E/15/065

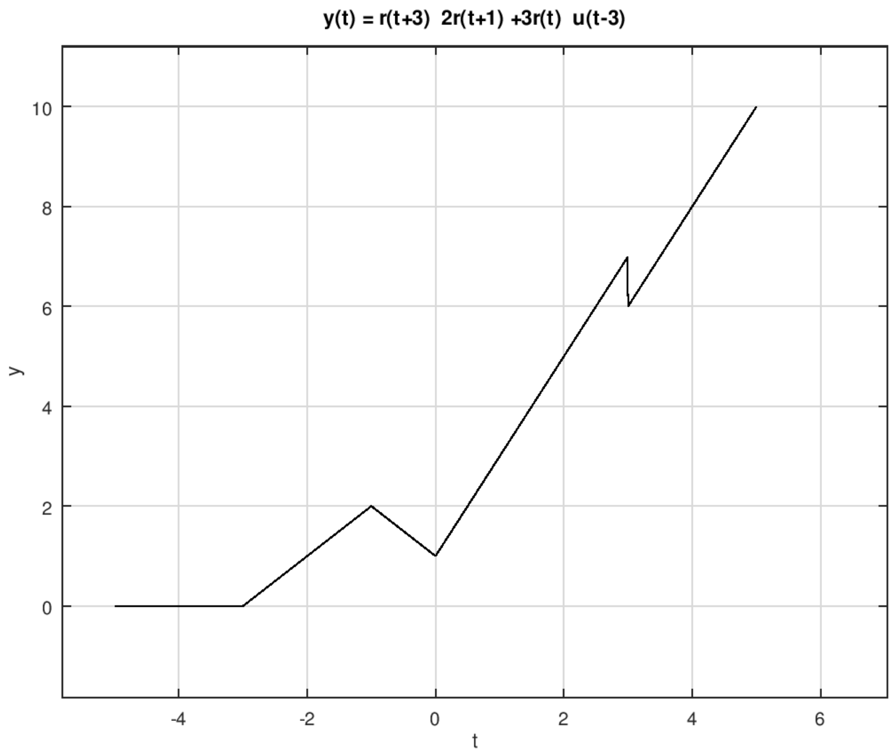
Lab 1: Basic Signal Representation and Convolution in MATLAB

PART 1: Basic Signal Representation in MATLAB

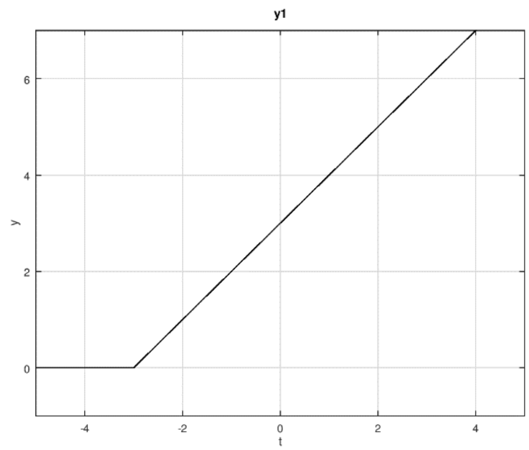
1) Write a Matlab program and necessary functions to generate the following signal:
   y(t) = r(t+3) – 2r(t+1) +3r(t) – u(t-3)

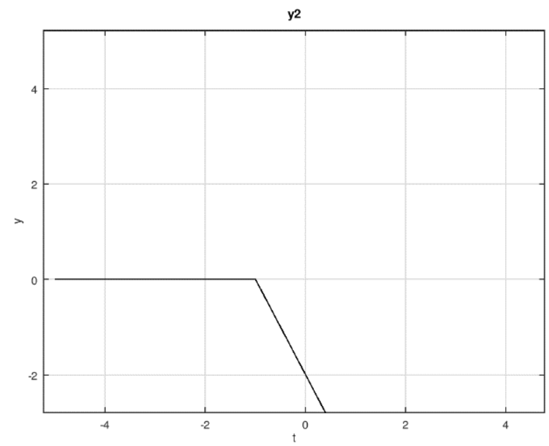
```
%In the given code there are some conflicts. So I modified the code.
clear all;
Ts=0.01;
 t= -5:Ts:5;
function y=ramp(t,m,ad)
  n=length(t);
  y=zeros(1,n);
  for i =1:n
    if(t(i)>=-ad)
      y(i)=m*(t(i)+ad);
    end
   end
end
function y = ustep(t,ad)
  y = zeros(size(t));
  y(t>=0-ad) = 1;
end
y1 = ramp(t,1,3);
y2 = ramp(t,-2,1);
y3 = ramp(t,3,0);
y4 = (-1).*ustep(t,-3);
```

