Iran University of Science and Technology

simulation of digital signal processing course (FIR Filter Design)

Seyed Amir Ali Mohaddes

Table of Contents

[Abstract 4](#_Toc116993093)

[**Examining the theory of the problem and presenting the relevant relationships** 4](#_Toc116993094)

[**Simulation method** 8](#_Toc116993095)

[**Simulation and presentation of results** 8](#_Toc116993096)

[Calculations of the number of multiplication and addition operations for a filter of length M and a signal to Length of N : 8](#_Toc116993097)

[Review part 1 9](#_Toc116993098)

[Examining the difference between passband and baseband 9](#_Toc116993099)

[Continue to review part 1 of the problem 10](#_Toc116993100)

[Review part 2 12](#_Toc116993101)

[Review of the third part 14](#_Toc116993102)

[Review of the fourth part: 15](#_Toc116993103)

[Review of the fifth part 16](#_Toc116993104)

[Review of the sixth part 19](#_Toc116993105)

[Review of the seventh episode 19](#_Toc116993106)

[Conclusion 22](#_Toc116993107)

[List of images and charts 23](#_Toc116993108)

# Abstract

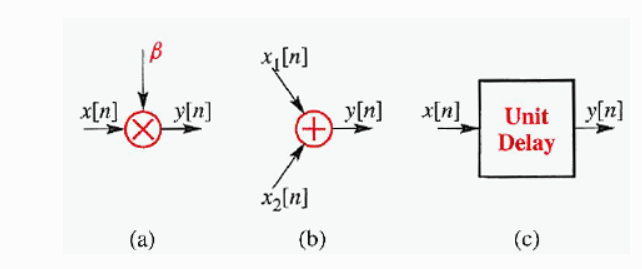
In this project, we are going to design an FIR filter for a noisy signal with a signal-to-noise ratio of 40. First, we count the number of operations required for filtering the signal, then we draw the signal and its fast Fourier transform, the noise with the function awgn ) to the signal. Then we design the filter. Finally, we perform decimation and design the filter again and report the results.

# **Examining the theory of the problem and presenting the relevant relationships**

In an FIR filter , the output is a linear combination of the input and past inputs and has no feedback from the output and is defined as follows:

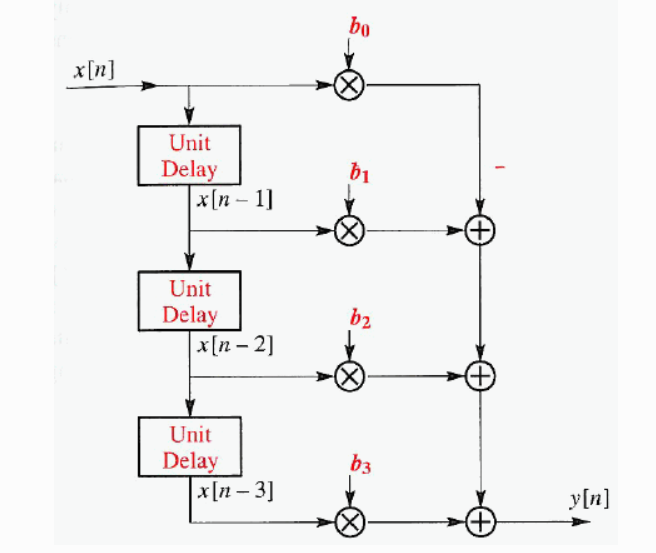
Finally, the goal is to determine the coefficients of this filter in a way to reach the desired goal.

FIR filter building blocks are: 1- Multiplier 2- Adder 3- Delay generator



resolution of R 1 basic systems in FIR filter

Below is the block diagram of an FIR filter :



resolution of R 2 Block diagram of a 3rd order FIR filter

for designing the FIR back filter , such as windowing, Fourier transform, frequency sampling, etc.

According to the demands of the design problem, it is supposed to [[1]](#footnote-1)be done by the Kaiser window method, so the concepts of the windowing method, especially the Kaiser window, are explained below.

