Illinois Institute of Technology

Digital Signal Processing II

ECE 569

Project 1

Filter Design and Signal Filtering

Karthikeyan Kumar A20376688

Due April 6th 2017

Contents

Objectives:

- Designing a Linear phase, FIR (Finite Impulse Response) filters using Least square error minimisation, and optimum equiripple using Remez algorithm for the given filter specifications, and inferring the response
- Designing a IIR (Infinite Impulse Response) butterworth, and chebyshev filters for the given filter specifications, and inferring the response
- An overall comparison on the results of the different filters designed
- Applying the designed filters in order to process the Signal and compare the BER (Bit Error Rate) of the signal before and after application of filter.
- Inference of the filter characteristics from the BER values.

Filter Specifications:

FIR, Linear Phase, bandpass filter with the specifications:

Pass band range = 0.25π to 0.5π

Stop band range = 0 to 0.2π and 0.55π to π

Pass band ripple = 0.1

Stop band attenuation = 0.05

Description:

Design of FIR Filter:

Least-squares error minimisation:

Design of the filter involves minimising the square error, by employing some weighting function. A system of linear equations are solved to obtain the minimum square error. It is carried out by using an in-built MATLAB function 'firls'. This function takes in a series of inputs and returns a row vector consisting of the filter coefficients as the output. Inputs are namely the order of the filter, the frequency range for the filter operation, amplitude, weight.

Z=firls(order,freq,amplitude,weight)

Order – order of the filter. (Can be even or odd)

Freq – Frequency range of the filter (represented as an input row vector with increasing values)

Amplitude – An input row vector of zeros and ones describing which range of frequencies are pass band and stop band

Weight – An input row vector governing the attenuation of the ripples in pass band and stop band

Z – An output row vector consisting of (order+1) filter coefficients.

Note: Freq and Amplitude vector must be of same length and must be even. The filter coefficient Z is symmetric in nature.

Result

Implementing the procedure with the filter specifications, and order M= 75 provides the desired filter response. Upon decrementing the order of the filter, it is observed, there is also a need to alter the band edges to obtain a favourable response. A satisfactory response was observed with order M = 49 coupled with the first and second edge frequencies 0.2075 and 0.543 respectively. It can be inferred from the magnitude, phase response represented from Figure 1 through Figure 3 that the filter satisfies the linear phase criterion and the response is well within the desired range.

Optimum Equiripple design

This filter design method uses the Remez Algorithm and Chebyshev approximation to compute the optimum values by minimising the difference in the desired and actual frequency response. The design produces equal ripples in both stop and pass band, hence named Equiripple. 'firpm' is the MATLAB function that is employed, which takes in a series of inputs and returns the filter coefficients as the output in the form of a row vector.

Z=firpm(order,freq,amplitude,weight)

Order – order of the filter.

Freq – Frequency range of the filter (represented as an input row vector with increasing values)

Amplitude – An input row vector of zeros and ones describing which range of frequencies are pass band and stop band

Weight – An input row vector governing the attenuation of the ripples in pass band and stop band

Z – An output row vector consisting of (order+1) filter coefficients.

NOTE: Freq and Amplitude must be of same length and even number of values. The output coefficients are symmetric.

Result

The filter response for order M = 65 satisfies the design requirement, it is noted that as the value of the order is reduced, there is an adverse change in the value of the ripples in pass and stop bands. So the values of the cut-off frequencies are manipulated such that the ripples satisfy the design requirement. After successive trails, for M=55 and f1= 0.2 and f2= 0.5475, Figure 5 and 6 provide the magnitude and phase response proving how the filter meets the design criteria and is linear phase.

Designing of IIR filters:

Butterworth Design:

The Butterworth filter is maximally flat in the pass band and has monotonic in the stop bands. The flat response is traded off with the steepness of the filter. Like any IIR filter design, it is initially designed in the analog domain and later converted into digital through bilinear transformation and frequency warping. The MATLAB function 'butter' carries out this procedure and provides the digital filter coefficients as the output. The input to the filter are order of the filter, cut-off frequency, and filter type.

[x,y]=butter(order,cuttoff, filtert)

Order – Input defining the order of the filter employed

Cutoff - An input row vector containing the cut off frequencies in increasing fashion.

Filtert - Input defines the type of filter to be designed lowpass, highpass, bandpass, or bandstop.

x - An output row vector containing numerator of the transfer function of the filter.

y - An output row vector containing denominator of the transfer function of the filter.

NOTE: length of x and y is (2*order) +1. The true order of the filter is twice the order value for bandpass and bandstop filter, and same as order value for lowpass and highpass.

Result

The filter steepness is traded for the flat response at the passband region, for N = 35 the filter's response meets the design specifications. Upon decreasing the order, it is observed that there is a rapid change in the filter transition band, and steepness of the filter. By careful alteration of the cut-off frequencies, and steady decrease of the filter order we obtained the best response with N=17 and f1=0.24 and f2=0.51. The resulting magnitude response in Figure 8 supports this claim.

Chebyshev Design:

Chebyshev filters are a type of IIR filters that produce monotonicity and equiripple in specific region depending on the type of filter used. Type 1 filter has ripple in the passband and Type 2 has ripple in stop band. The filter is first designed in the analog domain and converted to the digital front. The MATLAB function 'cheby2' generates the filter transfer function depending on the inputs.

[x,y]=cheby2(order,attenuation,freq,filtert)

Order – An integer scalar input defining the order of the filter

Attenuation – A scalar input value defining the attenuation value in dB.

Freq – Scalar row vector input containing the stop band edge frequencies in increasing fashion

Filtert – Input string defines the type of filter to be designed lowpass, highpass, bandpass, or bandstop.

x - An output row vector containing numerator of the transfer function of the filter.

y - An output row vector containing denominator of the transfer function of the filter.

Result

Type 2 Chebyshev filter meets the design specifications for the given band edge frequencies with order N=10. On reducing the order slowly and modifying the stop band frequencies, an agreeable response was observed for N=6 and f1=0.21 and f2=0.54. The magnitude response for these values is plotted in Figure 11. It is evident from the plot that the design is satisfactory.

Comparison of designs

The least squares error approximation design of FIR filter minimises overall square error between desired and designed response, while Equiripple design focus mainly on the need to reduce the error between the design and desired response rather than the overall error square as in least square filter.

In least square the ripples are not equally spaced in the stop and pass band thereby providing different attenuation, and distortion for various regions of the signal. Whereas equiripple ensures equal attenuation in all the bands. Overall the Least squares can be used in cases where there is a need for a sharp transition, uneven attenuation in the pass band and fairs poorly in systems requiring a steady attenuation in the pass bands. Whereas equiripple filter fairs good in situation needing equal attenuation over the entire frequency with a slower transition.

Butterworth filters produce maximally flat response on the pass band, therefore they're used in scenarios requiring zero attenuation in the pass band, but trades off with a longer transition band making it a poor choice for applications requiring sharp transitions. Whereas Chebyshev filters have a quick transition band making them quite useful in places requiring sharper transition, but they produce ripples in the pass band or stop band depending on the filter type, making them a poor choice in cases requiring least deviation from the unit value in the pass or stop band.

The type of filter design used depends on the application we wish to employ them in, some of the key difference between FIR and IIR are

- FIR is linear phase, IIR have no particular phase
- FIR systems are more stable than IIR
- FIR can be implemented directly in digital domain while IIR are derived from analog domain.

BER performance:

From table 1 it is seen that the Equiripple filter design provides considerable clean up against corruption due to noise, while Butterworth provides the least. As the attenuation produced by the equiripple is optimal, and uniform that filter works better against noise than other filter. Presence of ripples in the pass band plays a major role in determining the filter's response to noise corruption. As the presence of noise scales up the incoming signal's DC value, and ripples in the passband help in reducing those values. Yet another reason why the filters with ripples perform better than the ones without them. The slope of the transition band determines the frequencies allowed in between stop and pass band.

All filter design provide exceptional filtering in signals corrupted by interference. Proper selection of stop band and phase aid in tackling interference. Interference is the distortion due to the successive channel signal which might alter the phase of the signal, resulting in mutual cancellation, hence it becomes mandatory to design filter with proper phase characteristics to avoid mismatch while decoding the signal.

The filter phase delay for FIR is obtained from the symmetry condition, if proper phase is not employed, we won't achieve desired BER for even a clean signal. But IIR filters have irregular phase, therefore it must be found by trial and error.

The figure 13 to 16 correspond to Least Square approximation design, starting from clean, noise, interferer, and both. Followed by Equiripple design from figure 17 to 20, Butterworth figure 21 to 24 and chebyshev from figure 25 to 28.

Filter Design Optimisation:

All the filter designs, except for Butterworth seem to produce a BER of 3.5% or lower, by modifying the attenuation value. The attenuation value plays a major factor in cancelling out the noise of the channel. Upon changing the values of the stop band ripple, the BER of least square was reduced. This is a direct result of the noise being attenuated by the filter. Stop band ripple = 0.75, f1=0.249 and f2=0.501.

In the case of equiripple, changing only the stop band proves unfruitful, as the weighting function depends on both stop and pass band attenuation. Stop band attenuation = 0.8 passband = 0.09. f1= 0.2 f2=0.5475.

