Full Name: Orkun İbrahim Kök

Department: EEE

Course Code: EEE321

Section: 02

Experiment Number: 03

Date: 23.03.2024

Lab 3 Report

Introduction:

The purpose of this lab is examining the system properties of the ideal and non-ideal (imperfect) integrators and analyzing their integration performance.

Analysis:

Part 1.1 Ideal (Perfect) Integrator

The system property derivations can be found at the end of this report.

Part 1.2 Another System

The system property derivations for continuous time linear time invariant system can also be found at the end of the report.

Part 1.3 Discretization of the Two Systems

In this part, above two system is discretized with a sampling interval T_S=0.01s. The input/output relation for ideal integrator is given in below equation (Eqn. 1).

$$y[n] = y[n-1] + T_S * x[n] \qquad \text{(Eqn. 1)}$$

This system is called as “accumulator” (discrete-time integrator). The impulse response h[n] and unit-step response s[n] of the accumulator is given below.

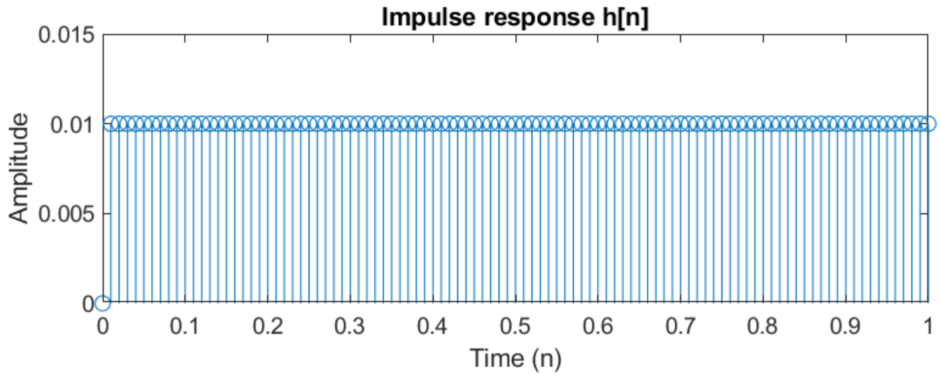


Figure 1: impulse response of accumulator

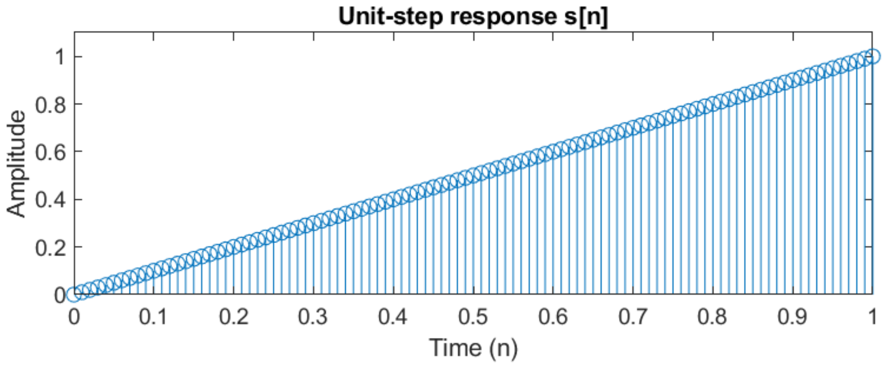


Figure 2: unit-step response of accumulator

The discretized version of the second system is called "discrete-time exponential decay filter". The impulse response h[n] and unit-step response s[n] of the system is given below.

