RYERSON UNIVERSITY

Faculty of Engineering, Architecture and Science

Department of Electrical and Computer Engineering

Course Number	**ELE 532**
Course Title	**Signals and Systems 1**
Sem/Year	**Fall 2020**
Instructor	**Dr. Soosan Beheshti**

Lab/Tutorial Report # 1

Report Title	**System Properties and Convolution**

Submission Date	**12**
Due Date	**25th October 2020**

Student Name	**Pak Hung Chu**
Student ID	**500894595**
Signature	

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work <u>will result in an investigation of Academic Misconduct and may result in a "0" on the work,</u> an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:*
http://www.ryerson.ca/senate/policies/pol60.pdf.

Problem A1 (C=1uF)

Code:

```
% CH2MP1.m : Chapter 2, MATLAB Program 1
% Script M-file determines characteristic roots of op-amp circuit.
% Set component values:
R = [1e4, 1e4, 1e4];
C = [1e-6, 1e-6];
% Determine coefficients for characteristic equation:
A = [1, (1/R(1)+1/R(2)+1/R(3))/C(2), 1/(R(1)*R(2)*C(1)*C(2))];
% Determine characteristic roots:
lambda = roots(A);
p = poly(A);
```

Command Window

```
>> CH2MP1
>> lambda

lambda =

 -261.8034
  -38.1966

>> p

p =

   1.0e+06 *

    0.0000   -0.0103    3.0103   -3.0000

>>
```

Problem A1 (C=1uF->1nF)

Code:

```
% CH2MP1.m : Chapter 2, MATLAB Program 1
% Script M-file determines characteristic roots of op-amp circuit.
% Set component values:
R = [1e4, 1e4, 1e4];
C = [1e-9, 1e-6];
% Determine coefficients for characteristic equation:
A = [1, (1/R(1)+1/R(2)+1/R(3))/C(2), 1/(R(1)*R(2)*C(1)*C(2))];
% Determine characteristic roots:
lambda = roots(A);
p = poly(A);
```

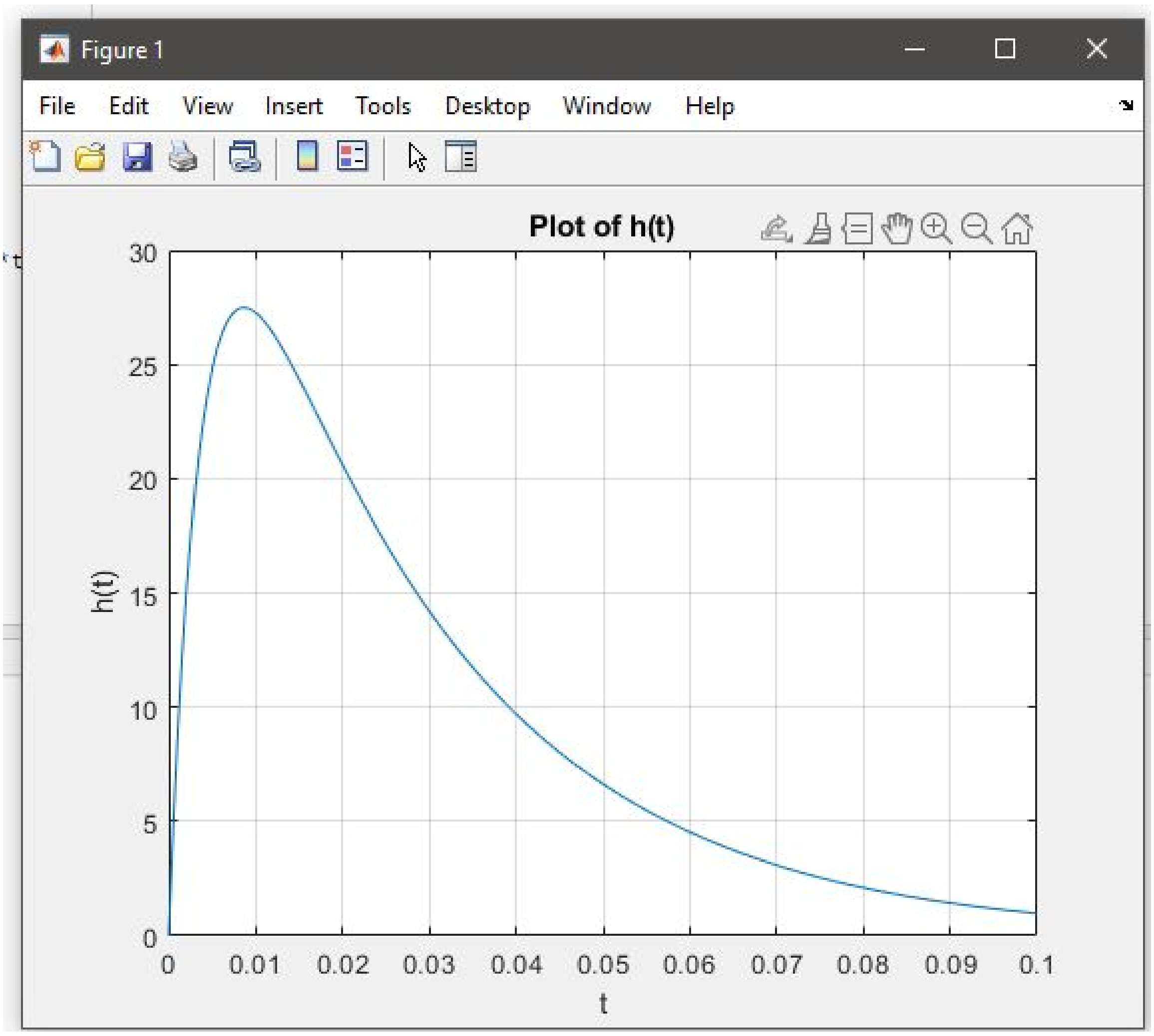
Command Window

```
>> CH2MP1
>> lambda

lambda =

   1.0e+03 *

  -0.1500 + 3.1587i
  -0.1500 - 3.1587i

>> p

p =

   1.0e+09 *

    0.0000   -0.0100    3.0100   -3.0000

>>
```

Problem A.2

Code:

```
% CH2MP1.m : Chapter 2, MATLAB Program 1
% Script M-file determines characteristic roots of op-amp circuit.
% Set component values:
R = [1e4, 1e4, 1e4];
C = [1e-6, 1e-6];
% Determine coefficients for characteristic equation:
A = [1, (1/R(1)+1/R(2)+1/R(3))/C(2), 1/(R(1)*R(2)*C(1)*C(2))];
% Determine characteristic roots:
lambda = roots(A);
p = poly(A);

% solve for LTE of h(t) function accordingly with roots given
t = (0:0.0005:0.1);
u = @(t) 1.0.*(t>=0);
h = @(t) -44.721359 .* exp(-261.8034 .*t)+ 44.721359 .* exp(-38.1966.*t).*u(t);

plot (t, h(t));
xlabel('t');
ylabel('h(t)');
title('Plot of h(t)');
axis ([0 0.1 0 30]);
grid;
```



Problem A3

Code:

```
function [lambda] = CH2MP2(R,C)
% CH2MP2.m : Chapter 2, MATLAB Program 2
% Function M-file finds characteristic roots of op-amp circuit.
% INPUTS: R = length-3 vector of resistances
% C = length-2 vector of capacitances
% OUTPUTS: lambda = characteristic roots
% Determine coefficients for characteristic equation:
A = [1, (1/R(1)+1/R(2)+1/R(3))/C(2), 1/(R(1)*R(2)*C(1)*C(2))];
% Determine characteristic roots:
lambda = roots(A);
```

