Xiaobin Zhuang - 300519184

ECEN220 Lab 3

18/07/2020

Objectives

The primary object of this lab was to do Fourier transform to several different signals by using MATLAB.

Methodology

1.

(a) Plotting x(t) = cos(w0*t) on t = [-5, 5] for w0 = 2*pi/10, with step size 1/50 (if the step size was small, cos wave would not be smooth).

(b) Then plotted Fourier transform of cosine X(w). Because X(w) is complex, I plotted two graph: Magnitude and Phase of Fourier transform. To have a clear graph of Magnitude and Phase, I used fftshift() to shift the impulses of Magnitude (on the right and left) to the centre.

Matlab code and graph are shown in figure below.

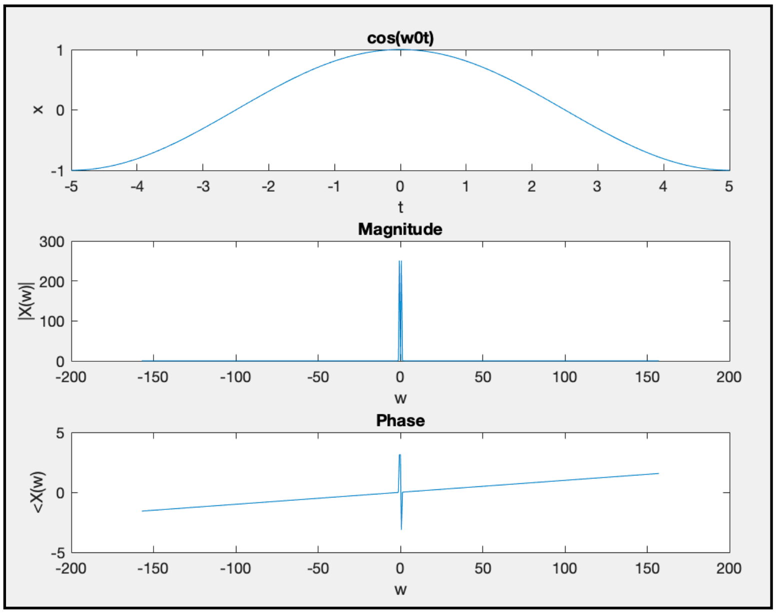

fig. 1

```
clear all
t = -5:1/50:5;
w0 = 2*pi/10;
x = cos(w0*t);
subplot(311)
plot(t,x)
title('cos(w0t)')
xlabel('t')
ylabel('x')

%(b)
y = fftshift(fft(x));
f = (t*w0/(1/50));
subplot(312)
plot(f,abs(y))
title('Magnitude')
xlabel('w')
ylabel('|X(w)|')
subplot(313)
plot(f,angle(y))
title('Phase')
xlabel('w')
ylabel('<X(w)')
```

fig. 2

(c) The theoretical Fourier transform of cosine is $\pi[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)]$, it isn't a complex number. However, in Matlab, after doing fft(), it became a complex number. Which shown in figure 3.

fig. 3

y

1x501 complex double

	1	2	3	4	5	
1	3.8810e-...	3.4930e-...	9.7039e-...	1.9023e-...	3.1452e-...	4.

(d) when I reduced the step size to 1/100, the impulse of magnitude and phase became narrower. Which shown together with Matlab code in figure 4.

```
%% (d)
clear all
t = -5:1/100:5; % step size reduced to 1/100
w0 = 2*pi/10;
x = cos(w0*t);
subplot(311)
plot(t,x)
title('cos(w0t)')
xlabel('t')
ylabel('x')

y = fftshift(fft(x));
f = (t*w0/(1/100));
subplot(312)
plot(f,abs(y))
title('Magnitude')
xlabel('w')
ylabel('|X(w)|')
subplot(313)
plot(f,angle(y))
title('Phase')
xlabel('w')
ylabel('<X(w)')
```