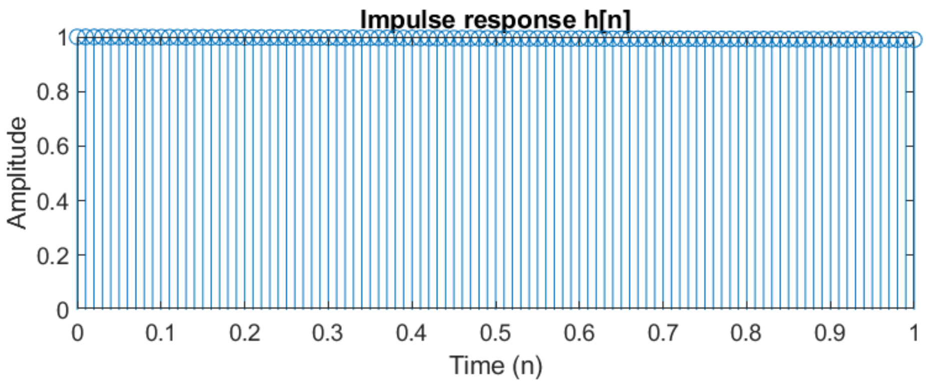


Figure 3: impulse response of discrete time exponential decay filter

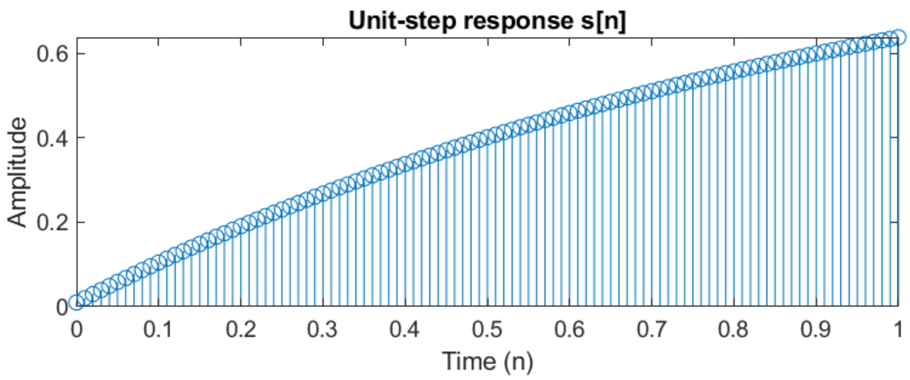


Figure 4: unit-step response of discrete time exponential decay filter

Part 2

In this part, BIBO stability of the system in Part 1.2 is considered. I wrote a MATLAB function called “sumElements” to observe the BIBO stability of the systems found in Part 1.3. For discrete-time exponential decay filter system, five a values (0,0.05,0.10,0.25, and 0.5) are considered. The behavior of the system for five different a values can be observed in the figure below (Figure 5).

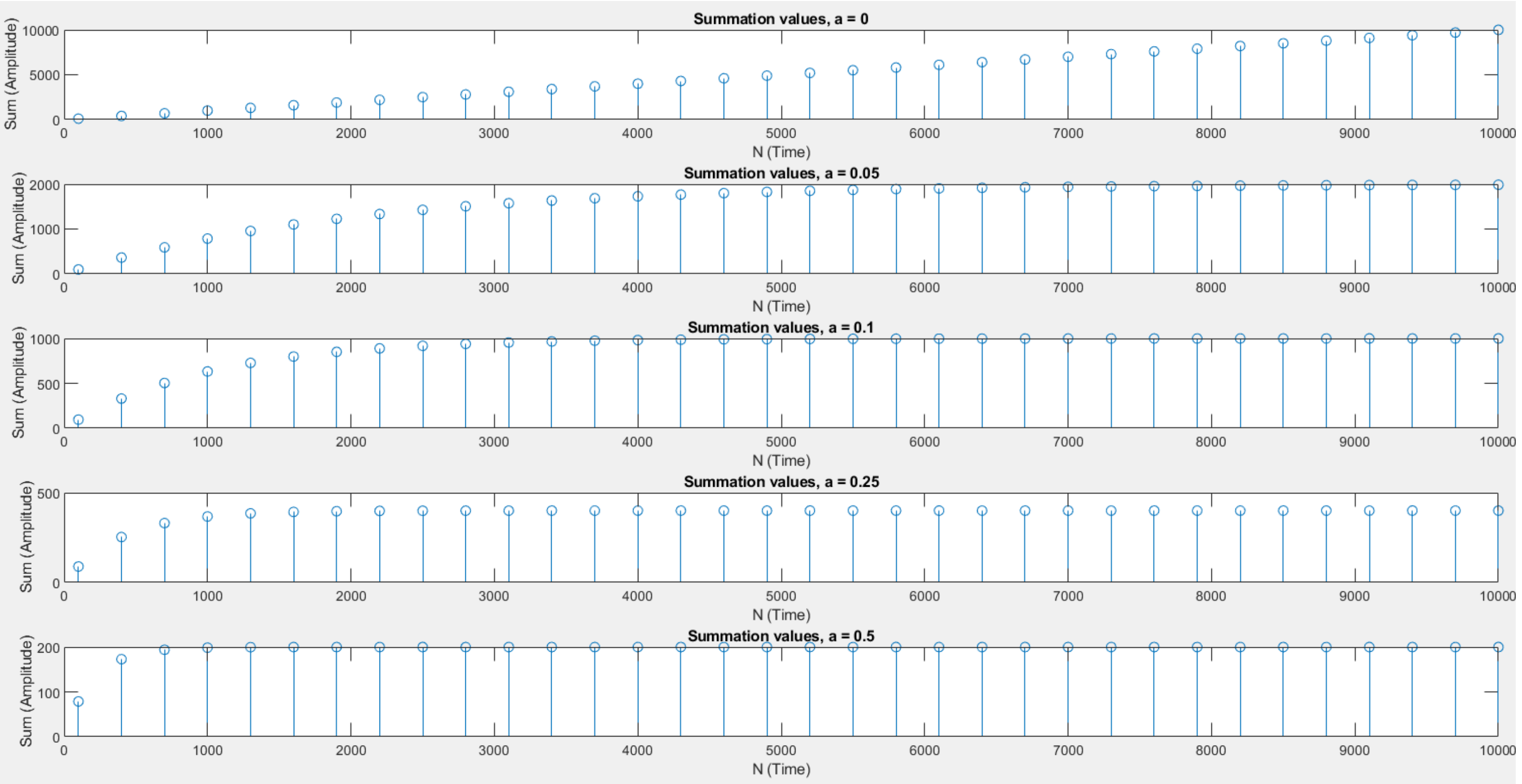


Figure 5: sumElements for discrete-time exponential decay filter

As seen from the plots in (Figure 5), as “a” value increases, the magnitude of the sum decreases substantially, because it is a discrete-time exponential decay filter system.

