EEE 321

Report: Signals and Systems Lab 3

Yigit Turkmen – 22102518

March 23, 2024

Introduction:

In this lab, ideal and non-ideal integrators analyzed through theoretical analysis and MATLAB simulations. The study began with evaluating the ideal integrator's properties, then compared these with a second system characterized by an exponential impulse response, enhancing their understanding of system behaviors. The assignment progressed to practical exercises, including discretizing these systems and evaluating their stability with MATLAB. Additionally, I delved into first- and second-order differentiation, applying theoretical concepts in practical scenarios. This report aims to succinctly convey the insights and knowledge gained during the lab, showcasing a thorough grasp of the material.

Note that, since codes related to each parts will be added to the end of this report. I won't also add these codes to each part of this assignment while investigating them seperately.

Part 1

1.1 Ideal (Perfect) Integrator

For this part see the appendix for ease of use.

1.2 Another System

For this part see the appendix for ease of use.

1.3 Discretization of the Two Systems

See the figure 1 of plots below.

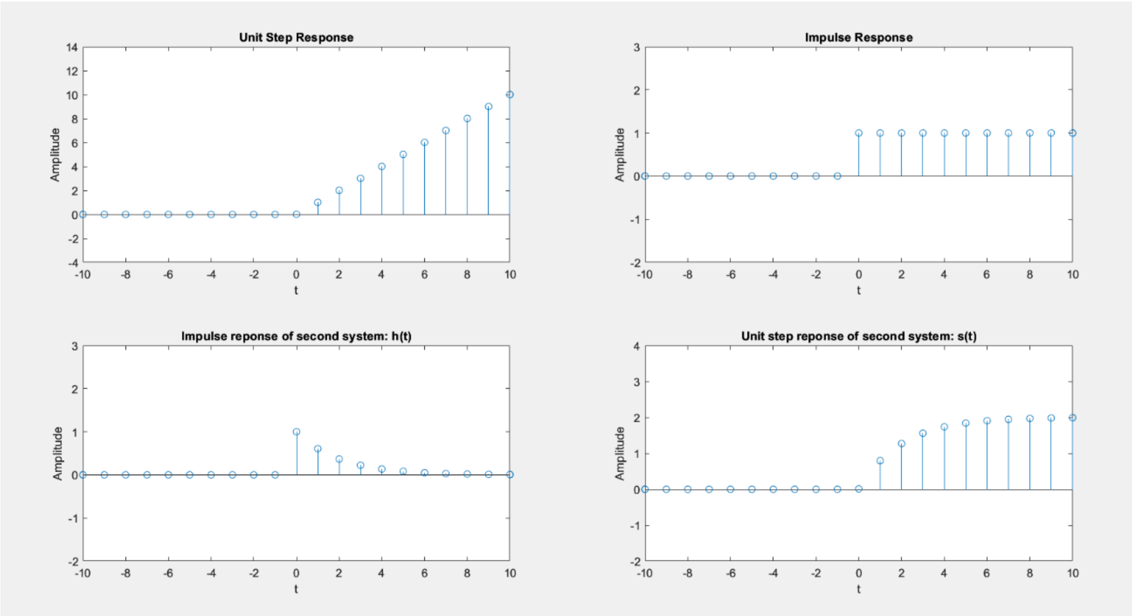


Figure 1: Plots of part 1.3 with associated titles for each of them

Question: What is this system called?

This system is commonly referred to as a discrete-time integrator or an accumulator because it accumulates the sum of the past input samples.

Input-Output equation:

$$\int_{-\infty}^{t} x(T)dT \rightarrow \sum_{n=-\infty}^{t} \sum_{m=n/Ts}^{n+1/Ts} x(m-Ts)Ts \rightarrow accumulator$$

Part 2

See the figure 2 of plots of the related part.

