SS EXPERIMENT LAB 6

TITLE: Verification of different properties of LTI system: Commutation,

Association, Distribution, Identity and Fourier analysis and synthesis

NAME: Yash Gupta

ROLL NO: S20200010234

OBSERVATION: In this lab, I learned how to verify different properties of LTI system and Fourier analysis and synthesis.

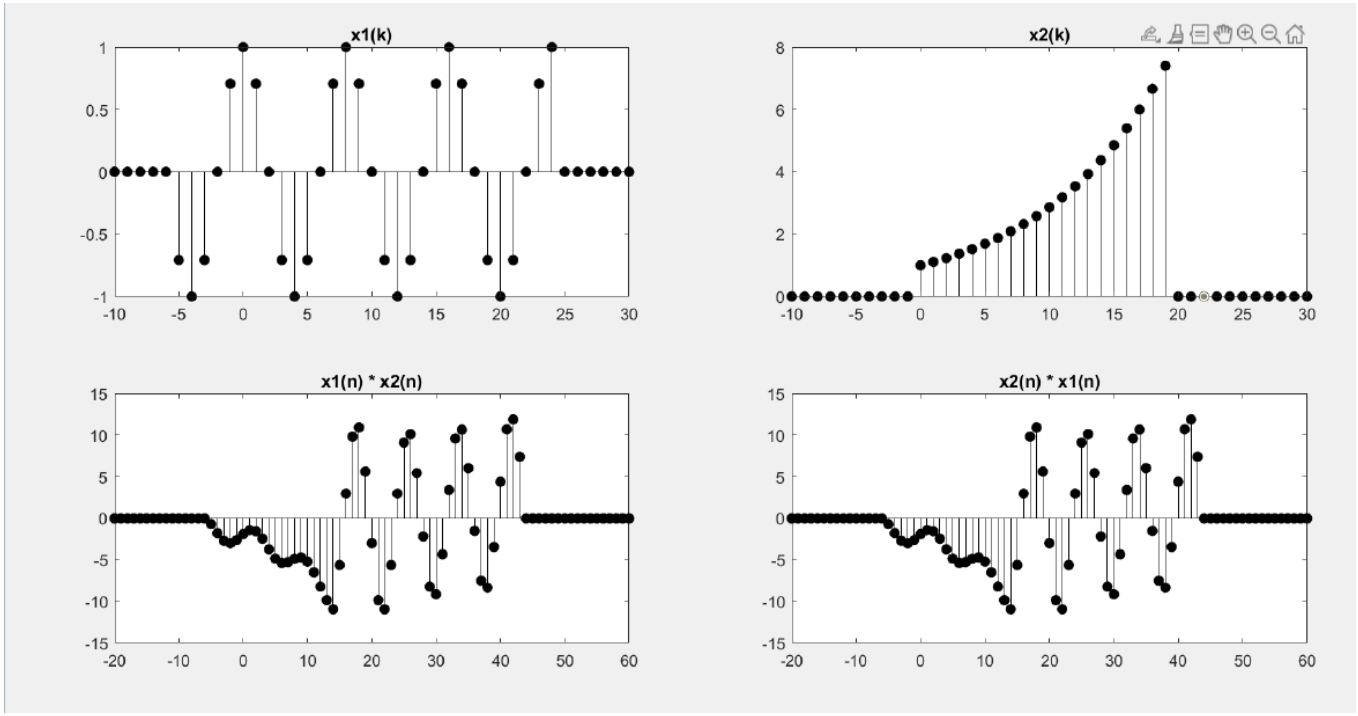
Verify the above properties for the following signals **by writing a MATLAB script**. Next, also verify the result obtained in MATLAB script with Analytical method.

Convolution function:

```
function [ny,y] = convulation(nx,x,nh,h)
nyb=nx(1)+nh(1);
nye=nx(length(x))+nh(length(h));
ny=nyb:nye;
y=zeros(1,length(ny));
for i=1:length(ny)
    [hf, nhf] =sigfold(h,nh);
    [shf, nshf] = sigshift(hf,nhf, ny(i));
    [xnew, nm] = sigmult(x,nx,shf,nshf);
    y(i) = sum(xnew);
end
```