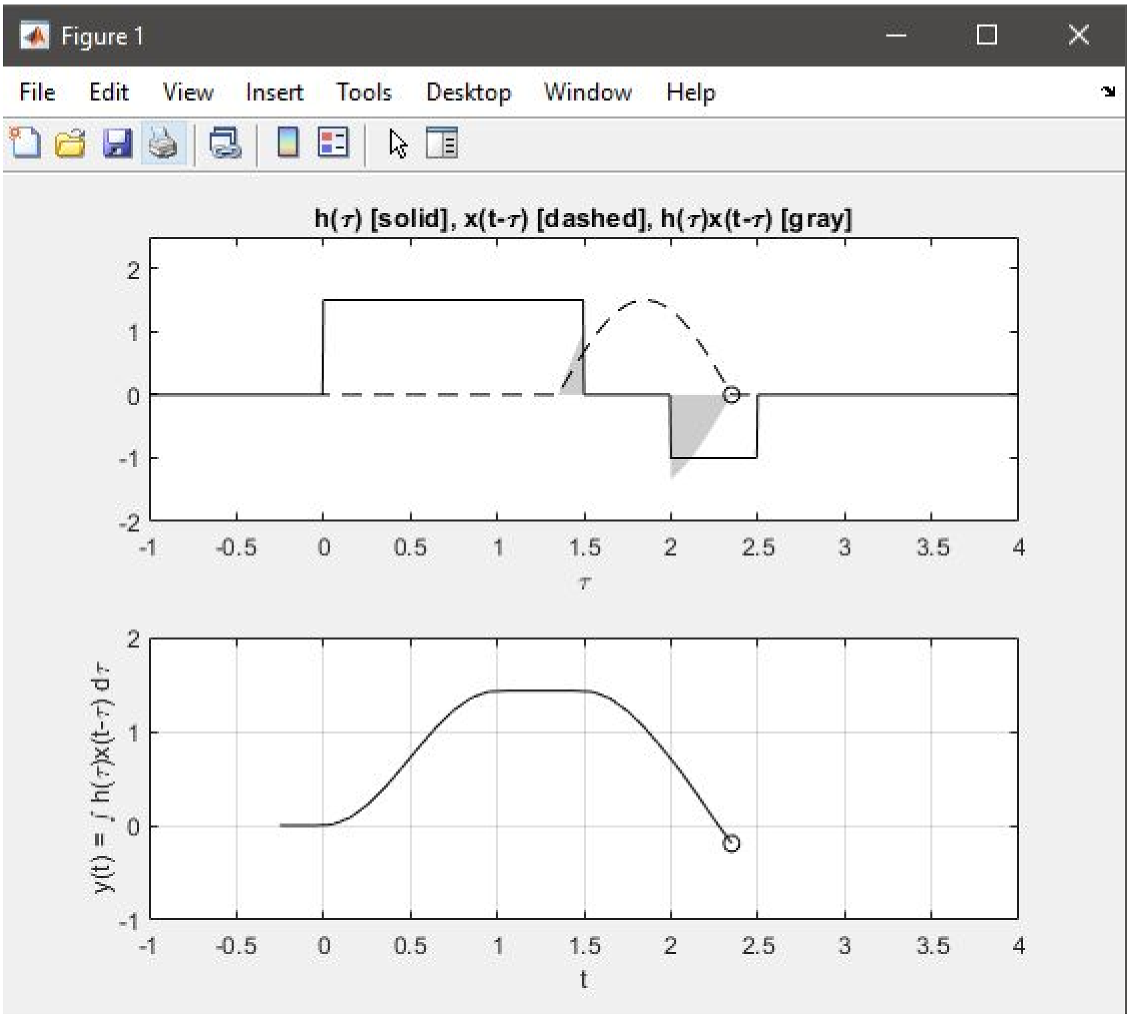
Command Window

```
>> lambda = CH2MP2([1e4, 1e4, 1e4],[1e-9, 1e-6])

lambda =

   1.0e+03 *

  -0.1500 + 3.1587i
  -0.1500 - 3.1587i
```

Problem B.1

Code:

```
% CH2MP4.m : Chapter 2, MATLAB Program 4
% Script M-file graphically demonstrates the convolution process.
figure(1) % Create figure window and make visible on screen
u = @(t) 1.0*(t>=0);
x = @(t) 1.5*sin(pi*t).*(u(t)-u(t-1));
h = @(t) 1.5*(u(t)-u(t-1.5))-u(t-2)+u(t-2.5);
dtau = 0.005; tau = -1:dtau:4;
ti = 0; tvec = -.25:.1:3.75;
y = NaN*zeros(1,length(tvec)); % Pre-allocate memory
for t = tvec,
ti = ti+1; % Time index
xh = x(t-tau).*h(tau); lxh = length(xh);
y(ti) = sum(xh.*dtau); % Trapezoidal approximation of convolution integral
subplot(2,1,1),plot(tau,h(tau),'k-',tau,x(t-tau),'k--',t,0,'ok');
axis([tau(1) tau(end) -2.0 2.5]);
patch([tau(1:end-1);tau(1:end-1);tau(2:end);tau(2:end)],...
[zeros(1,lxh-1);xh(1:end-1);xh(2:end);zeros(1,lxh-1)],...
[.8 .8 .8],'edgecolor','none');
xlabel('\tau'); title('h(\tau) [solid], x(t-\tau) [dashed], h(\tau)x(t-\tau) [gray]');
c = get(gca,'children'); set(gca,'children',[c(2);c(3);c(4);c(1)]);
subplot(2,1,2),plot(tvec,y,'k',tvec(ti),y(ti),'ok');
xlabel('t'); ylabel('y(t) = \int h(\tau)x(t-\tau) d\tau');
axis([tau(1) tau(end) -1.0 2.0]); grid;
drawnow;
end
```



Problem B.2

Convolute

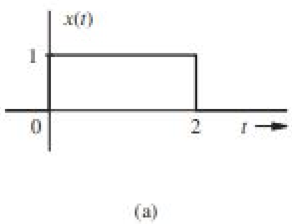
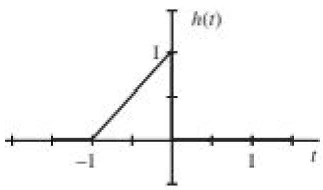


Figure P2.4-28

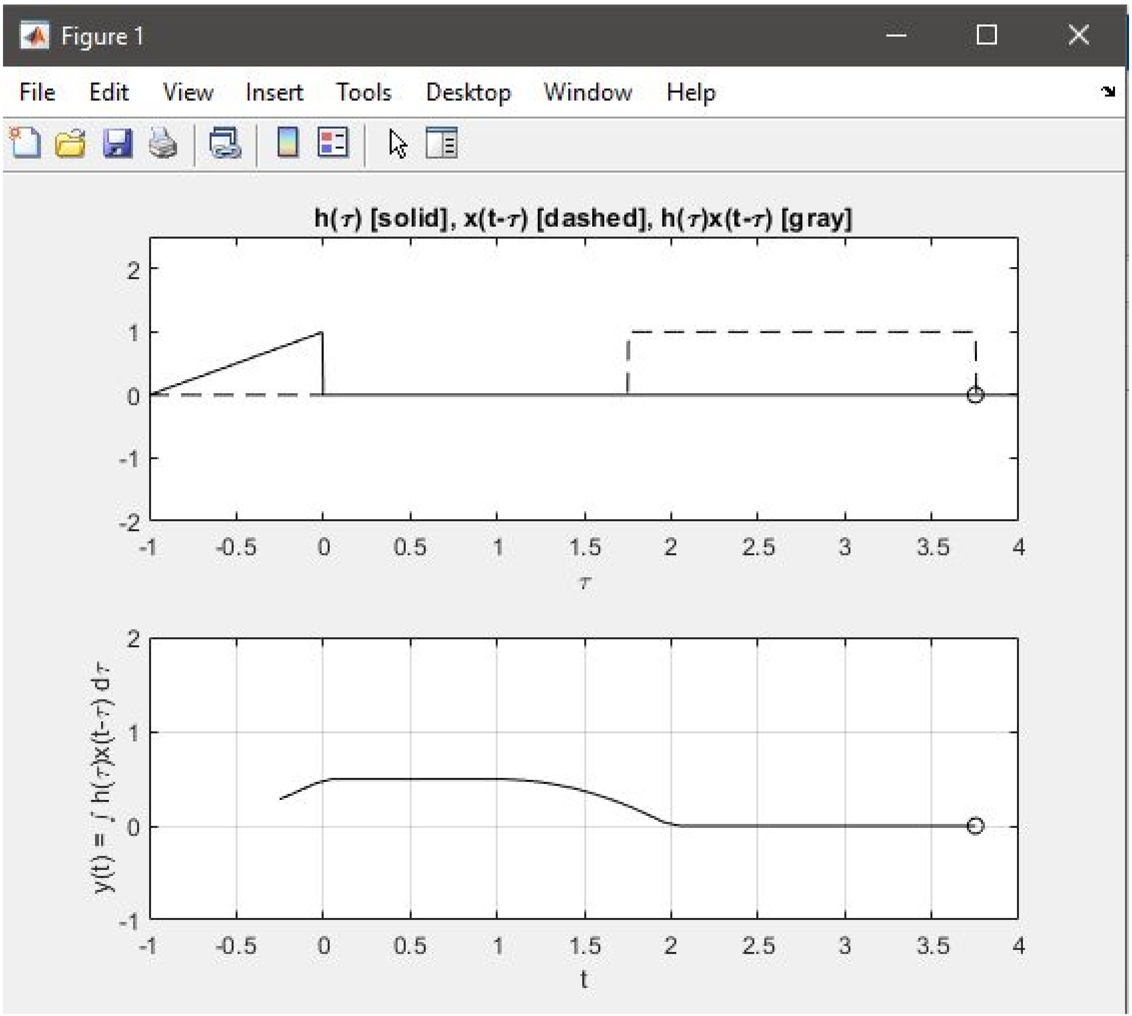
with



Code:

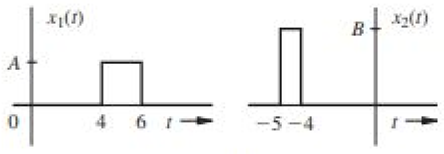
```
% CH2MP4.m : Chapter 2, MATLAB Program 4
% Script M-file graphically demonstrates the convolution process.
figure(1) % Create figure window and make visible on screen
u = @(t) 1.0.*(t>=0);
x = @(t) 1.0.*(u(t) - u(t-2));
h = @(t) 1.0.*(t+1).*(u(t+1)-u(t));
dtau = 0.005; tau = -1:dtau:4;
ti = 0; tvec = -.25:.1:3.75;
y = NaN*zeros(1,length(tvec)); % Pre-allocate memory
for t = tvec,
```

```
ti = ti+1; % Time index
xh = x(t-tau).*h(tau); lxh = length(xh);
y(ti) = sum(xh.*dtau); % Trapezoidal approximation of convolution integral
subplot(2,1,1),plot(tau,h(tau),'k-',tau,x(t-tau),'k--',t,0,'ok');
axis([tau(1) tau(end) -2.0 2.5]);
patch([tau(1:end-1);tau(1:end-1);tau(2:end);tau(2:end)],...
[zeros(1,lxh-1);xh(1:end-1);xh(2:end);zeros(1,lxh-1)],...
[.8 .8 .8],'edgecolor','none');
xlabel('\tau'); title('h(\tau) [solid], x(t-\tau) [dashed], h(\tau)x(t-\tau) [gray]');
c = get(gca,'children'); set(gca,'children',[c(2);c(3);c(4);c(1)]);
subplot(2,1,2),plot(tvec,y,'k',tvec(ti),y(ti),'ok');
xlabel('t'); ylabel('y(t) = \int h(\tau)x(t-\tau) d\tau');
axis([tau(1) tau(end) -1.0 2.0]); grid;
drawnow;
end
```