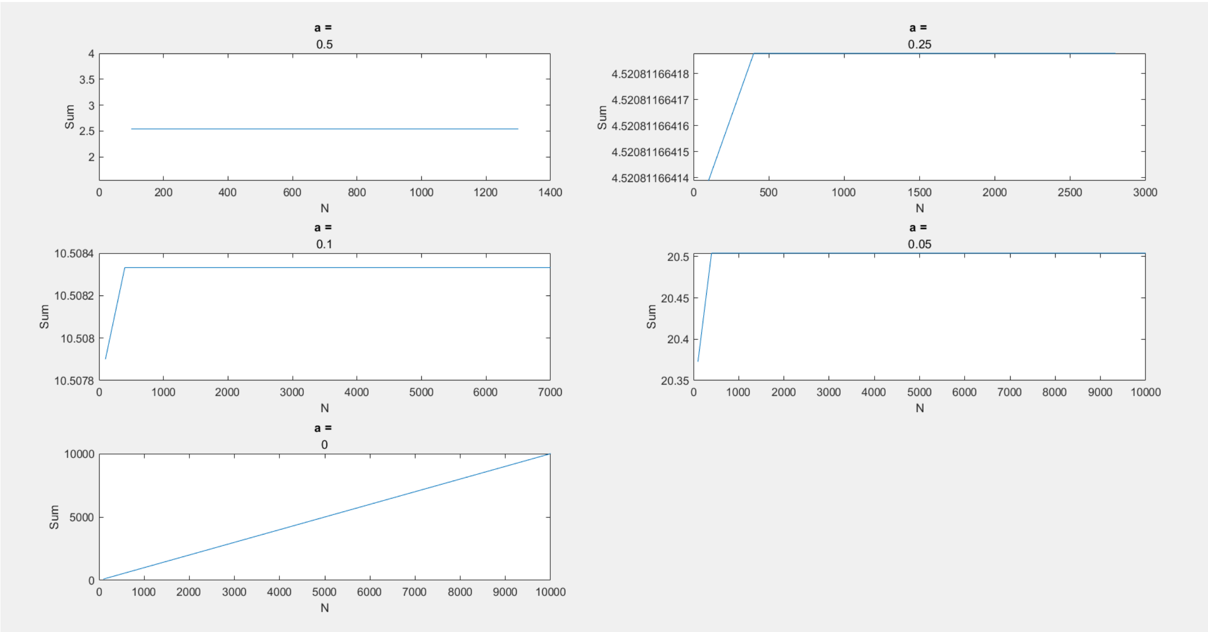


Figure 2: Plots of part 2 with associated titles =”a” for each of them

Comments on BIBO:

- For the exponentially decaying functions (*a*>0), as the input gets larger (i.e., larger *t*), the output approaches 0, indicating BIBO stability. The system can handle bounded inputs and produce bounded outputs.
- For the constant function (*a*=0), the output remains constant and therefore sum of it goes to infinity, indicating a not BIBO stability.

Part 3

See the figure 3-4-5 of plots of the related part.

