Wade Campney

DSP Project #1

1) Matlab script: Set up for question #2

clear all

Trep = 1e-6;

t1 = [0:Trep:.01];

f\_tone1 = 1000;

% Declaring x(t) and plotting it

xt = cos(2\*pi\*f\_tone1\*t1);

subplot(3,2,1);

plot(t1,xt);

title('x(t)');

% Calculating and plotting X(jw)

spectrum1 = fft(xt);

shiftedspectrum1 = fftshift(spectrum1);

f\_axis = linspace(-1/Trep/2,1/Trep/2,length(shiftedspectrum1));

subplot(3,2,2);

plot(f\_axis,abs(shiftedspectrum1));

title('X(jw)');

% Sampling frequency

ws = 5000;

Ts = 1/ws;

% Defining p(t) the impulse train

samprate = Ts/Trep;

range = [0:1:length(t1)-1];

pt = [1:1:length(t1)];

pt(find(pt>0)) = 0;

select = find(rem(range,samprate)==0);

pt(select) = ones(size(select));

% Calculating xs(t) the sampled signal

xst = pt.\*xt;

% Calculating and plotting Xs(jw)

sampledspectrum = fft(xst);

shiftedspectrum2 = fftshift(sampledspectrum);

shiftedspectrum2(find(abs(shiftedspectrum2)<10))=0;

f\_axis2 = linspace(-1/Trep/2,1/Trep/2,length(shiftedspectrum2));

subplot(3,2,3);

plot(f\_axis2,abs(shiftedspectrum2));

title('Xs(jw)');

% Creating the low pass filter Hr(jw)

Hrjw = zeros(1,length(t1));

for k = -ws/200:ws/200

Hrjw(k+5000) = Ts;

end

% Calculating and plotting Xr(jw)

Xrjw = shiftedspectrum2.\*Hrjw;

f\_axis3 = linspace(-1/Trep/2,1/Trep/2,length(Xrjw));

subplot(3,2,4);

plot(f\_axis3,abs(Xrjw));

title('Xr(jw)');

% Using a seperate ifft function to calculate xr(t)

xrt = ifftfunction(Xrjw);

% Plotting x(t) and xr(t) side by side to for comparison

subplot(3,2,5);

plot(t1,xt);

title('x(t)');

subplot(3,2,6);

plot(t1,xrt);

title('xr(t)');

IFFT function:

function [ xrt ] = ifftfunction( Xrjw )

%function that shifts and calculated the ifft

% Detaiailed explanation

shiftedsignal = ifftshift(Xrjw);

xrt = ifft(shiftedsignal);

