Ryerson University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	Signals and Systems I
Course Number:	ELE532
Semester/Year (e.g.F2016)	F2019

Instructor:	**Dimitri Androutsos**

Assignment/Lab Number:	2
Assignment/Lab Title:	System Properties and Convolution

Submission Date:	October 20, 2019
Due Date:	October 20, 2019

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Hashem	Khaled	500708209	3	
Yousuf	**Syed**	500833920	3	

Reset Form

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a [illegible] on the work, 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/current/pol60.pdf

Lab 2: System Properties and Convolution

By Syed Yousuf and Khaled Hashem

A:
A.1:
The “poly” command is used to turn a matrix that contains the roots of an equation back into the original polynomial equation.

Matlab

```
% Set component values:
R = [1e4, 1e4, 1e4]; C = [1e-6, 1e-6];
% Determine the coefficients for characteristic equation:
A1 = [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(A1);

%The poly command takes in the matrix of roots and  returns the original
%polynomial equation.
poly(lambda)
```

A.2:

Matlab:

```
% Set component values:
R = [1e4, 1e4, 1e4]; C = [1e-6, 1e-6];
% Determine the coefficients for characteristic equation:
A1 = [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(A1);

%The poly command takes in the matrix of roots and  returns the original
%polynomial equation.
poly(lambda);

t = [0:0.0005:0.1];
u = @(t) 1.0* (t>=0);
```

```
h = @(t) (C(1).* exp(lambda(1).* t) + C(2).* exp(lambda(2).*t)).*(u(t));

plot(t,h(t))
```

A.3:
Matlab:

```
function[lambda] = CH2MP2(R,C)

%Determine the coefficients for characteristics 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)
```

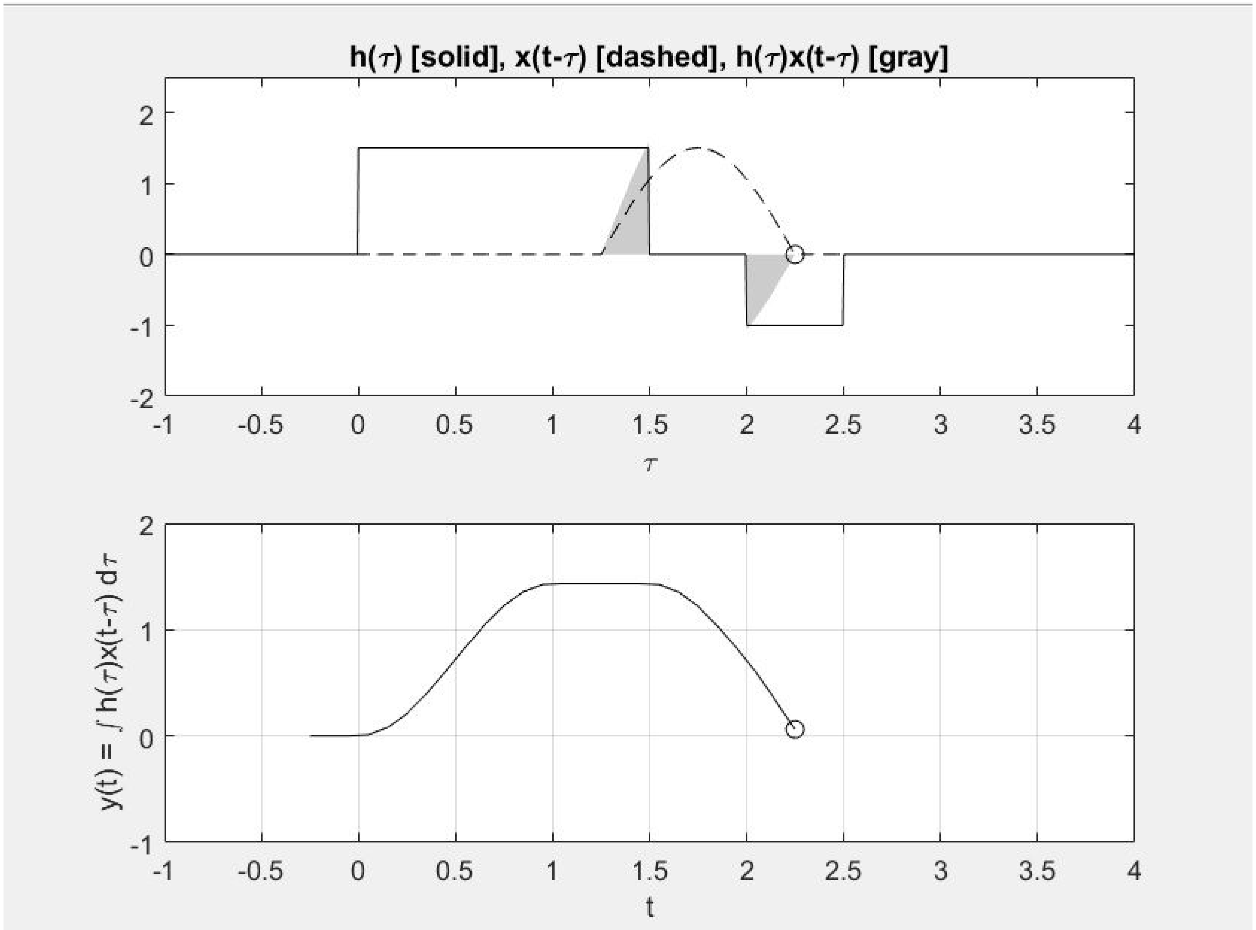
B:
B.1:

- x (t) = [1.5sin(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) = x(t) * h(t)
- * refers to convolution.

Also, here is the matlab code for CH2MP4, which creates the fiqure below.

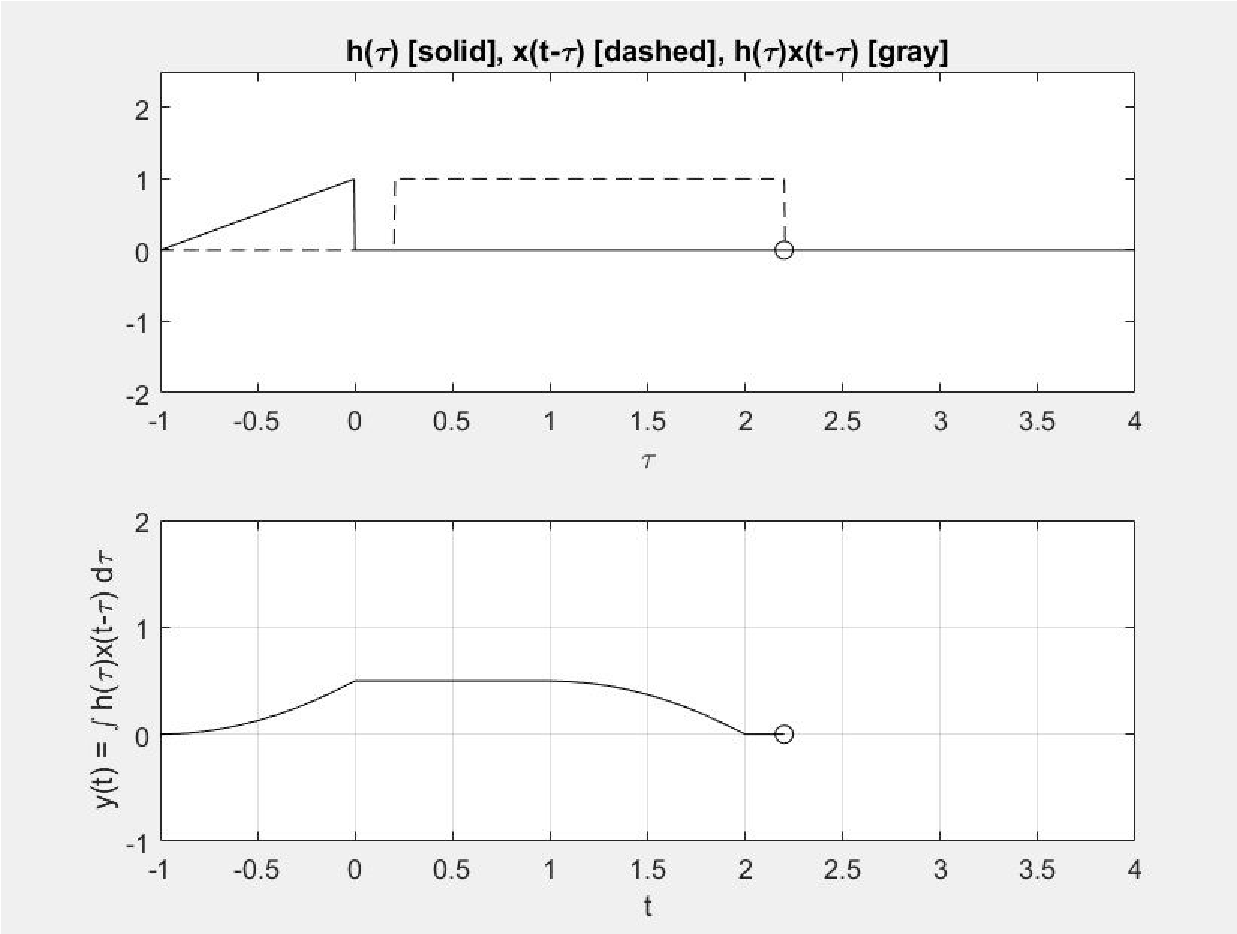
```
% CH2MP4.m : Chapter 2, MATLAB Program 4
% Script M-file graphically demonstrates the convolution process.figure(1)
% Create figure window and make visible on screen
```

```
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;
      pause;
end
```



