Computer Assignment #3

ECE 2101 Electrical Circuit Analysis 2: Section 3

Instructor: Adrian Gonzalez

Kyler Martinez

12/11/2020

Objective

The objective of the third computer assignment is to use MATLAB to find the transfer function of a circuit and then use the transfer function to find the responses of various inputs. We are also to be able to use MATLAB to find the convolution of two signals and finally find the Laplace transform of a signal and use that in circuit analysis. We also use MATLAB to automate some processes to reduce the work and time needed to solve some problems.

<u>Results</u>

Problem 1:

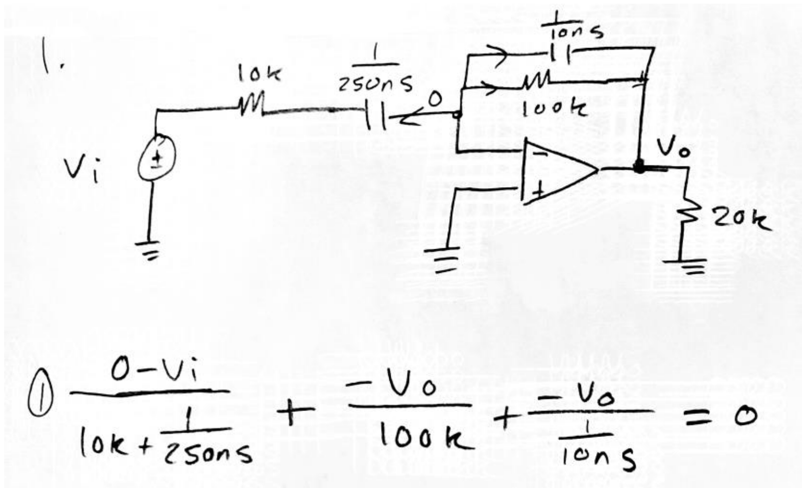

Figure 1: Circuit Diagram & Node Equation For Problem 1

The transfer function of the circuit is $H(s) = \frac{-10000s}{s^2+1400s+400000}$

The output due to a unit step response is $y_0(t) = \left(\frac{50}{3}e^{-1000t} - \frac{50}{3}e^{-400t}\right)u(t)$

The output due to a unit ramp response is $y_0(t) = \left(\frac{1}{24}e^{-400t} - \frac{1}{60}e^{-1000t} - \frac{1}{40}\right)u(t)$

The steady state expression for V_o for various values of ω if V_i =2cos(ωt)

ω = 100 rad/s	V_o = 4.8266cos(100t-109.75^0)
ω = 200 rad/s	V_o = 8.7706cos(200t-127.87^0)
ω = 400 rad/s	V_o = 13.131cos(400t-156.8^0)
ω = 800 rad/s	V_o = 13.969cos(800t+167.91^0)
ω = 1600 rad/s	V_o = 10.293cos(1600t+136.04^0)
ω = 3200 rad/s	V_o = 5.9194cos(3200t+114.48^0)
ω = 6400 rad/s	V_o = 3.0815cos(6400t+102.46^0)
ω = 12800 rad/s	V_o = 1.557cos(12800t+96.257^0)
ω = 25600 rad/s	V_o = .78056cos(25600t+93.132^0)
ω = 51200 rad/s	V_o = .39054cos(51200t+91.567^0)
ω = 102400 rad/s	V_o = 4.8266cos(102400t+90.783^0)

Problem 2:

a. y = (heaviside(t - 2)*(t - 2)^2)/2 - heaviside(t - 1)*(t - 1)^2 + t^2/2 for t>=0