H d (n ) to Title Response Sample A frequency filter unit Select an idea Consider Al with a linear phase:

because H d (n) In general, it has an infinite length We have to approximate it using an FIR filter . With the window design method, the desired filter using the window The sample response of the unit is designed :

In this relation, w(n) is a window with a finite length, which is equal to zero outside the interval 0 ≤ n ≤ N and is symmetric with respect to its midpoint :

The effect of the window on the frequency response can be obtained from the mixed convolution theory :

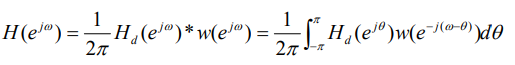
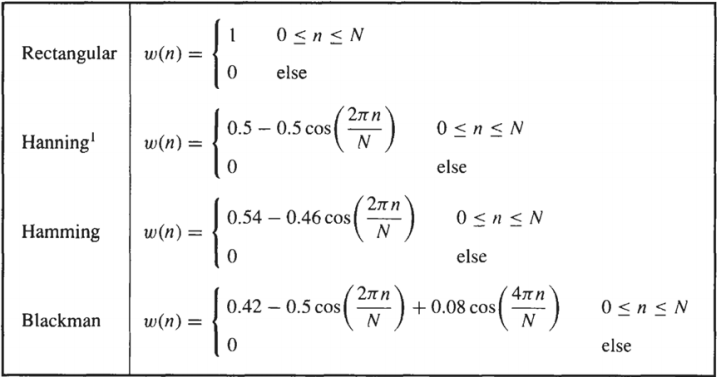
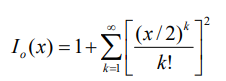
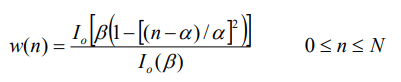


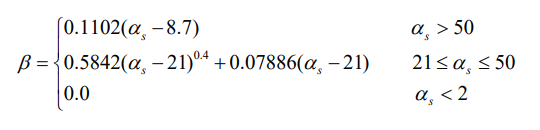
Table 1 some From window Hi Usual



In addition to these windows introduced in table 1 became Kiser, a new family from the window introduced which are described by the following relation :

that in this regard α=N/2 and I 0 Bessel function corrected It is the first type of bile that is used It is obtained from the expansion of the following power series Aide :

There are two empirical relations for using the Kaiser window in the design of FIR filters , which simplify the design more does The first is the relation of the attenuation coefficient of the stop band of the low filter. Go to \_ parameter β It relates :



and the second N to the transient bandwidth Stop band attenuation coefficient It relates :

# **Simulation method**

We perform the simulation in MATLAB software step by step according to the provided instructions, and the relevant calculations and answers to the questions follow:

# **Simulation and presentation of results**

### Calculations of the number of multiplication and addition operations for a filter of length M and a signal to Length of N :

Number of addition operations:

In order to create a complete overlap of the filter, an addition operation is performed on the M-1 signal , which actually determines only the delay. After N overlapping , the addition operation takes place. Considering the delay, the number of additions is equal to:

If we do not consider the delay, we have N addition operations

(The number of times the sigma index changes in convolution was considered as the number of addition operations.)

Number of multiplication operations:

If Takhbar is calculated, the number of multiplication operations is obtained through arithmetic expansion as follows. We assume that N > M :

:Delay

: full overlap

:Exit

Number of delayed multiplication operations

Number of multiplication operations without delay

### Review part 1

### Examining the difference between passband and baseband

In passband mode The signal is placed in a specific central frequency (modulated), but in baseband mode The signal is without modulation (baseband).