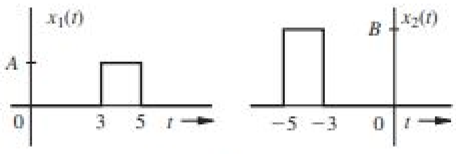


Problem B.3

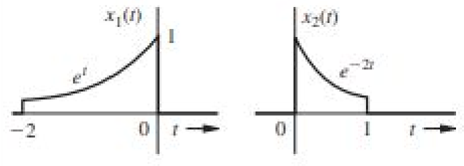
Convolute a, b, h



(a)



(b)

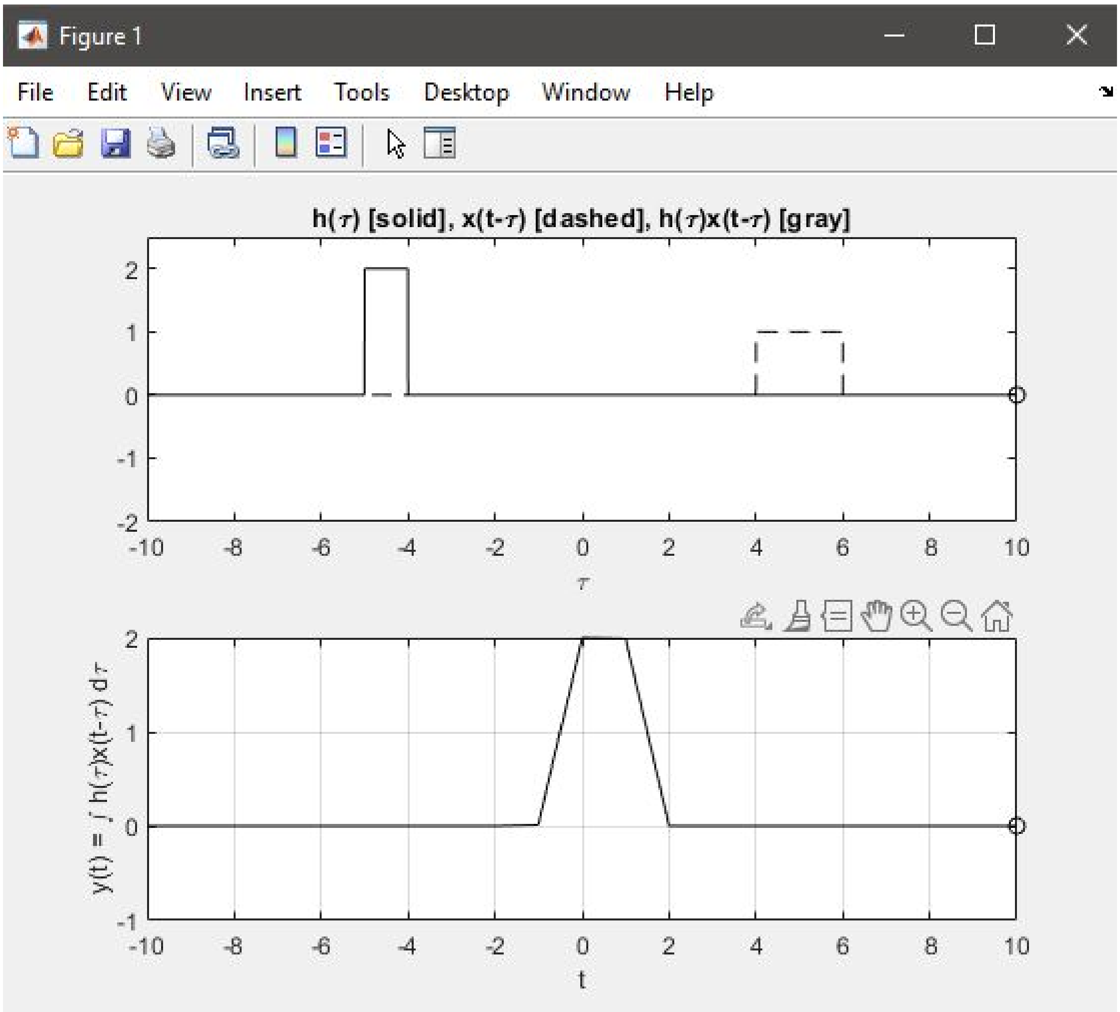


(h)

Convolute A:

Code:

```
% CH2MP4.m : Chapter 2, MATLAB Program 4
% Script M-file graphically demonstrates the convolution process.
figure(1) % Create figure window and make visible on screen
u = @(t) 1.0.*(t>=0);
x = @(t) 1.0.*(u(t-4) - u(t-6));
h = @(t) 2.0.*(u(t+5) - u(t+4));
dtau = 0.005; tau = -10:dtau:10;
ti = 0; tvec = [-10:1:10];
y = NaN*zeros(1,length(tvec)); % Pre-allocate memory
for t = tvec,
ti = ti+1; % Time index
xh = x(t-tau).*h(tau); lxh = length(xh);
y(ti) = sum(xh.*dtau); % Trapezoidal approximation of convolution integral
subplot(2,1,1),plot(tau,h(tau),'k-',tau,x(t-tau),'k--',t,0,'ok');
axis([tau(1) tau(end) -2.0 2.5]);
patch([tau(1:end-1);tau(1:end-1);tau(2:end);tau(2:end)],...
[zeros(1,lxh-1);xh(1:end-1);xh(2:end);zeros(1,lxh-1)],...
[.8 .8 .8],'edgecolor','none');
xlabel('\tau'); title('h(\tau) [solid], x(t-\tau) [dashed], h(\tau)x(t-\tau) [gray]');
c = get(gca,'children'); set(gca,'children',[c(2);c(3);c(4);c(1)]);
subplot(2,1,2),plot(tvec,y,'k',tvec(ti),y(ti),'ok');
xlabel('t'); ylabel('y(t) = \int h(\tau)x(t-\tau) d\tau');
axis([tau(1) tau(end) -1.0 2.0]); grid;
drawnow;
end
```