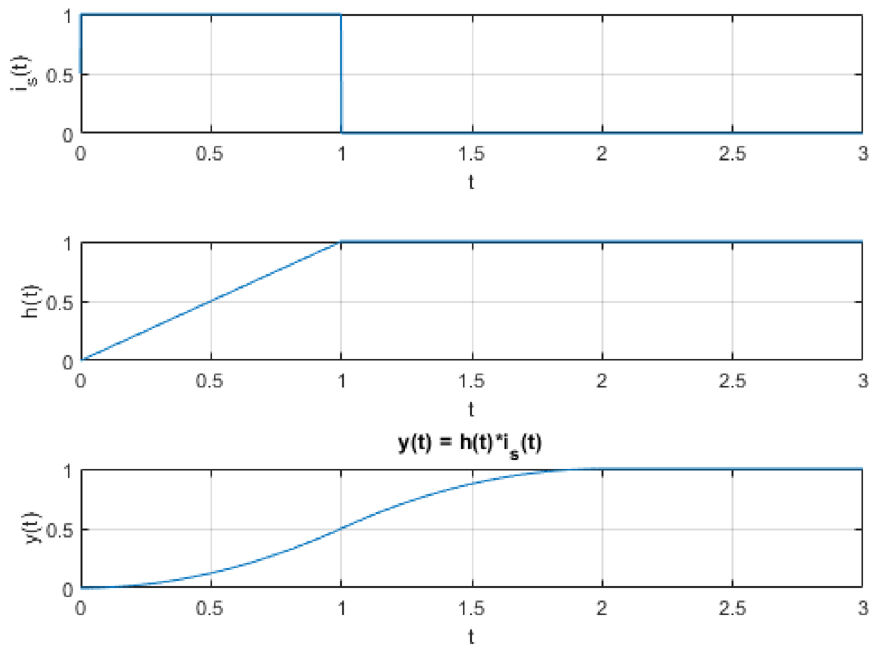

Figure 2: Graphical Convolution With MATLAB

b. y = 2 - 2*exp(-t) for t>=0

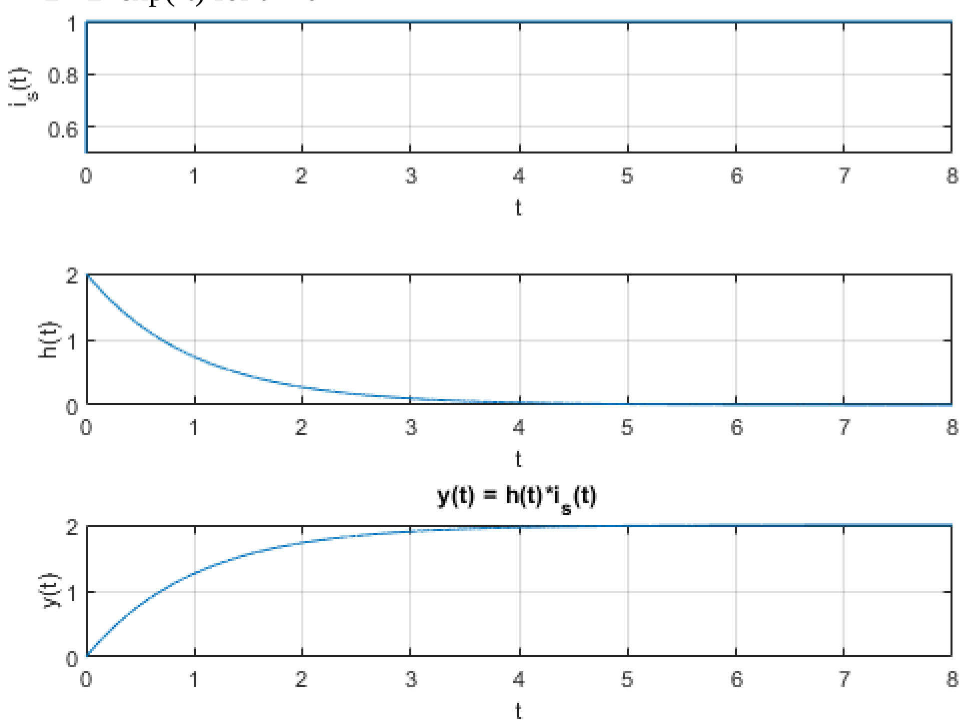

Figure 3: Graphical Convolution With MATLAB

Problem 3:

a. y = t - 3*heaviside(t - 1)*(t - 1) + 3*heaviside(t - 2)*(t - 2)-heaviside(t - 3)*(t - 3) for t>0