### Continue to review part 1 of the problem

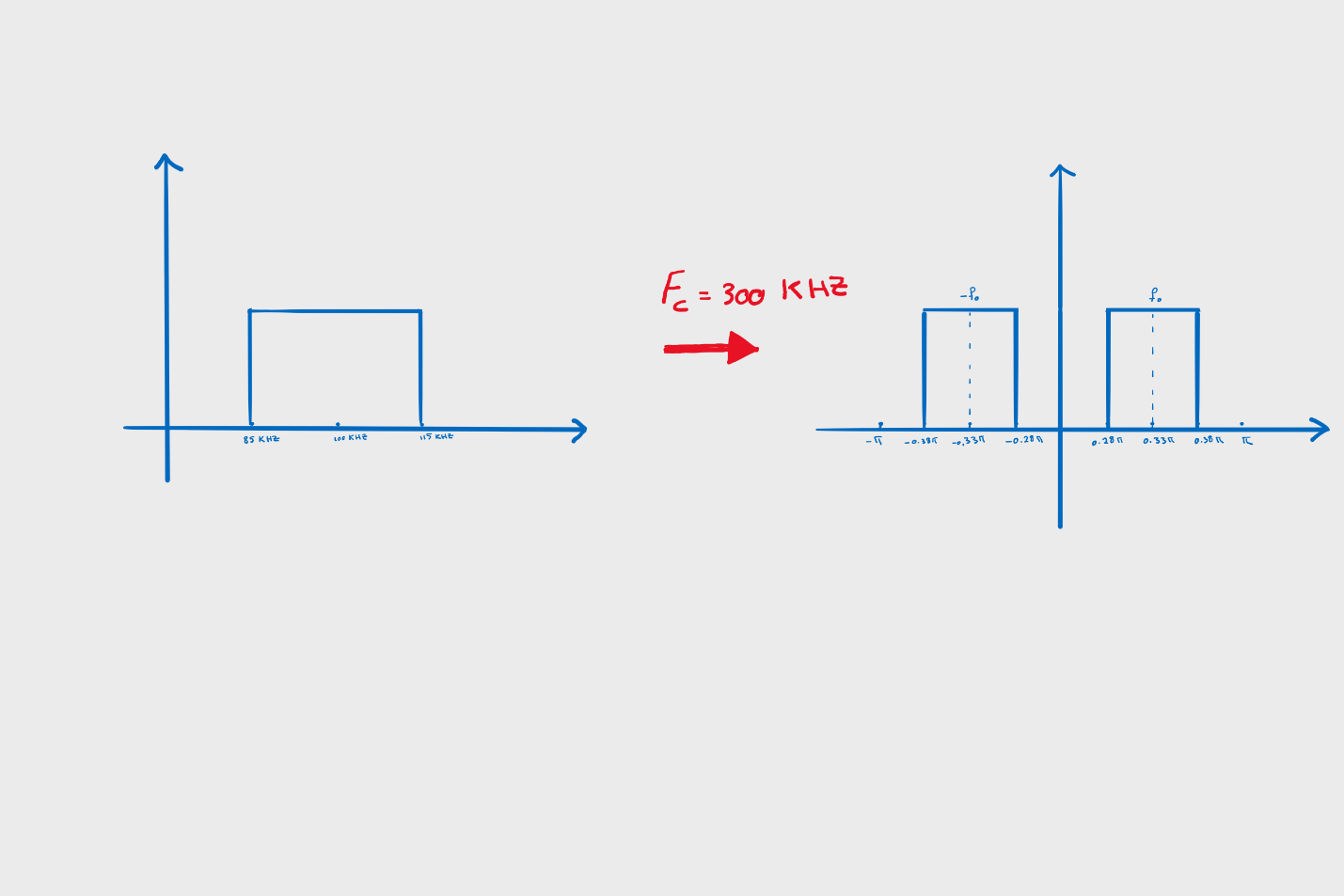


Image 3 At Shape Above Response The frequency of K \_ Letter \_ \_ It 's passing oh ten \_ Ah \_ At Interval (- pi,pi ) draw round ten \_ is \_

The hit response of this filter will be as follows:

So the frequency response in terms of H(w) will be as follows:

Now the image of the Fourier transform of this filter is calculated as follows:

Now that we have advanced the calculations theoretically, we will perform the simulation using the following code:

clc

clear

close all

n=-100:100;

w0=0.33\*2\*pi;

w=0.01\*2\*pi;

bn=(2\*w/ pi)\* (sin(w\*n)./(pi\*n)).\*cos(w0\*n);

bn( 101)=w/pi; Hospital \_

subplot(2,1,1 ),stem ( n,bn )

title( 'Impulse Response' )

Hw = fft (bn);

