Furkan Büyüksarıkulak

22002097

14.03.2025

EEE 321

Signals and Systems

Spring 2024-2025

Lab Assignment 3

In this lab task, we aimed to learn discrete form of ideal integrator and its impulse and unit step responses. We investigate linearity, time-invariance, causality, memory and stability of the signals. Then we learn how to approach first order differentiation of discrete signals and by using this approach we get second order differentiation and applied the signal tests required.

Part 1

Part 1.1: Ideal (Perfect) Integrator

Part 1.2: Another System

Note: Since Part 1.1 and Part 1.2 were made on paper, large figures were placed starting from the next page for easier reading.

EEE 321 Lab 3

Part 1

Part 1.1 Ideal (Perfect) Integrator

Ideal Integrator $\rightarrow$ $y(t) = \int_{-\infty}^{t} x(\tau)\, d\tau$

$$\delta(x-x_0) = \begin{cases} \infty & x = x_0 \\ 0 & x \neq x_0 \end{cases} \quad \Big\{ \text{ Dirac delta function}$$

when input is Dirac delta function in the ideal integrator the output is unit step function

$$h(t) = \int_{-\infty}^{t} \delta(\tau)\, d\tau = u(t) \quad \Big\{ \text{ unit step function}$$

for unit step response we need the input as unit step function

$$s(t) = \int_{-\infty}^{t} u(\tau)\, d\tau = \int_{0}^{t} 1\, d\tau = t\,u(t) = r(t) \quad \Big\{ \text{ unit ramp signal with slope 1}$$

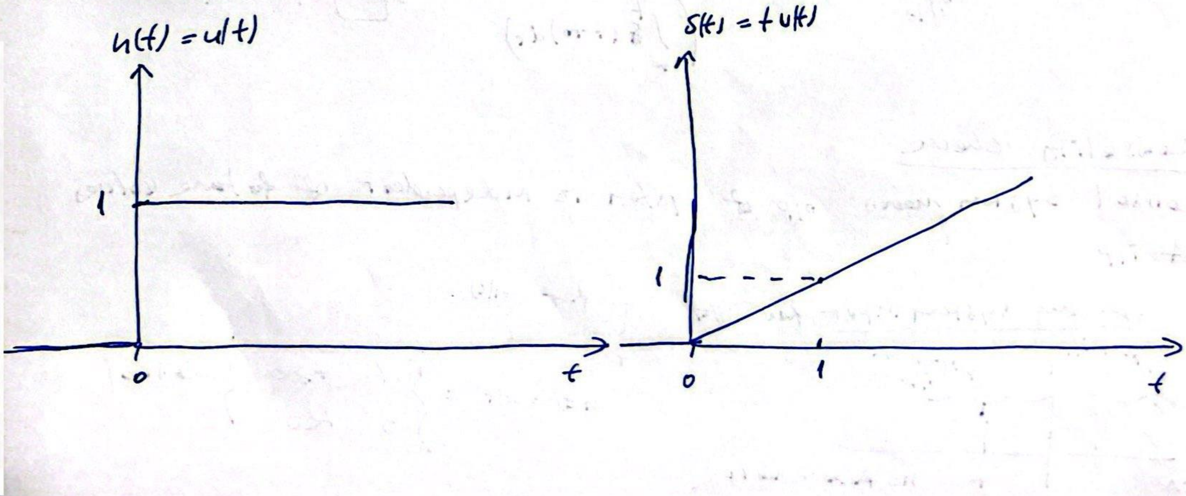


figure 1 : Part 1.1

Linearity Check

a) Law of Additivity

$x_1(t) \rightarrow \text{system} \rightarrow y_1(t) = \int_{-\infty}^{t} x_1(\tau)\, d\tau$

$x_2(t) \rightarrow \text{sys} \rightarrow y_2(t) = \int_{-\infty}^{t} x_2(\tau)\, d\tau$

$\Big\}\ y_1(t) + y_2(t) = \int_{-\infty}^{\tau} [x_1(\tau) + x_2(\tau)]\, d\tau$

$x_1(t) + x_2(t) \rightarrow \text{sys} \rightarrow y'(t) = \int_{-\infty}^{\tau} [x_1(\tau) + x_2(\tau)]\, d\tau$

LOA applies

b) Law of Homogeneity

$k\, y(t) = k \int_{-\infty}^{t} x(\tau)\, d\tau \Rightarrow k\, x(t) \rightarrow \text{system} \rightarrow y'(t) = \int_{-\infty}^{\tau} k\, x(\tau)\, d\tau = k\, y(t)$

LOH applies

so system is Linear

Time Invariance Check

$x(t) = \delta(t) \rightarrow \text{sys} \rightarrow u(t) \xrightarrow{\text{delay by } t_0} u(t - t_0)$

$\hookrightarrow \text{delay by } t_0 \rightarrow \delta(t - t_0) \rightarrow \text{sys} \rightarrow u(t - t_0)$

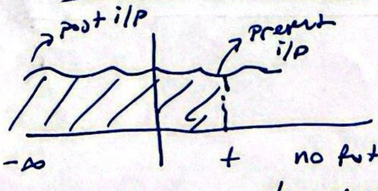
$\left(\int_{-\infty}^{t} \delta(t - t_0)\, dt\right)$

} Time Invariant system

Causality Check

causal system means o/p of system is independent of future values of i/p

for any system input for y(t)



no future value dependence causal

for h(t)

$h(t) = u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$ } causal

figure 2 : Part 1.1 Continiued

Part 1.2

$h(t) = e^{-at}\, u(t)$

for unit step response $(s(t)) \Rightarrow x(t) = u(t)$

$$\mathcal{L}\{u(t)\} = \int_{-\infty}^{\infty} u(t)\, e^{-st}\, dt = \int_{-\infty}^{0} \cancel{0\, e^{-st}} + \int_{0}^{\infty} 1\, e^{-st} = \left[\frac{e^{-st}}{-s}\right]_0^{\infty}$$

$$F(s) = \frac{-1}{s}\left[e^{-\infty} - e^{0}\right] = \frac{1}{s} \quad \text{for } \mathcal{L}\{u(t-a)\} = \frac{e^{-a}}{s}$$

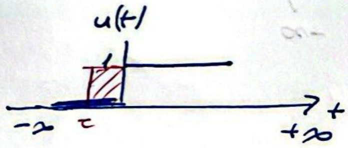
$$x(t) \longrightarrow \boxed{h(t)} \longrightarrow y(t) = h(t) * x(t)$$

$\downarrow$ Laplace $\qquad \downarrow$ Laplace $\qquad \uparrow$ Inverse Laplace

$$X(s) \longrightarrow \boxed{H(s)} \longrightarrow Y(s) = H(s) \cdot X(s)$$

$$s(t) = u(t) * h(t) = \int_{-\infty}^{\infty} u(\tau)\, \underbrace{h(t-\tau)}_{e^{-a(t-\tau)}\, u(t-\tau)}\, d\tau$$

$= 0$ as $t < 0$



$$s(t) = \int_{0}^{t} e^{-a\tau}\, d\tau = \left(\frac{1}{a} - e^{-at}\right) = \frac{1 - e^{-at}}{a}\, u(t)$$