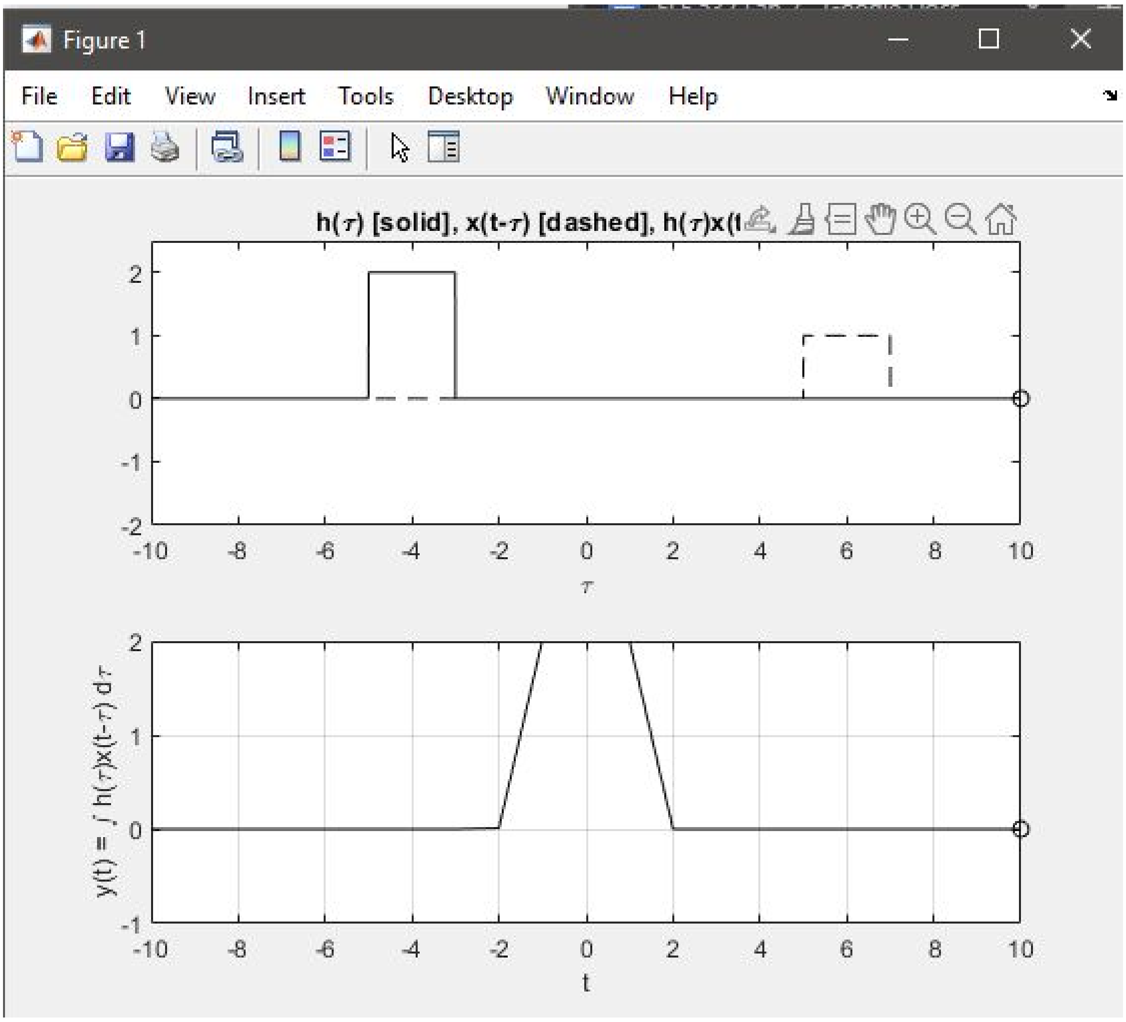


Convolute B

Code:

```
% CH2MP4.m : Chapter 2, MATLAB Program 4
% Script M-file graphically demonstrates the convolution process.
figure(1) % Create figure window and make visible on screen
u = @(t) 1.0.*(t>=0);
x = @(t) 1.0.*(u(t-3) - u(t-5));
h = @(t) 2.0.*(u(t+5) - u(t+3));
dtau = 0.005; tau = -10:dtau:10;
ti = 0; tvec = [-10:1:10];
y = NaN*zeros(1,length(tvec)); % Pre-allocate memory
for t = tvec,
ti = ti+1; % Time index
xh = x(t-tau).*h(tau); lxh = length(xh);
y(ti) = sum(xh.*dtau); % Trapezoidal approximation of convolution integral
subplot(2,1,1),plot(tau,h(tau),'k-',tau,x(t-tau),'k--',t,0,'ok');
axis([tau(1) tau(end) -2.0 2.5]);
patch([tau(1:end-1);tau(1:end-1);tau(2:end);tau(2:end)],...
[zeros(1,lxh-1);xh(1:end-1);xh(2:end);zeros(1,lxh-1)],...
[.8 .8 .8],'edgecolor','none');
xlabel('\tau'); title('h(\tau) [solid], x(t-\tau) [dashed], h(\tau)x(t-\tau) [gray]');
c = get(gca,'children'); set(gca,'children',[c(2);c(3);c(4);c(1)]);
subplot(2,1,2),plot(tvec,y,'k',tvec(ti),y(ti),'ok');
```

```
xlabel('t'); ylabel('y(t) = \int h(\tau)x(t-\tau) d\tau');
axis([tau(1) tau(end) -1.0 2.0]); grid;
drawnow;
end
```

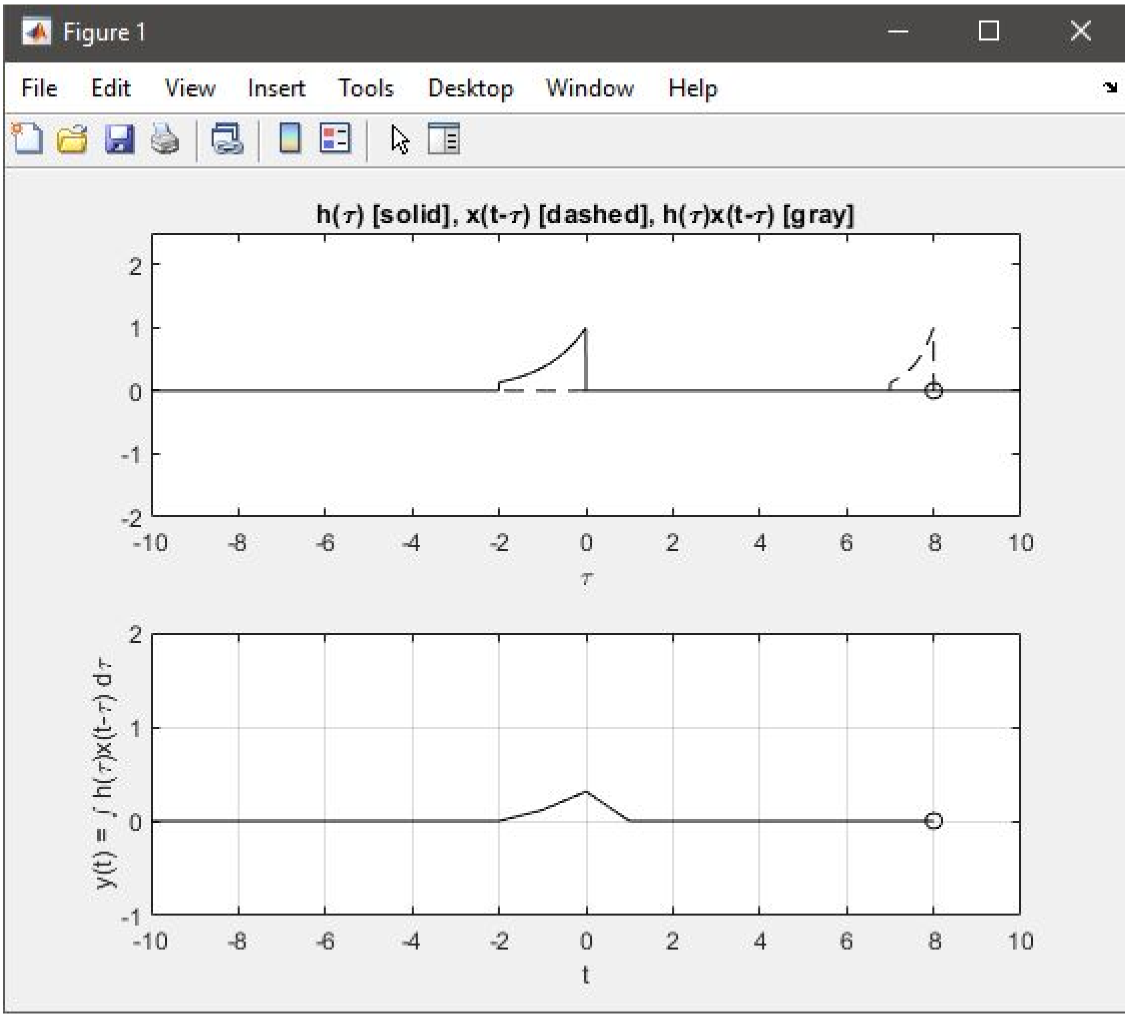


Convolute h

Code:

```
% CH2MP4.m : Chapter 2, MATLAB Program 4
% Script M-file graphically demonstrates the convolution process.
figure(1) % Create figure window and make visible on screen
u = @(t) 1.0.*(t>=0);
x = @(t) exp(-2*t).*(u(t) - u(t-1));
h = @(t) exp(t).*(u(t+2) - u(t));
dtau = 0.005; tau = -10:dtau:10;
ti = 0; tvec = [-10:1:8];
y = NaN*zeros(1,length(tvec)); % Pre-allocate memory
for t = tvec,
ti = ti+1; % Time index
xh = x(t-tau).*h(tau); lxh = length(xh);
y(ti) = sum(xh.*dtau); % Trapezoidal approximation of convolution integral
subplot(2,1,1),plot(tau,h(tau),'k-',tau,x(t-tau),'k--',t,0,'ok');
axis([tau(1) tau(end) -2.0 2.5]);
```

```
patch([tau(1:end-1);tau(1:end-1);tau(2:end);tau(2:end)],...
[zeros(1,lxh-1);xh(1:end-1);xh(2:end);zeros(1,lxh-1)],...
[.8 .8 .8],'edgecolor','none');
xlabel('\tau'); title('h(\tau) [solid], x(t-\tau) [dashed], h(\tau)x(t-\tau) [gray]');
c = get(gca,'children'); set(gca,'children',[c(2);c(3);c(4);c(1)]);
subplot(2,1,2),plot(tvec,y,'k',tvec(ti),y(ti),'ok');
xlabel('t'); ylabel('y(t) = \int h(\tau)x(t-\tau) d\tau');
axis([tau(1) tau(end) -1.0 2.0]); grid;
drawnow;
end
```