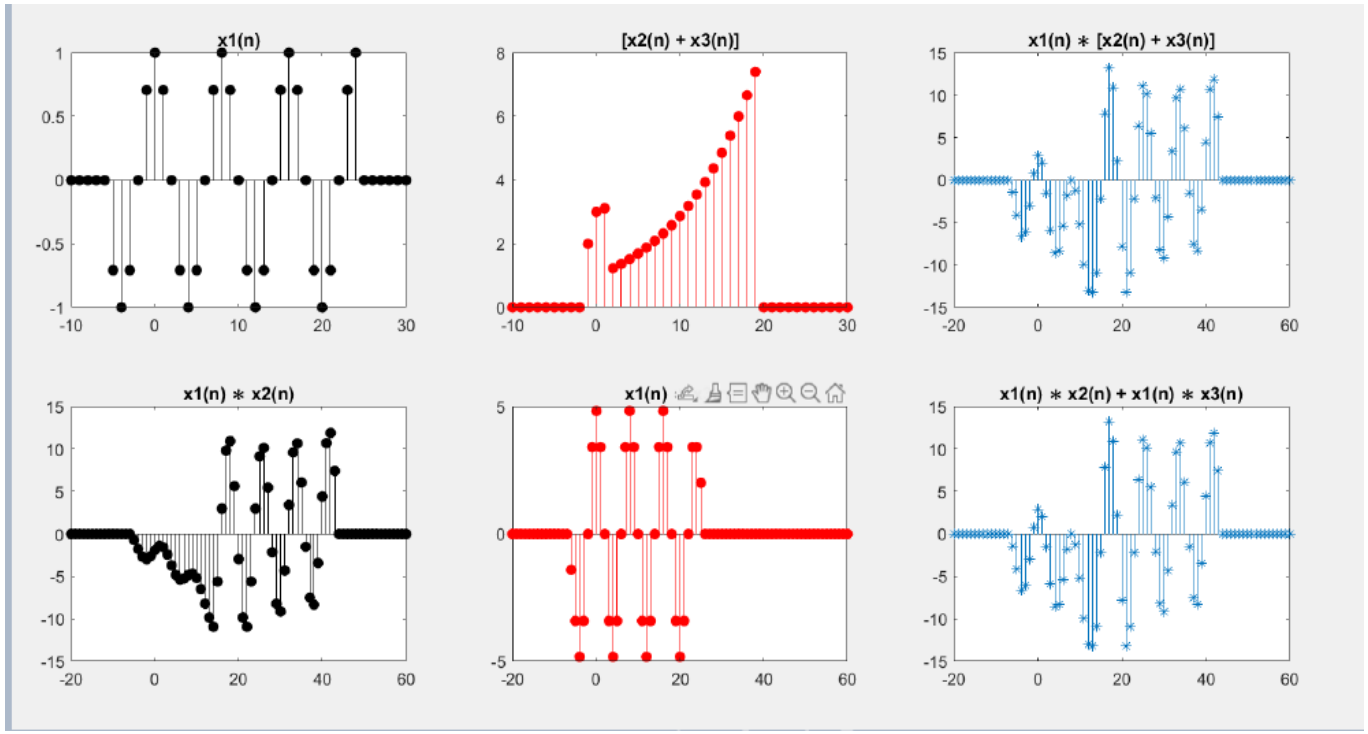
Commutative law

```
clear;
close all;
nx1 = -10:30 ;
x1 = cos((pi * nx1)/4).*(stepseq(-5,-10,30) - stepseq(25,-10,30));
nx2 = -10:30 ;
x2 = ((0.9).^-nx2).*(stepseq(0,-10,30) - stepseq(20,-10,30));
%%commutative law:
[ny,y] = convulation(nx2,x2,nx1,x1);

subplot(2,2,1);
stem(nx1, x1, 'k', 'filled');
title('x1(k)');
subplot(2,2,2);
stem(nx2, x2, 'k', 'filled');
title('x2(k)');
subplot(2,2,3);
stem(ny, y,'k','filled');
title('x1(n) * x2(n)');

[ny,y] = convulation(nx2,x2,nx1,x1);

subplot(2,2,4);
stem(ny, y,'k','filled');
title('x2(n) * x1(n)');
```


x1(k)
x2(k)
x1(n) * x2(n)
x2(n) * x1(n)

