M. Enes İnanç

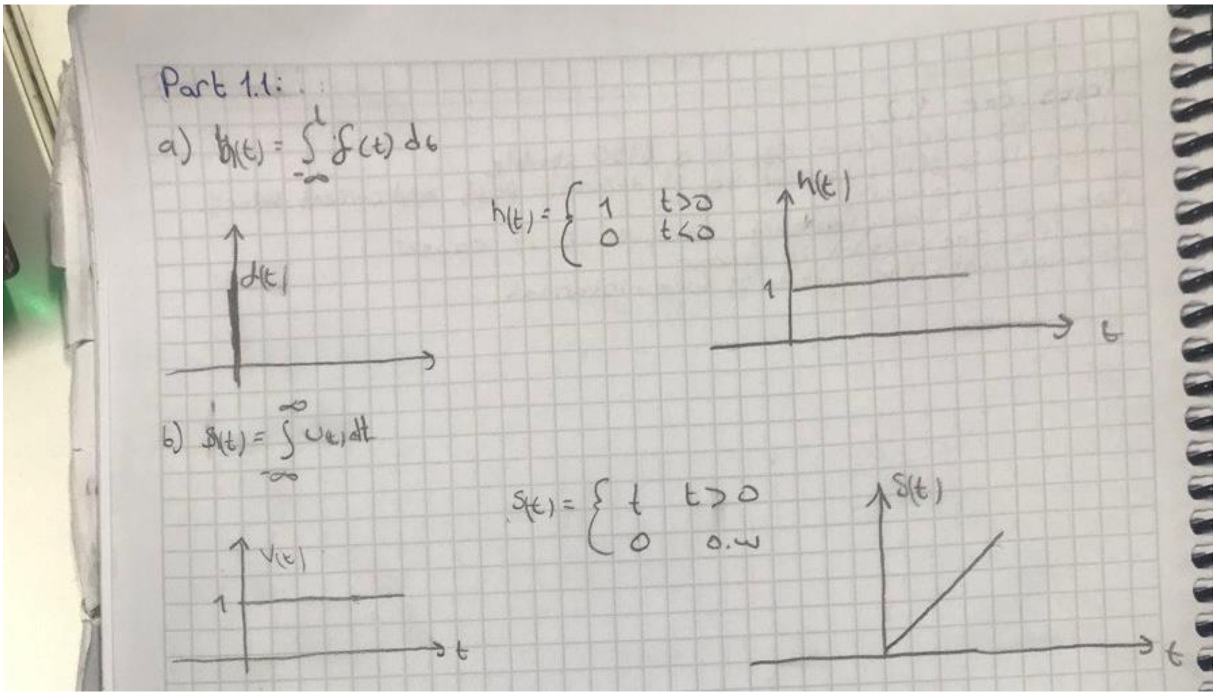
22001799

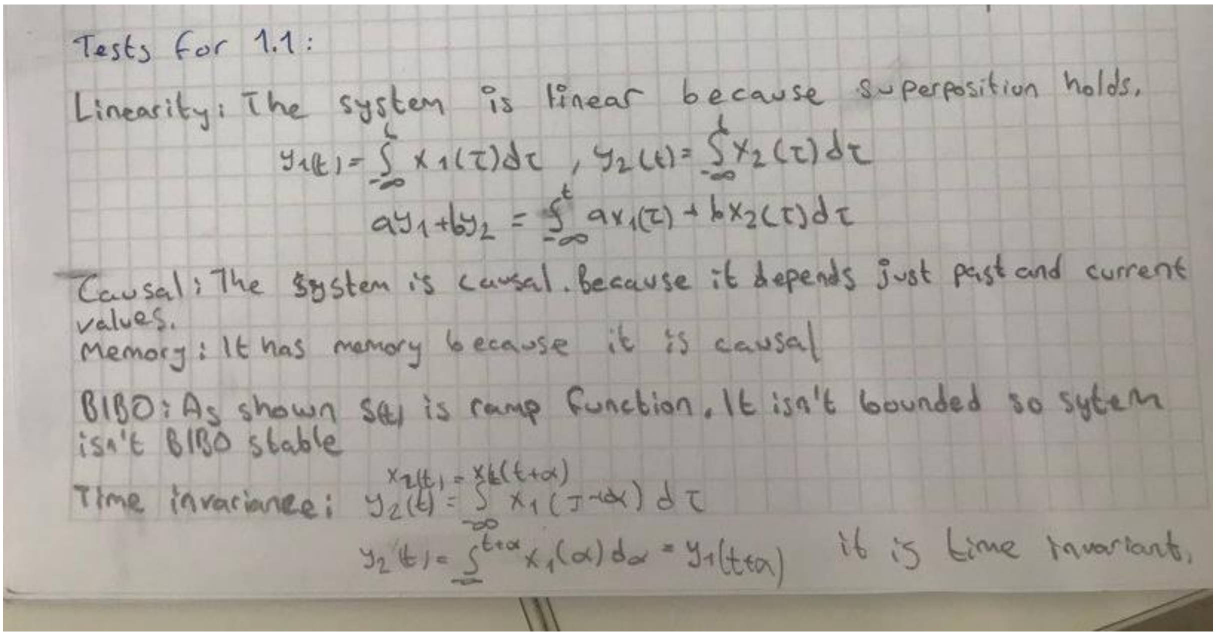
23.03.2024

EEE-321 Lab-3

Part 1:

Part 1.1





Tests for 1.1:

Linearity: The system is linear because superposition holds.

$$y_1(t) = \int_{-\infty}^{t} x_1(\tau)\,d\tau \quad , \quad y_2(t) = \int_{-\infty}^{t} x_2(\tau)\,d\tau$$

$$ay_1 + by_2 = \int_{-\infty}^{t} ax_1(\tau) + bx_2(\tau)\,d\tau$$

Causal: The system is causal. because it depends just past and current values.

Memory: It has memory because it is causal

BIBO: As shown S(t) is ramp function. It isn't bounded so sytem isn't BIBO stable

Time invariance: $x_2(t) = x_1(t+\alpha)$

$$y_2(t) = \int_{-\infty}^{t} x_1(\tau+\alpha)\,d\tau$$

$$y_2(t) = \int_{-\infty}^{t+\alpha} x_1(\alpha)\,d\alpha = y_1(t+\alpha)$$ it is time invariant.

Part 1.2:

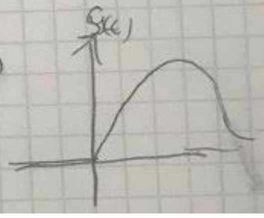
Part 1.2:

$$h(t) = \begin{cases} e^{-at} & t>0 \\ 0 & t<0 \end{cases}$$

$$u(t) * h(t) = g(t)$$

$$g_t = \int_{-\infty}^{\infty} u(\tau)\, h(t-\tau)\, d\tau$$

$$g(t) = \begin{cases} \dfrac{e^{-at}}{-a} & t>0 \\ 0 & t<0 \end{cases}$$



Tests for 1.2 :