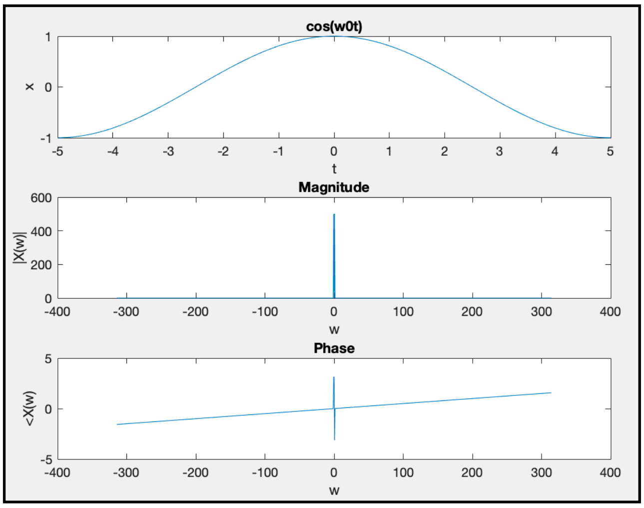

fig. 4

Because after I reduced the step size, sampling point 't' was increased, and 'f' was also increased. It needed to do Fourier transform with more sampling points, so the graph looked narrower.

2.

(a) x(t) = sin(ω_0t) (similar with cosine)

```
%% Question 2 (a) 1.sine
clear all
t1 = 0:1/50:5;
w0 = 2*pi/10;
x1 = sin(w0*t1);
subplot(311)
plot(t1,x1)
title('sin(w0t)')
xlabel('t1')
ylabel('x1')

y1 = fftshift(fft(x1));
f1 = (t1*w0/(1/50));
subplot(312)
plot(f1,abs(y1))
title('Magnitude')
xlabel('w')
ylabel('|X(w)|')
subplot(313)
plot(f1,angle(y1))
title('Phase')
xlabel('w')
ylabel('<X(w)')
```

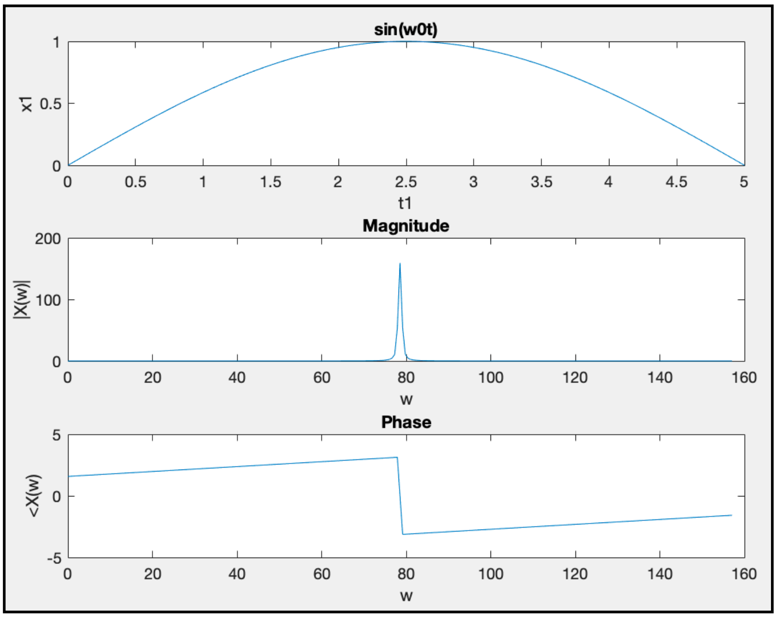

fig. 5

However, the Fourier transform of sine wave had only one impulse over one period, different with cosine which had two impulse.

$x(t) = e^{j\omega_0 t}$

```
%% 2.exponetial
clear all
t2 = 0:1/50:10;
w0 = 2*pi/10;
x2 = exp(j*w0*t2);
subplot(411)
plot(t2,real(x2))
title('Real')
xlabel('t2')
ylabel('x2')
subplot(412)
plot(t2,imag(x2))
title('Imaginary')
xlabel('t2')
ylabel('x2')

y2 = fftshift(fft(x2));
f2 = (t2*w0/(1/50));
subplot(413)
plot(f2,abs(y2))
title('Magnitude')
xlabel('w')
ylabel('|X(w)|')
subplot(414)
plot(f2,angle(y2))
title('Phase')
xlabel('w')
ylabel('<X(w)')
```