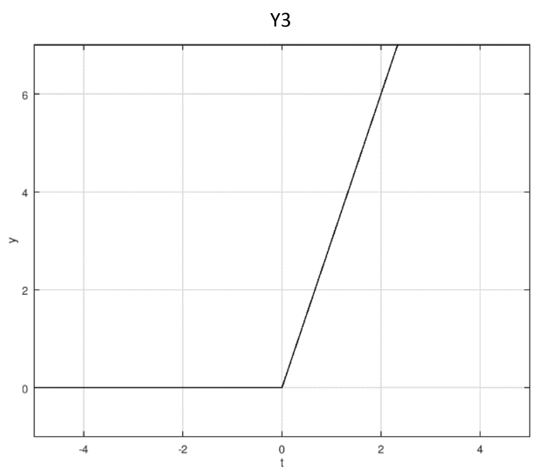
```
y = y1+y2+y3+y4;

plot(t,y,'k');

axis([-5 5 -1 7]); xlabel('t');ylabel('y');

title('y(t) = r(t+3) – 2r(t+1) +3r(t) – u(t-3)')
```

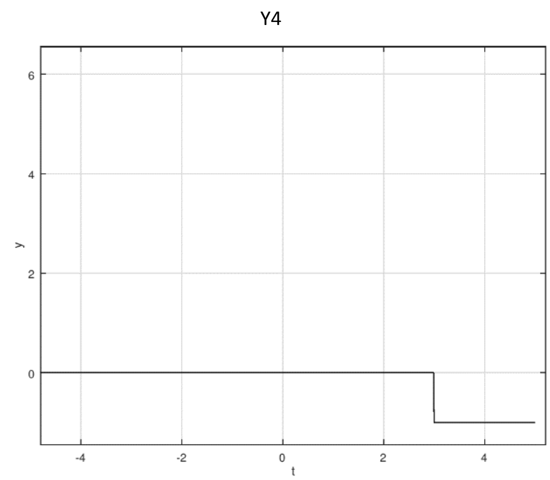


- Y(t)=0 for t<-3 and t>3, therefore -5<t<5 shows the signal well.
- For -3 <= t <= -1, y(t) is r(t+3)=(t+3) which is 0 at t= -3 and 2 at t=-1
- For -1 <= t <= 0, y(t) is r(t+3) – 2r(t+1) =(t+3) – 2(t+1) = -t+1 which is 2 at t=-1 and 1 t=0
- For 0 <= t <=3, r(t+3) – 2r(t+1) +3r(t) = -t+1 + 3(t) =2t+1 which is 1 at t=0 and 7 at t=3
- For t >= 3, r(t+3) – 2r(t+1) +3r(t) – u(t-3) = 2t+1 –1 = 2t = 6