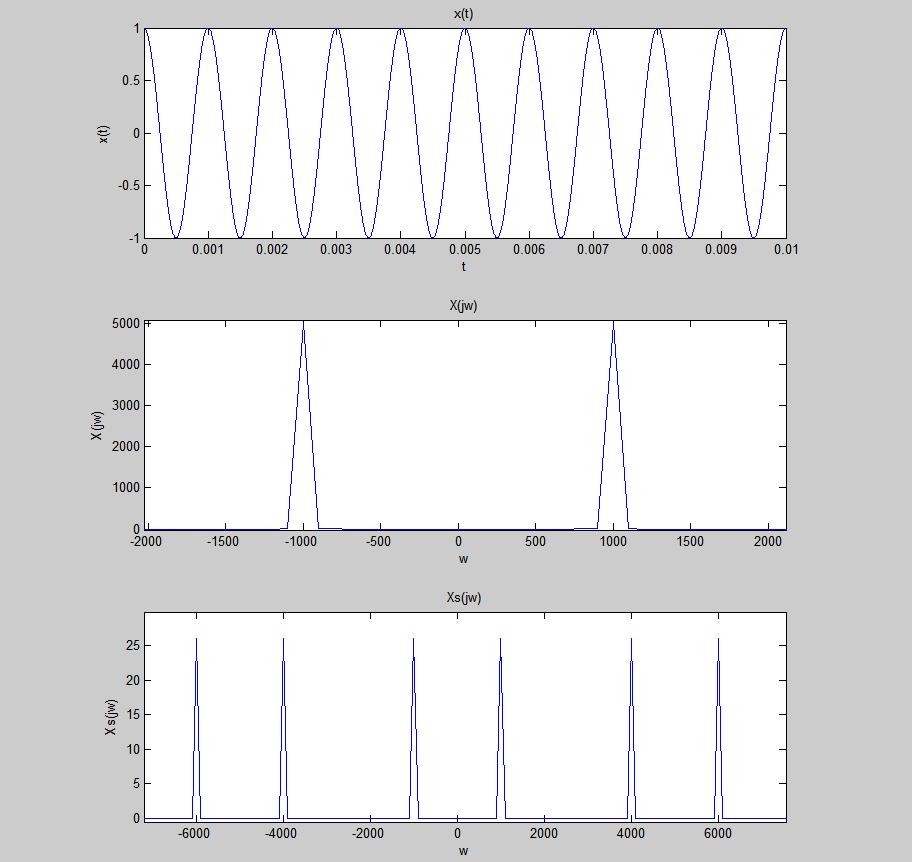
end

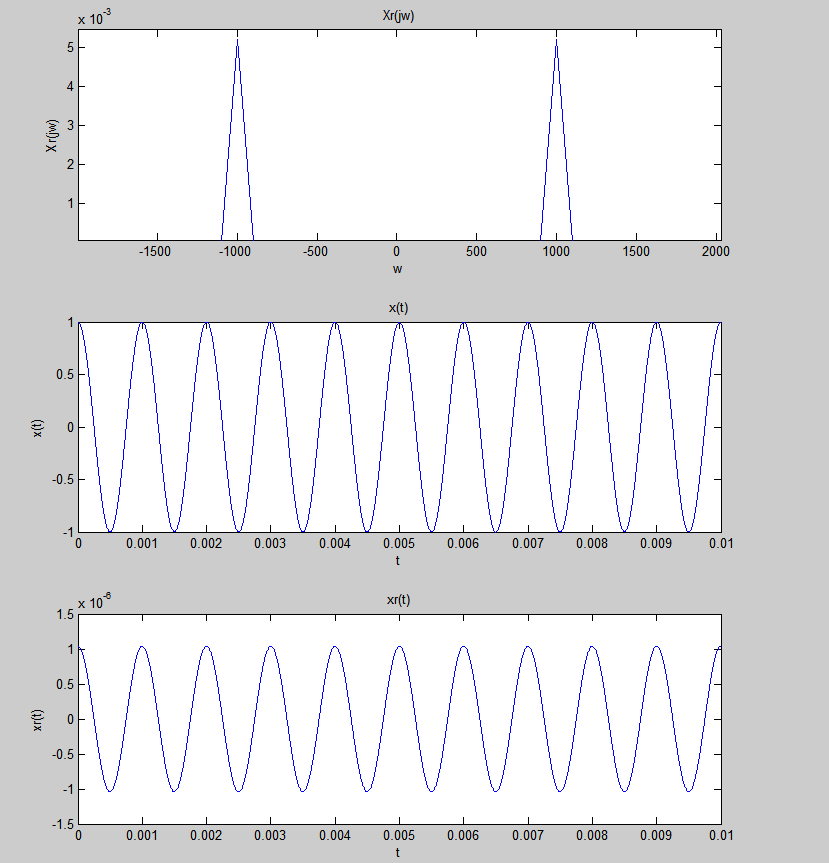
2)

a)The theoretical Fourier transform would be 2 delta functions with amplitudes equal to 1 , one at -1000 Hz and one at 1000 Hz. The sampled spectrum would created shifted versions of the original spectrum at an interval of± 5000 Hz. So there would be a delta at ±4000 Hz and ±6000 Hz and that pattern would continue on in each direction infinitely.

b) The reconstructed spectrum would be 2 delta functions with amplitudes equal to 1 , one at -1000 Hz and one at 1000 Hz.

c)





d) The matlab results matched the theoretical calculations exactly.

e) There is no aliasing in this case because the sampling frequency is greater than two times the greatest frequency in the signal. 2\*1000 Hz = 2000 5000 > 2000

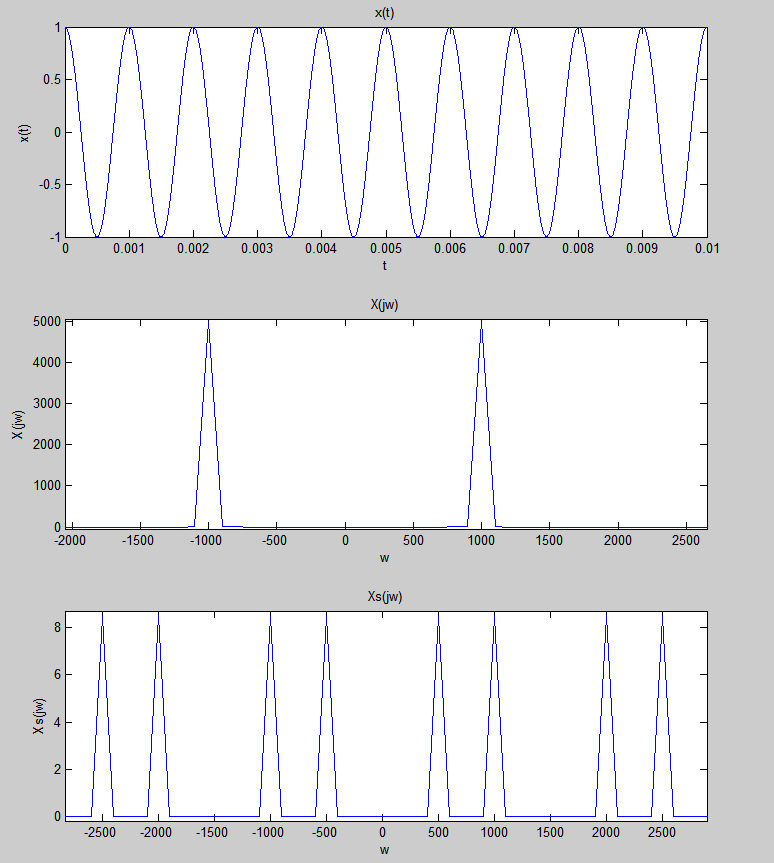
f) They appear to be the same and the only noticable difference is that they have different amplitudes due to the scaling done by matlab during the fft, filtering, and ifft. They have the same frequency.