BIBO: As seen from g(t) it is BIBO stable

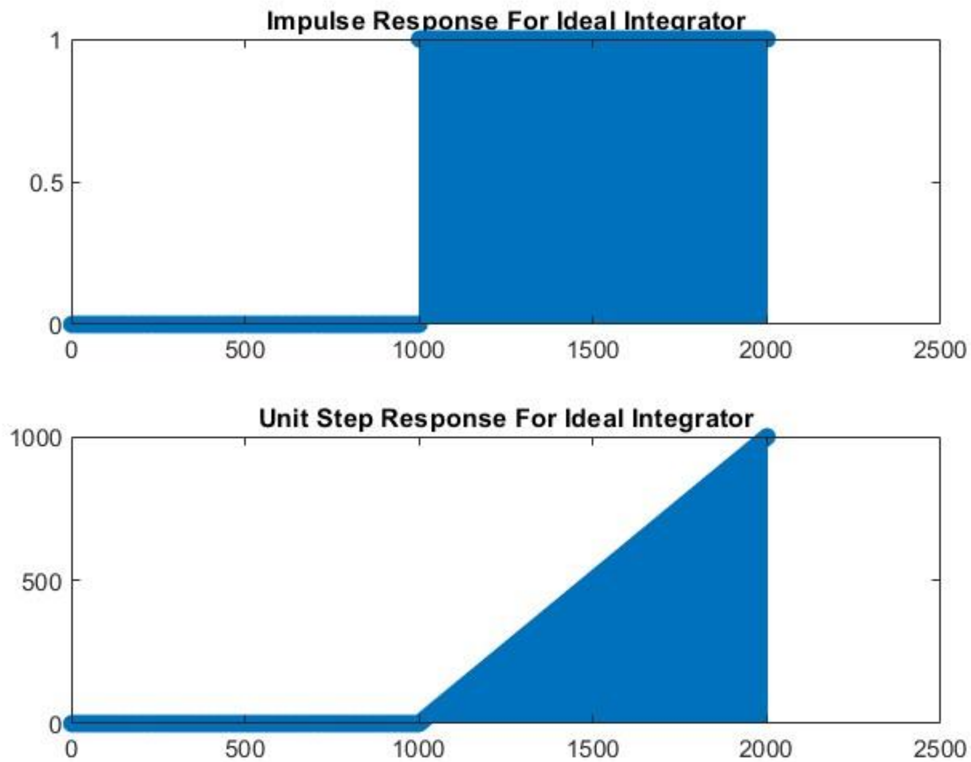
Causal: It is still integrator so it depends past and current values. so it is causal

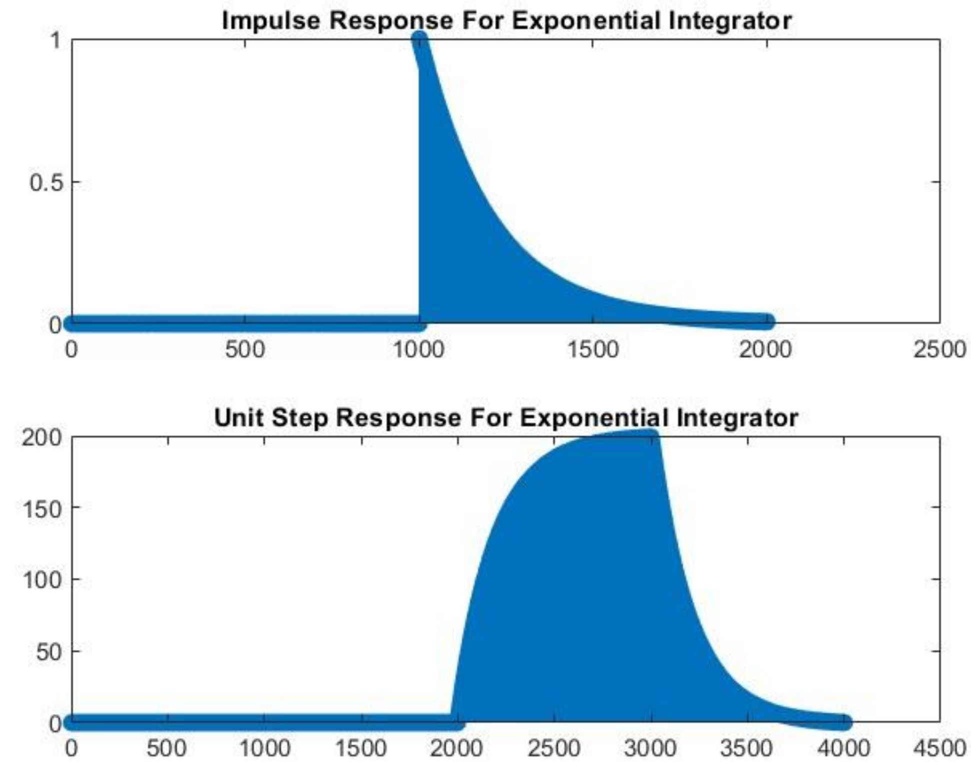
Memory: it has memory because it is causal

Linearity: The system is linear

Time-invariant: The system is time-invariant.

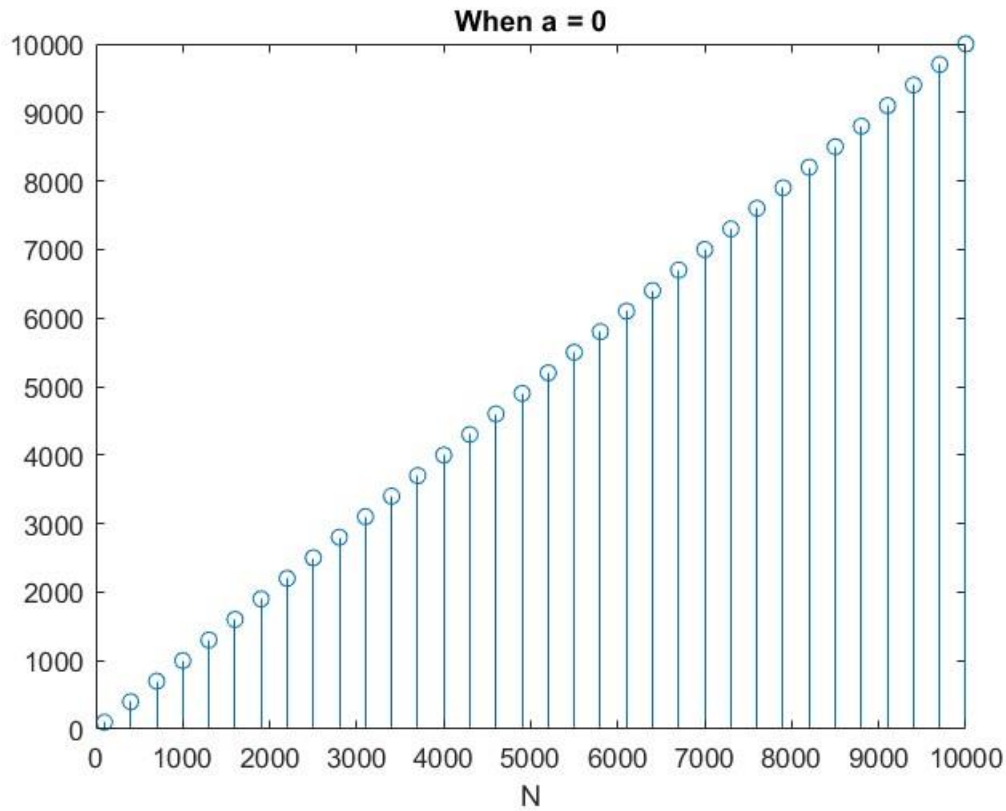
Part 1.3:

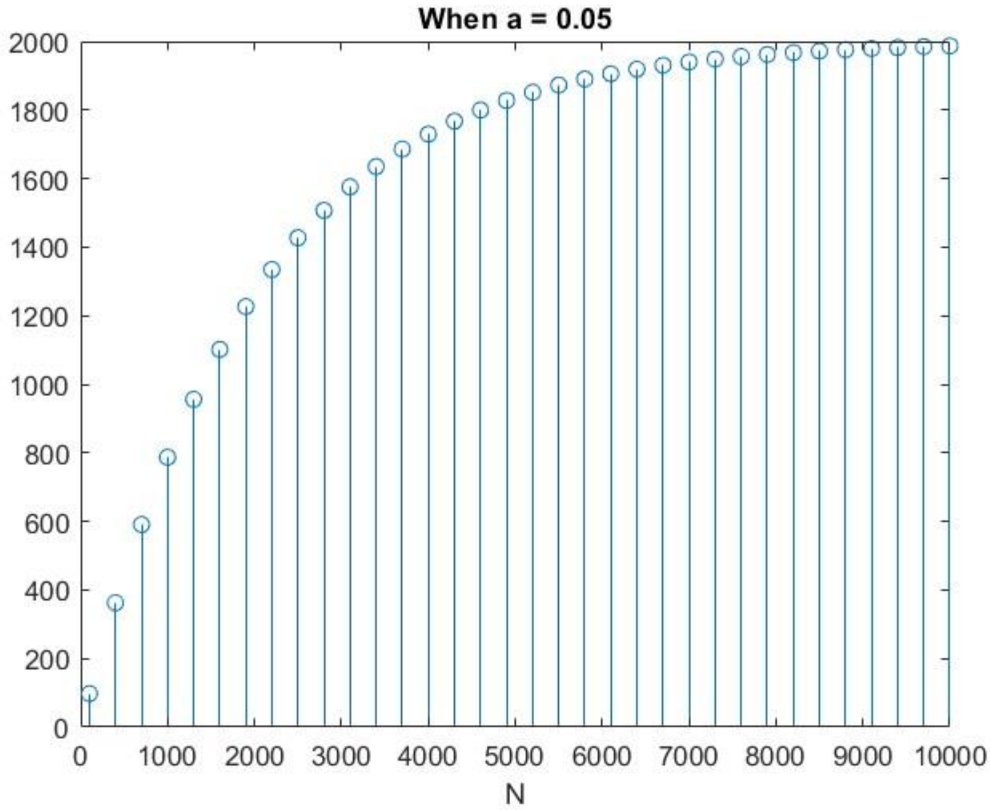

Impulse Response For Ideal Integrator
1
0.5
0
0
500
1000
1500
2000
2500
Unit Step Response For Ideal Integrator
1000
500
0
0
500
1000
1500
2000
2500

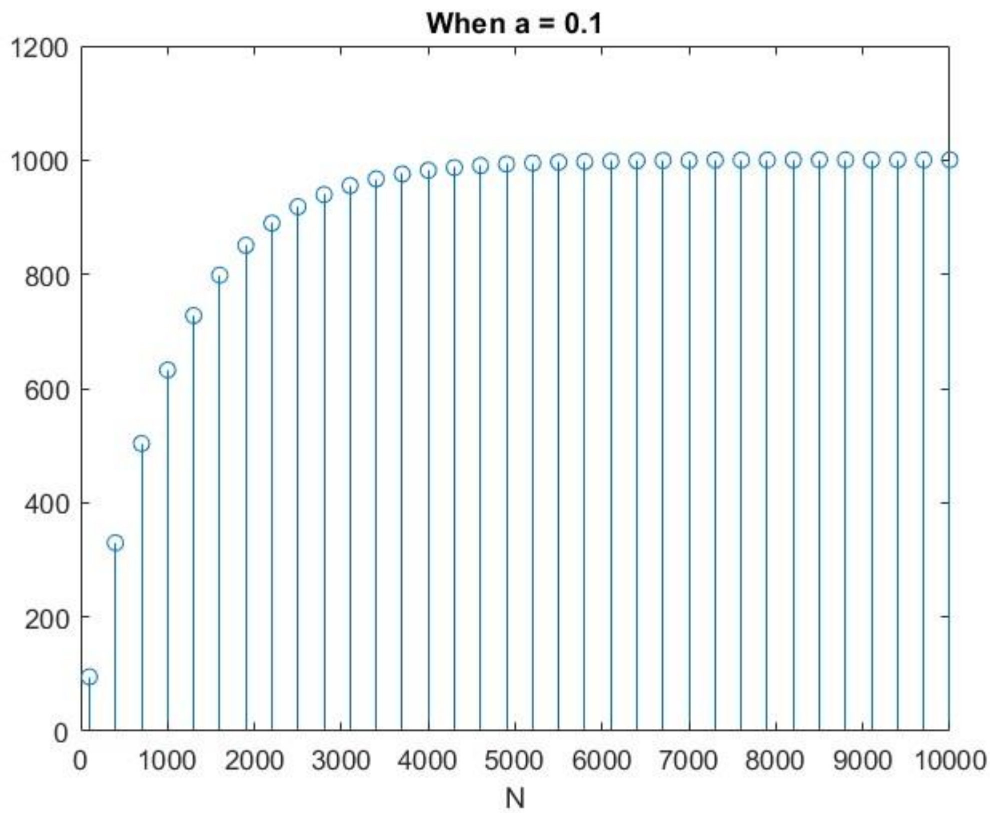

Impulse Response For Exponential Integrator
1
0.5
0
0
500
1000
1500
2000
2500
Unit Step Response For Exponential Integrator
200
150
100
50
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500

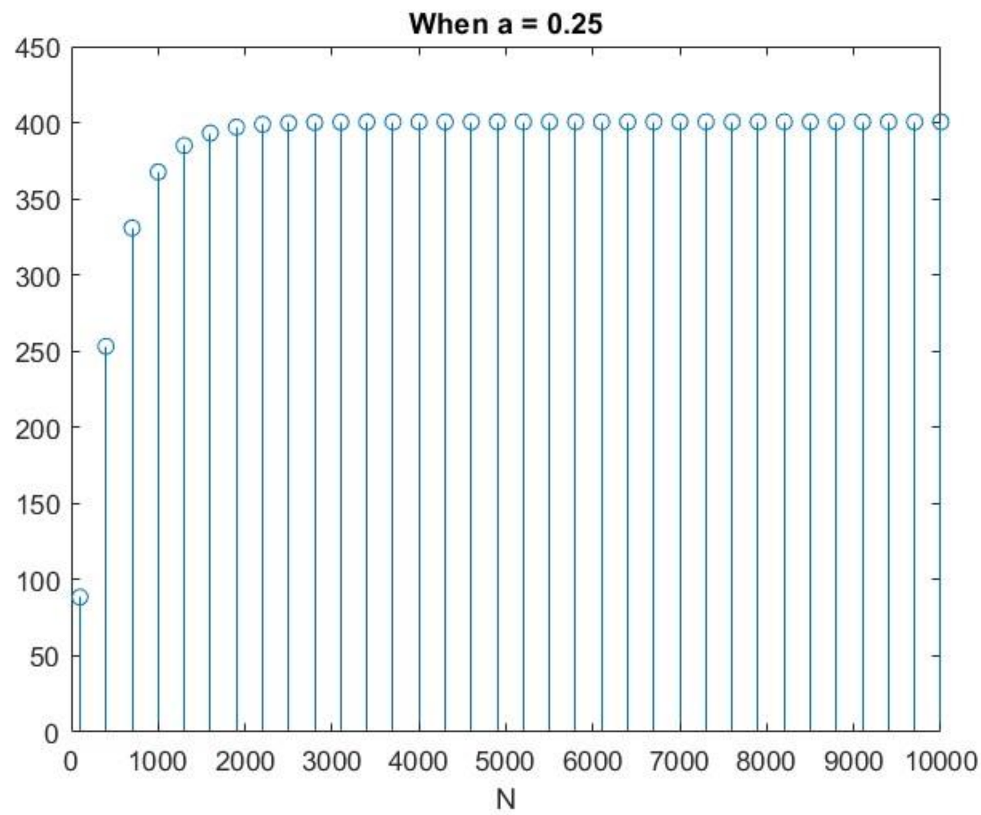
Part 2:

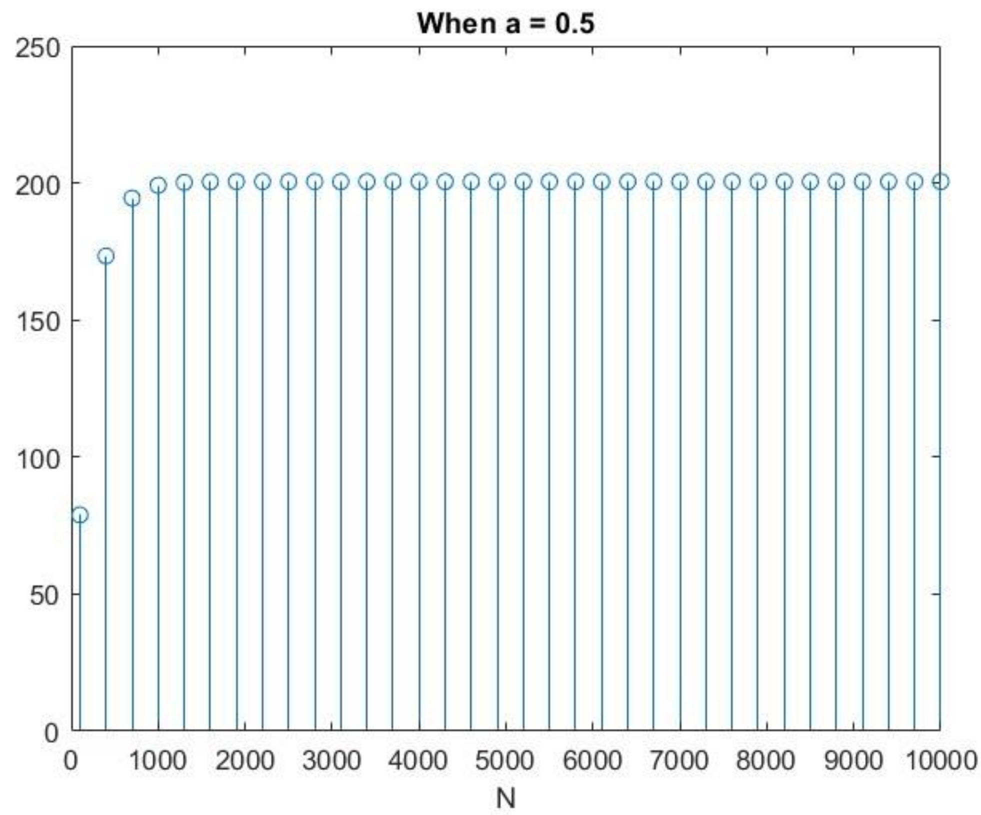
To determine whether a system is BIBO stable or not, we can sum its abs of impulse response as given Equation 2. We can observe whether the unit step response remains bounded. By building Equation 2 on MATLAB we can analyze it. If the summation converges to a finite value as the interval increases, we can conclude that the system is BIBO stable. The results which are asked for this lab task is following.



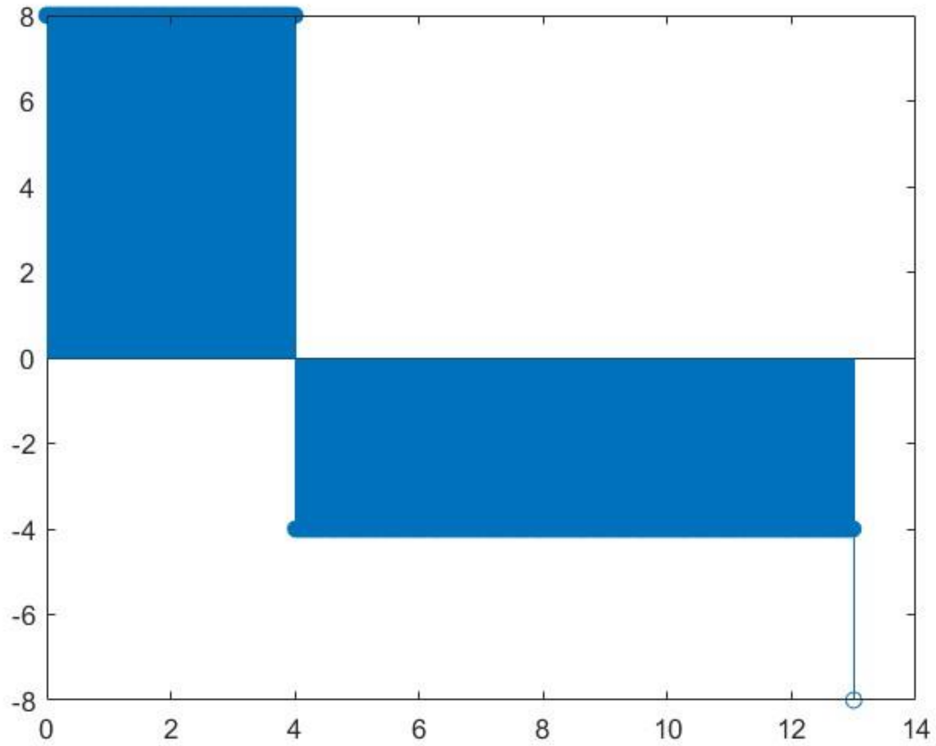

When a = 0.05
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
N


When a = 0.1
1200
1000
800
600
400
200
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
N


When a = 0.25
450
400
350
300
250
200
150
100
50
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
N

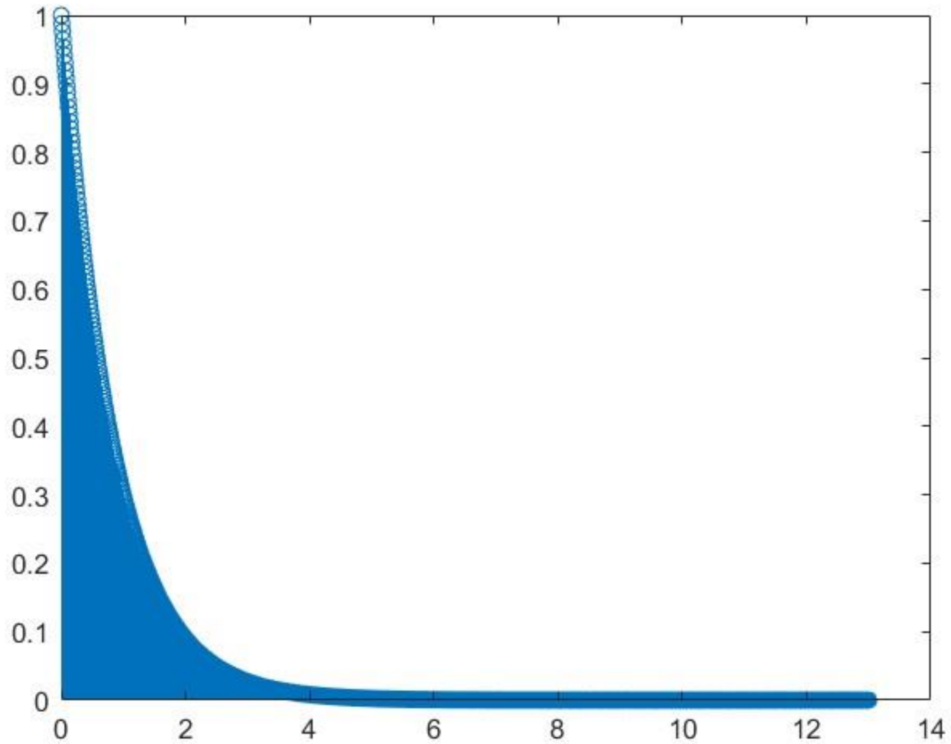

When a = 0.5
250
200
150
100
50
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
N

Part 3:



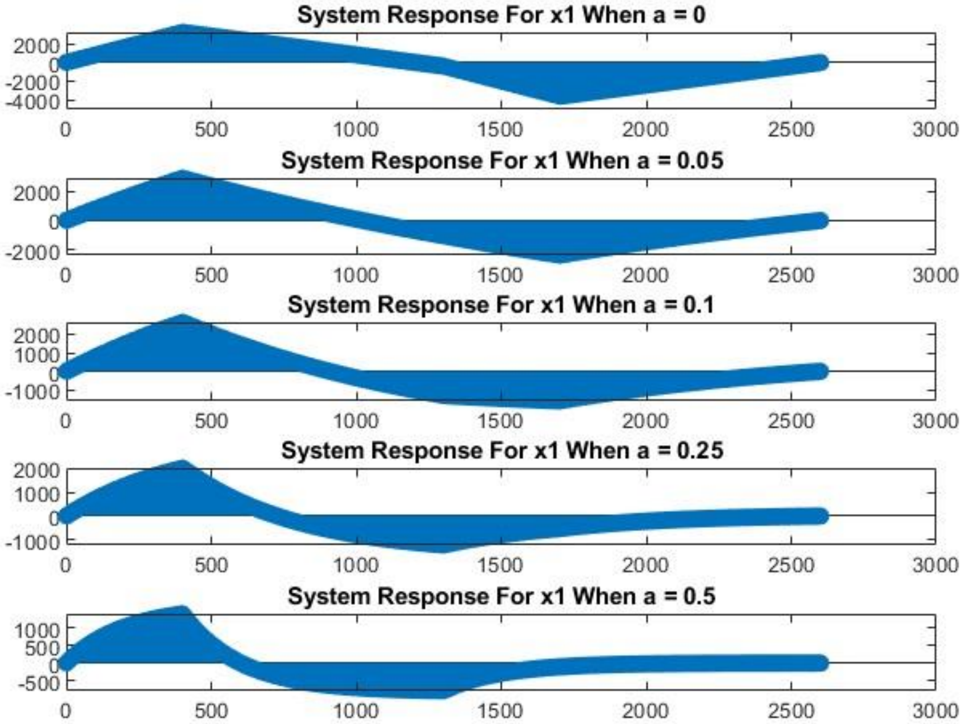
8
6
4
2
0
-2
-4
-6
-8
0
2
4
6
8
10
12
14