y1
y
t
6
4
2
0
-4
-2
0
2
4

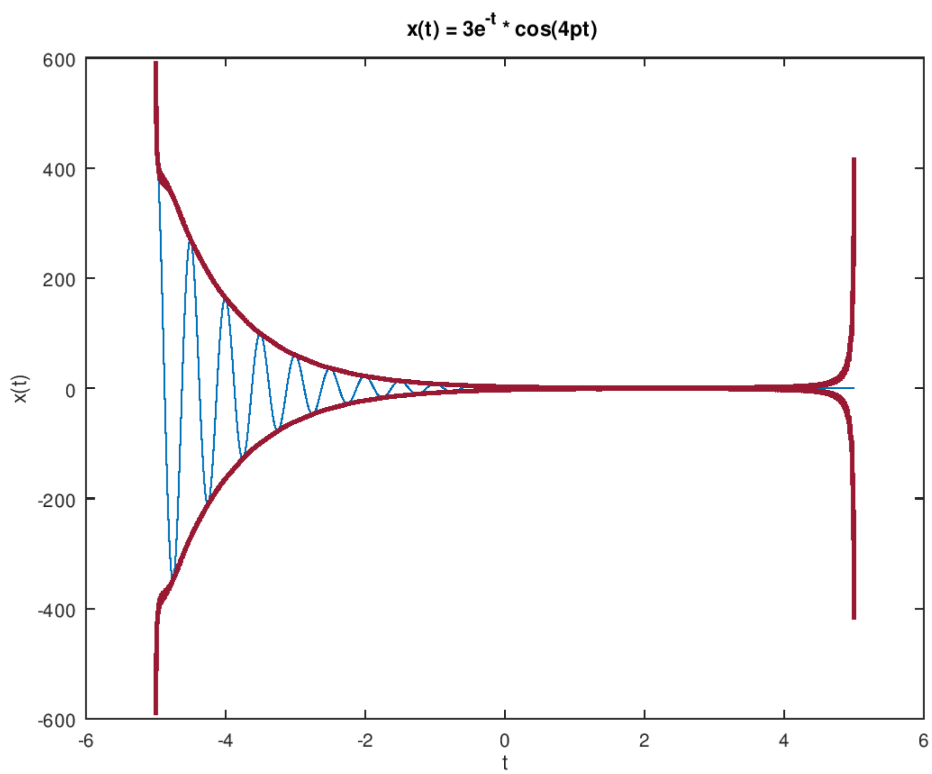

y2
y
t
4
2
0
-2
-4
-2
0
2
4


Y3
y
t
6
4
2
0
-4
-2
0
2
4


Y4
y
t
6
4
2
0
-4
-2
0
2
4

2) For the damped sinusoidal signal x(t) = 3e -t cos(4πt) write a MATLAB program to generate x(t) and its envelope, then plot.

```
Ts=0.01;

t= -5:Ts:5;

x=3.*exp(-t) .* cos(4*pi*t);

temp = hilbert(x);

env=abs(temp);


plot_param = {'Color', [0.6 0.1 0.2],'Linewidth',2}

plot(t,x);

hold on

plot(t,[-1;1]*env,plot_param{:});

xlabel('t');ylabel('x(t)');title('x(t) = 3e^-^t * cos(4πt) ');

hold off
```



PART 2: Time-Domain Convolution

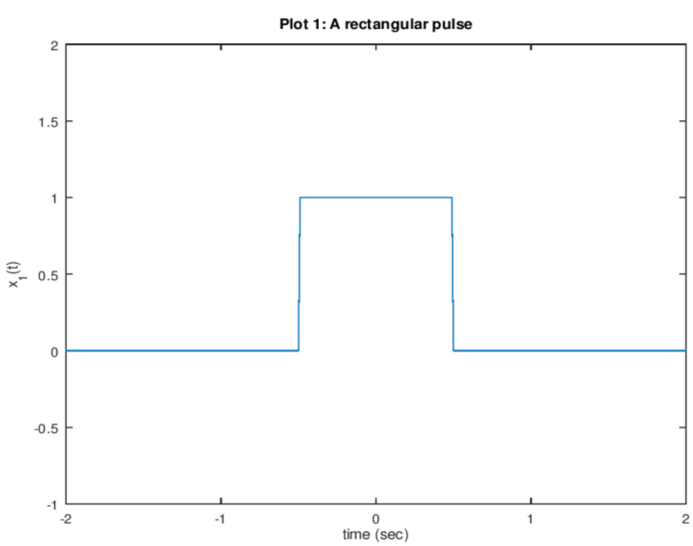
Creating a rectangular pulse in MATLAB

Are there any disadvantages if a high sampling frequency is used?

Yes, there may be some disadvantages,
Then t = -5:T_s:5 t will be very large, so it takes a lot of space(memory) and takes more time to calculate the values.

