National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering and Computer Science

Faculty Member: Dr. Salman Ghafoor Dated: 10/10/2022

Semester: 5th Section: BEE 12C

EE-232: Signals and Systems

Lab 5: Introduction to Properties of Systems

Group Members

		PLO4 – CLO3	PLO5 - CLO3	PLO8 - CLO4	PLO9 - CLO4
Name	Reg. No	**Viva / Quiz / Lab Performance**	**Analysis of data in Lab Report**	**Modern Tool Usage**	**Ethics and Safety**
		5 Marks	**5 Marks**	**5 Marks**	**5 Marks**
Danial Ahmad	331388				
Muhammad Umer	345834				
Syeda Fatima Zahra	334379				

1 Table of Contents

2 Introduction to Properties of Systems

2.1 Objectives

The goal of this exercise is to gain familiarity with properties of systems. It is important to understand how to demonstrate when a system does or does not satisfy a given property. MATLAB can be used to create counter examples demonstrating that certain properties are not satisfied.

- How to determine if systems satisfy a particular property or not
- Properties of Linear Time Invariant Systems

2.2 Equipment

Software

- *MATLAB*

2.3 Lab Instructions

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

- Lab objectives
- MATLAB codes
- Results (Graphs/Tables) duly commented and discussed
- Conclusion

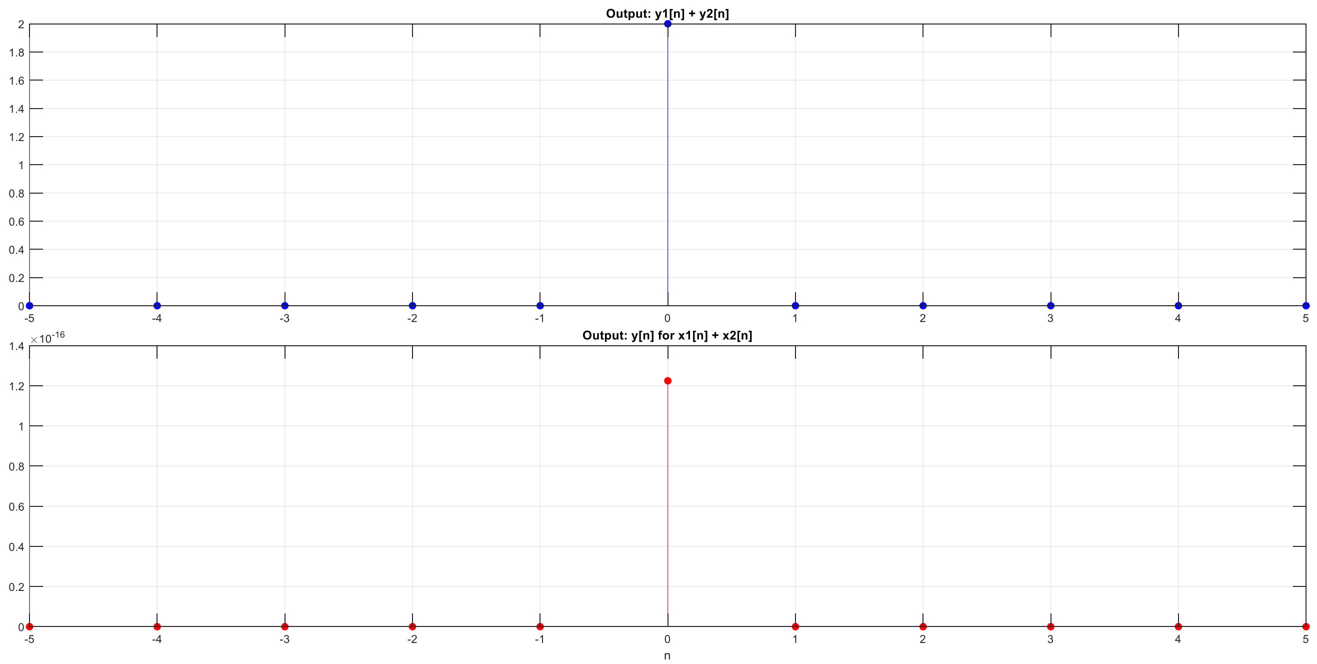
3 Lab Tasks

3.1 Task 1

1. Verify that the system $y[n] = \sin((\pi/2)\, x[n])$ is not linear. Use signal $x_1[n] = \delta[n]$ and $x_2[n] = 2\delta[n]$ to demonstrate if the system violates linearity.

```
%% Task 1
n = -5:5;
% Definitions
x1 = n == 0;
x2 = (2 * n) == 0;
y1 = sin((pi / 2) .* x1);
y2 = sin((pi / 2) .* x2);
y_add = y1 + y2; % Sum of y
x = x1 + x2;
y = sin((pi / 2) .* x); % Additive y

% Plots
subplot(211)
stem(n, y_add, 'filled', 'b')
title('Output: y1[n] + y2[n]')
grid
subplot(212)
stem(n, y, 'filled', 'r')
title('Output: y[n] for x1[n] + x2[n]')
grid
xlabel('n')
```

