Course Project

Digital Filtering

Author: Xingzhou Tang

0. Preface

In signal processing, a digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal. We can divide the filter into high-pass filters and low-pass filters by the frequency relative to the cutoff frequency it passes. We can also divide the filters into Infinite impulse response (IIR) and Finite impulse response (FIR) by the length of its impulse response. In this lab, we will design four kinds of filters and test their ability to filter the signals with noise.

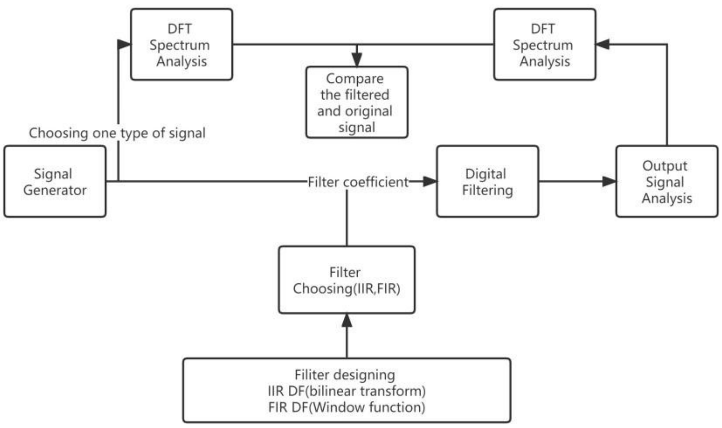

Figure 1*:* The flow chart of this class project

1. Generating Sample Signal

We need to create a signal with noise first to for the consequent filtering. We will create a signal that is the sum of the sine function in order to be easier to forecast how the outcome should be like. We will also use add an extra noise on that and analyze the frequency response(both magnitude and phase response). We will also compare this response with the filtered response in chapter 3.

1.1 Create a signal `10cos(2πt)+10cos(4πt)+10cos(6πt)` and set the t = 1 and the sample interval 0.01. The noise (you can use `randn` function to do that) is added after the twentieth sample to the end. Use hold on function to show the original signal and the signal with noise in the same plot. Include your code and plot.

Answer : The code is as follows::

```
function exe1_1()
dalt=0.01;
t=0:0.01:1;
rn=randn(1,length(t));rn(1:20)=0;
figure(1)
y1=10*cos(2*pi*t)+10*cos(2*pi*t*2)+10*cos(2*pi*t*3);
plot(t,y1)
hold on
y = 10*cos(2*pi*t)+10*cos(2*pi*t*2)+10*cos(2*pi*t*3)+rn;
plot(t,y)
```

The figure is as follows:

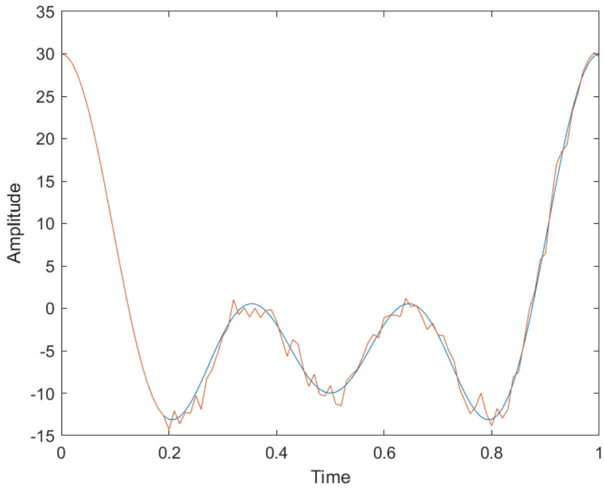

Figure 1*:* The amplitude of the original signal and the signal with noise （since the noise is created by random function, your plot will be the same as figure 1 exactly）

1.2 Make an sample of the signal with noise. Set the sample interval equal to 0.01 and set the sample time to 1s. Include the code and signal.

