1. **Analysis of signal**

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

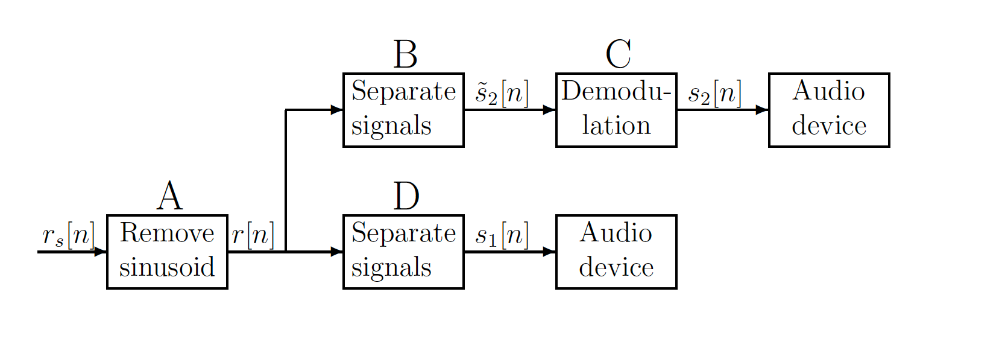
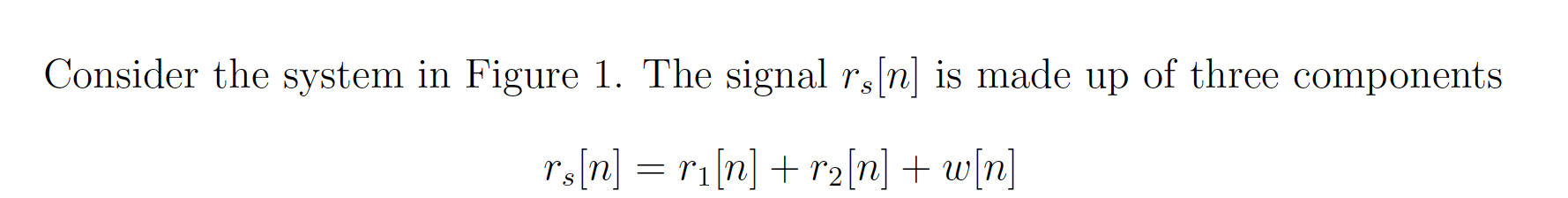


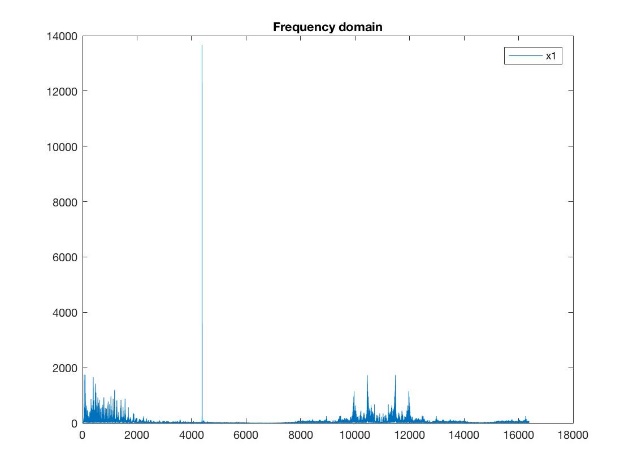
Figure1: Signal removal and separation



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.



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].

*Figure 3: the original signal in frequency domain*

**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 .

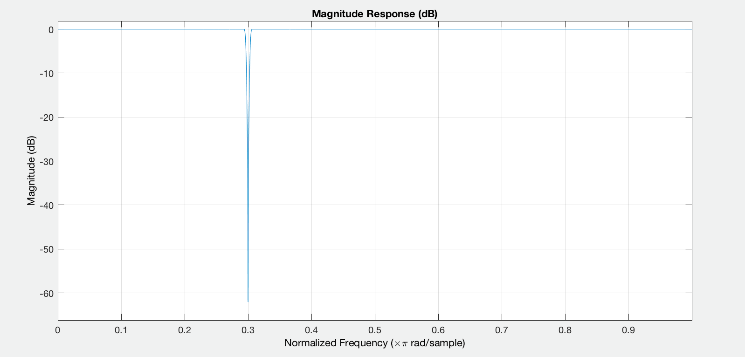
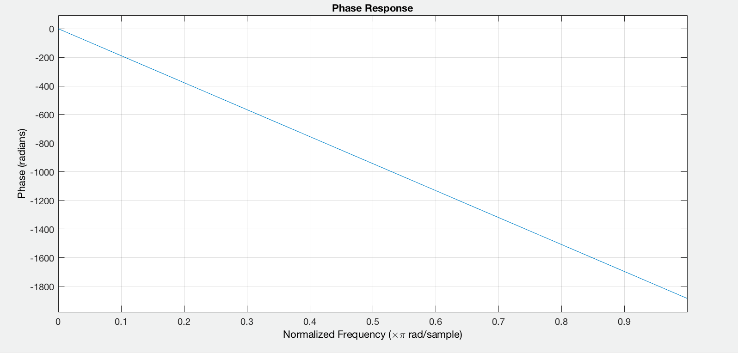


Figure 4: The magnitude and phase response of the notch filter

As figure 4 shows, the gain of the filter at 0.3pi is smaller than -60dB, which satisfy with the design requirement of the stopband, moreover, using the function *islinphase()* in Matlab, the filter has the linear phase. hence the filter satisfies the requirement of linear phase.

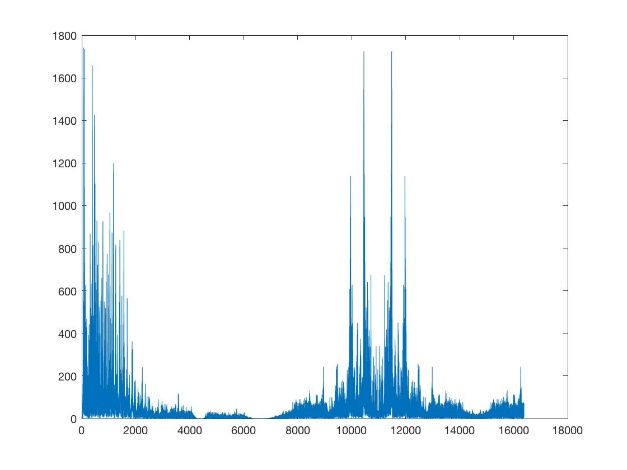


Figure 5: The output of the notch filter

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 .

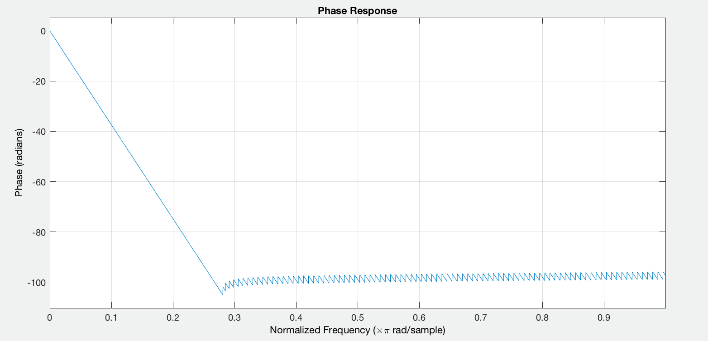
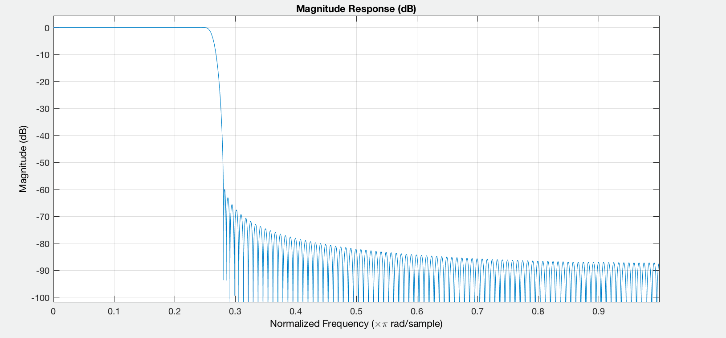
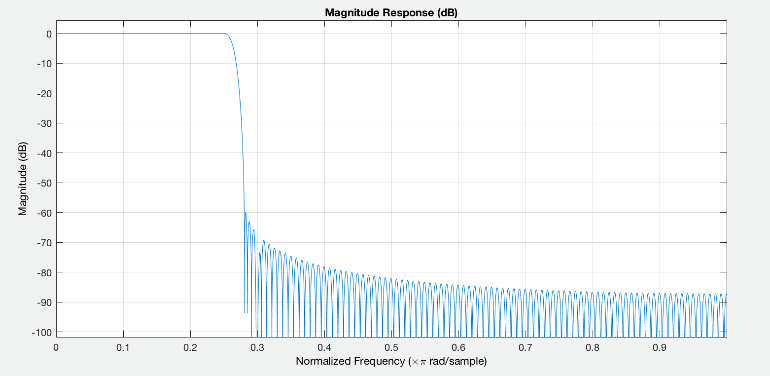


Figure 5: The magnitude and phase response of Block D lowpass filter

*islinphase()* was used to check if the cascaded filter has the linear phase and the filter satisfies the requirement of linear phase.