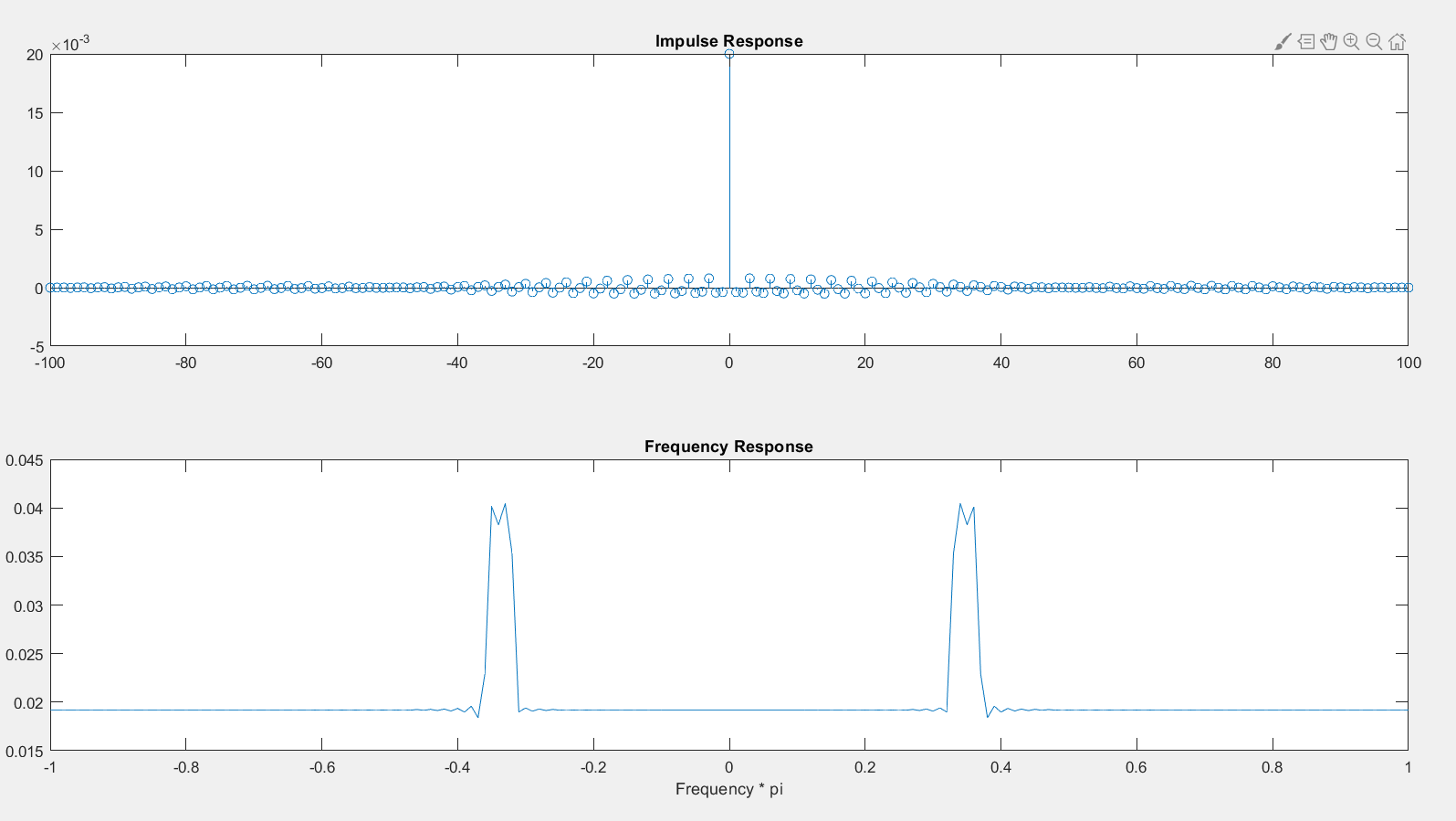
f=-1:1/(( length( Hw )-1)/2):1;

subplot( 2,1,2), plot( f,abs ( Hw ))

title( 'Frequency Response' )

xlabel ( 'Frequency \* pi' )

The impulse response diagram and the Fourier transform of the signal will be as follows



resolution of R 4 Impulse response diagram of signal B along with its Fourier transform diagram

### Review part 2

According to part 1, the Fourier transform of the signal was drawn according to Figure 4.

Due to the non-causality of the rectangular pulse signal, in general, this signal cannot be drawn correctly, and the spectrum drawn in matlab always has differences from our expected spectrum.

FFT command closer to our desired value.

The results obtained can be seen in the graph below.