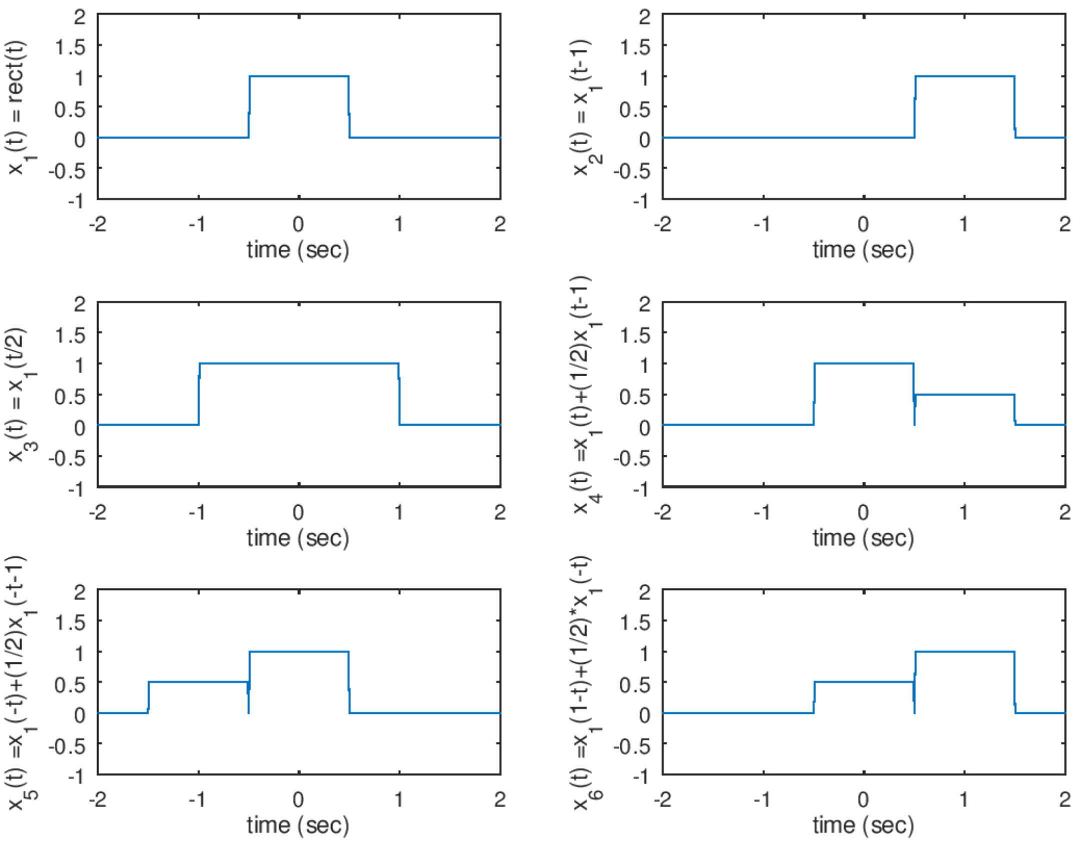
```
function x= rect(t=0)

  x=ones(1,numel(t)).*(abs(t)<1/2) % a is the width of the pulse

end
```

```
clear all

f_s=100;

T_s = 1/f_s;

t = -5:T_s:5;

x1=rect(t);

plot(t,x1);

axis( [-2 2 -1 2]); %resize the axis

xlabel('time (sec)');

ylabel('x_1(t)')

title ('Plot 1: A rectangular pulse')
```