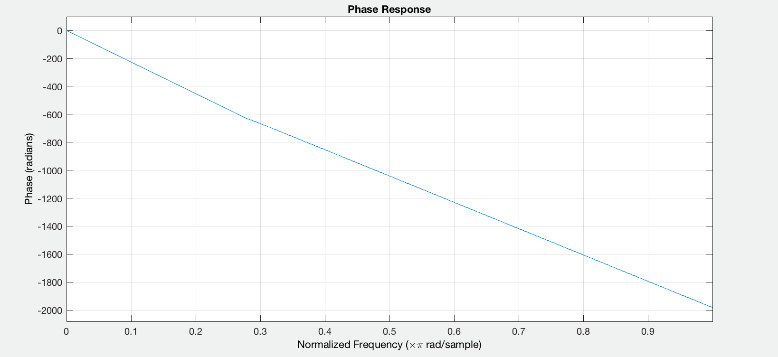
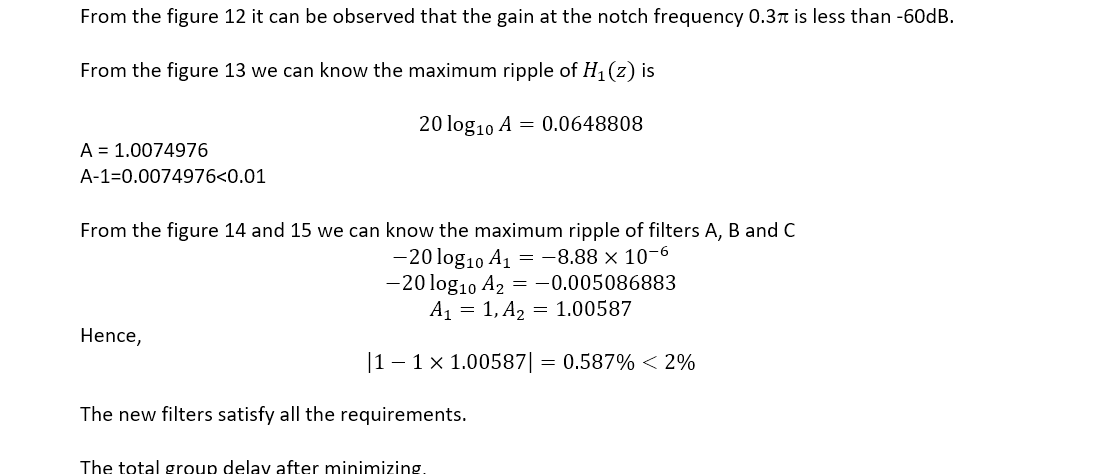


Figure 6: The magnitude and phase response of

Ripple analysis

检查0.25pi edge of passband 处的distortion确实小于1%

像这种



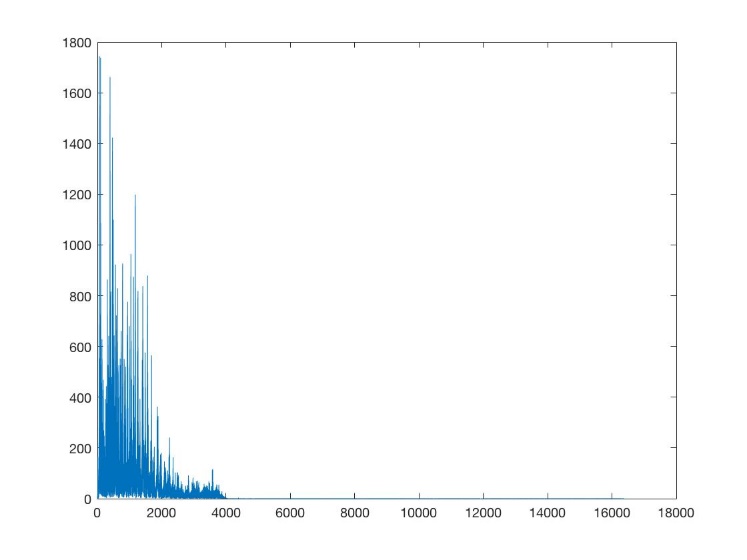


Figure 7: The output of Block D

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,

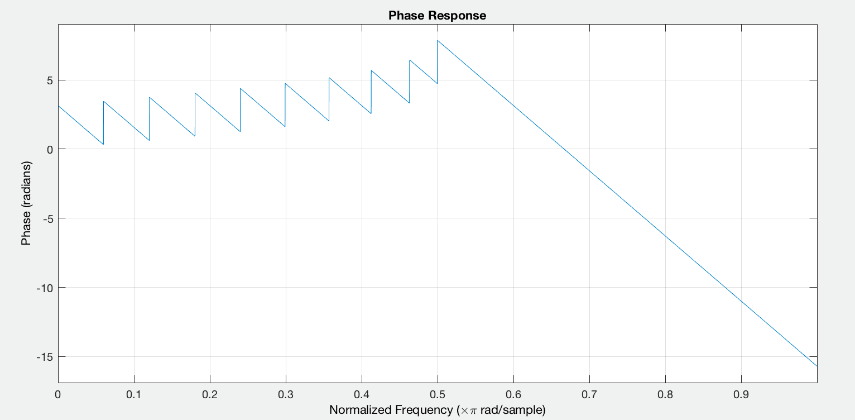
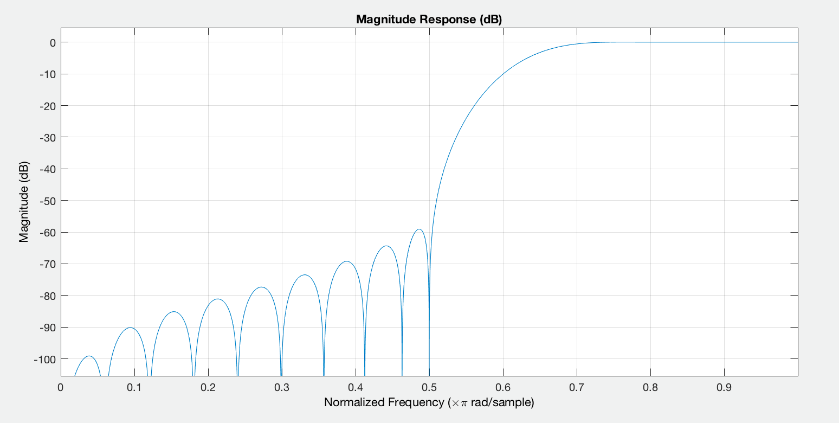
the lowest order is .

Figure 8: the magnitude and phase response of highpass filter

using the function *islinphase()* in Matlab, the filter has the linear phase. hence the filter satisfies the requirement of linear phase.

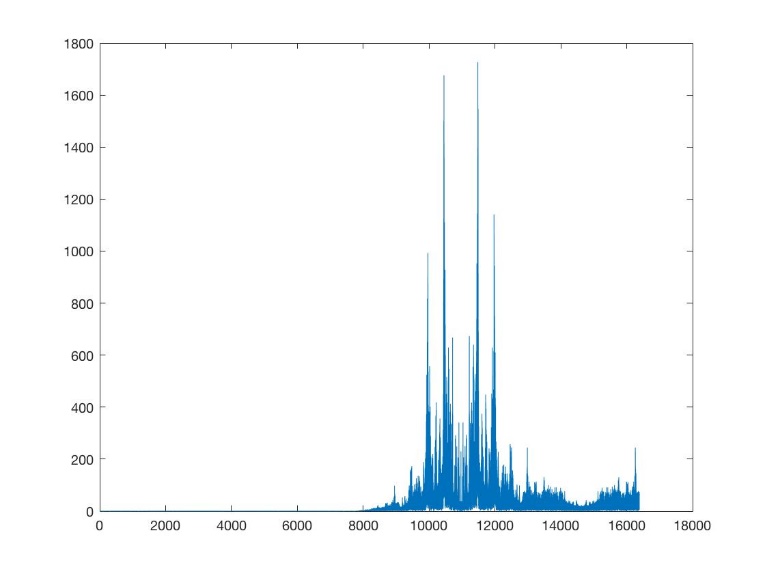


Figure 9: the output of Block B

Demodulation(求phase shift的过程)

我现在写这个

2.4 Block C

Using the Kaiser Window to design a lowpass filter

Passband: [0 4096Hz];

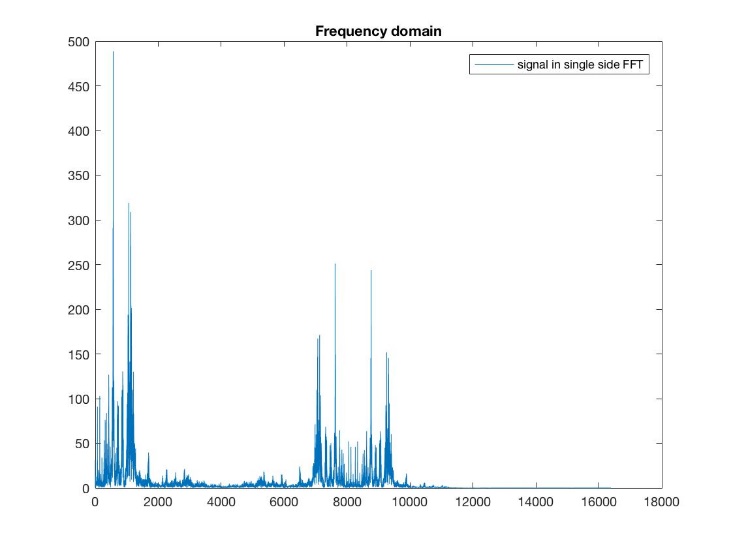
Stopband: [4596Hz ];

Passband ripple: 0.001

Stopband ripple: 0.001

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

the lowest order is .