Answer: The code is as follows:

```
function exe1_2()
figure(1)
dalt=0.01;
t=0:0.01:1;
rn=randn(1,length(t));rn(1:20)=0;
y=10*cos(2*pi*t)+10*cos(2*pi*t*2)+10*cos(2*pi*t*3)+rn;
subplot(2,1,1);plot(t,y);
title('Test signal');
```

```
xlabel('time');ylabel('amplitude');
subplot(2,1,2);stem(t,y,'.');
title('Signal sample');
xlabel('time');ylabel('amplitude');
end
```

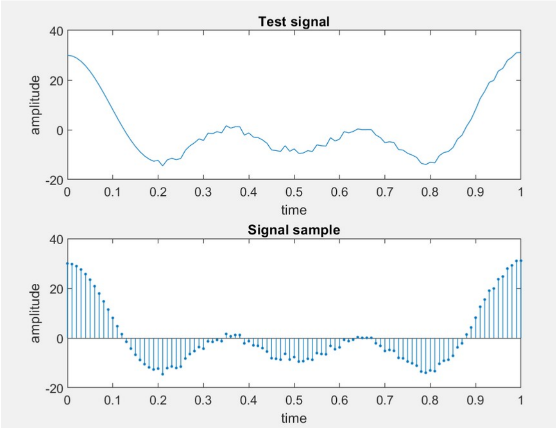

Figure 2*:* The original signal and signal sample

1.3 Compute the frequency response of the signal with noise (both magnitude response and phase response) in 32 samples and 64 samples. In order to facilitate reading the frequency value, it is better to make normalization and take w/pi as the abscissa.(Hint: `k=0:31;wk=2*k/32;`). Use the `subplot` to create four figures in the same figure. Include your code and figure. The plot should be like this.

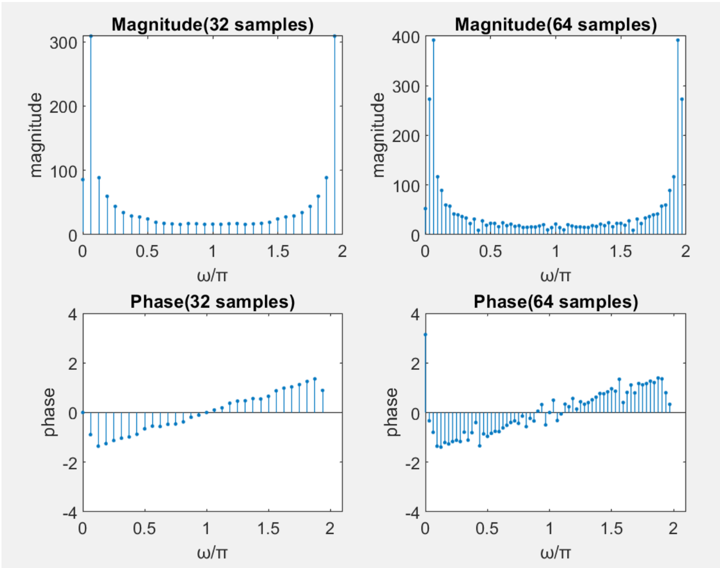

Figure 3*:* The amplitude of the original signal and the signal with noise

Answer: The code is as follows:

```
function exe1_3()
dalt=0.01;
```

```
t=0:0.01:1;
rn=randn(1,length(t));rn(1:20)=0;
y = 10*cos(2*pi*t)+10*cos(2*pi*t*2)+10*cos(2*pi*t*3)+rn;
Yk32=fft(y,32);
k=0:31;wk=2*k/32;
subplot(2,2,1);stem(wk,abs(Yk32),'.');
title('Magnitude(32 samples)');xlabel('ω/π');ylabel('magnitude');
subplot(2,2,3);stem(wk,angle(Yk32),'.');
title('Phase(32
samples)');xlabel('ω/π');ylabel('phase');axis([0,2.1,-4,4])
Yk64=fft(y,64);
k=0:63;wk=2*k/64;
subplot(2,2,2);stem(wk,abs(Yk64),'.');
title('Magnitude(64 samples)');xlabel('ω/π');ylabel('magnitude');
subplot(2,2,4);stem(wk,angle(Yk64),'.');
title('Phase(64
samples)');xlabel('ω/π');ylabel('phase');axis([0,2.1,-4,4])
```

1.4 Make an observation on the plot in the 1.2. Is there any difference between the 32 samples and 64 samples?

Answer: In the plot of magnitude, there is two pits near 0.5 and 1.5 when the sample number is 64. And in the plot of phase, there is some peak near the 1 when the sample number is 64.