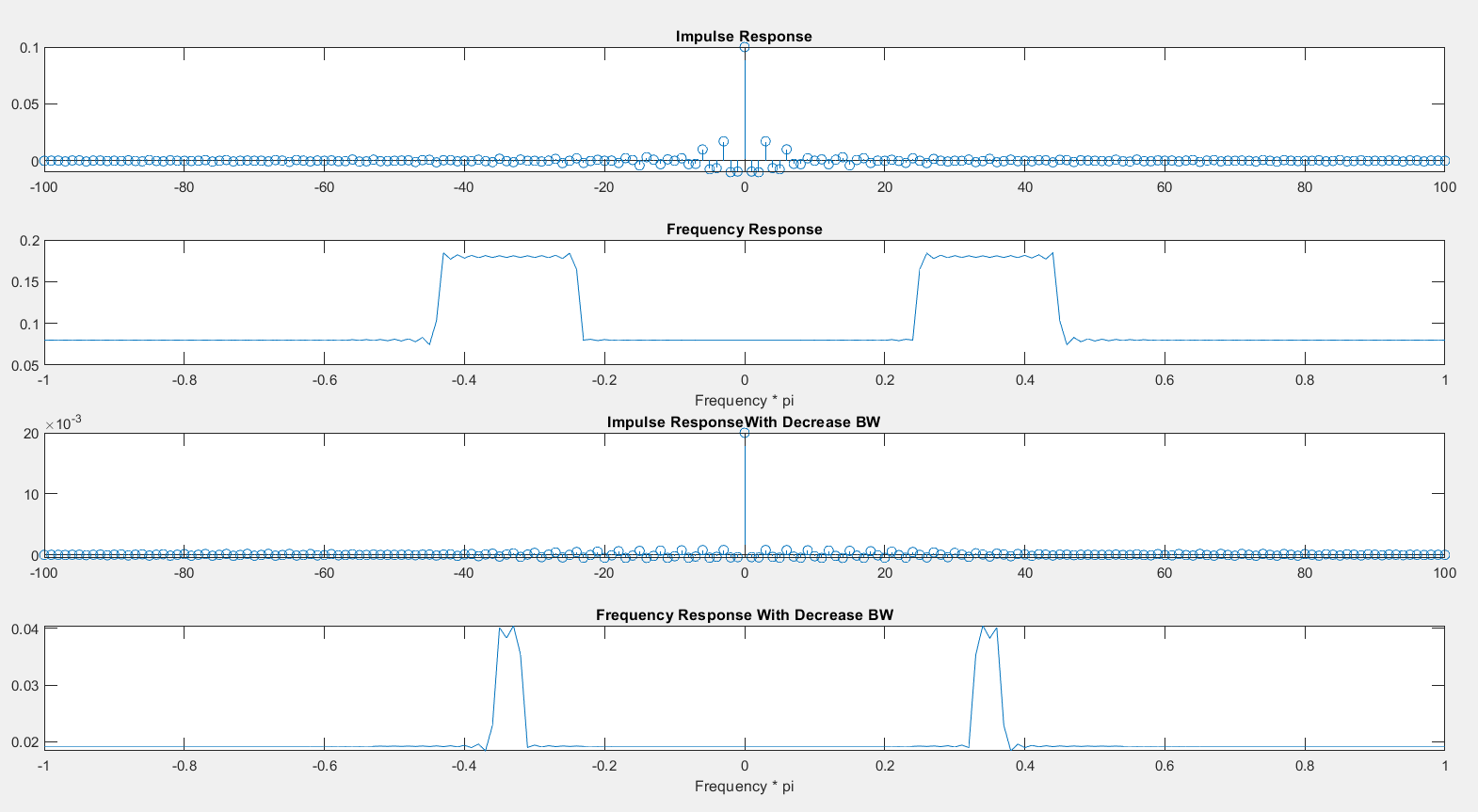


Image 5 Comparison of spectrum and impulse response of the signal after reducing the bandwidth with the initial state

The corresponding MATLAB code can be seen below

clc

clear

close all

n=-100:100;

w0=0.33\*2\*pi;

w=0.05\*2\*pi;

w2=0.01\*2\*pi;

bn=(2\*w/ pi)\* (sin(w\*n)./(pi\*n)).\*cos(w0\*n);

bn(101)=w/pi; Hospital

bn2=(2\*w2/ pi)\* (sin(w2\*n)./(pi\*n)).\*cos(w0\*n);

bn2(101)=w2/pi; Hospital

subplot(4,1,1),stem(n,bn)

title( 'Impulse Response' )

Hw=fft(bn);

Hw2=fft(bn2);

f=-1:1/((length(Hw)-1)/2):1;

subplot(4,1,2), plot(f,abs(Hw))

title( 'Frequency Response' )

xlabel( 'Frequency \* pi' )

subplot(4,1,3),stem(n,bn2)

title( 'Impulse ResponseWith Decrease BW' )

subplot(4,1,4), plot(f,abs(Hw2))

title( 'Frequency Response With Decrease BW' )

xlabel ( 'Frequency \* pi' )

### Review of the third part

#### awgn() function function

    The MATLAB communication toolbox has an internal function called awgn() , which can be used to add an incremental Gaussian white noise to obtain the desired SNR (signal to noise ratio). The main use of this function is to add AWGN to a clean signal with unlimited SNR In order to obtain a signal with an assumed SNR .

