Name: Diptangshu Dey

Roll No: 20CS8018

CSC-453 Signals Processing Laboratory Assignment

Lab Assignment 4:

1. Write a program to convolve two discrete time sequences. Plot all the sequences.

$$x(t) = \begin{cases} 3, & t = -2 \\ -1, & t = 1 \\ 2, & t = 3 \\ 0, & t \neq -2, 1, 3 \end{cases}$$

$$h(t) = \begin{cases} 1, & -4 \leq t < 4 \\ 0, & t > 4 \quad \text{or} \quad t < -4 \end{cases}$$

Theoretical Explanation :

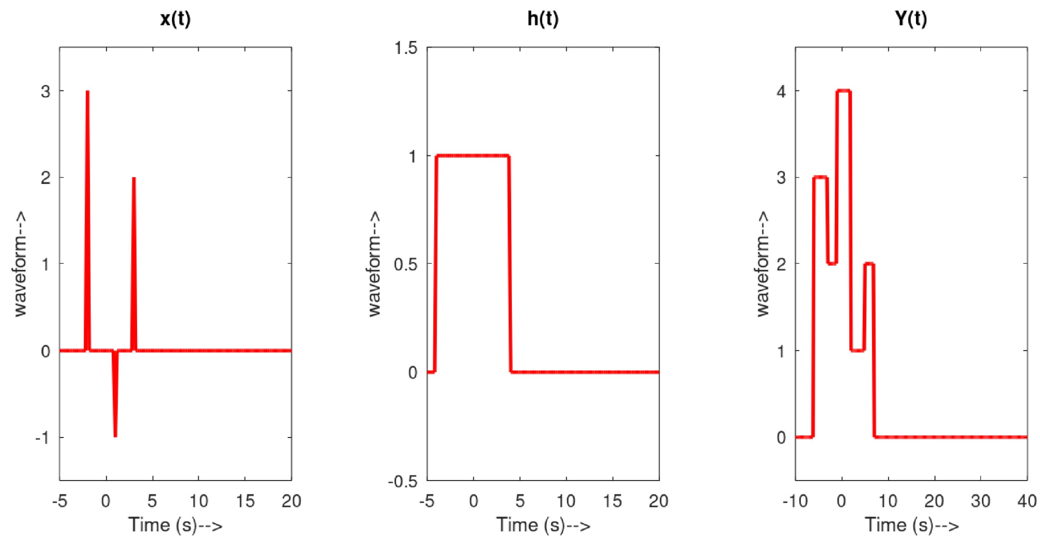
Here we have taken 2 discrete signal functions namely denoted by X(t) & Y(t) . Now based on the functions defined it has a vector of pulses formed namely like for instance in range of [-4,4] : X(t) : [0 , 0 , 3 , 0 , 0 , -1 , 0 , 2] H(t) : [1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1] Hence to convolute them is basically multiplying both the vectors and getting a result to generate some expression based on the multiplicative result obtained , which provides us with the convo(X,H).

Code:

```
t = -5:0.2:20
x = zeros(size(t));
h = zeros(size(t));
%% Generating x(t)
k = 0;
for i = t
if i == -2
x(k+1) = 3;
```

```
elseif i == 1
x(k+1) = -1;
elseif i == 3
x(k+1) = 2;
else
x(k+1) = 0;
end
k=k+1;
end
subplot(1,3,1)
plot(t,x,'r','LineWidth',2)
xlim([min(t) max(t)])
ylim([min(x)-0.5 max(x)+0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
title('x(t)')
%% Generating h(t)
k = 0;
for i = t
   if i >= -4 && i < 4
      h(k+1) = 1;
   else
      h(k+1) = 0;
   end
   k = k+1;
end
subplot(1,3,2)
plot(t,h,'r','LineWidth',2)
xlim([min(t) max(t)])
ylim([min(h)-0.5 max(h)+0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
title('h(t)')
%% Generating Y(t) for Y(t) = x(t)*h(t)
t1=-10:0.2:40;
n=length(x);
m=length(h);
H=[x,zeros(1,m)];
X=[h,zeros(1,n)];
for i = 1 : n + m - 1
y(i)=0;
for j=1:m
if(i-j+1 > 0)
```

```
y(i) = y(i) + X(j) * H(i-j+1);
end
end
end
subplot(1,3, 3)
plot(t1,y ,'r','LineWidth',2)
xlim([min(t1) max(t1)])
ylim([min(y) - 0.5 max(y) + 0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
title('Y(t)')
```



2. Write a program to convolve two discrete time sequences. Plot all the sequences. Hence verify the result as shown :

$$x(t) * h(t) = h(t) * x(t)$$

Theoretical Explanation :

Basically this question is same as the above Q1 just we have to experimentally prove the commutative property of the convolution as a whole so we shall be using the above code and the above data just plotting 2 convolution based curves one for H(t)*X(t) another for X(t)*H(t).