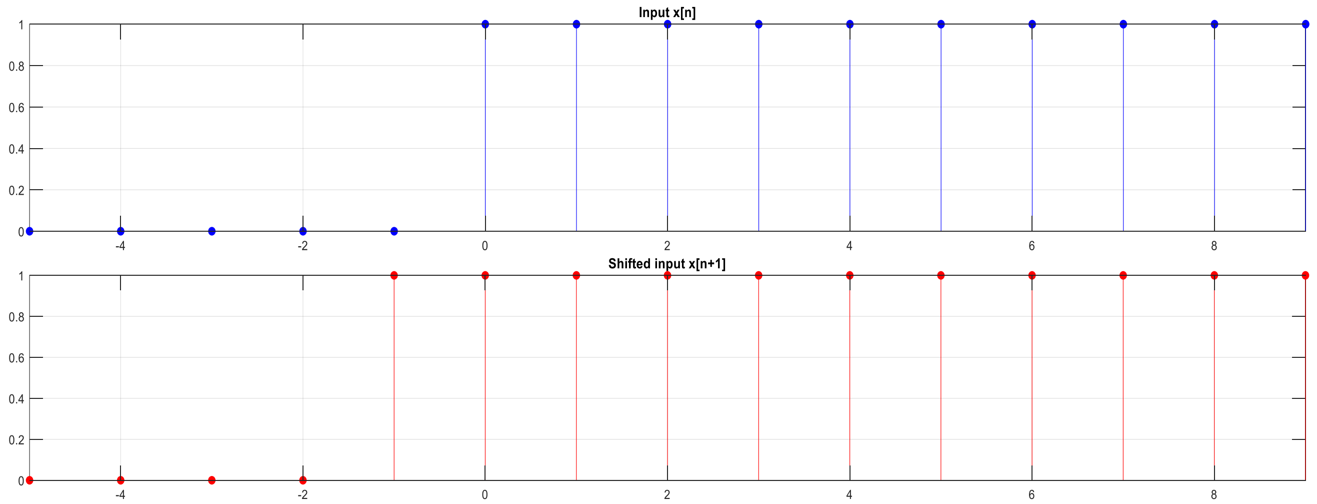


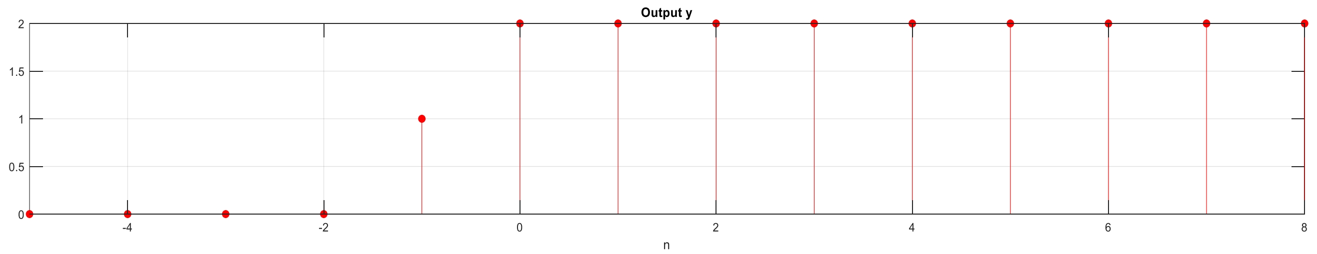
Comments: From the above plots for output *"y1[n] + y2[n]"* and output *"y[n] for input x1[n] + x2[n]"*, we can verify that the system y[n] = sin ((π/2) x[n]) is not linear as for linearity, both the outputs should be the same.

2. Verify if the following system y[n] = x[n] + x[n+1] is not causal. Use the signal x[n]=u[n] to demonstrate this. Define vectors x and y to represent the input on the interval $-5 \leq n \leq 9$ and output on the interval $-5 \leq n \leq 8$.

```
%% Task 1.2
n = -5:9;
% Definitions
x = n >= 0;
x_shifted = (n + 1) >= 0;
y = x + x_shifted;

% Plots
subplot(311)
stem(n, x, 'filled', 'b')
title('Input x[n]')
grid
axis([-5 9 0 1])
subplot(312)
stem(n, x_shifted, 'filled', 'r')
title('Shifted input x[n+1]')
axis([-5 9 0 1])
grid
subplot(313)
stem(n, y, 'filled', 'r')
title('Output y')
axis([-5 8 0 2])
grid
xlabel('n')
```