This function takes the input signal and the desired SNR as input, and the output is a signal to which Gaussian noise has been added.

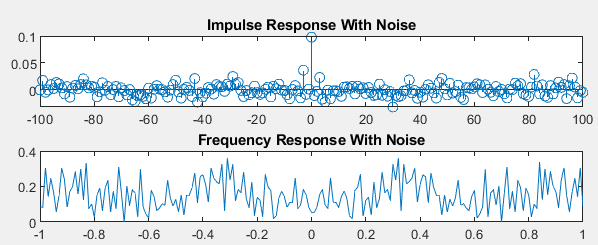


Image 6 hit response and the Fourier transform of the signal along with awgn noise

The relevant MATLAB simulation code:

clc

clear

close all

n=-100:100;

w0=0.33\*2\*pi;

w=0.05\*2\*pi;

bn=(2\*w/ pi)\* (sin(w\*n)./(pi\*n)).\*cos(w0\*n);

bn( 101)=w/pi; Hospital \_

NoiseBn =awgn(bn,40);

DS\_NoiseBn = NoiseBn ( 1:2:end );

Hw\_N = fft ( NoiseBn );

DS\_Hw\_N = fft ( DS\_NoiseBn );

f=-1:1/(( length( Hw\_N )-1)/2):1;

f2=-1:1/(( length( DS\_NoiseBn )-1)/2):1;

subplot(4,1,1 ), stem ( n,NoiseBn );

title( 'Impulse Response With Noise' )

subplot(4,1,2 ),plot ( f,abs ( Hw\_N ))

title( 'Frequency Response With Noise' )

### Review of the fourth part:

with the fdatool toolbox. The specifications of the filter are as follows:

Order=56

Wstop1=0.48

Wpass1=0.56

Wpass2=0.76

Wstop2=0.84

Astop1=Astop2=40 db

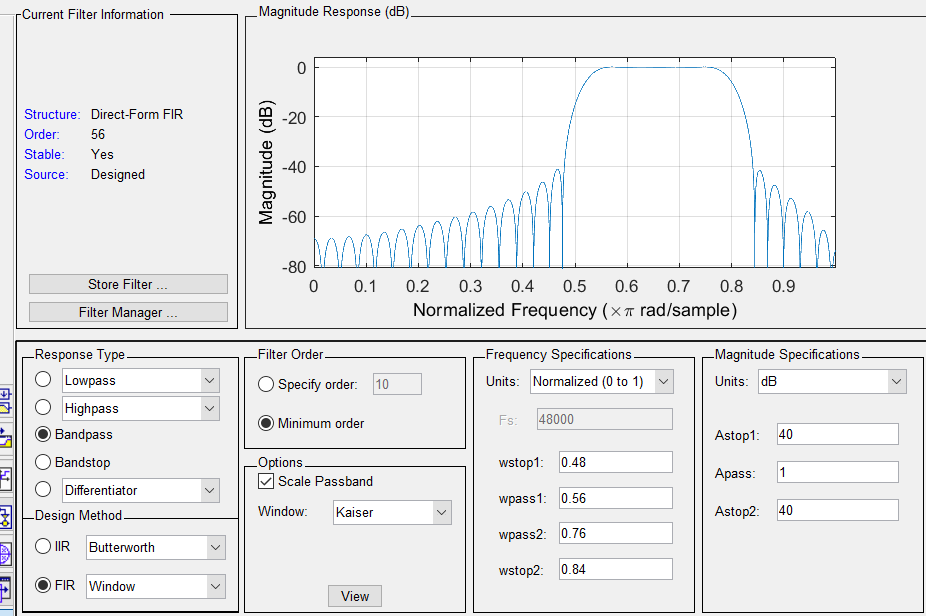


Image 7 filters designed in fdatool

### Review of the fifth part

First, we write and execute the following code in MATLAB, then we analyze the results:

clc

clear

close all

n=-100:100;

w0=0.33\*2\*pi;

w=0.05\*2\*pi;

bn=(2\*w/ pi)\* (sin(w\*n)./(pi\*n)).\*cos(w0\*n);

bn( 101)=w/pi; Hospital \_

NoiseBn =awgn(bn,40);

DS\_NoiseBn = NoiseBn ( 1:2:end );

Hw\_N = fft ( NoiseBn );

