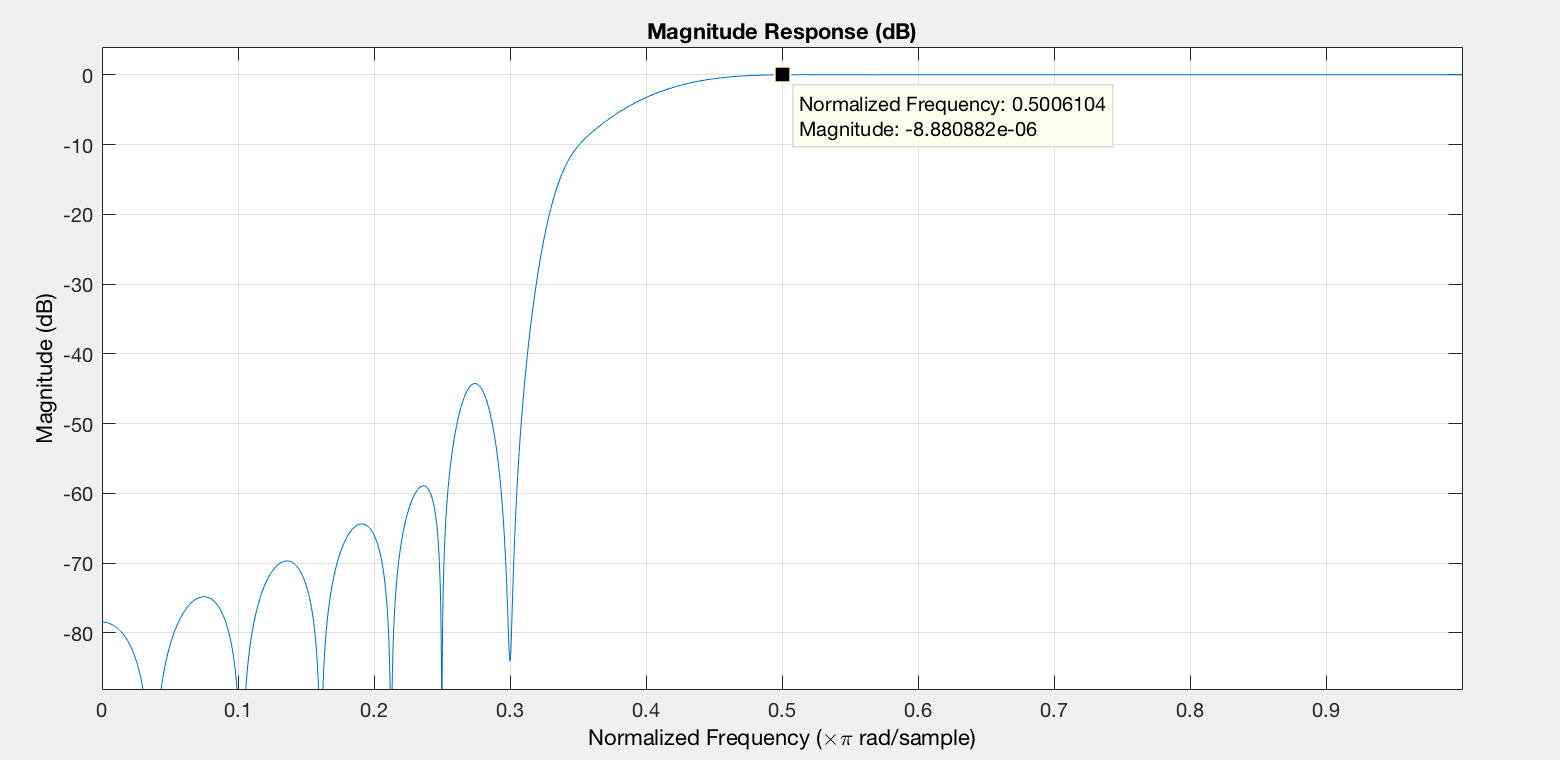


Figure 13: the response of