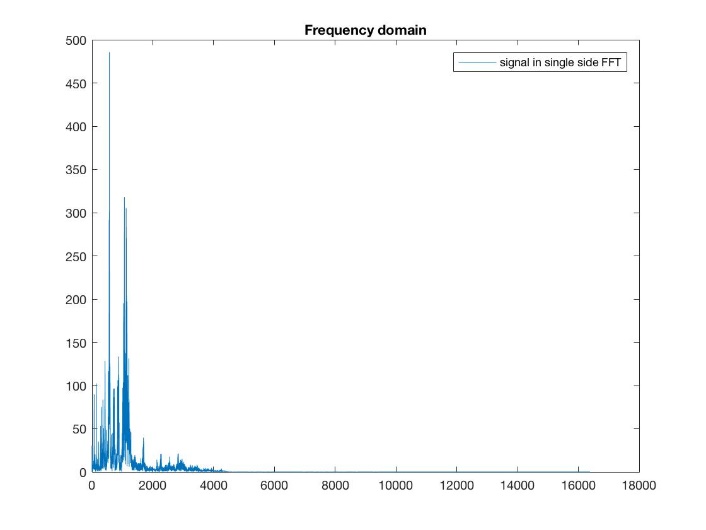


Figure 10: after demodulation Figure 11: the output of demodulated signal after lowpass filter

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.

在这里加检查A+B+C的ripple确实小于2%

**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

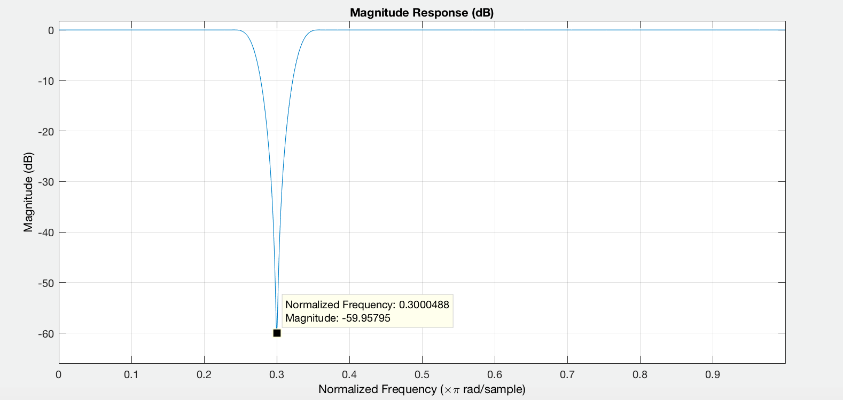
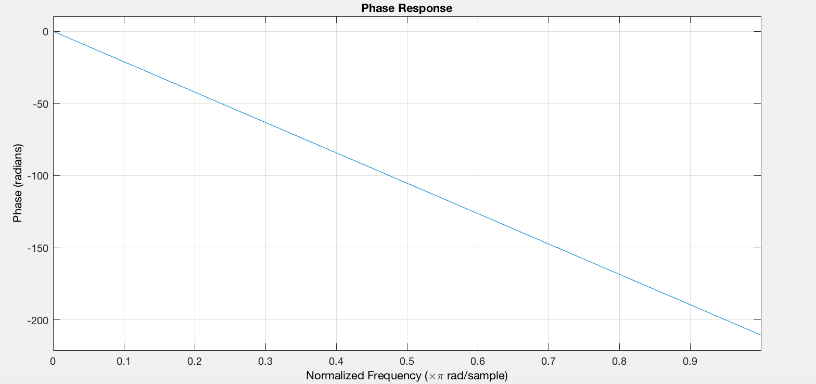
Passband: [8192Hz ];

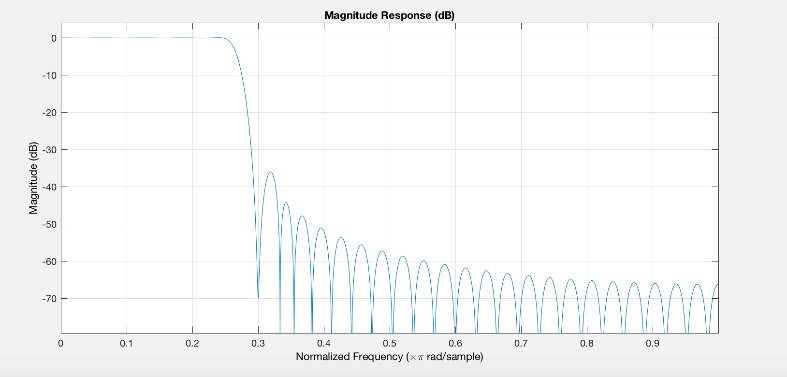
Stopband:[0 4096Hz];

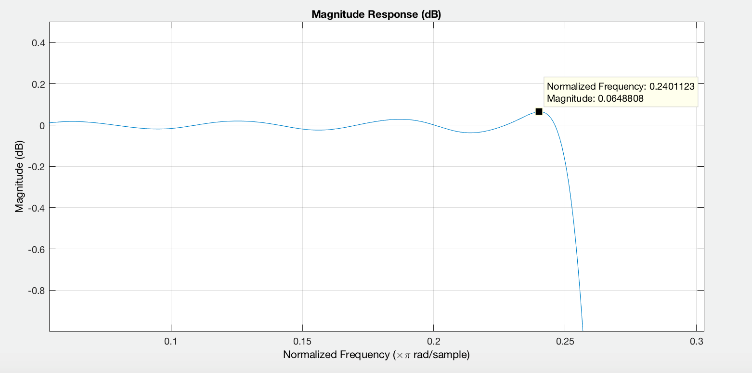
Passband ripple: 0.001;

Stopband ripple: 0.001;

Now the order of the highpass filter is 30.



Figure 12: the new notch filter bode diagram



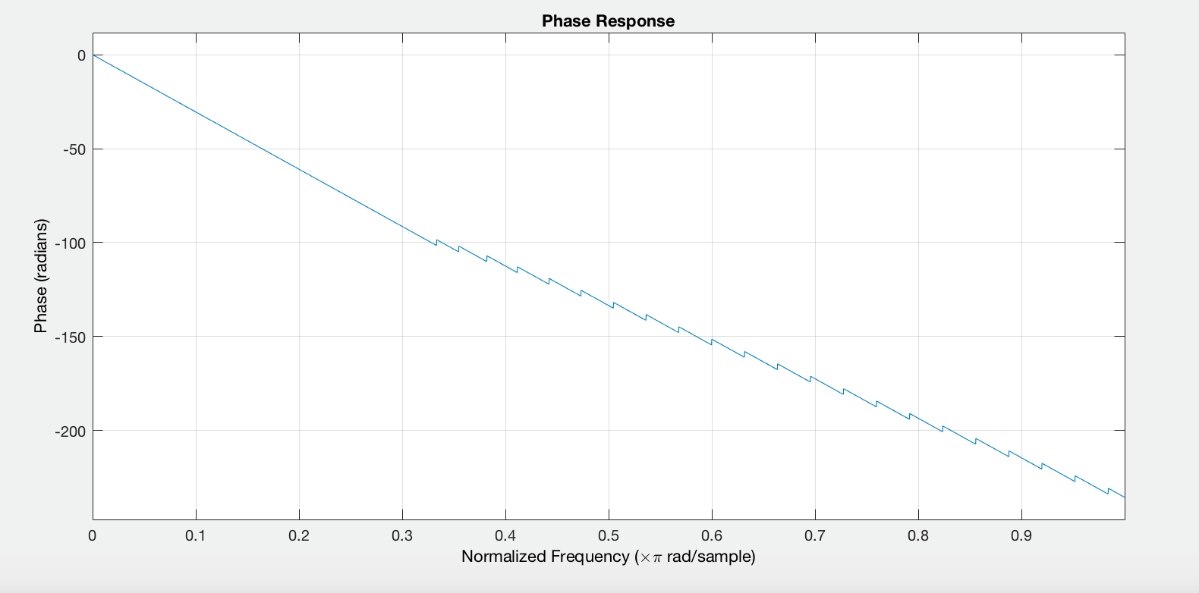
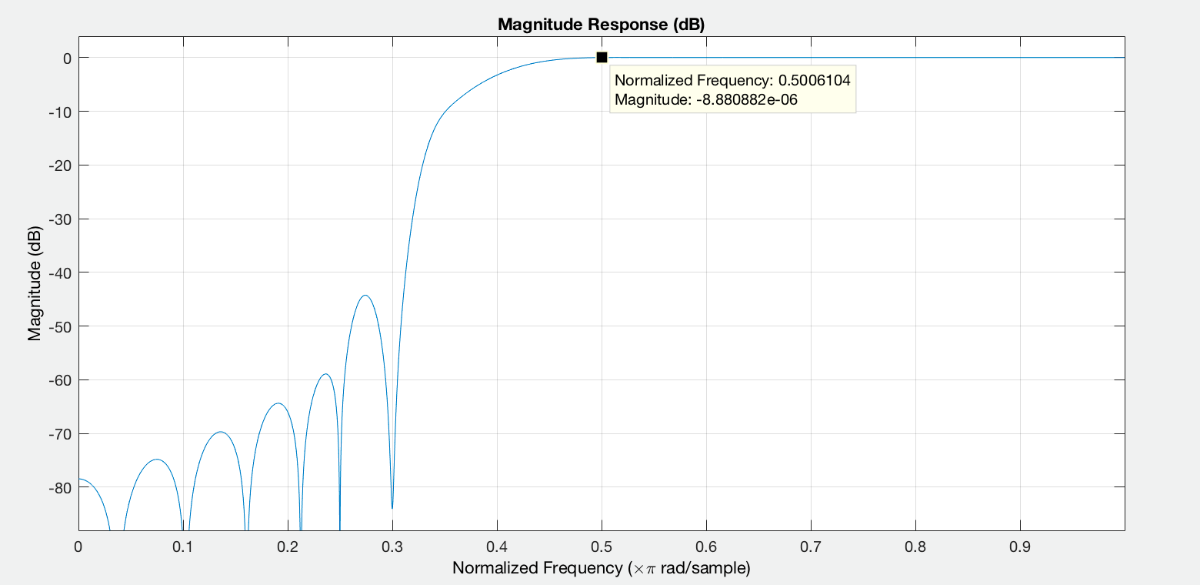