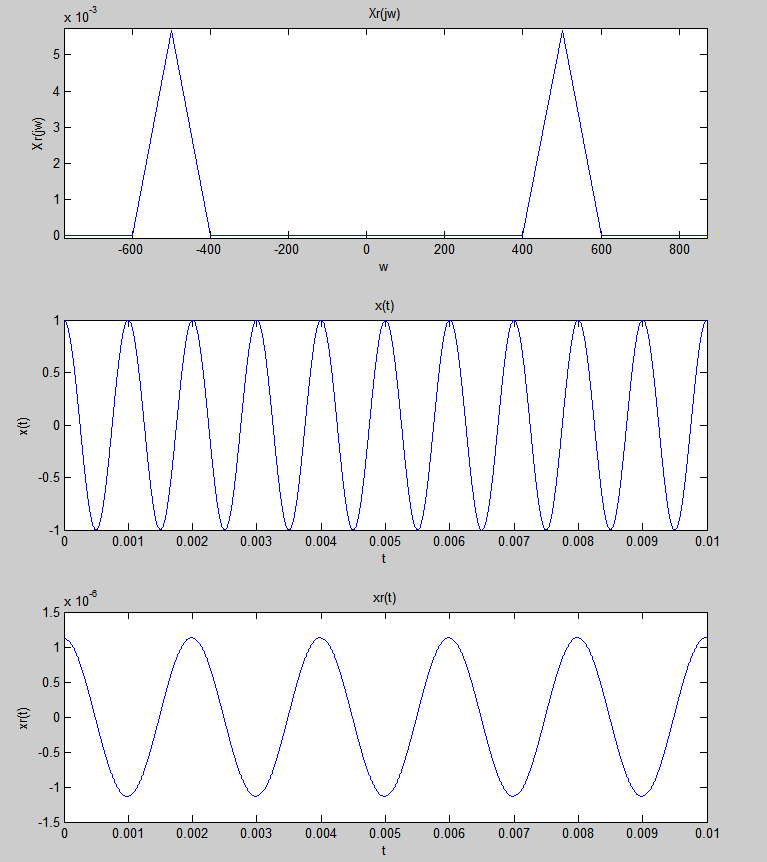
3)

a)The theoretical Fourier transform would be 2 delta functions with amplitudes equal to 1 , one at -1000 Hz and one at 1000 Hz. The sampled spectrum would created shifted versions of the original spectrum at an interval of ±1500 Hz. So there would be a delta at ±2500 Hz and ±500 Hz and that pattern would continue on in each direction infinitely.

b) The reconstructed spectrum would be 2 delta functions with amplitudes equal to 1 , one at -1000 Hz and one at 1000 Hz.

c)





d) The matlab results matched the theoretical calculations exactly.

e) There is aliasing in this case, the signal resulting from the reconstruction is not the same as the input. This occured because the sampling frequency was not greater than twice the greatest frequency in the signal. 2\*1000 Hz = 2000 1500 < 2000