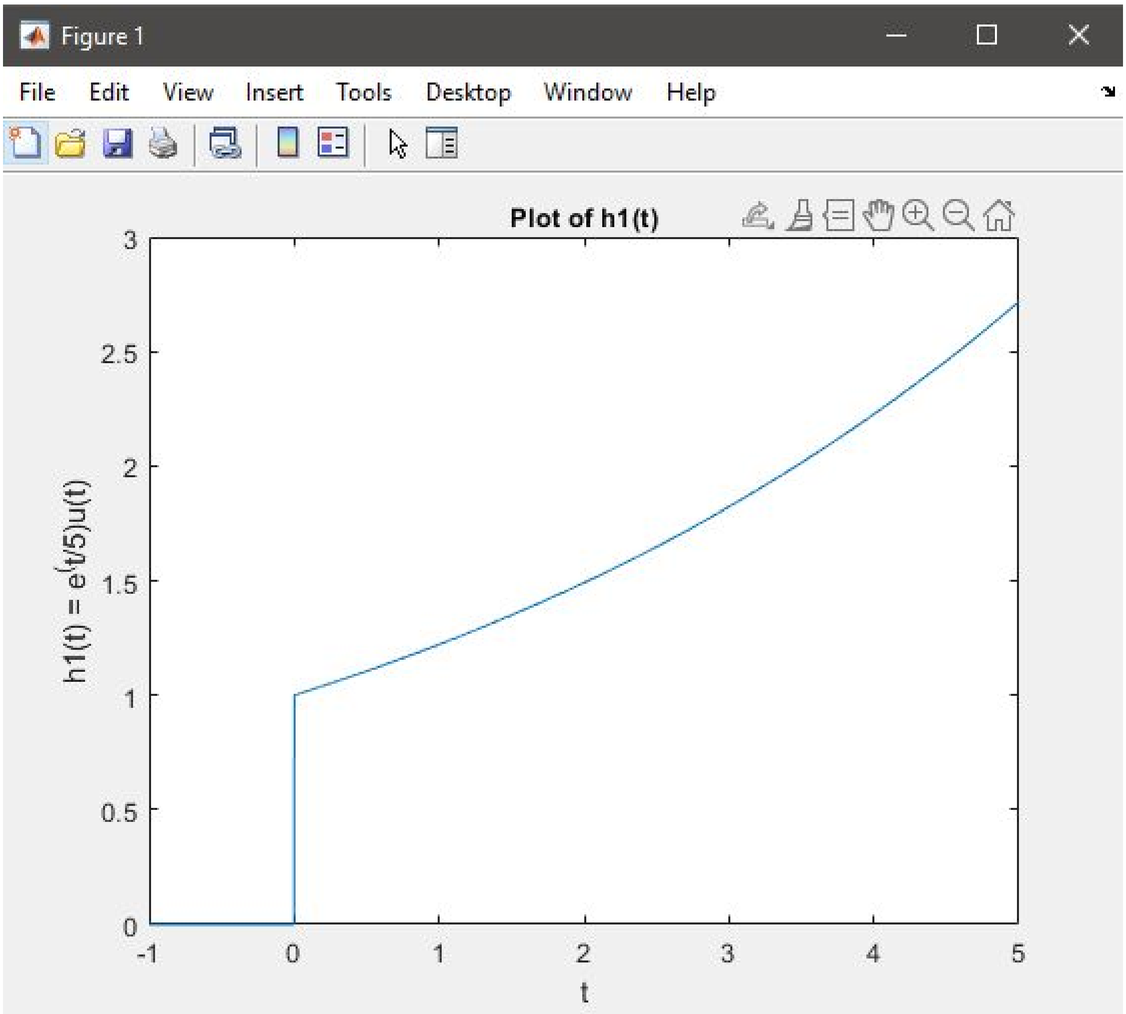


Problem C.1

Plot h1(t)

Code:

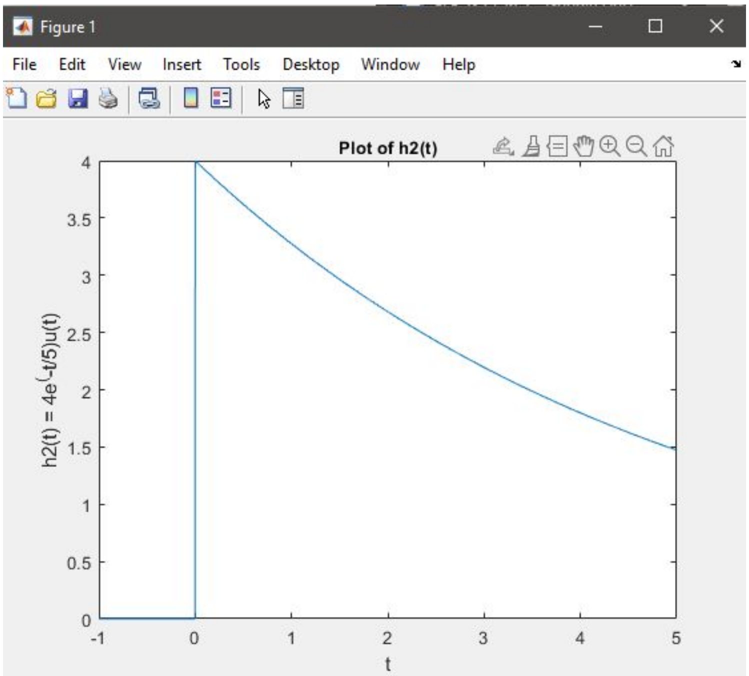
```
u = @(t) 1.0.*(t>=0);
t = (-1:0.001:5);
h1= @(t) exp(t/5).*u(t);
plot(t,h1(t));
xlabel('t');
ylabel('h1(t) = e^(t/5)u(t)');
title('Plot of h1(t)')
```



Plot h2(t):

Code:

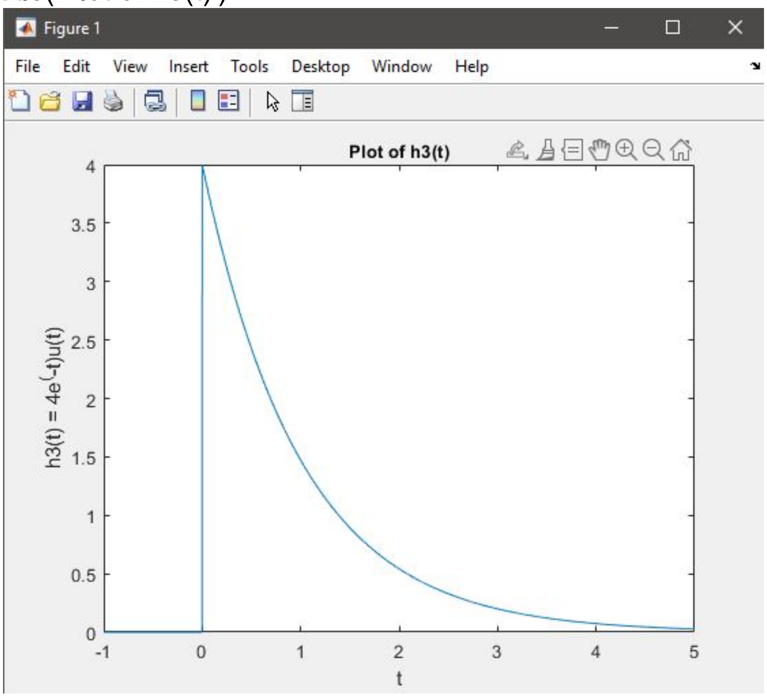
```
u = @(t) 1.0.*(t>=0);
t = (-1:0.001:5);
h2= @(t) 4*exp(-t/5).*u(t);
plot(t,h2(t));
xlabel('t');
ylabel('h2(t) = 4e^(-t/5)u(t)');
title('Plot of h2(t)')
```



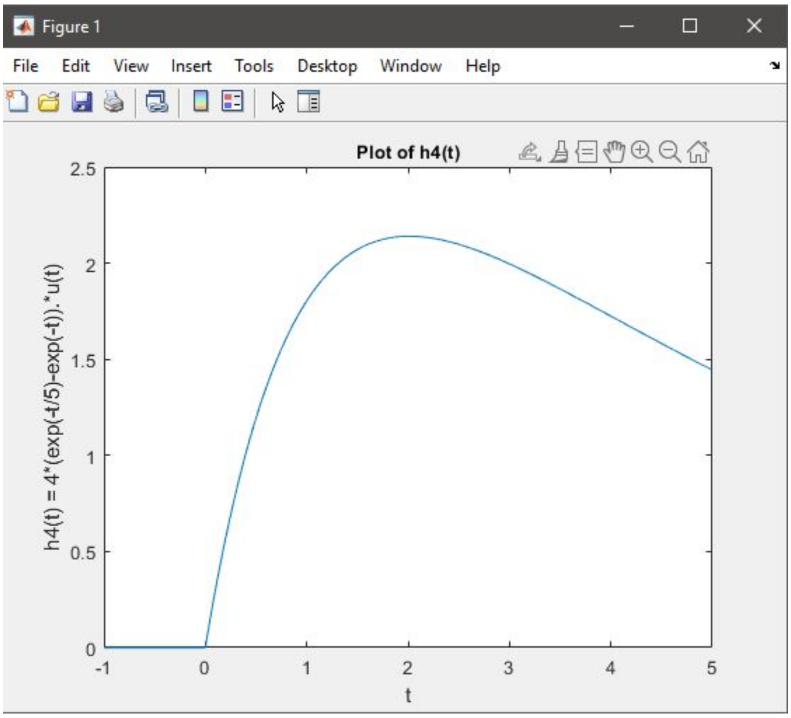
Plot h3(t)

Code:

```
u = @(t) 1.0.*(t>=0);
t = (-1:0.001:5);
h3= @(t) 4*exp(-t).*u(t);
plot(t,h3(t));
xlabel('t');
ylabel('h3(t) = 4e^(t)u(t)');
title('Plot of h3(t)')
```



Plot h4(t)
Code:
```
u = @(t) 1.0.*(t>=0);
t = (-1:0.001:5);
h4= @(t) 4*(exp(-t/5)-exp(-t)).*u(t);
plot(t,h4(t));
xlabel('t');
ylabel('h4(t) = 4*(exp(-t/5)-exp(-t)).*u(t)');
title('Plot of h4(t)')
```