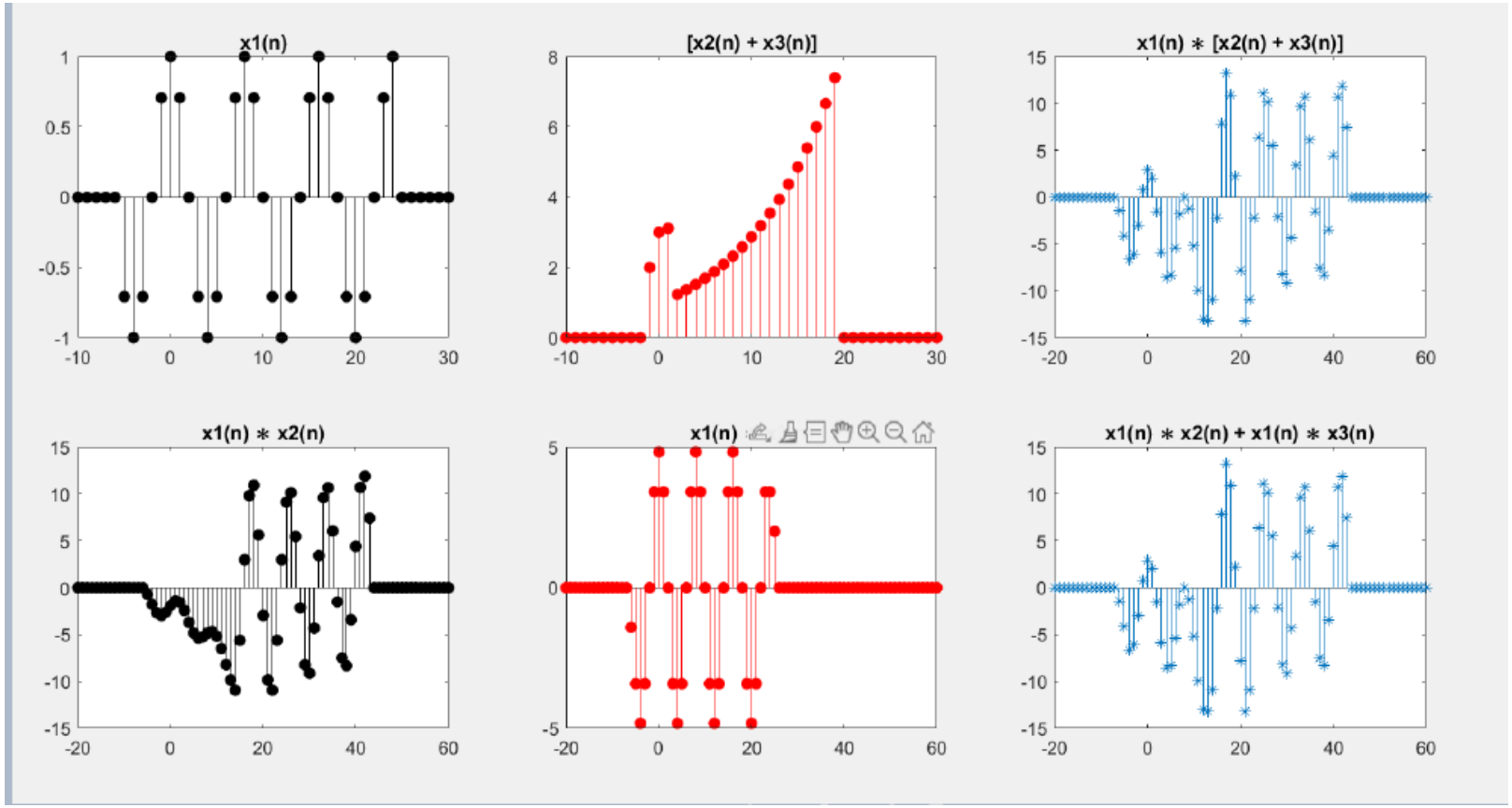
Association property

```
clear;
close all;
nx1 = -10:30 ;
x1 = cos((pi * nx1)/4).*(stepseq(-5,-10,30) - stepseq(25,-10,30));
nx2 = -10:30 ;
x2 = ((0.9).^-nx2).*(stepseq(0,-10,30) - stepseq(20,-10,30));
nx3 = -10:10 ;
wn = [0,0,0,0,0,0,0,0,0,1/3,1/3,1/3,0,0,0,0,0,0,0,0,0];
x3 = round(5 .* wn);
%association property%
[ny,y] = convulation(nx1,x1,nx2,x2);
subplot(2,3,1);
stem(ny, y, 'k', 'filled');
title('x1(n) * x2(n)');
[nz,z] = convulation(ny,y,nx3,x3);
subplot(2,3,2);
stem(nx3, x3, 'k', 'filled');
title('x3(n)');
subplot(2,3,3);
stem(nz, z, 'k', 'filled');
title('[x1(n) * x2(n)] * x(n)');
[ny,y] = convulation(nx2,x2,nx3,x3);
subplot(2,3,4);
stem(ny, y, 'k', 'filled');
title('x2(n) * x3(n)');
[nz,z] = convulation(nx1,x1,ny,y);
subplot(2,3,5);
stem(nx1, x1, 'k', 'filled');
title('x1(n)');
subplot(2,3,6);
stem(nz, z, 'k', 'filled');
title('x1(n) * [x2(n) * x3(n)]');

```


x1(n)
[x2(n) + x3(n)]
x1(n) ∗ [x2(n) + x3(n)]
x1(n) ∗ x2(n)
x1(n)
x1(n) ∗ x2(n) + x1(n) ∗ x3(n)