Linearity check

LOA $\rightarrow x_1(t) = u_1(t) \rightarrow \text{sys} \rightarrow \frac{1-e^{at}}{a} u_1(t)$, $u_2(t) \rightarrow \text{sys} \rightarrow \frac{1-e^{at}}{a} u_2(t)$

$$y_1(t) + y_2(t) = (u_1(t) + u_2(t))\left[\frac{1-e^{-at}}{a}\right]$$

$$(x_1(t) + x_2(t)) \rightarrow \text{sys} \rightarrow y(t) = (u_1(t) + u_2(t))\left[\frac{1-e^{-at}}{a}\right] \quad \text{LOA} \checkmark$$

LOH $\rightarrow k\, y(t) = k\left[\frac{1-e^{at}}{a}\right] u(t)$

$$k\, u(t) \rightarrow \text{sys} \rightarrow k\left[\frac{1-e^{at}}{a}\right] u(t) \quad \text{LOH} \checkmark$$

→ because system is causal

} system is Linear

figure 3 : Part 1.2

Time Invariance

This system is time invariant because impulse response does not depend on absolute time

Causality

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

output of system is independent of future values of i/p

BIBO check

$$\int_{-\infty}^{\infty} |h(t)|\,dt \text{ must be } < \infty$$

$$= \int_{-\infty}^{\infty} |e^{-a\tau} u(\tau)|\,d\tau = \int_{0}^{\infty} e^{-a\tau}\,d\tau = \left[\frac{e^{-a\tau}}{-a}\right]_{0}^{\infty} = \frac{1}{a} < \infty$$

BIBO stable

we have BIBO stability compared to part 1.1

figure 4 : Part 1.2 continiued

Part 1.3: Discretization of the Two Systems

In this part we discritize ideal integrator by using sampling period of 0.01 s. The input-output equation of the system given in figüre 5. The system we obtained is the ideal accumulator and we calculated both the impulse response and the unit step response of the system we obtained.

Part 1.3 lets discritize Ideal Integral for → impulse response $h[n]$ → unit step response $u[n]$

$$s[n] = \sum_{k=-\infty}^{n} h[k] \Rightarrow h[n] = s[n] - s[n-1]$$

ideal accumulator

and

$$[\delta[n] - \delta[n-1]] * u[n] = \delta[n]$$

$$y[n] = [x[n] - x[n-1]] * [s[n]] = \sum_{k} [x[k] - x[k-1]]\, s[n-k]$$

figure 5 : Part 1.3

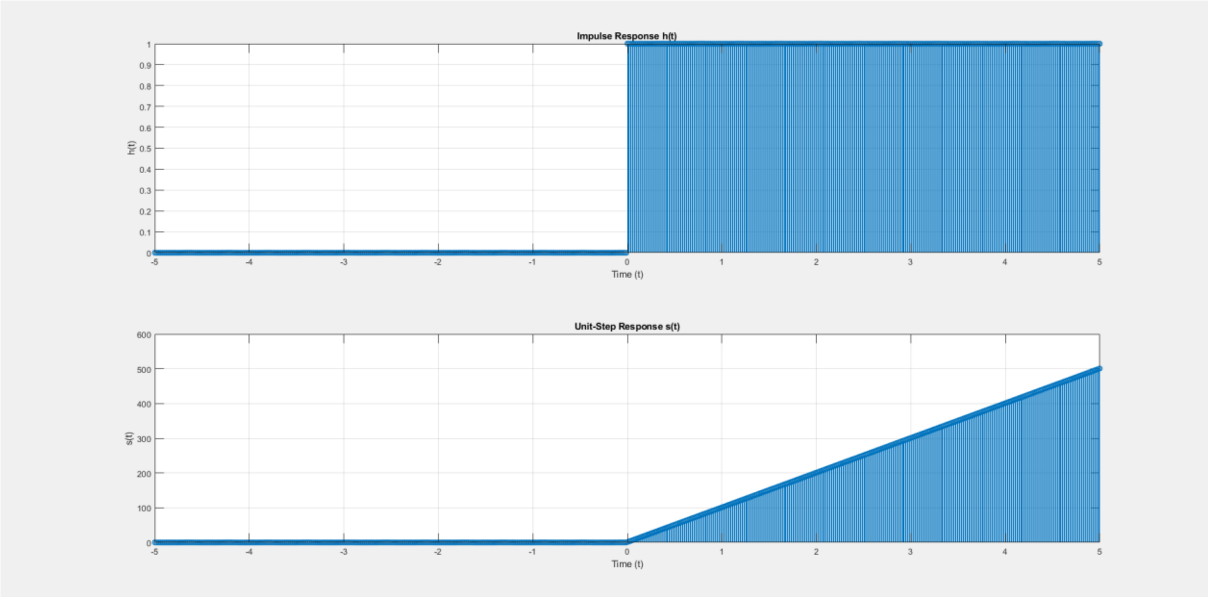


figure 6 : Part 1.3 Impulse and Unit Step Responses Plot

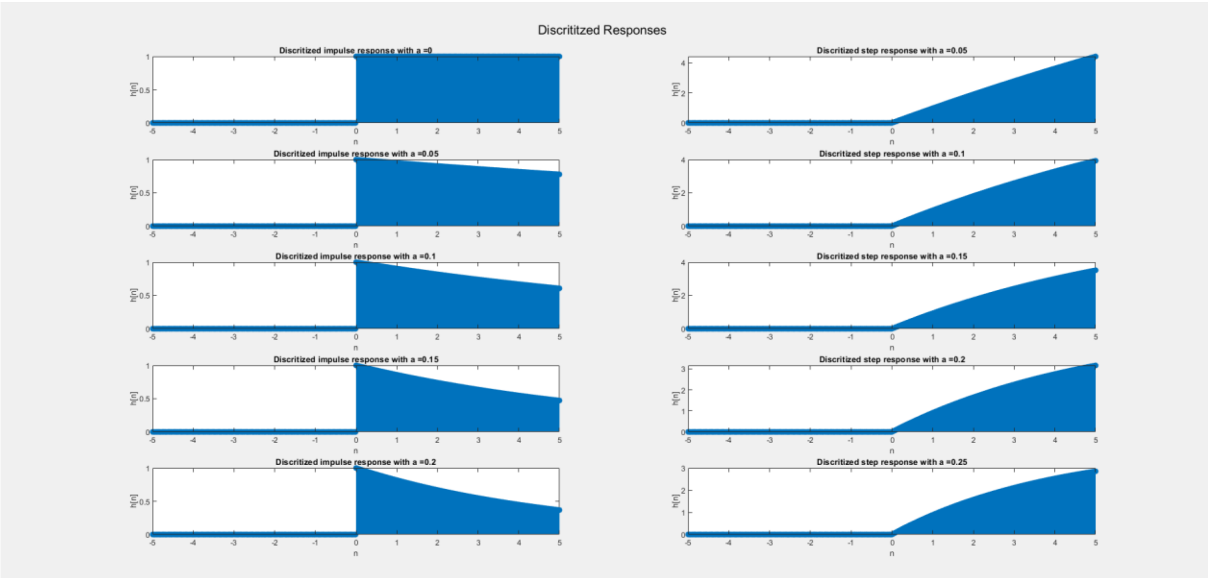


figure 7 : Part 1.3 Discritized Responses with respect to changing a values

<u>Part 2</u>

In this part, the BIBO stability of the discrete system we obtained in part 1.3 was tested. As we saw in part 1.2, we expect to see the stability in the system in exponential systems. If the cumulative sum of the resulting system is less than infinite, we consider it stable. Since we were not allowed to use the cumsum function in MATLAB, we wrote a function called sumElements. The desired template for this function was as follows:

function [sum array] = sumElements(h, N range) where

• h represents the impulse response to be processed

• N range represents the array of N values for which the summation is conducted

• sum array represents an array whose ith element is the summation of the elements of the given impulse response between the indices −i and i

We performed stability tests by changing the a value in the system where we spaced the range as N_range = [100:300:10000] and plotted them on the graph. The a = 0 value represents the ideal integrator.

