**Digital Signal Processing Lab**

**EEL-325**

Lab Journal: 03



* + - 1. Name: M.OMAR
      2. Class: BCE-6
      3. Enrollment No: 01-132182-024

**Experiment No. 3**

**Implement Discrete time signal (LTI System) Using MATLAB**

**Objectives:**

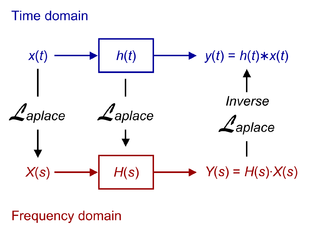
1. We will be able to know about discrete time linear time invariant (LTI) system using MATLAB.
2. Linear or nonlinear system behavior checked using MATLAB plotting.

## Introduction:

The defining properties of any LTI system are *linearity* and *time invariance*.

* *Linearity* means that the relationship between the input and the output are the result of [linear differential equations](https://en.wikipedia.org/wiki/Linear_differential_equation), that is, differential equations employing only [linear operators](https://en.wikipedia.org/wiki/Linear_map#Definition_and_first_consequences). A linear system that maps an input **x(t)** to an output **y(t)** will map a *scaled* input **ax(t)** to an output **ay(t)** likewise scaled by the same factor **a**. And the [superposition principle](https://en.wikipedia.org/wiki/Superposition_principle) applies to a linear system: if the system maps inputs **x1(t)** and **x2(t)** to outputs **y1(t)** and **y2(t)** respectively, then it will map **x3(t) = x1(t) + x2(t)** to the output **y3(t)** where **y3(t) = y1(t) + y2(t)**.
* *Time invariance* means that whether we apply an input to the system now or *T* seconds from now, the output will be identical except for a time delay of *T* seconds. That is, if the output due to input {\displaystyle x(t)} is {\displaystyle y(t)}, then the output due to input {\displaystyle x(t-T)} is {\displaystyle y(t-T)}. Hence, the system is time invariant because the output does not depend on the particular time the input is applied.

The fundamental result in LTI system theory is that any LTI system can be characterized entirely by a single function called the system's [impulse response](https://en.wikipedia.org/wiki/Impulse_response). The output of the system **y(t)** is simply the [convolution](https://en.wikipedia.org/wiki/Convolution) of the input to the system **x(t)** with the system's impulse response **h(t)**. This is called a [continuous time](https://en.wikipedia.org/wiki/Continuous_time) system. Similarly, a discrete-time linear time-invariant (or, more generally, "shift-invariant") system is defined as one operating in [discrete time](https://en.wikipedia.org/wiki/Discrete_time): **yi = xi \* hi** where y, x, and h are [sequences](https://en.wikipedia.org/wiki/Sequences) and the convolution, in discrete time, uses a discrete summation rather than an integral.

[](https://en.wikipedia.org/wiki/File:LTI.png)

Relationship between the **time domain** and the **frequency domain**

LTI systems can also be characterized in the [*frequency domain*](https://en.wikipedia.org/wiki/Frequency_domain) by the system's [transfer function](https://en.wikipedia.org/wiki/Transfer_function), which is the [Laplace transform](https://en.wikipedia.org/wiki/Laplace_transform) of the system's impulse response (or [Z transform](https://en.wikipedia.org/wiki/Z_transform) in the case of discrete-time systems). As a result of the properties of these transforms, the output of the system in the frequency domain is the product of the transfer function and the transform of the input. In other words, convolution in the time domain is equivalent to multiplication in the frequency domain.

For all LTI systems, the [eigenfunctions](https://en.wikipedia.org/wiki/Eigenfunction" \o "Eigenfunction), and the basis functions of the transforms, are [complex](https://en.wikipedia.org/wiki/Complex_number) [exponentials](https://en.wikipedia.org/wiki/Exponential_function). This is, if the input to a system is the complex waveform {\displaystyle A\_{s}e^{st}} for some complex amplitude {\displaystyle A\_{s}} and complex frequency {\displaystyle s}, the output will be some complex constant times the input, say {\displaystyle B\_{s}e^{st}} for some new complex amplitude {\displaystyle B\_{s}}. The ratio {\displaystyle B\_{s}/A\_{s}} is the transfer function at frequency {\displaystyle s}.