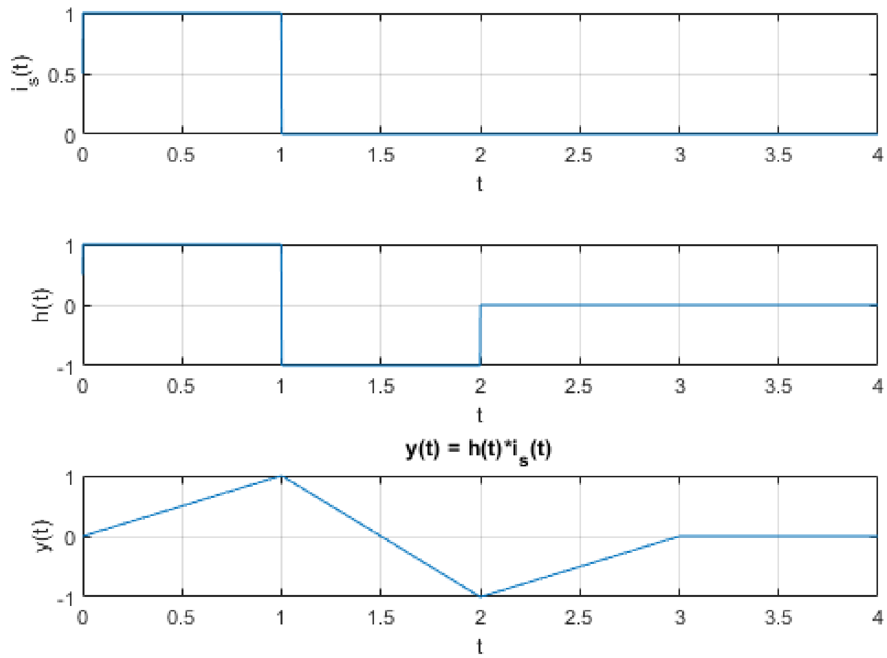

Figure 4: Graphical Convolution With MATLAB

b. y = heaviside(t - 6)*(t - 6) - heaviside(t - 3)*(t - 3) + (heaviside(t - 2)*(t - 2)^2)/2 - (heaviside(t - 3)*(t - 3)^2)/2 - (heaviside(t - 5)*(t - 5)^2)/2 + (heaviside(t - 6)*(t - 6)^2)/2 for t>0

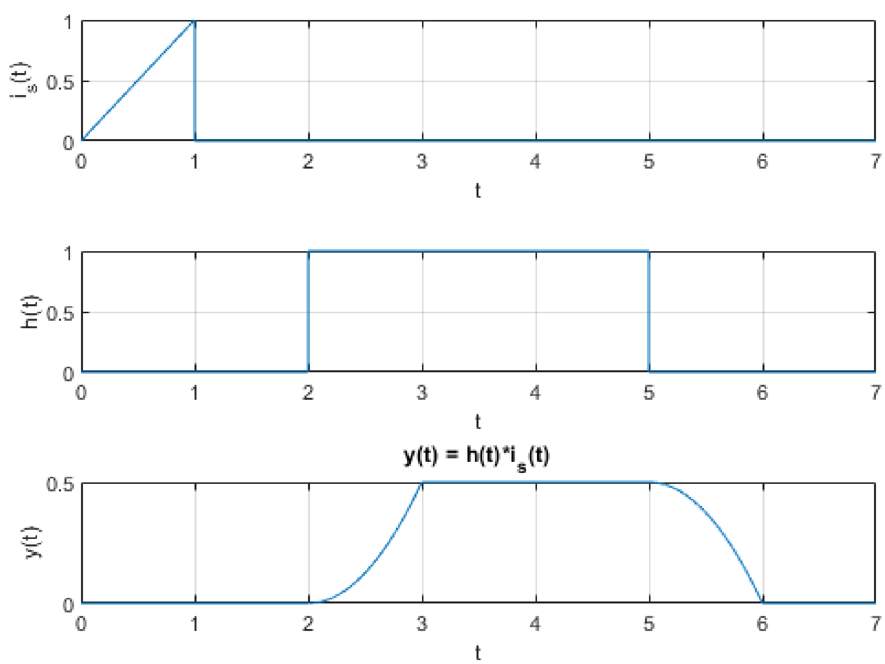

Figure 5: Graphical Convolution With MATLAB

Problem 4:

y = 8*t - 16*heaviside(t - 2)*(t - 2) + 8*heaviside(t - 6)*(t - 6) + 8*heaviside(t - 8)*(t - 8) - 16*heaviside(t - 12)*(t - 12) + 8*heaviside(t - 14)*(t - 14) for t>0