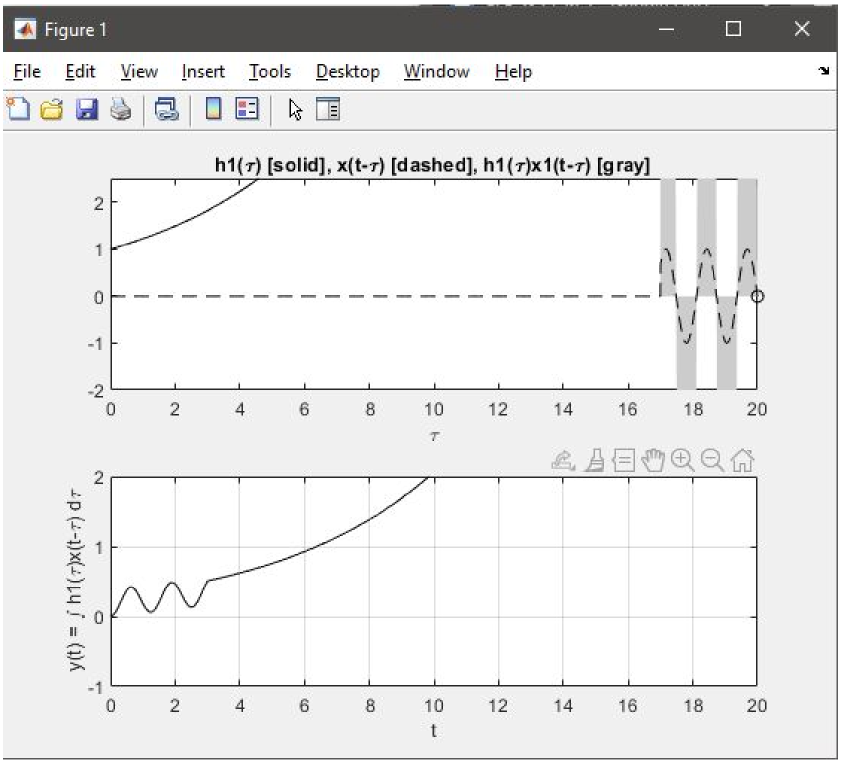
Problem C.2

Eigenvalues:
h1(t) = ⅕
h2(t) = -⅕
h3(t) = -1
h4(t) = -⅕ and -1

Problem C.3

For h1(t)
Code:
```
% CH2MP4.m : Chapter 2, MATLAB Program 4
% Script M-file graphically demonstrates the convolution process.
figure(1) % Create figure window and make visible on screen
u = @(t) 1.0.*(t>=0);
x = @(t) sin(5.*t).*(u(t)-u(t-3));
h = @(t) exp(t/5).*u(t);
dtau = 0.005; tau = 0:dtau:20;
ti = 0; tvec = [0:0.1:20];
y = NaN*zeros(1,length(tvec)); % Pre-allocate memory
```

```
for t = tvec,
ti = ti+1; % Time index
xh = x(t-tau).*h(tau);lxh = length(xh);
y(ti) = sum(xh.*dtau); % Trapezoidal approximation of convolution integral
subplot(2,1,1),plot(tau,h(tau),'k-',tau,x(t-tau),'k--',t,0,'ok');
axis([tau(1) tau(end) -2.0 2.5]);
patch([tau(1:end-1);tau(1:end-1);tau(2:end);tau(2:end)],...
[zeros(1,lxh-1);xh(1:end-1);xh(2:end);zeros(1,lxh-1)],...
[.8 .8 .8],'edgecolor','none');
xlabel('\tau'); title('h1(\tau) [solid], x(t-\tau) [dashed], h1(\tau)x1(t-\tau) [gray]');
c = get(gca,'children'); set(gca,'children',[c(2);c(3);c(4);c(1)]);
subplot(2,1,2),plot(tvec,y,'k',tvec(ti),y(ti),'ok');
xlabel('t'); ylabel('y(t) = \int h1(\tau)x(t-\tau) d\tau');
axis([tau(1) tau(end) -1.0 2.0]); grid;
drawnow
end
```

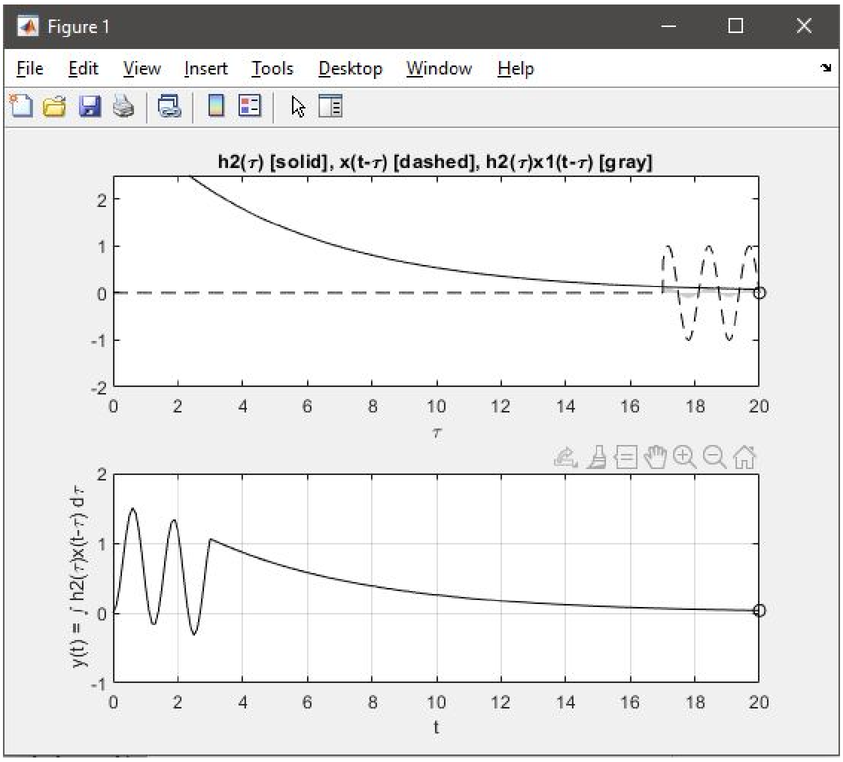


For h2(t)

Code:

```
% CH2MP4.m : Chapter 2, MATLAB Program 4
% Script M-file graphically demonstrates the convolution process.
figure(1) % Create figure window and make visible on screen
u = @(t) 1.0.*(t>=0);
x = @(t) sin(5.*t).*(u(t)-u(t-3));
h = @(t) 4*exp(-t/5).*u(t);
dtau = 0.005; tau = 0:dtau:20;
ti = 0; tvec = [0:0.1:20];
y = NaN*zeros(1,length(tvec)); % Pre-allocate memory
for t = tvec,
ti = ti+1; % Time index
xh = x(t-tau).*h(tau);lxh = length(xh);
```

```
y(ti) = sum(xh.*dtau); % Trapezoidal approximation of convolution integral
subplot(2,1,1),plot(tau,h(tau),'k-',tau,x(t-tau),'k--',t,0,'ok');
axis([tau(1) tau(end) -2.0 2.5]);
patch([tau(1:end-1);tau(1:end-1);tau(2:end);tau(2:end)],...
[zeros(1,lxh-1);xh(1:end-1);xh(2:end);zeros(1,lxh-1)],...
[.8 .8 .8],'edgecolor','none');
xlabel('\tau'); title('h2(\tau) [solid], x(t-\tau) [dashed], h2(\tau)x1(t-\tau) [gray]');
c = get(gca,'children'); set(gca,'children',[c(2);c(3);c(4);c(1)]);
subplot(2,1,2),plot(tvec,y,'k',tvec(ti),y(ti),'ok');
xlabel('t'); ylabel('y(t) = \int h2(\tau)x(t-\tau) d\tau');
axis([tau(1) tau(end) -1.0 2.0]); grid;
drawnow
end
```

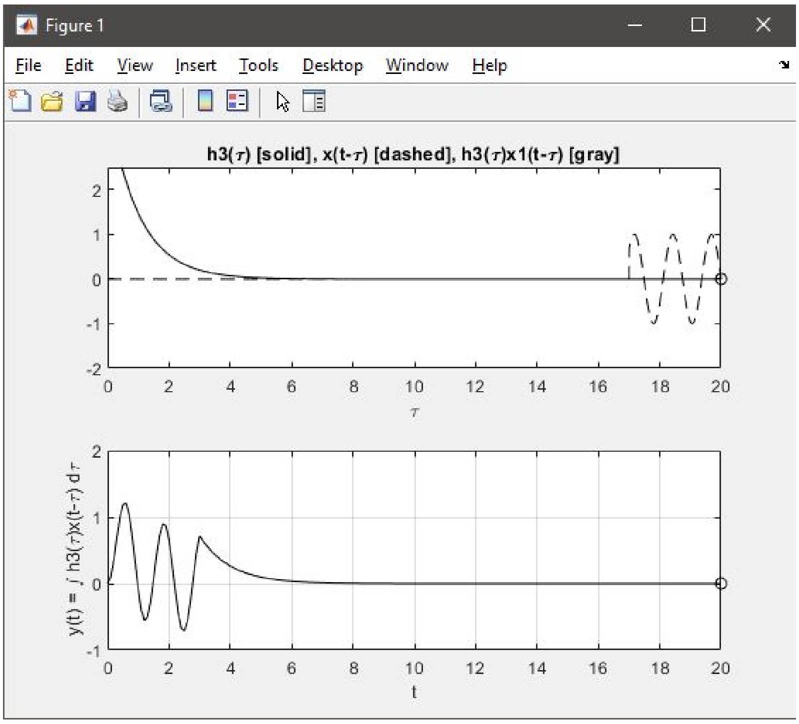


For h3(t)

