Name: Manaal Waseem **Reg. No:** FA18-BCE-074

Lab 7

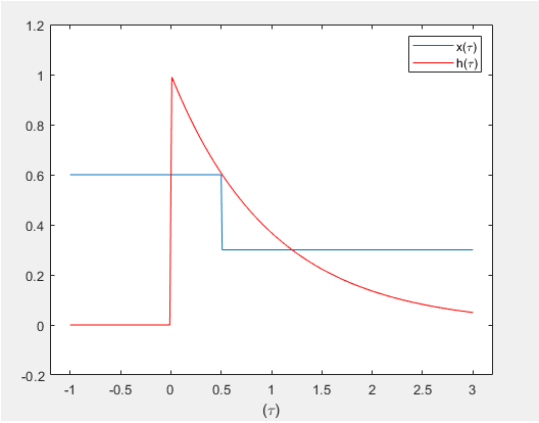
Task 1:

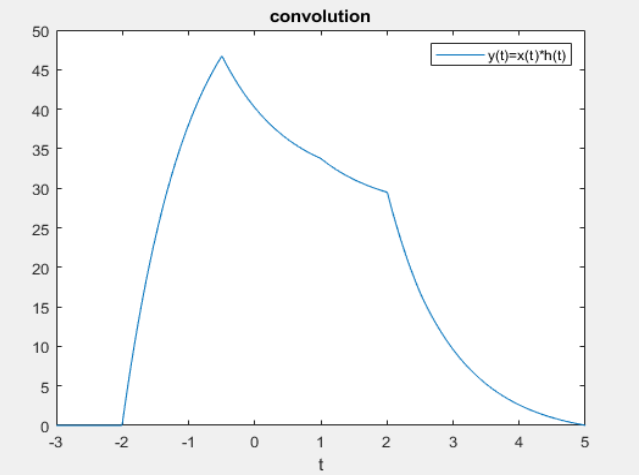
Code:

```
step = 0.01;
tx1 = -1:step:0.5; % j:i:k creates a regularly-spaced
% vector using i as the increment between elements j & k
tx2 = 0.5+step:step:3;
t = [tx1 tx2];

x1 = ones(size(tx1)).*0.6;
x2 = ones(size(tx2)).*0.3;
x = [x1 x2];

h = exp(-t).*heaviside(t); % a discontinuous function that
% returns 0 for x < 0, 1/2 for x = 0, and 1 for x > 0
plot(t,x,t,h,'r')
xlim([-1.2 3.2])
ylim([ -0.2 1.2])
legend('x(\tau)','h(\tau)')
xlabel('(\tau)')
```

```
y = conv(x,h); % convolution
yt = -3:0.01:5; % j:i:k creates a regularly-spaced
% vector using i as the increment between elements j & k
plot(yt,y)
legend('y(t)=x(t)*h(t)')
xlabel('t')
title convolution
```

Graph:





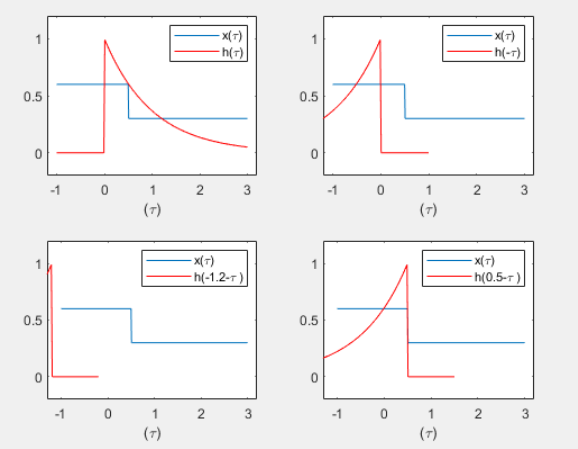
Second Procedure:

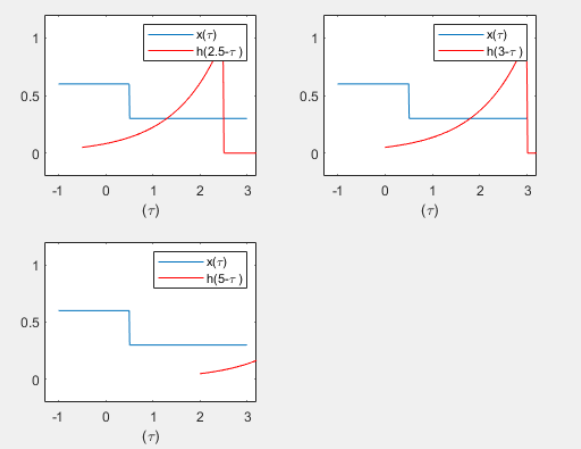
Code:

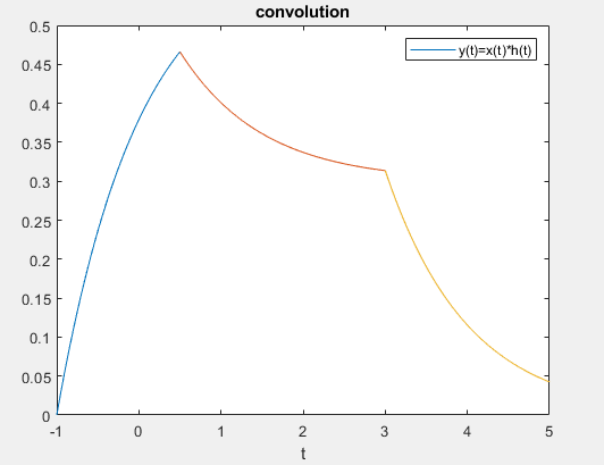
```
step = 0.01;
tx1 = -1:step:0.5;
tx2 = 0.5+step:step:3;
t = [tx1 tx2];

x1 = ones(size(tx1)).*0.6; % ones(n) returns an n -by- n matrix of 1's
x2 = ones(size(tx2)).*0.3;
x = [x1 x2];

h = exp(-t).*heaviside(t);
subplot(2,2,1)            %Step1
plot(t,x,t,h,'r')
xlim([-1.2 3.2])
ylim([ -0.2 1.2])
legend('x(\tau)','h(\tau)')
xlabel('(\tau)')

subplot(2,2,2)            %Step2
plot(t,x,-t,h,'r')        %flipped
xlim([-1.2 3.2])
ylim([ -0.2 1.2])
legend('x(\tau)','h(-\tau)')
xlabel('(\tau)')

subplot(2,2,3)            %Step3
p = -1.2;                 %No Overlap
plot(t,x,-t+p,h,'r')
xlim([-1.3 3.2])
ylim([ -0.2 1.2])
legend('x(\tau)','h(-1.2-\tau )')
xlabel('(\tau)')

subplot(2,2,4)            %Step4
p = 0.5;                  %Partial Overlap
plot(t,x,-t+p,h,'r')
xlim([-1.3 3.2])
ylim([ -0.2 1.2])
legend('x(\tau)','h(0.5-\tau )')
xlabel('(\tau)')


subplot(2,2,1)            %Step5
p = 2.5;                  %Partial Overlap
plot(t,x,-t+p,h,'r')
xlim([-1.3 3.2])
ylim([ -0.2 1.2])
legend('x(\tau)','h(2.5-\tau )')
xlabel('(\tau)')

subplot(2,2,2)            %Step6
p = 3;                    %Exiting Overlap
plot(t,x,-t+p,h,'r')
xlim([-1.3 3.2])
ylim([ -0.2 1.2])
legend('x(\tau)','h(3-\tau )')
xlabel('(\tau)')

subplot(2,2,3)            %Step7
p = 5;                    %No Overlap
```

```
plot(t,x,-t+p,h,'r')
xlim([-1.3 3.2])
ylim([ -0.2 1.2])
legend('x(\tau)','h(5-\tau )')
xlabel('(\tau)')
syms t tau
f = 0.6.*exp(-(t-tau));
y1 = int(f,tau,-1,t);

syms t tau % Separate different variables by spaces.
% syms clears all assumptions from the variables.
x = 0.6.*exp(-(t-tau));
z = 0.3.*exp(-(t-tau));
y2 = int(x,tau,-1,0.5); % int(expr,var,a,b) computes
% the definite integral of expr with respect to the
% symbolic scalar variable var from a to b.
y3 = int(z,tau,0.5,t);
y4 = y2+y3;

syms t tau
p = 0.6.*exp(-(t-tau)); % returns the exponential
k = 0.3.*exp(-(t-tau));
y5 = int(p,tau,-1,0.5);
y6 = int(k,tau,0.5,3);
y7 = y5+y6;

tt1 = -1:0.01:0.5; %  creates a regularly-spaced vector
tt2 = 0.5:0.01:3;
tt3 = 3:0.01:5;

yt1 = 3/5 - (3*exp(- tt1 - 1))/5;
yt2 = (3*exp(- tt2 - 1)*(exp(3/2) - 1))/5 - (3*exp(-tt2)*exp(1/2))/10 + 3/10;
yt3 = (3*exp(- tt3 - 1)*(exp(3/2) - 1))/5 + (3*exp(-tt3)*exp(1/2)*(exp(5/2) - 1))/10;
plot(tt1,yt1,tt2,yt2,tt3,yt3);
legend('y(t)=x(t)*h(t)')
xlabel('t')
title convolution
```