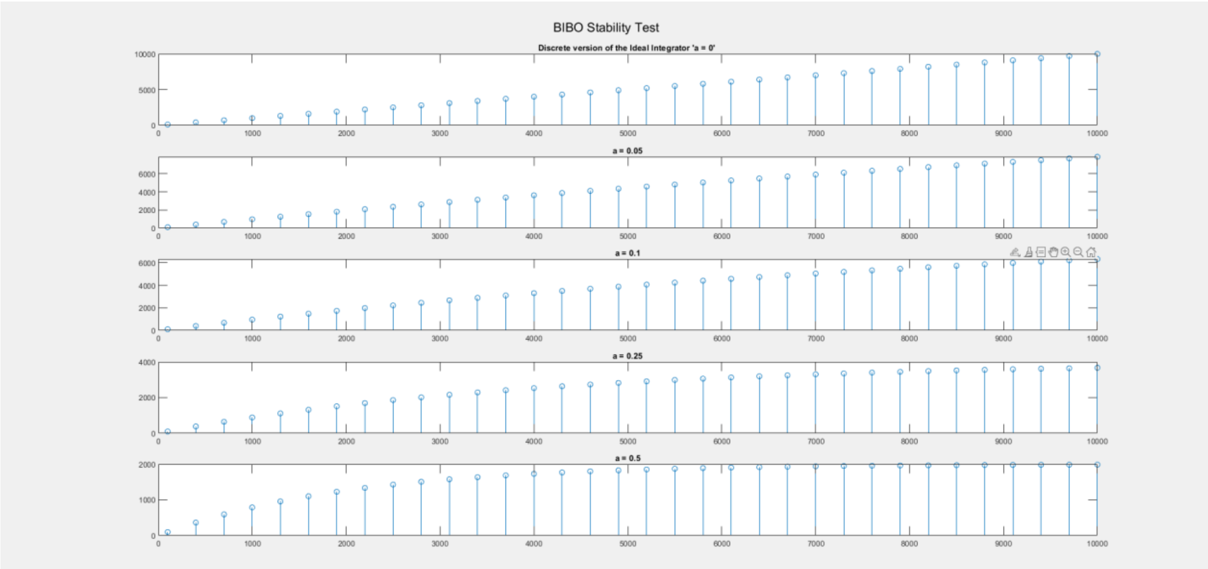


figure 8 : Part 2 Discrete Version of Ideal Integrator with Respect to Changing a values

Part 3

In this part, we will examine the difference in the output of the two systems we have examined in the previous parts. The functions we will give as input to these systems are given as follows.

$$x1[n] = 8\,(u[n] - u[n-4]) - 4\,(u[n-4] - u[n-13])$$

$$x2[n] = (0.3)n\, u[n]$$

First, we do the discrete plot of the ideal integrator for these two systems using the value a = 0. Then, we performed the following subtraction operation for the values a = [0, 0.05, 0.10, 0.25, 0.5].

$$E1[n] = |\, y\_ideal\,1\,[n] - y1[n]\, |$$

$$E2[n] = |\, y\_ideal\,2\,[n] - y2[n]\, |$$

The graphs we obtained as a result of these operations are as follows.