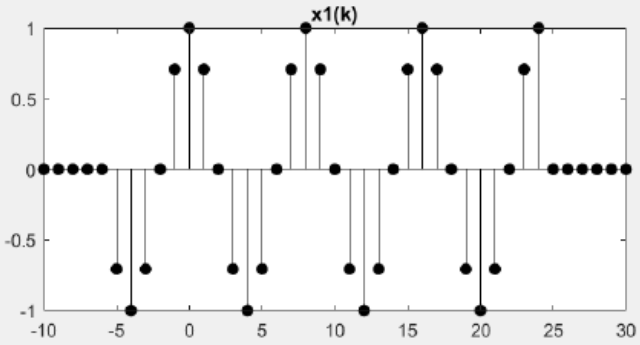
Distribution Law

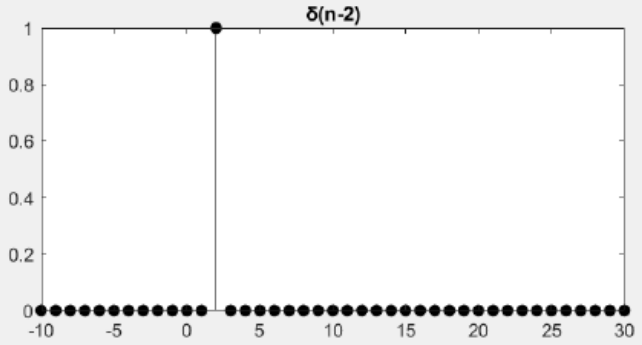
```
clear;
close all;
nx1 = -10:30 ;
x1 = cos((pi * nx1)/4).*(stepseq(-5,-10,30) - stepseq(25,-10,30));
nx2 = -10:30 ;
x2 = ((0.9).^-nx2).*(stepseq(0,-10,30) - stepseq(20,-10,30));
nx3 = -10:30 ;
wn = [0,0,0,0,0,0,0,0,0,1/3,1/3,1/3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
x3 = round(5 .* wn);
nx4 = -10:30;
x4 = x2 + x3 ;
subplot(2,3,1);
stem(nx1, x1, 'k', 'filled');
title('x1(n)');
subplot(2,3,2);
stem(nx4, x4, 'r', 'filled');
title('[x2(n) + x3(n)]')
[ny,y] = convulation(nx1,x1,nx4,x4);
subplot(2,3,3);
stem(ny, y, '*', 'filled');
title('x1(n) * [x2(n) + x3(n)]');
[ny,y] = convulation(nx1,x1,nx2,x2);
[nz,z] = convulation(nx1,x1,nx3,x3);
nw = -20:60;
w = y + z;
subplot(2,3,4);
stem(ny, y, 'k', 'filled');
title('x1(n) * x2(n)');
subplot(2,3,5);
stem(nz, z, 'r', 'filled');
title('x1(n) * x3(n)');
subplot(2,3,6);
stem(nw, w, '*', 'filled');
title('x1(n) * x2(n) + x1(n) * x3(n)');
```

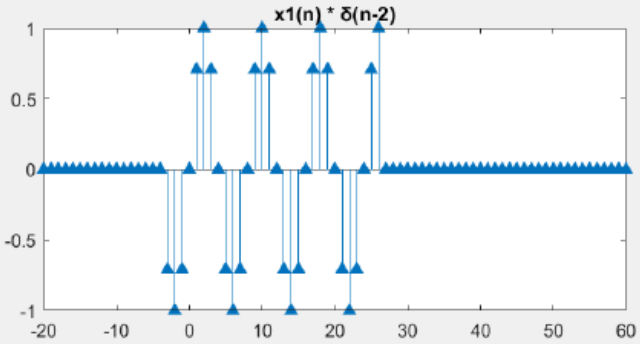

x1(n)
[x2(n) + x3(n)]
x1(n) ∗ [x2(n) + x3(n)]
x1(n) ∗ x2(n)
x1(n)
x1(n) ∗ x2(n) + x1(n) ∗ x3(n)

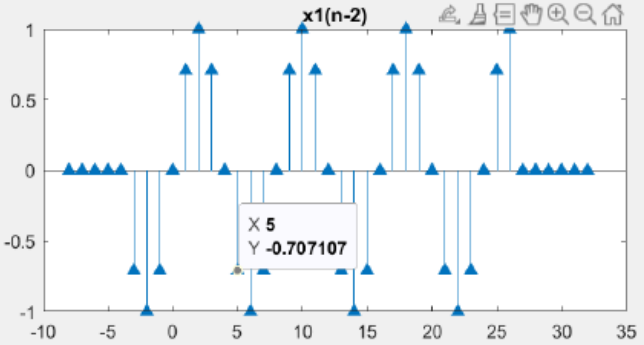
Identity Property

```
clear;
close all;
nx1 = -10:30 ;
x1 = cos((pi * nx1)/4).*(stepseq(-5,-10,30) - stepseq(25,-10,30));
nh = -10:30;
h = impseq(2,-10,30);
% identity property%
[ny,y] = convulation(nx1,x1,nh,h);
subplot(2,2,1);
stem(nx1, x1, 'k', 'filled');
title('x1(k)');
subplot(2,2,2);
stem(nh, h, 'k', 'filled');
title('δ(n-2)');
subplot(2,2,3)
stem(ny, y, '^','filled');
title('x1(n) * δ(n-2)');
[z,n] = sigshift(x1,nx1,2);
subplot(2,2,4);
stem(n, z,'^','filled');
title('x1(n-2)');

```


x1(k)


δ(n-2)

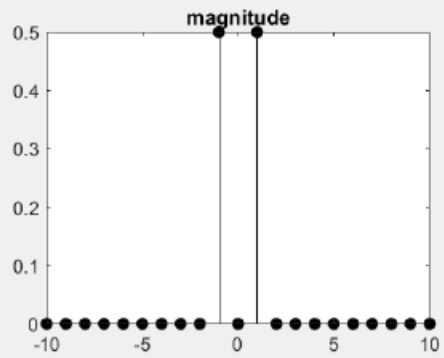

x1(n) * δ(n-2)