Chebyshev filter produced significantly lower BER on altering the stop band attenuation and a slight variation in the cut-off frequencies. Stop band attenuation = 20.0206 dB. f1=0.2495 f2 = 0.5115.

But as the Butterworth filter is maximally flat in the pass band and monotonic at stop bands, we can't control the nature of the attenuation to a higher degree as in other filter designs. Altering the order of the filter and the cut off frequencies is a viable option for Butterworth. The resultant BER values are presented in table 2.

APPENDIX

Function explanation:

Freqz:

Computes the frequency response of the given filter by taking filter coefficients and order as the input.

[x,y]=freqz(b,a,n)

b – Input row vector containing the numerator polynomial coefficients

a – Input row vector containing the denominator polynomial coefficients

n – Input scalar containing the order of the filter.

x – Output frequency response of the filter returned as a row vector.

y - Output angular frequencies of the filter returned as a row vector.

Phasez:

Computes the phase response of the given filter from the coefficients and order.

[x,y]-phasez(b,a,n)

b – Input row vector containing the numerator polynomial coefficients

a – Input row vector containing the denominator polynomial coefficients

n – Input scalar containing the order of the filter.

x – Output phase response of the filter returned as a row vector.

y - Output frequency of the filter returned as a row vector.

Figure:

Used to create a new figure window to plot the response.

Contents


```
clc; clear all;
```

Least Square Filter

```
f = [0 0.2075 0.25 0.5 0.543 1]; % Set the bandpass frequencies
a=[0 0 1 1 0 0];                 % Amplitude values at each region
w=[0.05,0.1,0.05];               % Scaling values at each region
Mls=49;                          %49 [[0 0.2075 0.25 0.5 0.543 1]
b=firls(Mls,f,a,w);              % Coefficients of Least square filter
N = 1024;
z = [0:2*N]*pi/N;
[H,W]=freqz(b,1,z);              % Magnitude and phase response
figure();
box_plot();
plot(W,abs(H));                  % Log-scale Magnitude response Plot
title('Filter Magnitude Response log scale');
ylabel('|H(w)|');
xlabel('w[rad/samples]');
set(gca,'YScale','log')
axis([0 1*Lw 0.0001 Htop]);
figure();
box_plot();
plot(W,abs(H));                  %Magnitude response Plot
title('Filter Magnitude Response');
ylabel('|H(w)|');
xlabel('w[rad/samples]');
figure();
phasez(b);
title('Filter phase Response');
ylabel('Phase(radians)');
xlabel('w[rad/samples]');
```

Optimum Equiripple

```
M=55;
f1=[0 0.2 0.255 0.5 0.5475 1]; % frequencies
b1=firpm(M,f1,a,w);            % Coefficients of filter
[H1,W1]=freqz(b1,1,z);
figure();
box_plot();
plot(W1,abs(H1));              % Log-scale Magnitude response Plot
title('Filter Magnitude Response log scale');
ylabel('|H(w)|');
xlabel('w[rad/samples]');
set(gca,'YScale','log')
axis([0 1*Lw 0.0001 Htop]);
figure();
box_plot();
```

```
plot(W1,abs(H1));                % Magnitude response Plot
title('Filter Magnitude Response');
ylabel('|H(w)|');
xlabel('w[rad/samples]');
figure();
phasez(b1);
title('Filter phase Response');
ylabel('Phase(radians)');
xlabel('w[rad/samples]');
```

Butterworth

```
%%Butterworth
Mbutter=17;                          % Orded of the filter
[b2,a2] = butter(Mbutter,[0.24 0.51],'bandpass'); %Coefficients
%0.2425 0.51, 0.24 0.51 12
[Hb,Wb]=freqz(b2,a2,z);
figure ();
box_plot();
plot(Wb,abs(Hb));                    % Log-scale Magnitude response Plot
title('Filter Magnitude Response log scale');
ylabel('|H(w)|');
xlabel('w[rad/samples]');
set(gca,'YScale','log')
axis([0 1*Lw 0.0001 Htop]);
figure ();
box_plot();
plot(Wb,abs(Hb));                    % Magnitude response Plot
title('Filter Magnitude Response');
ylabel('|H(w)|');
xlabel('w[rad/samples]');
figure();
phasez(b2,a2);
title('Filter phase Response');
ylabel('Phase(radians)');
xlabel('w[rad/samples]');
```

Chebyshev

```
Mcheby=6;             %order of the filter
[b3,a3] = cheby2(Mcheby,26.1206,[0.21 0.54],'bandpass'); %Coefficients
[Hc,Wc]=freqz(b3,a3,z);
figure ();
box_plot();
plot(Wc,abs(Hc));                    % Log-scale Magnitude response Plot
title('Filter Magnitude Response log scale');
ylabel('|H(w)|');
xlabel('w[rad/samples]');
set(gca,'YScale','log')
axis([0 1*Lw 0.0001 Htop]);
figure ();
box_plot();
plot(Wc,abs(Hc));                    % Magnitude response Plot
title('Filter Magnitude Response');
ylabel('|H(w)|');
xlabel('w[rad/samples]');
figure();
phasez(b3,a3);
```

```
title('Filter phase Response');
ylabel('Phase(radians)');
xlabel('w[rad/samples]');
```

Published with MATLAB® R2014b

Contents


```
clc;
```

Least Square Filter

```
fprintf('Response of the Least Square Filter\n');
ECE569Project(b,[1],(Mls-1)/2,'clean');
ECE569Project(b,[1],(Mls-1)/2,'noise');
ECE569Project(b,[1],(Mls-1)/2,'interferer');
ECE569Project(b,[1],(Mls-1)/2,'both');
```

Remez Algorithm

```
fprintf('Response of the Optimum Equiripple\n');
ECE569Project(b1,[1],(M-1)/2,'clean');
ECE569Project(b1,[1],(M-1)/2,'noise');
ECE569Project(b1,[1],(M-1)/2,'interferer');
ECE569Project(b1,[1],(M-1)/2,'both');
```

Butterworth

```
fprintf('Response for the Butterworth filter\n');
z=25;
ECE569Project(b2,a2,z,'clean');
ECE569Project(b2,a2,z,'noise');
ECE569Project(b2,a2,z,'interferer');
ECE569Project(b2,a2,z,'both');
```

Chebyshev

```
fprintf('Response for the Chebyshev type 2 filter\n');
y=6;
ECE569Project(b3,a3,y,'clean');
ECE569Project(b3,a3,y,'noise');
ECE569Project(b3,a3,y,'interferer');
ECE569Project(b3,a3,y,'both');
```


Published with MATLAB® R2014b

PART C

```
clc; clear all;
%%least Square finished
f = [0 0.249 0.25 0.5 0.501 1]; % Set the bandpass frequencies Works pakka!!
a=[0 0 1 1 0 0];                % Amplitude values at each region
w=[0.075,0.1,0.075];              % Scaling values at each region
Mls=49;
b=firls(Mls,f,a,w);             % Coefficients of Least square filter
N = 1024;
z = [0:2*N]*pi/N;
[H,W]=freqz(b,1,z);           % Magnitude and phase response

%%Cheby
Mcheby=6;             %order of the filter 3.54%
[b3,a3] = cheby2(Mcheby,20.0206,[0.2495 0.5115],'bandpass'); %Coefficients
[Hc,Wc]=freqz(b3,a3,z);

%%REMEZ
M=45;
a=[0 0 1 1 0 0];                 % Amplitude values at each region
w=[0.8,0.09,0.8];               % Scaling values at each region
N = 1024;
z = [0:2*N]*pi/N;
f1=[0 0.2 0.255 0.5 0.5475 1]; % frequencies
b1=firpm(M,f1,a,w);            % Coefficients of filter
[H1,W1]=freqz(b1,1,z);
```