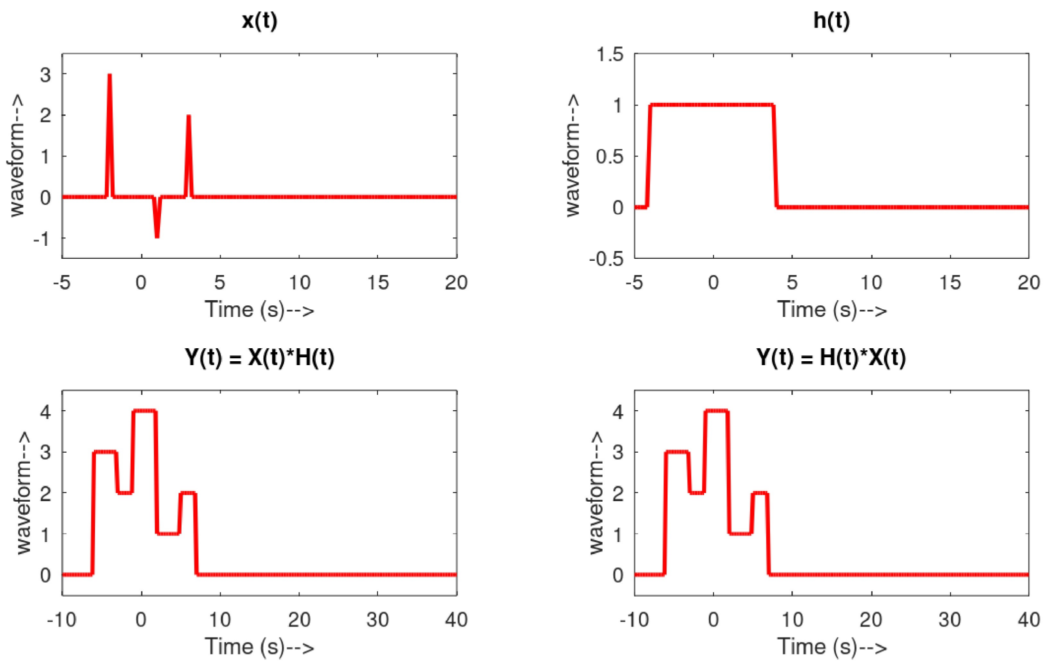
Code:

```
t = -5:0.2:20;
```

```
x = zeros(size(t));
h = zeros(size(t));
%% Generating x(t)
k = 0;
for i = t
if i == -2
x(k+1) = 3;
elseif i == 1
x(k+1) = -1;
elseif i == 3
x(k+1) = 2;
else
x(k+1) = 0;
end
k=k+1;
end
subplot(2, 2,1)
plot(t,x,'r','LineWidth',2)
xlim([min(t) max(t)])
ylim([min(x)-0.5 max(x)+0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
title('x(t)')
%% Generating h(t)
k = 0;
for i = t
if i >= -4 && i < 4
h(k+1) = 1;
else
h(k+1) = 0;
end
k = k+1;
end
subplot(2, 2,2)
plot(t,h,'r','LineWidth',2)
xlim([min(t) max(t)])
ylim([min(h)-0.5 max(h)+0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
title('h(t)')
%% Generating Y(t) for Y(t) = x(t)*h(t)
t1=-10:0.2:40;
n=length(x);
```

```
m=length(h);
H=[x,zeros(1,m)];
X=[h,zeros(1,n)];
for i = 1 : n + m - 1
y(i)=0;
for j=1:m
if(i-j+1 > 0)
y(i) = y(i) + X(j) * H(i-j+1);
end
end
end
subplot(2, 2, 3)
plot(t1,y ,'r','LineWidth',2)
xlim([min(t1) max(t1)])
ylim([min(y) - 0.5 max(y) + 0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
title('Y(t) = X(t)*H(t)')
%% Generating Y(t) for Y(t) = h(t)*x(t)
t1=-10:0.2:40;
n=length(x);
m=length(h);
H=[x,zeros(1,m)];
X=[h,zeros(1,n)];
for i = 1 : n + m - 1
y(i)=0;
for j=1:m
if(i-j+1 > 0)
y(i) = y(i) + X(j) * H(i-j+1);
end
end
end
subplot(2, 2, 4)
plot(t1,y ,'r','LineWidth',2)
xlim([min(t1) max(t1)])
ylim([min(y) - 0.5 max(y) + 0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
```