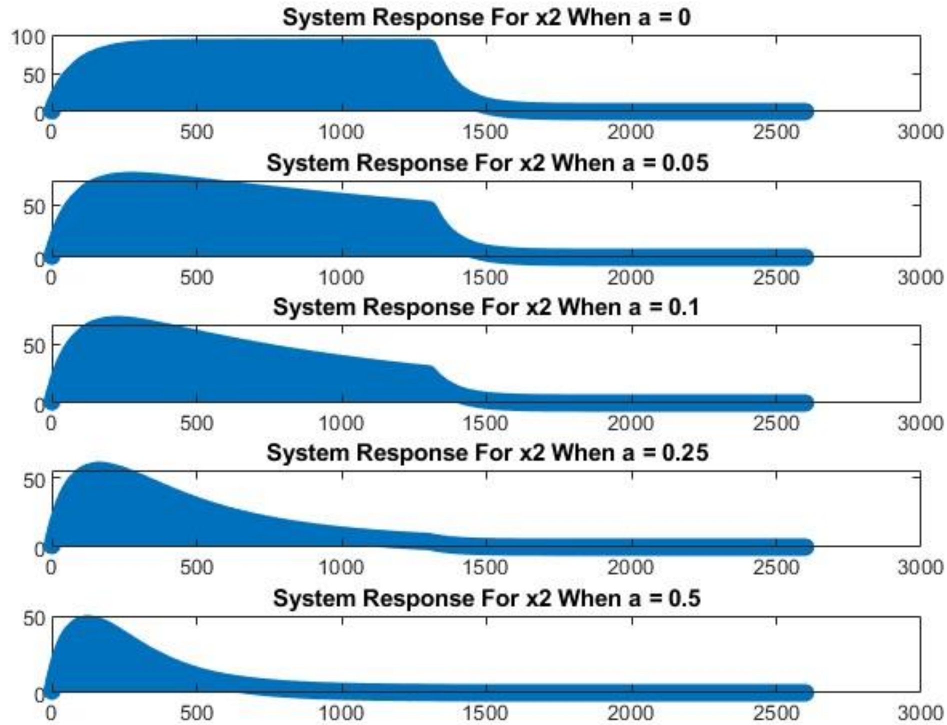
X1

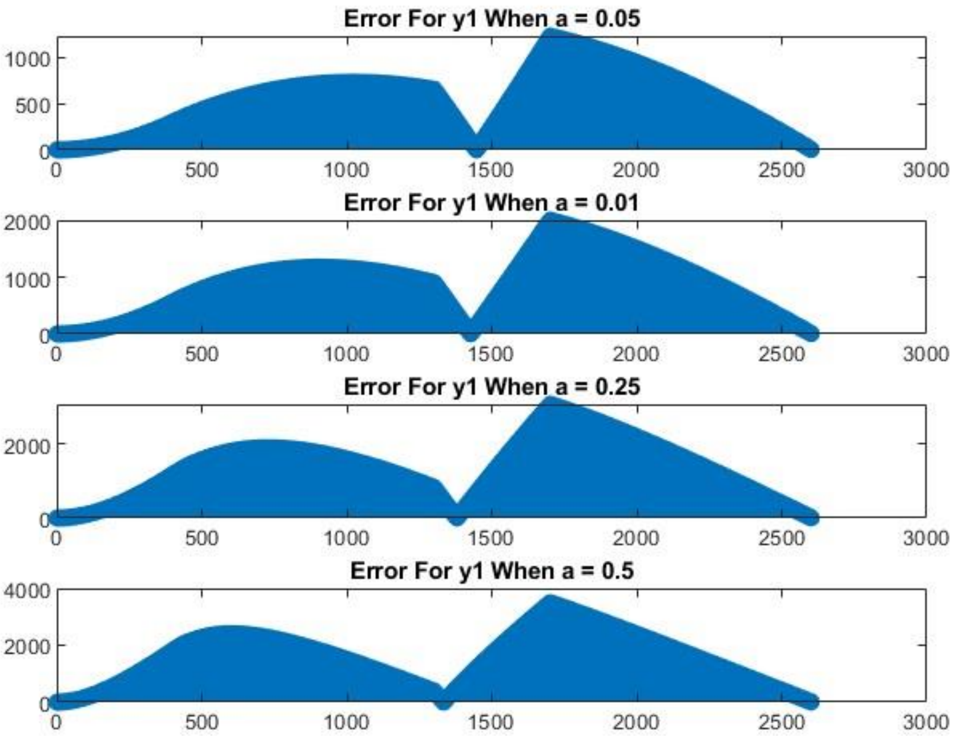


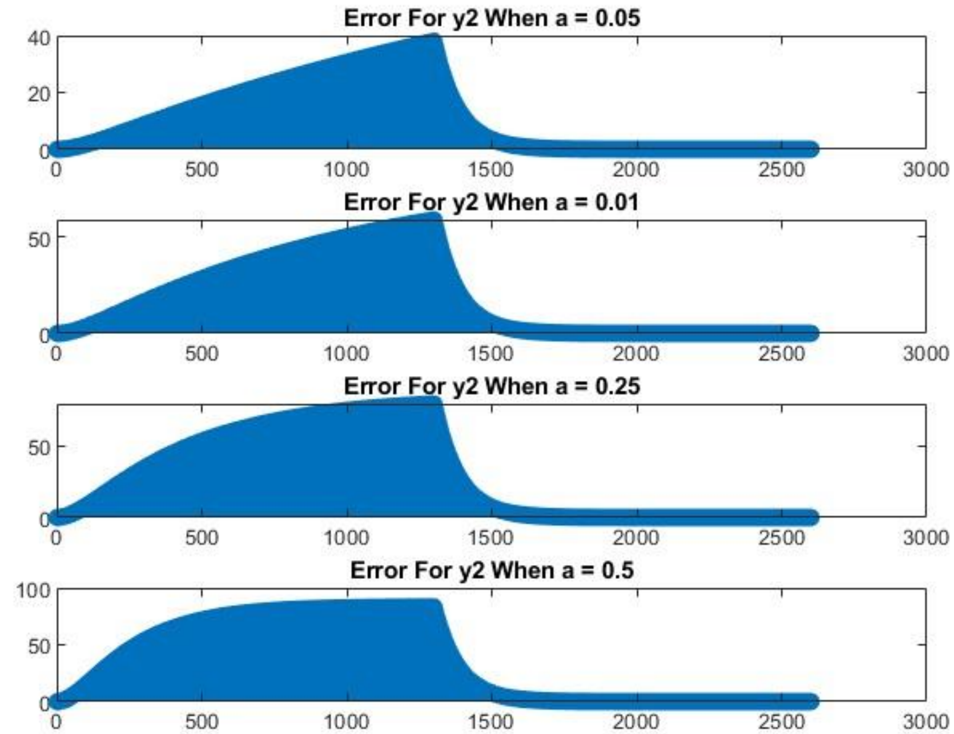
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2
4
6
8
10
12
14

X2


System Response For x1 When a = 0
System Response For x1 When a = 0.05
System Response For x1 When a = 0.1
System Response For x1 When a = 0.25
System Response For x1 When a = 0.5


System Response For x2 When a = 0
System Response For x2 When a = 0.05
System Response For x2 When a = 0.1
System Response For x2 When a = 0.25
System Response For x2 When a = 0.5