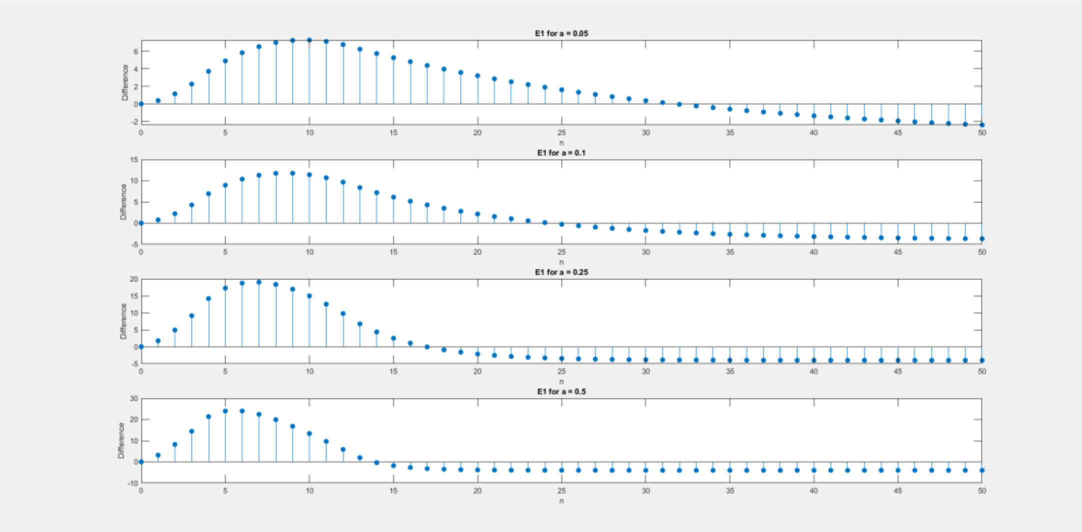


figure 9 : Part 3 E1 Differerence Sequence for different a values

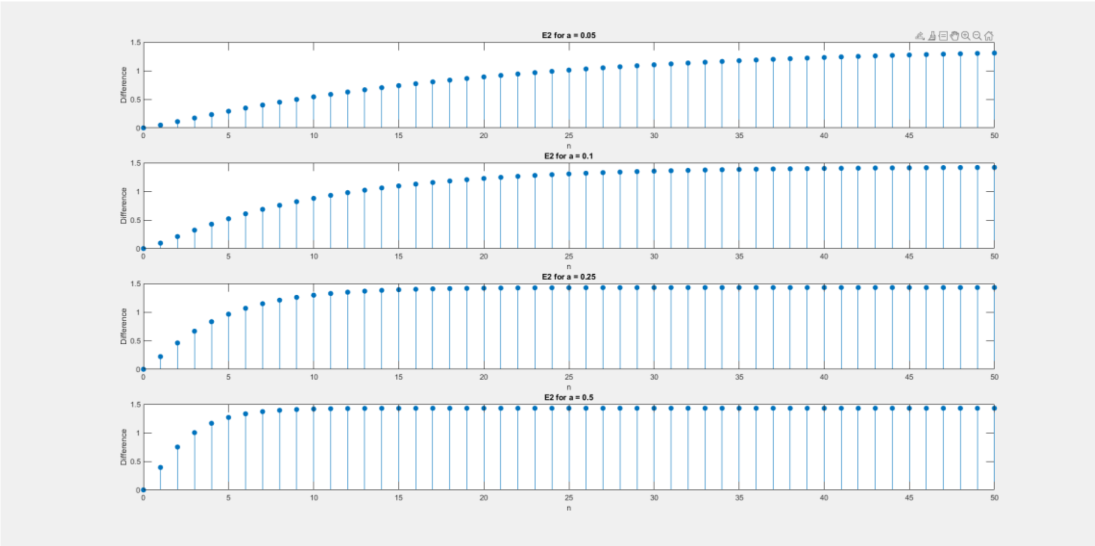


figure 10 : E2 Difference Sequence for different a values

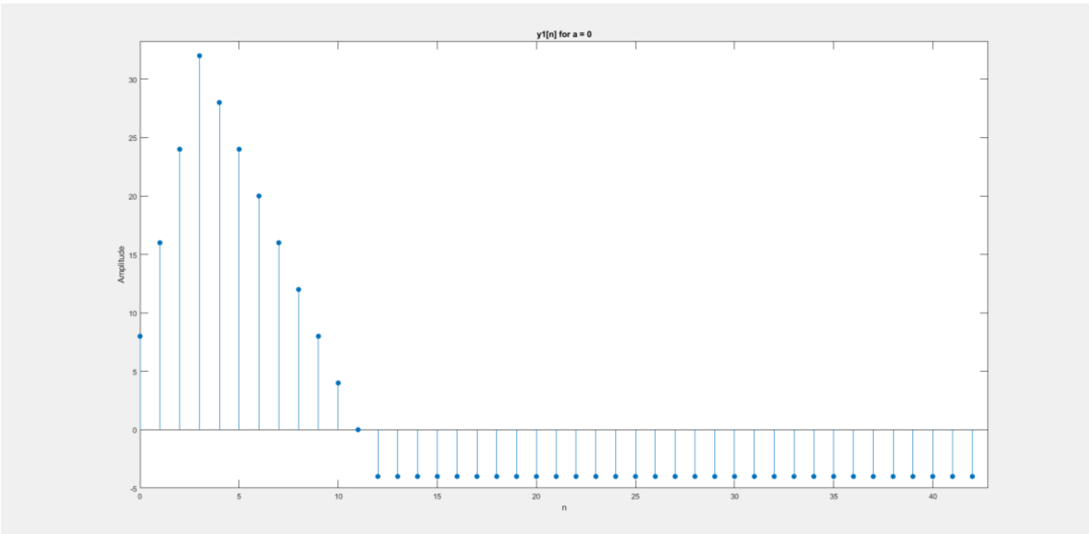


figure 11 : Part 3 Response of Discrete X1 signal

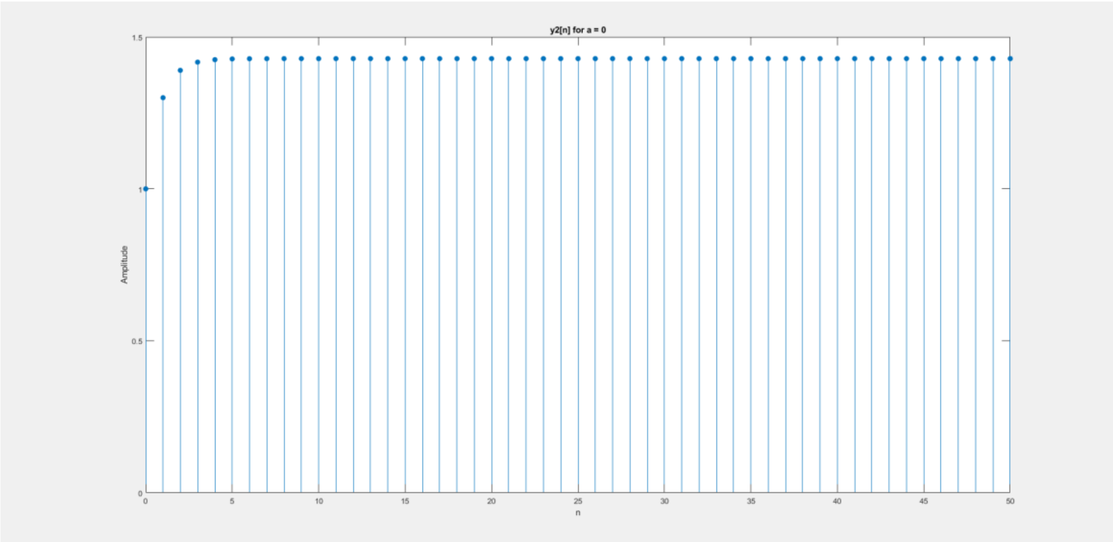


figure 12 : Part 3 Response of Discrete X2 signal

<u>Part 4</u>:

Part 4.1 First- and Second-Order Differentiation

It is given how to obtain the fist order difference equation for a discrete system using the CT derivative function. Using this approach, we obtain the second-order differentiation. As a result, the system we obtain is the finite impulse response (FIR). This derivation process is done manually as in the figures below. Then the system was implemented in MATLAB and the Impulse response was examined both manually and by plotting via Matlab. Then the BIBO stability of the system was tested and the x1 and x2 responses were plotted.

Part 4

Part 4.1

firstly we need to understand how to reach first difference equation

Consider LTI system with impulse response $h[n] = u[n]$ and i/p with $x[n]$. Using convolution sum to calculate response to input

$$y[n] = \sum_{-\infty}^{+\infty} x[k]\, u[n-k] \qquad \begin{cases} u[n-k] = 0 & \text{if } n-k<0 \\ u[n-k] = 1 & \text{if } n-k \geq 0 \end{cases}$$

so the equation becomes