title('Y(t) = H(t)*X(t)')



From the graphs it is more or less clear that both the graphs are same hence the commutative law is proved true.

3. Find the convolution of two Non Causal Signals :

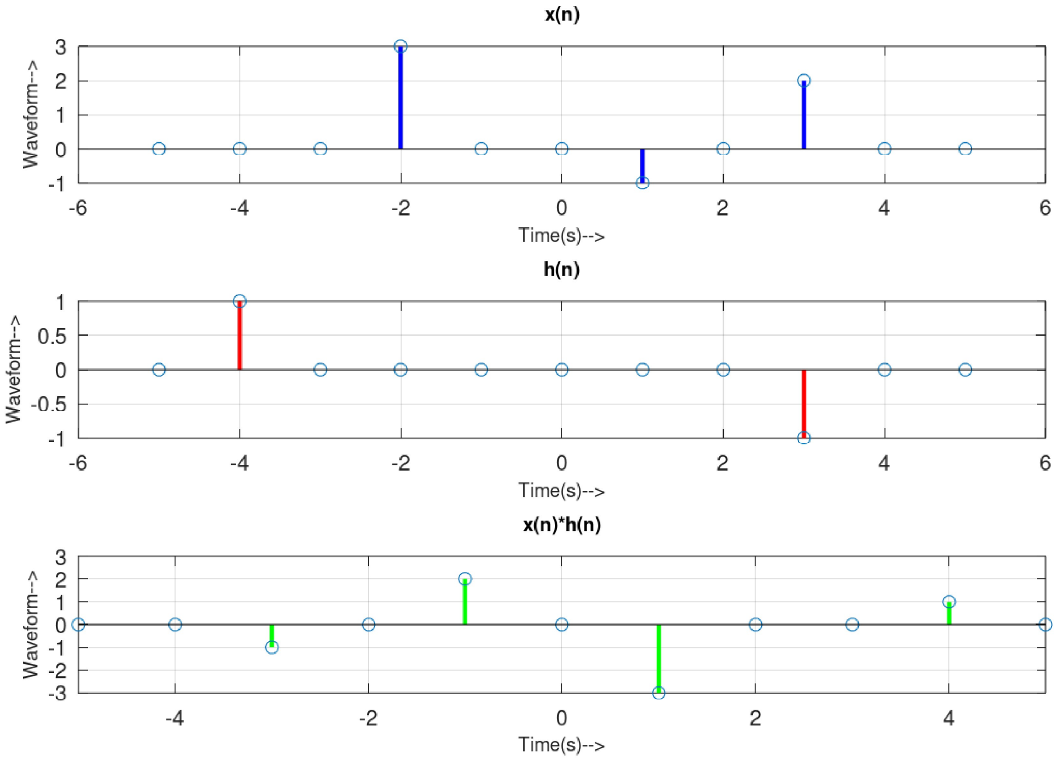
$$x(n) = 3\delta(n+2) - \delta(n-1) + 2\delta(n-3)$$
$$\text{and}$$
$$g(n) = u(n+4) - u(n-3)$$

Theoretical Explanation :

