COMSATS University Islamabad, Lahore Campus
Department of Computer Engineering

EEE223 Signals and Systems

Lab Manual for Spring 2025 & Onwards

Lab Resource Persons

Engr.Abu Bakar Talha
Engr.Tabassum Nawaz Bajwa

Theory Resource Persons

Dr. Arsla Khan
Engr.Tabassum Nawaz Bajwa

Name:____________________________ **Registration Number:** ____________________________

Program:__________________________ **Batch:** ____

Semester ______________________________

Revision History

S.No.	Update	Date	Performed By
1	Lab Manual Preparation	Sep-12	Mr. Muhammad Ijaz
2	Lab Manual Review	Sep-12	Dr. Ejaz Ahmad Ansari
3	Lab Manual Modifications	March-15	Ms. Tabassum Nawaz Bajwa Mr. Assad Ali
4	Lab Manual Review	March-15	Dr. Ejaz Ahmad Ansari
5	Lab Manual Modification	Sep -15	Muhammad Usman Iqbal
6	Lab Manual Review	Sep-15	Dr. Ejaz Ahmad Ansari
7	Lab Manual Modification	Aug-16	Mr. Assad Ali Ms. Maida Farooq
8	Lab Manual Review	Aug-16	Dr. Khurram Ali
9	Lab Manual Modification	Sep-17	Ms. Maida Farooq
10	Lab Manual Review	Sep-17	Dr. Khurram Ali
11	Lab Manual Modification	Feb-2020	Ms. Sidra Saleem
12	Lab Manual Modification	Sep-2021	Mr. Abu Bakar Talha
13	Lab Manual Modification	Feb-2022	Mr. Abu Bakar Talha
14	Lab Manual Modification	Feb-2023	Mr. Abu Bakar Talha
15	Lab Manual Modification	June-2024	Mr. Abu Bakar Talha

Preface

This course allows the students to analyze in depth the signals and systems in time, frequency and z- domains respectively. Thorough understanding of signals and systems in time domain further helps the students to study their applications in frequency domain quite comprehensively. Based on our experience, working in frequency domain is always a fun and more often simplifies many engineering problems and disseminates a more compact overview of the problem. The rationale behind this course is to make the student able to think in various dimensions especially in frequency domain. It is expected that student should acquire familiarity with mathematical representation of signals and systems.

After the completion of this course, following key educational aims are expected to be achieved and the student should be able to:

- Understand the representation of continuous and discrete time signals and systems in time domain.
- Analyze continuous and discrete – time signals and systems in time domain.
- Study and analyze frequency domain versions of different systems along with their Characteristics.
- Determine Fourier series/and Fourier Transforms representation of Periodic and A-periodic Signals.
- Grasp concepts of Fourier Transform and Discrete time Fourier Transform, and understand their properties.
- Grasp concepts of Laplace transform, analysis of properties and characterization of LTI systems.
- Grasp concepts of z transform, analysis of properties and characterization of LTI.
- Acquire enough understanding to build a comprehensive foundation for later higher level courses such as communication systems, control systems and digital signal processing.
- Implement these concepts on MATLAB.
- Utilize mathematical skills for analyzing and solving complex systems.
- Use his/her software skills of MATLAB in processing signals through systems for various engineering applications.
- Apply the knowledge of this course towards learning of higher level modules such as Analog and Digital Communication Systems, Modern Control Systems and Digital Signal Processing.

This is a basic course that is intended to provide the fundamentals of signals, systems and transforms to the electrical engineering students. The student should acquire familiarity with mathematical representation of signals and systems. S/he should be able to think in frequency domain. The course is aimed to build a comprehensive foundation for later higher level courses in communication system, control systems and digital signal processing. Both discrete-time and continuous-time signals, systems and transforms are covered in the course.

Books

Text Books

1. Continuous and Discrete Time Signals and System by Mrinal Mandal and Amir Asif, Latest Edition, Cambridge.

Reference Books

1. Linear Systems by and Signals by B.P Lathi, Latest Edition, OxfordPress.
2. Signals and Systems by Simon Haykin & Bary Van Veen, Latest Edition, John Wiley & Sons.
3. Signals and Systems by Alan V Oppenheim & Willisky, Latest Edition, Prentice Hall.