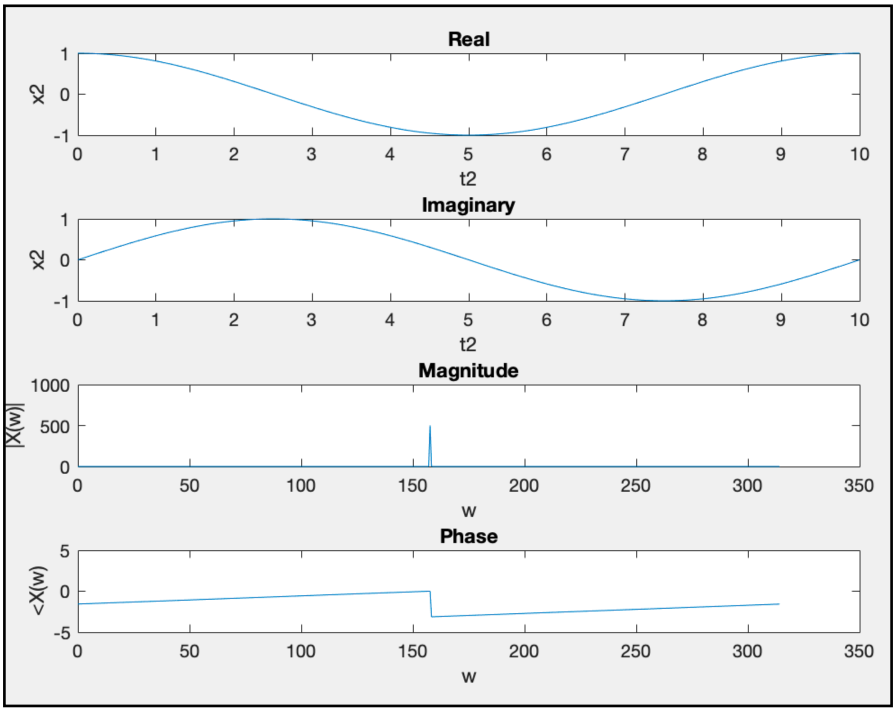

fig. 6

Exponential signal was a complex number, so I plot real and imaginary part separately.

Shown in figure 6.

Fourier transform of e^j*w0*t is $2\pi\delta(\omega-\omega_0)$, there was an impulse on the plot of magnitude, the position is more right than the Fourier transform of sine. Because the value of $2\pi\delta(\omega-\omega_0)$ is larger than $\frac{\pi}{j}[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)]$

$x(t) = \delta(t)$

```
%% 3.delta
clear all
t3 = -1:0.1:1;
x3 = dirac(t3);
idx = x3 == Inf; % find Inf
x3(idx) = 1;     % set Inf to finite value
subplot(311)
stem(t3,x3)

y3 = fft(x3);
f3 = (t3/0.1);
subplot(312)
plot(f3,abs(y3))
title('Magnitude')
xlabel('w')
ylabel('|X(w)|')
subplot(313)
plot(f3,angle(y3))
title('Phase')
xlabel('w')
ylabel('<X(w)')
```

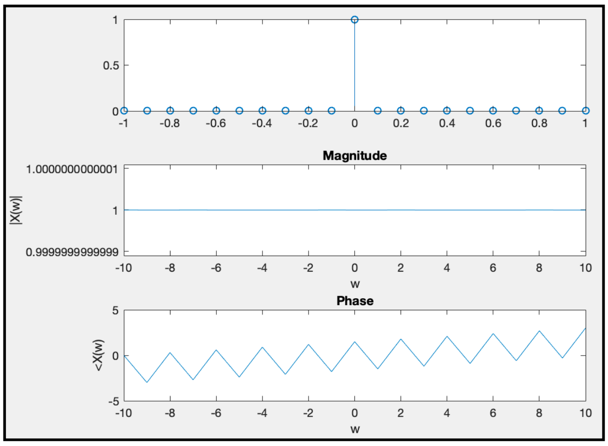

fig. 7

Figure 7 showed the code and plot of delta(t) Fourier transform. The Fourier transform of delta(t) is 1, which match the magnitude plot.