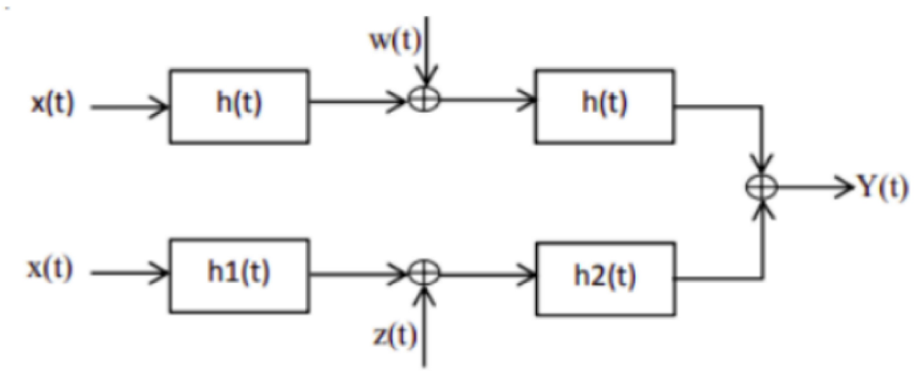
Non causal signals are those systems that depend upon the future values of the input at any instant of the time then the system is said to be non-causal system. For example here we have considered 2 functions x and g whose values are different and time variated based on all values of n. Now using the built-in convo(x,y,z) module we have convoluted the x and g graphs to get the resultant signal.

Code:

```
n0 = -2;
n1 = 1;
n2 = 3;
n3 = -4;
n = -5:5;
%% generating x(n)
xn = 3*((n-n0)==0)-((n-n1)==0)+2*((n-n2)==0);
%% generating h(n)
yn = ((n-n3)==0) - ((n-n2)==0);
%% convolution
h = conv(xn, yn, 'same');
%% plot
figure
subplot(3,1,1)
stem(n,xn,'linewidth',2,'color','b')
a= title('x(n)');
set(a,'fontsize',9);a= xlabel('Time(s)-->');
set(a,'fontsize',9);
a = ylabel('Waveform-->');
set(a,'fontsize',9);
grid
subplot(3,1,2)
stem(n,yn,'linewidth',2,'color','r')
a= title('h(n)');
set(a,'fontsize',9);
a= xlabel('Time(s)-->');
set(a,'fontsize',9);
a = ylabel('Waveform-->');
set(a,'fontsize',9);
grid
subplot(3,1,3)
stem(n,h,'linewidth',2,'color','g')
xlim([-5,5]);
ylim([-3,3]);
a= title('x(n)*h(n)');
set(a,'fontsize',9);
a= xlabel('Time(s)-->');
set(a,'fontsize',9);
a = ylabel('Waveform-->');
set(a,'fontsize',9);
grid
```



5. Write a program to find the response of the given system below. Plot all the sequences. Where each function serve as an inverter.



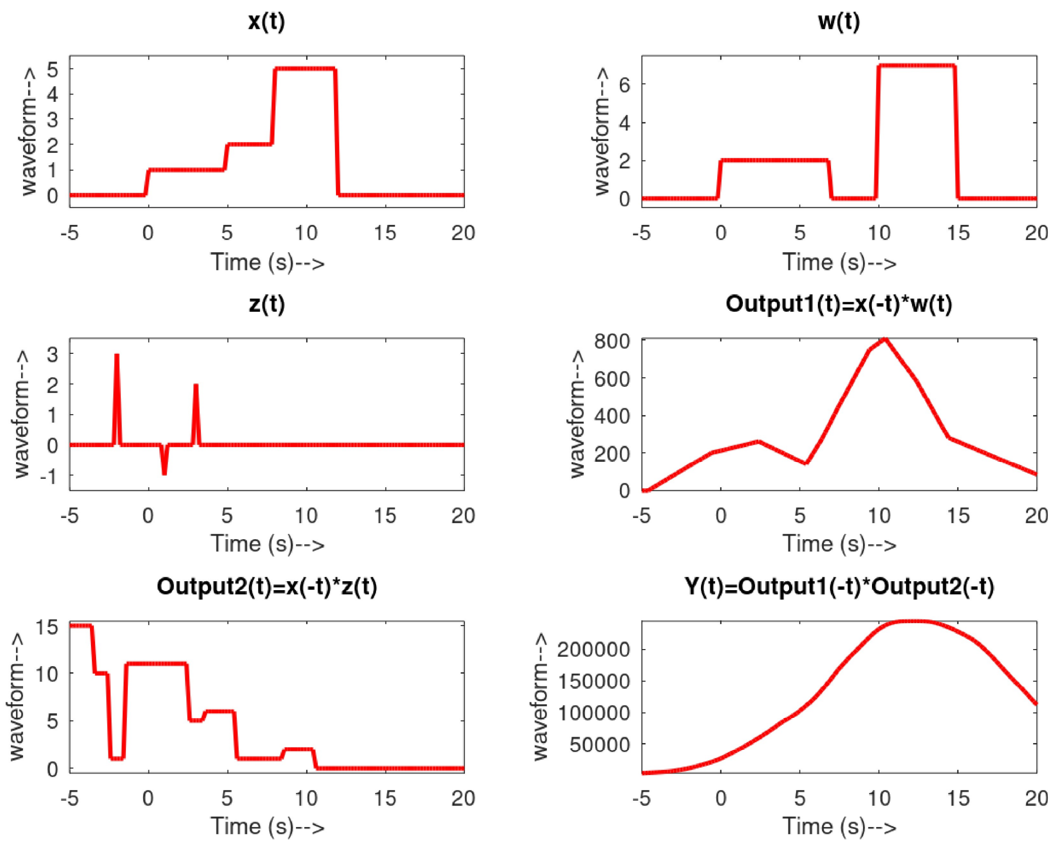
Now here we have a machine or a collection of systems working simultaneously each function h(t) or h1(t) will basically invert the result and the XOR sign signifies convolution of the incoming signals to form an output signal. We shall be using discrete values of x derived from Lab 3 assignment above.