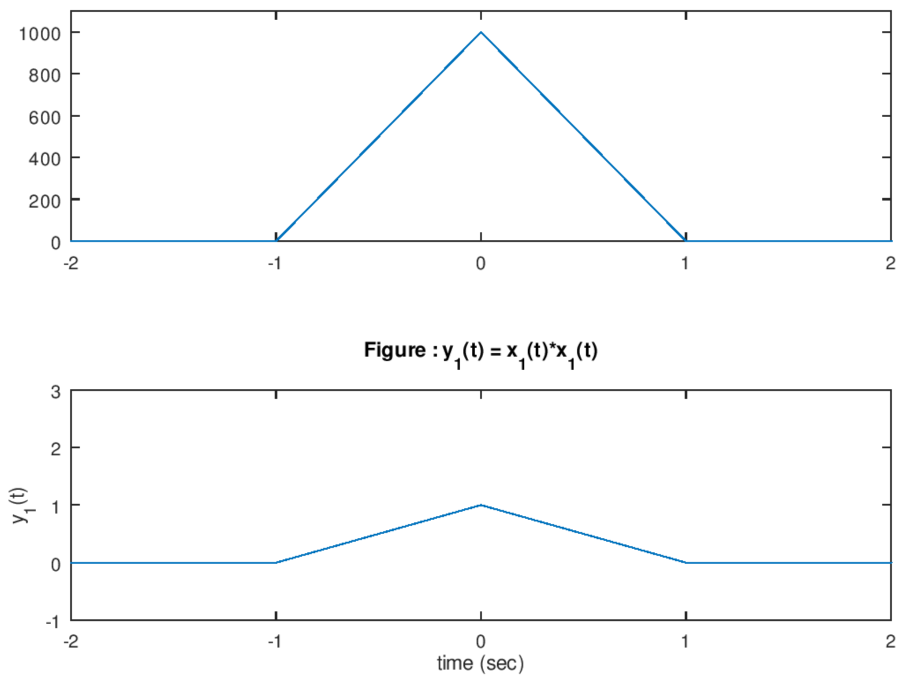
Elementary signal operations

```
clear all
f_s=100;
T_s = 1/f_s;
t = -5:T_s:5;
x1=rect(t);
x2 = rect(t-1); %time-delay
x3 = rect(t/2); %time-scaling
x4=rect(t)+(1/2).*rect(t-1);
x5=rect(-t)+(1/2)*rect(-t-1);
x6=rect(1-t)+(1/2)*rect(-t);
subplot(3,2,1)
plot(t,x1) ;
axis( [-2 2 -1 2]); xlabel( 'time (sec)' );ylabel('x_1(t) = rect(t)');
subplot(3,2,2)
plot(t,x2); axis( [-2 2 -1 2]); xlabel( 'time (sec)' );ylabel('x_2(t) = x_1(t-1)');

subplot(3,2,3)
plot(t,x3); axis( [-2 2 -1 2]); xlabel( 'time (sec)' );ylabel('x_3(t) = x_1(t/2)');

subplot(3,2,4)
plot(t,x4); axis( [-2 2 -1 2]); xlabel( 'time (sec)' );ylabel('x_4(t) =x_1(t)+(1/2)x_1(t-1)');

subplot(3,2,5)
plot(t,x5);axis( [-2 2 -1 2]);xlabel( 'time (sec)' );ylabel('x_5(t) =x_1(-t)+(1/2)x_1(-t-1)');

subplot(3,2,6)
plot(t,x6);axis( [-2 2 -1 2]);xlabel( 'time (sec)' );ylabel('x_6(t) =x_1(1-t)+(1/2)*x_1(-t)')
```


x_1(t) = rect(t)
x_2(t) = x_1(t-1)
x_3(t) = x_1(t/2)
x_4(t) =x_1(t)+(1/2)x_1(t-1)
x_5(t) =x_1(-t)+(1/2)x_1(-t-1)
x_6(t) =x_1(1-t)+(1/2)*x_1(-t)
time (sec)

Convolution

```
f_s=1000

T_s = 1/f_s;

t = -5:T_s:5


x1=rect(t);

y = conv(x1,x1);

t_y = -10:T_s:10;


close all

subplot(2,1,1)

plot( t_y, y);

axis( [-2 2 -1 2]);


y1 = T_s*conv(x1,x1);

subplot(2,1,2)

plot(t_y, y1);

axis( [-2 2 -1 2] ) ;xlabel( 'time (sec)');ylabel('y_1(t)');

title('Figure : y_1(t) = x_1(t)*x_1(t)');
```