2. Filter Design

In this chapter, we will design different kinds of filters to process the signal. We will use the bilinear transformation method to design the IIR filter and use the window function method to design FIR filter.

Bilinear transformation for IIR Filter

The bilinear transform is a first-order approximation of the natural logarithm function that is an exact mapping of the z-plane to the s-plane. When the Laplace transform is performed on a discrete-time signal (with each element of the discrete-time sequence attached to a correspondingly delayed unit impulse), the result is precisely the Z transform of the discrete-time sequence with the substitution of

$$\begin{aligned} z &= e^{sT} \\ &= \frac{e^{sT/2}}{e^{-sT/2}} \\ &\approx \frac{1 + sT/2}{1 - sT/2} \end{aligned}$$

The inverse of this mapping (and its first-order bilinear approximation) is:

$$\begin{aligned} s &= \frac{1}{T}\ln(z) \\ &= \frac{2}{T}\left[\frac{z-1}{z+1} + \frac{1}{3}\left(\frac{z-1}{z+1}\right)^3 + \frac{1}{5}\left(\frac{z-1}{z+1}\right)^5 + \frac{1}{7}\left(\frac{z-1}{z+1}\right)^7 + \cdots\right] \\ &\approx \frac{2}{T}\frac{z-1}{z+1} \\ &= \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \end{aligned}$$

There is some index that we need to take into account when designing the filter. The Wp and Ws are respectively the passband and stopband edge frequencies of the filter. And Butterworth filter with no more than Rp dB of passband and at least Rs dB of sttenuation in the stopband. The index will be provided, you filter should be designed based on the provide code.

2.1 Create a buttord low pass filter based on the bilinear transformation. Since it is difficult to create the whole filter. Some code about the index and buttord filter is provided. Based on the code below plot the frequency response(magnitude response in dB and phase response) and the group delay of the filter. Include your figure. Your figure should like this. Hint: you can use the bilinear function in MATLAB to help you.

```
%some index for the filter to the signal
wplp=0.1*pi;
wslp=0.5*pi;
wphp=0.8*pi;
Rp=1;
As=4;

wp=2*tan(wplp/2); %T=1;
ws=2*tan(wslp/2);
ep=sqrt(10^(Rp/10)-1); %Passband fluctuation parameter ε
Ripple=sqrt(1/(1+ep*ep)); %Minimum passband amplitude
Attn=1/(10^(As/20)); %Maximum ripple of stopband

[N,wn]=buttord(wp,ws,Rp,As,'s') %Calculation of order and cutoff
frequency of simulated Butterworth filter. [n,Wn]
= buttord(Wp,Ws,Rp,Rs) returns the lowest order, n, of the digital
Butterworth filter with no more than Rp dB of passband ripple and at
least Rs dB of attenuation in the stopband.

[z,p,k]=buttap(N); %The zeros and poles of the simulated prototype
low-pass filter system function are designed
[bp,ap]=zp2tf(z,p,k)
[bs,as]=lp2lp(bp,ap,wplp) %The analog low-pass prototype filter is
converted to the actual low-pass filter
```

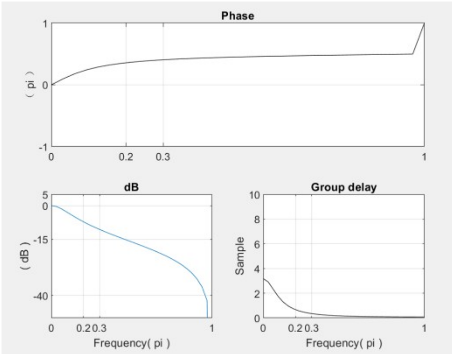

Figure 4*:* The frequency response and group delay of the filter.