B.2:

- $x(t) = u(t) - u(t-2)$,
- $h(t) = (t+1)(u(t+1)-u(t))$,
- $y(t) = x(t) * h(t)$
- * refers to convolution.



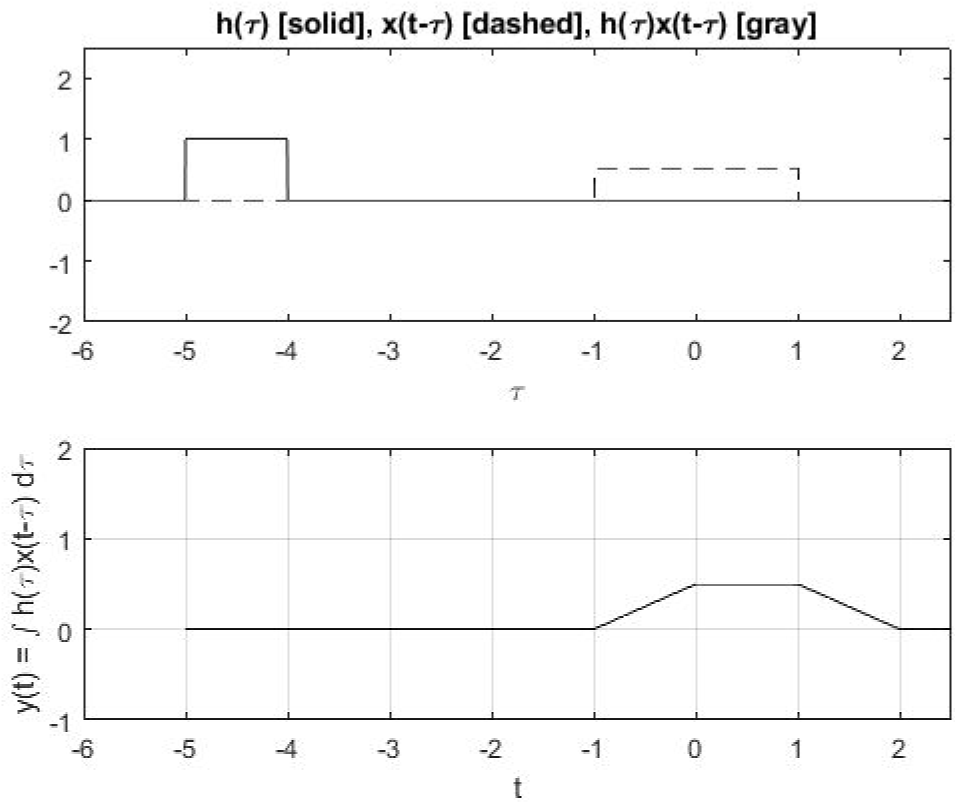
B.3

(a) Assuming A = 0.5 and B = 1

Matlab:

```
u = @(t) 1.0*(t>=0);
A = 0.5; B = 1; % Say A = 0.5 and B = 1
```

```
x = @(t) A*(u(t-4) - u(t-6));
h = @(t) B*(u(t+5) - u(t+4));
dtau = 0.005;
tau = -6:dtau:2.5;ti = 0;
tvec = -5:.1:5; 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;
      pause;
end
```

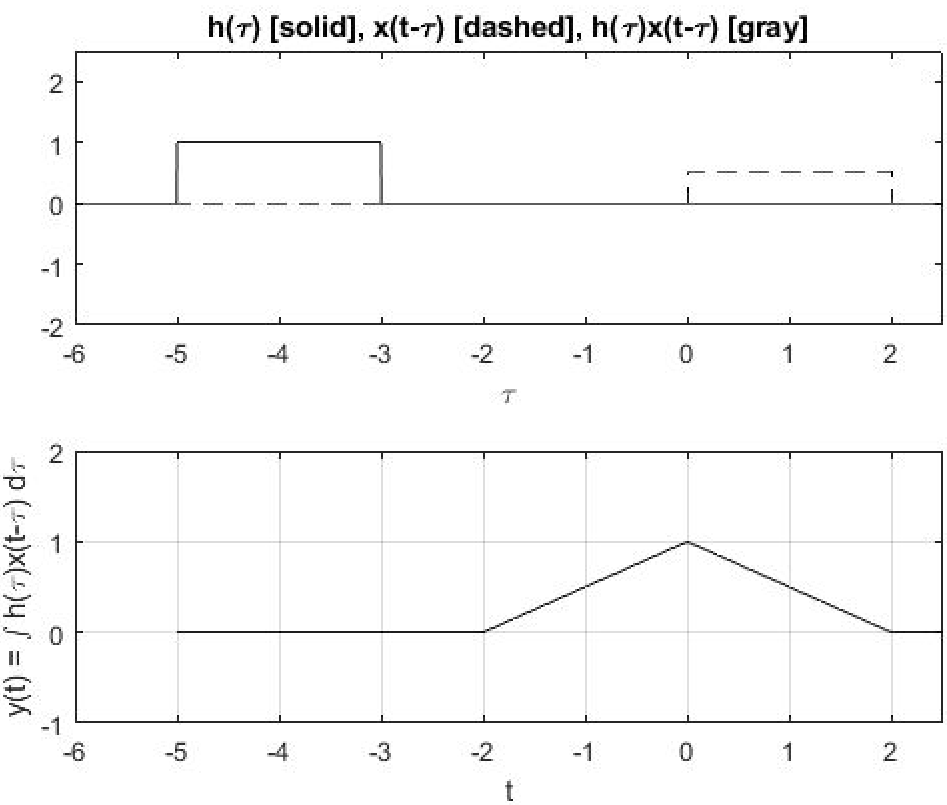


(b) Say A = 0.5 and B = 1

Matlab:

```
u = @(t) 1.0*(t>=0);
A = 0.5; B = 1; % Say A = 0.5 and B = 1
x = @(t) A*(u(t-3) - u(t-5));
h = @(t) B*(u(t+5) - u(t+3));
dtau = 0.005;
tau = -6:dtau:2.5;ti = 0;
tvec = -5:.1:5; 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");
```