Figure 13: the response of



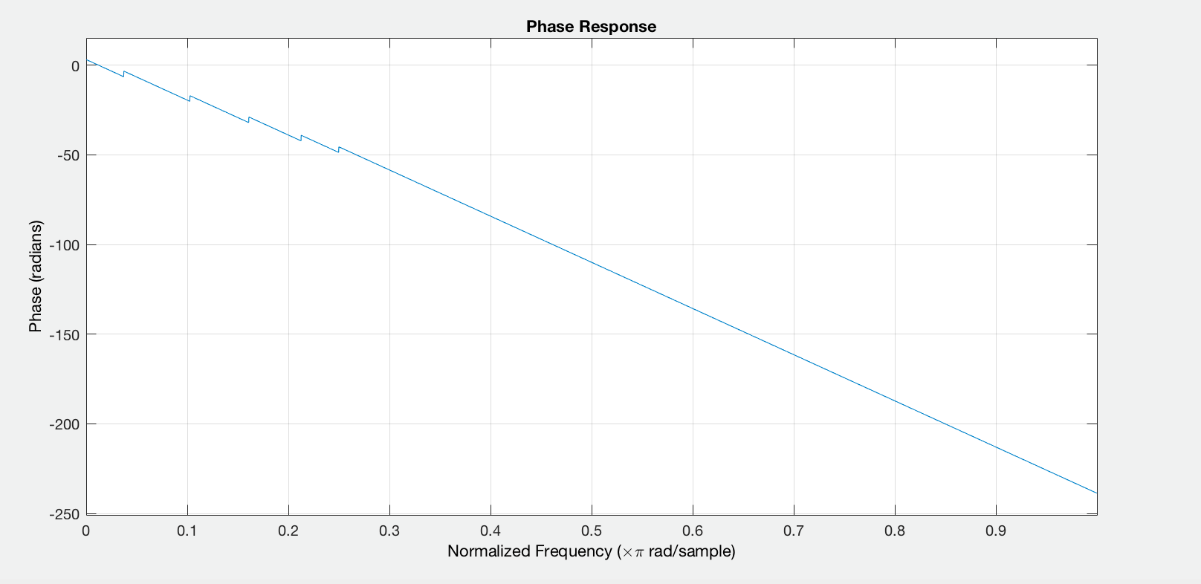
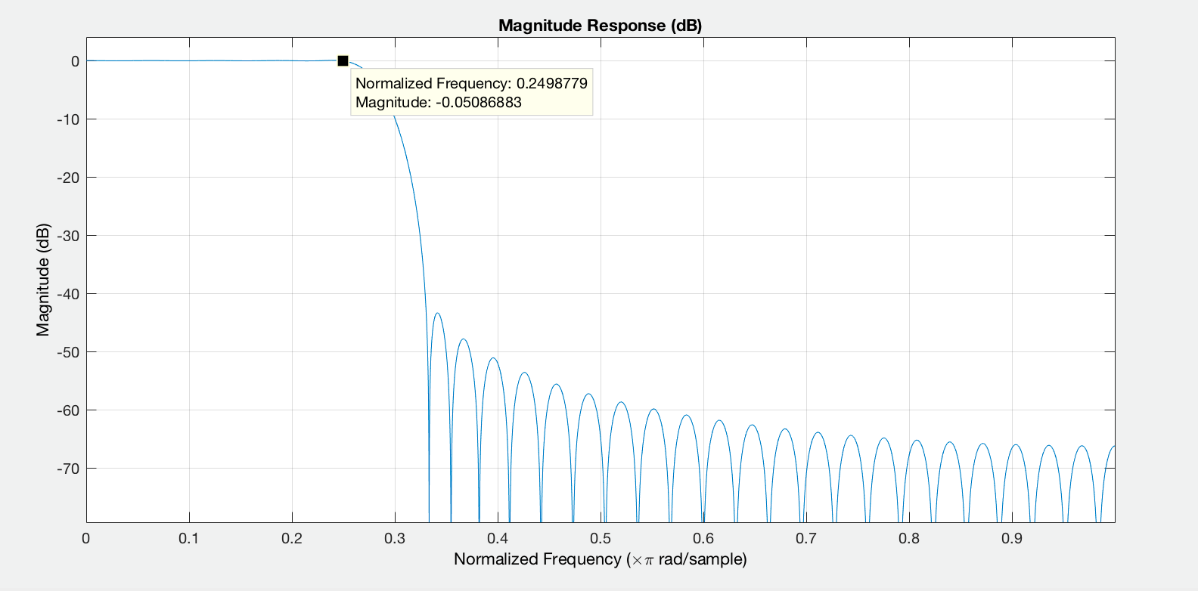


Figure 14: the response of cascade of filters A and B



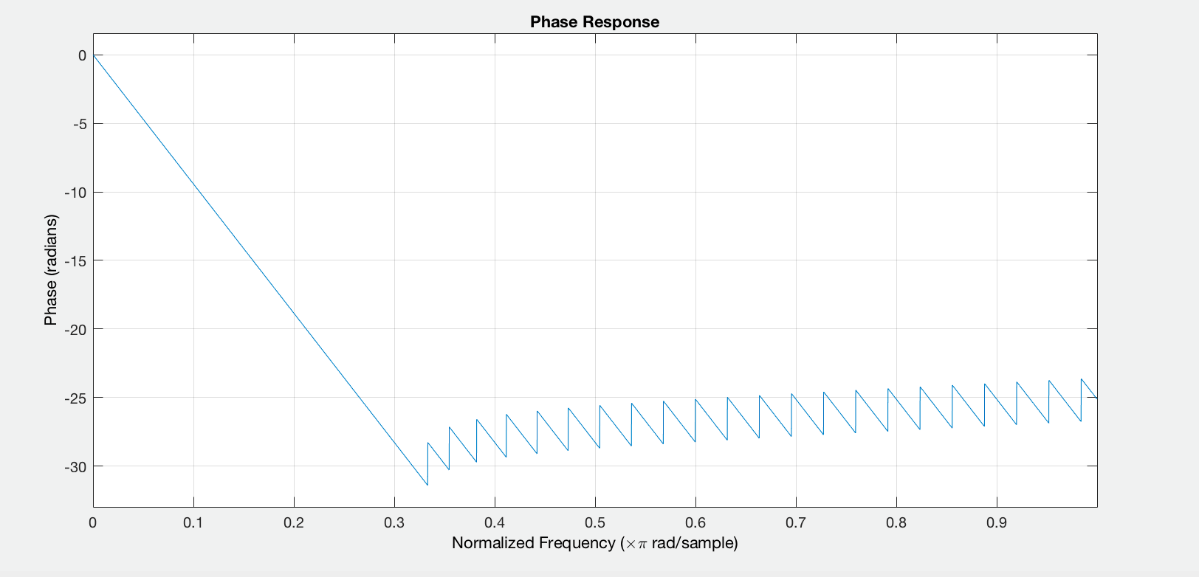


Figure 15: the response of filter in Block C

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

## IIR filter Design

**Block A: Second order Notch filter**

Central frequency

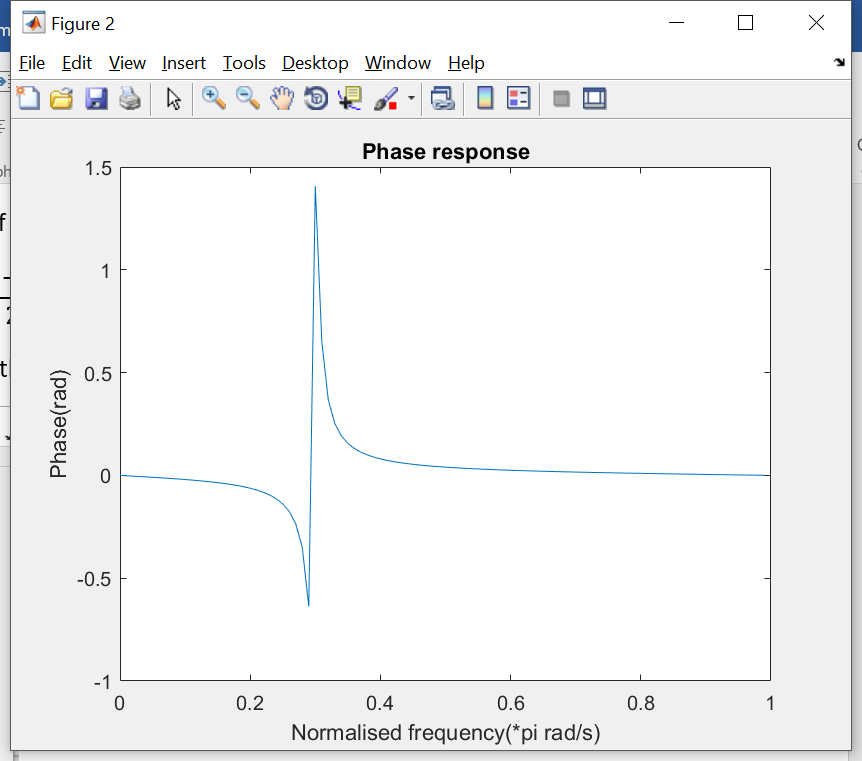
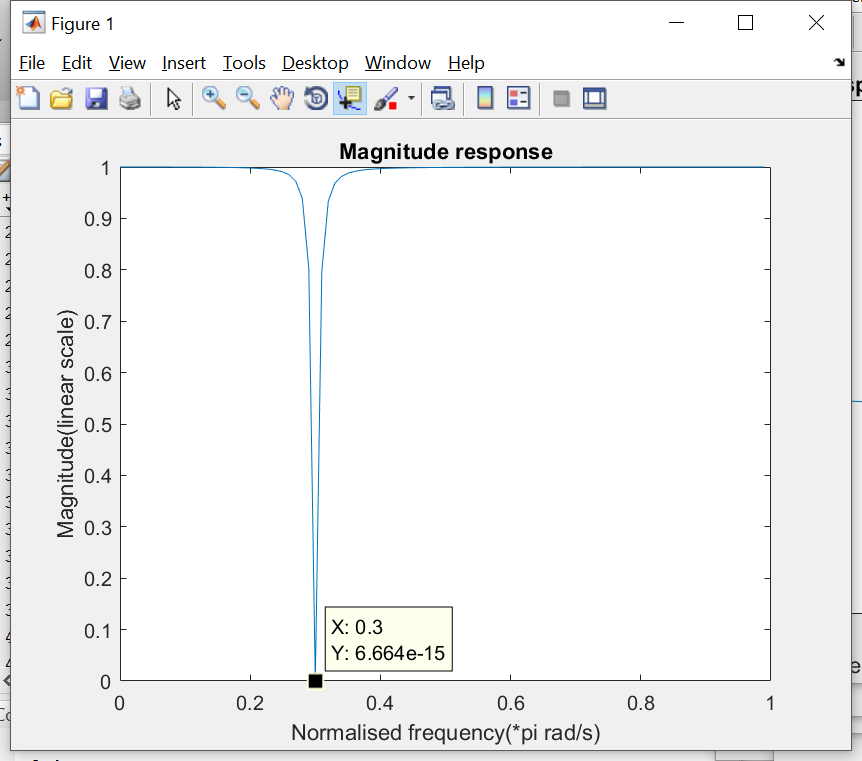
Filter parameters