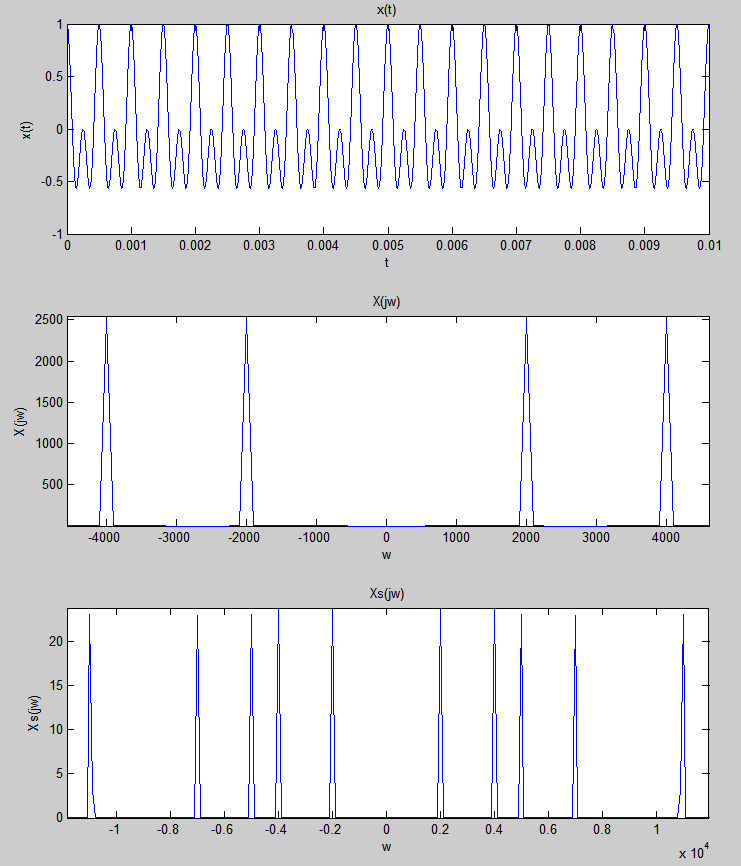
f) The reconstructed signal and the original are not the same, they have a different frequency and also amplitude but the difference in amplitude was caused by the scaling done by matlab during the fft, filtering, and ifft. The frequency difference was caused by aliasing.

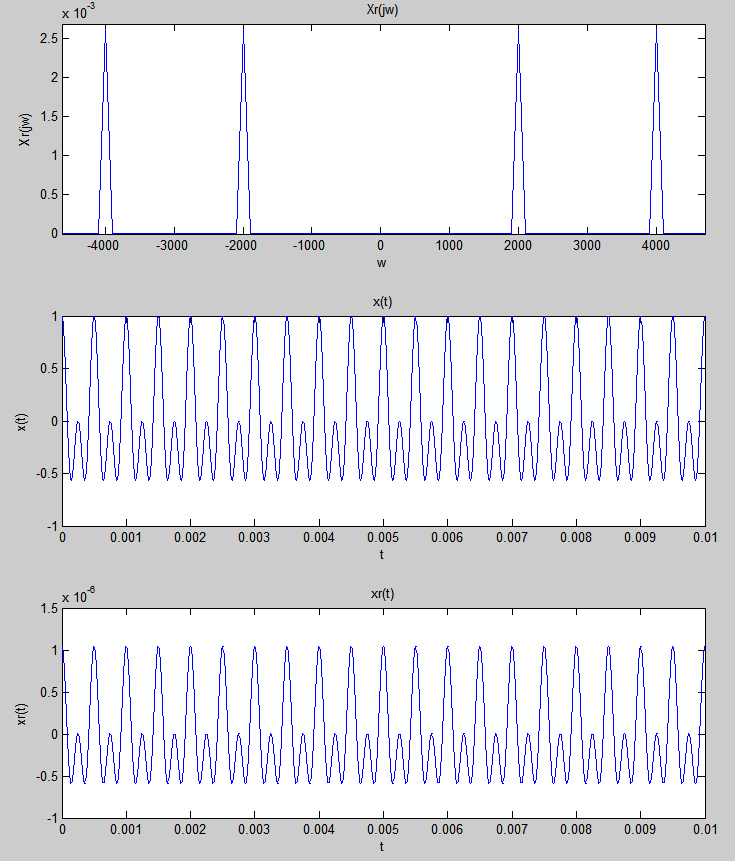
4)

a)The theoretical Fourier transform would be equal to 4 delta functions, each at an amplitude of 1/2 because of the cos modulation. This would also cause there to be two shifted spectrums each at ±3000 Hz. Then from there would be two deltas each offset ±1000 from the 3000 offset already in place. Then because of the sampling, this would be replicated every ±9000 Hz infinitely in both directions.

b) The reconstructed spectrum would be 4 delta functions, at ±2000 Hz and ±4000 Hz.

c)





d) The matlab results matched the theoretical calculations exactly.

e) There is no aliasing in this case because the sampling frequency is greater than two times the greatest frequency in the signal. 2\*4000 Hz = 8000 9000 > 6000