Error For y1 When a = 0.05
1000
500
0
0 500 1000 1500 2000 2500 3000
Error For y1 When a = 0.01
2000
1000
0
0 500 1000 1500 2000 2500 3000
Error For y1 When a = 0.25
2000
0
0 500 1000 1500 2000 2500 3000
Error For y1 When a = 0.5
4000
2000
0
0 500 1000 1500 2000 2500 3000


Error For y2 When a = 0.05
40
20
0
0 500 1000 1500 2000 2500 3000
Error For y2 When a = 0.01
50
0
0 500 1000 1500 2000 2500 3000
Error For y2 When a = 0.25
50
0
0 500 1000 1500 2000 2500 3000
Error For y2 When a = 0.5
100
50
0
0 500 1000 1500 2000 2500 3000

Part 4:

Part 4.1:

Part 4.1:

$y(t) = \frac{d\,x(t)}{dt} \approx \frac{x(t)-x(t-\Delta t)}{\Delta t}$

from definition

$y_{bwd}[n] = x[n] - x[n-1]$

Namely; $x'[n] = x[n] - x[n-1]$

$$(x'[n])' = (x[n] - x[n-1])'$$

$$x''[n] = x'[n] - x'[n-1]$$

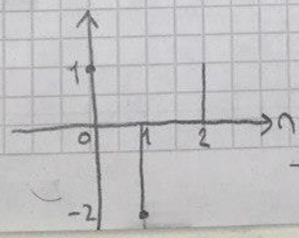
$$x''[n] = x[n] - x[n-1] - (x[n-1] - x[n-2]) = x[n] - 2x[n-1] + x[n-2]$$

So the system of sec. diff. when output ($x[n] = \delta[n]$)

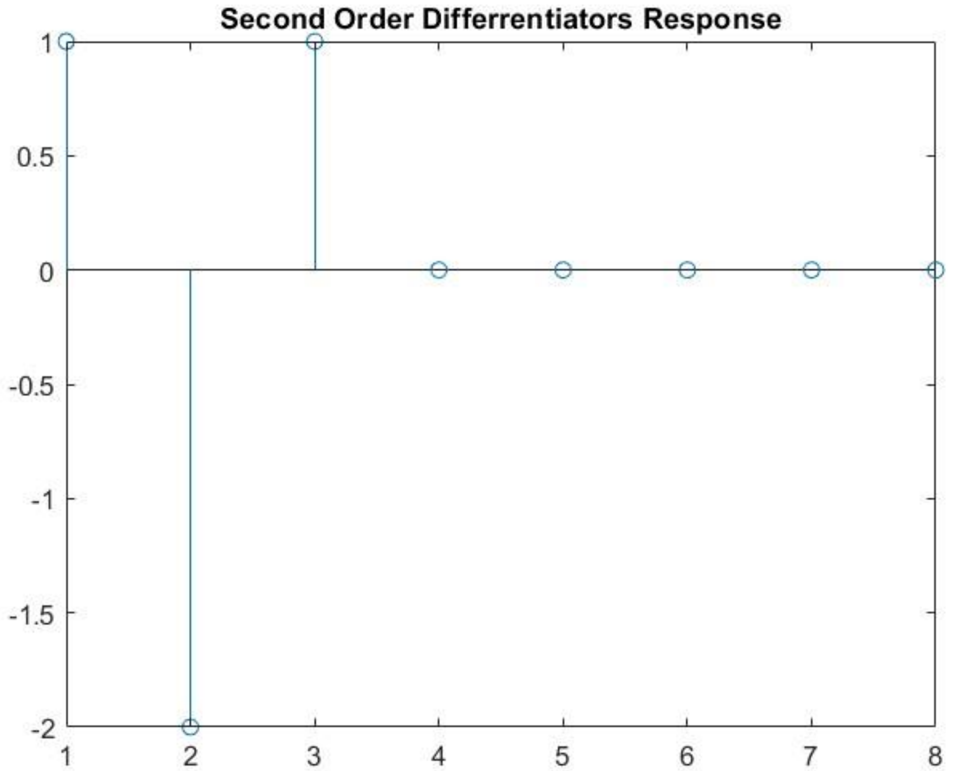
$$h[n] = \delta[n] - 2\delta[n-1] + \delta[n-2]$$