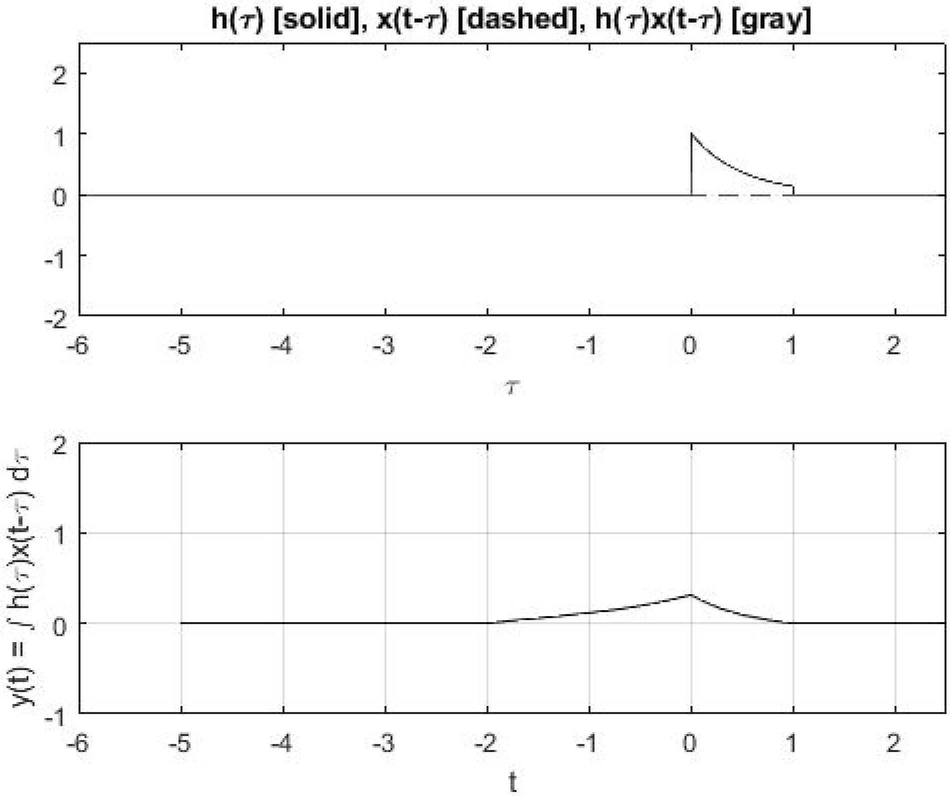
```
        axis([tau(1) tau(end) -2.0 2.5]);
        patch([tau(1:end-1);tau(1:end-1);tau(2:end);tau(2:end)],...
        [zeros(1,lxh-1);xh(1:end-1);xh(2:end);zeros(1,lxh-1)],...
        [.8 .8 .8],"edgecolor","none");
        xlabel("\tau"); title("h(\tau) [solid], x(t-\tau) [dashed],
h(\tau)x(t-\tau) [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
```



(h)

Matlab:

```
u = @(t) 1.0*(t>=0);
x = @(t) exp(t).*(u(t+2) - u(t));
h = @(t) exp(-2*t).*(u(t) - u(t-1));
dtau = 0.005;
tau = -6:dtau:2.5;ti = 0;
tvec = -5:.1:5; 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
```

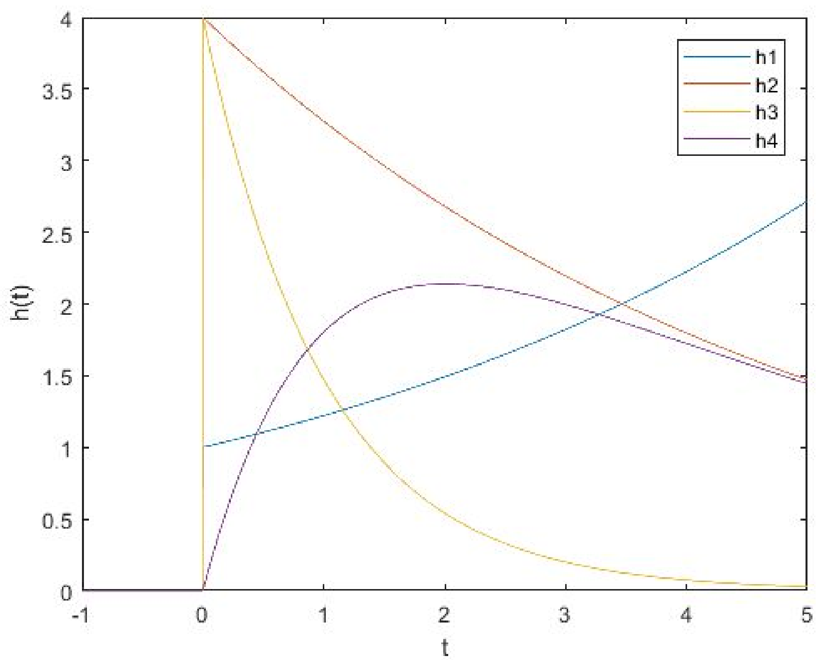


C:
C.1:

Matlab

```
t = [-1:0.001:5];

% Create function
u = @(t) 1.0.* (t>=0);
h1 = @(t) exp(t/5).*u(t);
h2 = @(t) 4*exp(-t/5).*u(t);
h3 = @(t) 4*exp(-t).*u(t);
h4 = @(t) 4*(exp(-t/5) - exp(-t)).*u(t);

plot(t,h1(t));
xlabel("t");
```

```
ylabel("h(t)");
hold on;

plot(t,h2(t));
plot(t,h3(t));
plot(t,h4(t));

legend("h1", "h2", "h3", "h4");
%Plot all the graphs
hold off;
```



C.2:

Characteristic values of the systems S1-S4

S1:

λ1 = ⅕

S2:

λ1 = -⅕

S3:
λ1 = -1

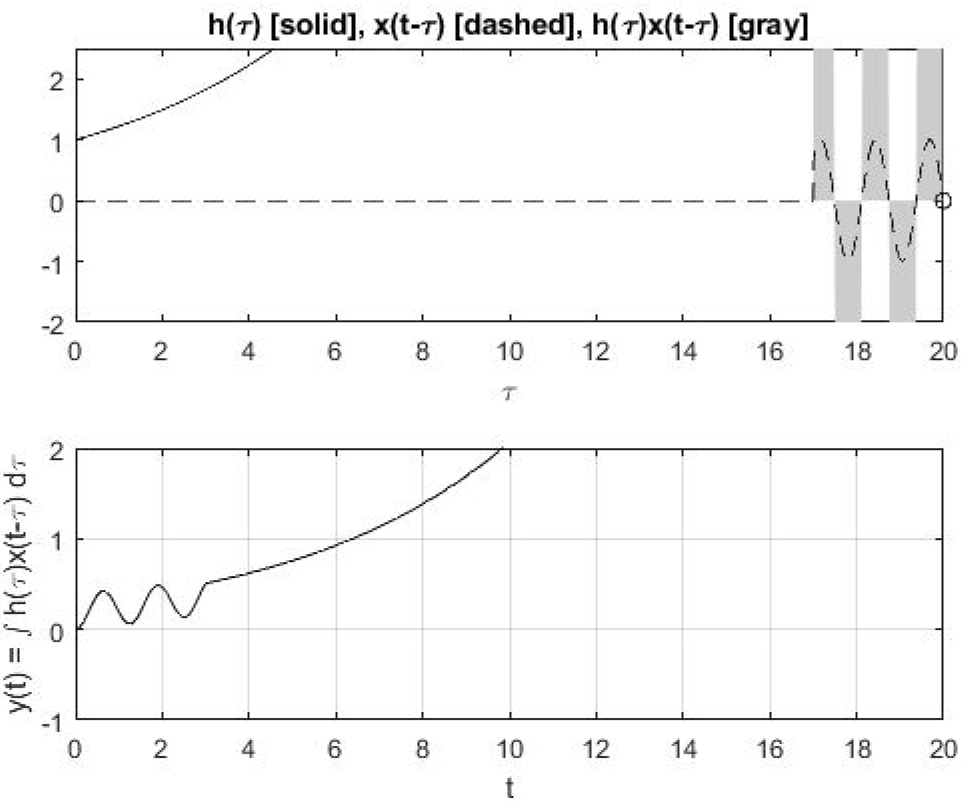
S4:
λ1 = -⅕
λ2 = -1

C.3

For h1:
Matlab:

```
% First create the u(t) function
u = @(t) 1.0.* (t>=0);

% Create the x(t) function.
x = @(t) sin(5*t).*(u(t) - u(t - 3));

% Now truncate each of the impluse response funtions.
h = @(t) exp(t/5).*(u(t)-u(t-20)); % h1

% Modified CH2MP4 from B.1
dtau = 0.005;
tau = 0:dtau:20; ti = 0;
tvec = 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("h(\tau) [solid], x(t-\tau) [dashed],
h(\tau)x(t-\tau) [gray]");
```

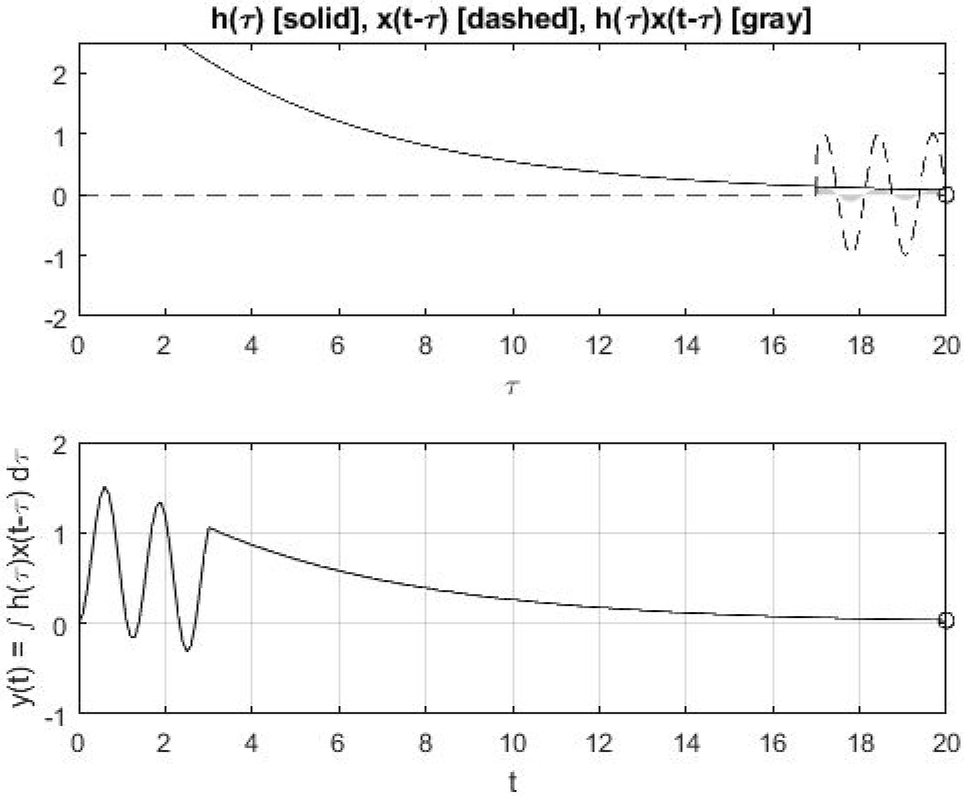
```
        c = get(gca,'children'); set(gca,'children',[c(2);c(3);c(4);c(1)]);
        subplot(2,1,2),plot(tvec,y,"k",tvec(ti),y(ti),"ok");
        xlabel("t"); ylabel("y(t) = \int h(\tau)x(t-\tau) d\tau");
        axis([tau(1) tau(end) -1.0 2.0]); grid;
        drawnow;
end
```



For h2:
Matlab:

```
% First create the u(t) function
u = @(t) 1.0.* (t>=0);

% Create the x(t) function.
```

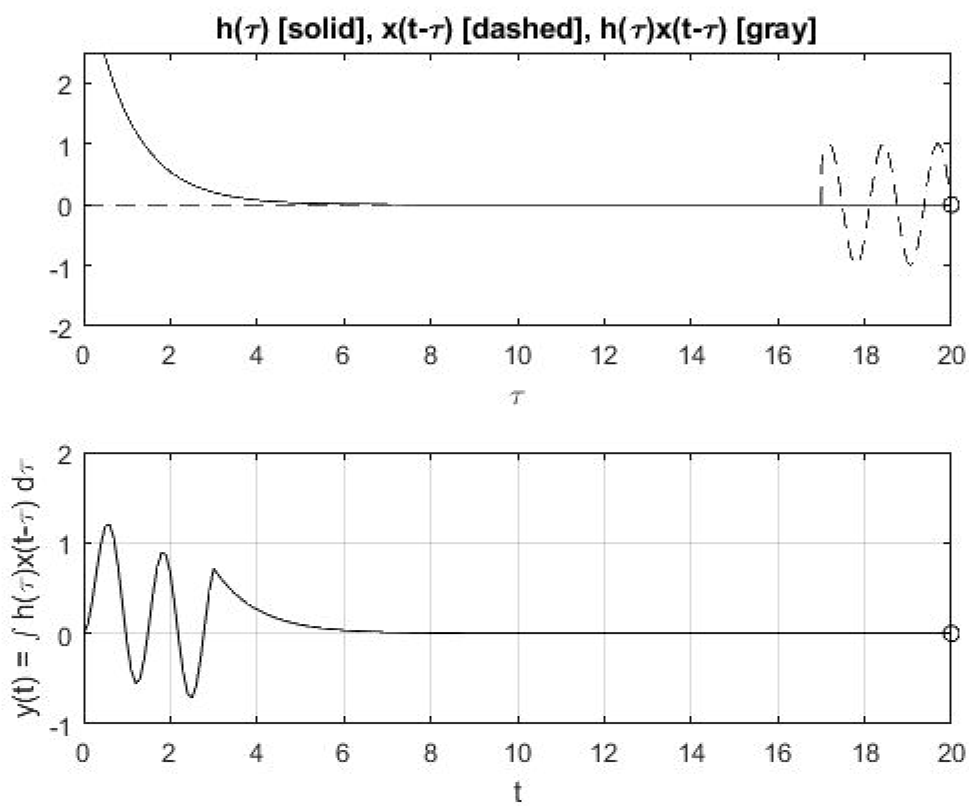
```
x = @(t) sin(5*t).*(u(t) - u(t - 3));

% Now truncate each of the impluse response funtions.
h = @(t) 4*exp(-t/5).*(u(t)-u(t-20)); %h2

% Modified CH2MP4 from B.1
dtau = 0.005;
tau = 0:dtau:20; ti = 0;
tvec = 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("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
```



For h3:
Matlab:

```
% First create the u(t) function
u = @(t) 1.0.* (t>=0);

% Create the x(t) function.
x = @(t) sin(5*t).*(u(t) - u(t - 3));

% Now truncate each of the impluse response funtions.
h = @(t) 4*exp(-t).*(u(t)-u(t-20));%h3

% Modified CH2MP4 from B.1
dtau = 0.005;
tau = 0:dtau:20; ti = 0;
tvec = 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);
```