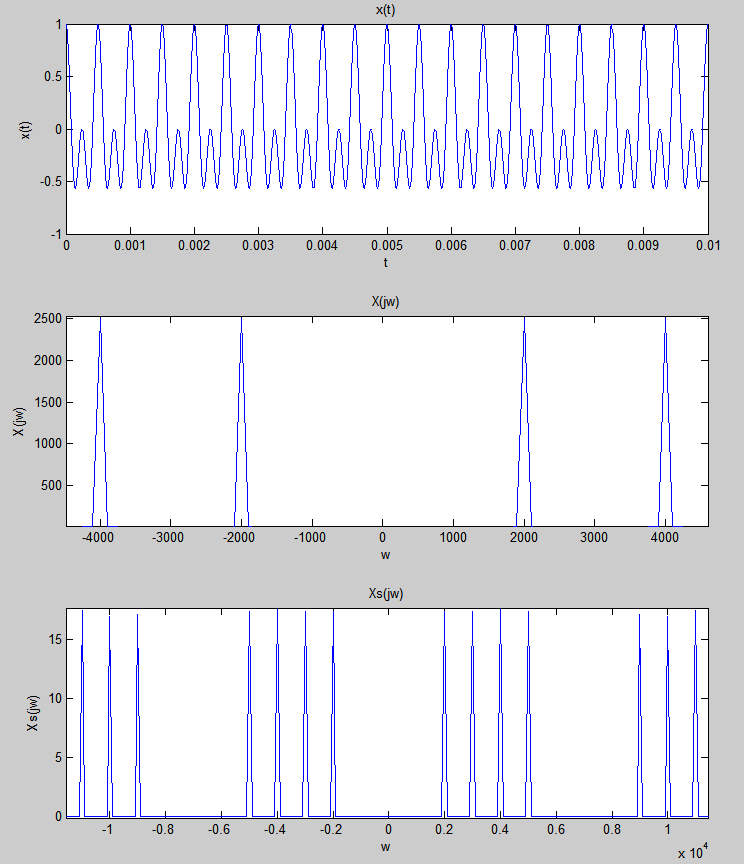
f) They appear to be the same and the only noticable difference is that they have different amplitudes due to the scaling done by matlab during the fft, filtering, and ifft. They have the same frequency.

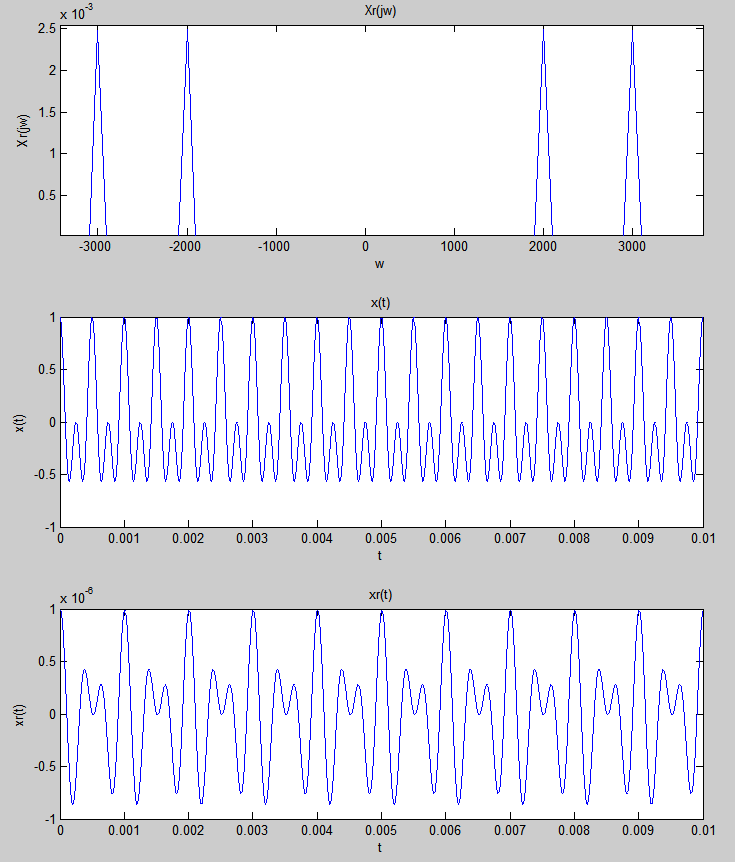
5)

a)The theoretical Fourier transform would be equal to 4 delta functions, each at an amplitude of 1/2 because of the cos modulation. This would also cause there to be two shifted spectrums each at ±3000 Hz. Then from there would be two deltas each offset ±1000 from the 3000 offset already in place. Then because of the sampling, this would be replicated every ±7000 Hz infinitely in both directions.

b) The reconstructed spectrum would be 4 delta functions, at ±2000 Hz and ±3000 Hz.

c)