x(t) = u(t)

```
clear all
t4 = -1:0.0001:1;
x4 = heaviside(t4);
subplot(311)
plot(t4,x4)
title('u(t)')
xlabel('t4')
ylabel('x4')

y4 = fftshift(fft(x4));
f4 = (t4/0.0001);
subplot(312)
plot(f4,abs(y4))
title('Magnitude')
xlabel('w')
ylabel('|X(w)|')
subplot(313)
plot(f4,angle(y4))
title('Phase')
xlabel('w')
ylabel('<X(w)')
```

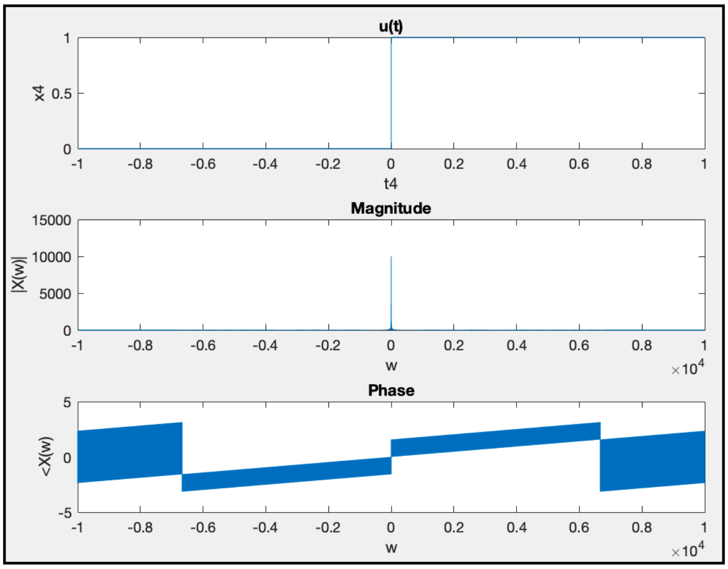

fig. 8

For unit step function, I set step size into a small number (0.0001) was because the unit step graph would be more accurate if there were enough points.

x(t)=δ(t–t0)

```
%% 5.delta(t-t0)
clear all
t5 = -1:0.1:1;
x5 = dirac(t5-0.6);
idx = x5 == Inf; % find Inf
x5(idx) = 1;     % set Inf to finite value
subplot(311)
stem(t5,x5)
title('delta(t-0.6)')
xlabel('t5')
ylabel('x5')

y5 = fft(x5);
f5 = (t5/0.1);
subplot(312)
plot(f5,abs(y5))
title('Magnitude')
xlabel('w')
ylabel('|X(w)|')
subplot(313)
plot(f5,angle(y5))
```

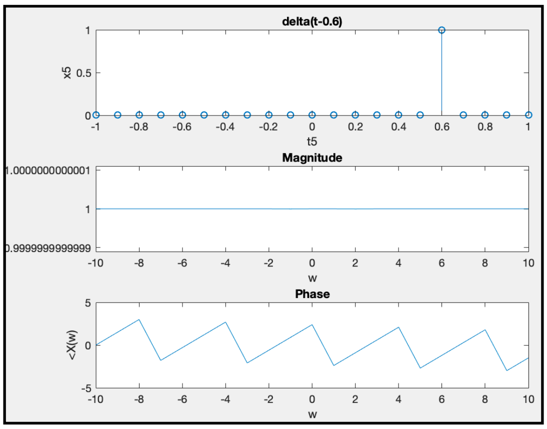

fig. 9

Shift delta(t) to delta(t-0.6), the impulse shift from 0 to 0.6. Which shown in figure 9.

