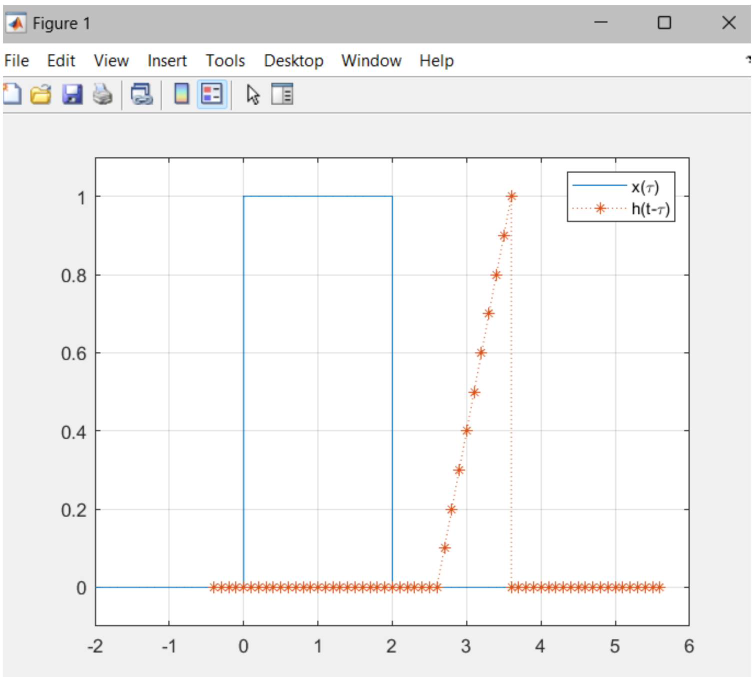
Lab 7: Analysis of Continuous Time LTI Systems using Convolution Integral

Name	**DANISH ZULFIQAR**
Registration Number	**FA21-BEE-042**
Class/section	**4-A**
Instructor's Name	**DR. BILAL TARIQ**

Example 1:

```
>> tx1=-2:0.1:0;
tx2=0:0.1:2;
tx3=2:0.1:4;
tx=[tx1 tx2 tx3];
x1=zeros(size(tx1));
x2=ones(size(tx2));
x3=zeros(size(tx3));
x=[x1 x2 x3];
>> th1=-2:0.1:0;
th2=0:0.1:1;
th3=1:0.1:4;
th=[th1 th2 th3];
h1=zeros(size(th1));
h2=1-th2;
h3=zeros(size(th3));
h=[h1 h2 h3];
plot(tx,x,th,h,':*')
ylim([-0.1 1.1])
legend('x(\tau)','h(\tau)')
grid
>> plot(tx,x,-th,h,':*')
ylim([-0.1 1.1])
legend('x(\tau)','h(-\tau)')
grid
>> t=-2;
plot(tx,x,-th+t,h,':*')
ylim([-0.1 1.1])
legend('x(\tau)','h(t-\tau)')
grid
>> t=0.5;
plot(tx,x,-th+t,h,':*'),grid on
```

```
>> t=1.6;
plot(tx,x,-th+t,h,':*'),grid on
ylim([-0.1 1.1])
legend('x(\tau)','h(t-\tau)')
% the following code
% produce the shadowed
% area plot
T=1;
r=t-T:0.1:t;
a=1/T*r+1-t/T;
hold on; area(r,a);
hold off;
>> t=2.4;
plot(tx,x,-th+t,h,':*'),grid on
ylim([-0.1 1.1])
legend('x(\tau)','h(t-\tau)')
% the following code
% produce the shadowed
% area plot
T=1;
r=t-T:0.1:2;
a=1/T*r+1-t/T;
hold on; area(r,a);
hold off;
>> t=3.6;
plot(tx,x,-th+t,h,':*'),grid on
ylim([-0.1 1.1])
legend('x(\tau)','h(t-\tau)')
```



Task 1:

```
>> step = 0.01;
tx1 = -1:step:0.5;
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);
plot(t,x,t,h)
y = conv(x,h);
yt = -3:0.01:5;plot(yt,y)
title convolution
%Analytical Method:
step = 0.01;
tx1 = -1:step:0.5;
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);
subplot(2,2,1) %Step1
plot(t,x,t,h)
subplot(2,2,2) %Step2
plot(t,x,-t,h)
subplot(2,2,3) %Step3
p = -1.2;
plot(t,x,-t+p,h)
subplot(2,2,4) %Step4
p = 0.5;
plot(t,x,-t+p,h)
```

```
plot(t,x,-t+p,h)
subplot(2,2,4) %Step4
p = 0.5;
plot(t,x,-t+p,h)
subplot(2,2,1) %Step5
p = 2.5;
plot(t,x,-t+p,h)
subplot(2,2,2) %Step6
p = 3;
plot(t,x,-t+p,h)
```