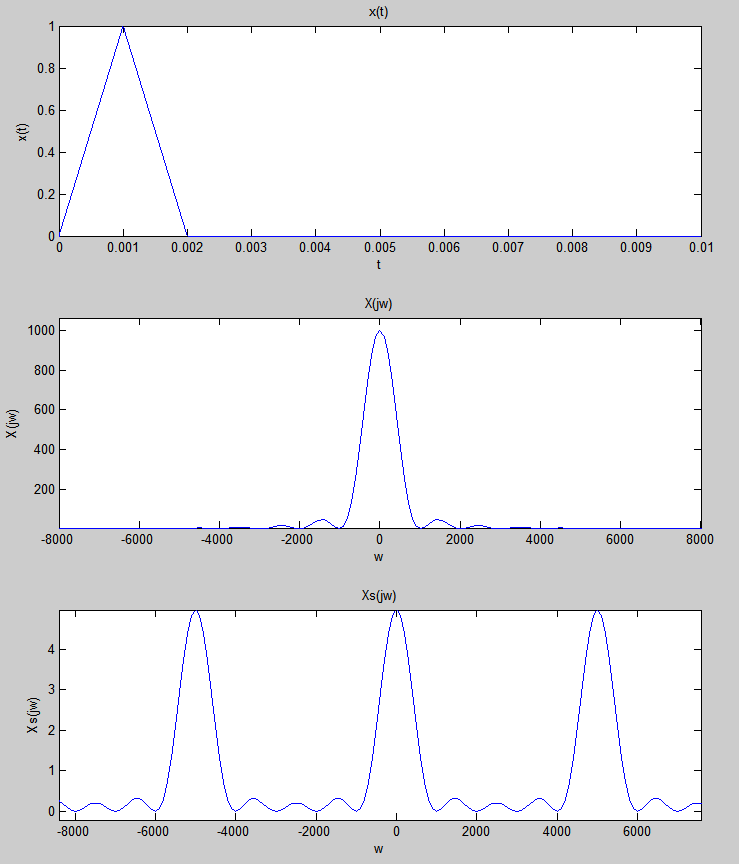
d) The matlab results matched the theoretical calculations exactly.

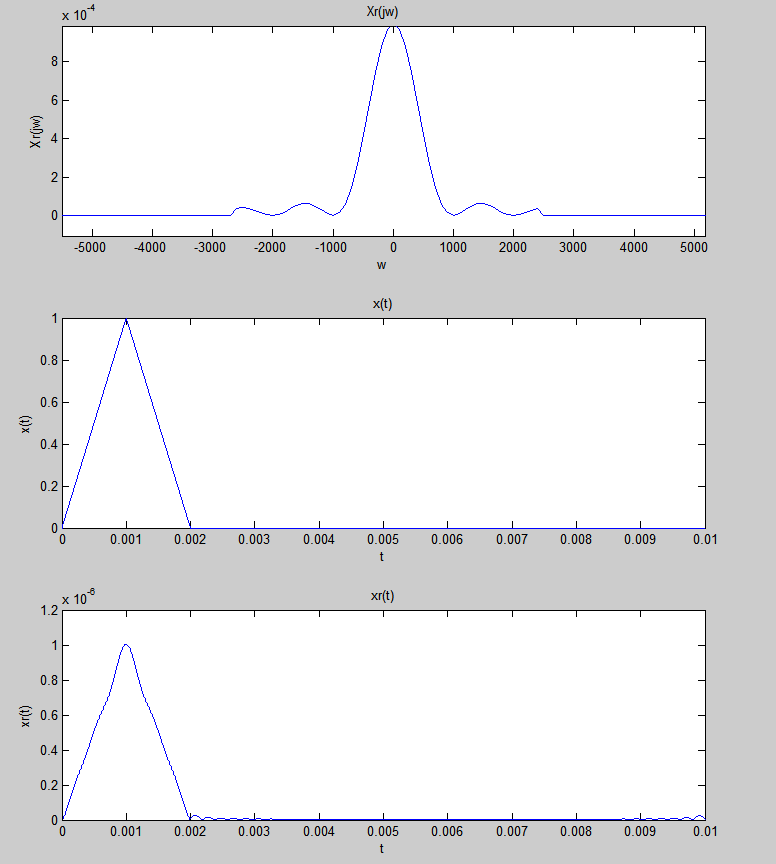
e) There is aliasing in this case, the signal resulting from the reconstruction is not the same as the input. This occured because the sampling frequency was not greater than twice the greatest frequency in the signal. 2\*4000 Hz = 8000 7000 < 8000

f) The reconstructed signal is still a combination of two cosine waves like the input but they are not the same frequency as the original cosines were. They are also of a different amplitude but that is due to the scaling done by matlab during the fft, filtering, and ifft.

6) The sinc2(jw) function that results from the Fourier transform on the triangular pulse is not bandlimited because the function never goes to zero, the oscillations just grow very small over as the frequency increases. So because of this, it should be impossible to get back a perfect triangular pulse unless the sampling frequency is huge. There are still visable oscilations in the plot of the spectrum X(jw) at around 5000 Hz, so the frequency should be greated than that to achieve minimal aliasing. I Believe there will be aliasing for each of the following cases.