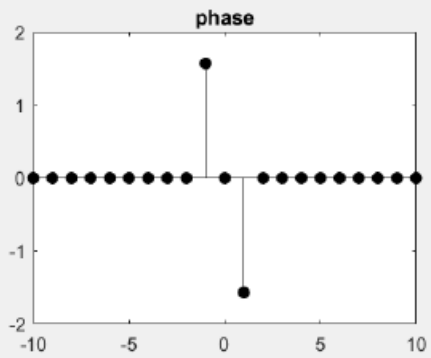

x1(n-2)
X 5
Y -0.707107

Task 2: Q1 a

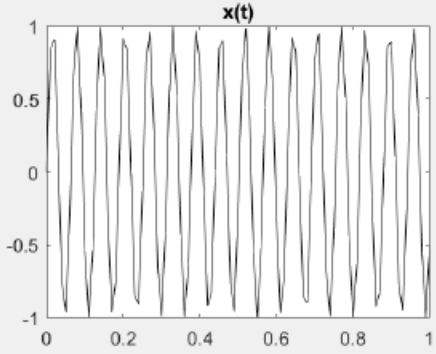
```
close all;
clear;syms x;syms t;
x = sin(100*t);
w0 = 100;
N = 10;
ak = zeros(1,2*N+1);
T = 2*pi/w0; %claculating the period and store in T
syms t;
%loop for calculating fourier coefficient
for k = -N:N
ak(1,1+k+N) = 1/T * int(x * exp(-1j*k*w0*t), 0, T) ;% ak is fourier coefficient
disp(ak(1+k+N));
end
k = -N:N;
mag = sqrt(real(ak).*real(ak) + imag(ak).* imag(ak));%magnitude of fourier coefficient
subplot(2,3,1);
stem(k,mag,'k','filled');
title('magnitude');

phase = angle(ak);
subplot(2,3,2);
stem(k,phase,'k','filled');
title('phase');
```

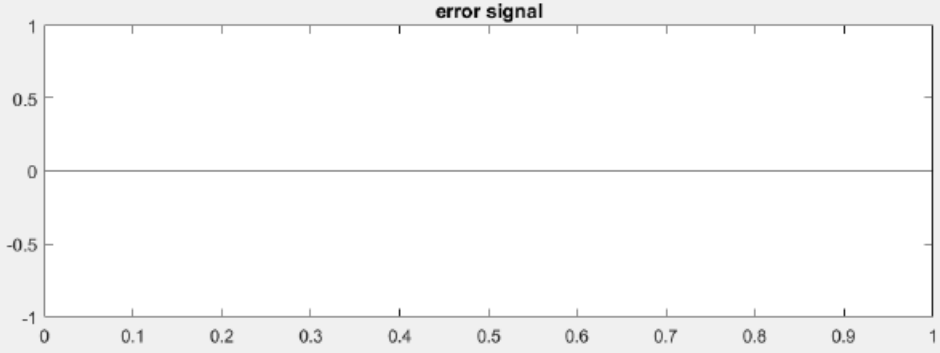
```
%synthesis of fourirer series
t = 0:0.01:1;
yt = zeros(size(t));
for k = -N:N
    yt = yt + (ak(1+k+N).*exp(1j*k*w0*t));
end
subplot(2,3,3);
plot(t,yt,'k');
title('xn(t)');

subplot(2,3,4);
x = sin(100*t);
plot(t,x,'k');
title('x(t)');

subplot(2,3,[5,6]);
plot(t,x-yt,'k');%errorsignal(e(t))
title('error signal');