Published with MATLAB® R2014b

Contents

Least Square Filter

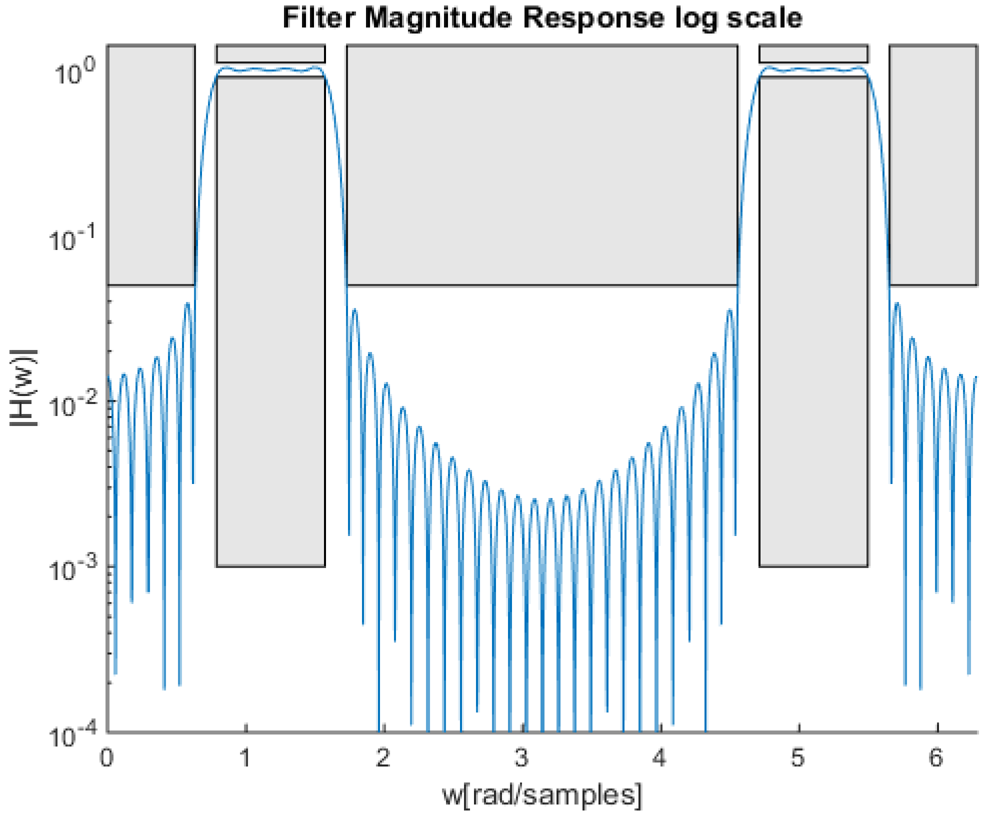


FIGURE 1

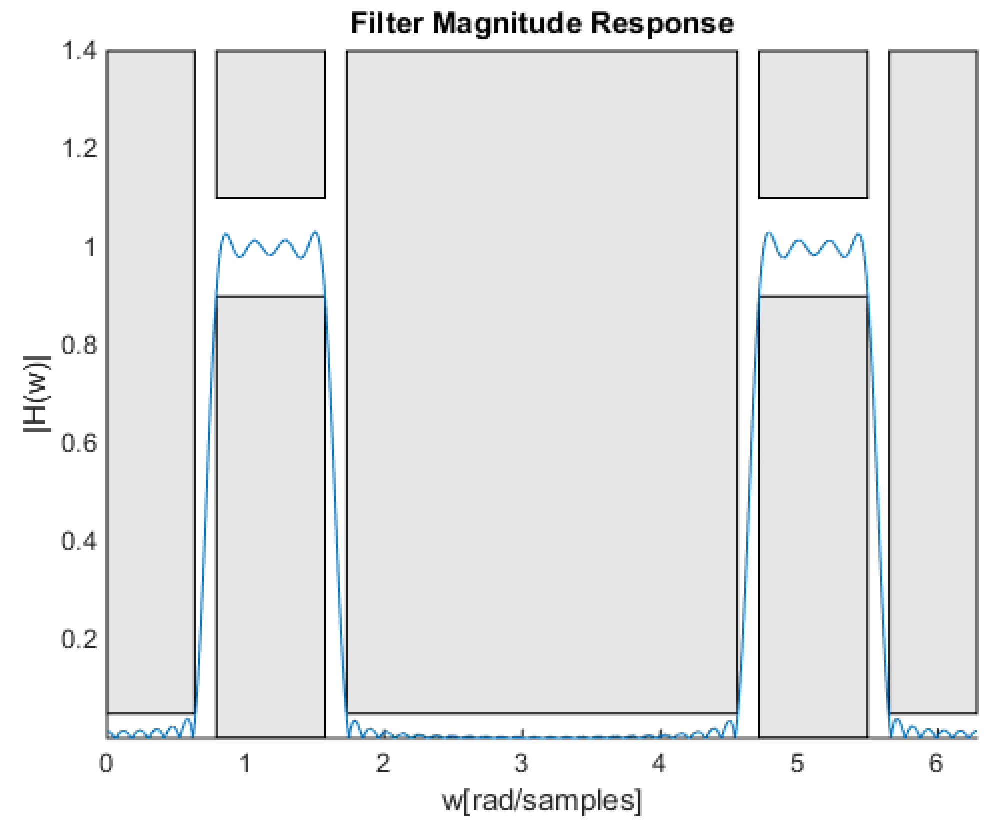


FIGURE 2

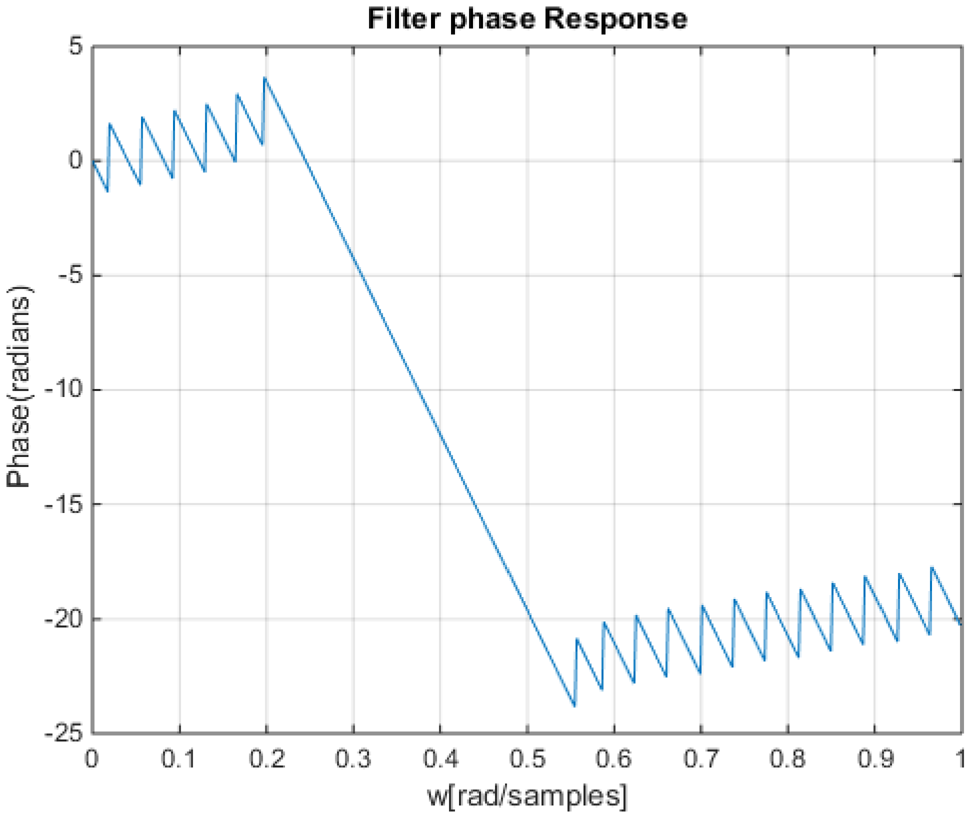


Figure 3

Published with MATLAB® R2014b

Contents