$$\Rightarrow y[n] = \left.\sum_{k=-\infty}^{n} x[k] \right\} \text{accumulator}$$

$y[n] = x[n] - x[n-1]$ ⇒ so from given definition we reach

$$s[n] = u[n] * h[n] \rightarrow h[n] = s[n] - s[n-1]$$

$$\Rightarrow h(t) = \frac{d\,s(t)}{dt} = s'(t)$$

so for second order differentiation we need

$$\frac{d^2 x(t)}{dt^2} = \frac{d}{dt}\left(\frac{d\,x(t)}{dt}\right) = x(t) * u_1(t) * u_1(t)$$

$$\frac{d^2 x(t)}{dt^2} \xrightarrow{\text{discrete}} \left[\, x[n] - x[n-1] \,\right] * u[n]$$

$$\text{second order differentiation} = \sum_{-\infty}^{\infty} \left[\, x[k] - x[k-1] \,\right] u[n-k]$$

figure 13 : Part4 Handwritten Solution

$\Rightarrow \left(x[n] - x[n-1]\right) * u[n]$ $\Rightarrow$ let $h[n] = y[n]$, $x = \delta[n]$

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

<u>causality check</u>

for $n < 0 \rightarrow 0$
for $n = 0 \rightarrow 1$
" $n = 1 \rightarrow -2$

for $n = 2 \rightarrow 1$
for $n > 2 \rightarrow 0$
outputs are independent of future values of i/p
system is <u>causal</u>

<u>memory check</u> = system is dynamic because output of system depend on past values of input with the variables ("$-\delta[n-1]$", "$\delta[n-2]$")
(system has memory = dynamic)

<u>BIBO check</u>

$$\sum_{k=-\infty}^{\infty} |h[n]| = \sum_{k=-\infty}^{\infty} \left| \delta[n] - 2\delta[n-1] + \delta[n-2] \right| = 4 < \infty \quad \Bigg\} \text{ BIBO stable}$$

System has finite outputs so that it is a (FIR) finite impulse response.