Code:

```
t = -5:0.2:20;
x = zeros(size(t));
w = zeros(size(t));
z = zeros(size(t));
%% Generate x(t)
k = 0;
for i = t
if i<0
x(k+1) = 0;
elseif i >= 0 && i<5
x(k+1) = 1;
elseif i >= 5 && i<8
x(k+1) = 2;
elseif i >= 8 && i < 12
x(k+1) = 5;
else
x(k+1) = 0;
end
k = k+1;
end
subplot(3,2,1)
plot(t,x,'r','LineWidth',2)
xlim([min(t) max(t)])
ylim([min(x)-0.5 max(x)+0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
title('x(t)')
%% Generate w(t)
k=0;
for i = t
if i<0
w(k+1)=0;
elseif i>=0 && i<7
w(k+1)=2;
elseif i>=7 && i<10
w(k+1)=0;
elseif i>=10 && i<15
w(k+1)=7;
else
w(k+1)=0;
end
```

```
k=k+1;
end
subplot(3,2,2)
plot(t,w,'r','LineWidth',2)
xlim([min(t) max(t)])
ylim([min(w)-0.5 max(w)+0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
title('w(t)')
%% generate z(t)
k=0;
for i = t
if i==-2
z(k+1)=3;
elseif i==1
z(k+1)=-1;
elseif i==3
z(k+1)=2;
else
z(k+1)=0;
end
k=k+1;
end
subplot(3,2,3)
plot(t,z,'r','LineWidth',2)
xlim([min(t) max(t)])
ylim([min(z)-0.5 max(z)+0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
title('z(t)')
%% x(t)->x(-t)
c=fliplr(x);
%% x(-t)*w(t); x(-t)*z(t)
o1 = conv(c,w,'same');
o2 = conv(c,z,'same');
subplot(3,2,4)
plot(t,o1,'r','LineWidth',2)
xlim([min(t) max(t)])
ylim([min(o1)-0.5 max(o1)+0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
title('Output1(t)=x(-t)*w(t)')
subplot(3,2,5)
```

```
plot(t,o2,'r','LineWidth',2)
xlim([min(t) max(t)])
ylim([min(o2)-0.5 max(o2)+0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
title('Output2(t)=x(-t)*z(t)')
%% o1(t)->o1(-t); o2(t)->o2(-t)
o1flip = fliplr(o1);
o2flip = fliplr(o2);
%% Y(t)=o1flip(t)*o2flip(t)
Y = conv(o1flip,o2flip,'same');
%% plot Y(t)
subplot(3,2,6)
plot(t,Y,'r','LineWidth',2)
xlim([min(t) max(t)])
ylim([min(Y)-0.5 max(Y)+0.5])
xlabel('Time (s)-->')
ylabel('waveform-->')
```




```
title('Y(t)=Output1(-t)*Output2(-t)')
```