Optimum Equiripple

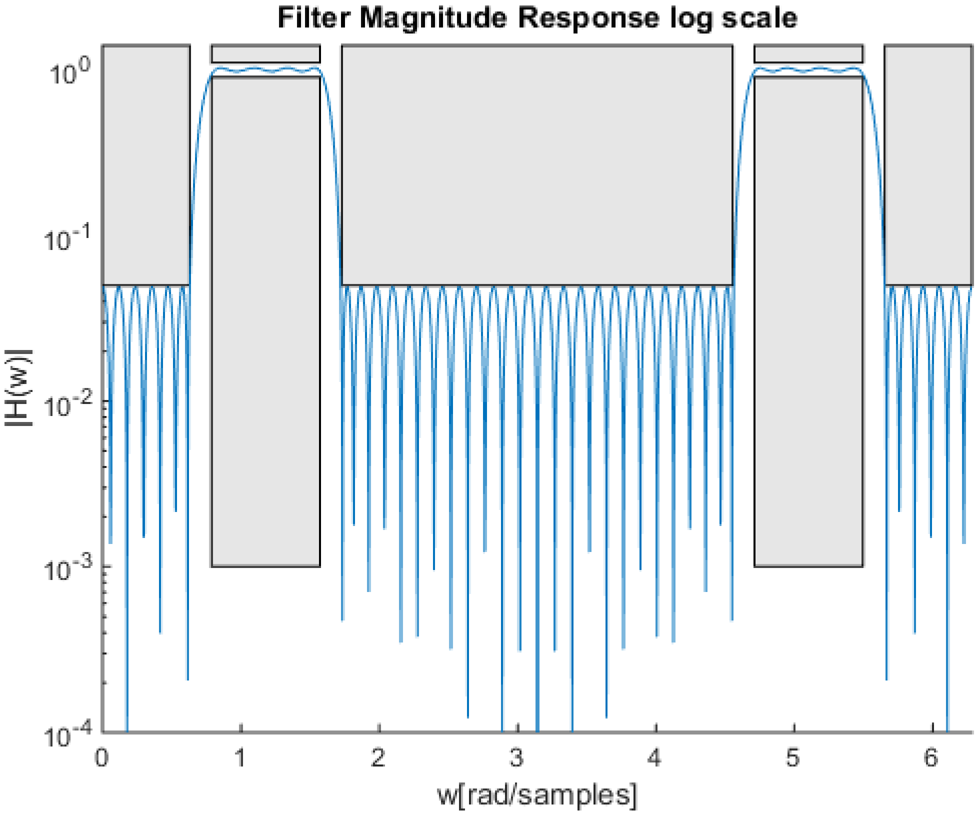


Figure 4

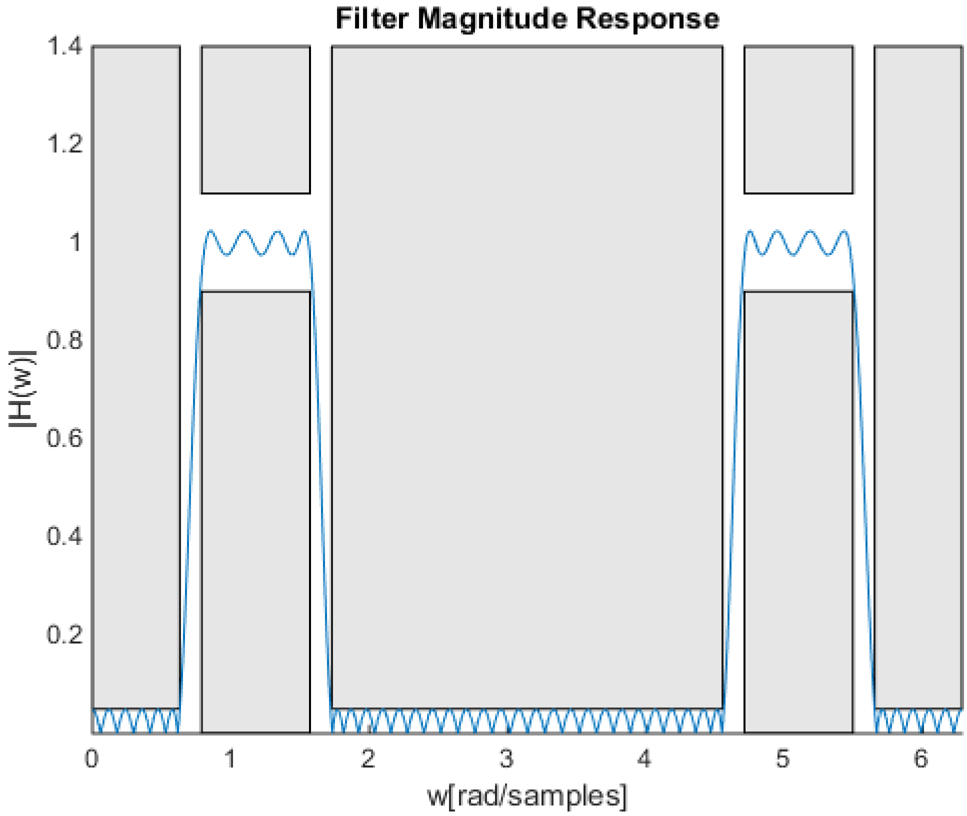


Figure 5

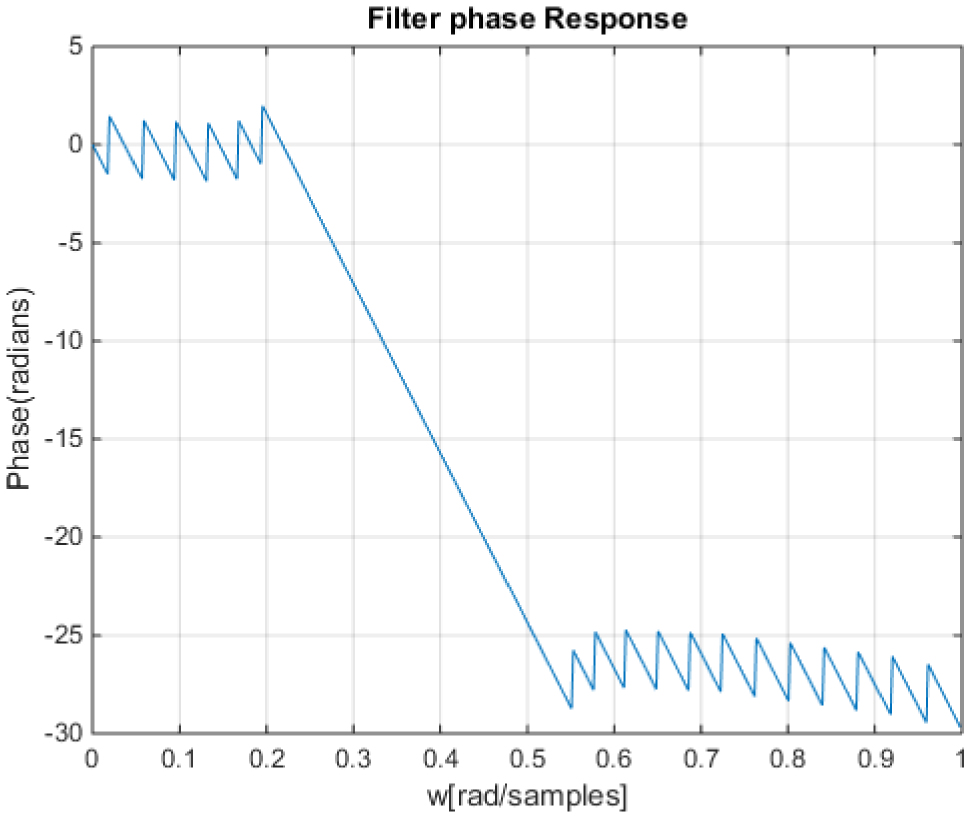


Figure 6

Published with MATLAB® R2014b

Contents

Butterworth

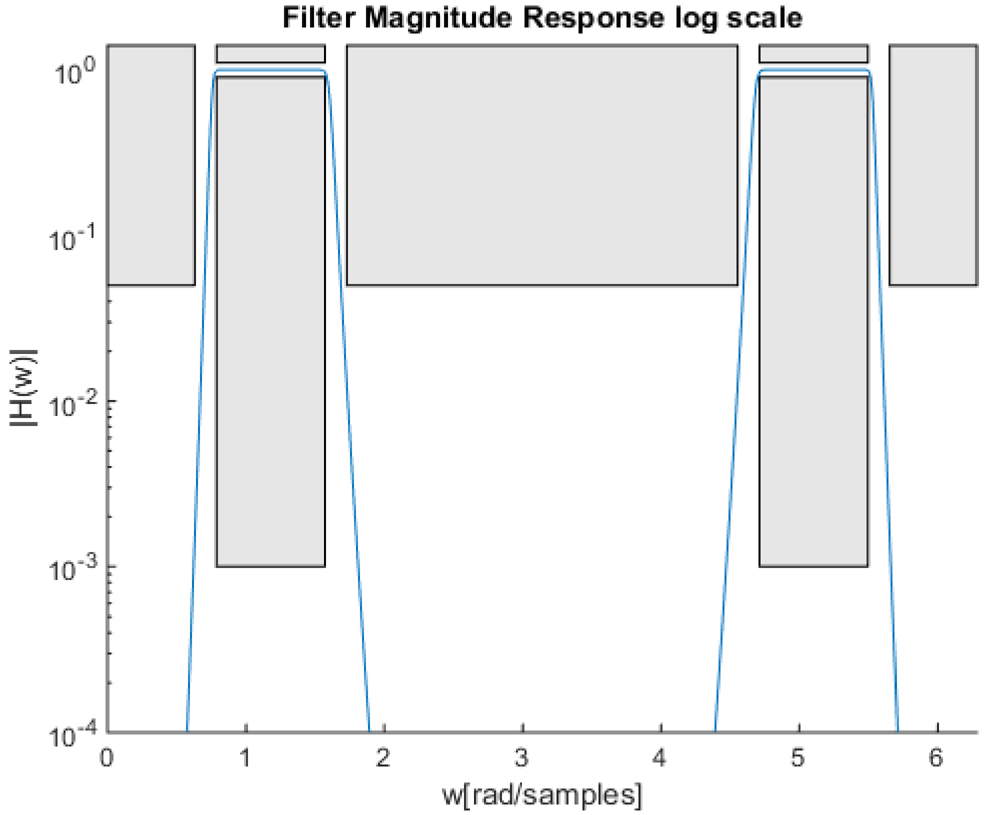


