Xiaobin Zhuang - 300519184

ECEN220 Lab 5

09/10/2020

Objectives

The object of this lab was to upsample the signal.

Methodology

1. First, we sampled the original signal at 2kHz, then take 8k samples of pulse. Code and graph Shown in figure 1.

```
% Q1
clear all
f0 = 100;
T = 0.5;
fs1 = 2000; % sampling rate fs = 2kHz
fs2 = 8000; % upsampling rate fs = 8kHz
Tmax = 1;
t1 = 0:1/fs1:Tmax;
t2 = 0:1/fs2:Tmax;
Q = 2;
x = zeros(1,length(t1));
for q = 0:Q
    S = sin(2.*pi.*(2.*q+1).*f0.*t1)/(2.*q+1)
    x = x + (4/pi).*S  % sampling
end

x2 = zeros(1,length(t2));
M = 4;
x2(1:M:end-1) =  x(1:(end-1))   % upsampling

plot(t1,x)
hold on
stem(t2,x2)
xlim([0.03,0.04])
hold off
legend('sampled at 2kHz','upsampled at 8kHz')
```

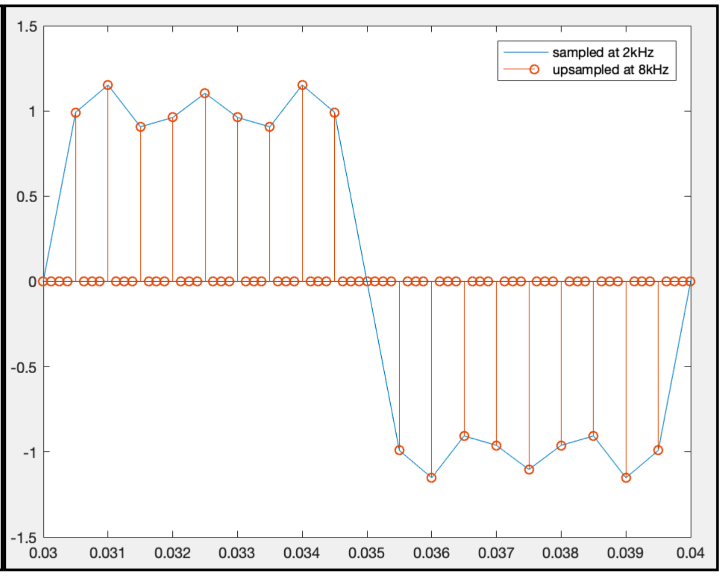

fig. 1

2. We then created a low pass filter (sinc function). Code shown in figure 2, we limited x-axis from -50 to 50, to make plot (since function) more observable.

```
% Q2
L = 100;
n = -L:L;
W = 0:1/length(n):pi;
h = zeros(1,length(n));
h = M.*sin((pi.*n)/M)./(pi.*n);   % filter response
figure()
h (n == 0) = 1;
freqz(h,1,W)
figure()
plot(n,h)
xlim([-50 50]) % limit x-axis from -50 to 50, to make plot more observable
xlabel('n')
ylabel('h')
title('Sinc')

xL = xlim;
yL = ylim;
line([0 0], yL, 'color', 'black');  %x-axis
line(xL, [0 0], 'color', 'black');  %y-axis
```