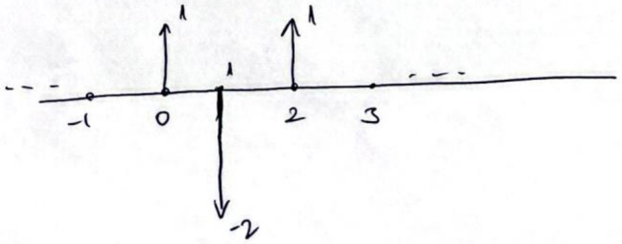


figure 14 : Part4 Handwritten Solution Continiued

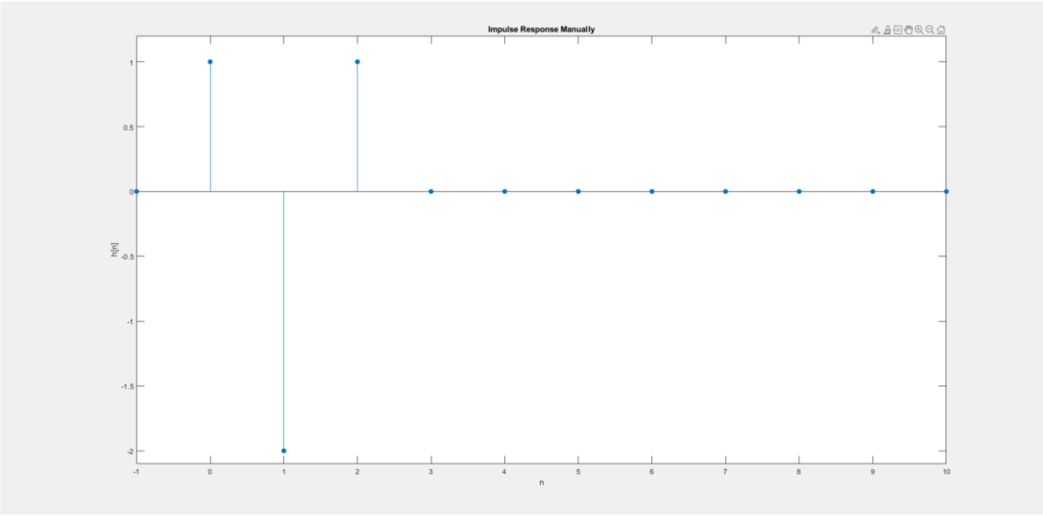


figure 15 : Part 4 Discrete Impulse Response Plot Manually

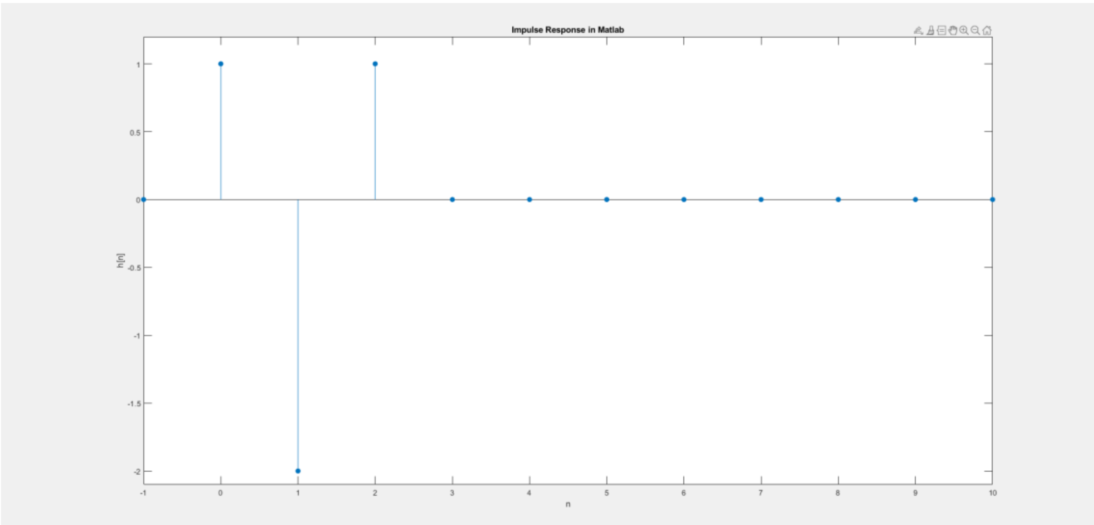


figure 16 : Part 4 Discrete Impulse Response by MATLAB

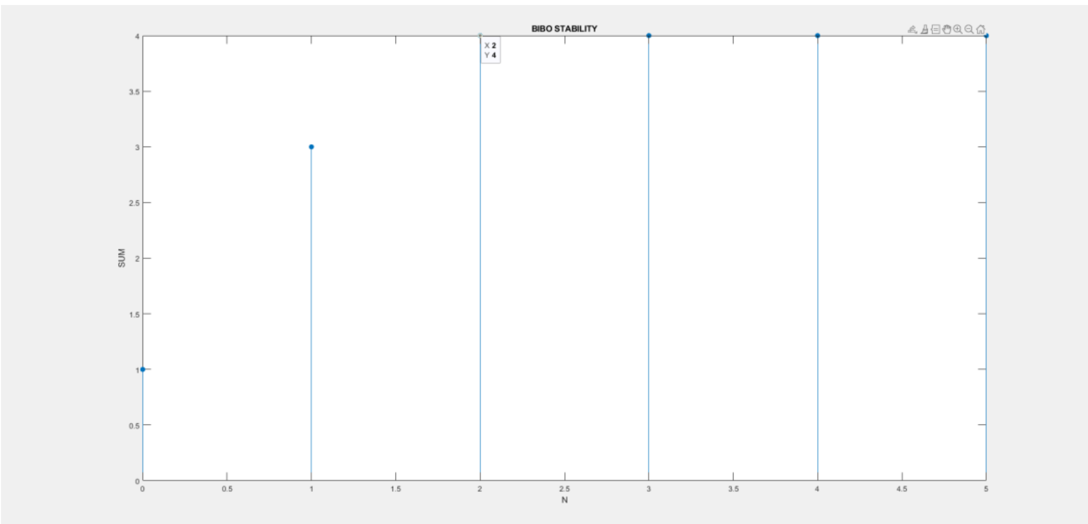


figure 17 : Part 4 BIBO Stability Test

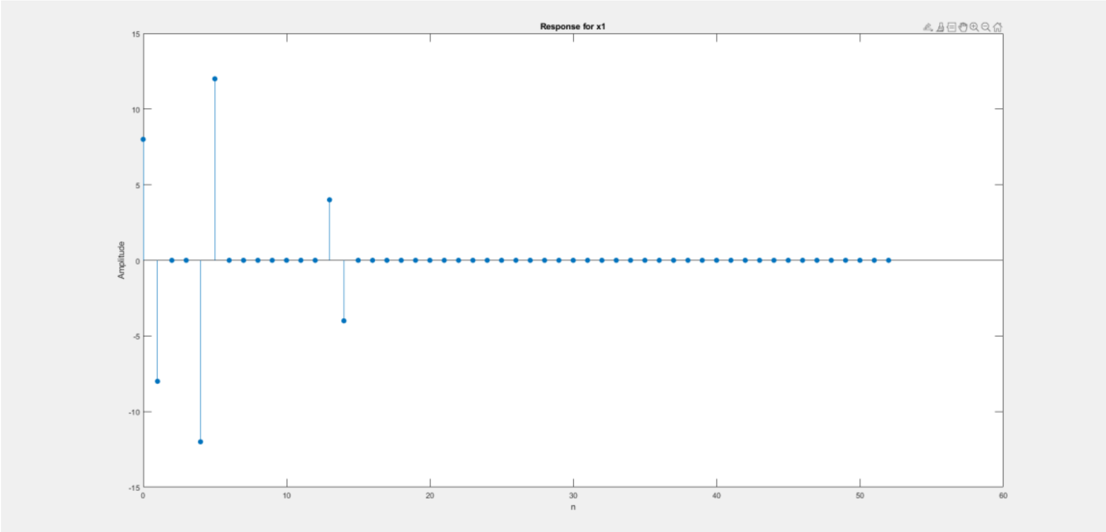


figure 18 : Part 4 Response of Discrete X1 signal

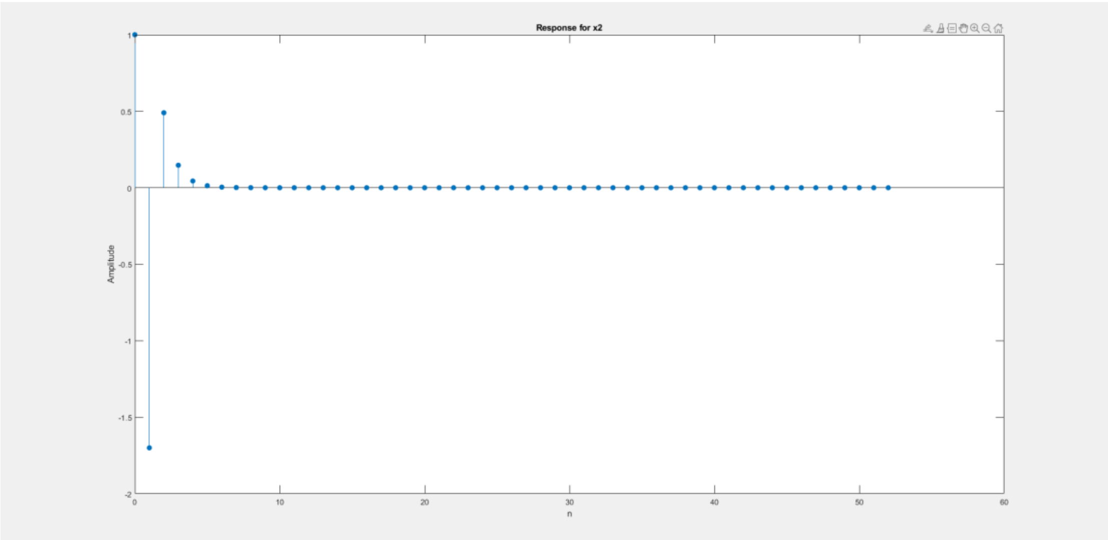


figure 19 : Part 4 Response of Discrete X2 signal

Part 4.2 : Invertibility of Second-Order Difference

In this part, we were asked to test whether the second order difference system is invertible. In order to test whether the system is invertible, we had to use the following equation.

$$h[n] * h^{-1}[n] = \delta[n]$$

When we convolve the inverse of the system with the main system, the result we obtain should give the dirac delta function and the graph in Figure 20 shows that our system is invertible. Additionally, the impulse response of the inverse system and the x1 and x2 responses are given graphically. Our system is linear and time-invariant also when we check causality memory and invertibility by MATLAB our result gives us system is invertible, causal and has memory.