Since [sinusoids](https://en.wikipedia.org/wiki/Sine_wave) are a sum of complex exponentials with complex-conjugate frequencies, if the input to the system is a sinusoid, then the output of the system will also be a sinusoid, perhaps with a different [amplitude](https://en.wikipedia.org/wiki/Amplitude) and a different [phase](https://en.wikipedia.org/wiki/Phase_(waves)), but always with the same frequency upon reaching steady-state. LTI systems cannot produce frequency components that are not in the input.

LTI system theory is good at describing many important systems. Most LTI systems are considered "easy" to analyze, at least compared to the time-varying and/or [nonlinear](https://en.wikipedia.org/wiki/Nonlinear) case. Any system that can be modeled as a linear [differential equation](https://en.wikipedia.org/wiki/Differential_equation) with constant coefficients is an LTI system. Examples of such systems are [electrical circuits](https://en.wikipedia.org/wiki/Electrical_network) made up of [resistors](https://en.wikipedia.org/wiki/Resistor), [inductors](https://en.wikipedia.org/wiki/Inductor), and [capacitors](https://en.wikipedia.org/wiki/Capacitor) (RLC circuits). Ideal spring–mass–damper systems are also LTI systems, and are mathematically equivalent to RLC circuits.

Most LTI system concepts are similar between the continuous-time and discrete-time (linear shift-invariant) cases. In image processing, the time variable is replaced with two space variables, and the notion of time invariance is replaced by two-dimensional shift invariance. When analyzing [filter banks](https://en.wikipedia.org/wiki/Filter_bank) and [MIMO](https://en.wikipedia.org/wiki/MIMO_(systems_theory)) systems, it is often useful to consider [vectors](https://en.wikipedia.org/wiki/Matrix_(mathematics)) of signals.

A linear system that is not time-invariant can be solved using other approaches such as the [Green function](https://en.wikipedia.org/wiki/Green%27s_function) method. The same method must be used when the initial conditions of the problem are not null.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**Procedure:**

* Open MATLAB Software.
* We create the new project.
* Make a new script and name it on the name of your lab.
* In order to get voice signal in MATLAB use built-in function **“audiorecorder”** for recording Voice signal.
* To read and write the voice signal use **“audiowrite”.**
* To get voice signal as a vector, we use **“getaudiodata”** function.- Record the voice signal for sometimes. I.e. t=10sec- Inter sound and plot command.
* Run the mail file and observe the results.

## Problem No. 01:

**Task:** Whether the system is linear or not.

**Y[n] = x[n] + 4x [n-1]**

Codes and Results: LINEAR FUNCTION

n=0:30;

x1\_n=cos(2\*pi\*0.1\*n);

x2\_n=cos(2\*pi\*0.4\*n);

num=[1 4 ];

den=[1];

y1=filter(num,den,x1\_n);

y2=filter(num,den,x2\_n);

a=2;

b=-3;

x3=a\*x1\_n+b\*x2\_n;

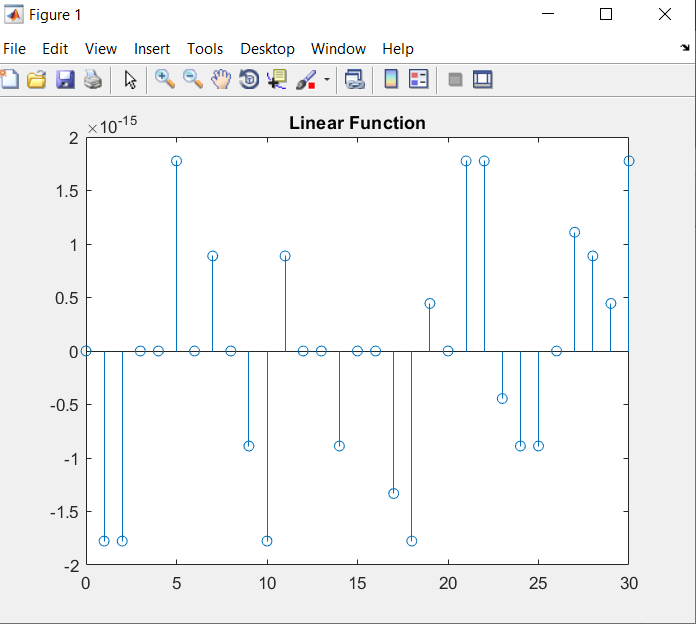
y3=filter(num,den,x3);

z=a\*y1+b\*y2;

p=y3-z;

stem(n,p)

title('Linear Function')