By trial and error in Matlab, we found that when the bandwidth = 0.015, the gain of the filter and the distortion of the filter is the most suitable for our application, further analysis will be done in the later part of this report about this.

Therefore .

The notch filter has transfer function of

The magnitude and phase response of the filter has been plotted below



As the magnitude response shows, the gain of the filter at is 20log(6.664e-15) = -283.52dB << -60dB, therefore this filter will size up the design requirement.

The phase response of the block A filter is not linear, which is different from FIR filters.

**Block D: LPF**

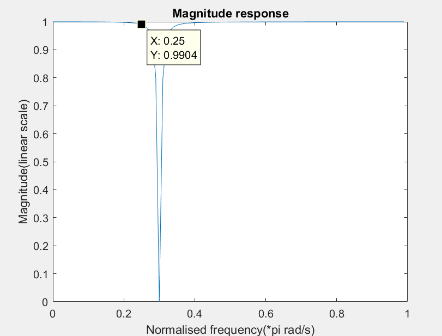
Passband: [0, 4096]

Stopband: (4096, inf)

Passband ripple: <4E-4

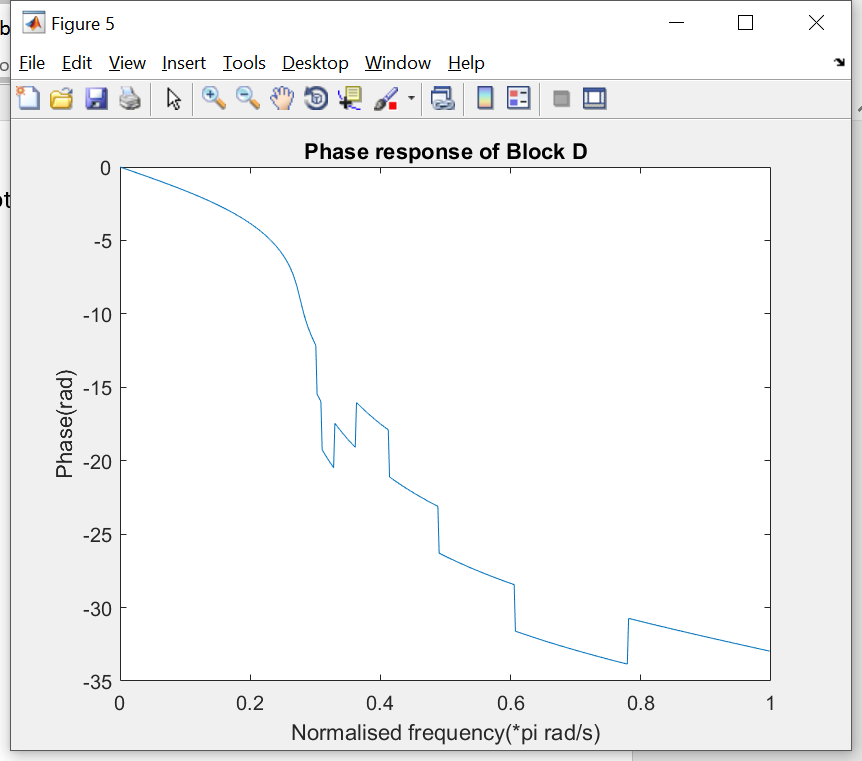
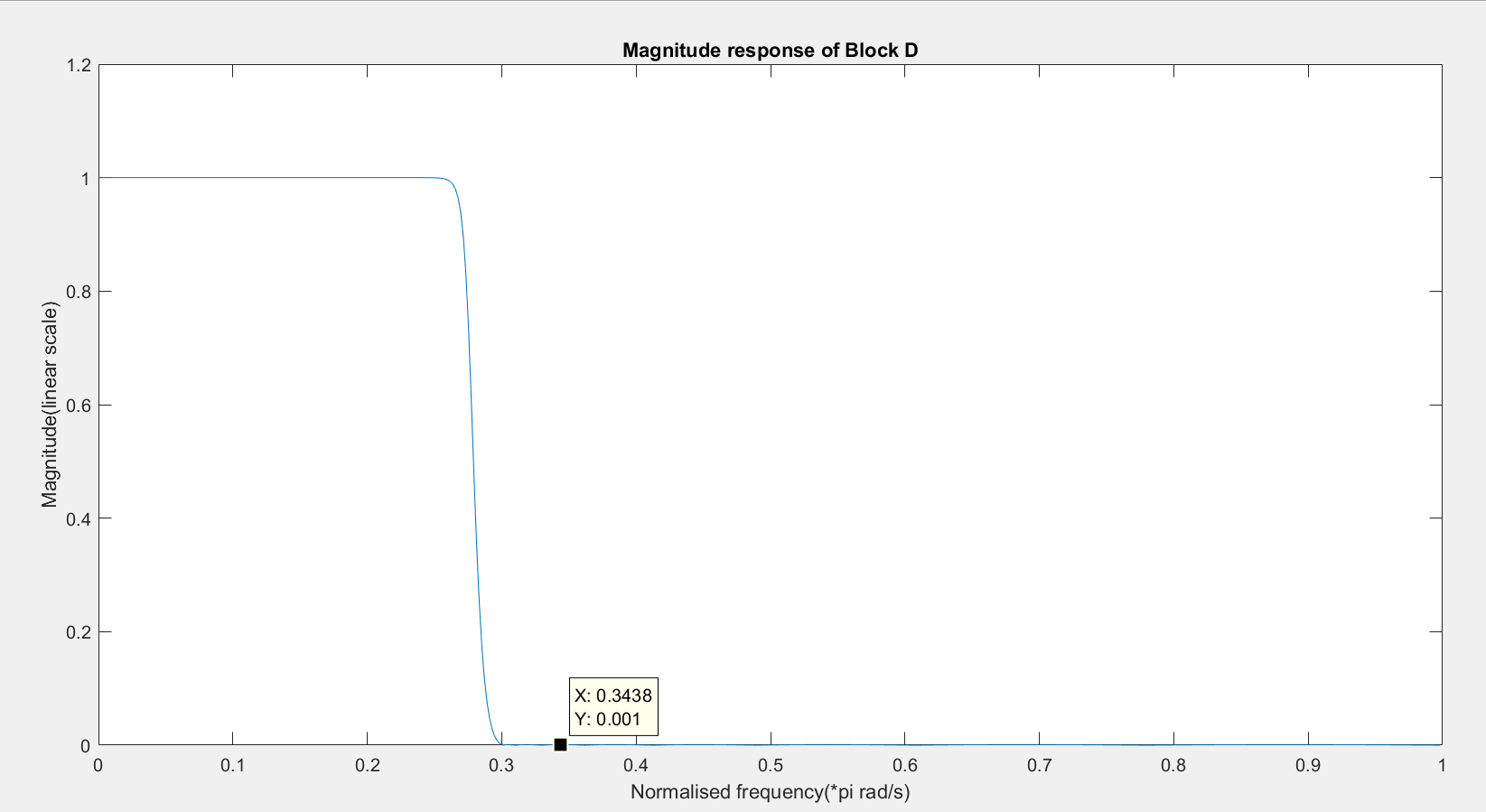
Stopband ripple: 0.001