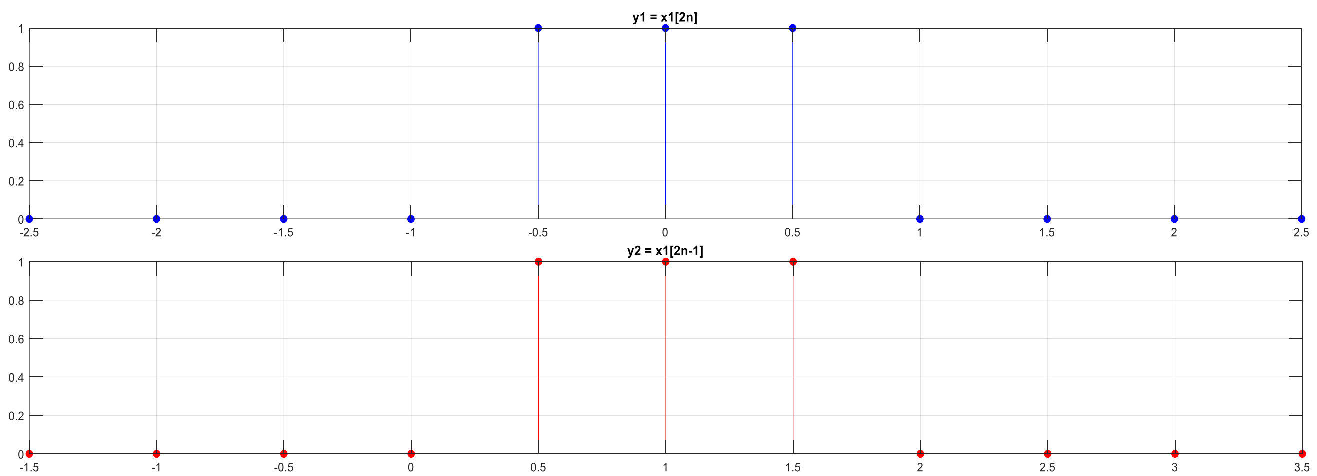


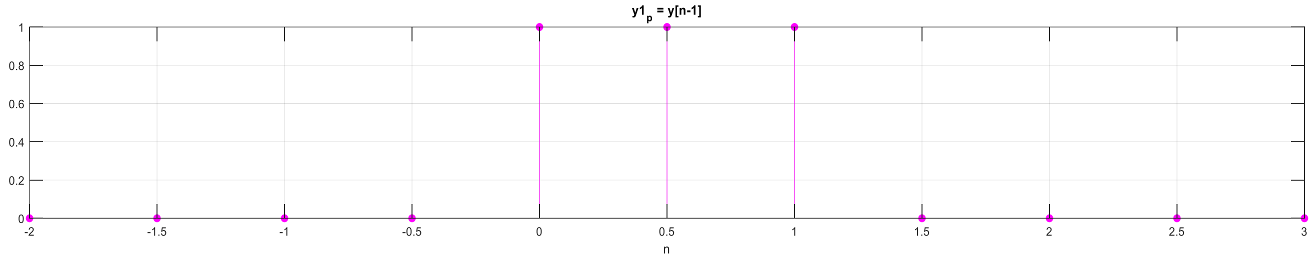
Answer: The given system is not causal as the output depends on input values present in the future.

3. Verify if the following system y[n] = x[2n] is time variant or invariant? Use a signal of your choice.

```
n_def = -5:5;
% Definitions
x = (n >= -1) - (n > 1);
n1 = 0.5 * n_def;
n2 = (0.5 * n_def) + 1;
n1_p = 0.5 * (n_def + 1);
y = x;

% Plots
subplot(311)
stem(n1, y, 'filled', 'b')
title('y1 = x1[2n]')
grid
subplot(312)
stem(n2, y, 'filled', 'r')
title('y2 = x1[2n+1]')
grid
subplot(313)
stem(n1_p, y, 'filled', 'r')
title('y1_p = y[n+1]')
grid
xlabel('n')
```





Answer: The given system is not time invariant as the output *"y[n-1]"* does not yield the same plot as that of *"y = x[2n-1]"*.

3.2 Task 2

1. Given the signals x[n] = [1 2 3 4 5] and y[n] = [1 1 1 1 1], verify using 'conv or convn' function that commutative property holds.

```
x_n = [1 2 3 4 5];
y_n = [1 1 1 1 1];

lhs = conv(x_n, y_n)
rhs = conv(y_n, x_n)

if lhs == rhs
    disp('Commutative property holds.')
end
```

Output:

```
lhs =
     1     3     6    10    15    14    12     9     5
rhs =
     1     3     6    10    15    14    12     9     5
Commutative property holds.
```

2. Assume a 2-D signal (i.e., some image). Load image and assume it to be signal x. Next assume that instead of having a 2-D filter you have two one D filters h1[n] = [0.25 0.5 0.25] and h2[n] = [0.25; 0.5; 0.25].

```
x = imread('image.jpg');
h1_n = [0.25 0.5 0.25];
h2_n = [0.25; 0.5; 0.25];

% x[n] * (h1[n] * h2[n]) = (x[n] * h1[n]) * h2[n]
h3_n = convn(h1_n, h2_n);
lhs = convn(x, h3_n);
size_lhs = size(lhs)
h4_n = convn(x, h1_n);
rhs = convn(h4_n, h2_n);
```

```
size_rhs = size(rhs)

if lhs == rhs
    disp('Associative property holds.')
```

```
size_lhs =
   185   278     3
size_rhs =
   185   278     3
Associative property holds.
```

4 Conclusion

In this lab, we familiarized ourselves with systems and verified the properties of said systems using discrete time signals. We also introduced ourselves to convolution in discrete time and verified intrinsic properties of convolution through the use of equality operator.