**Task2:** Whether the system is linear or not.

**Y[n] – 2Y [n-1] +** **3Y [n-3]** **= 3x[n] + 4x [n-1]**

Codes and Results: NOT A LINEAR FUNCTION

n=0:70;

x1\_n=cos(2\*pi\*0.1\*n);

x2\_n=cos(2\*pi\*0.4\*n);

num=[3 4 ];

den=[1 -2 0 3];

y1=filter(num,den,x1\_n);

y2=filter(num,den,x2\_n);

a=2;

b=-3;

x3=a\*x1\_n+b\*x2\_n;

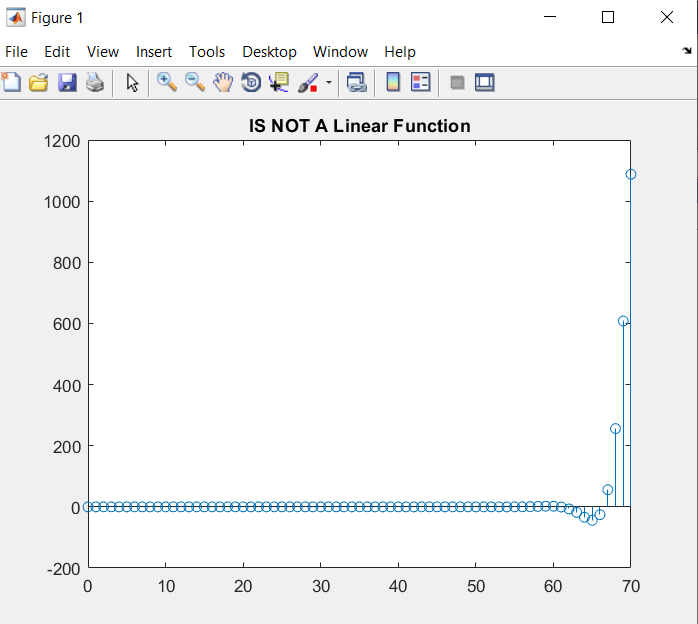
y3=filter(num,den,x3);

z=a\*y1+b\*y2;

p=y3-z;

stem(n,p)

title('IS NOT A Linear Function')



## Problem No. 02:

**Task1:** Whether the system is Time invariant or not.

1. **Y[n] = x[n] - x [n-1]**

% time invariant

n=0:30;

x1\_n=cos(2\*pi\*0.1\*n);

x2\_n=cos(2\*pi\*0.4\*n);

num=[1 -1 ];

den=[1];

x=x1\_n+x2\_n;

subplot(2,2,1);

stem(n,x);

title('Original input');

x\_delay= delayseq(x,10);

subplot(2,2,2);

stem(n,x\_delay);

title('delayed input');

y1=filter(num,den,x);

subplot(2,2,3);

stem(n,y1);

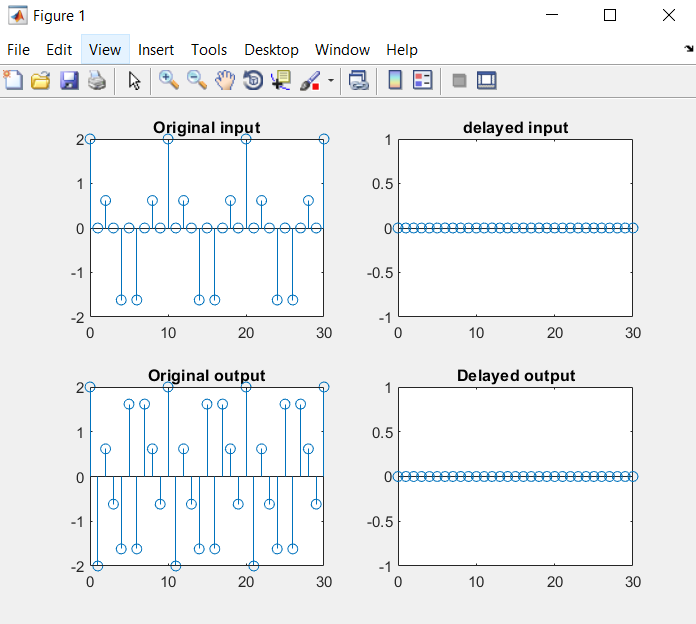
title('Original output');

y2=filter(num,den,x\_delay);

subplot(2,2,4);

stem(n,y2);

title('Delayed output');



1. **Y[n] = n** **x[n]**

Codes and Results:

%NOT A TIME INVARIANT FUNCTION.