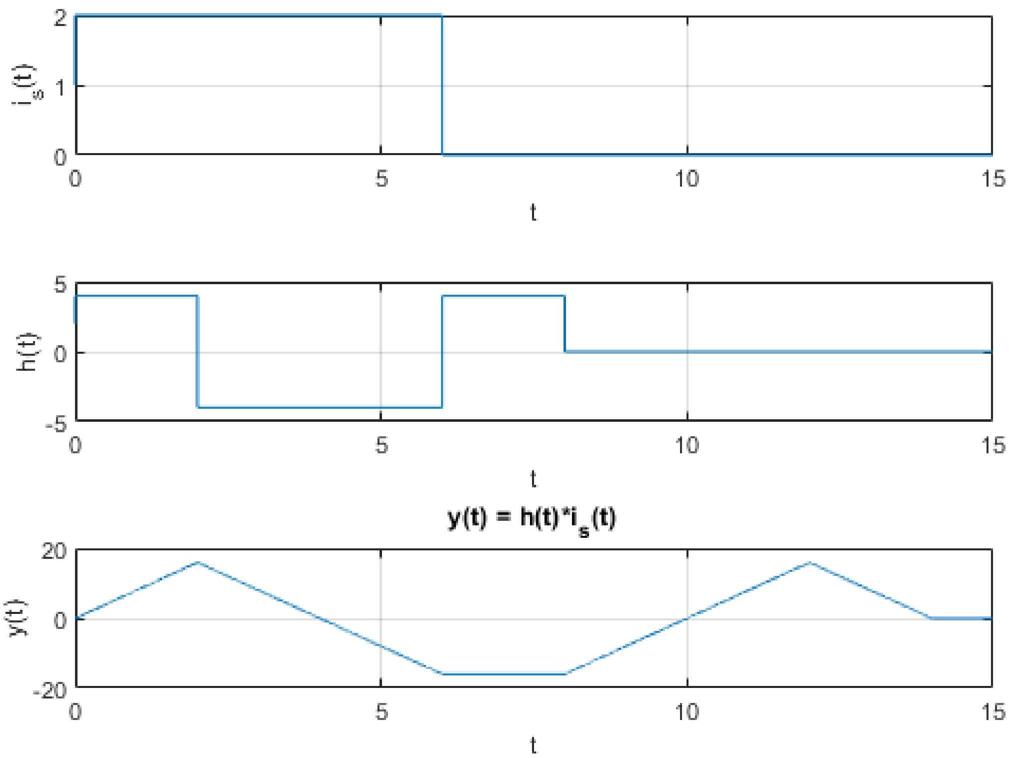

Figure 6: Graphical Convolution With MATLAB

Problem 5:

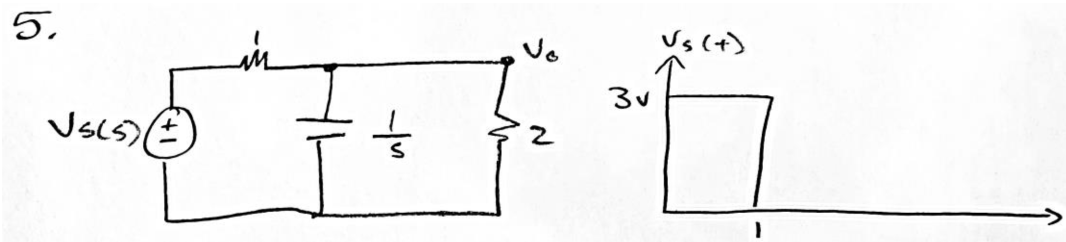

Figure 7: S- Domain Circuit & Time Domain Signal

a. $v_s(s) = \frac{3}{s} - \frac{3\,e^{-s}}{s}$

b. $v_o(t) = -2\,e^{-\frac{3t}{2}} + \text{heaviside}(t-1)\,\left(\frac{e^{\frac{3}{2}-\frac{3t}{2}}\,2}{3} - \frac{2}{3}\right)\,3 + 2$ for t>0, or the function multiplied by the Heaviside function.