fig. 2

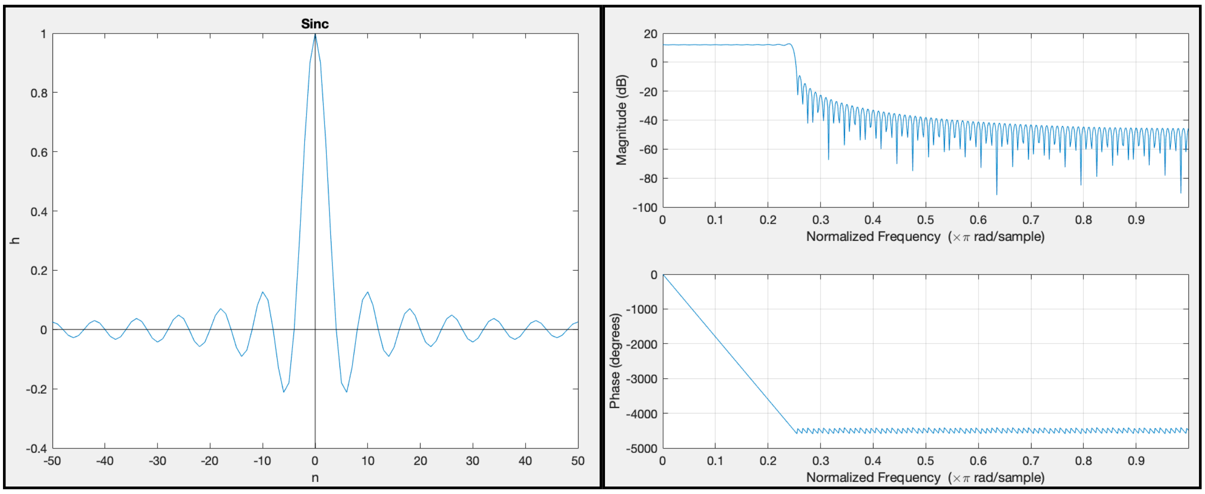

fig. 3

Here are the plot of sinc function and frequency response.

3. Implement the low pass filter by using command, which shown in the figure below.

```
% Q3
Y = filter(h,1,x2) % original LPF
```

fig. 4

4. Next, we convolve the low pass filter with pulses, then compared the original signal and upsampled signal over the range 30ms to 40ms. Code shown in figure5.

```
%Q4
figure()
plot(t2,Y)  % upsampled signal
hold on
plot(t1,x)  % original signal
hold off
xlim([0.03,0.04])
legend('upsampled signal','original signal')
```

fig. 5

Figure 6 showed the comparison of two signals, we can see two waveforms are not aligned.

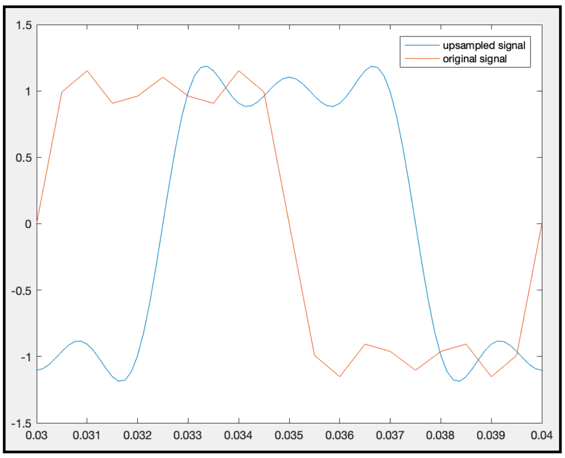

fig. 6

5. Implement a delay into the filter I created before, and make new filter. Then, use this filter to filter zero-inserted signal. We shifted the red waveform to the right. Code shown in figure 7.

```
%Q5
delay = zeros(1,L+1);
delay (L+1) = 1;
Y2 = filter(delay,1,x2) % new LPF (with delay)
figure()
plot(t2,Y2) % the original zero-inserted signal with new LPF
hold on
plot(t2,Y)  % the original zero-inserted signal with original LPF
hold off
xlim([0.03 0.04])
legend('signal with new LPF','signal with old LPF')
```

fig. 7

Figure 8 showed a good comparison of two waveforms. We can see the original signal has been recovered.

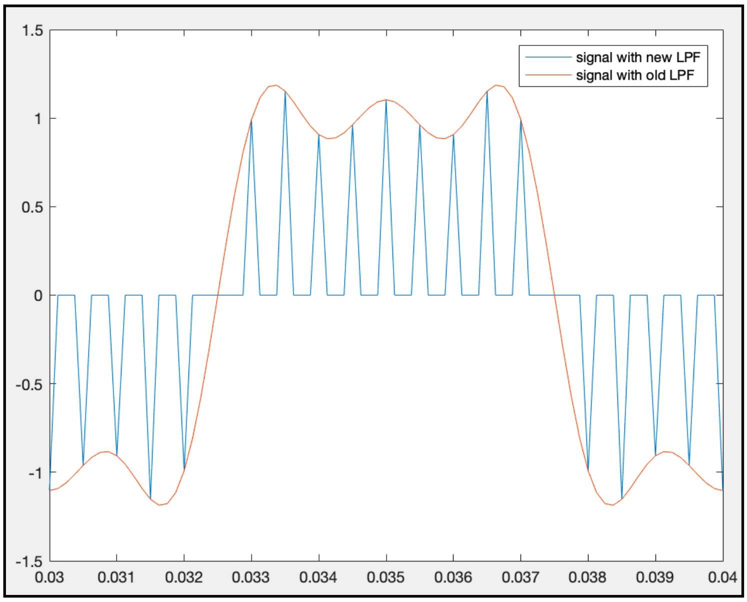

fig. 8