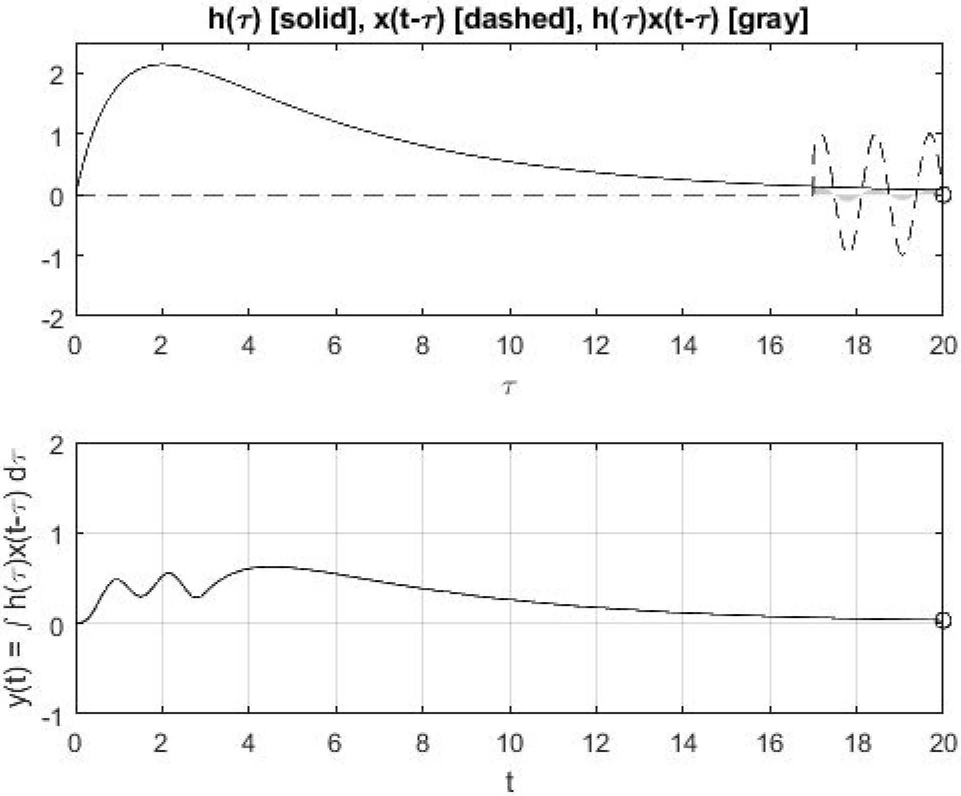
```
      lxh = length(xh);
      y(ti) = sum(xh.*dtau);
      % Trapezoidal approximation of convolution integral
      subplot(2,1,1),plot(tau,h(tau),"k-",tau,x(t-tau),"k--",t,0,"ok");
      axis([tau(1) tau(end) -2.0 2.5]);
      patch([tau(1:end-1);tau(1:end-1);tau(2:end);tau(2:end)],...
      [zeros(1,lxh-1);xh(1:end-1);xh(2:end);zeros(1,lxh-1)],...
      [.8 .8 .8],"edgecolor","none");
      xlabel("\tau"); title("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
```



For h4:

Matlab:

```
% First create the u(t) function
u = @(t) 1.0.* (t>=0);

% Create the x(t) function.
x = @(t) sin(5*t).*(u(t) - u(t - 3));

% Now truncate each of the impluse response funtions.
h = @(t) 4*(exp(-t/5)-exp(-t)).*(u(t)-u(t-20)); %h4

% Modified CH2MP4 from B.1
dtau = 0.005;
tau = 0:dtau:20; ti = 0;
tvec = 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("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
```

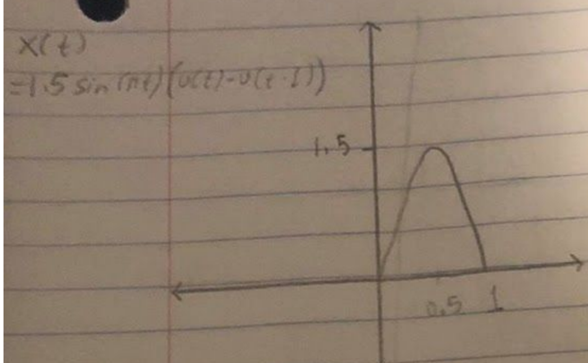


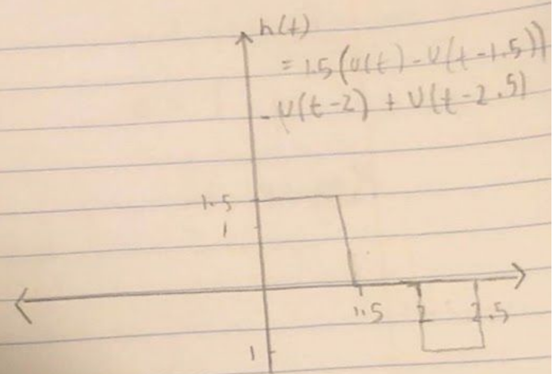
I observed that all the convolutions has parts of them that were similar to the sin(t) function. I also observed that the duration of the signal resulting from the convolution of two signals is that the duration of the convolution is equal to the sum of the duration of the functions. There is a relationship between S2, S3, and S4. S2 and S3 have about the same convolution. S4 has a similar convolution to S2 and S3.

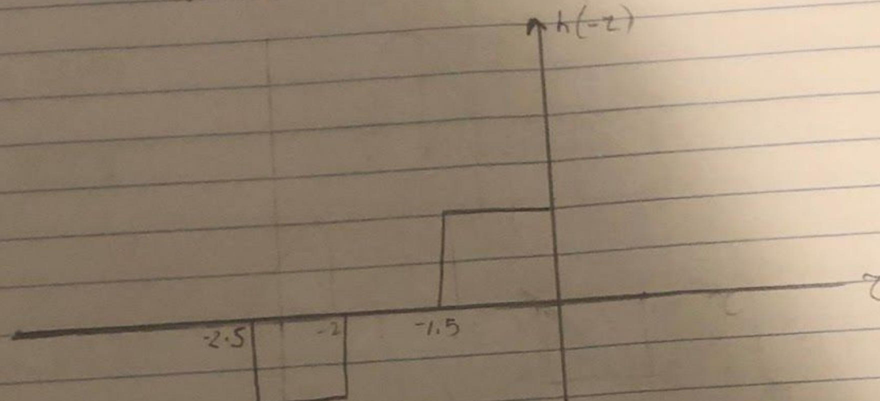
D:

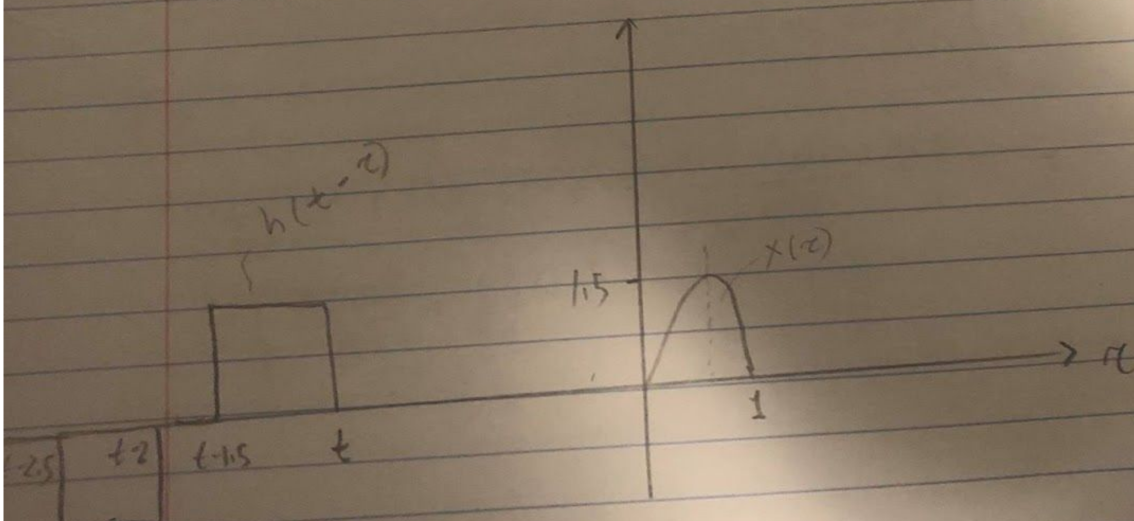
<u>B.1 manually:</u>

D) B.1 manually





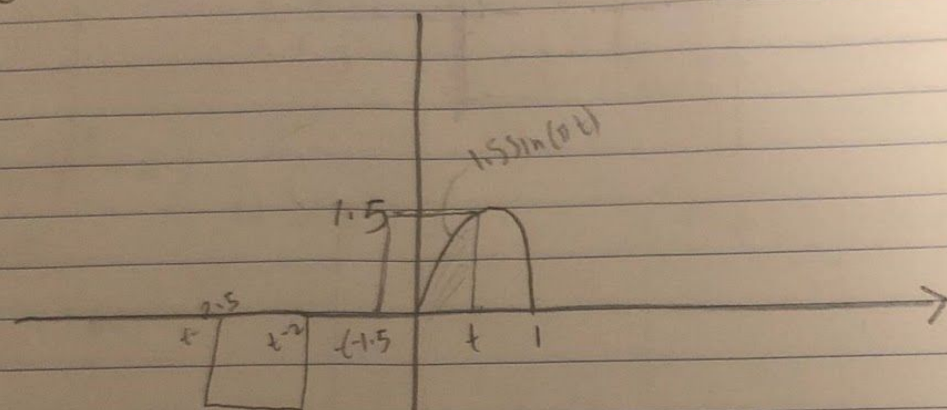




Region 1

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

Region 2



$0 \le t \le 1$

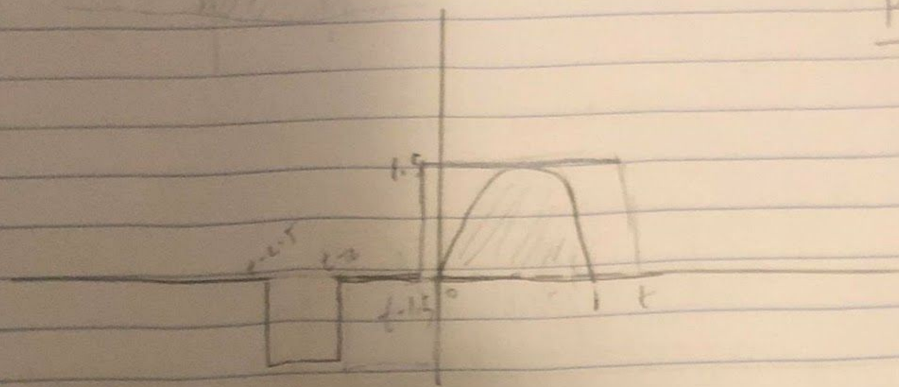
$$y(t) = \int_0^t (1.5)(1.5 \sin \pi t)\, dt$$

$$= \int_0^t 2.25 \sin(\pi t)\, dt$$

$$= -\frac{2.25 \cos(\pi t)}{\pi}\Bigg|_0^t = -\frac{2.25 \cos(\pi t)}{\pi} + \frac{2.25 \cos(0)}{\pi}$$

$$= 2.25\left(\frac{1}{\pi} - \frac{\cos(\pi t)}{\pi}\right)$$

Region 3



$t - 1.5 < 0$

$t \geq 1$

$$= \int_0^1 (1.5)(1.5 \sin \pi t)\, dt$$

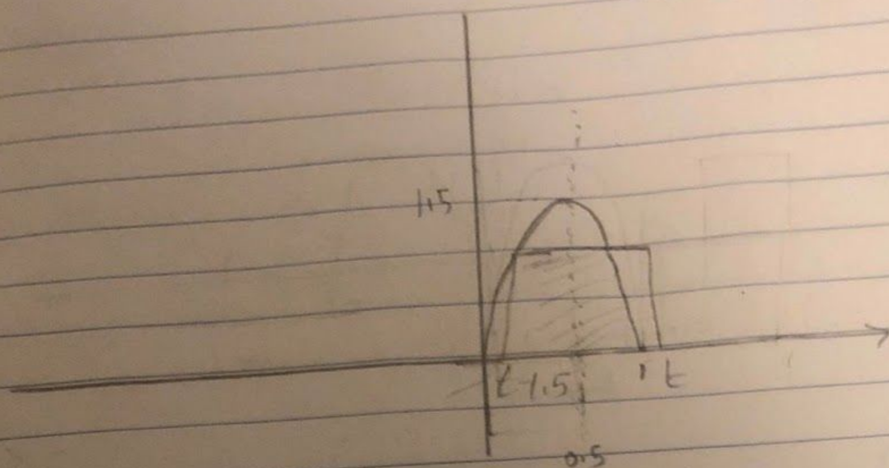
$\boxed{1 \leq t < 1.5}$

$$= -2.25 \frac{\cos(\pi t)}{\pi} \Big|_0^1$$

$$= -2.25 \frac{\cos(\pi)}{\pi} + \frac{2.25 \cos(0)}{\pi}$$

$$= \frac{2.25}{\pi} + \frac{2.25}{\pi} = \boxed{1.432}$$

Region 4



$t-1.5 \leq 0.5$

$\boxed{t \leq 2}$

$t-1.5 \geq 0$

$\boxed{1.5 \leq t \leq 2}$

$$\int_{t-1.5}^{1} (1.5)(1.5\sin \pi t)\ dt$$

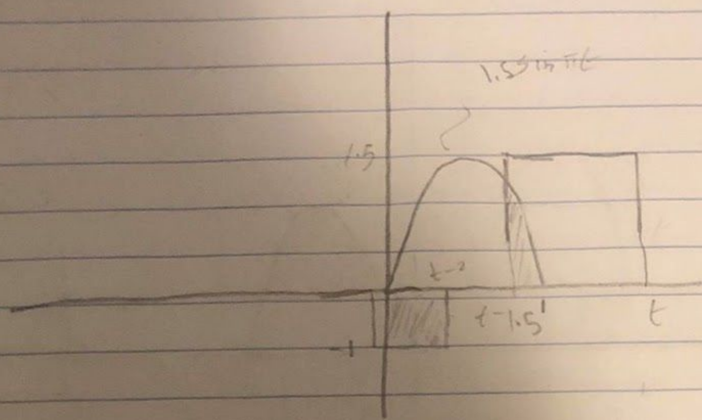
$$\int_{t-1.5}^{1} 2.25\sin(\pi t)\ dt$$

$$\left. \frac{-2.25\cos \pi t}{\pi} \right|_{t-1.5}^{1}$$

$$= \frac{-2.25(\cos \pi)}{\pi} + \frac{2.25\cos(\pi(t-1.5))}{\pi}$$

$$= \frac{2.25}{\pi}\left(1 + \cos(\pi(t-1.5)) \right)$$

Region 5



$t-2 \geq 0$

$t-1.5 \leq 1$

$$\int_0^{t-2} (-1)(1.5\sin\pi t)\,dt + \int_{t-1.5}^{1} (1.5)(1.5\sin\pi t)\,dt$$