DS\_Hw\_N = fft ( DS\_NoiseBn );

f=-1:1/(( length( Hw\_N )-1)/2):1;

f2=-1:1/(( length( DS\_NoiseBn )-1)/2):1;

subplot(4,1,1 ), stem ( n,NoiseBn );

title( 'Impulse Response With Noise' )

subplot(4,1,2 ),plot ( f,abs ( Hw\_N ))

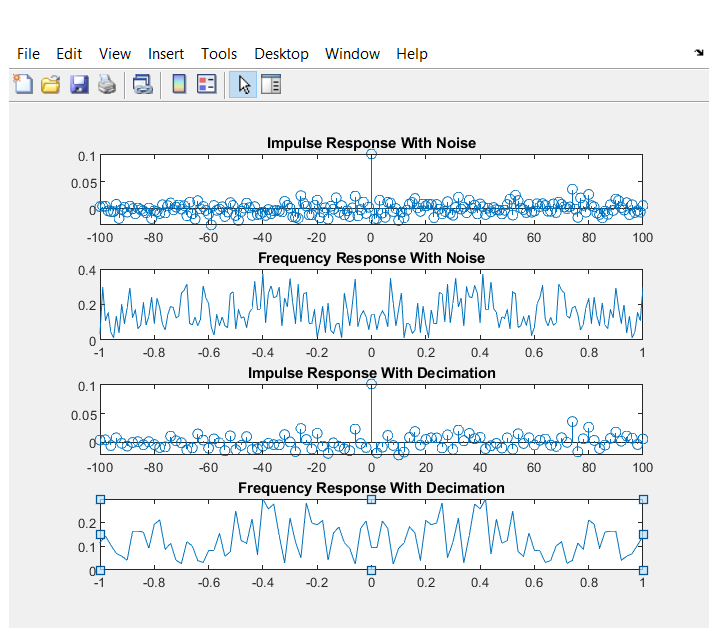
title( 'Frequency Response With Noise' )

subplot(4,1,3 ), stem (n(1:2:end), DS\_NoiseBn );

title( 'Impulse Response With Decimation' )

subplot(4,1,4 ),plot (f2,abs( DS\_Hw\_N ))

title( 'Frequency Response With Decimation' )



تصویر 8 طیف سیگنال که با ضریب 2 downsample شده است

Analysis

of the results

By performing the decimation operation On a signal with a factor of 2, as expected, the bandwidth of the signal is doubled.

Suppose that the signal x(t) is sampled at the rate T. Then we have:

Now, if this signal is at the rate of M downsample , we have:

### Review of the sixth part

In this case, with attention By halving the sampling rate and doubling the bandwidth, only passband is needed is doubled in the filter and the order of the filter does not change.

Order=56

Wstop1=0.38

Wpass1=0.46

Wpass2=0.86

Wstop2=0.94

Astop1=Astop2=40 db

### Review of the seventh episode

Coefficients of filters in two variables Coef\_Filter and Coef\_Filter2 have been saved.

Now the signal contains Gaussian white noise (section 3) as well as the decimate signal (Section 5) we pass through the designed filters Coef\_Filter and Coef\_Filter2 respectively and check the results.

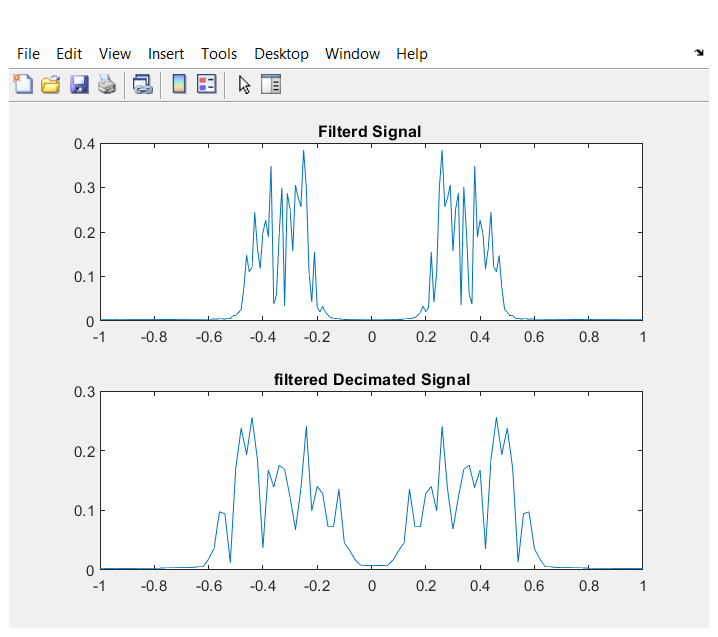


Image 9 Figure signal of step 3 and 5 after passing through the designed filters

The relevant MATLAB code

clc

clear

close all

n=-100:100;

w0=0.33\*2\*pi;

w=0.05\*2\*pi;

bn=(2\*w/ pi)\* (sin(w\*n)./(pi\*n)).\*cos(w0\*n);

bn( 101)=w/pi; Hopital \_

NoiseBn =awgn(bn,40);