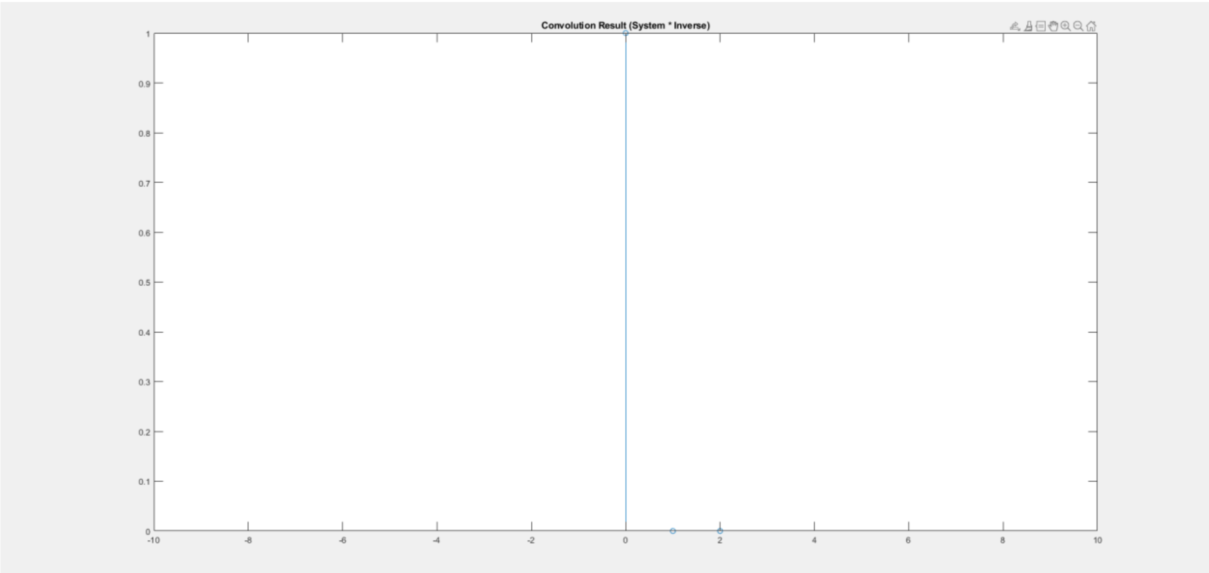


figure 20 : Convolution result between system and its inverse

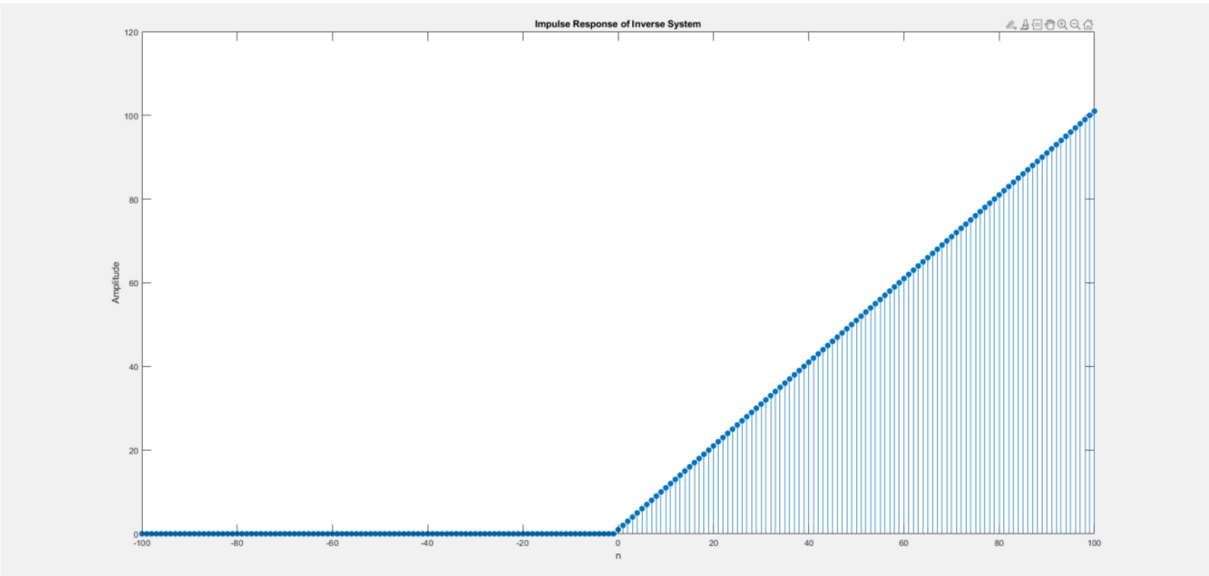


figure 21 : Impulse response of Inverse system

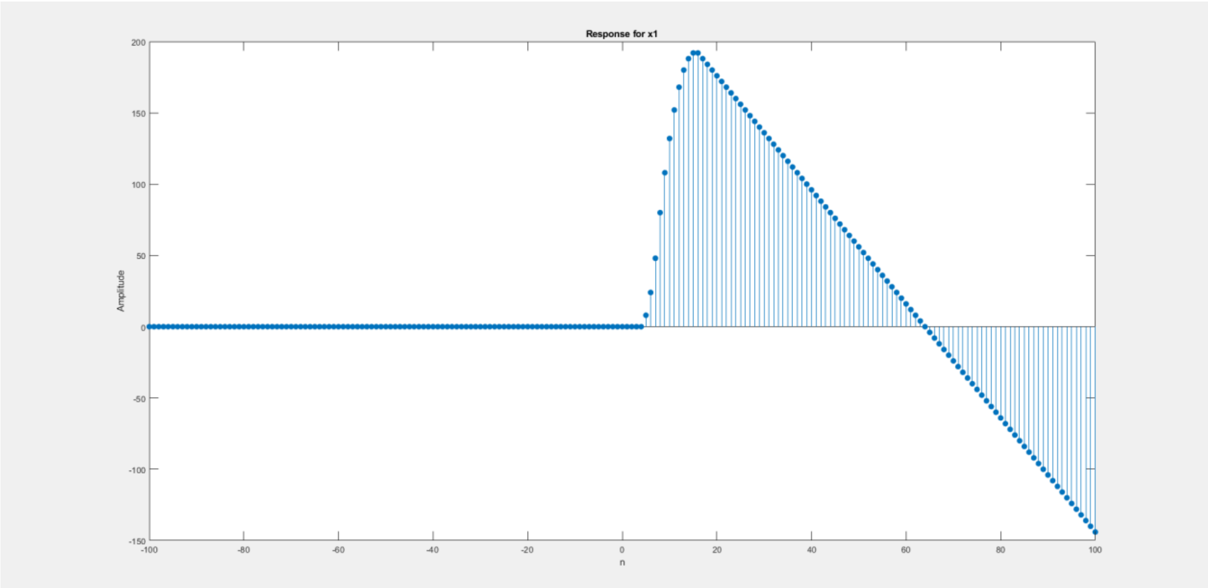


figure 22 : Response of X1 on the Inverse system

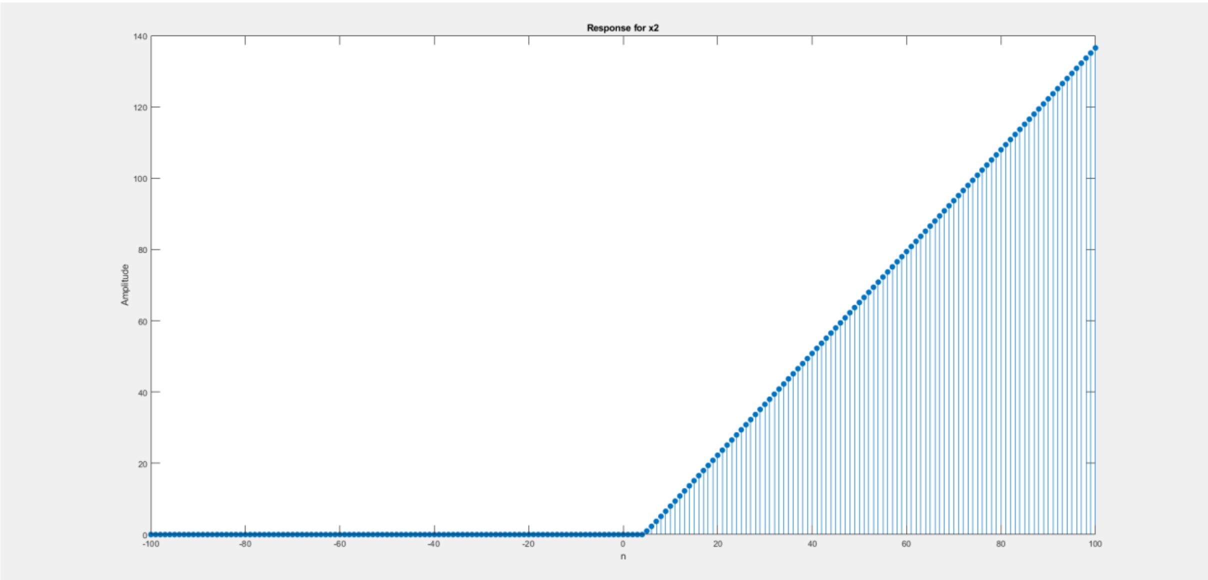


figure 23 : Response of X2 on the Inverse System

```
>> lab3_4_2
The inverse system is causal.
The inverse system has memory.
The system is Invertible
```

figure 24 : System Result for Part 4.2

Conclusion

In this lab, we examined the properties of the ideal integrator. We checked the properties of the system such as air linearity stability and looked at its responses according to different imputs. We discretized the system we examined and performed our operations in the discrete time domain. We saw that the BIBO stability of the system was not a problem in the ideal integrator but was not a problem for the exponential. We observed how the difference between the air results of the two systems behaved according to the changing N values. Then, we examined the counterpart of the derivative operator in the discrete time domain. We obtained the second derivative by using the approach used here. In the last part, we checked whether the system we obtained was invertible and plotted the results.