$\boxed{2 \leq t \leq 2.5}$

$$\int_0^{t-2} -1.5\sin(\pi t)\,dt + \int_{t-1.5}^{1} 2.25\sin(\pi t)\,dt$$

$$-1.5\int_0^{t-2} \sin(\pi(t))\,dt$$

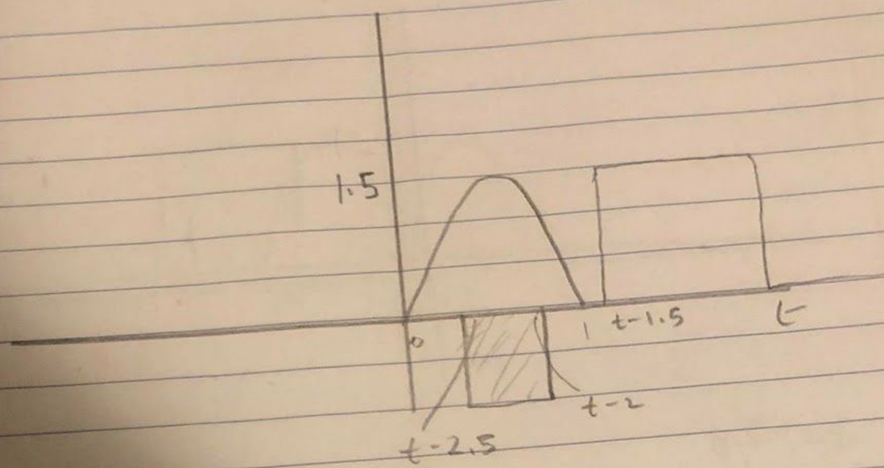
$$\left.\frac{1.5\cos(\pi(t))}{\pi}\right|_0^{t-2}$$

from last part

$$\frac{1.5\cos(\pi(t-2))}{\pi} - \frac{1.5\cos(0)}{\pi}$$

$$= \frac{1.5}{\pi}\left(\cos(\pi(t-2)) - 1\right) + \frac{2.25}{\pi}\left(1 + \cos(\pi(t-1.5))\right)$$

Region 6



$t-2 \leq 1$

$\boxed{t \leq 3}$

$t-2.5 \geq 0$

$\boxed{t \geq 2.5}$

$2.5 \leq t \leq 3$

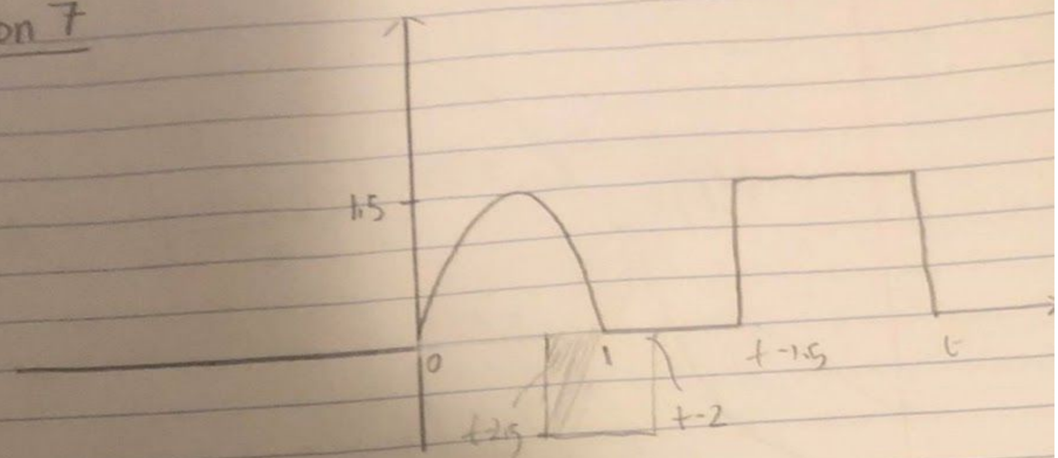
$$\int_{t-2.5}^{t-2} (1)1.5\sin(\pi t)\ dt$$

$$\frac{1.5\cos(\pi t)}{\pi}\Bigg|_{t-2.5}^{t-2}$$

$$= \frac{1.5\cos(\pi(t-2))}{\pi} - \frac{1.5\cos(\pi(t-2.5))}{\pi}$$

$$= \frac{1.5}{\pi}\left(\cos(\pi(t-2)) - \cos(\pi(t-2.5))\right)$$

Region 7



$t-2.5 \leq 1$

$t-2 \geq 1$

$\boxed{3 \leq t \leq 3.5}$

$$\int_{t-2.5}^{1} (-1)\, 1.5 \sin(\pi(t))\, dt$$

$$= \frac{1.5 \cos(\pi(t))}{\pi} \Bigg|_{t-2.5}^{1}$$

$$= \frac{1.5}{\pi}\left(-1 \quad - \cos(\pi(t-2.5)) \right)$$

Region 8

no Overlap $\Longrightarrow$ $y(t) = 0$

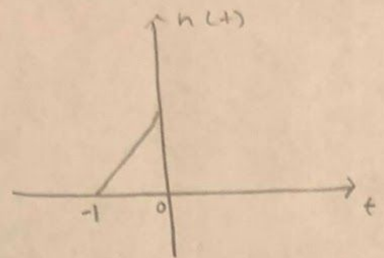
$$
y(t) = \begin{cases}
0 & t \le 0 \\
2.25\left(\frac{1}{\pi} - \frac{\cos(\pi t)}{\pi}\right) & 0 \le t \le 1 \\
1.432 & 1 \le t \le 1.5 \\
\frac{2.25}{\pi}\left(1 + \cos(\pi(t-1.5))\right) & 1.5 \le t \le 2 \\
\frac{1.5}{\pi}\left(\cos(\pi(t-2)) - 1\right) + \frac{2.25}{\pi}\left(1 + \cos(\pi(t-1.5))\right) & 2 \le t \le 2. \\
\frac{1.5}{\pi}\left(\cos\pi(t-2)) - \cos(\pi(t-2.5))\right) & 2.5 \le t \le 3 \\
\frac{1.5}{\pi}\left(-1 - \cos(\pi(t-2.5))\right) & 3 \le t \le 3.5 \\
0 & t \geq 3.5
\end{cases}
$$

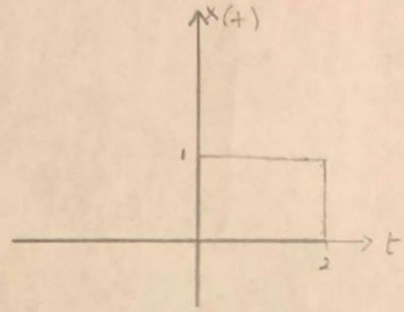
B.2 manually:

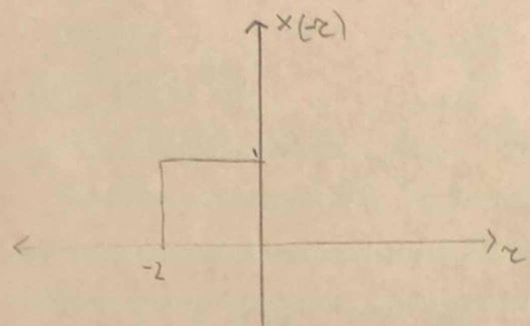
D) B.2 manually

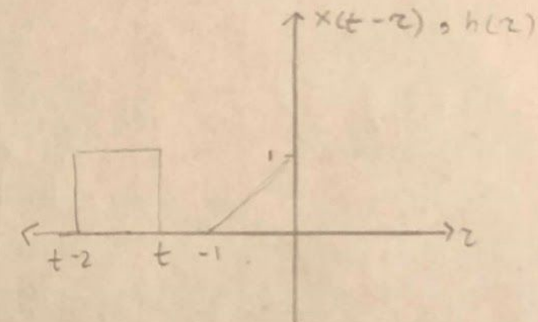
$x(t) = u(t) - u(t-2)$

$h(t) = (t+1)\big(u(t+1) - u(t)\big)$



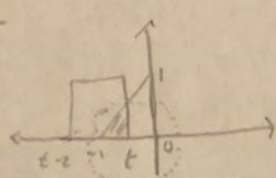






Region 1 no overlap $y(t) = 0$, $t \leq -1$

Region2



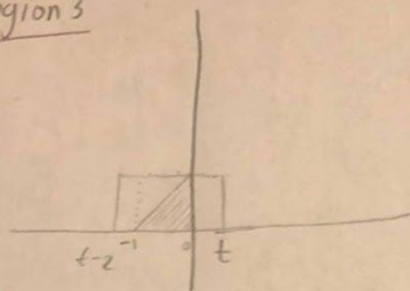
$-1 \leq t < 0$

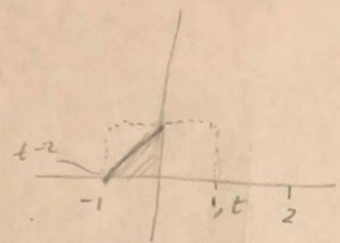
$$\int_{-1}^{t} (t+1)\, dt$$

$$\frac{t^2}{2} + t \,\Big|_{-1}^{t} = t\left(\frac{t}{2} + 1\right)\Big|_{-1}^{t}$$

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

Region 3





$t \leq 1$

$t \geq 0$

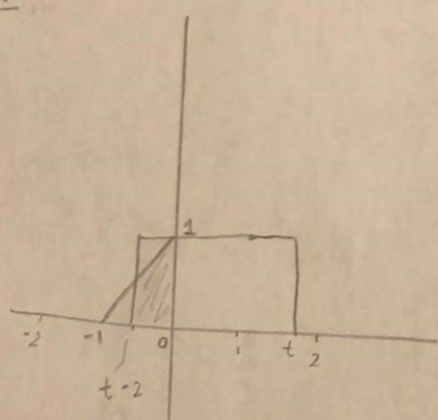
$\boxed{0 \leq t \leq 1}$

$$\int_{-1}^{0} (t+1) \, dt$$

$$\frac{t^2}{2} + t \Big|_{-1}^{0}$$

$$\boxed{\frac{1}{2} - 1 = -0.5}$$

Region 4



$1 \leq t \leq 2$

$$\int_{t-2}^{0} (1)(t+1) = \frac{t^2}{2} + t \Big|_{t-2}^{0}$$

$$= -\left[\frac{(t-2)^2}{2} + t-2\right]$$

Region 5

$y(t) = 0$ (no overlap)

$t \geq 2$

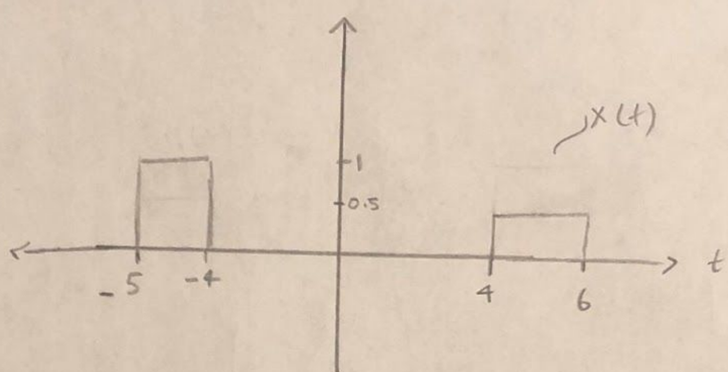
$$y(t) = \begin{cases} 0 & t \leq -1 \\ \dfrac{t^2}{2} + t + \dfrac{1}{2} & -1 \leq t \leq 0 \\ 0.5 & 0 \leq t \leq 1 \\ -\left[\dfrac{(t-2)^2}{2} + t-2\right] & 1 \leq t \leq 2 \\ 0 & t \geq 2 \end{cases}$$

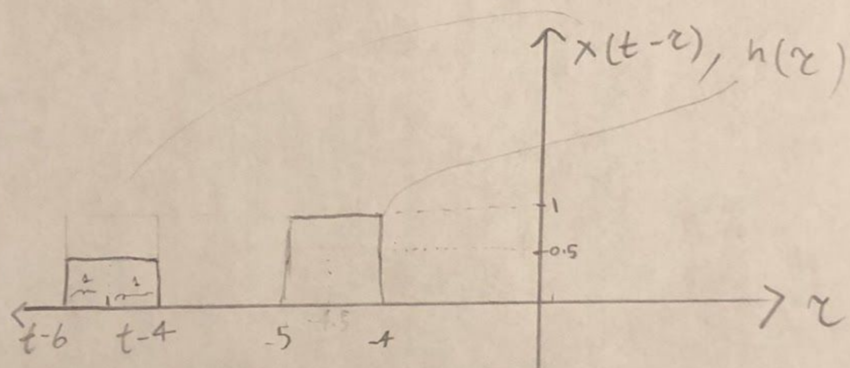
B.3 manually:

D) B.3

$x(t) = \frac{1}{2}(t-4) - u(t-6)$

$h(t) = (u(t+5) - u(t+4))$





Region 1

$t-4 \le -5$ $\quad y(t) = 0$

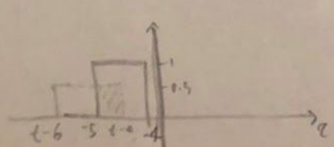
$t \le -1$ $\quad$ (no overlap

Region 2

$-5 < t-4 \le -4$

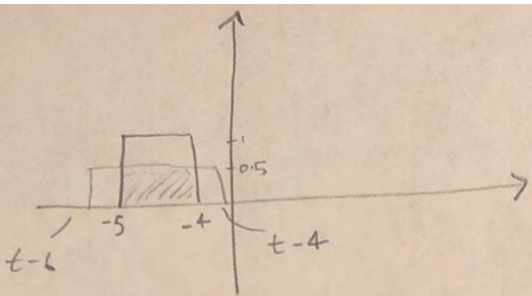
$-1 < t \le 0$

$$\int_{-5}^{t-4} 0.5 \, dt = 0.5t \Big|_{-5}^{t-4}$$



$= 0.5(t-4) - 0.5(-5)$

$= \frac{1}{2}t - 2 + 2.5 = \frac{1}{2}t + 0.5$



Region 3

$t-4 \geq -4$
$t-6 \leq -5$

$0 \leq t \leq 1$

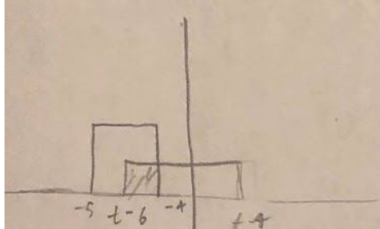
$$\int_{-5}^{-4} 0.5 \, dt$$

$$0.5t \Big|_{-5}^{-4}$$

$$0.5(-4) - 0.5(-5)$$

$$= 0.5$$

Region 4



$1 \leq t \leq 2$

$$\int_{t-6}^{-4} 0.5 \, dt$$

$$= 0.5t \Big|_{t-6}^{-4}$$

$$= 0.5(-4) - 0.5(t-6)$$

$$= -2 - 0.5t + 3 = -0.5t + 1$$

overall $y(t)$

$$y(t) = \begin{cases} 0 & t \leq -1 \\ \frac{1}{2}t + 0.5 & -1 \leq t \leq 0 \\ 0.5 & 0 \leq t \leq 1 \\ -\frac{1}{2}t + 1 & 1 \leq t \leq 2 \end{cases}$$

Overall convolution for all regions performed by hand is similar to those performed using Matlab, same results were obtained.

D2: The observations that can be made about the width/duration of the signal resulting from the convolution of two signals is that the duration of the convolution is equal to the sum of the duration of the functions.