its causal

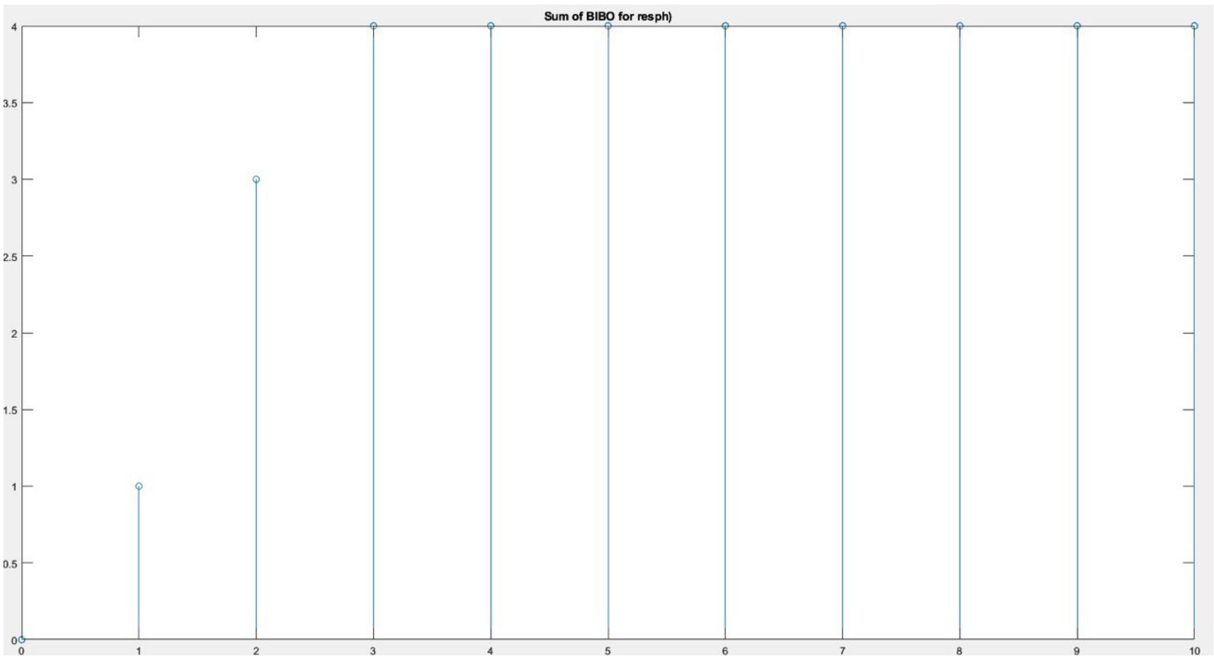
it has memory



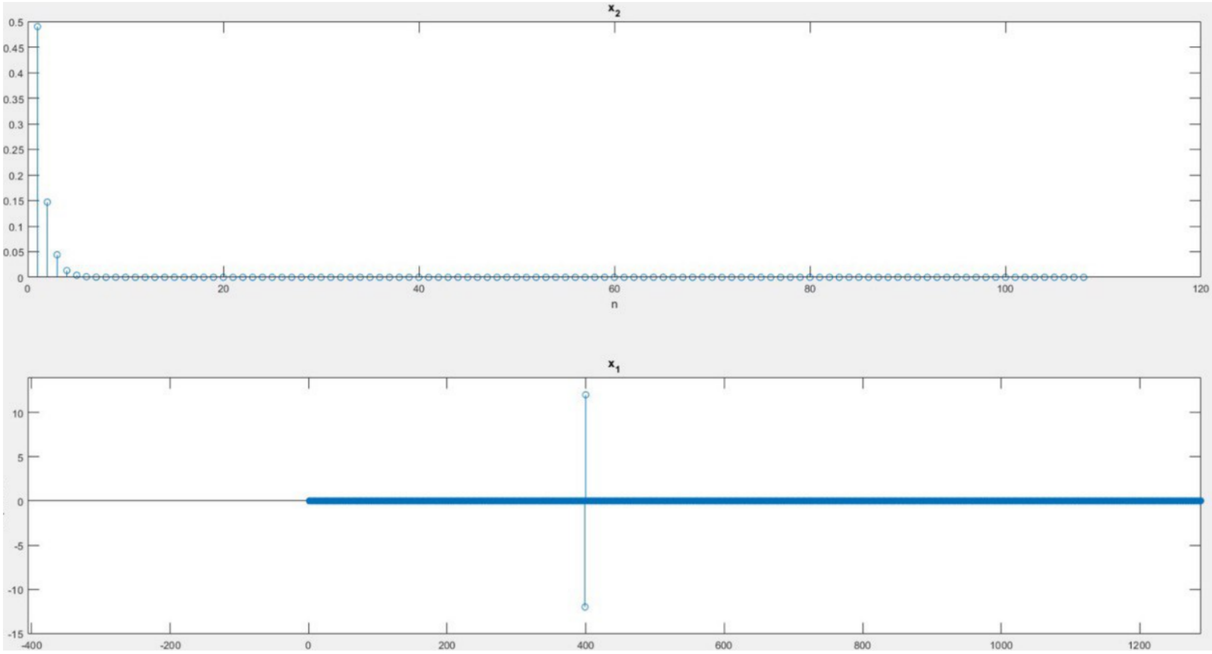
$\rightarrow$ FIR



Second Order Differrentiators Response



Sum of BIBO for resph)

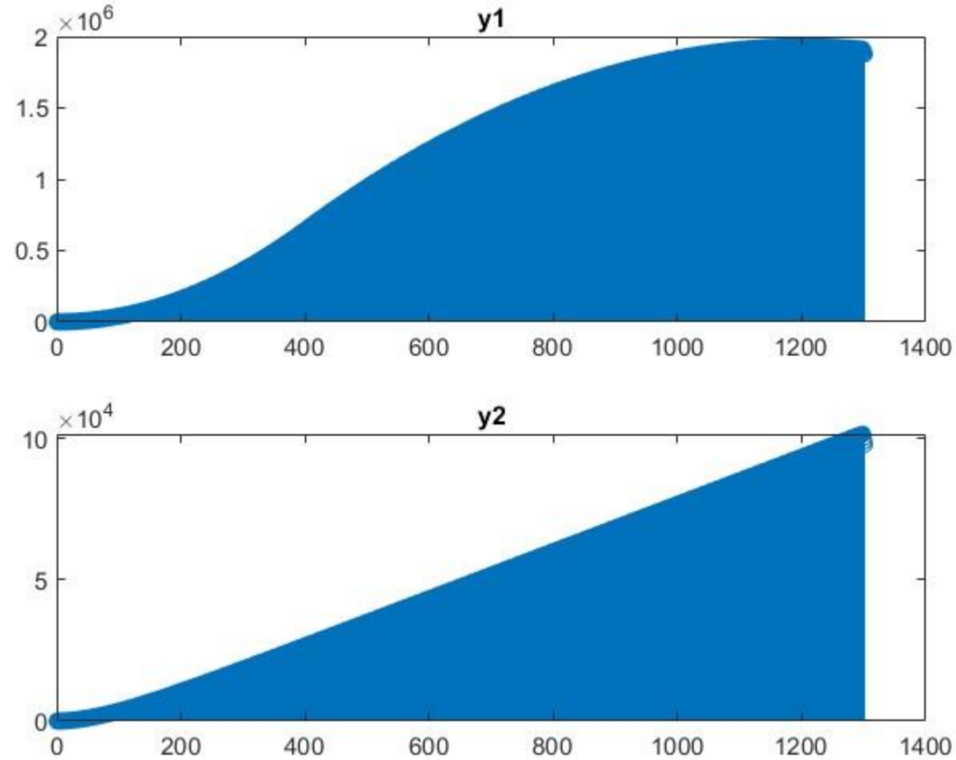


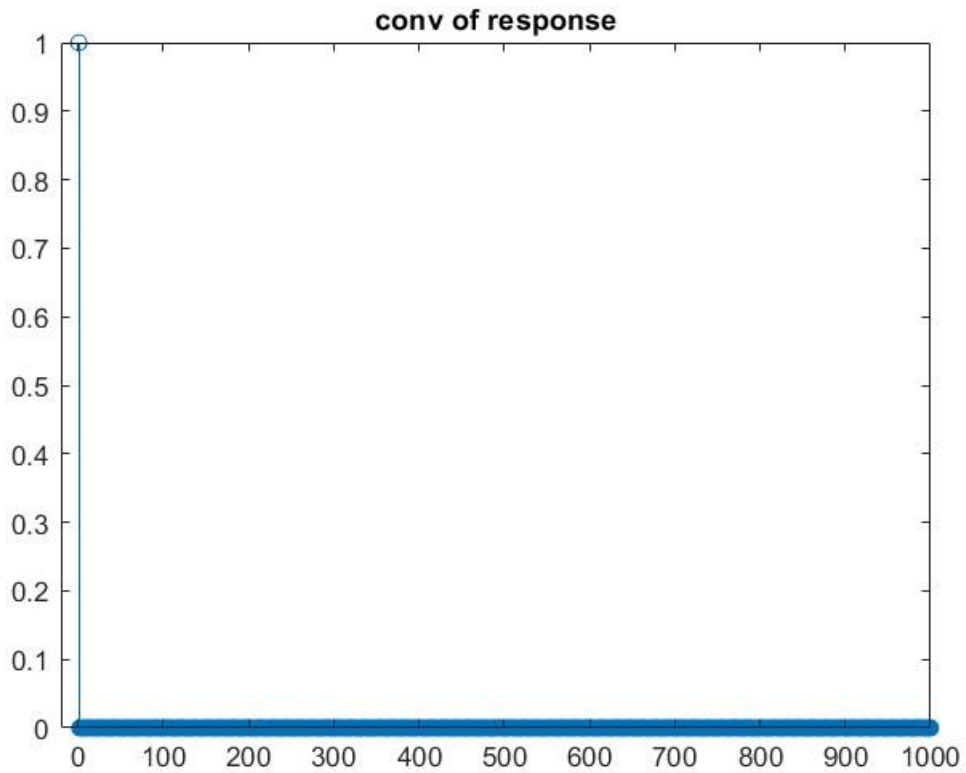
Part 4.2

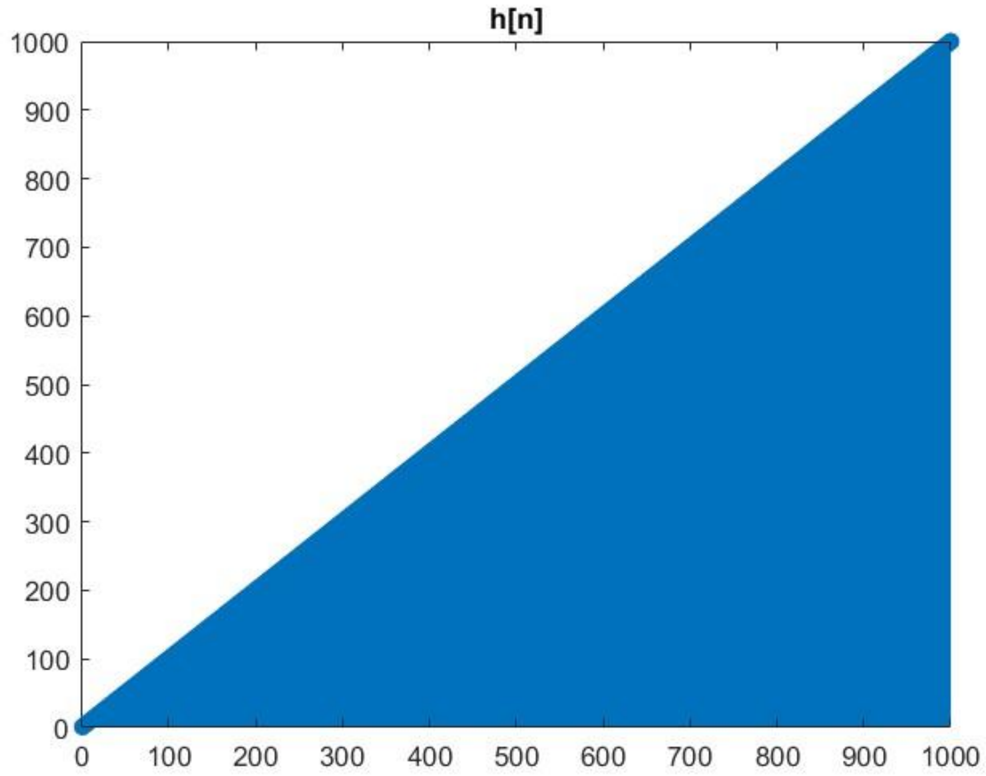
As given in the hint

function you get. Comment on this. (**Hint:** To obtain the inverse system, you may use the fact that $h[n] * h^{-1}[n] = \delta[n]$.)