Graph:





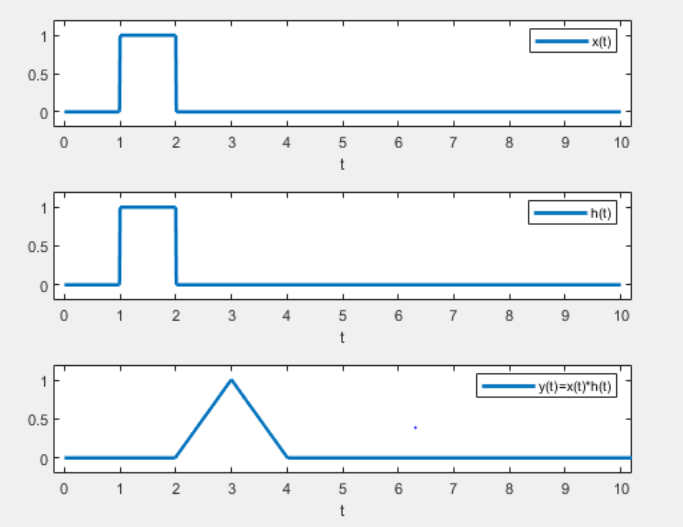


Task 2:

Code:

```
step = 0.01;
t1 = 0:step:1-step; % j:i:k creates a regularly-spaced
% vector using i as the increment between elements j & k
t2 = 1:step:2;
t3 = 2+step:step:10;
t = [t1 t2 t3];

x1 = zeros(size(t1)); % zeros(n) returns an n-by-n matrix of zeros
x2 = ones(size(t2)); % ones(n) returns an n -by- n matrix of 1's
x3 = zeros(size(t3));
x = [x1 x2 x3];

h = x;
subplot(3,1,1)
plot(t,x,'LineWidth',2)
legend('x(t)')
xlabel('t')
xlim([-0.2 10.2])
ylim([-0.2 1.2])

subplot(3,1,2)
plot(t,h,'LineWidth',2);
legend('h(t)')
xlabel('t')
xlim([-0.2 10.2])
ylim([-0.2 1.2])

subplot(3,1,3)
y = conv(x,h).*step;
yt = 0:step:20;
plot(yt,y,'LineWidth',2);
legend('y(t)=x(t)*h(t)')
xlabel('t')
xlim([-0.2 10.2])
ylim([-0.2 1.2])
```

Graph:



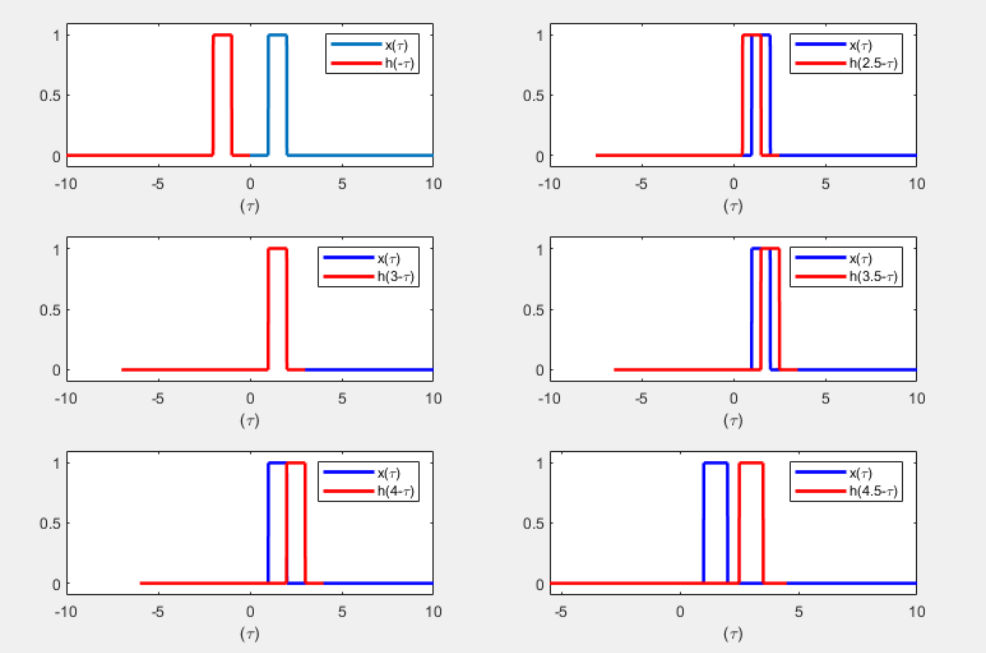
Second Procedure:

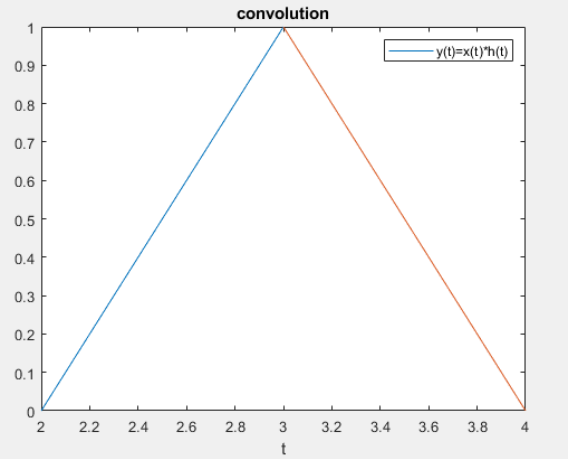
Code:

```
step = 0.01;
t1 = 0:step:1-step;
t2 = 1:step:2;
t3 = 2+step:step:10;
```

```
t = [t1 t2 t3];

x1 = zeros(size(t1)); % zeros(n) returns an n-by-n matrix of zeros
x2 = ones(size(t2)); % ones(n) returns an n -by- n matrix of 1's
x3 = zeros(size(t3));
x = [x1 x2 x3];

h = x;

subplot(3,2,1)
plot(t,x,-t,h,'r','LineWidth',2)    %flipped
legend('x(\tau)','h(-\tau)')        %No Overlap
xlabel('(\tau)')
ylim([-0.1 1.1])

p = 2.5;
subplot(3,2,2)
plot(t,x,'b',-t+p,h,'r','LineWidth',2)  %Partial Overlap
legend('x(\tau)','h(2.5-\tau)')
xlabel('(\tau)')
ylim([-0.1 1.1])

p = 3;
subplot(3,2,3)
plot(t,x,'b',-t+p,h,'r','LineWidth',2)  %Full Overlap
legend('x(\tau)','h(3-\tau)')
xlabel('(\tau)')
ylim([-0.1 1.1])

p = 3.5;
subplot(3,2,4)
plot(t,x,'b',-t+p,h,'r','LineWidth',2)  %Partial Overlap
legend('x(\tau)','h(3.5-\tau)')
xlabel('(\tau)')
ylim([-0.1 1.1])

p = 4;
subplot(3,2,5)
plot(t,x,'b',-t+p,h,'r','LineWidth',2)  %Exiting Overlap
legend('x(\tau)','h(4-\tau)')
xlabel('(\tau)')
ylim([-0.1 1.1])

p = 4.5;
subplot(3,2,6)

plot(t,x,'b',-t+p,h,'r','LineWidth',2)  %No Overlap
legend('x(\tau)','h(4.5-\tau)')
xlabel('(\tau)')
ylim([-0.1 1.1])

syms t x % Separate different variables by spaces.
% syms clears all assumptions from the variables.
f = 1;
p1 = int(f,x,1,t);
p2 = int(f,x,1,2);
p3 = p1-p2;

syms t x
p4 = int(f,x,t,2);  % int(expr,var,a,b) computes
% the definite integral of expr with respect to the
% symbolic scalar variable var from a to b.
p5 = int(f,x,3,2);
p6 = int(f,x,2,1);
p7 = p4 - p5 - p6;

t1 = 2:0.01:3; %  creates a regularly-spaced vector
t2 = 3:0.01:4;

y1 = t1-2;
y2 = -t2+4;
plot(t1,y1,t2,y2);
legend('y(t)=x(t)*h(t)')
xlabel('t')
title convolution
```