Answer for 2.1:

The code is as follows:

```
[bz,az]=bilinear(bs,as,1) %use the bilinear transformation to change
into a digital IIR filter
[H,w]=freqz(bz,az,64,'whole');
H= (H(1:1:33))';w= (w(1:1:33))';
maglp=abs(H); %to calculate the magnituded
dblp=20*log10((maglp+eps)/max(maglp)); %change into dB
phalp=angle(H); %calcuate the phase
grdlp=grpdelay(bz,az,w); %calculate the groupdelay

figure(3);subplot(1,1,1)

subplot(2,2,3);plot(w/pi,dblp);title('dB');
xlabel('Frequency( pi )');ylabel('( dB )');axis([0,1,-50,5]);
set(gca,'XTickMode','manual','Xtick',[0,0.2,0.3,1]);
set(gca,'YTickMode','manual','Ytick',[-40,-15,0,5]);grid
subplot(2,2,[1 2]);plot(w/pi,phalp/pi,'k');title('Phase');
ylabel(' ( pi ) ');axis([0,1,-1,1]);
set(gca,'XTickMode','manual','Xtick',[0,0.2,0.3,1]);
set(gca,'YTickMode','manual','Ytick',[-1,0,1]);grid
subplot(2,2,4);plot(w/pi,grdlp,'k');title('Group delay');
xlabel('Frequency( pi )');ylabel('Sample');axis([0,1,0,10])
set(gca,'XTickMode','manual','Xtick',[0,0.2,0.3,1]);
set(gca,'YTickMode','manual','Ytick',[0:2:10]);grid
```

Window function for FIR Filter

The window method is easily understood in terms of the convolution theorem for Fourier transforms, making it instructive to study after the Fourier theorems and windows for spectrum analysis.

The window method consists of simply ``windowing'' a theoretically ideal filter impulse response $h(n)$ by some suitably chosen window function $w(n)$, yielding:

$$h_w(n) = w(n) \cdot h(n), \quad n \in \mathcal{Z}.$$

The index of the filter will still be provided. In this lab, we will use the kaiser window to design the filter. There are some advantage of kaiser window when comparing to other windows. In Kaiser window, the passband and stopband ripple size are not affected much by the change in the window length(when compared to other windows). It is affected by how the coefficients roll off(the shape factor).The window length mainly affects the transition band.

So keeping the window length constant, we can adjust the shape factor, to design for the passband and stopband ripples. This is a huge advantage over other windows, where the window length, ripple size and transition bandwidth have a three-way tradeoff.

The coefficients of a Kaiser window are computed from the following equation:

$$w(n) = \frac{I_0\left(\beta\sqrt{1-\left(\frac{n-N/2}{N/2}\right)^2}\right)}{I_0(\beta)}, \quad 0 \le n \le N,$$

where I0 is the zeroth-order modified Bessel function of the first kind. The length L = N + 1.

2.2 Use the `kaiser` and `fir1` function to design a filter with kaiser window. Plot the frequency response(magnitude in dB and phase) of the filter. Include the plot and code. The parameters of the filter is provided.

The parameter settings are as follows:

```
fp=4;fc=14;As=3;Ap=1;Fs=100;% settings for filter
wc=2*pi*fc/Fs; wp=2*pi*fp/Fs;
wdel=wc-wp;
beta=0.112*(As-1);% the shape factor used in beta
N=ceil((As-0.5)/2.285/wdel);
```

Answer:

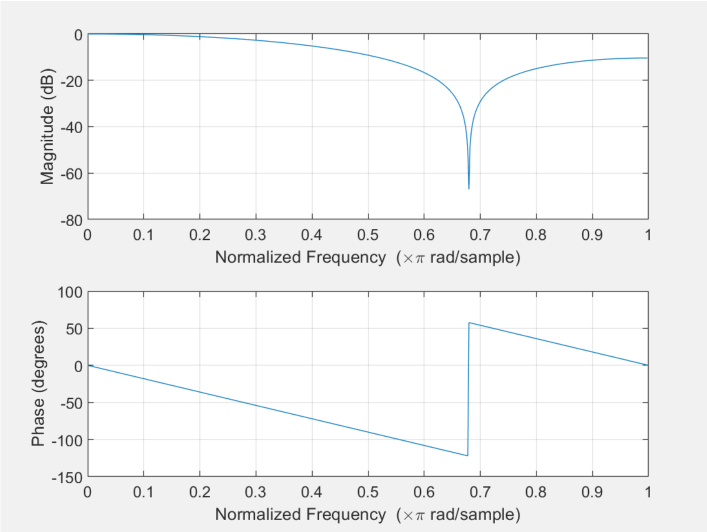

Figure 5*:* The frequency response of the lowpass FIR based on kaiser

The code is as follows:

```
wn= kaiser(N+1,beta); % returns an L-point Kaiser window with shape
factor beta.
ws=(wp+wc)/2/pi;
b=fir1(N,ws,wn);%design a kaiser Window-based FIR filter
freqz(b,1);
```