Figure 7

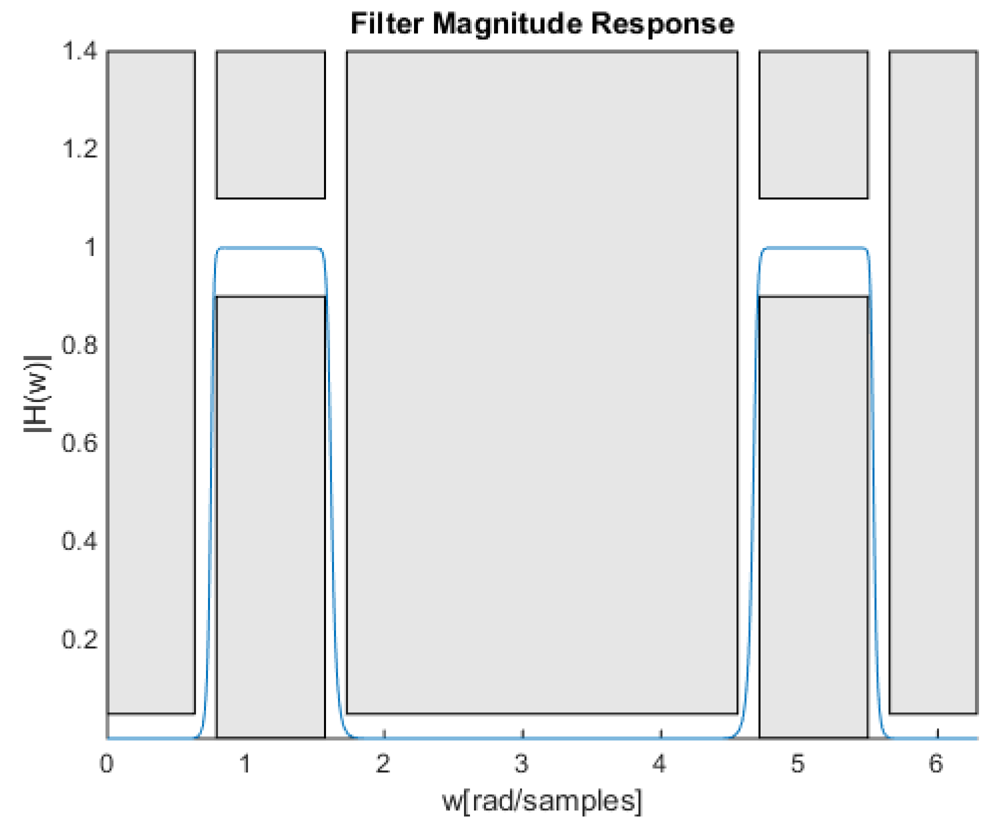
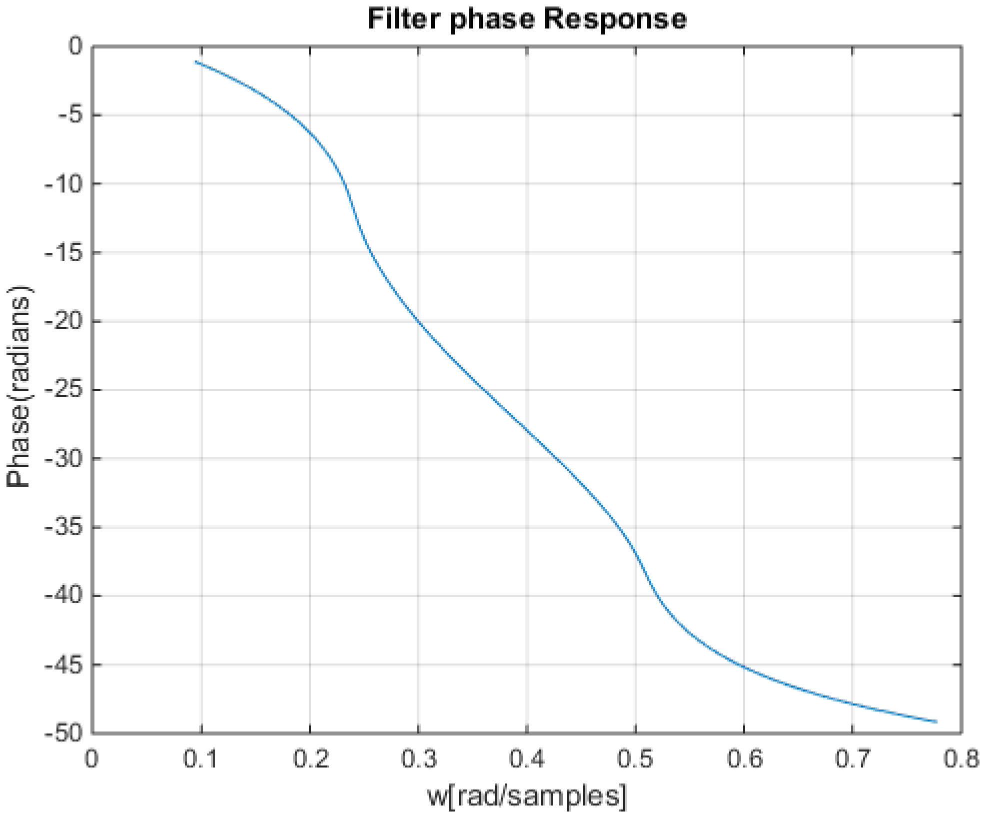


Figure 8



Filter phase Response
Phase(radians)
w[rad/samples]
0
-5
-10
-15
-20
-25
-30
-35
-40
-45
-50
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