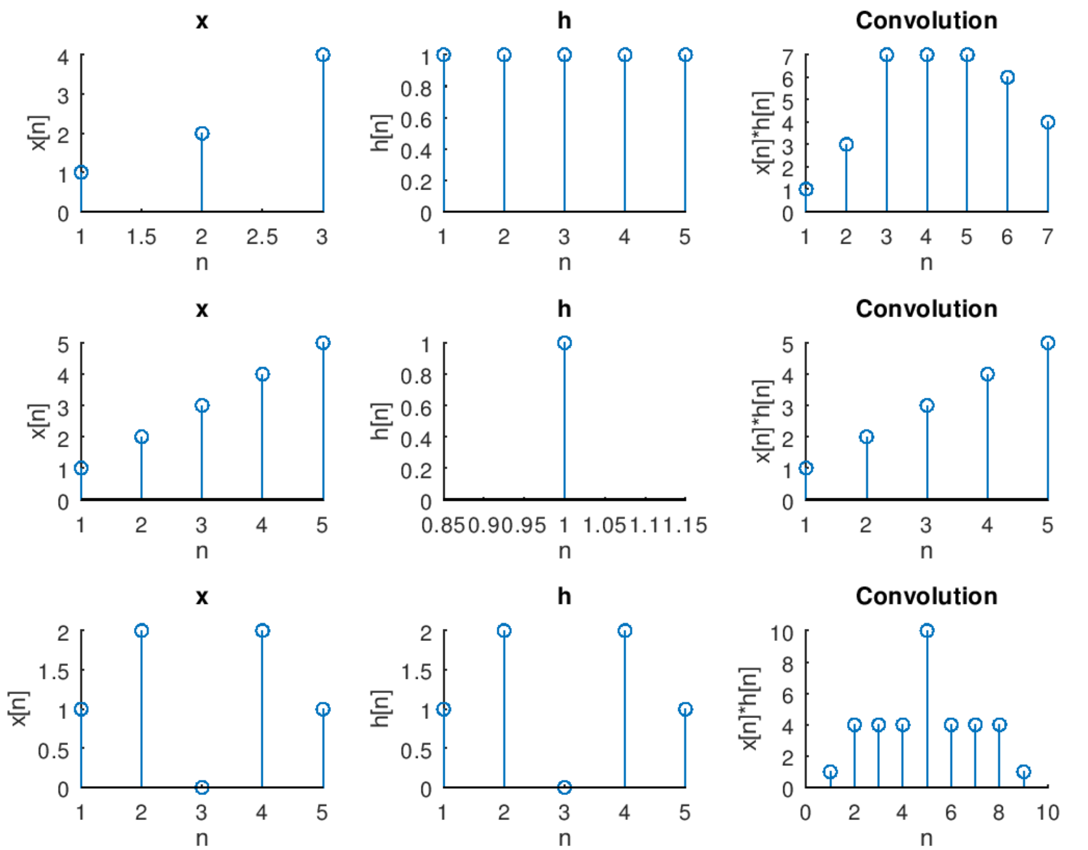


Exercise

1) Perform convolution on discrete time signals x(n) and h(n), i.e., y(n) = x(n)*h(n) using MATLAB. For each set of signals, plot x(n), h(n) and y(n) as subplots in the same figure
   - x(n) = { 1,2,4 }, h(n) = {1,1,1,1,1}
   - x(n) = { 1,2,3,4,5 }, h(n) = {1}
   - x(n) = h(n) ={ 1,2,0,2,1}

```
x1=[1,2,4];

h1=[1 1 1 1 1];

y1=conv(x1,h1);


x2=[1 2 3 4 5];

h2=[1];

y2=conv(x2,h2);


x3=[1 2 0 2 1];

h3=[1 2 0 2 1];

y3=conv(x3,h3);
```

```
% x(n) = { 1,2,4 }, h(n) = {1,1,1,1,1}
subplot(3,1,1)
stem(x1);title('x');xlabel('n');ylabel('x[n]')
subplot(3,1,2)
stem(h1);title('h');xlabel('n');ylabel('h[n]')
subplot(3,1,3)
stem(y1);title('Convolution');;xlabel('n');ylabel('x[n]*h[n]')

% x(n) = { 1,2,3,4,5 }, h(n) = {1}
subplot(3,3,4)
stem(x2);title('x');xlabel('n');ylabel('x[n]')
subplot(3,3,5)
stem(h2);title('h');xlabel('n');ylabel('h[n]')
subplot(3,3,6)
stem(y2);title('Convolution');;xlabel('n');ylabel('x[n]*h[n]')

%x(n) = h(n) ={ 1,2,0,2,1}
subplot(3,3,7)
stem(x3);title('x');xlabel('n');ylabel('x[n]')
subplot(3,3,8)
stem(h3);title('h');xlabel('n');ylabel('h[n]')
subplot(3,3,9)
stem(y3);title('Convolution');;xlabel('n');ylabel('x[n]*h[n]')
```



2) Assume a system with the following impulse response:

$h(n) = (0.5)n$ for $0 <= n < 4$

$= 0$ elsewhere

Determine the input x (n) that will generate the output sequence y(n) = {1, 2, 2.5, 3, 3, 3, 2, 1,0...}. Plot h(n), y (n) and x (n) in one figure.

```
n=10;

h=zeros(1,n);

i=[0:3];

h(i<4)=(0.5).^(i);

y=[1, 2, 2.5, 3, 3, 3, 2, 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

x=deconv(y,h)

subplot(3,1,1);

stem(y);title('y');;xlabel('n');ylabel('x[n]*h[n]');

subplot(3,1,2);

stem(h);title('h');xlabel('n');ylabel('h[n]');

subplot(3,1,3);

stem(x);title('x');xlabel('n');ylabel('x[n]');
```