```


magnitude
0.5
0.4
0.3
0.2
0.1
0
-10
-5
0
5
10


phase
2
1
0
-1
-2
-10
-5
0
5
10


xr
1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1

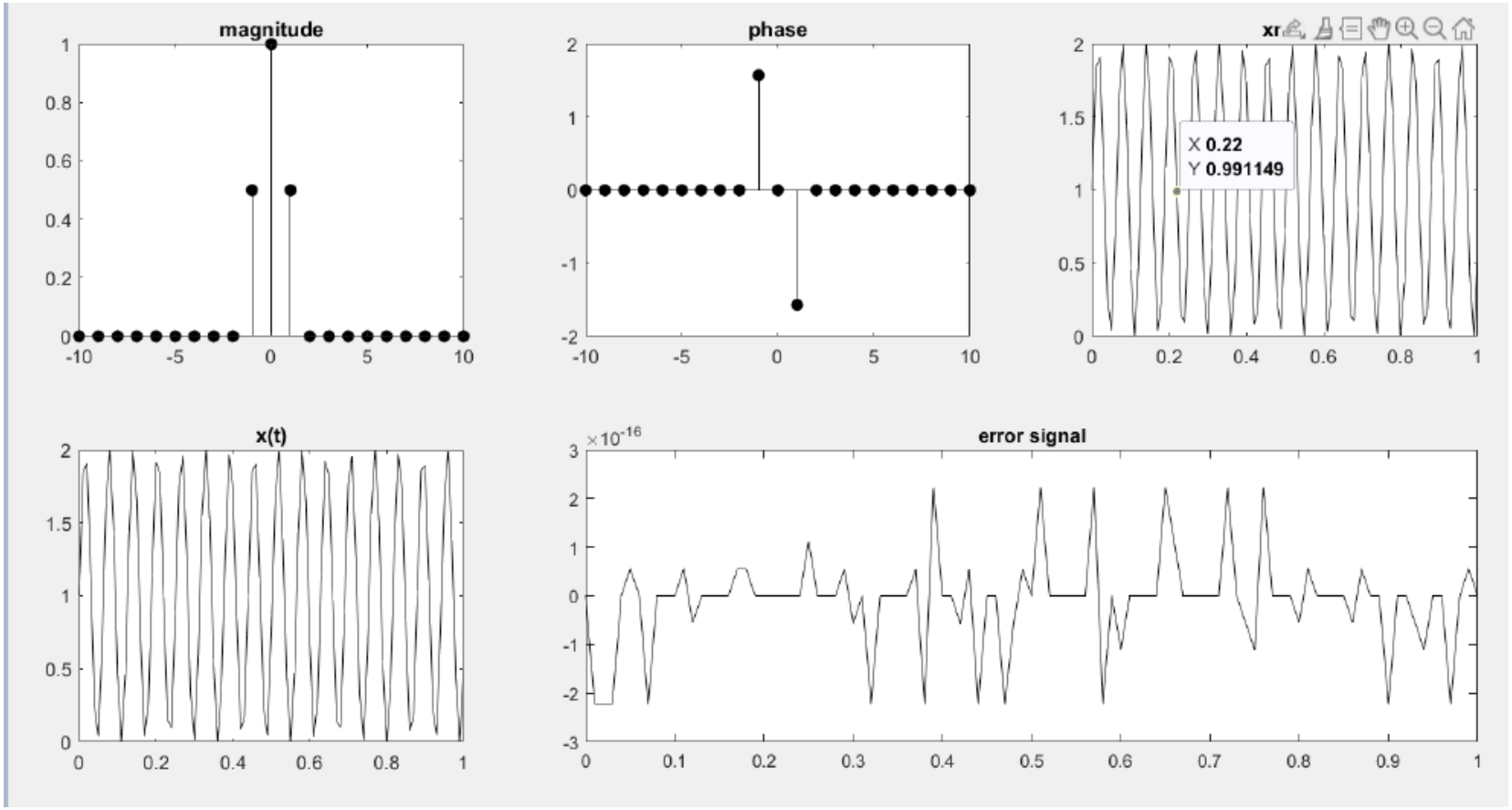

x(t)
1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1


error signal
1
0.5
0
-0.5
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Q1b

```
syms x;
syms t;
x = 1 + sin(100*t);
w0 = 100;
N = 10;
ak = zeros(1,2*N+1);
T = 2*pi/w0; %calculate the period and store in T
syms t;
%loop for calculating fourier coefficient
for k = -N:N
ak(1,1+k+N) = 1/T * int(x * exp(-1j*k*w0*t), 0, T) ;% ak is fourier coefficient
disp(ak(1+k+N));
end
k = -N:N;
mag = sqrt(real(ak).*real(ak) + imag(ak).* imag(ak));%magnitude of fourier coefficient
subplot(2,3,1);
stem(k,mag,'k','filled');
title('magnitude');

phase = angle(ak);
subplot(2,3,2);
stem(k,phase,'k','filled');
title('phase');
```