Figure 9

Contents

Chebyshev

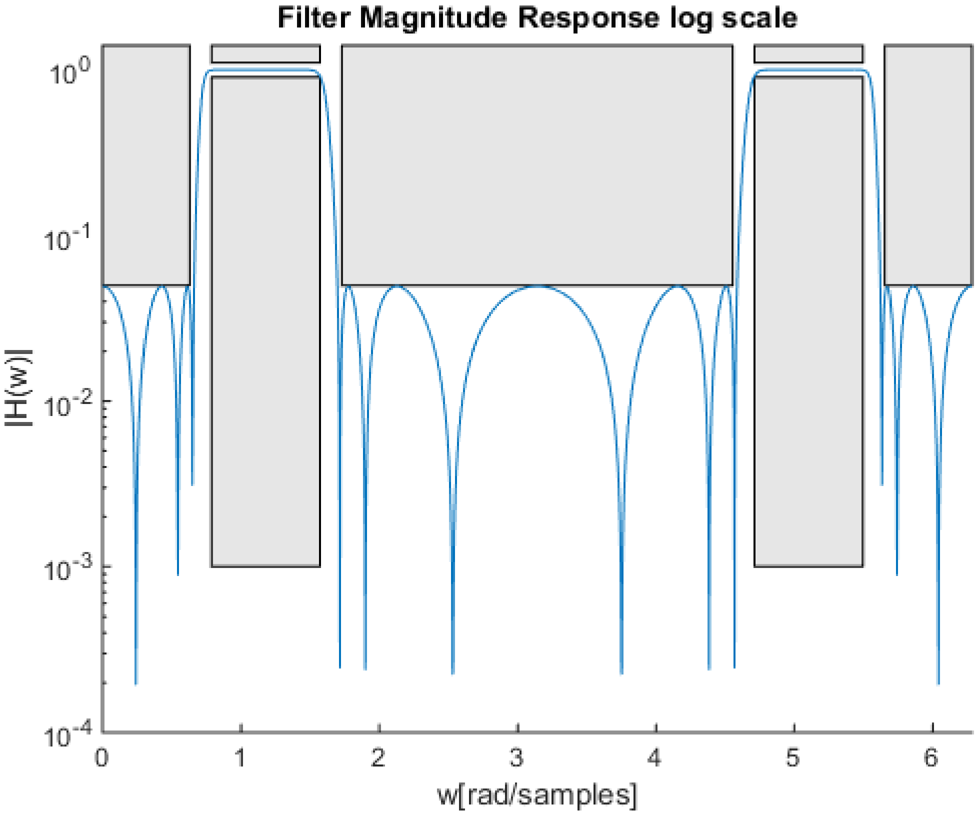


Figure 10

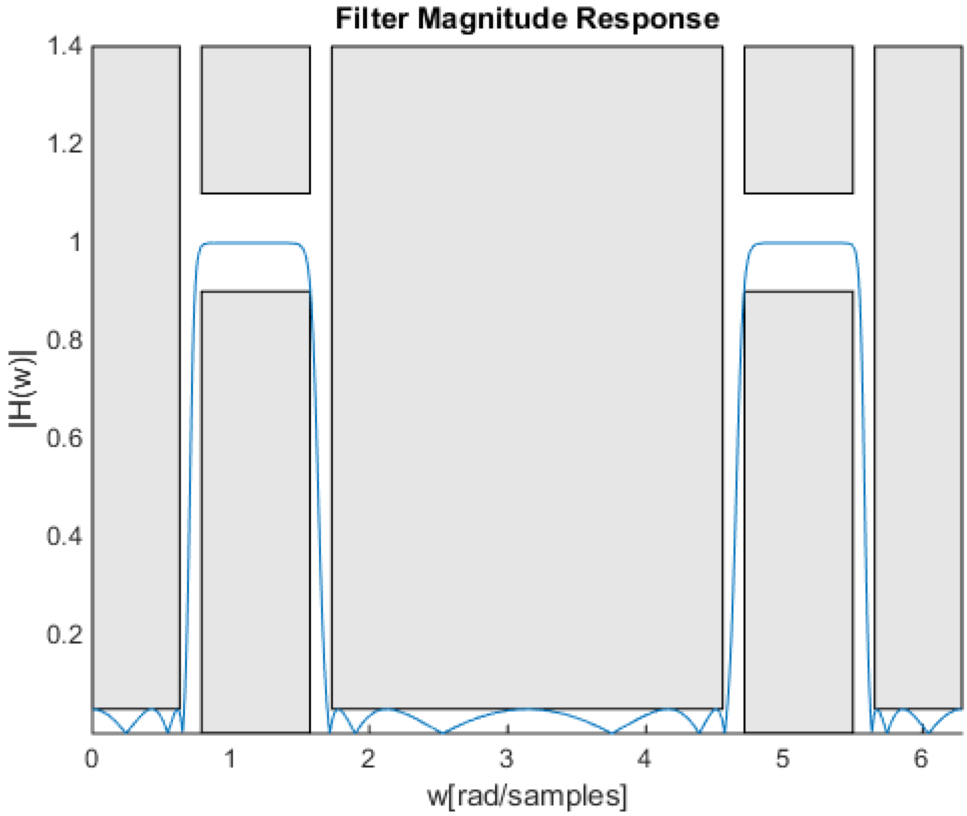


Figure 11

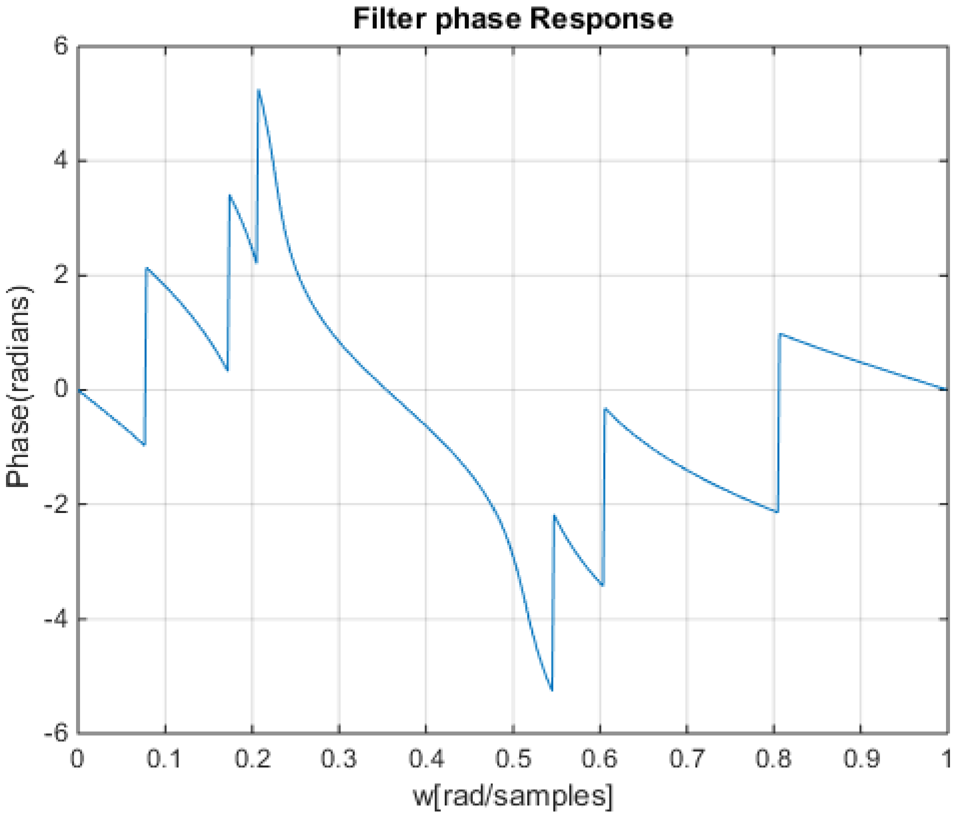


Figure 12

Published with MATLAB® R2014b

Least Square

Figure 13 (clean signal)

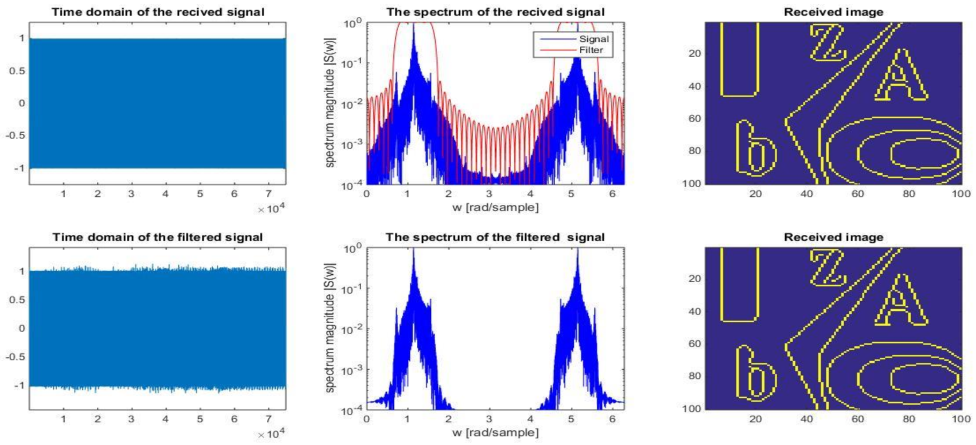


Figure 14 (Noise)

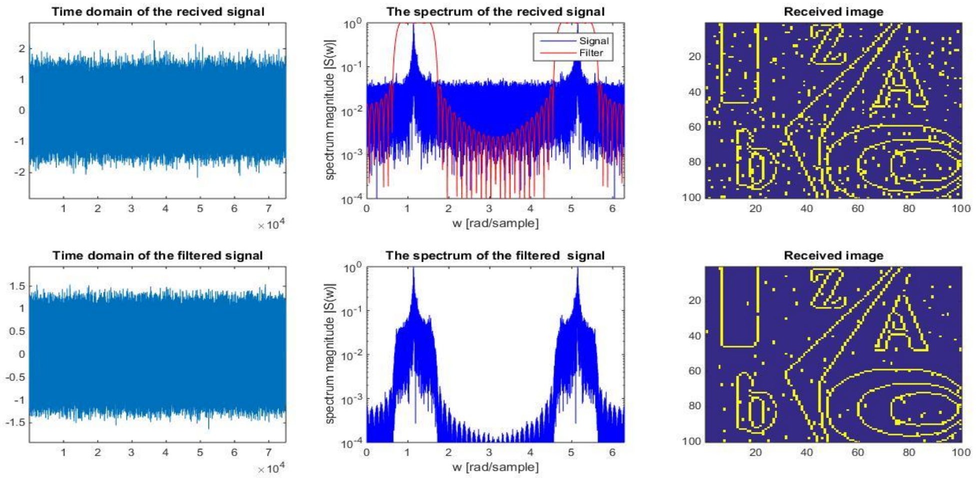


Figure 15 (Interferer)

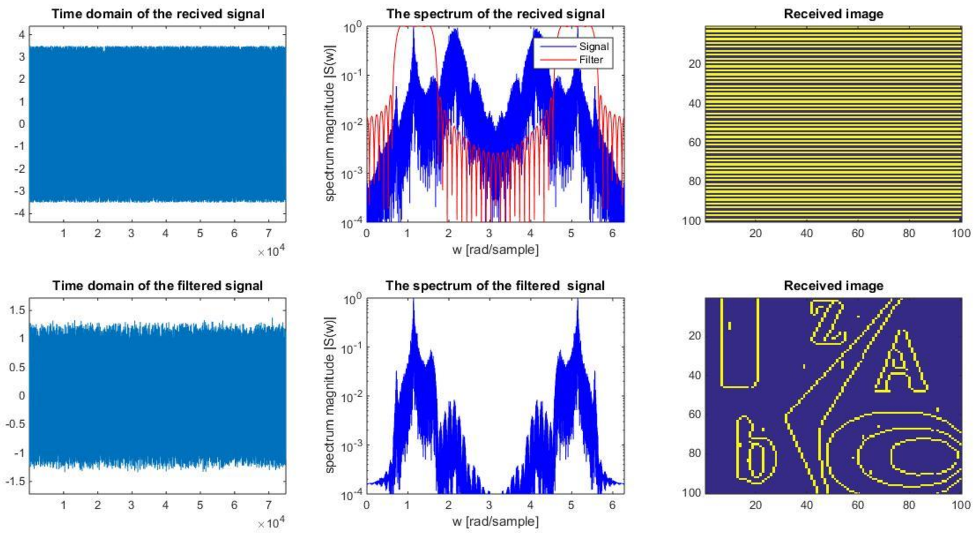
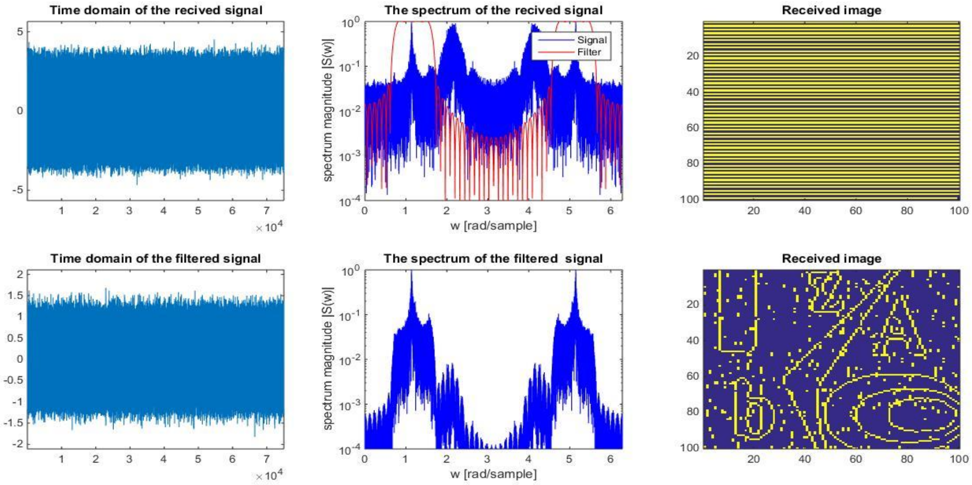


Fig 16 (both)



Equiripple filter

Fig 17 (clean)