Part 3

For this part, difference between the outputs of the two systems will be analyzed. Two input sequences are given as:

$$x_1[n] = 8(u[n] - u[n\ -4]) - 4(u[n\ -4] - u[n\ -13])$$

$$x_2[n] = (0.3)^n\, u[n]$$

The difference between the outputs for a=0 (denoted as $y_1^{ideal}[n]$ and $y_2^{ideal}[n]$) and the outputs for the remaining four cases of a (denoted as $y_1[n]$ and $y_2[n]$) can be found by the following equations.

$$\varepsilon_1[n] = |y_1{}^{ideal}[n] - y_1[n]|$$
$$\varepsilon_2[n] = |y_2{}^{ideal}[n] - y_2[n]| \qquad \text{(Eqn. 2)}$$

The differences between the output for 5 different a values are given in the figures below. (Figure 6) belongs to $x_1[n]$ and (Figure 7) belongs to $x_2[n]$.

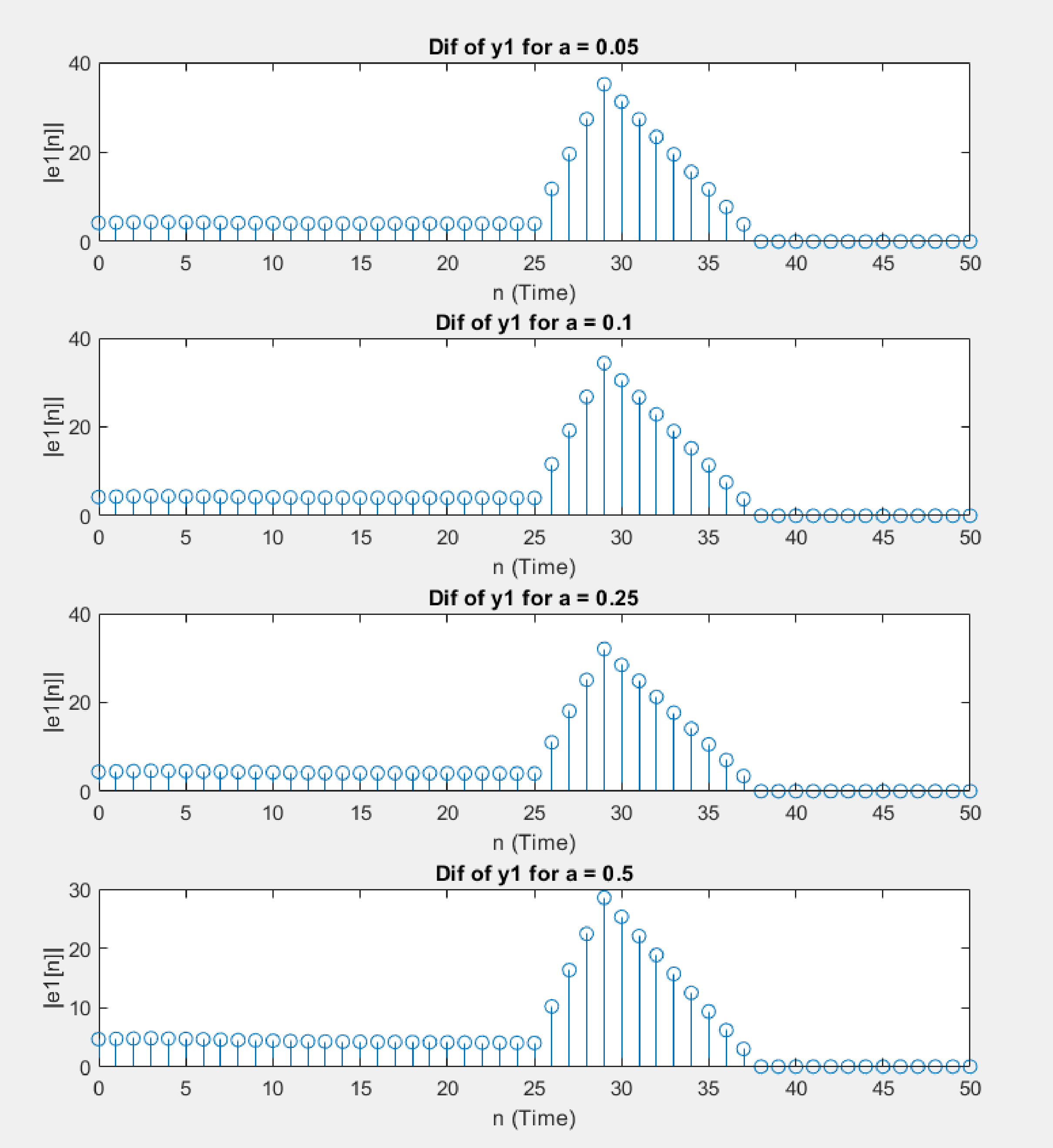


Figure 6: $x_1[n]$ input sequence plots for different a values

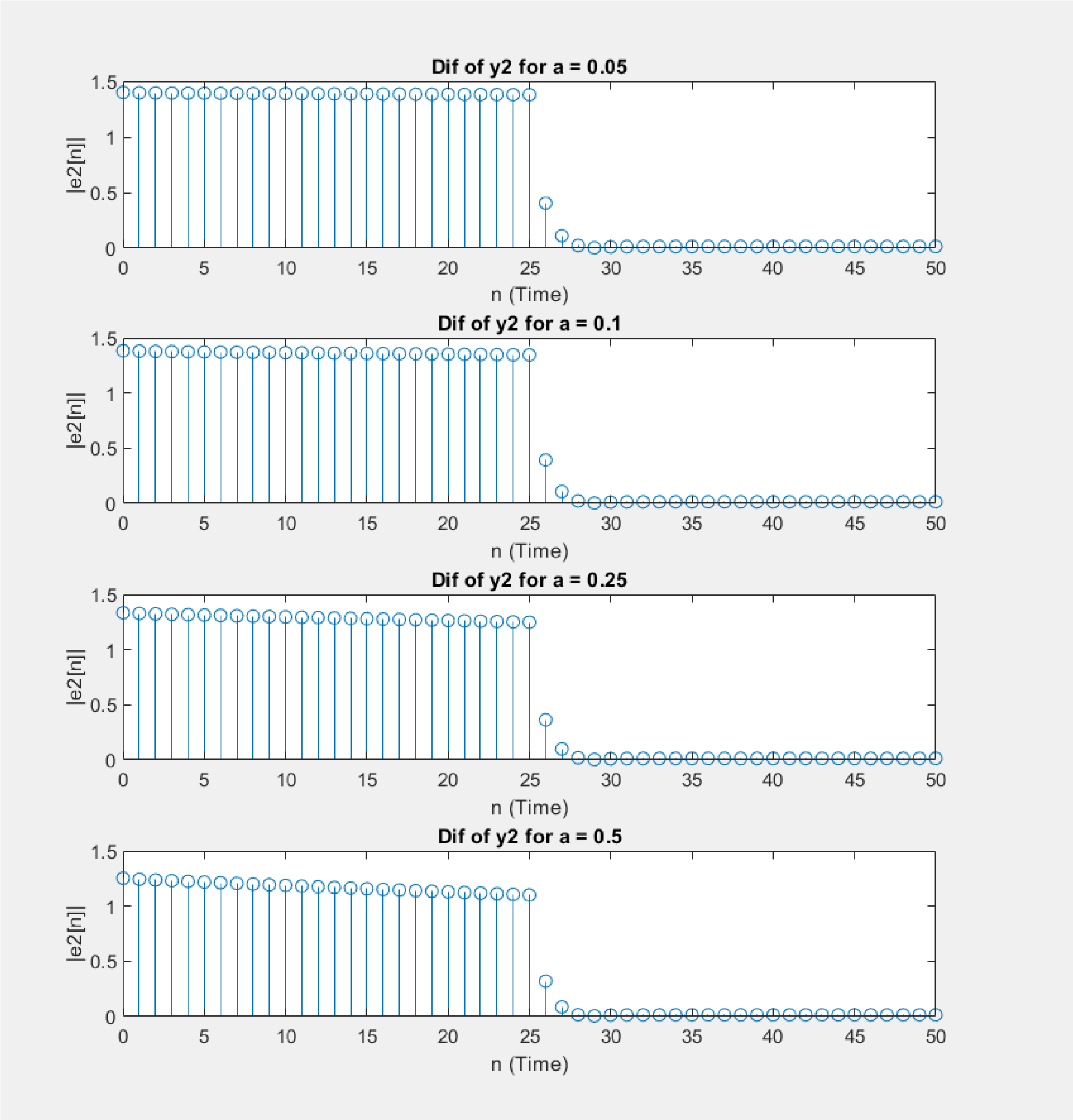


Figure 7: x2[n] input sequence plots for different a values

In both figures (Figure 6 and Figure 7), as a values increase, the amplitude of $\varepsilon_1[n]$ and $\varepsilon_2[n]$ decreases.