Theoretical Fourier transform of delta(t-t0) is e^(-jwt0), however, the magnitude graph showed the value is constant 1.

x(t) = e^(–at) *u(t), R]{a} > 0

```
%% x(t) = e^-at*u(t)
clear all
t6 = -5:0.0001:5;
ut = heaviside(t6);
x6 = exp(-t6).*ut;
subplot(311)
plot(t6,x6)
title('exp(-t6).* u(t)')
xlabel('t6')
ylabel('x6')

y6 = fftshift(fft(x6));
f6 = t6/0.0001;
subplot(312)
plot(f6,abs(y6))
title('Magnitude')
xlabel('w')
ylabel('|X(w)|')
subplot(313)
plot(f6,angle(y6))
title('Phase')
```

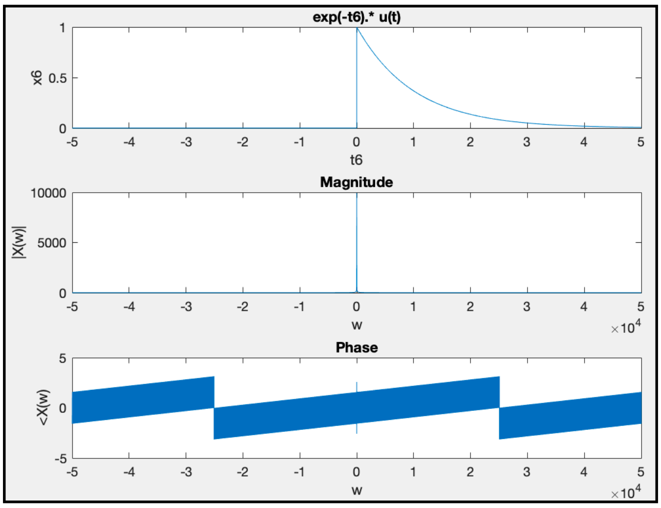

fig. 10

Set a = 1, when t<0, u(t) = 0, x(t) = 0; t>0, u(t) = 1, x(t) = e^-t. The graph shown in figure 10. Theoretical Fourier transform of e^-at is 1/(a+jw), however, according to the magnitude graph I plotted, it only had value when w=0.

- $x(t) = \begin{cases} 1, & |t| < T_1 \\ 0, & T_1 < |t| < \frac{T}{2} \end{cases}$

I tried to use for loop and if statement to plot x(t), however it didn't work..

Code shown in figure 11.

```
%% x(t) = 1, |t|<T1; 0, T1<|t|<T/2
clear all
for t7 = -10:10
T = 4;
T1 = 1;
if abs(t7) < T1
   x7 = 1;
else
    T1 < abs(t7) && abs(t7) < T/2
        x7 = 0;
    end
end
plot(t7, x7*ones(size(t7)))
```

fig. 11

Instead of using that method, I used the other way to plot square wave with 30% duty cycle. Although it was little bit different with the required square wave. Shown in figure 12.

```
t7 = -10:0.01:10;
x7 = square(t7,30); % 30% duty cycle
subplot(311)
plot(t7,x7,'linewidth',2)
title('square wave')
ylabel('x7')
xlabel('t7')

y7 = fftshift(fft(x7));
f7 = t7/0.01;
subplot(312)
plot(f7,abs(y7))
title('Magnitude')
xlabel('w')
ylabel('|X(w)|')
subplot(313)
plot(f7,angle(y7))
title('Phase')
xlabel('w')
ylabel('<X(w)')
```

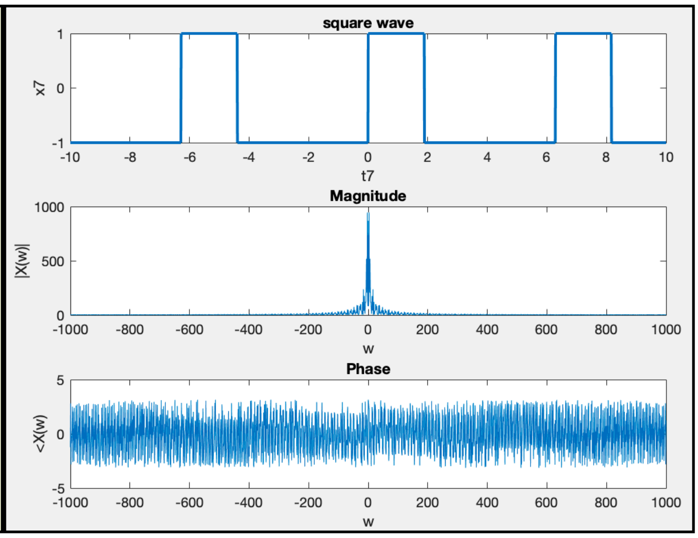

fig. 12