2.3 Still use kaiser and follow the way in 2.2 to design a highpass filter. Include your plot, but not your code.

The parameter is provided as follow:

```
fp=12;fc=15;As=100;Ap=1;Fs=100;
```

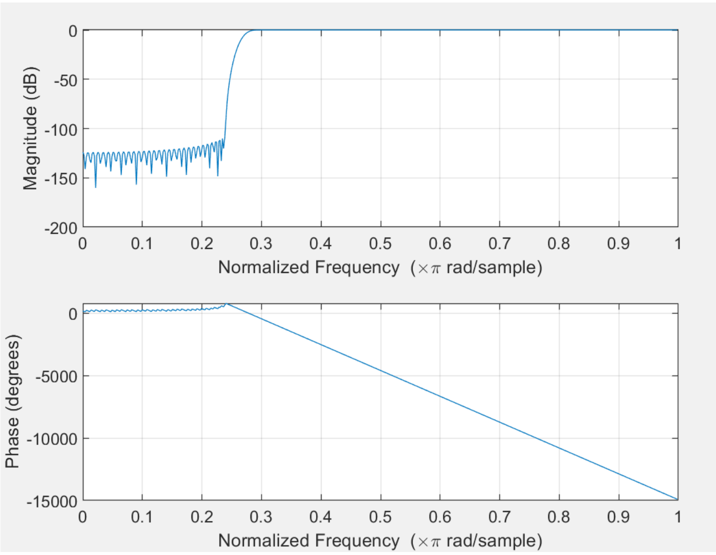

Figure 6*:* The frequency response of the highpass FIR based on kaiser

3. Analyze the output of different filters

In chapter 2, we have created a filter by the bilinear transformation and 2 filters by the window functions. Now we can use the signal with noise as the input and analyze the output of different kinds of filters.

3.1 Use the signal with noise as input to the IIR low pass filter. Plot the frequency response(magnitude and phase response). (use `axis` function to set the x range from 0 to1.6)

Answer: The code is as follows:

```
t=0:0.01:1;
rn=randn(1,length(t));rn(1:20)=0;
y=10*cos(2*pi*t)+10*cos(2*pi*t*2)+10*cos(3*pi*t*3)+rn;
ynlp=filter(bz,az,y)
Fs=100;
N=40;n=0:N-1;
Yk64lp=fft(ynlp)/Fs;
figure(7)
k=0:length(Yk64lp)-1;f=k*N/pi/Fs;
subplot(2,1,1);stem(f/pi,abs(Yk64lp),'.');
axis([0,1.6,0,15]);xlabel('Frequency( pi )');ylabel('Magnitude');
subplot(2,1,2);stem(f/pi,angle(Yk64lp),'k.');
axis([0,1.6,-4,4]);xlabel('Frequency( pi )');ylabel('Phase');
```

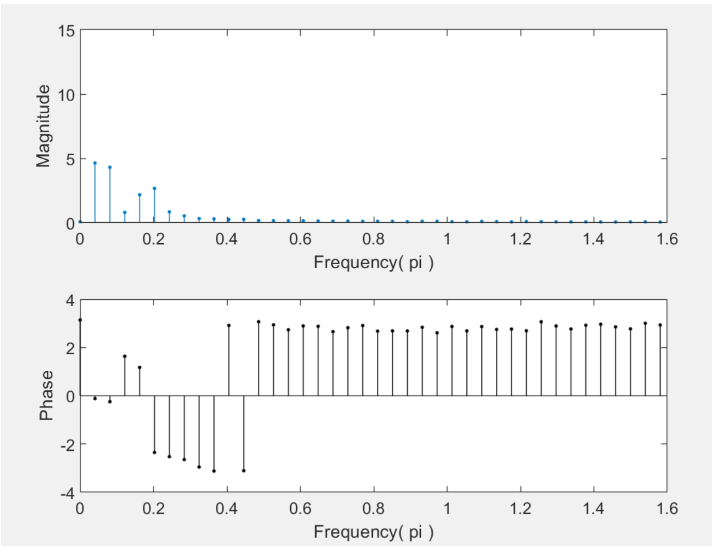

Figure 7: The frequency response of the lowpass IIR by bilinear transformation

3.2 Use the signal with noise as input to the FIR low pass filter. Plot the frequency response(magnitude) and the waveform of both original and filtered signal. Use subplot function to plot four in one figure. Include your code and figure.