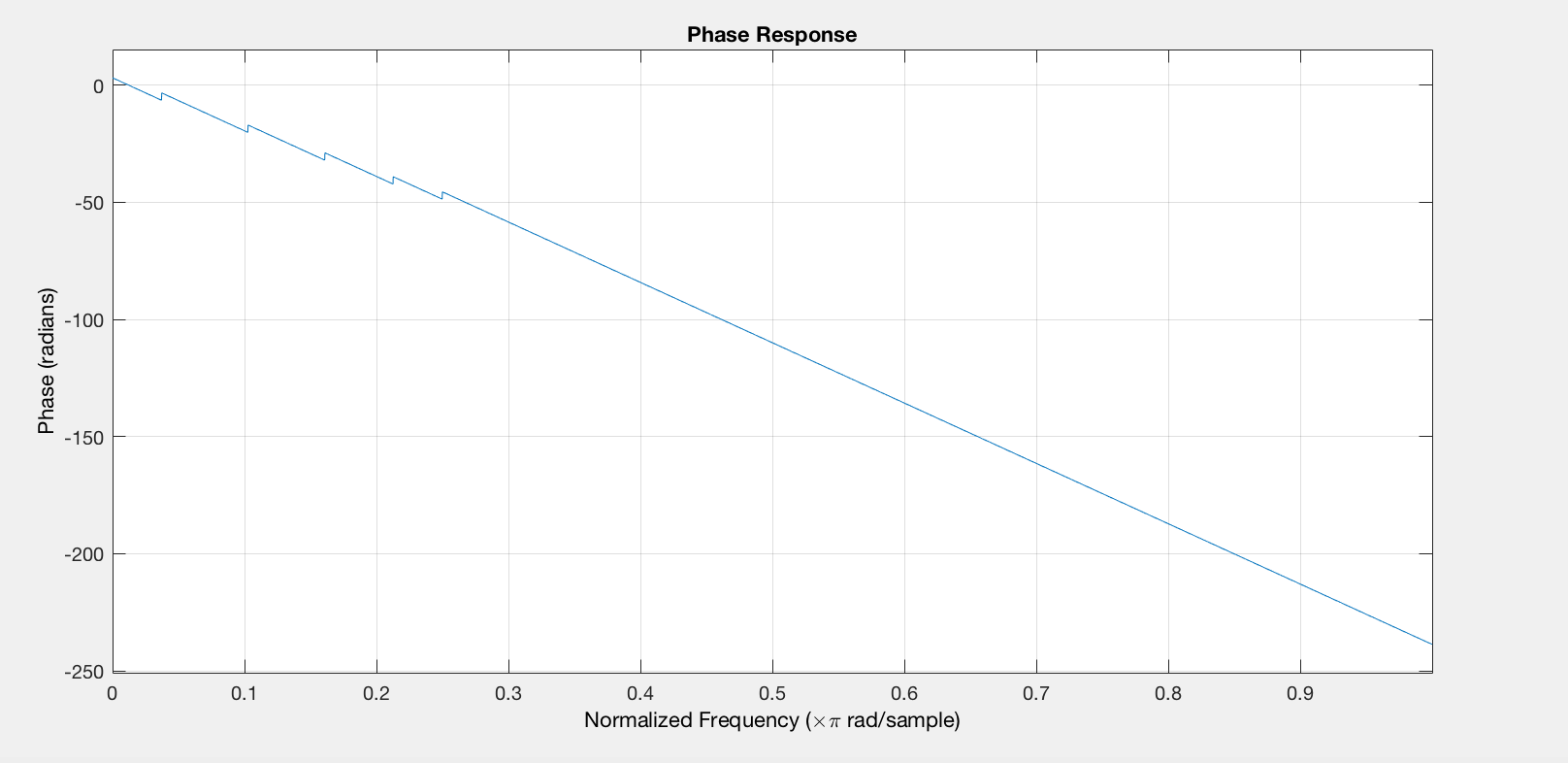
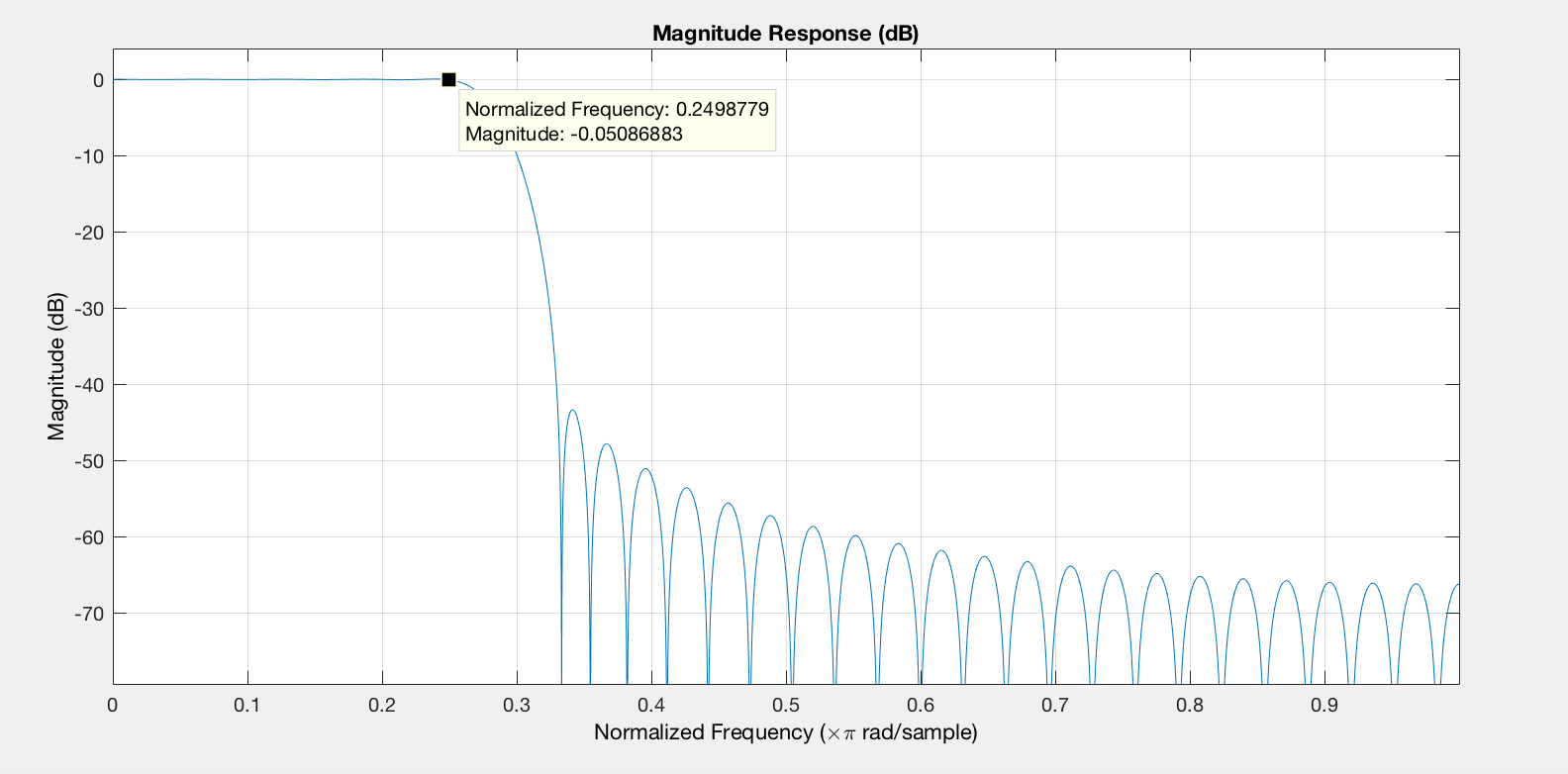