Part 4.1 First- and Second-Order Differentiation

The derivation for this part can also be found at the end of this report. Using backward approximation, second order difference system is obtained (Eqn. 3).

$$y[n] = \delta[n] - 2\delta[n-1] + \delta[n-2] \qquad \text{(Eqn. 3)}$$

This system is causal, has memory, BIBO stable, and FIR. As said, explanations can be found at the end of this report.

The manually derived impulse response of the system is given below.

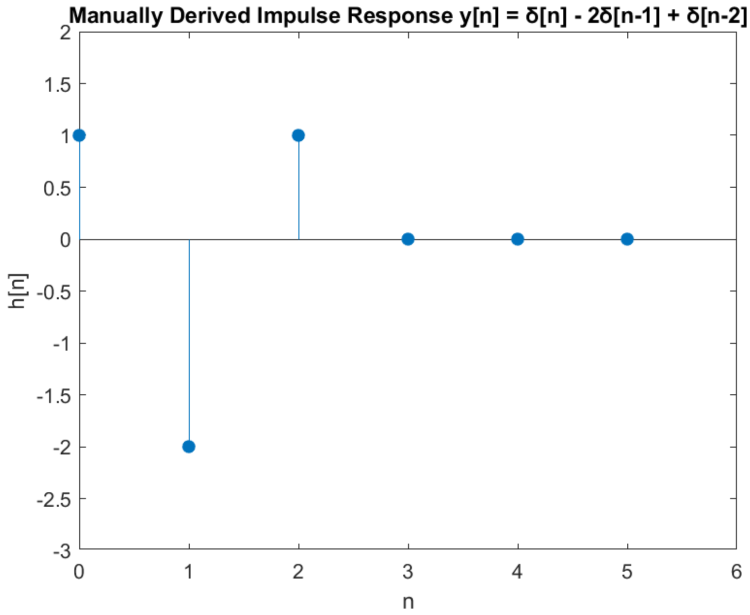


Figure 8: Manually derived impulse response of the system

Impulse response of the input signal is given below.

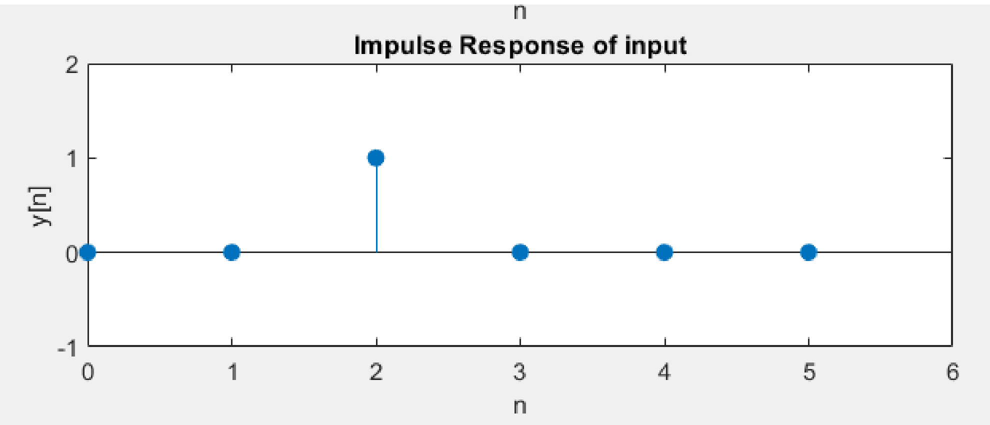


Figure 9: Impulse response of the input

Looking at the output of the matlab code, system is turned out to be BIBO stable. Also, by looking at the plot, it can be seen that the inputs and outputs are finite, which shows us that the signal is BIBO stable.

4.2 Invertibility of Second-Order Difference

The derivation of inverse system and its properties are given at the end of the report. The inverse of the signal is called as “ramp signal”. The convolution of second order difference system and its inverse gives us the delta function (Figure 10).