7) Triangular Pulse



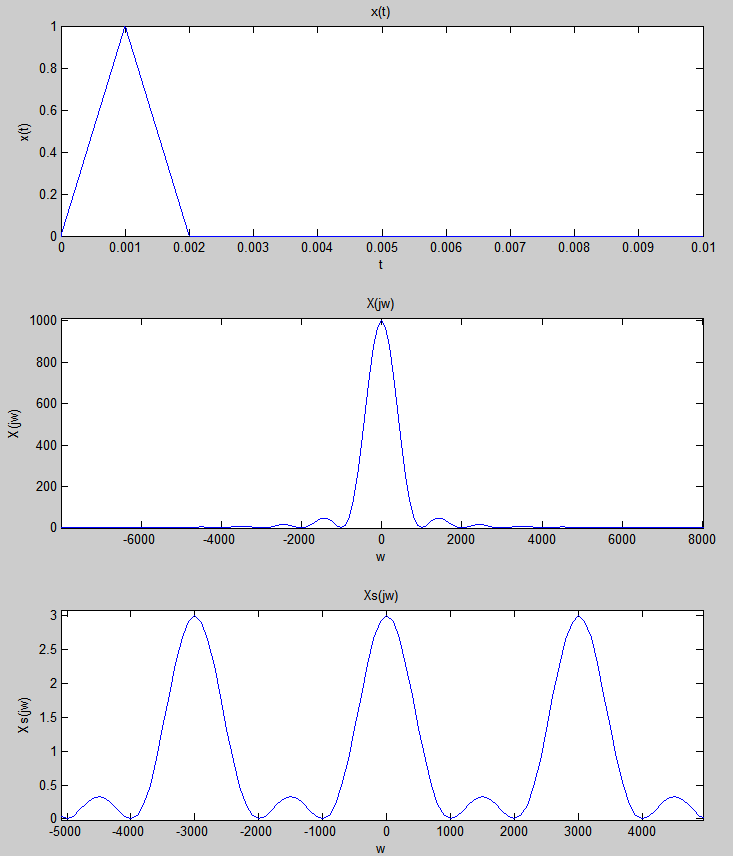


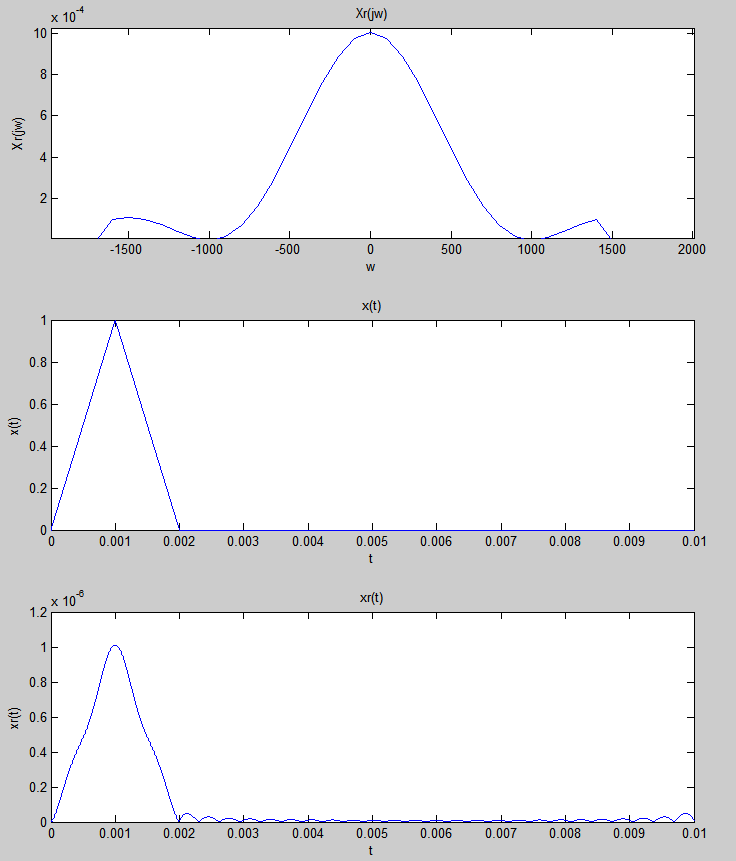
b) The matlab results matched the theoretical expectations.

c) There is minor aliasing in this case because the sampling frequency of 5000 Hz is pretty large but still not large enough to eliminate the aliasing because the sinc2(jw) function is not band limited.

d) The reconstructed signal is not exactly the same as the original because of the aliasing caused by the function not being band limited. It is not a perfect triangular pulse and there are also some small oscillations after the pulse caused by the aliasing.