So from hint and our knowledge we get Dirac Delta Function if we integrate of the second order derivative is second order integral. The results are following.


y1
×10^6
y2
×10^4


conv of response


h[n]
1000
900
800
700
600
500
400
300
200
100
0
0
100
200
300
400
500
600
700
800
900
1000

Appendices:

Codes For Part 1.3:

```
samp = 0.01
t = -10:samp:10
dir = zeros(size(t))
dir(1001)=1

stepfunc = zeros(size(t))
stepfunc(1001:end)=1

ydir = zeros(size(t))
ystep = zeros(size(t))

for n=2:(length(t))
    ydir(n) = ydir(n-1)+dir(n)
    ystep(n) = ystep(n-1)+stepfunc(n)
end
subplot(2,1,1);
stem(ydir);
title('Impulse Response For Ideal Integrator');

subplot(2,1,2);
stem(ystep)
title('Unit Step Response For Ideal Integrator');

a=0.5
impulseres = exp((-a)*t).*stepfunc
exporesstep = conv(impulseres,stepfunc)
figure;
subplot(2,1,1);
stem(impulseres);
title('Impulse Response For Exponential Integrator');

subplot(2,1,2);
stem(exporesstep)
title('Unit Step Response For Exponential Integrator');
```

Codes for Part 2:

a is changed manually for % a= 0, 0.05, 0.10, 0.25, and 0.

```
N_range = 100:300:10000
orreut = sumElements(N_range)
stem(N_range,orreut);
title('When a = 0.25');
xlabel('N');
% a= 0, 0.05, 0.10, 0.25, and 0.

function [sum_array] = sumElements(N_range)
    sum_array = zeros(1,length(N_range));
    uzuNran = length(N_range);
    a = 0.25;
    for i=1:uzuNran
        t = 0:0.01:N_range(i);
        h = exp((-a)*t);
```

```
        sum_array(i) = sum(abs(h(1:N_range(i))));
    end
end
```

Codes for Part 3:

```
samp = 0.01
t = 0:samp:13
stepfunc = ones(size(t))
stepfuncmin4 = stepfunc
stepfuncmin4(1:401) = 0;
stepfuncmin13 = stepfunc
stepfuncmin13(1:1300) = 0
x1 = (8*stepfunc) - (12*stepfuncmin4) - (4*stepfuncmin13)
x2 = 0:1:100;
x2 = (0.3).^t;


respa0 = exp((-0)*t)
respa005 = exp((-0.05)*t)
respa01 = exp((-0.1)*t)
respa025 = exp((-0.25)*t)
respa05 = exp((-0.5)*t)


resx1a0 = conv(respa0,x1)
resx1a005 = conv(respa005,x1)
resx1a01 = conv(respa01,x1)
resx1a025 = conv(respa025,x1)
resx1a05 = conv(respa05,x1)

resx2a0 = conv(respa0,x2)
resx2a005 = conv(respa005,x2)
resx2a01 = conv(respa01,x2)
resx2a025 = conv(respa025,x2)
resx2a05 = conv(respa05,x2)

figure;
stem(t,x1)

figure;
stem(t,x2)

figure;
subplot(5,1,1);
stem(resx1a0);
title('System Response For x1 When a = 0');

subplot(5,1,2);
stem(resx1a005);
title('System Response For x1 When a = 0.05');

subplot(5,1,3);
stem(resx1a01);
title('System Response For x1 When a = 0.1');

subplot(5,1,4);
stem(resx1a025);
```

```
title('System Response For x1 When a = 0.25');

subplot(5,1,5);
stem(resx1a05);
title('System Response For x1 When a = 0.5');



figure;
subplot(5,1,1);
stem(resx2a0);
title('System Response For x2 When a = 0');

subplot(5,1,2);
stem(resx2a005);
title('System Response For x2 When a = 0.05');

subplot(5,1,3);
stem(resx2a01);
title('System Response For x2 When a = 0.1');

subplot(5,1,4);
stem(resx2a025);
title('System Response For x2 When a = 0.25');

subplot(5,1,5);
stem(resx2a05);
title('System Response For x2 When a = 0.5');

figure;
subplot(4,1,1);
stem(abs(resx1a0-resx1a005));
title('Error For y1 When a = 0.05');

subplot(4,1,2);
stem(abs(resx1a0-resx1a01));
title('Error For y1 When a = 0.01');

subplot(4,1,3);
stem(abs(resx1a0-resx1a025));
title('Error For y1 When a = 0.25');

subplot(4,1,4);
stem(abs(resx1a0-resx1a05));
title('Error For y1 When a = 0.5');


figure;
subplot(4,1,1);
stem(abs(resx2a0-resx2a005));
title('Error For y2 When a = 0.05');

subplot(4,1,2);
stem(abs(resx2a0-resx2a01));
title('Error For y2 When a = 0.01');

subplot(4,1,3);
stem(abs(resx2a0-resx2a025));
title('Error For y2 When a = 0.25');
```

```
subplot(4,1,4);
stem(abs(resx2a0-resx2a05));
title('Error For y2 When a = 0.5');
```

Part 4.1

```
N_range = 0:1:10
resph = zeros(1,10)
resph(3)=1
resph = diff(resph,2)


samp = 0.01
t = 0:samp:13
stepfunc = ones(size(t));
stepfuncmin4 = stepfunc
stepfuncmin4(1:401) = 0;
stepfuncmin13 = stepfunc
stepfuncmin13(1:1300) = 0;
x1 = (8*stepfunc) - (12*stepfuncmin4) - (4*stepfuncmin13);
x2 = (0.3).^t;

secdiffconvforx1 = conv(x1,resph);
secdiffconvforx2 = conv(x2,resph);

figure;
stem(resph)
title('Second Order Differrentiators Response')

figure;
stem(N_range, sumElements(N_range));
title('Sum of BIBO for resph');

figure;
subplot(2,1,1);
stem(secdiffconvforx1);
title('x1');

subplot(2,1,2);
stem(secdiffconvforx2);
title('x2');
```

Part 4.2

```
ts=0.01
t=0:ts:13
toflen=length(t)
ocaas = zeros(1, toflen);
ourimp = ones(1, toflen);
ourimp(toflen+1)=0
ourimp(toflen+2)=0
ourimp=zeros(4,toflen+2)
ourimp(1)=0
ourimp(2)=0
ourimp(3)=1
```

```
for i = 3:1300
    ocaas(i) = 2*ocaas(i-1) - ocaas(i-2) + ourimp(i);
end

stepfunc = ones(size(t));
stepfuncmin4 = stepfunc
stepfuncmin4(1:401) = 0;
stepfuncmin13 = stepfunc
stepfuncmin13(1:1300) = 0;
ourimp_1 = (8*stepfunc) - (12*stepfuncmin4) - (4*stepfuncmin13);

h=zeros(1,10);
h(1)=1
h(2)=-2
h(3)=1
impconv = conv(ocaas,h);
impconv = impconv(3:end-4);
imp = ocaas(3:end);
ourimp_2 = (0.3).^t;
iska = conv(imp,ourimp_1);
lasc = conv(imp,ourimp_2);



figure;
subplot(2,1,1);
stem(iska(1:toflen));
title('y1');
subplot(2,1,2);
stem(lasc(1:toflen));
title('y2');
figure;
stem(impconv);
title("conv of response");
figure;
stem(imp);
title("h[n]");
```