n=0:30;

x1\_n=cos(2\*pi\*0.1\*n);

x2\_n=cos(2\*pi\*0.4\*n);

num=[n ];

den=[1];

x=x1\_n+x2\_n;

subplot(2,2,1);

stem(n,x);

title('Original input');

x\_delay= delayseq(x,10);

subplot(2,2,2);

stem(n,x\_delay);

title('delayed input');

y1=filter(num,den,x);

subplot(2,2,3);

stem(n,y1);

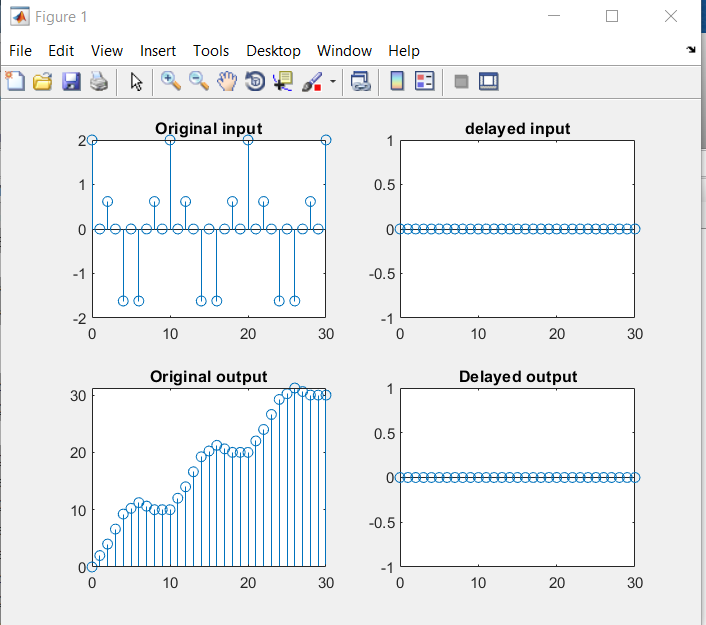
title('Original output');

y2=filter(num,den,x\_delay);

subplot(2,2,4);

stem(n,y2);

title('Delayed output');



**Task2:** Whether the system is Time invariant or not.

**Y[n] = 3x[n] + 5x [n-1]** + **10x [n-4]**

Codes and Results:

% time invariant

n=0:70;

x1\_n=cos(2\*pi\*0.1\*n);

x2\_n=cos(2\*pi\*0.4\*n);

num=[3 5 0 0 10 ];

den=[1];

x=x1\_n+x2\_n;

subplot(2,2,1);

stem(n,x);

title('Original input');

x\_delay= delayseq(x,10);

subplot(2,2,2);

stem(n,x\_delay);

title('delayed input');

y1=filter(num,den,x);

subplot(2,2,3);

stem(n,y1);

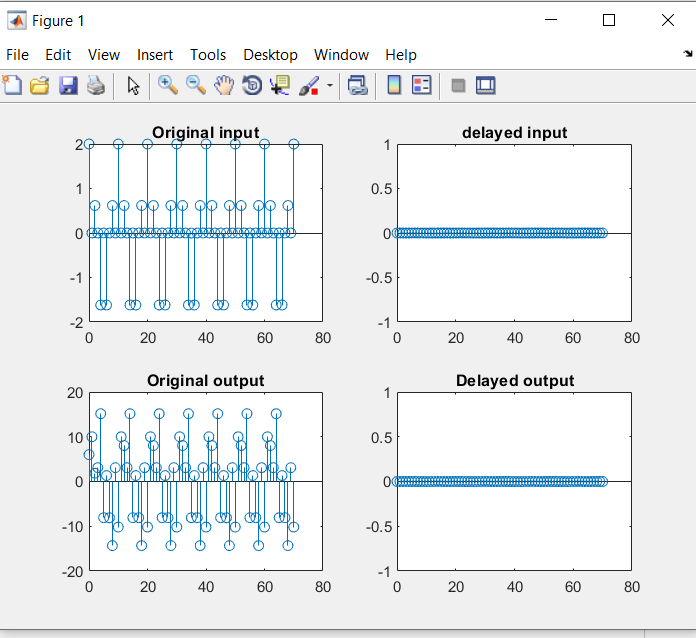
title('Original output');

y2=filter(num,den,x\_delay);

subplot(2,2,4);

stem(n,y2);

title('Delayed output');



**Conclusion: Today we learned about Linear Time Invariant functions and also proved that a function is LTI or not**