Ripple and filter type analysis

Ripple in passband, cascading with Block A: <0.01

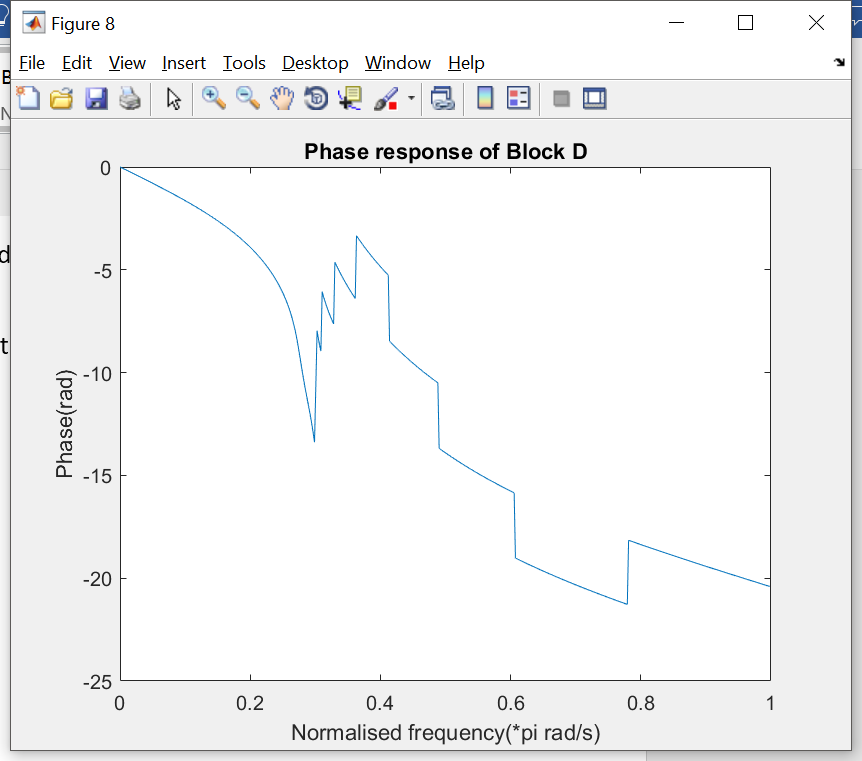
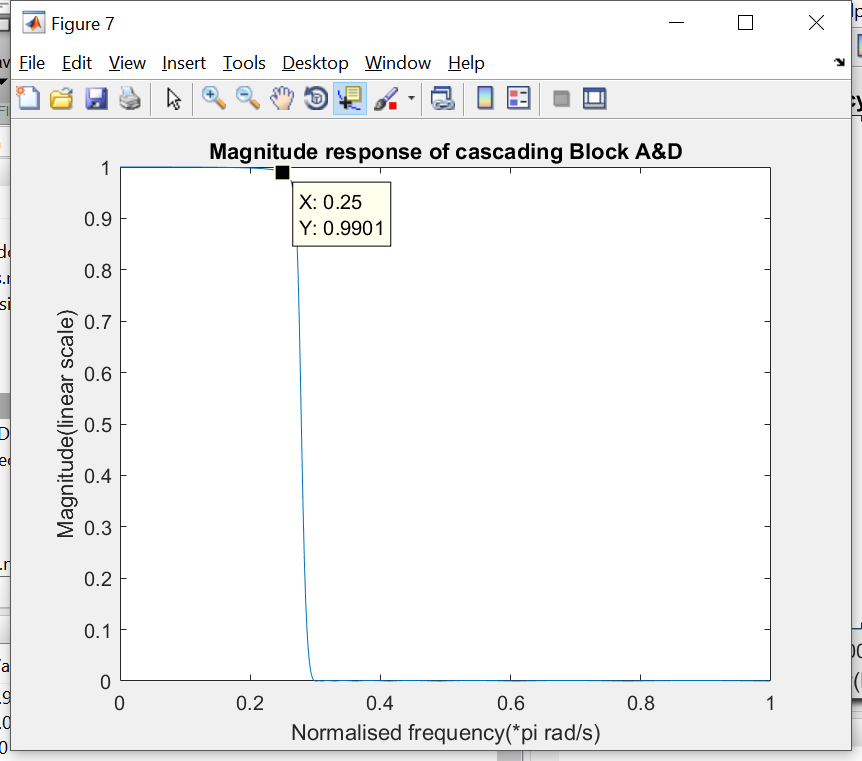
It was known from the magnitude response of the filter A that it will introduce a ripple at the passband edge ( = 0.25) of filter D, and the ripple has been estimated to be 1-0.9904 = 9.6E-3. Therefore, , rp\_D<0.000403877, rp\_D = 4E-4 was set to satisfy this condition. So, it’s preferably to have a flat passband from the low pass filter in this case. Therefore, a **Chebyshev Type 2 filter** has been implemented.

The frequency and phase response of the filter has been plotted in the diagram below



As the magnitude plot shows, the filter has a flat passband and stopband with ripple of 0.001, as we designed.

By cascading A and D together, we have phase and magnitude response as shown below



As analysed before, the passband ripple of filter A+D is 1-0.9901 = 9.9E-3 < 1%, therefore, the design of A and D satisfy the requirements.

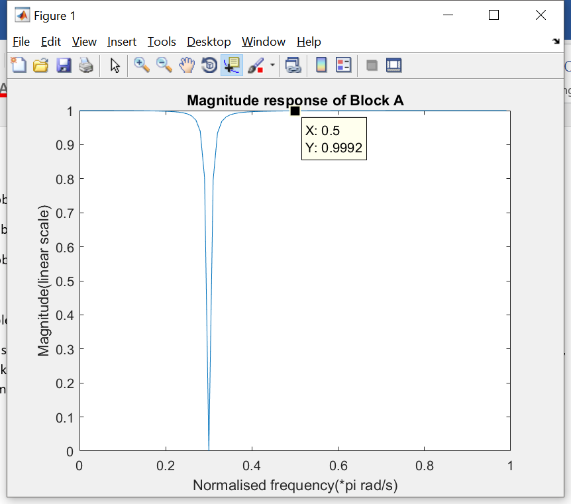
**Block B: HPF**

Passband: [8192, inf ), because the modulated signal has been shifted and centred at fc = 12288Hz, and the lower bound of this signal is 12288-bandwidth = 12288-4096 = 8192.

Stopband: [0, 8192)

Passband ripple: 4.5E-3

Stopband ripple: 0.001



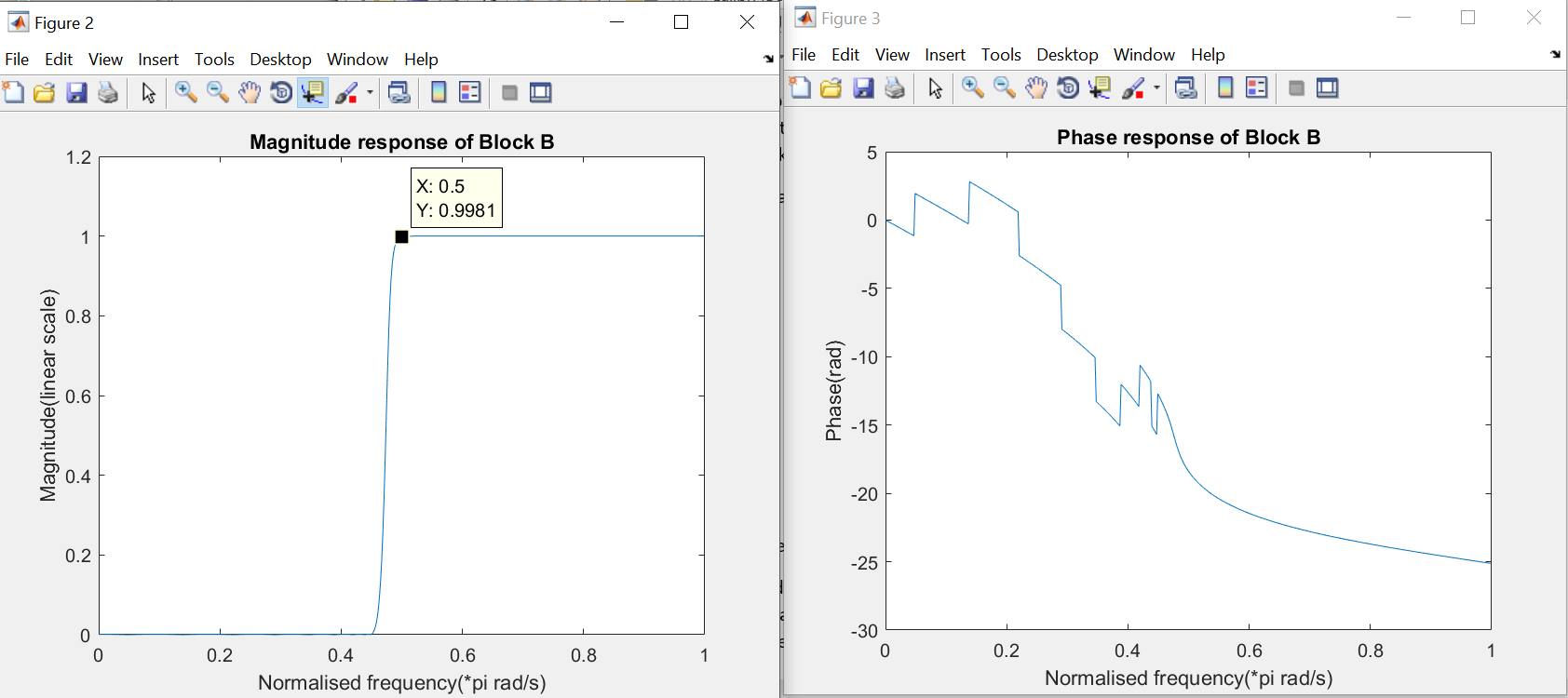
Ripple and filter type analysis

It has been required that cascading A B and C should have ripple less than 2% of the original signal, and knowing that the gain of block A at the passband edge of block B, f = 8192, i.e, 0.5 at normalised angular velocity, is 0.9992. Therefore, the ripple allowance for cascading B and C will be calculated as below.

For the convenience of calculation, we set , therefore,

Therefore, the passband ripple for Block B is 4.5E-3. Since a flat response in the passband is always the best option to minimise the ripple, therefore we still choose **Chebyshev Type 2 filter** to implement Block B.

The magnitude and phase response of the filter has been plotted as below.