8) Triangular Pulse



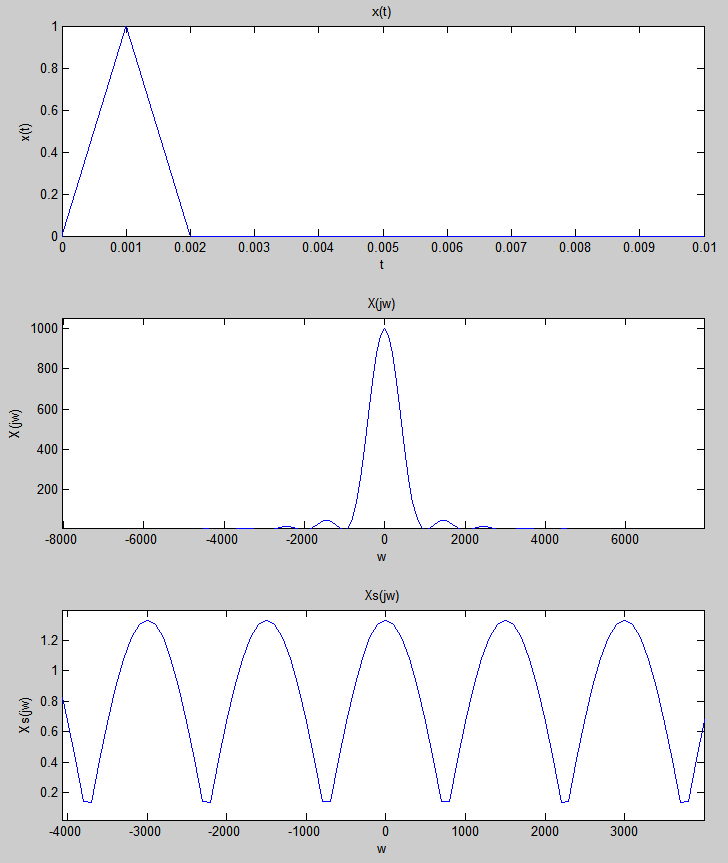


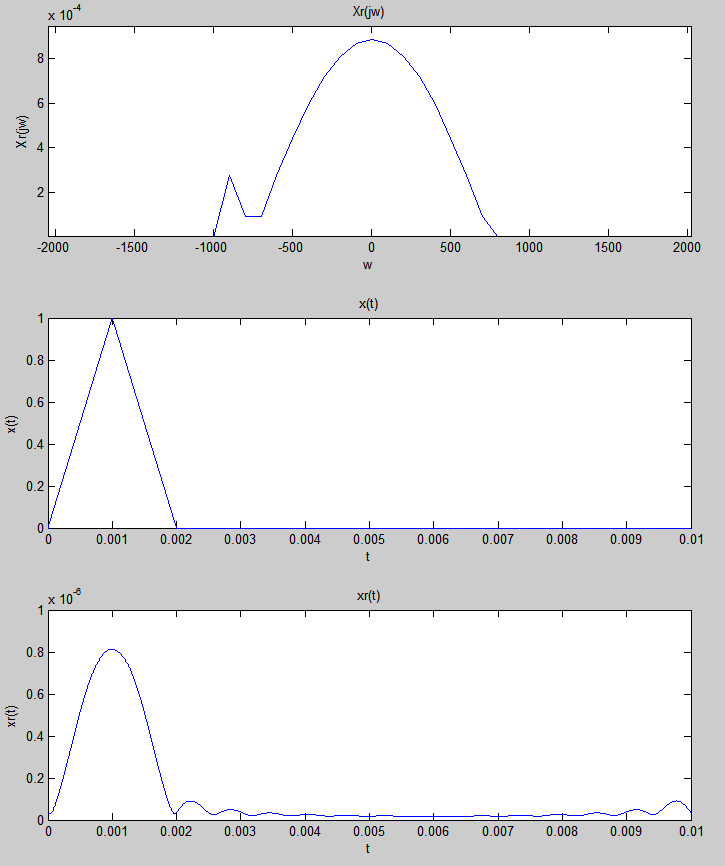
b) The matlab results matched the theoretical expectations.

c) There is a little more aliasing in this case than in the 5000 Hz case because the sampling frequency of 3000 Hz is pretty even less so it just made the aliasing slightly worse. It is still almost a triangular pulse but the oscillations after the pulse became worse. Again, this aliasing is caused because the sinc2(jw) function is not band limited.

d) The reconstructed signal is not exactly the same as the original because of the aliasing caused by the function not being band limited. It is not a perfect triangular pulse and there are some larger oscillations than last time after the pulse caused by the aliasing

8) Triangular Pulse





b) The matlab results matched the theoretical expectations.

c) There is some pretty bad aliasing in the case because the sampling frequency of 1500 Hz is so small. The sinc2(jw) functions are really overlapping in the sampled spectrum so when it is filtered, there is a big difference between X(jw) and Xr(jw). This causes the output to even more resemble a sinc2 wave rather than a triangular pulse like we wanted. Again, the aliasing is because the sampling frequency was too small and the sinc2 wave is not band limited.

d) No the reconstructed signal and the original do not match, they are actually quite different in this case. The reconstructed signal became almost like a sinc2 signal, it is way more rounded and the oscillations after it have become quite large and it is basically a sinc2 function now.