Code:

```
% CH2MP4.m : Chapter 2, MATLAB Program 4
% Script M-file graphically demonstrates the convolution process.
figure(1) % Create figure window and make visible on screen
u = @(t) 1.0.*(t>=0);
x = @(t) sin(5.*t).*(u(t)-u(t-3));
h = @(t) 4*exp(-t).*u(t);
dtau = 0.005; tau = 0:dtau:20;
ti = 0; tvec = [0:0.1:20];
y = NaN*zeros(1,length(tvec)); % Pre-allocate memory
for t = tvec,
ti = ti+1; % Time index
xh = x(t-tau).*h(tau);lxh = length(xh);
y(ti) = sum(xh.*dtau); % Trapezoidal approximation of convolution integral
subplot(2,1,1),plot(tau,h(tau),'k-',tau,x(t-tau),'k--',t,0,'ok');
axis([tau(1) tau(end) -2.0 2.5]);
```

```
patch([tau(1:end-1);tau(1:end-1);tau(2:end);tau(2:end)],...
[zeros(1,lxh-1);xh(1:end-1);xh(2:end);zeros(1,lxh-1)],...
[.8 .8 .8],'edgecolor','none');
xlabel('\tau'); title('h3(\tau) [solid], x(t-\tau) [dashed], h3(\tau)x1(t-\tau) [gray]');
c = get(gca,'children'); set(gca,'children',[c(2);c(3);c(4);c(1)]);
subplot(2,1,2),plot(tvec,y,'k',tvec(ti),y(ti),'ok');
xlabel('t'); ylabel('y(t) = \int h3(\tau)x(t-\tau) d\tau');
axis([tau(1) tau(end) -1.0 2.0]); grid;
drawnow
end
```

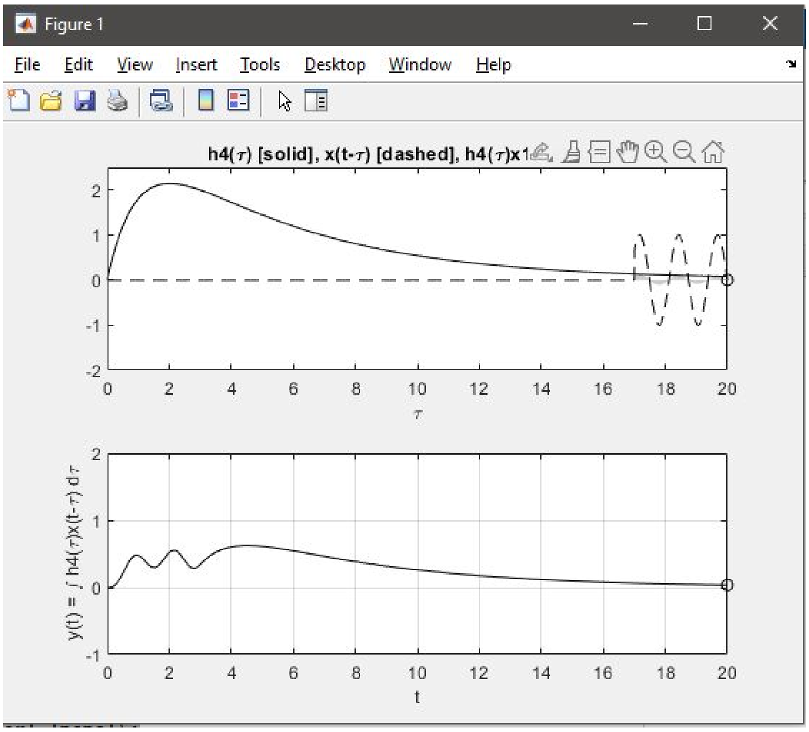


For h4(t)

Code:

```
% CH2MP4.m : Chapter 2, MATLAB Program 4
% Script M-file graphically demonstrates the convolution process.
figure(1) % Create figure window and make visible on screen
u = @(t) 1.0.*(t>=0);
x = @(t) sin(5.*t).*(u(t)-u(t-3));
h = @(t) 4*(exp(-t/5)-exp(-t)).*u(t);
dtau = 0.005; tau = 0:dtau:20;
ti = 0; tvec = [0:0.1:20];
y = NaN*zeros(1,length(tvec)); % Pre-allocate memory
for t = tvec,
ti = ti+1; % Time index
xh = x(t-tau).*h(tau);lxh = length(xh);
y(ti) = sum(xh.*dtau); % Trapezoidal approximation of convolution integral
subplot(2,1,1),plot(tau,h(tau),'k-',tau,x(t-tau),'k--',t,0,'ok');
axis([tau(1) tau(end) -2.0 2.5]);
patch([tau(1:end-1);tau(1:end-1);tau(2:end);tau(2:end)],...
[zeros(1,lxh-1);xh(1:end-1);xh(2:end);zeros(1,lxh-1)],...
[.8 .8 .8],'edgecolor','none');
```

```
xlabel('\tau'); title('h4(\tau) [solid], x(t-\tau) [dashed], h4(\tau)x1(t-\tau) [gray]');
c = get(gca,'children'); set(gca,'children',[c(2);c(3);c(4);c(1)]);
subplot(2,1,2),plot(tvec,y,'k',tvec(ti),y(ti),'ok');
xlabel('t'); ylabel('y(t) = \int h4(\tau)x(t-\tau) d\tau');
axis([tau(1) tau(end) -1.0 2.0]); grid;
drawnow
end
```



The relationship between these three systems is that y4(t) is an interconnected parallel system of y2(t) and the negative of y3(t). Thus y4(t) = y2(t) + (- y3(t))

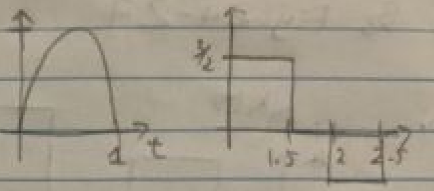
Problem D.1
B.1

D1 B1

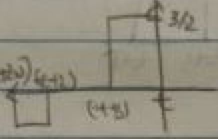
$x(t) = 1.5\sin(\pi t)[u(t) - u(t-1)]$

$h(t) = 1.5[u(t) - u(t-1.5)] - u(t-2) + u(t-2.5)$

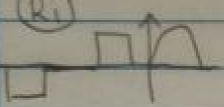
$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$



$h(t-\tau)$:

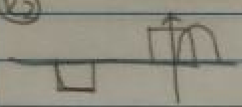


R1



$t < 0 \therefore y(t) = 0$

R2

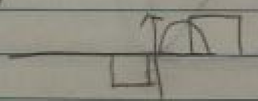


$0 \le t \le 1 \quad y(t) = \int_0^t \frac{3}{2}\left(\frac{3}{2}\sin(\pi\tau)d\tau\right) = -\frac{9}{4}\pi\cos(\pi\tau)\Big|_0^t = -\frac{9}{4\pi}\cos(\pi t) + \frac{9}{4\pi}$

R3

$1 \le t \le 3/2 \quad y(t) = \int^1 \frac{3}{2}d\tau = \frac{3}{2}$

R4



$3/2 \le t \le 2 \quad y(t) = \int_{t+3/2}^{1} \frac{3}{2}\left(\frac{3}{2}\sin(\pi\tau)\right)d\tau = -\frac{9}{4\pi}\cos(\pi\tau)\Big|_{t+3/2}^{1}$

$= \frac{9}{4\pi} + \frac{9}{4\pi}\cos\left(\pi\left(t+\frac{3}{2}\right)\right)$

R5

$2 \le t \le 2.5 \quad y(t) = \int_0^{t+2} -\frac{3}{2}\sin(\pi\tau)d\tau$

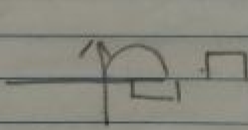
$= \frac{3}{2\pi}\cos(\pi\tau)\Big|_0^{t+2} = \frac{3}{2\pi}\cos(\pi(t+2)) - \frac{3}{2\pi}$

R6

$2.5 \le t \le 3 \quad y(t) = \int_{t+2.5}^{t+2} -3/2\sin(\pi\tau)d\tau = \frac{3}{2\pi}\cos(\pi\tau)\Big|_{t+2.5}^{t+2}$