Answer: The code is as follows:

```
figure(9)
t=0:0.01:1;
rn=randn(1,length(t));rn(1:20)=0;
y=10*cos(2*pi*t)+10*cos(2*pi*t*2)+10*cos(3*pi*t*3)+rn;
Y1=fft(y);
fp=4;fc=16;As=4;Ap=1;Fs=100;
```

```
wc=2*pi*fc/Fs; wp=2*pi*fp/Fs;
wdel=wc-wp;
beta=0.112*(As-1);
N=ceil((As-0.5)/2.285/wdel);
wn= kaiser(N+1,beta);
ws=(wp+wc)/2/pi;
b=fir1(N,ws,wn);
freqz(b,1);
x=fftfilt(b,y);
X=fft(x);
subplot(2,2,1);plot(abs(Y1));
title('Magnitude before filter');
subplot(2,2,2);plot(abs(X));
title('Magnitude after filter');
subplot(2,2,3);plot(y);
title('Waveform before filter');
subplot(2,2,4);plot(x);
title('Waveform after filter');
```

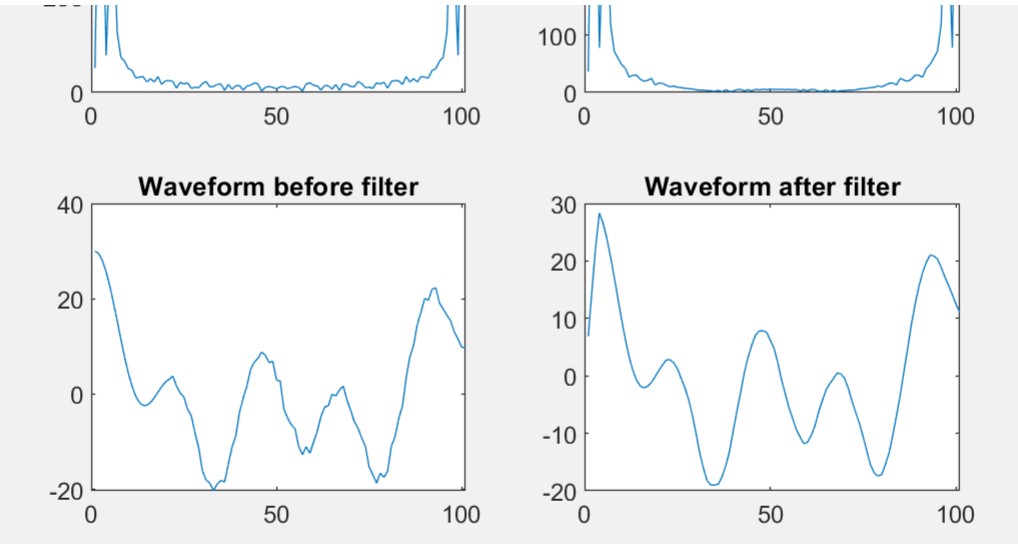

Figure 8*:* The frequency response and waveform before and after the Filter

3.3 Use the signal with noise as input to the FIR high pass filter. Plot the frequency response and waveform(like what we did in 3.2). This time just your figure.