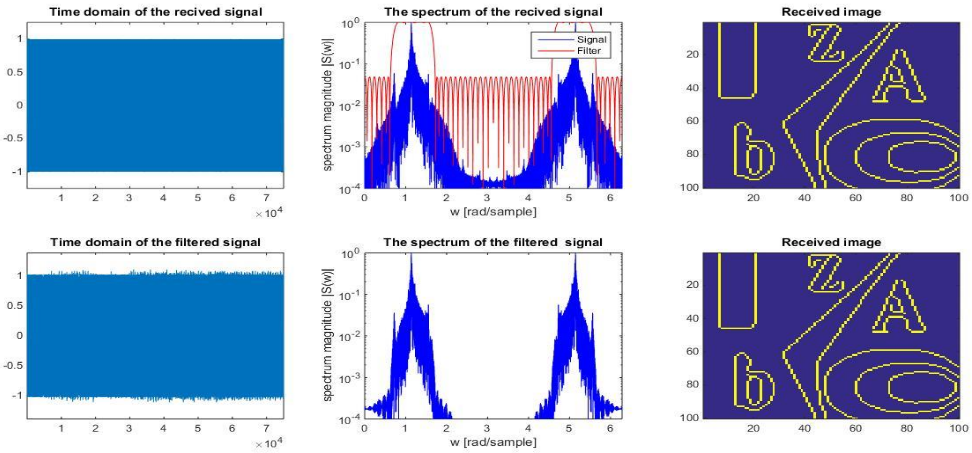


Fig 18 (noise)

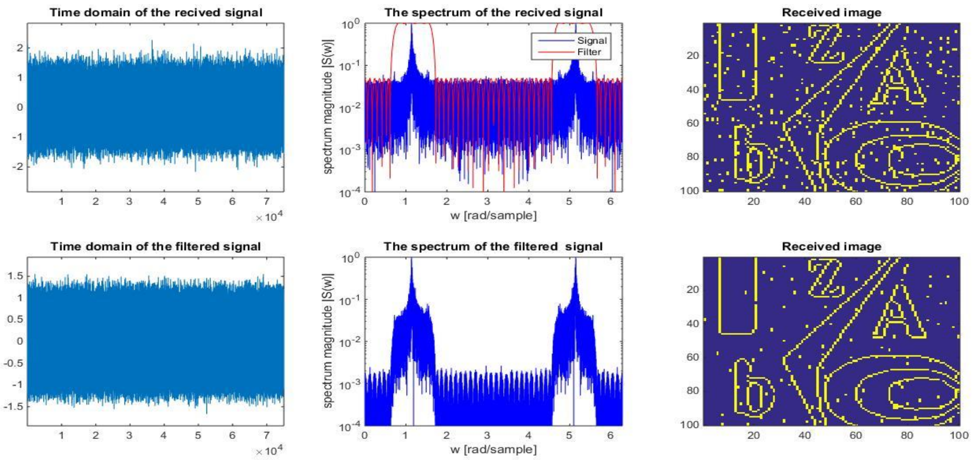


Fig 19(interferer)

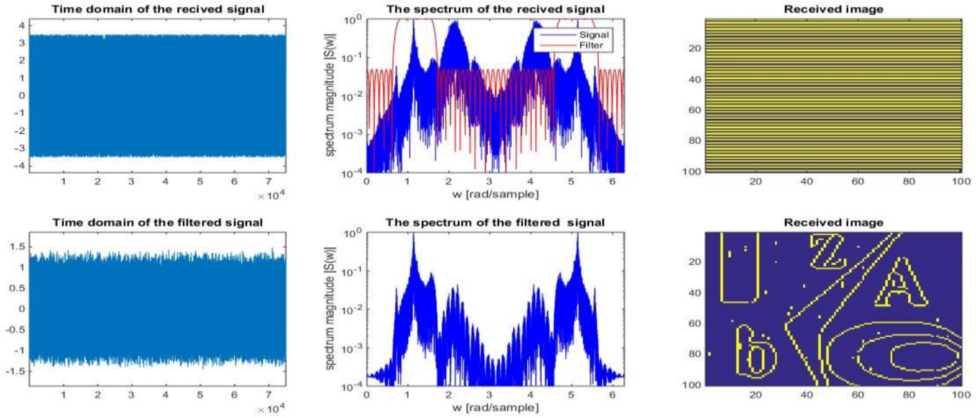
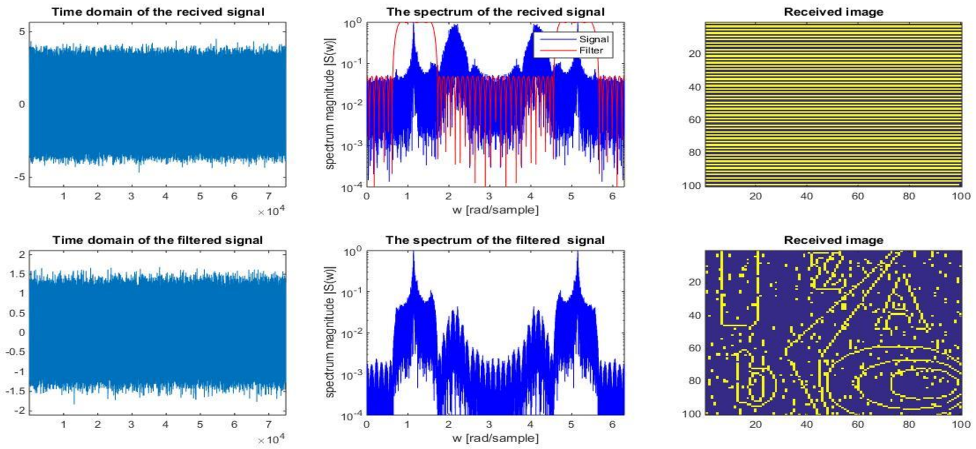


Fig 20 (both)



<u>Butterworth</u>

Fig 21(clean)

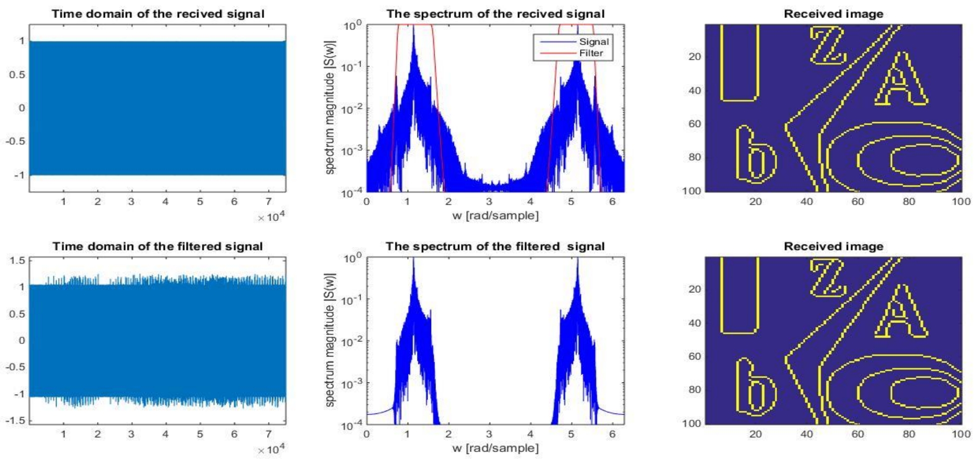


Fig 22 (noise)

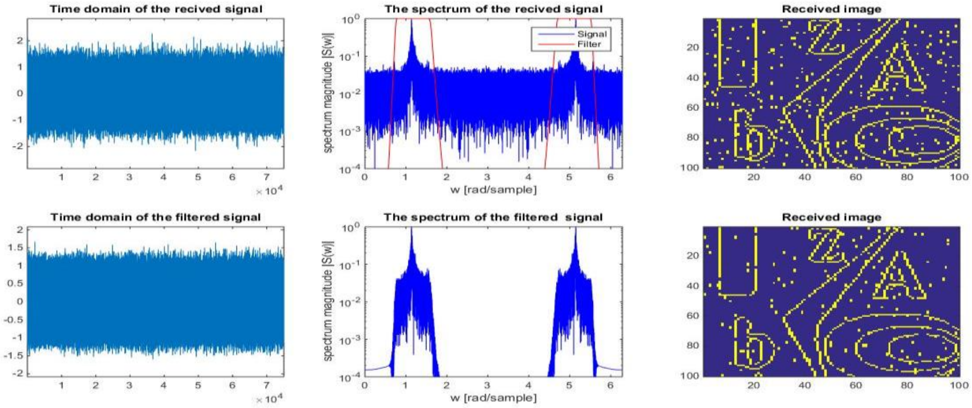


Fig 23 (interferer)