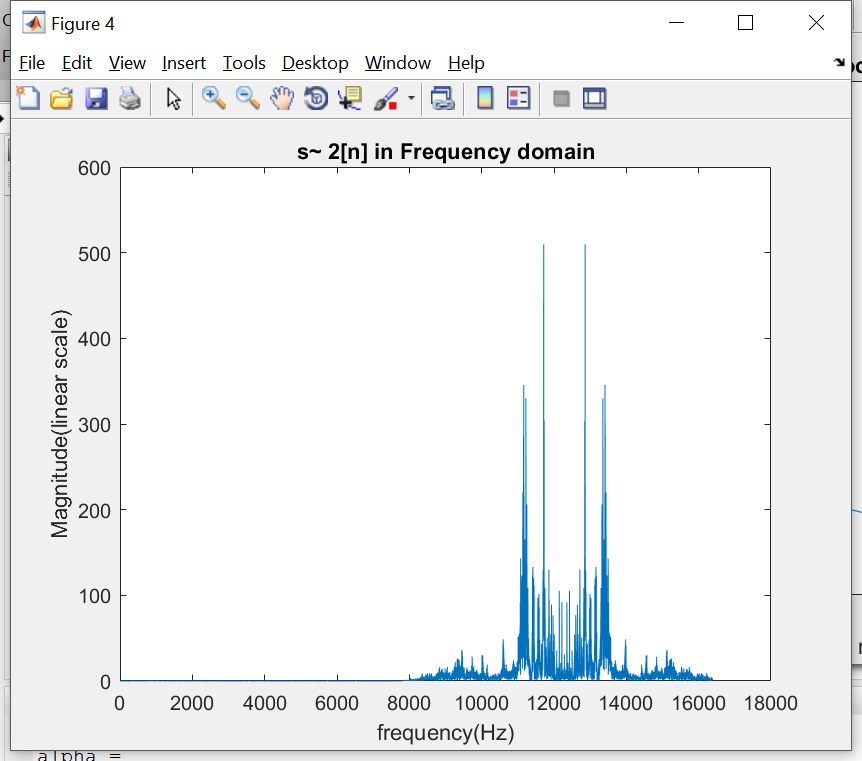
As expected, the stopband has ripple of 0.001 and the magnitude response for the passband is flat.

Further analysis on designing Block C

As the magnitude response show, the ripple for this filter at 0.5pi (passband edge) has gain of 0.9939, that means the ripple 1 - 0.9981 = 1.9E-3, recall the magnitude at the 0.5pi for block A is 0.9992, This means the ripple allowance for cascading block C is 1-0.9992\*0.9981\*(1-rp\_C)<0.01.

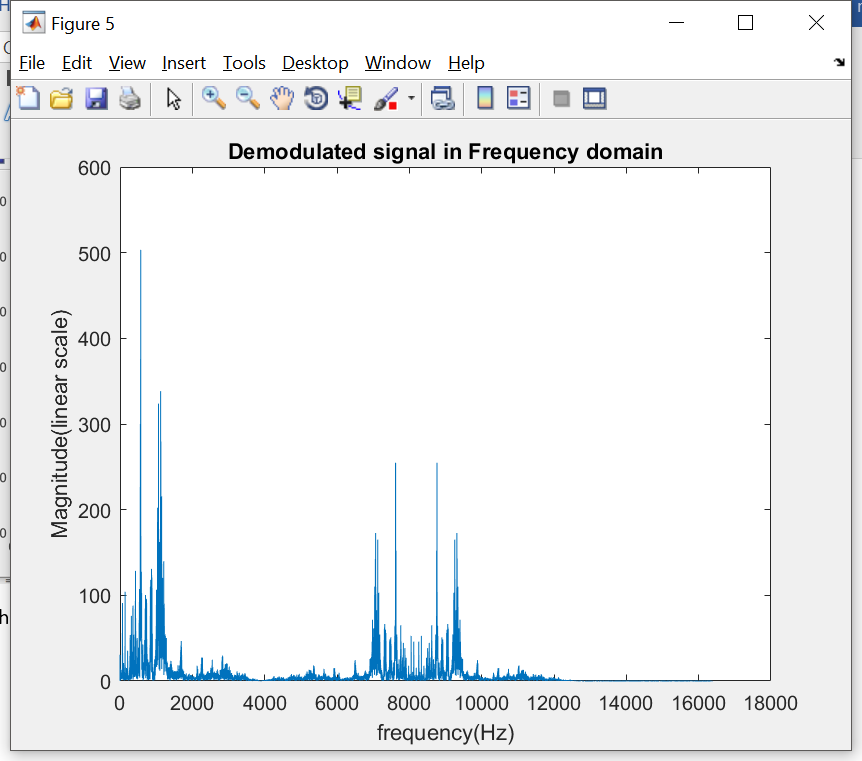
Rp\_C<0.00732128 = 7.32E-3. This is a sanity check for the result of Rp\_C<4.61E-3, it means that the performance of block B is better than expected.

Continue with the verification of block B, the signal after Block B HPF is shown below as



This is the modulated signal bandpass signal.

Using the same demodulation method, we have the signal as below show



**Block C: LPF**

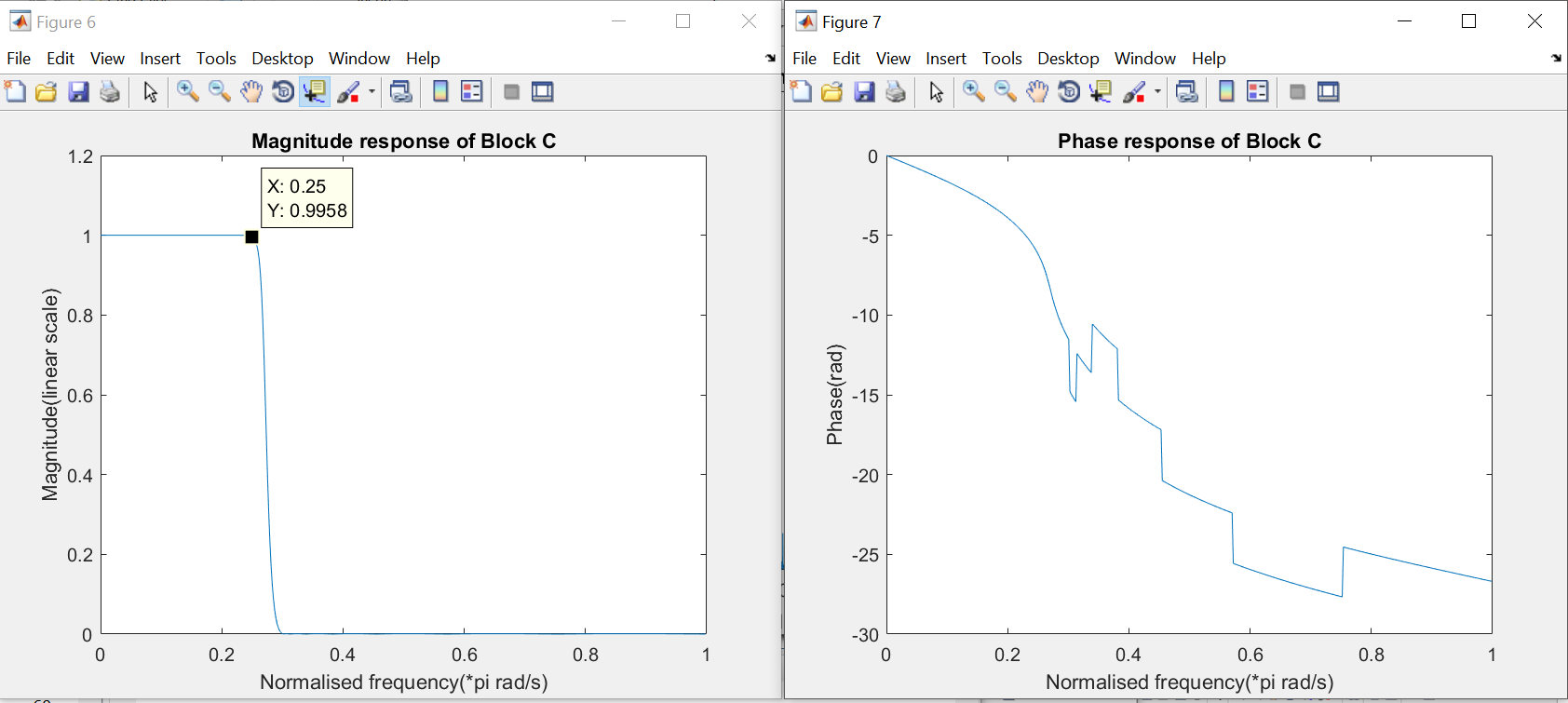
Passband: [0, 4096),

Stopband: (4096, inf)

Passband ripple: 4.5E-3, as discussed previously

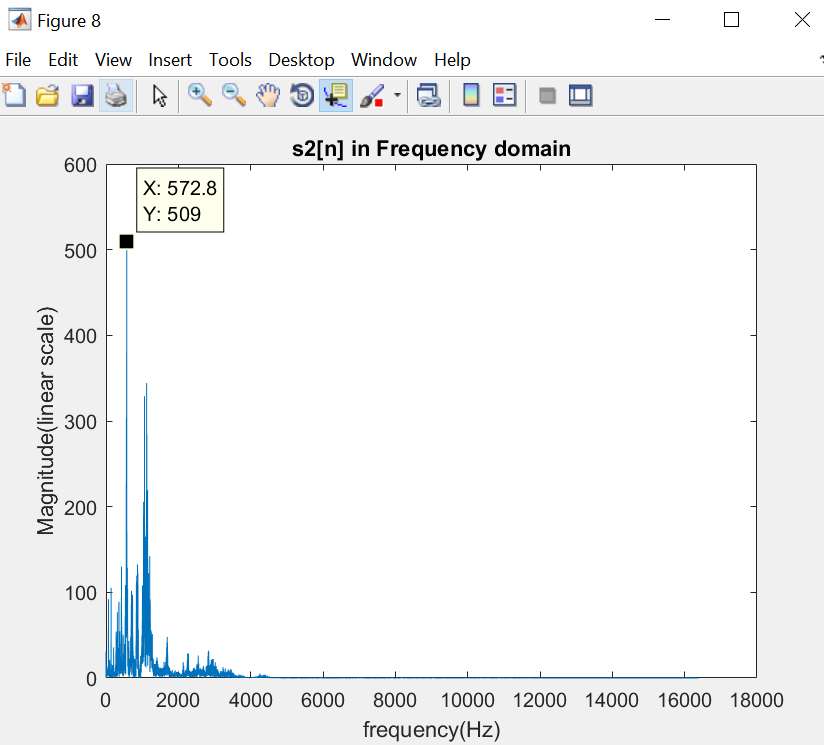
Stopband ripple: 0.001

As the diagram below is the magnitude and phase response of the block C lowpass filter



As expected,

Signal s2[n] after Block C,



The phase offset has been set to +pi/3 to obtain the original magnitude, by test and error.