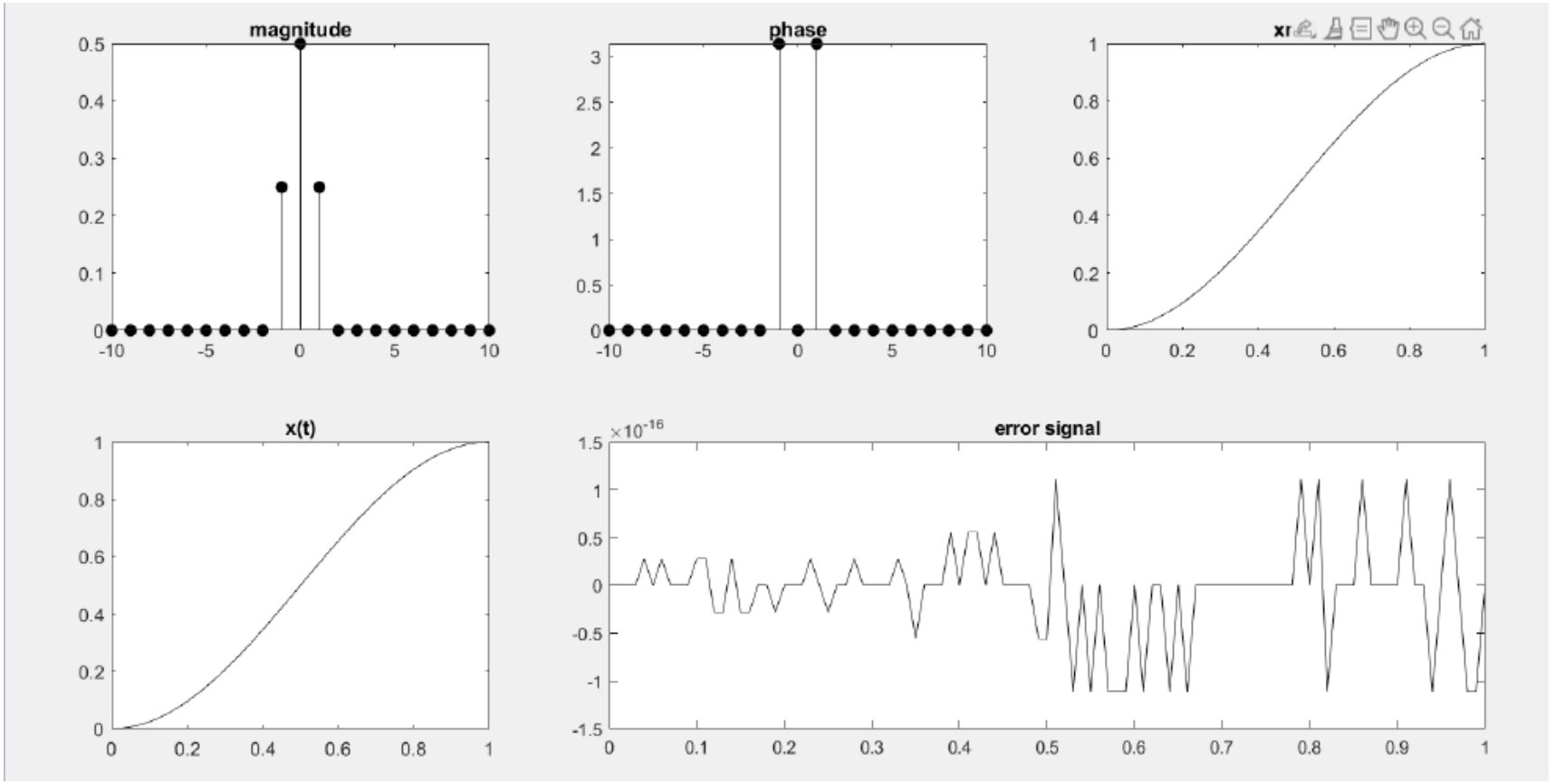
```
%synthesis of fourier series
t = 0:0.01:1;
yt = zeros(size(t));
for k = -N:N
    yt = yt + (ak(1+k+N).*exp(1j*k*w0*t));
end

subplot(2,3,3);
plot(t,yt,'k');
title('xn(t)');
subplot(2,3,4);

x = 1 + sin(100*t);
plot(t,x,'k');
title('x(t)');

subplot(2,3,[5,6]);
plot(t,x-yt,'k');%error signal(e(t))
title('error signal');

```


magnitude
phase
xr
X 0.22
Y 0.991149
x(t)
error signal
×10^-16

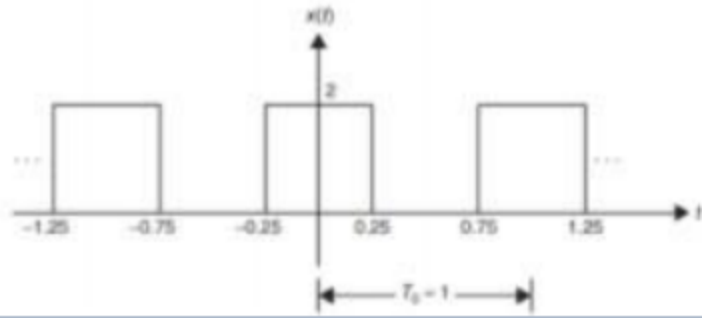
Q1c

```
syms x;
syms t;
w0 = pi;
x = (1/2)*(1-cos(w0 * t));
N = 10;
ak = zeros(1,2*N+1);
T = 2*pi/w0; %calculate the period and store in T
syms t;
%loop for calculating fourier coefficient
for k = -N:N;
ak(1,1+k+N) = 1/T * int(x * exp(-1j*k*w0*t), 0, T) ;%ak is fourier coefficient
disp(ak(1+k+N));
end
k = -N:N;
mag = sqrt(real(ak).*real(ak) + imag(ak).* imag(ak));%magnitude of fourier coefficient
subplot(2,3,1);
stem(k,mag,'k','filled');
title('magnitude');

phase = angle(ak);
subplot(2,3,2);
stem(k,phase,'k','filled');
title('phase');
```

```
t = 0:0.01:1;
yt = zeros(size(t));
for k = -N:N
    yt = yt + (ak(1+k+N).*exp(1j*k*w0*t));
end
subplot(2,3,3);
plot(t,yt,'k');
title('xn(t)');
subplot(2,3,4);
x = (1/2)*(1-cos(w0 * t));
plot(t,x,'k');
title('x(t)');
subplot(2,3,[5,6]);
plot(t,x-yt,'k');%error signal(e(t))
title('error signal');