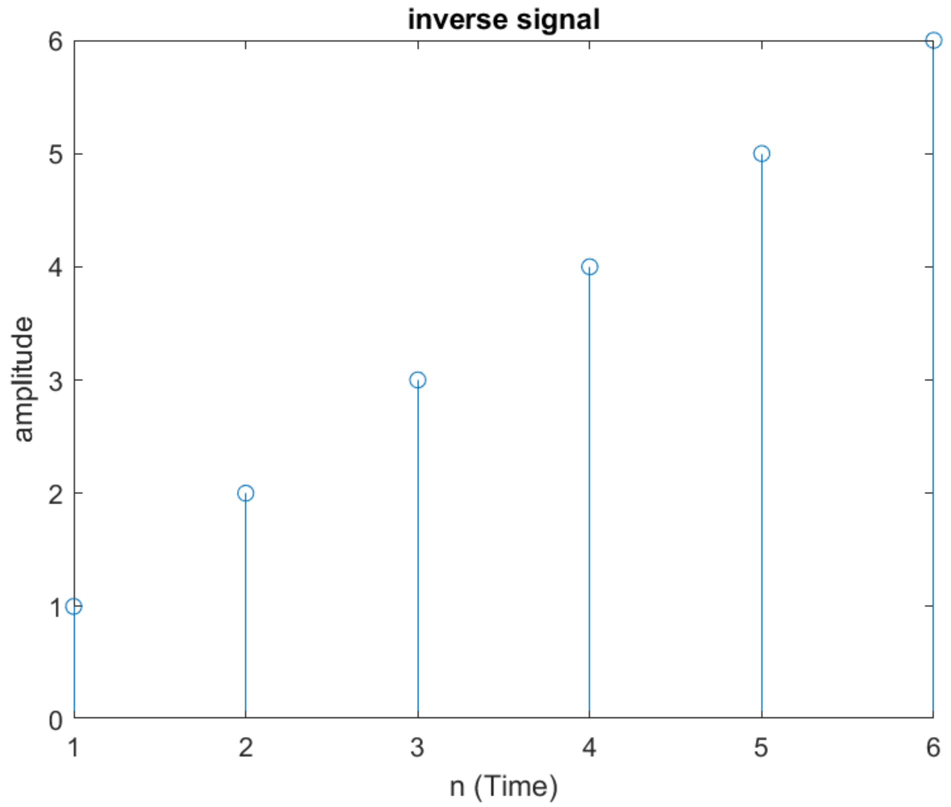


Figure 10: inverse signal plot

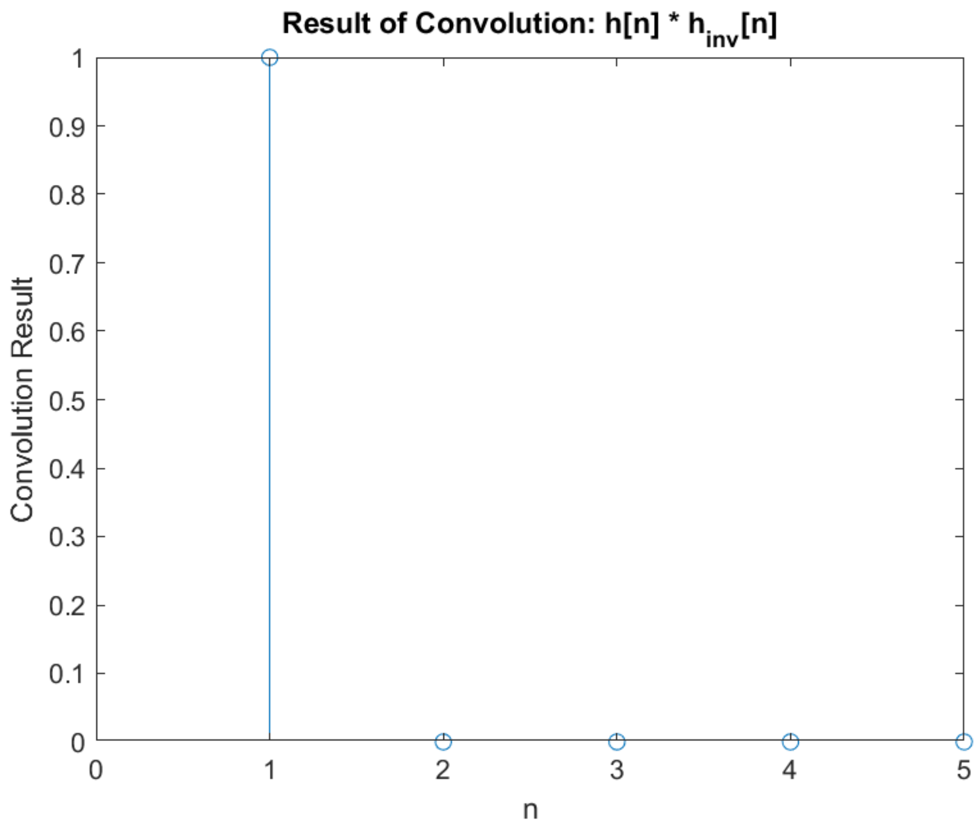


*Figure 11: h*h_inv convolution*

Part 1.3.1

```
Ts = 0.01;
n = 0:Ts:1;
% y[n] = y[n-1] + Ts * x[n] (discrete version of ideal integrator)
% discrete-time integrator or accumulator
h = zeros(size(n));
d = zeros(size(n));
d(1) = 1;

s = zeros(size(n));

for i = 2:length(n)
    h(i) = h(i-1) + Ts * d(i-1);
    s(i) = s(i-1) + Ts * 1;
end

subplot(211)
stem(n, h)
title('Impulse response h[n]')
xlabel('Time (n)')
ylabel('Amplitude')
ylim([0 0.015])

subplot(212)
stem(n, s)
title('Unit-step response s[n]')
xlabel('Time (n)')
ylabel('Amplitude')
ylim([0 1.1])
```