Codes

Part 1.3: Discretization of the Two Systems

```
%Sampling Period
Ts = 0.01;
Tt = -5:Ts:5;

%define sequence responses

dirdelta = zeros(size(Tt));
dirdelta(Tt==0) = 1;

u = @(t) double(t>=0);

h_t = cumsum(dirdelta); %cumulative sum of dirac delta
s_t = cumsum(u(Tt)); %cumulative sum of heaviside step func



figure;

subplot(2,1,1);
stem(Tt, h_t);
title('Impulse Response h(t)');
xlabel('Time (t)');
ylabel('h(t)');
grid on;
subplot(2,1,2);
stem(Tt, s_t);
title('Unit-Step Response s(t)');
xlabel('Time (t)');
ylabel('s(t)');
grid on;

%--for part 2
figure ;
n = 0;
```

```
for i = 1:5
 f_1 = exp(-Tt * 0.05*n) .* u(Tt);
 sgtitle('Discrititzed Responses');
 subplot(5,2,2*i-1);
 stem(Tt, f_1);
 title(['Discritized impulse response with a =',num2str(n.*0.05)]);
 xlabel('n');
 ylabel('h[n]');
 f_2 = ((1-exp(-Tt * 0.05*i))/(0.05*i)) .* u(Tt);
 subplot(5,2,2*i);
 stem(Tt, f_2);
 title(['Discritized step response with a =',num2str(i.*0.05)]);
 xlabel('n');
 ylabel('h[n]');
 n = n+1;

end
```

sumElements

```
function [sum_array] = sumElements(h, N_range)

    sum_array = zeros(size(N_range));
    h_mid = floor(length(h)/2)+1;

    for i = 1:length(N_range)
        N = N_range(i);
        N_min = max(1, h_mid -N);
        N_max = min(length(h),h_mid +N);
        sum_array(i) = sum(abs(h(N_min:N_max)));
    end
end
```

Part 2

```
N_range = 100:300:10000;
t = -10000:0.001:10000;

u = @(t) double(t>=0);
a = [0,0.05,0.10,0.25,0.5];
disp(a(1));
n=0;
for i = 1:length(a)
sgtitle("BIBO Stability Test");


subplot (5, 1, i);
f_f = exp(-t * (a(i)));
h = f_f .* u(t);
y = sumElements(h, N_range);
stem(N_range, y);

if i == 1
    title("Discrete version of the Ideal Integrator 'a = 0' ");
else
    title(['a =',' ',num2str(a(i))]);
end

n = n+1;
```

```
end
```

Part 3

```
% Define the input sequences
a = [0, 0.05, 0.1, 0.25, 0.5];
n = 0:50;


%unit step function
u = @(n) double(n>=0);
%define sequences
x1 = @(n) 8*(u(n) - u(n-4)) - 4*(u(n-4) - u(n-13)); %x1(t)
x2 = @(n) (0.3).^n .* u(n); %x2(t)

y_1 = zeros(length(n) + length(n) - 1, length(a));
y_2 = zeros(length(n) + length(n) - 1, length(a));

for i = 1:length(a)
    h = exp(-a(i) * n) .* u(n);
    y_1(:,i) = conv(x1(n),h,'full');
    y_2(:,i) = conv(x2(n),h,'full');
end

E1 = zeros(size(y_1, 1), length(a) - 1);
E2 = zeros(size(y_2, 1), length(a) - 1);


s = 1;
for i = 2:length(a)
    E1(:, i-1) = y_1(:, 1) - y_1(:, i);
    E2(:, i-1) = y_2(:, 1) - y_2(:, i);


end

figure;
stem(n, y_1(1:length(n), 1), 'filled');
title('y1[n] for a = 0');
xlabel('n');
ylabel('Amplitude');
figure;
stem(n, y_2(1:length(n), 1), 'filled');
title('y2[n] for a = 0');
xlabel('n');
ylabel('Amplitude');



figure;
for i = 1:length(a)-1
 subplot(size(E1, 2), 1, i);
 stem(n, E1(1:length(n), i), 'filled');
 title(['E1 for a =',' ',num2str(a(i+1))]);
 xlabel('n');
 ylabel('Difference');
end
```

```
figure;
for i = 1:length(a)-1
 subplot(size(E2, 2), 1, i);
 stem(n, E2(1:length(n), i), 'filled');
 title(['E2 for a =',' ',num2str(a(i+1))]);
 xlabel('n');
 ylabel('Difference');
end
```

Part 4

Part 4.1: First- and Second-Order Differentiation

```
a = 1;
t = -10:1:10;
u = @(n) double(n>=0);
func_coeff = [1, -2, 1];

%Dirac delta func shift operations
dirdelta = zeros(size(t));
dirdelta(t==0) = 1;

f_1 = filter(func_coeff, a, dirdelta);

y_out = conv(dirdelta, func_coeff);

figure;
stem(t, y_out(1:length(t)), 'filled');
title('Impulse Response Manually');
xlabel('n');
ylabel('h[n]');
ylim([-2.1 1.2]);
xlim([-1 10]);

figure;
stem(t, f_1, 'filled');
title('Impulse Response in Matlab');
xlabel('n');
ylabel('h[n]');
ylim([-2.1 1.2]);
xlim([-1 10]);

figure;
N = 0:5;
f_sum = sumElements(f_1,N);

disp(f_sum);
stem(N, f_sum, 'filled');
title('BIBO STABILITY');
xlabel('N');
ylabel('SUM');


figure; %Responses for x1 and x2

n= 0:50;
x1 = @(n) 8*(u(n) - u(n-4)) - 4*(u(n-4) - u(n-13)); %x1(t)
```

```
x2 = @(n) (0.3).^n .* u(n); %x2(t)
y_1 = convFUNC(x1(n),func_coeff);
y_2 = convFUNC(x2(n),func_coeff);


stem(0:length(y_1)-1, y_1, 'filled');
title('Response for x1');
xlabel('n');
ylabel('Amplitude');

figure;
stem(0:length(y_2)-1, y_2, 'filled');
title('Response for x2');
xlabel('n');
ylabel('Amplitude');
```

Part 4.1: Invertibility of Second-Order Difference

```
clear;


t = -100:1:100;
u = @(n) double(n>=0);
%Dirac delta func shift operations
dirdelta = zeros(size(t));
dirdelta(t==0) = 1;


func_coeff = [1, -2, 1];
%func_coeff = @(x,n) x(n) - 2*x(n-1) + x(n-2);


h_inv = cumsum(cumsum(dirdelta));

f_i =  conv(func_coeff, h_inv,'same');



h_inv = cumsum(cumsum(dirdelta));




if f_i == dirdelta(101:103)
    Invertibility = "Invertible";
else
    Invertibility = "Not Invertible";
end


n = -5:1:20;
%n= 0:50;
x1 = @(n) 8*(u(n) - u(n-4)) - 4*(u(n-4) - u(n-13)); %x1(t)
x2 = @(n) (0.3).^n .* u(n); %x2(t)
y_11 =  convFUNC(x1(n), h_inv);
y_22 =  convFUNC(x2(n), h_inv);

figure;
stem([0:1:2],f_i);
```

```
title('Convolution Result (System * Inverse)');
xlim([-10 10]);

figure;
stem(t, h_inv, 'filled');
title('Impulse Response of Inverse System');
xlabel('n');
ylabel('Amplitude');



figure;
stem(t, y_11(1:length(t)), 'filled');
title('Response for x1');
xlabel('n');
ylabel('Amplitude');

figure;
stem(t, y_22(1:length(t)), 'filled');
title('Response for x2');
xlabel('n');
ylabel('Amplitude');

    if all(h_inv(t < 0) == 0)  % Negatif zamanlı bileşenler sıfır mı?
        fprintf('The inverse system is causal.\n');
    else
        fprintf('The inverse system is not causal.\n');
    end


    if sum(h_inv ~= 0) == 1  % Sadece bir yerde sıfırdan farklı mı?
        fprintf('The inverse system is memoryless.\n');
    else
        fprintf('The inverse system has memory.\n');
    end

A = "The system is";
fprintf('%s %s\n', A, Invertibility);
```