DS\_NoiseBn=NoiseBn(1:2:end);

Coef\_Filter=[-5.28314437091259e-05,-0.00228983863622678,0.00415496601204029,-1.55126665749000e-18,-0.00602105960956121,0.00483544554379611,0.00016452453780058 4,0.00207934538532895,-0.00674641209654254,-0.00179325730571808,0.0173633493118839,-0.0153111953217375,-0.00312337750034839,0.00813947292652343,0.00287567598899596,0.00476712701010950, -0.0332922096283142,0.0315376044106861,0.0156287144823543,-0.0438612557552704,0.0166943762843342,-0.00193265939917407,0.0477567425359735,-0.0581809161140468,-0.0650250634442802,0.202959507396785,-0.131210039079736,-0.131087715097407,0.281589325730372,-0.131087715097407,-0.131210039079736,0.202959507396785,-0.0650250634442802,-0.0581809161140468,0.0477567425359735,- 0.00193265939917407,0.0166943762843342,-0.0438612557552704,0.0156287144823543,0.0315376044106861,-0.0332922096283142,0.00476712701010950,0.00287567598899596,0.00813947292652343,-0.00312337750034839,-0.0153111953217375,0.0173633493118839,-0.00179325730571808,-0.00674641209654254,0.002079345 38532895,0.000164524537800584,0.00483544554379611,-0.00602105960956121,-1.55126665749000e-18,0.00415496601204029,-0.00228983863622678,-5.28314437091259e-05];

Coef\_Filter2=[0.000162411235492125,0.00355729806723443,-0.00311893875684266,8.58353706454336e-19,-0.00451972798793100,0.00751193591331289,0.000505771403864582,-0.00137996542519361,-0.00669382686865763,-0.00199446584387669,0.0145748499670944,-0.00406340651682391,0.00218096041782039,-0.0248554627922029,0.0205987512329366,0.00324959089545704,0.0144021676114968,-0.0285386879134216, -0.0155068957287943,0.0287606613450254,0.0111901304239521,0.0257854081217173,-0.0966161610750685,0.0419414272706542,-0.00823209040488874,0.160196934896505,-0.168628407776185,-0.209113661650226,0.478961939013600,-0.209113661650226,-0.168628407776185,0.160196934896505,-0.00823209040488874,0.0419414272706542,-0.0966161610750685,0.0257854081217173,0.0111901304239521,0.0287606613450254,- 0.0155068957287943,-0.0285386879134216,0.0144021676114968,0.00324959089545704,0.0205987512329366,-0.0248554627922029,0.00218096041782039,-0.00406340651682391,0.0145748499670944,-0.00199446584387669,-0.00669382686865763,-0.0013799654251 9361,0.0005571403864582,0.0075119359128128 9, -0.004519727983100,8.58353706454545436e-19, -0.003113189386666666666666666666666666666666666666666666666666524444

tic

y = filter(Coef\_Filter,1,NoiseBn);

toc

DS\_NoiseBn = NoiseBn ( 1:2:end );

tic

y2 = filter( Coef\_Filter2,1,DS\_NoiseBn);

toc

Hw\_F = fft (y);

DS\_Hw\_F = fft (y2);

f=-1:1/(( length( Hw\_F )-1)/2):1;

f2=-1:1/(( length( DS\_NoiseBn )-1)/2):1;

subplot( 2,1,1), plot( f,abs ( Hw\_F ))

title( ' Filtered Signal' )

subplot( 2,1,2), plot(f2,abs( DS\_Hw\_F ))

title( 'filtered Decimated Signal' )

Also, the time spent in the process of filtering the signals is as follows:

First signal:

Elapsed time is 0.000045 seconds

Second signal:

Elapsed time is 0.000016 seconds

# Conclusion

The conclusion of each part of the simulation has been reviewed while presenting the simulation results.

# List of images and charts

1. Kaiser [↑](#footnote-ref-1)