Part 1.3.2

```
Ts = 0.01;
n = 0:Ts:1;
a = 1;
% y[n] = e^-(a*n*t)*u[n]
h = exp(-a * n * Ts) ;
s = filter(Ts, [1, -(1 - Ts*a)], ones(1, 101));
subplot(211)
stem(n, h)
title('Impulse response h[n]')
xlabel('Time (n)')
ylabel('Amplitude')

subplot(212)
stem(n, s)
title('Unit-step response s[n]')
xlabel('Time (n)')
ylabel('Amplitude')
```

Part 2

```
a = [0 0.05 0.1 0.25 0.5];
Ts = 0.01;
N_range = 100:300:10000;
sum_arrays = zeros(length(a), length(N_range));
```

```
for i = 1:length(a)
    a_values = a(i);
    if a_values == 0
        h = ones(1, max(N_range) + 1);
    else
        h = exp(-(a_values) * (0:max(N_range)) * Ts);
    end
    subplot(5, 1, i);
    stem(N_range, sumElements(h, N_range));
    xlabel('N (Time)');
    ylabel('Sum (Amplitude of signal)');
    title(['Sum values, a = ', num2str(a_values)]);
end
```

sumElements

```
function sum_array = sumElements(h, N_range)
    sum_array = zeros(size(N_range));%create an empty array
    for i = 1:length(N_range)
        N = N_range(i);%select N values
        sum_result = 0;
        for k = -N:N %determine the limits of the sum operation
            if k >= 1 && k <= numel(h)%calculates the total number of elements
                sum_result = sum_result + h(k);
            end
        end
        sum_array(i) = sum_result; %store results in the array called sum_array
    end
end
```

Part 3

```
Ts = 0.01;
n = 0:50;
u = @(n) double(n >= 0);
a_values = [0, 0.05, 0.10, 0.25, 0.5];
%Given input sequences in the lab manual
x1 = 8 * (u(n) - u(n-4)) - 4 * (u(n-4) - u(n-13));
x2 = (0.3).^n .* u(n);

ideal_y1 = Ts * cumsum(x1);
ideal_y2 = Ts * cumsum(x2);

figure;
for i = 2:length(a_values)
    a = a_values(i);
    h = exp(-a * n * Ts);  %second given signal
    y1 = conv(x1, h, 'same');  % Convolve input with impulse response
    y2 = conv(x2, h, 'same');

    e1 = abs(ideal_y1 - y1);
    e2 = abs(ideal_y2 - y2);

    subplot(length(a_values)-1, 2, 2*i-3);
    stem(n, e1);
    title(['Dif of y1 for a = ', num2str(a)]);
    xlabel('n (Time)');
    ylabel('|e1[n]|');
```

```
    subplot(length(a_values)-1, 2, 2*i-2);
    stem(n, e2);
    title(['Dif of y2 for a = ', num2str(a)]);
    xlabel('n (Time)');
    ylabel('|e2[n]|');
end
```

Part 4.1

```
n = 0:5;
%create impulse signal
len = 5;
impulse_index = 1;
impulse = zeros(1, len);
impulse(impulse_index) = 1;
y = zeros(1, length(n));
for i = 3:length(n)
    y(i) = x(i) - 2*x(i-1) + x(i-2);
end

%second order differnece equation = δ[n] - 2δ[n-1] + δ[n-2]
h = [1, -2, 1, zeros(1,3)];

% Plot the manually derived impulse response
subplot(211)
stem(n, h, 'filled');
title('Manually Derived Impulse Response y[n] = δ[n] - 2δ[n-1] + δ[n-2]');
xlabel('n');
ylabel('h[n]');
xlim([0 5]);
ylim([-3 2]);


subplot(212)
stem(n, y, 'filled');
title('Impulse Response of input');
xlabel('n');
xlim([0 5]);
ylim([-3 2]);
ylabel('y[n]');

%check bibo stability
BIBO_stability = sum(abs(h)) < Inf;

if BIBO_stability
    disp('BIBO stable.');
else
    disp('not BIBO stable.');
end
```

Part 4.2

```
% Given impulse response
h = [1, -2, 1 0 0 0];
```

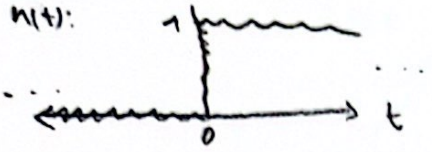
```matlab
% Inverse of the signal
h_inv = [1 2 3 4 5 6];
figure;
stem(h_inv)
title("inverse signal")
ylabel('amplitude')
xlabel("n (Time)")

figure;
stem(conv(h, h_inv));
title('Result of Convolution: h[n] * h_{inv}[n]');
xlabel('n (Time)');
ylabel('Convolution Result');
xlim([0 5])
```

Orkun İbrahim Kök

Part 1.1)