Answer: The code is as follows:

```
figure(10);
t=0:0.01:1;
rn=randn(1,length(t));rn(1:20)=0;
y=10*cos(2*pi*t)+10*cos(2*pi*t*2)+10*cos(3*pi*t*3)+rn;
Y1=fft(y);
fp=12;fc=15;As=100;Ap=1;Fs=100;
wc=2*pi*fc/Fs; wp=2*pi*fp/Fs;
wdel=wc-wp;
```

```
beta=0.112*(As-1);
N=ceil((As-0.8)/2.285/wdel);
wn= kaiser(N,beta);
ws=(wp+wc)/2/pi;
b=fir1(N-1,ws,'high',wn);
freqz(b,1);
x=fftfilt(b,y);
X=fft(x);
subplot(2,2,1);plot(abs(Y1));
title('Magnitude before filter');
subplot(2,2,2);plot(abs(X));
title('Magnitude after filter');
subplot(2,2,3);plot(y);
title('Waveform before filter');
subplot(2,2,4);plot(x);
title('Waveform after filter');
```

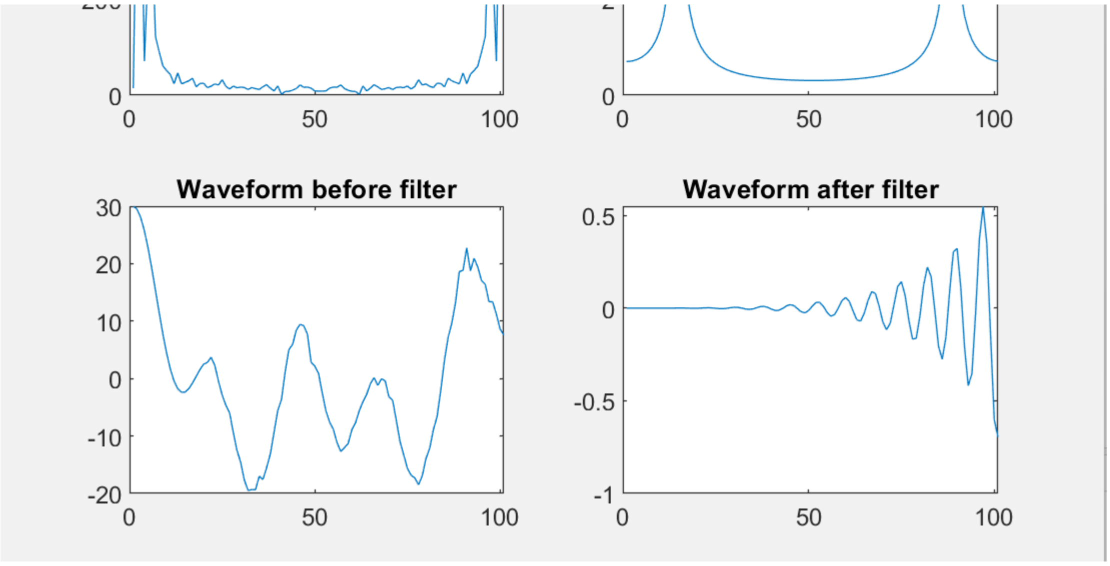

Figure 9*:* The frequency response and waveform before and after the Filter

3.4 Observe the waveform in the plot of 3.3 and 3.2.Why the waveform after filter is so different from the waveform before filter in 3.3?

Because the filter in 3.3 is a high pass filter. But the original signal is with low frequency, so only most of the signal is filtered by the FIR filter. While the filter in 3.2 is a low pass filter, it can preserve most of the original signal and preserve the shape well.

3.5 Compare the outcome of different kinds of filters. And analyze the advantage and disadvantage of each of them. (ie. Linear phase, the frequency overlap, complexity of designing)

Analysis of Bilinear Transformation method:

The main advantage of the bilinear transformation: the bilinear transformation will not be confused to the low frequency part because the high frequency part exceeds the folding frequency.
The disadvantage of the bilinear transformation method: it will produce the frequency mixing phenomenon, so that the frequency noise offset of the digital filter simulates the frequency noise of the filter.
Analysis on the window functions:
Window function method: the phase response is strictly linear, there is no stability problem, simple design.

4. Summary

In this class project, we first create a sample signal. Given by the setting for the filter we design a low pass IIR filter based on the bilinear transformation and FIR filter based on the windows function. After we analyze the output signal, we then conclude the advantage and disadvantage of different methods. We can also have a general idea on how to design a filter in matlab with filter parameters(like w_p ,As) provided.

5. Acknowledgement

I want to thank Dr. Gibson for giving me some instruction on the diagram in the proposal.