```

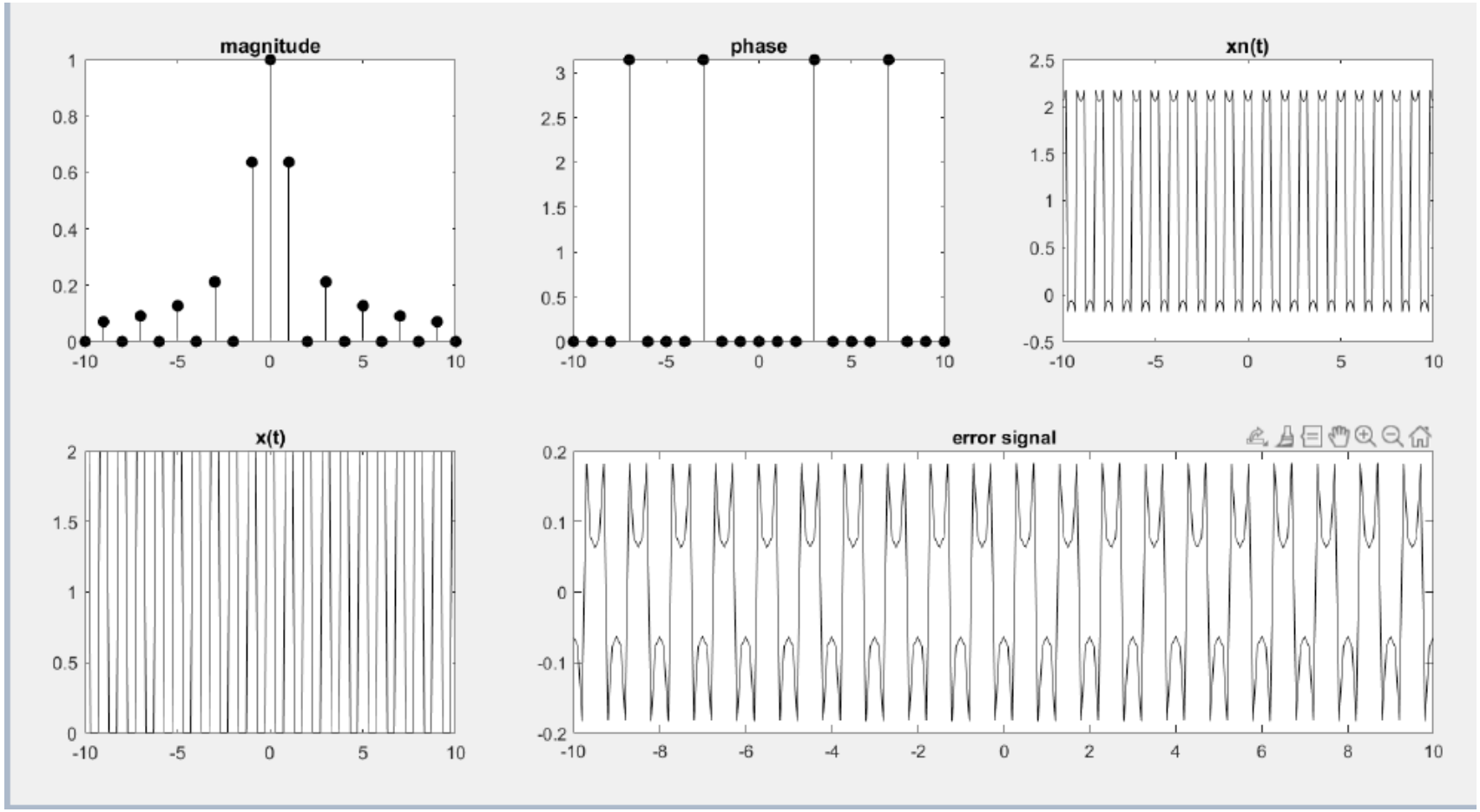

magnitude
phase
xr
x(t)
error signal
×10^-16

(d)




```
x = zeros(size(t));
for p = -10:10
    x = x + 2*(heaviside(t+p+0.25) - heaviside(t+p-0.25));
end
w0 = 2*pi;
N = 10;
ak = zeros(1,2*N+1);
T = 2*pi/w0;
syms t;
for k = -N:N;
ak(1,1+k+N) = 1/T * int(x * exp(-1j*k*w0*t), 0, T) ;%ak is fourier coefficient
disp(ak(1+k+N));
end
k = -N:N;
mag = sqrt(real(ak).*real(ak) + imag(ak).*imag(ak));
subplot(2,3,1);
stem(k,mag,'k','filled');
title('magnitude');

phase = angle(ak);
subplot(2,3,2);
stem(k,phase,'k','filled');
title('phase');
```

```
%synthesis of fourier series;
t = -10:0.1:10;
yt = zeros(size(t));
for k = -N:N
    yt = yt + (ak(1+k+N).*exp(1j*k*w0*t));
end
subplot(2,3,3);
plot(t,yt,'k');
title('xn(t)');
subplot(2,3,4);
t = -10:0.1:10;
x = zeros(size(t));
for p = -10:10
    x = x + 2*(heaviside(t+p+0.25) - heaviside(t+p-0.25));
end
plot(t,x,'k');
title('x(t)');
subplot(2,3,[5,6]);
plot(t,x-yt,'k');%error signal(e(t))
title('error signal');

```


magnitude
phase
xn(t)
x(t)
error signal

THANK YOU