Figure 14: the response of cascade of filters A and B



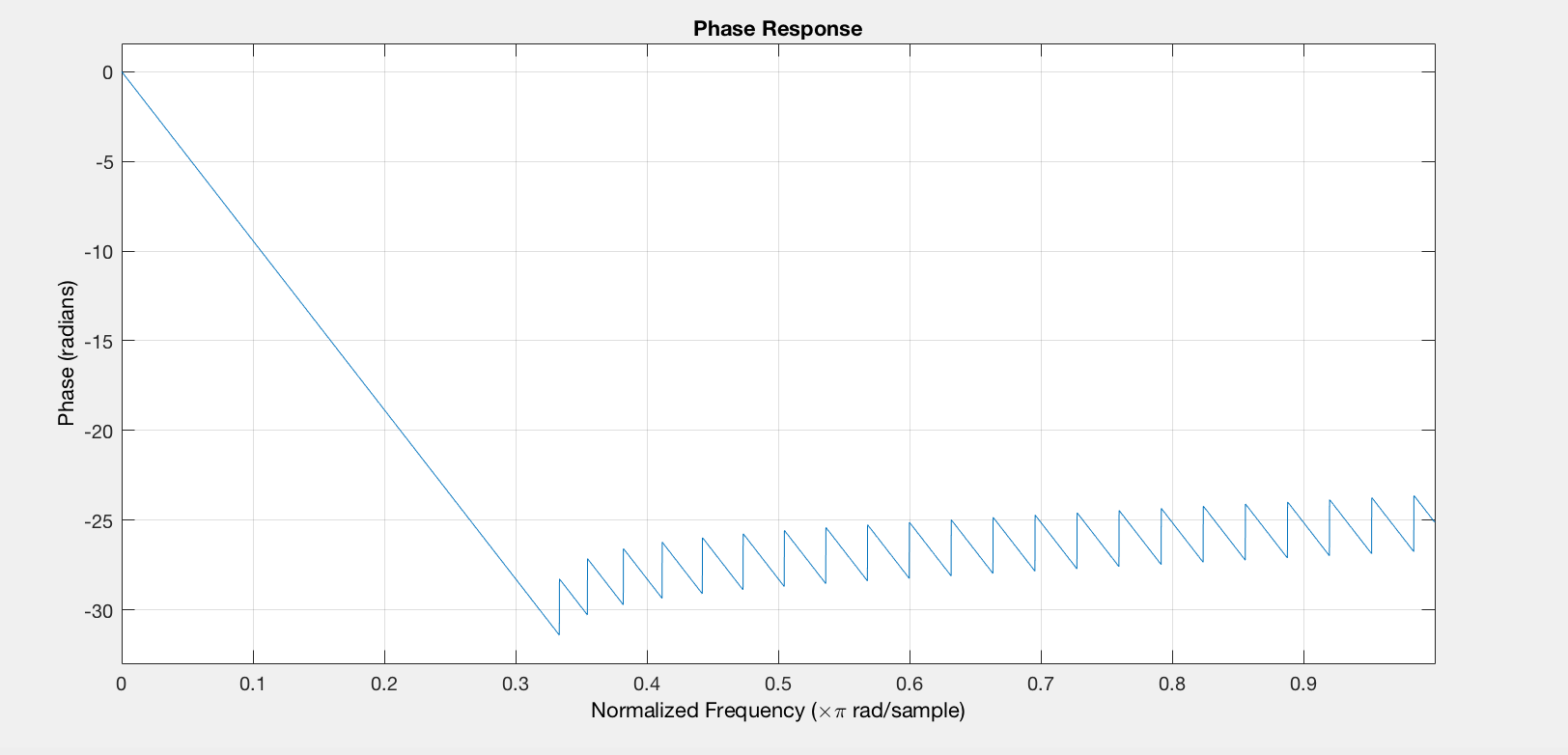


Figure 15: the response of filter in Block C

From the figure 12 it can be observed that the gain at the notch frequency 0.3π is less than -60dB.

From the figure 13 we can know the maximum ripple of is

A = 1.0074976

A-1=0.0074976<0.01

From the figure 14 and 15 we can know the maximum ripple of filters A, B and C

Hence,

The new filters satisfy all the requirements.

The total group delay after minimizing,

For

For cascade of the filters in blocks A, B and C

Appendix

FIR fliters

clear;

close all;

load projsignal0.mat;

rs = rs(1:25E3);%only take the first 25000 data point

fvtool(rs);

fsamp = 32768;

% block A notch filter

Bw1 = 4096;

fcuts = [4015 4915.4 4915.6 5815];

mags = [1 0 1];

devs = [0.001 0.001 0.001];

[n,Wn,beta,ftype] = kaiserord(fcuts,mags,devs,fsamp);

n = n + rem(n,2);

hbs = fir1(n,Wn,ftype,kaiser(n+1,beta),'noscale');

Ybs = filter(hbs,1,rs);

fvtool(Ybs);

%block D lowpass filter

fcuts = [4096 5415];

mags = [1 0];

devs = [0.01 0.008];

[n,Wn,beta,ftype] = kaiserord(fcuts,mags,devs,fsamp);

n = n + rem(n,2);

hlp = fir1(n,Wn,ftype,kaiser(n+1,beta),'noscale');

Ylp = filter(hlp,1,Ybs);