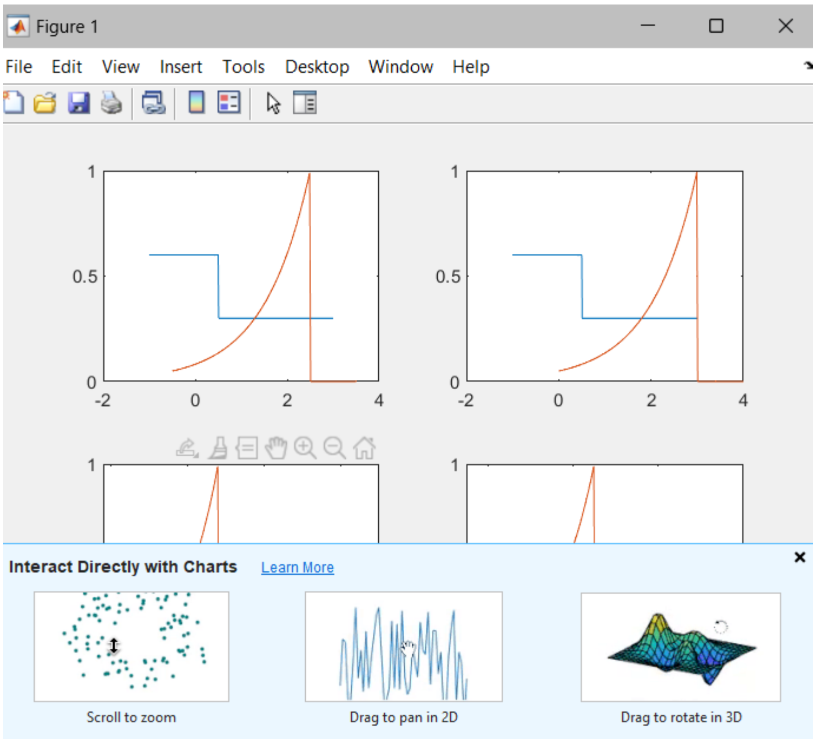
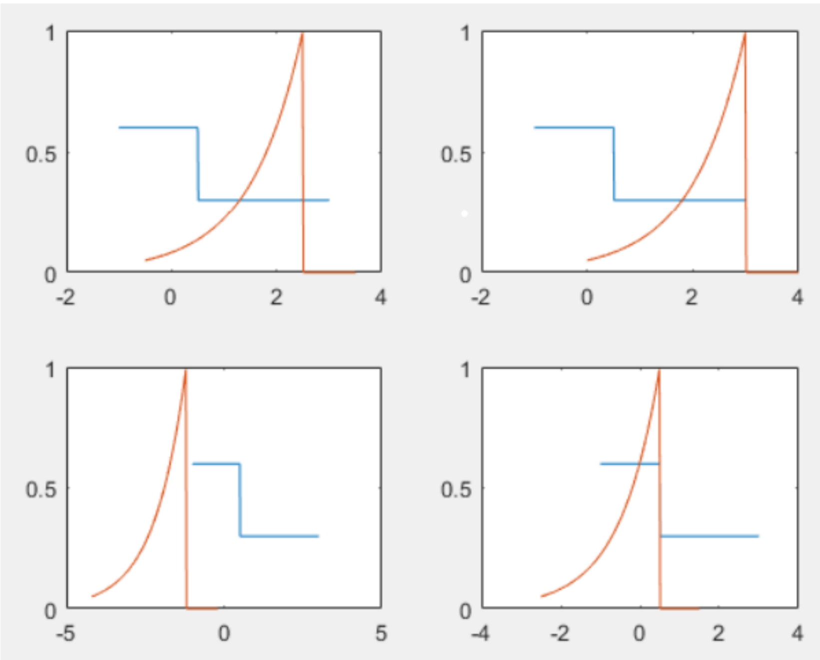
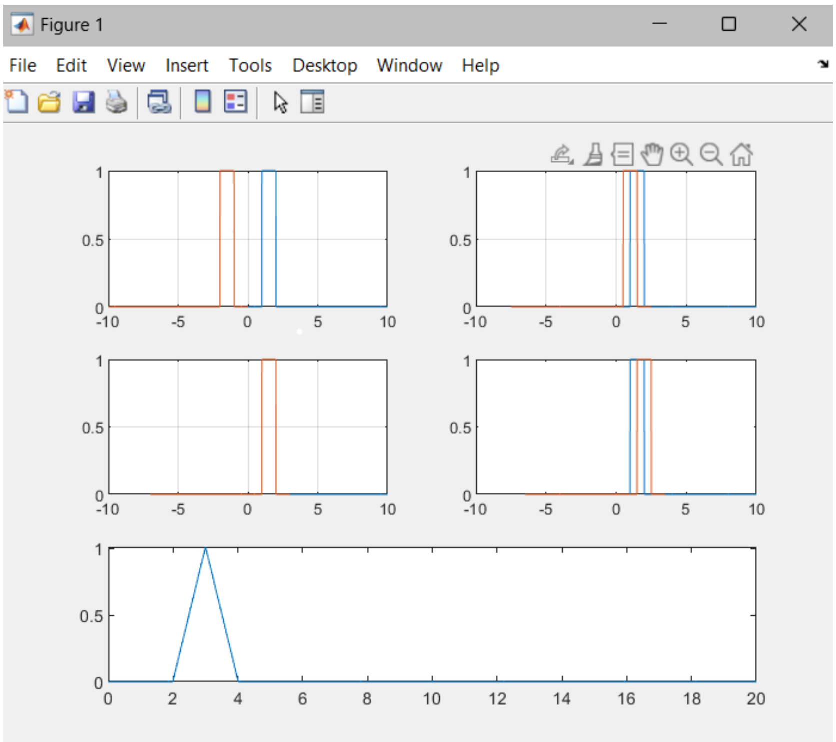