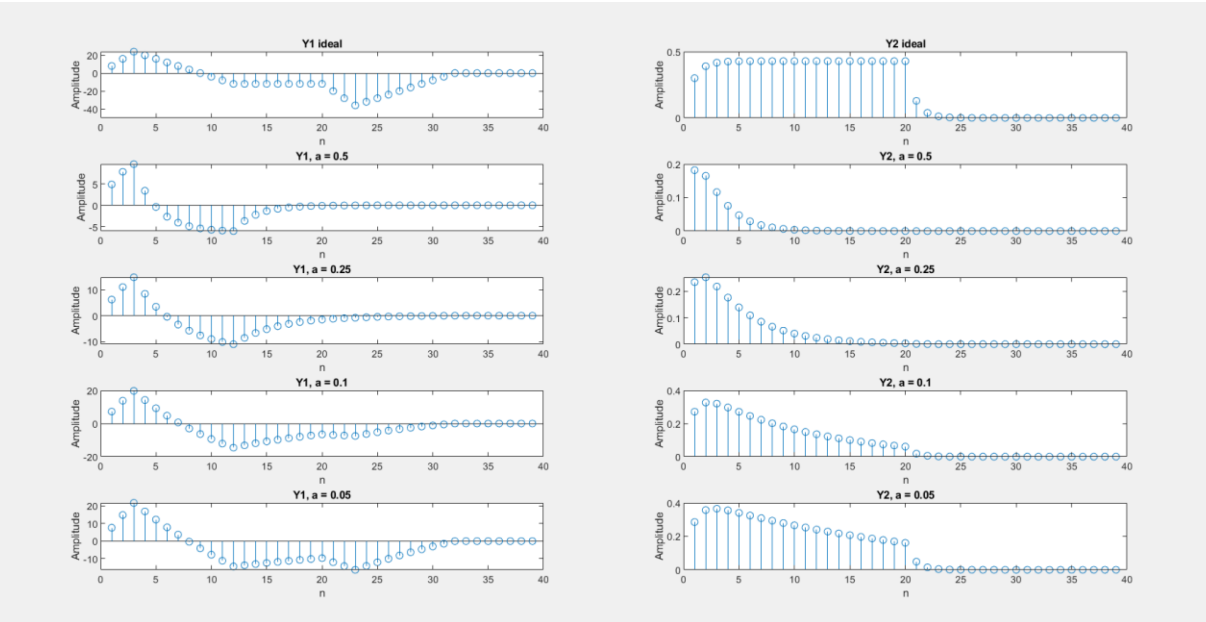


Figure 3: plots of Y1 and Y2 with related a values

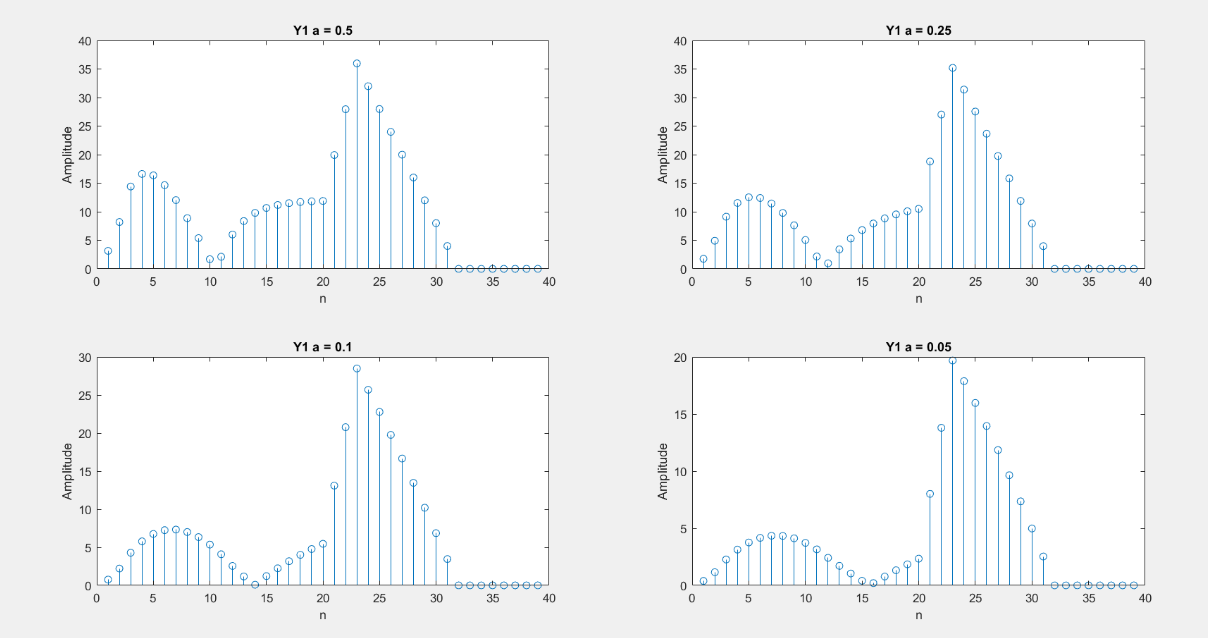


Figure 4: Difference plots of Y1 with related a values

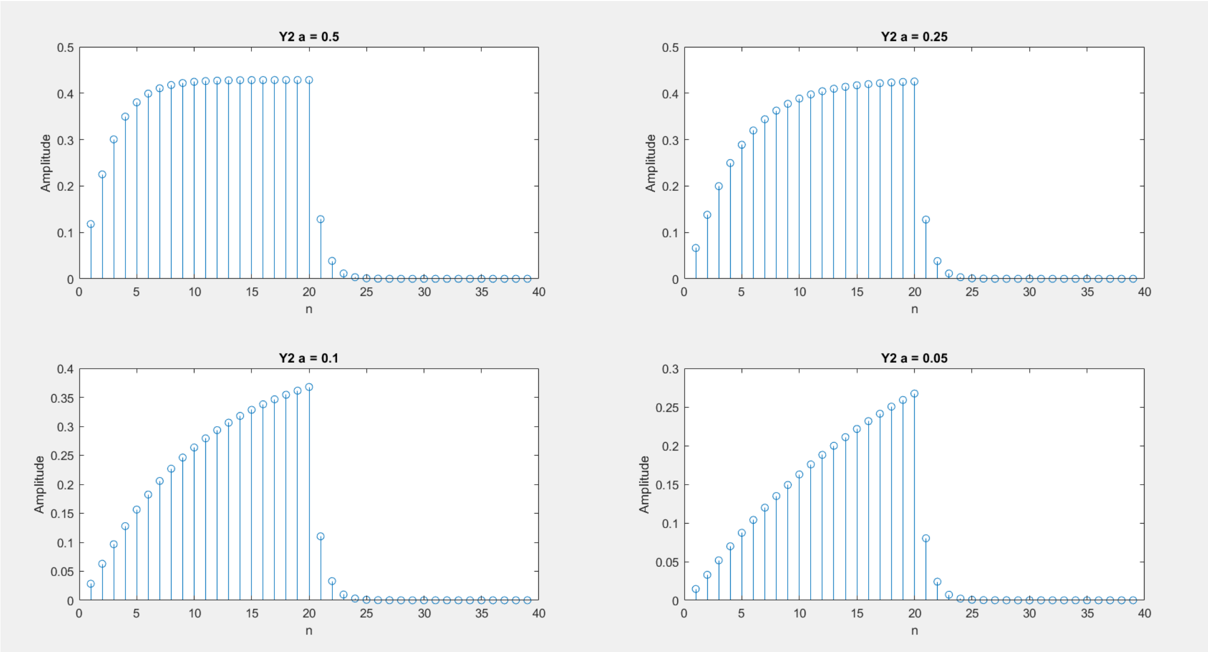


Figure 5: Difference plots of Y2 with related a values

Part 4

4.1 First- and Second-Order Differentiation

See the figure 6-7-8-9 of plots of the related part. Also see the picture below to answer the questions in this part.