Graph:





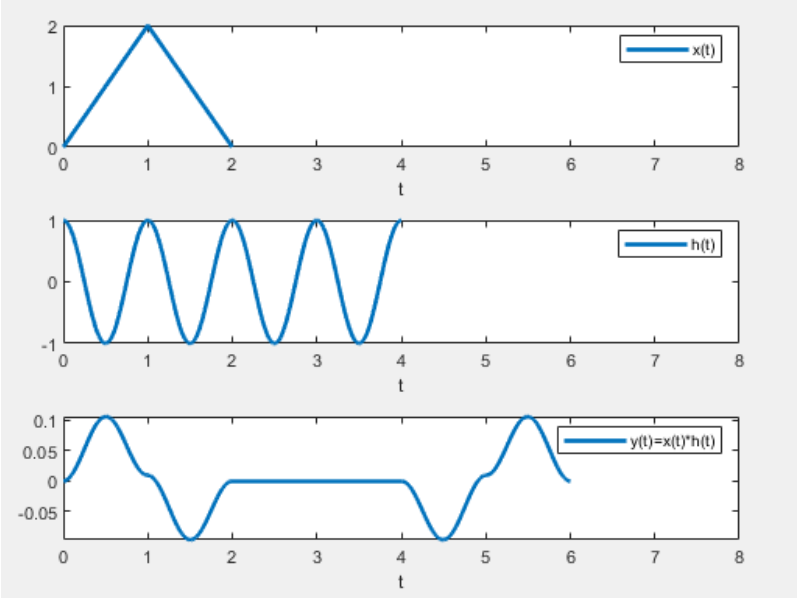
Task 3:

Code:

```
step = 0.01;
t1 = 0:step:1;
t2 = 1+step:step:2;
t = [t1 t2];

x1 = 2.*t1;
x2 = -2.*t2 + 4;
x = [x1 x2];

subplot(3,1,1)
plot(t,x,'LineWidth',2)
legend('x(t)')
xlabel('t')
xlim([0 8])

subplot(3,1,2)
th = 0:step:4;
h1 = ones(size(th));
h2 = cos(2.*pi.*th);
```

```
h = h2.*h1;
plot(th,h,'LineWidth',2)
legend('h(t)')
xlabel('t')
xlim([0 8])

subplot(3,1,3)
y = conv(x,h).*step;
yt = 0:step:6;
subplot(3,1,3)
plot(yt,y,'LineWidth',2)
legend('y(t)=x(t)*h(t)')
xlabel('t')
xlim([0 8])
```

Graph:



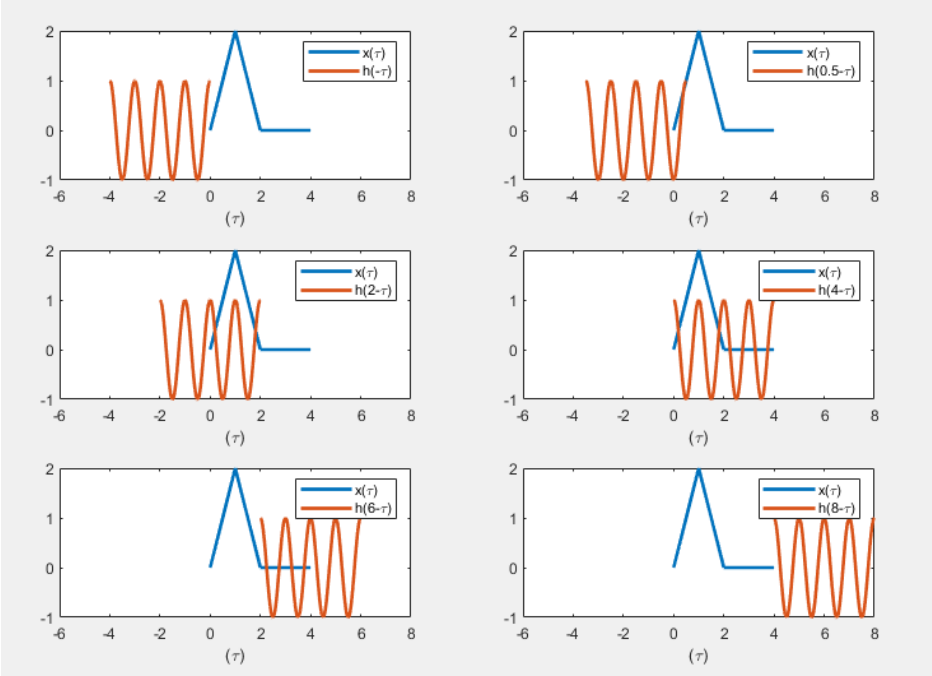
Second Procedure:

Code:

```
step = 0.01;
t1 = 0:step:1;  % j:i:k creates a regularly-spaced vector
% using i as the increment between elements j and k
t2 = 1+step:step:2;
t3 = 2+step:step:4;
t = [t1 t2 t3];

x1 = 2.*t1;
x2 = -2.*t2 + 4; % Element-wise multiplication
x3 = zeros(size(t3)); % zeros(n) returns an n-by-n matrix of zeros
x = [x1 x2 x3];

th = 0:step:4;
h1 = ones(size(th));
h2 = cos(2.*pi.*th);
h = h2.*h1;

subplot(3,2,1)
plot(t,x,-t,h,'LineWidth',2)
legend('x(\tau)','h(-\tau)')
xlabel('(\tau)')
xlim([-6 8])

p = 0.5;
subplot(3,2,2)
plot(t,x,-t+p,h,'LineWidth',2)
legend('x(\tau)','h(0.5-\tau)')
xlabel('(\tau)')
xlim([-6 8])

p = 2;
```

```
subplot(3,2,3)
plot(t,x,-t+p,h,'LineWidth',2)
legend('x(\tau)','h(2-\tau)')
xlabel('(\tau)')
xlim([-6 8])

p = 4;
subplot(3,2,4)
plot(t,x,-t+p,h,'LineWidth',2)
legend('x(\tau)','h(4-\tau)')
xlabel('(\tau)')
xlim([-6 8])

p = 6;
subplot(3,2,5)
plot(t,x,-t+p,h,'LineWidth',2)
legend('x(\tau)','h(6-\tau)')

xlabel('(\tau)')
xlim([-6 8])

p = 8;
subplot(3,2,6)
plot(t,x,-t+p,h,'LineWidth',2)
legend('x(\tau)','h(8-\tau)')
xlabel('(\tau)')
xlim([-6 8])
```

Graph:



Critical Analysis:

In this lab I learnt:

- The term ‘response’ denotes the output of a system.
- the term, ‘impulse’ denotes that the input signal applied to the system is the unit impulse or Dirac Delta function.
- There are certain steps for doing convolution. The signal is to be calculated as it sweeps over the input signal, making overlaps.
- The conv command can also be used to calculate the convolution of signals.

The mathematical expression of the convolution relationship of continuous time signal is $y(t) = x(t) * h(t)$.

THE END