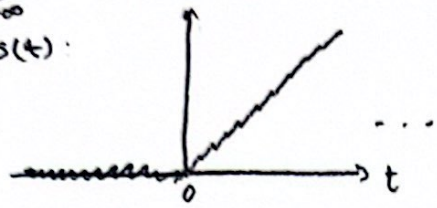
$$\int_{-\infty}^{t} \delta(\tau)\, d\tau = h(t) = u(t)$$

h(t):



$$\int_{-\infty}^{t} u(\tau)\, d\tau = s(t) = \text{ramp signal}$$

s(t):



Linearity:

$$\int_{-\infty}^{t} [a x_1(\tau) d\tau + b x_2(\tau) d\tau] \overset{?}{=} \int_{-\infty}^{t} a x_1(\tau) d\tau + \int_{-\infty}^{t} b x_2(\tau) d\tau$$

using the distributive property of integral and assuming $x_1(t)$ and $x_2(t)$ are continuous signals, we can say that ideal integrator is linear.

Time - Invariance:

say $x_2(t) = x_1(t - t_0) \longmapsto y_2(t)$

$$= \int_{-\infty}^{t} x_2(\tau)\, d\tau = \int_{-\infty}^{t} x_1(\tau - t_0)\, d\tau$$

apply change of variables:

$\tau' = \tau - t_0 \Rightarrow \tau = \tau' + t_0$ and $d\tau = d\tau'$

$$\Rightarrow y_2(t) = \int_{-\infty}^{t-t_0} x_1(\tau' + t_0 - t_0)\, d\tau' = \int_{-\infty}^{t-t_0} x_1(\tau')\, d\tau' = y_1(t - t_0)$$

Because $y_1(t-t_0) = y_2(t)$, we can conclude that ideal integrator is time-invariant.

Causality:

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

Because integration uses only past and present values, and do not depend on any future value, we can say that the ideal integrator system is causal.

can also be shown as follows:

$y(t) = \int_{-\infty}^{t} \delta(\tau)\, d\tau \Rightarrow y(t) = 0$ for $t < 0 \Rightarrow$ causal

Memory:

$y(t)=\int_{-\infty}^{t} x(\tau)\,d\tau$ => Because this system data from past, it has memory.

BIBO stability:

say $|x(t)| \leq B$ => $y(t)=\int_{-\infty}^{t} x(\tau)\,d\tau$ => $y(t) \leq B \cdot \underbrace{\int_{-\infty}^{t} d\tau}$

$\tau \big|_{-\infty}^{t} = t-(-\infty)$

$= \underline{t+\infty}$

not finite

Because $y(t)$ is not finite,

$y(t)=\int_{-\infty}^{t} x(\tau)\,d\tau$ is not BIBO stable.

Orhun İbrahim Kşk

Part 1.2)

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

$y(t) = e^{-at} \cdot u(t) * u(t)$

$y(t) = \int_{-\infty}^{t} e^{-a\tau} \cdot u(t-\tau)\, d\tau$

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

Linearity:

$k \cdot y_1(t) + m \cdot y_2(t) \overset{?}{=} k \cdot x_1(t) + m \cdot y_2(t)$

$k\left(\frac{-1}{a}\left[e^{-at_1} - 1\right]\right) + m\left(\frac{-1}{a}\left[e^{-at_2} - 1\right]\right) \overset{?}{=} k e^{-at_1} + m e^{-at_2}$

$\frac{k}{a} - \frac{m}{a} = (k+1) e^{-at_1} + (m+1) e^{-at_2}$

Because right and left hand sides of the equality are not same, we can say that this system is not linear.

Time - Invariance:

$y(t, t_0) = y(t)\big|_{t = t - t_0} = \frac{-1}{a}\left(e^{-a(t-t_0)} - 1\right)$

$y(t - t_0) = \frac{-1}{a}\left(e^{-a(t-t_0)} - 1\right)$

Because both equations are same of each other, we can say that this system is time-invariant.

Causality:

This system is causal because output does not depend on future values.

Memory:

System is memoryless

BIBO stability:

say $x(t) = e^{-at}$

$|x(t)| < B$

then $y(t) = \frac{-1}{a}\left(\underbrace{x(t)}_{<B} - 1\right) \Rightarrow y(t) < \overbrace{\frac{-1}{a}(B - 1)}^{\text{finite}}$

This system is BIBO stable.

Scanned with CamScanner

Part 4.1

Causality: Second order difference system uses only past and present values, therefore system is causal.

Memory: Since it uses values from the past, it has memory.

BIBO Stability: Looking at the figure for BIBO stability, it is seen that the system's outputs are finite, therefore system is BIBO stable.

System is FIR because impulse response of the second order difference system is finite, also impulse response settles to zero in a finite number of steps.

Part 1.2:

using deconv-demo, inverse of the second order difference equation y_{inv} is found as [1,2,3,4,5,6].

$y[n]$ = second order difference system:

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

This system has memory because it uses past values. Therefore, it is also causal for not using future inputs.