fvtool(Ylp);

%Ylp is the output signal of block D, s1[n];

%block B highpass filter

fcuts = [4096 8192];

mags = [0 1];

devs = [0.01 0.001];

[n,Wn,beta,ftype] = kaiserord(fcuts,mags,devs,fsamp);

n = n + rem(n,2);

hhp = fir1(n,Wn,ftype,kaiser(n+1,beta),'noscale');

Yhp = filter(hhp,1,Ybs);

fvtool(Yhp);

% Yhp is the output signal of block B, use Yhp to demodulation

fs1 = fsamp;

t = (0:1:24999).\* (1/fs1);

demod = zeros(1, 25000);

for i = 1:1:25000

    demod(i) = Yhp(i).\*cos(2\*pi\*12288\*t(i)+ 0.3\*pi );

end

figure;

N=25000;

X1\_mags = abs(fft(demod));

fax\_bins = [0 : N-1]; %frequency axis in bins

N\_2 = ceil(N/2);

plot(fax\_bins(1:N\_2)\*fs1/N, X1\_mags(1:N\_2));

legend('signal in single side FFT');

title('Frequency domain');

%block C lowpass filter

fcuts = [4096 4596];

mags = [1 0];

devs = [0.001 0.001];

[n,Wn,beta,ftype] = kaiserord(fcuts,mags,devs,fsamp);

n = n + rem(n,2);

hlp = fir1(n,Wn,ftype,kaiser(n+1,beta),'noscale');

Ylp\_1 = filter(hlp,1,demod);

figure;

N=25000;

X1\_mags = abs(fft(Ylp\_1));

fax\_bins = [0 : N-1]; %frequency axis in bins

N\_2 = ceil(N/2);

plot(fax\_bins(1:N\_2)\*fs1/N, X1\_mags(1:N\_2));

legend('signal in single side FFT');

title('Frequency domain');