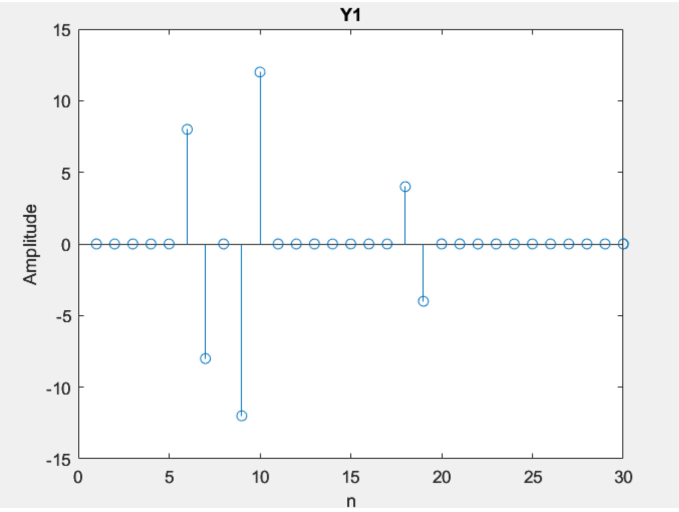


Figure 6: Y1 vs n

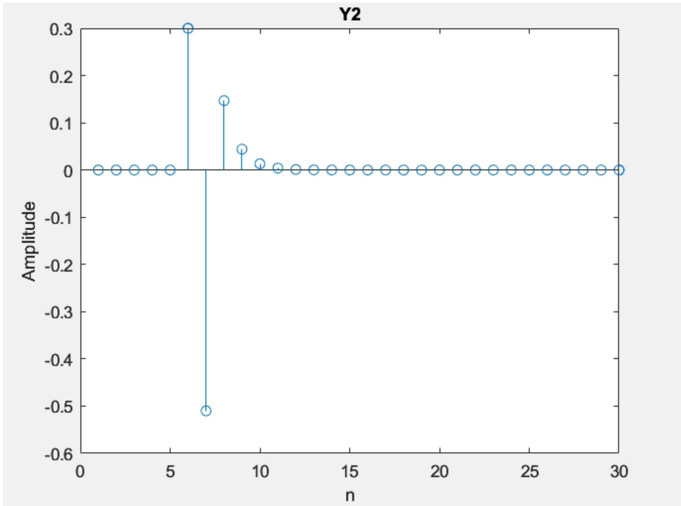


Figure 7: Y2 vs n

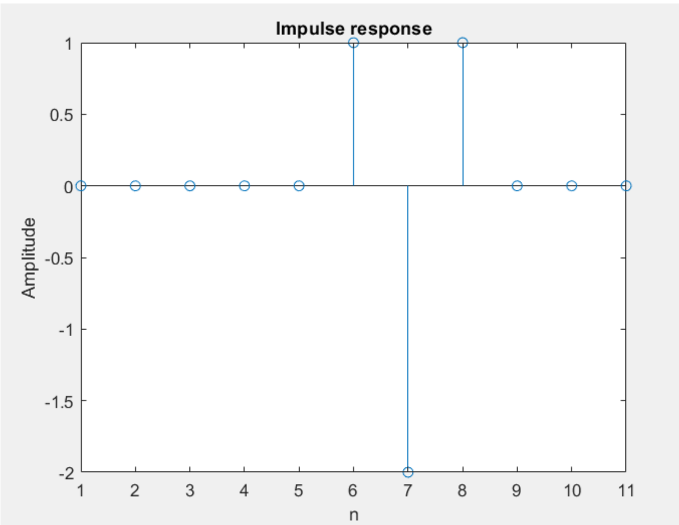


Figure 8: Impulse response vs n

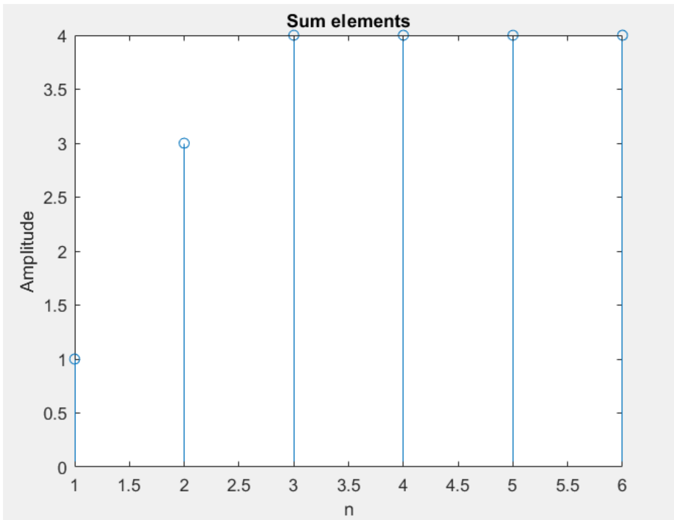


Figure 9: Sum elements vs n

Answers for the part 4.1:

Part 4.1

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

$z[n] = y[n] - y[n-1] = x[n] - 2x[n-1] + x[n-2]$

where $z[n]$ is the output of the second order system when input is x.