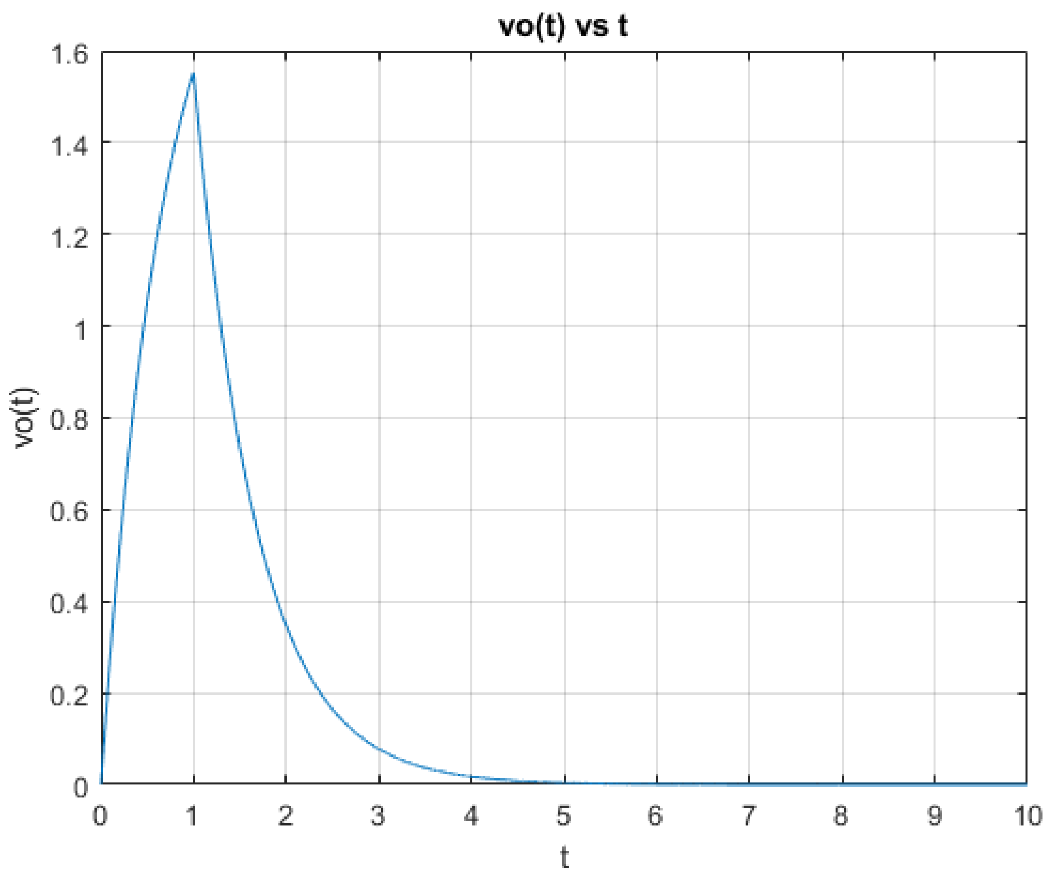

Figure 8: MATLAB plot of vo(t)

Analysis

We can see that the unit step response for the circuit in question 1 decays to 0V as time increase and that the unit ramp response decays to a steady state value. For the AC input, the magnitude of the response as the frequency increases, reaches a maximum near 800 rads/s and then decreases as the frequency increases. The magnitude response resembles that of a low pass filter and the phase increases as frequency increases. For the circuit in problem 5, we can see that the circuit is similar to a complex RC circuit and thus would expect for the voltage to attempt to reach a steady state value while the input signal was at 3V, but before the response could reach that value, the signal drops to 0V and we then see the response decay.

When convoluting two-unit step signals, we see the convolution function consisting of ramp functions and step, or only ramp if you consider the function to be 0*t + b. When a unit step and unit ramp function are convoluted, the response is consisted of quadratic segments. Finally, the convolution of an exponential and a unit step produces a exponential response, but is rising as opposed to decaying. These results occur because the integral that cause a unit step and ramp to turn into a quadratic and a decaying exponential to turn into a rising one, with a vertical shift.