2b. Minimise the filter orders

Its known that Elliptic filter has the narrowest transition band as well as the lowest required order among different type of filters.

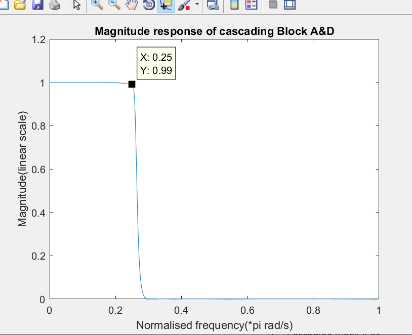
Therefore, we replace the Chebyshev type 2 filters in block B, C and D to complete this task.

Recall the design requirement for Block B, C and D are shown below

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Block B** | **Block C** | **Block D** |
| **Passband** | [8192, inf) | [0, 4096), | [0, 4096] |
| **Stopband** | [0, 8192) | (4096, inf) | (4096, inf) |
| **Passband ripple** | 4.5E-3 | 4.5E-3 | <4E-4 |
| **Stopband ripple** | 0.001 | 0.001 | 0.001 |
| **Filter Type** | Chebyshev Type 2 | Chebyshev Type 2 | Chebyshev Type 2 |
| **Minimum Order** | 18 | 15 | 17 |

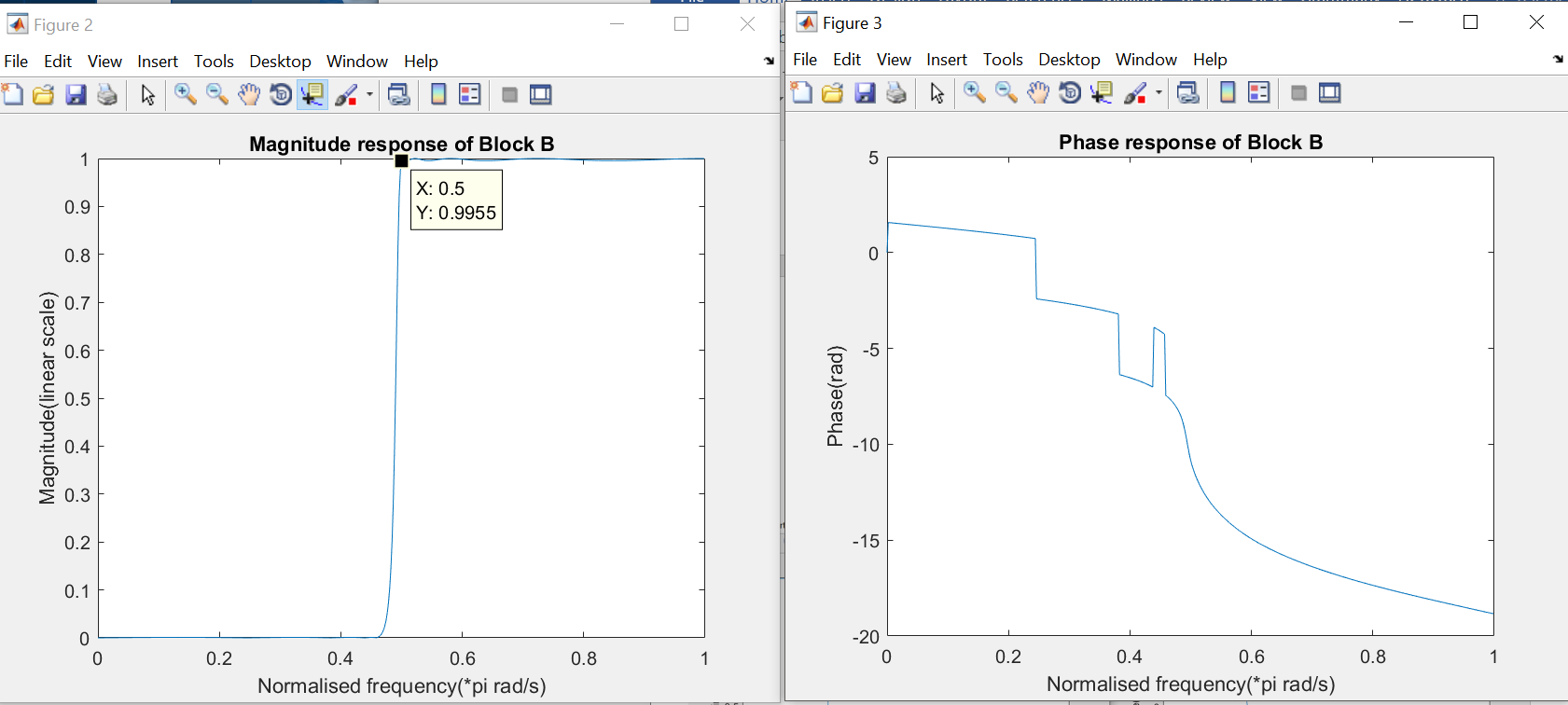
Optimised order design

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Block B** | **Block C** | **Block D** |
| **Filter Type** | Elliptic | Elliptic | Elliptic |
| **Minimum Order** | 9 | 8 | 9 |

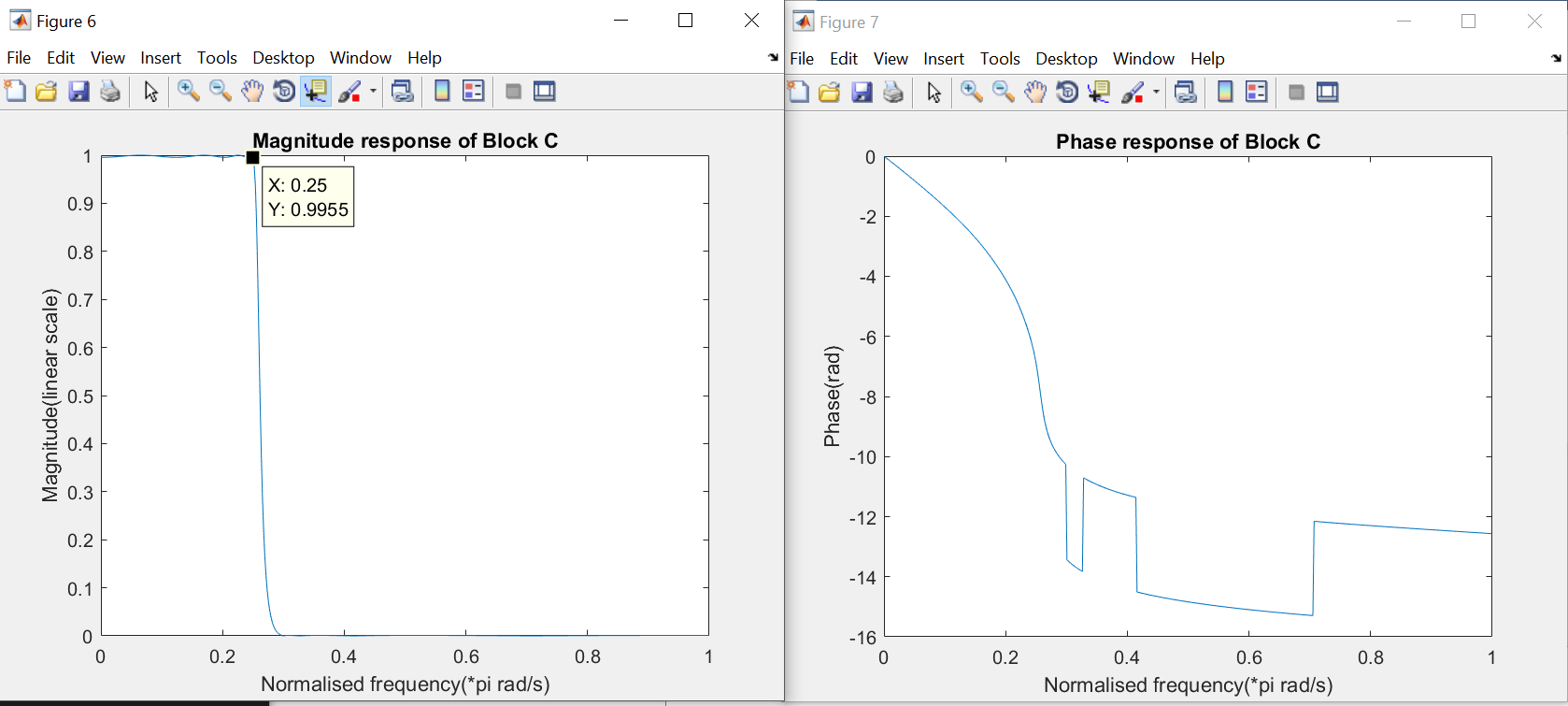
Requirement check

1. The filters are implemented to have minimum order
2. Cascading Block A and D has ripple of 1-0.99 = 0.1 at 0.25pi(the edge of passband)
3. The ripple of Block B has been estimated to be (1-0.9955), and that in C is (1-0.9955), therefore the total ripple of cascading A B and C is

Block B magnitude and phase response



Block C magnitude and phase response



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');