Causality

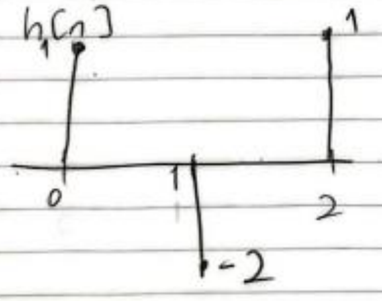
System is causal since it does not depend future x values.

Memory

System has memory since it depends $x[n-1]$ and $x[n-2]$.

For BIBO stability first find impulse response.

If we give delta δ to system as input, we can find impulse response. $h[n] = \delta[n] - 2\delta[n-1] + \delta[n-2]$



$\rightarrow$ same as Matlab plot.

BIBO stability $\rightarrow \sum_{-\infty}^{\infty} |h[n]| = 4 < \infty$, Therefore, it is BIBO stable.

Question: Is it a finite impulse response (FIR) or infinite impulse response (IIR) system?

It is a **FIR** system since the impulse response is finite and does not continue indefinitely.

4.2 Invertibility of Second-Order Difference

See the figure 10 and 11 of plots of the related part. Also see the below of this part to answer the questions in this part.

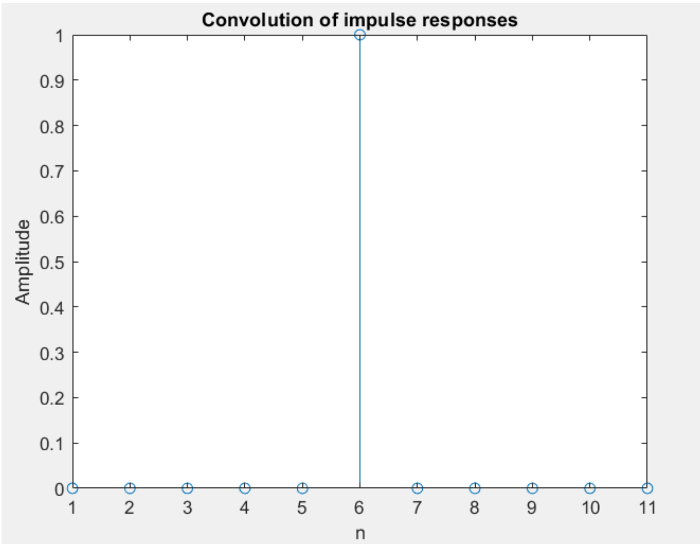


Figure 10: Plot of Convolution Of Impulse responses

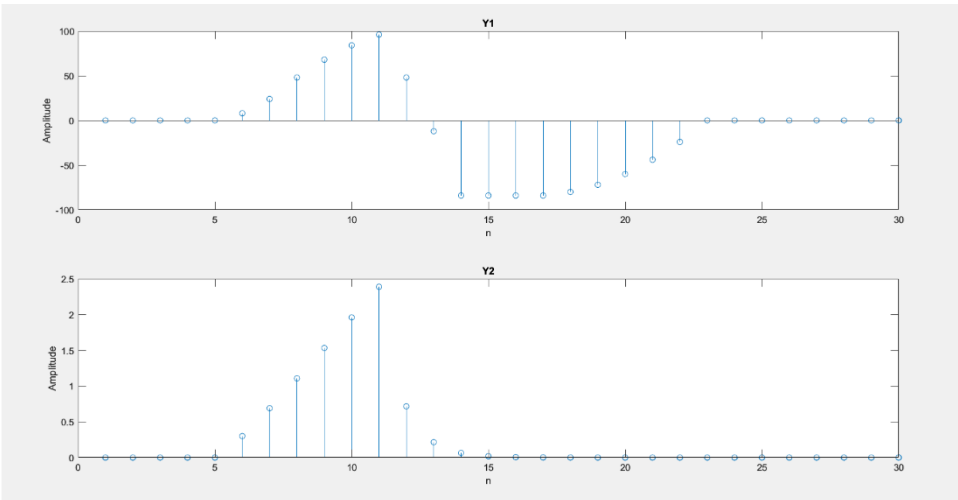


Figure 11: Y1 vs n and Y2 vs n

Answers for the part 4.2:

The system can be reversed. Reversing a first-order derivative equates to performing integration, akin to an accumulator system. To reverse a second-order derivative, one would need to apply the accumulator system twice in succession.

To determine the impulse response of the reversed system, one could use the accumulator twice in response to an impulse.