Conclusion

I was able to find the transfer function and different responses using MATLAB, the Laplace transform of a signal, and finally the convolution of two signals. MATLAB made tasks such as finding the steady state response to an AC input easier due to tis functionality as a programming langue so for loops expedited the process. The graphical process for convolution works well in MATLAB, however if the functions are not easy to recognize then you may not be able to derive the piecewise function without doing the convolution integral or Laplace multiplication. Finally, MATLAB did pose some issues with formatting since MATLAB would sometimes attempt to display the result to many decimal places or not reducing coefficients. Overall, MATLAB made circuit analysis easier and expedited the process when compared to doing it by hand.

Appendix

```
%MATLAB CODE Question One
%I removed the outputs to save space since they appear elsewhere.
syms vi vo s t %defining symbolic variables

%Defining Metric Vars
k = 10^3;
n = 10^-9;

%Defining Impedances
ZR1 = 10*k;
ZR2 = 100*k;
ZR3 = 20*k; %Never used; it is here to show its existance
ZC1 = 1/(s*250*n);
ZC2 = 1/(s*10*n);

eqn = -vi/(ZR1+ZC1) - vo/(ZR2) - vo/(ZC2) == 0; %Node Equation From Circuit

Vo(s)= solve(eqn,vo); %Solving Node Equation For Vo
H(s) = Vo/vi; %Finding Transfer Funciton
Y_Unit(s) = H(s)/s; %Unit Step Response In S Domain

y_unit(t) = (ilaplace(Y_Unit(s)) * heaviside(t)); %Unit Step Response In Time
Domain
Y_Ramp(s) = H(s)/s^2; %Unit Ramp Response In S Domain

y_ramp(t) = ilaplace (Y_Ramp(s)) * heaviside(t); %Unit Ramp Response In Time
Domian
w = 100;
Vi(t) = 2*cos(w*t);
while w<= 102400
    Vo_exact(t) = 2*abs(H(j*w))*cos(w*t+angle(H(j*w))*180/pi);%Exact Expression
    Vo = vpa(Vo_exact,5); %Expression Set To Lower SigFigs
    w = w*2; %Moving To Next Omega
end
```

```
%MATLAB Code Problem 2:
clc; clear; close all;
%Note: I have to comment out a plot before I run it, but it works nonetheless.

%Definitions:
tmin=0; tmax=3; stepSize=1e-3;
syms t

%Create time vector:
time=tmin:stepSize:tmax;

%Symbolic function:
is_sym=heaviside(t)-heaviside(t-1)
h_sym= t-(t-1)*heaviside(t-1)
is=double(subs(is_sym,t,time));
h=double(subs(h_sym,t,time));

%Convolution: y(t) = h(t)*i_s(t)
yfull=conv(is,h,'full')*stepSize;
y=yfull(1:length(time)); %Take only half the samples

%Plot results:
subplot(3,1,1); plot(time,is); %i_s(t)
xlabel('t'); ylabel('i_s(t)'); grid on;
subplot(3,1,2); plot(time,h); %h(t)
xlabel('t'); ylabel('h(t)'); grid on;
subplot(3,1,3); plot(time,y); %y(t)
xlabel('t'); ylabel('y(t)'); title('y(t) = h(t)*i_s(t)'); grid on;

Y = laplace(is_sym) * laplace(h_sym); %Laplace Method
y = ilaplace(Y)

%Symbolic function:
is_sym=heaviside(t);
h_sym=2*exp(-t);
is=double(subs(is_sym,t,time));
h=double(subs(h_sym,t,time));

%Convolution: y(t) = h(t)*i_s(t)
yfull=conv(is,h,'full')*stepSize;
y=yfull(1:length(time)); %Take only half the samples

%Plot results:
subplot(3,1,1); plot(time,is); %i_s(t)
xlabel('t'); ylabel('i_s(t)'); grid on;
subplot(3,1,2); plot(time,h); %h(t)