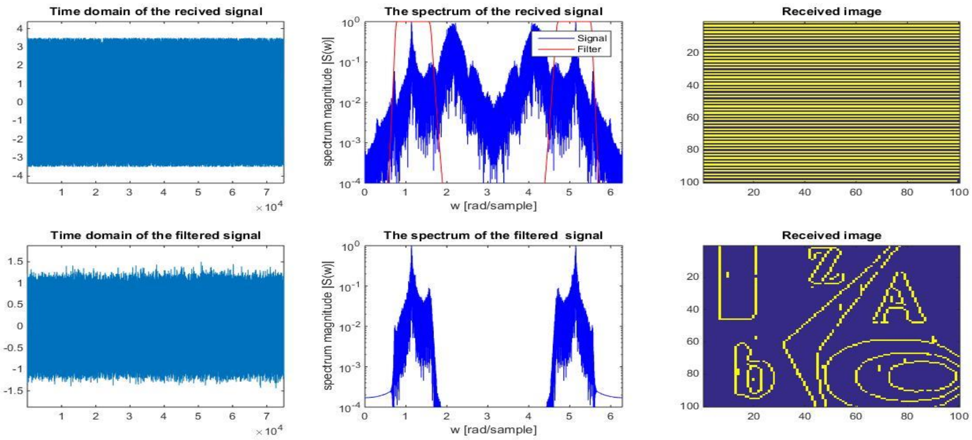
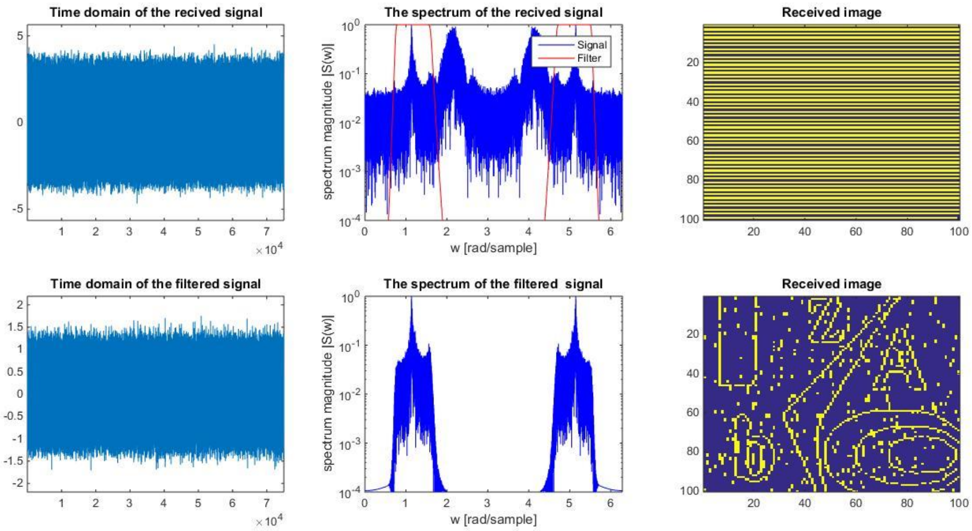


Fig 24 (both)



Chebyshev

Fig 25 (clean)

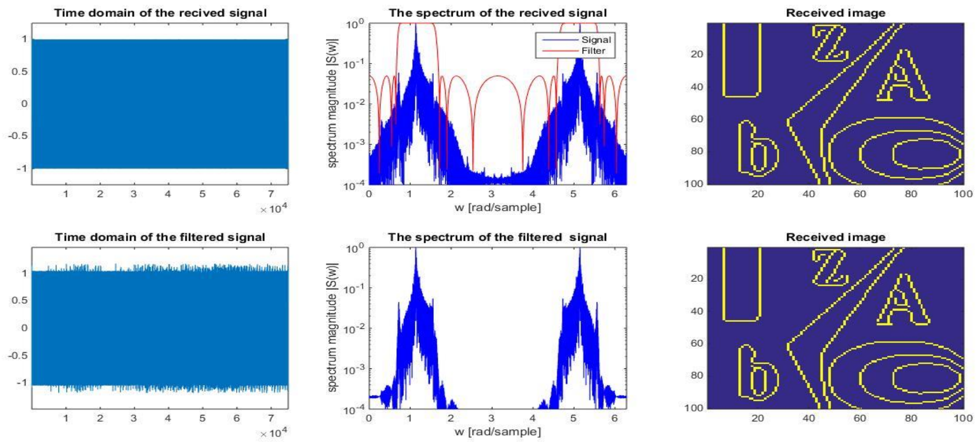


Fig 26 (noise)

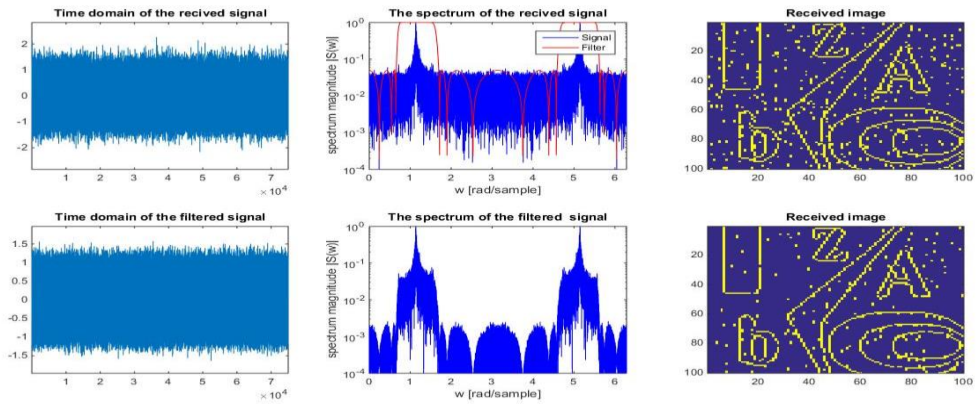


Fig 27 (interferer)

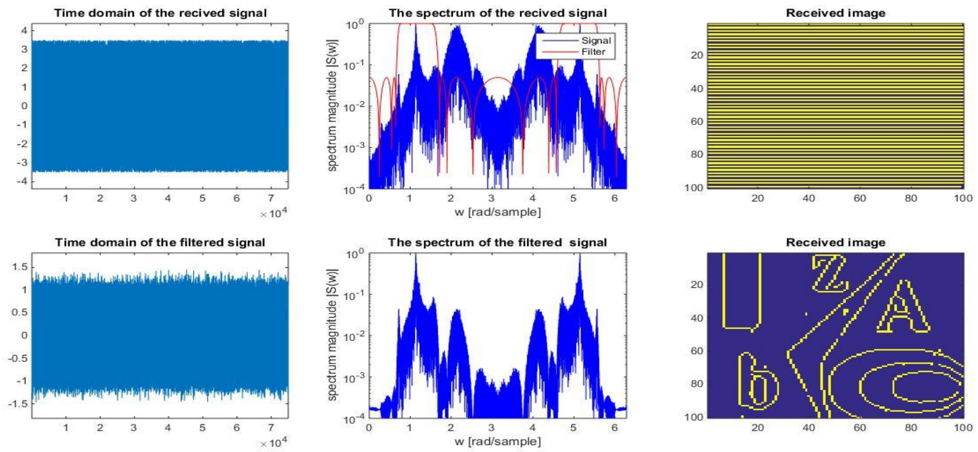


Fig 28 (both)

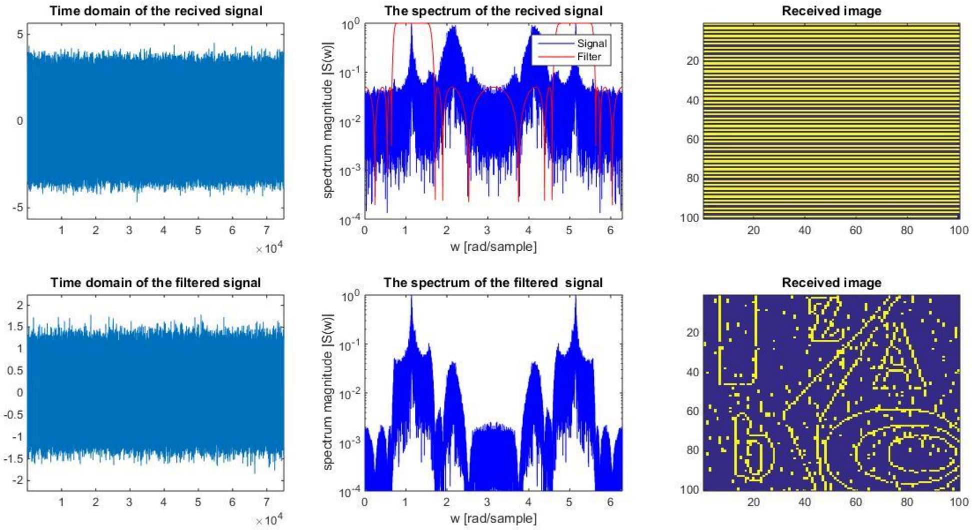

Time domain of the recived signal
The spectrum of the recived signal
Signal
Filter
spectrum magnitude |S(w)|
w [rad/sample]
Received image
Time domain of the filtered signal
The spectrum of the filtered signal
spectrum magnitude |S(w)|
w [rad/sample]
Received image

Table 1

Type of corruption	BER before Filter	BER after Filter			
		Least Square	Equiripple	Butterworth	Chebyshev
Clean	0 %	0 %	0 %	0 %	0 %
Noise	5.2 %	1.74 %	1.69 %	3.16 %	2.96 %
Interferer	50.13 %	0.33 %	0.63 %	0.63 %	0.48 %
Both	50.13 %	5.52 %	5.86 %	5.81 %	5.51 %

Table 2

Type of corruption	Least Square	Equiripple	Butterworth	Chebyshev
Both	3.49	3.55	-	3.54