Figure 1
File Edit View Insert Tools Desktop Window Help
Interact Directly with Charts
Learn More
Scroll to zoom
Drag to pan in 2D
Drag to rotate in 3D



Task 2:

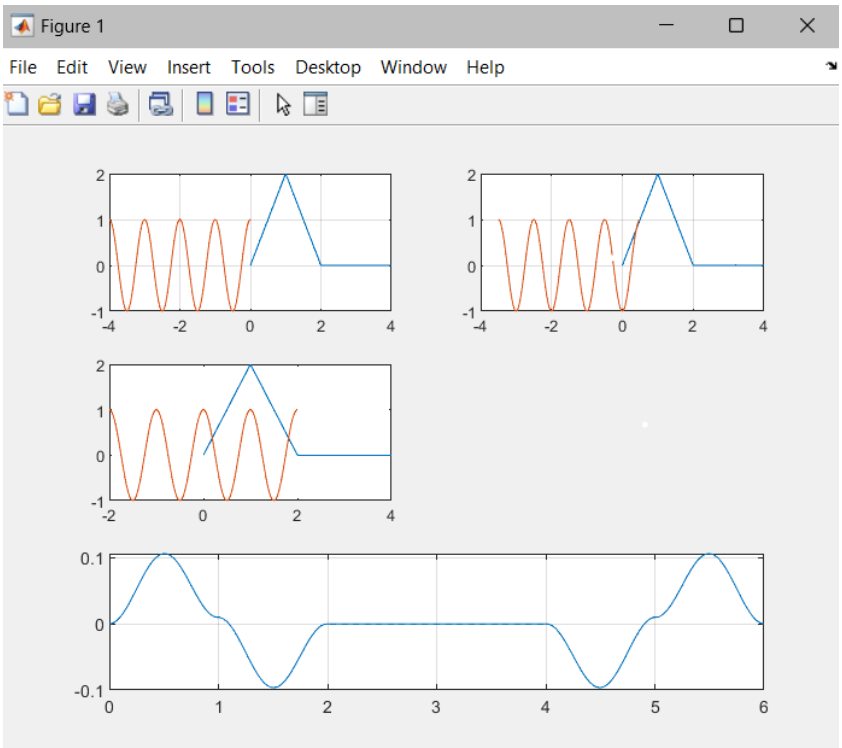
```
>> step = 0.01;
t1 = 0:step:1-step;
t2 = 1:step:2;
t3 = 2+step:step:10;
t = [t1 t2 t3];
x1 = zeros(size(t1));
x2 = ones(size(t2));
x3 = zeros(size(t3));
x = [x1 x2 x3];h = x;
subplot(3,1,1)
plot(t,x)
subplot(3,1,2)
plot(t,h);
subplot(3,1,3)
y = conv(x,h).*step;
yt = 0:step:20;
plot(yt,y);
%Analytical Method:
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));
x2 = ones(size(t2));
x3 = zeros(size(t3));
x = [x1 x2 x3];
h = x;
subplot(3,2,1)
plot(t,x,-t,h) %flipped
grid on
```



Task 3:

```
>> 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)
grid on
subplot(3,1,2)
th = 0:step:4;
h1 = ones(size(th));
h2 = cos(2.*pi.*th);
h = h2.*h1;
plot(th,h)
grid on
subplot(3,1,3);
>> y = conv(x,h).*step;
yt = 0:step:6;
subplot(3,1,3)
plot(yt,y)
grid on
%Analytical Method:
step = 0.01;
t1 = 0:step:1;
t2 = 1+step:step:2;
t3 = 2+step:step:4;
t = [t1 t2 t3];
x1 = 2.*t1;
x2 = -2.*t2 + 4;
```

```
step = 0.01;
t1 = 0:step:1;
t2 = 1+step:step:2;
t3 = 2+step:step:4;
t = [t1 t2 t3];
x1 = 2.*t1;
x2 = -2.*t2 + 4;
x3 = zeros(size(t3));
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)
grid on
p = 0.5;
subplot(3,2,2)
plot(t,x,-t+p,h)
grid on
p = 2;
>> subplot(3,2,3)
plot(t,x,-t+p,h)
```



Critical Analysis/Conclusion:

In this lab I learnt about Continuous Time Convolution.

The impulse response, which is commonly expressed by h[n], is perfectly described by an LTI Continuous-time Convolution. When the unit impulse sequence (or Dirac Delta function) is applied as input, the impulse response of an LTI Continuous time system is the output. The MATLAB command (In this case, h=deconv(y, x)) conv is used to compute the convolution between two discrete time signals.
If the input and output signals are known, the deconvolution technique can also be used to determine the impulse response of a system.