```

```
xlabel('t'); ylabel('h(t)'); grid on;
subplot(3,1,3); plot(time,y); %y(t)
xlabel('t'); ylabel('y(t)'); title('y(t) = h(t)*i_s(t)'); grid on;
Y = laplace(is_sym) * laplace(h_sym); %Laplace Method
y = ilaplace(Y);
```

```
%Note: MATLAB Code For Problem 3
clc; clear; close all;
%Note: I have to comment out a plot before I run it, but it works nonetheless.

%Definitions:
tmin=0; tmax=4; stepSize=1e-3;
syms t

%Create time vector:
time=tmin:stepSize:tmax;

%Symbolic function:
is_sym=heaviside(t)-heaviside(t-1);
h_sym= heaviside(t)-heaviside(t-1) - (heaviside(t-1)-heaviside(t-2));
is=double(subs(is_sym,t,time));
h=double(subs(h_sym,t,time));

%Convolution: y(t) = h(t)*i_s(t)
yfull=conv(is,h,'full')*stepSize;
y=yfull(1:length(time)); %Take only half the samples

%Plot results:
subplot(3,1,1); plot(time,is); %i_s(t)
xlabel('t'); ylabel('i_s(t)'); grid on;
subplot(3,1,2); plot(time,h); %h(t)
xlabel('t'); ylabel('h(t)'); grid on;
subplot(3,1,3); plot(time,y); %y(t)
xlabel('t'); ylabel('y(t)'); title('y(t) = h(t)*i_s(t)'); grid on;

Y = laplace(is_sym) * laplace(h_sym); %Laplace Method
y = ilaplace(Y);

%Definitions: It is necessary for a different time defintion.
tmin=0; tmax=7; stepSize=1e-3;
syms t

%Create time vector:
time=tmin:stepSize:tmax;
```

```
%Symbolic function:
is_sym=t*heaviside(t)-t*heaviside(t-1);
h_sym=heaviside(t-2) - heaviside(t-5);
is=double(subs(is_sym,t,time));
h=double(subs(h_sym,t,time));

%Convolution: y(t) = h(t)*i_s(t)
yfull=conv(is,h,'full')*stepSize;
y=yfull(1:length(time)); %Take only half the samples

%Plot results:
subplot(3,1,1); plot(time,is); %i_s(t)
xlabel('t'); ylabel('i_s(t)'); grid on;
subplot(3,1,2); plot(time,h); %h(t)
xlabel('t'); ylabel('h(t)'); grid on;
subplot(3,1,3); plot(time,y); %y(t)
xlabel('t'); ylabel('y(t)'); title('y(t) = h(t)*i_s(t)'); grid on;

Y = laplace(is_sym) * laplace(h_sym); %Laplace Method
y = ilaplace(Y);
```

```
%CA Question 4
clc; clear; close all;

%Definitions:
tmin=0; tmax=15; stepSize=1e-3;
syms t

%Create time vector:
time=tmin:stepSize:tmax;

%Symbolic function:
is_sym=2*heaviside(t)-2*heaviside(t-6);
h_sym= 4*heaviside(t)-8*heaviside(t-2) + 8*heaviside(t-6) - 4*heaviside(t-8);
is=double(subs(is_sym,t,time));
h=double(subs(h_sym,t,time));

%Convolution: y(t) = h(t)*i_s(t)
yfull=conv(is,h,'full')*stepSize;
y=yfull(1:length(time)); %Take only half the samples

%Plot results:
subplot(3,1,1); plot(time,is); %i_s(t)
```

```
xlabel('t'); ylabel('i_s(t)'); grid on;
subplot(3,1,2); plot(time,h); %h(t)
xlabel('t'); ylabel('h(t)'); grid on;
subplot(3,1,3); plot(time,y); %y(t)
xlabel('t'); ylabel('y(t)'); title('y(t) = h(t)*i_s(t)'); grid on;

Y = laplace(is_sym) * laplace(h_sym); %Laplace Method
y = ilaplace(Y);
```

```
%CA Question 5

syms s t
vs = 3*heaviside(t) -3*heaviside(t-1);
Vs = laplace (vs);
Z1 = 1;
Z2 = 1/s;
Z3 = 2;
Z23 = (1/Z2 + 1/Z3)^-1;
Vo = collect(Vs* Z23/(Z1 + Z23));
vo(t) = collect(ilaplace(Vs* Z23/(Z1 + Z23)));
w = linspace(0,10,10000);
plot(w,vo(w))
xlabel('t'); ylabel('vo(t)'); title('vo(t) vs t'); grid on;
```