$= \frac{3}{2\pi}\cos(\pi(t+2)) - \frac{3}{2\pi}\cos(\pi(t+2.5))$

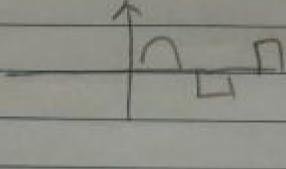
R7



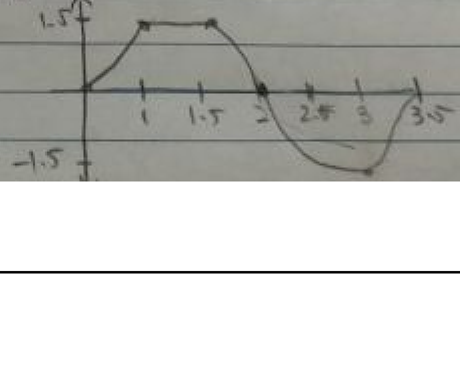
$3 \le t \le 3.5 \quad y(t) = \int_{t+2.5}^{1} -3/2\sin(\pi\tau)d\tau = \frac{3}{2\pi}\cos(\pi\tau)\Big|_{t+2.5}^{1}$

$= -\frac{3}{2\pi} - \frac{3}{2\pi}\cos(\pi(t+2.5))$

R8



$y(t) = 0$

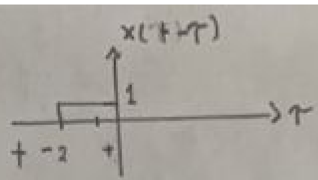


B.2

Doing B2

$x(t) = u(t) - u(t-2)$

$h(t) = (1 - \frac{t}{2})[u(t) - u(t-2)]$



Region 2

$0 \le t-2 \le 2$

$2 \le t \le 4$

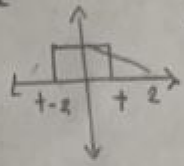
$y(t) = \int_t^2 (1 - \frac{t}{2}) d\tau = \tau - \frac{\tau^2}{4} \Big|_t^2$

$= (2-1) - (t - \frac{t^2}{4})$

$= 1 - (t-2) + \frac{(t-2)^2}{4}$

Region 1

$0 \le t \le 2$



$\int_0^t 1 - \frac{t}{2} = \tau - \frac{\tau^2}{4} \Big|_0^t$

$= t - \frac{t^2}{4}$

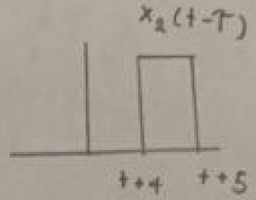
B.3a

B.3a

$x_1(t) = A(u(t-4) - u(t-6))$

$x_2(t) = B[u(t+5) - u(t+4)]$

$x_2(t-\tau)$



Region 1

No overlap for $t+5 \leq 4$

$t < -1$

R2

$4 \leq t+5 \leq 5$

$-1 \leq t \leq 0$

$y(t) = \int_4^{t+5} AB\,d\tau$

$= AB\tau \Big|_4^{t+5}$

$= AB(t+1)$

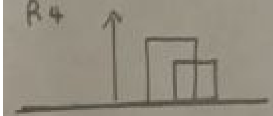
R3

$0 \leq t \leq 1$

$y(t) = \int_0^1 AB\,d\tau$

$= AB$

R4



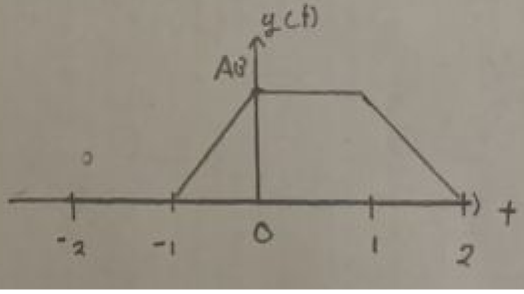
$1 \leq t \leq 2$

$y(t) = \int_{t+4}^{6} AB\,d\tau = 6-(t+4)AB$

$= (2-t)AB$

R5

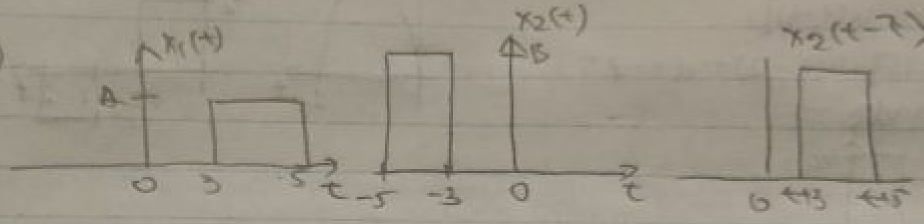
No overlap for $5 \leq t$



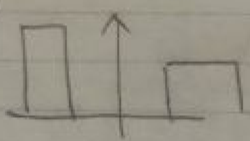
B.3b

D1 B3 Fig. 2.4-2.7

b)



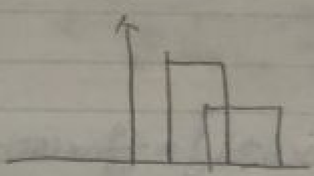
R1



$t < -2$

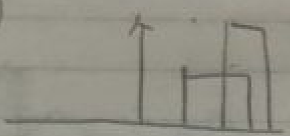
$y(t) = 0$

R2



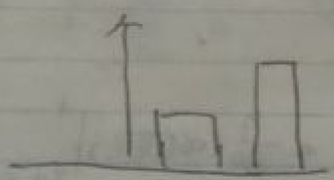
$-2 \le t \le 0$ $\quad y(t) = \int_3^{t+5} AB\,d\tau = AB\tau\Big|_3^{t+5} = AB(t+2)$

R3

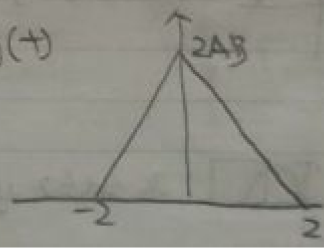


$0 \le t \le 2$ $\quad y(t) = \int_2^{t+2} AB\,d\tau = AB(2-t)$

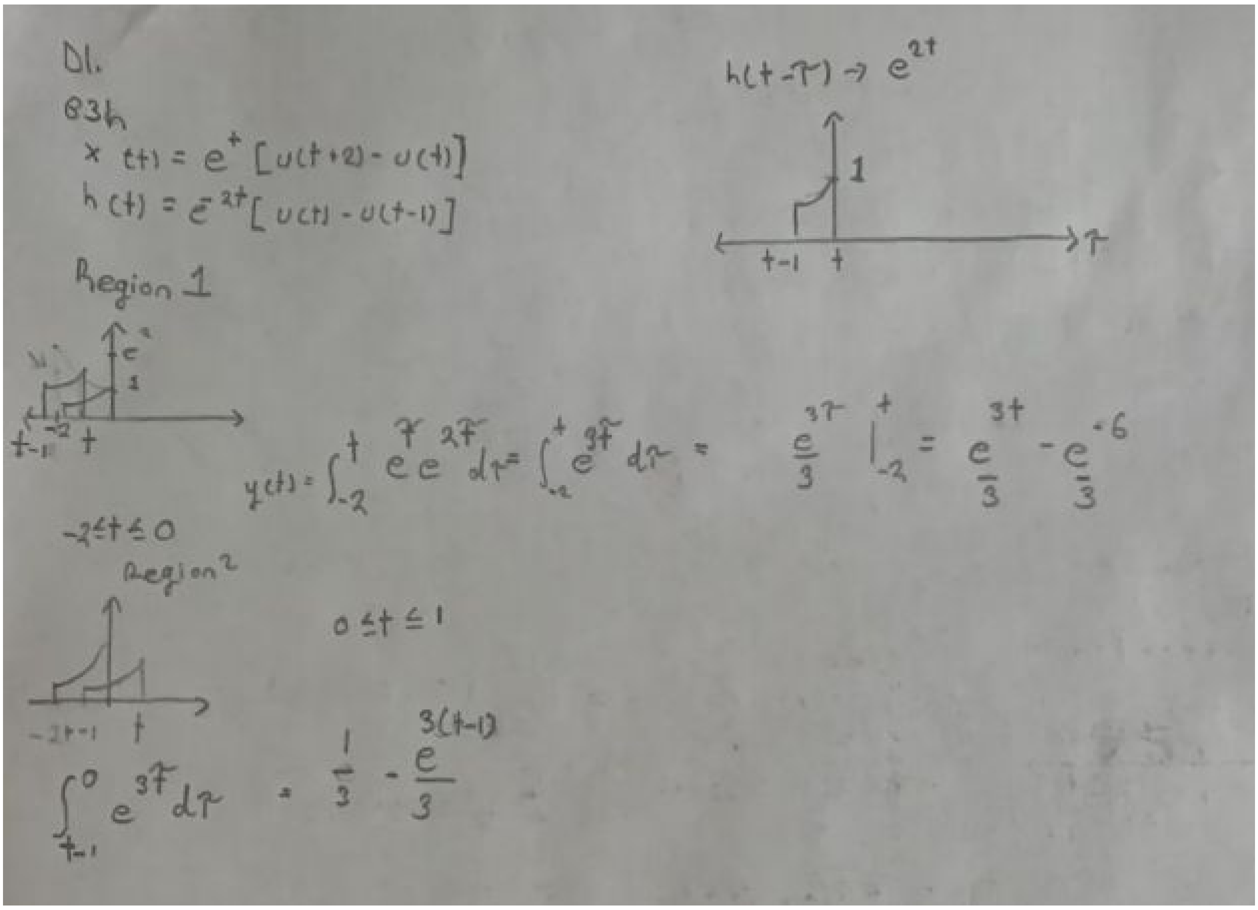
R4



$t \ge 2$ $\quad y(t) = 0$



B.3h



Problem D.2
The sum of the lengths of each function is equal to the length of the convolution of two functions