The equation u[n] represents the summation of the impulse function δ[k] from negative infinity to n. Similarly, r[n + 1] is the cumulative sum of u[k] from negative infinity to n.

The impulse response for the reversed system is a unit ramp function. Using this response, we can formulate the general equation for the system's input-output relationship:

The output y[n] can be computed by summing the product of the input x[n - k] and the factor (k + 1), starting from k equal to zero and continuing indefinitely.

The system is considered causal because its output relies solely on the current and previous inputs. It possesses memory, as the output is determined by the historical inputs.

Derivation from the formulas:

$$u[n] = \sum_{k=-\infty}^{n} \delta[\mathrm{k}]$$

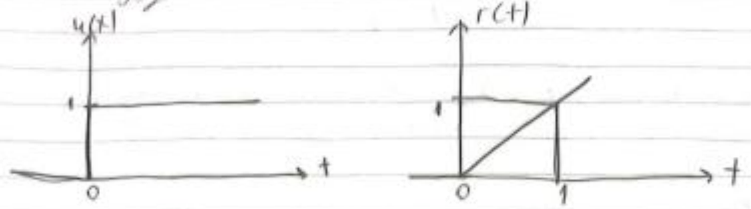
$$r[n+1] = \sum_{k=-\infty}^{n} \mathrm{u[k]}$$

$$y[n] = \sum_{k=-0}^{n} \mathrm{x[n-k]}(k+1)$$

Appendix:

Part 1.1

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



$s(t) = \int_{-\infty}^{t} u(\tau)\,d\tau = r(t) \rightarrow$ unit ramp function

Linearity

Assume $x(t) = \alpha y(t) + \beta z(t)$ and $Y(t) = \int_{-\infty}^{t} y(\tau)\,d\tau$, $Z(t) = \int_{-\infty}^{t} z(\tau)\,d\tau$

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

$= \alpha Y(t) + \beta Z(t)$ system is linear because integration operation is also liner.

Causality

Integration takes the area between the function and x-axis this system does this for $t = -\infty$ to current time, therefore system output depends on all past values and no future values, therefore system is causal.

Part 1.1

Part1.1 Question

Time Invariance

$$\int_{-\infty}^{t} x(\tau-k)\,d\tau = \int_{-\infty}^{t-k} x(u)\,du = X(t-k)-X(-\infty)$$

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

$\Rightarrow$ system is time invariant

BIBO stability

impulse response is $u(t)$ of this system for BIBO stability

$\int_{-\infty}^{\infty} |u(t)|\,dt < \infty$ should be satisfied

$= \int_{-\infty}^{0} 0\,dt + \int_{0}^{\infty} 1\,dt = t\Big|_{0}^{\infty} = \infty$. System is not BIBO stable which is the problem.

Memory

Since the integrals accumulates the effect of past inputs on the present input, system has memory.

Part 1.1

Part 1.2

BIBO stability

$$\int_{-\infty}^{\infty} |e^{-at}u(t)|\, dt = \int_{0}^{\infty} e^{-at} dt = -\frac{1}{a} e^{-at} \Big|_{0}^{\infty} = -\frac{1}{a}$$

for nonzero a values system is BIBO stable this system has no problem at BIBO unlike the first system in part 1.1.

impulse response is $u(t)$ of this system for BIBO stability

$\int_{-\infty}^{\infty} |u(t)| dt < \infty$ should be satisfied

$= \int_{-\infty}^{0} 0\, dt + \int_{0}^{\infty} 1\, dt = t \Big|_{0}^{\infty} = \infty$. $\rightarrow$ system is not BIBO stable

BIBO of the system is problematic.

Memory

since $\int_{-\infty}^{t} e^{-at} d\tau$ integral

accumulates the effect of the past inputs on the present output this indicates system has a memory.

Causality:

System output value again depends on only current and past values of the input therefore, system is causal.

Part 1.2

Part 1.2 dream

Time Invariance

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

$$\int_{-\infty}^{\infty} x(u) e^{-a(t+u-k)} u(t-u-k)\, du = \int_{-\infty}^{t-k} x(u) e^{-a(t-u-k)}\, du$$

$$\int_{-\infty}^{t} x(\tau) e^{-a(t-\tau)}\, d\tau \xrightarrow{t=t-k} \int_{-\infty}^{t-k} x(\tau) e^{-a(t-\tau-k)}\, d\tau$$

system is time invariant.

$$s(t) = \int_{-\infty}^{t} h(\tau)\, d\tau = \int_{-\infty}^{t} e^{-a\tau} u(\tau)\, d\tau = \int_{0}^{t} e^{-a\tau}\, d\tau = -\frac{1}{a} e^{-a\tau}\Big|_{0}^{t}$$

$$= \frac{1}{a} - \frac{1}{a} e^{-at}.$$

Linearity

$$x(t) = \alpha y(t) + \beta z(t), \quad X(t) = x(t) * h(t) = \int_{-\infty} x(\tau) e^{-a(t-\tau)} u(t-\tau)\, d\tau$$

$$= \int_{-\infty}^{t} x(\tau) e^{-a(t-\tau)}\, d\tau = \int_{-\infty}^{t} (\alpha y(\tau) + \beta z(\tau)) e^{-a(t-\tau)}\, d\tau$$

$$= \alpha \underbrace{\int_{-\infty}^{t} y(\tau) e^{-a(t-\tau)}\, d\tau}_{Y(t)} + \beta \underbrace{\int_{-\infty}^{t} z(\tau) e^{-a(t-\tau)}\, d\tau}_{Z(t)}$$

$$X(t) = \alpha Y(t) + \beta Z(t).$$

Part 1.2

Whole code for the assignment part by part:

```
Ts = 0.01;
a = 0.5;
rexp = @(t) exp(-a*t) * (t >= 0);
u = @(t) (t >= 0);
impulse = @(t) (abs(t) <= 0.01 / 2) * (1 / 0.01);
t = -10: 1: 10;
start = -20;

unit_response = DiscreteIntegrator(u,t,Ts,start);
impulse_response = DiscreteIntegrator(impulse,t,Ts,start);
s = DiscreteIntegrator(rexp,t,Ts,start);
h = exp(-a*t) .* (t >= 0);

subplot(2,2,1);
stem(t,unit_response);
title("Unit Step Response");
xlabel("t");
ylabel("Amplitude");
ylim([-4.0,14.0]);

subplot(2,2,2);
stem(t,impulse_response);
title("Impulse Response");
xlabel("t");
ylabel("Amplitude");
ylim([-2.0,3.0]);

subplot(2,2,3);
stem(t,h);
title("Impulse reponse of second system: h(t)");
xlabel("t");
ylabel("Amplitude");
ylim([-2.0,3.0]);

subplot(2,2,4);
stem(t,s);
title("Unit step reponse of second system: s(t)");
xlabel("t");
ylabel("Amplitude");
ylim([-2.0,4]);
```

"lab3part1_3.m"

```
function result = DiscreteIntegrator(func, arr, stepSize, start)
sum = 0;
result = zeros(1,length(arr));

    for n = 1:1:length(arr)
    for currentStep = start : stepSize : arr(n)
        sum = sum + func(currentStep) * stepSize;
    end
    result(n) = sum;
    sum = 0;
    end
```

"DiscreteIntegrator.m"

```matlab
n_range = 100:300:10000;
t = -10000: 1: 10000;

ha05 =  exp(-0.5*t) .* (t >= 0);

ha025 = exp(-0.25*t) .* (t >= 0);

ha01 =  exp(-0.1*t) .* (t >= 0);

ha005 = exp(-0.05*t) .* (t >= 0);

ha0 =  exp(-0.0*t) .* (t >= 0);

a_values = [0.5,0.25,0.1,0.05,0.0];
h_values = {ha05,ha025,ha01,ha005,ha0};

for i = 1:1:length(a_values)
    sum_array = sumElements(h_values{i},n_range);
    subplot(3,2,i);
    plot(n_range,sum_array);
    title("a = ",a_values(i));
    xlabel("N");
    ylabel("Sum")
end

function sum_array = sumElements(h, N_range)
    midIndex = 10001;
    sum_array = arrayfun(@(N) sum(abs(h(midIndex - N : midIndex + N))), N_range);
end
```

“lab3part2.m

```
t = 1: 1: 20;

ha05 =  exp(-0.5*t) .* (t >= 0);

ha025 = exp(-0.25*t) .* (t >= 0);

ha01 =  exp(-0.1*t) .* (t >= 0);

ha005 = exp(-0.05*t) .* (t >= 0);

ha0 =  exp(-0.0*t) .* (t >= 0);

x1_seq = (8*((t>=0)-((t-4)>=0))-4*(((t-4)>=0)-((t-13)>=0)));
x2_seq = 0.3.^t .* (t>=0);

y1ideal = conv(ha0,x1_seq);
y2ideal = conv(ha0,x2_seq);

y1005 = conv(ha005,x1_seq);
y2005 = conv(ha005,x2_seq);

y105 = conv(ha05,x1_seq);
y205 = conv(ha05,x2_seq);

y1025 = conv(ha025,x1_seq);
y2025 = conv(ha025,x2_seq);

y101 = conv(ha01,x1_seq);
y201 = conv(ha01,x2_seq);

subplot(5,2,1);
stem(y1ideal);
title("Y1 ideal");
xlabel("n")
ylabel("Amplitude")
subplot(5,2,2);
stem(y2ideal);
title("Y2 ideal");
xlabel("n")
ylabel("Amplitude")


subplot(5,2,3);
stem(y105);
title("Y1, a = 0.5");
xlabel("n")
ylabel("Amplitude")
subplot(5,2,4);
stem(y205);
title("Y2, a = 0.5");
xlabel("n")
ylabel("Amplitude")


subplot(5,2,5);
stem(y1025);
title("Y1, a = 0.25");
```

```
xlabel("n")
ylabel("Amplitude")
subplot(5,2,6);
stem(y2025);
title("Y2, a = 0.25");
xlabel("n")
ylabel("Amplitude")


subplot(5,2,7);
stem(y101);
title("Y1, a = 0.1");
xlabel("n")
ylabel("Amplitude")
subplot(5,2,8);
stem(y201);
title("Y2, a = 0.1");
xlabel("n")
ylabel("Amplitude")


subplot(5,2,9);
stem(y1005);
title("Y1, a = 0.05");
xlabel("n")
ylabel("Amplitude")
subplot(5,2,10);
stem(y2005);
title("Y2, a = 0.05");
xlabel("n")
ylabel("Amplitude")

aLabels = ["a = 0.5", "a = 0.25", "a = 0.1", "a = 0.05"];
errorsY1 = {abs(y1ideal-y105),abs(y1ideal-y1025),abs(y1ideal-y101),abs(y1ideal-
y1005)};
errorsY2 = {abs(y2ideal-y205),abs(y2ideal-y2025),abs(y2ideal-y201),abs(y2ideal-
y2005)};

figure;
for i = 1:length(errorsY1)
    subplot(2, 2, i);
    stem(errorsY1{i});
    title("Y1 " + aLabels{i});
    xlabel("n");
    ylabel("Amplitude");
end


figure;
for i = 1:length(errorsY2)
    subplot(2, 2, i);
    stem(errorsY2{i});
    title("Y2 " + aLabels{i});
    xlabel("n");
    ylabel("Amplitude");
end
```

“lab3part3.m

```
t = 1: 1: 20;
impulse = [0,0,0,0,0,1,0,0,0,0,0];
n_range = 0:5;

x1_seq = (8*((t>=0)-((t-4)>=0))-4*(((t-4)>=0)-((t-13)>=0)));
x2_seq = 0.3.^t .* (t>=0);

impulse_second_order_diff = second_order_diff(impulse,0,0);
y1 = conv(x1_seq,impulse_second_order_diff);
y2 = conv(x2_seq,impulse_second_order_diff);

stem(y1);
title("Y1")
xlabel("n")
ylabel("Amplitude")

figure;
stem(y2);
title("Y2")
xlabel("n")
ylabel("Amplitude")

figure;
stem(impulse_second_order_diff);
title("Impulse response")
xlabel("n")
ylabel("Amplitude")

figure;
sum_elements = sumElements(impulse_second_order_diff,n_range);
stem(sum_elements);
title("Sum elements")
xlabel("n")
ylabel("Amplitude")



function output = second_order_diff(sequence, seq_0, seq_n1)
    output = zeros(1, length(sequence));
    output(1) = sequence(1) - 2 * seq_0 + seq_n1;
    output(2) = sequence(2) - 2 * sequence(1) + seq_0;
    output(3:end) = sequence(3:end) - 2 * sequence(2:end-1) + sequence(1:end-2);
end


function sum_array = sumElements(h, N_range)
    sum_array = zeros(1, length(N_range));
    for n = 1:length(N_range)
        rangeIndices = (6 - N_range(n)):(6 + N_range(n));
        sum_array(n) = sum(abs(h(rangeIndices)));
    end
end
```

“lab3part4_1.m”

```
t = 1: 1: 20;

impulse = [0,0,0,0,0,1,0,0,0,0,0];
x1_seq = (8*((t>=0)-((t-4)>=0))-4*(((t-4)>=0)-((t-13)>=0)));
x2_seq = 0.3.^t .* (t>=0);

impulse_second_order_diff = second_order_diff(impulse,0,0);
impulse_inv_second_order_diff = inv_second_order_diff(impulse);

a = conv(impulse_inv_second_order_diff,impulse_second_order_diff);

y1 = conv(x1_seq,impulse_inv_second_order_diff);
y2 = conv(x2_seq,impulse_inv_second_order_diff);

stem(a(length(impulse) * 0.5 + 0.5:length(impulse)*1.5 - 0.5))
title("Convolution of impulse responses")
xlabel("n")
ylabel("Amplitude")

figure;
subplot(2,1,1);
stem(y1);
title("Y1")
xlabel("n")
ylabel("Amplitude")
subplot(2,1,2);
stem(y2);
title("Y2")
xlabel("n")
ylabel("Amplitude")


function output = second_order_diff(sequence, seq_0, seq_n1)
    output = zeros(1, length(sequence));
    output(1) = sequence(1) - 2 * seq_0 + seq_n1;
    output(2) = sequence(2) - 2 * sequence(1) + seq_0;
    output(3:end) = sequence(3:end) - 2 * sequence(2:end-1) + sequence(1:end-2);
end

function output = inv_second_order_diff(sequence)
    firstPass = cumsum(sequence);
    output = cumsum(firstPass);
end
```

“lab3part4_2.m”