Reference Books for Manual

1. Signals and Systems Laboratory with MATLAB by Alex Palamides and Anastasia Veloni.

Learning Outcomes

1. Understand the representation of continuous and discrete time signals and systems in time domain.
2. Analyze continuous and discrete – time signals and systems in time domain.
3. Study and analyze frequency domain versions of different systems along with their Characteristics.
4. Determine Fourier series/and Fourier Transforms representation of Periodic and A-periodic Signals.
5. Grasp concepts of Fourier Transform and Discrete time Fourier Transform, and understand their properties.
6. Grasp concepts of Laplace transform, analysis of properties and characterization of LTI systems.
7. Grasp concepts of z transform, analysis of properties and characterization of LTI.
8. Implement the course concepts on MATLAB.

Software Resources

✓ MATLAB Latest Version.

List of Equipment

✓ Personal Computers

Grading Policy

The final marks for lab would comprise of Lab Assignments (25%), Mid-Term (25%) and Lab Terminal (50%).

Mid-Term Lab Mid Term = 0.5*(Lab Mid Term) + 0.5*(average of lab evaluation of Lab 1-6)
Terminal 0.5*(Lab Terminal Exam) +0.375*(average of lab evaluation of Lab 7-12) + 0.125*(average of lab evaluation of Lab 1-6)

Lab Assignment Marks:

- Lab Assignment 1 marks = Lab evaluation marks from experiment 1-3.
- Lab Assignment 2 marks = Lab evaluation marks from experiment 4-6.
- Lab Assignment 3 marks = Lab evaluation marks from experiment 7-9.
- Lab Assignment 4 marks = Lab evaluation marks from experiment 10-12.

The minimum pass marks for both lab and theory shall be 50%. Students obtaining less than 50% marks (in either theory or lab, or both) shall be deemed to have failed in the course. The final marks would be computed with 75% weight to theory and 25% to lab final marks.

Lab Instructions

- ✓ This lab activity comprises of three parts: Pre-lab, Lab Tasks, Critical Analysis and Conclusion and Viva session.
- ✓ The students should perform and demonstrate each lab task separately for step-wise evaluation.
- ✓ Only those tasks that are completed during the allocated lab time will be credited to the students. Students are however encouraged to practice on their own in spare time for enhancing their skills.

Safety Instructions

1. Conduct yourself in a responsible manner at all times in the laboratory. Don’t talk aloud or crack jokes in lab.
2. Know the location of electrical panels and disconnect switches in or near your laboratory so that power can be quickly shut down in the event of a fire or electrical accident
3. Report any damages to equipment, hazards, and potential hazards to the laboratory instructor.
4. If in doubt about electrical safety, see the laboratory instructor. Regarding specific equipment, consult the instruction manual provided by the manufacturer of the equipment. Information regarding safe use and possible- hazards should be studied carefully.
5. Make sure that the computers should be properly turned off when you complete your lab task.
6. Make sure that the software used in Lab is properly handled.
7. During any problem regarding software consult the Lab instructor.
8. Do not eat food, drink beverages or chew gum in the laboratory and do not use laboratory glassware as containers for food or beverages. Smoking is strictly prohibited in lab area.
9. Do not open any irrelevant internet sites on lab computer.
10. Do not use a flash drive on lab computers.
11. Do not upload, delete or alter any software on the lab PC.

List of Experiments:

Experiment No.	Lab Title
01	To reproduce the Continuous and Discrete Time Signals Using MATLAB
02	To reproduce the Periodic and Aperiodic Continuous and Discrete Time Signals using Elementary Signals in MATLAB
03	To follow the Properties of Different Signal Types Using Signal Operations in MATLAB
04	To follow the Properties of the System Using I/O Relationship in MATLAB
05	To reproduce the Response of Linear Time Invariant Systems by Performing Convolution Using MATLAB
06	To reproduce the Line Spectrum of Periodic Signals Using Properties of Fourier Series Coefficients in MATLAB
07	To Reproduce the Continuous Time Fourier Transform (CTFT) Using MATLAB Functions
08	To Trace the Response of Continuous Time Signals Using Laplace Transform in MATLAB
09	To Reproduce the Properties of Laplace Transform Using MATLAB Functions
10	To Trace the Response of Discrete Time Signals Using z-Transform in MATLAB
11	To Reproduce the properties of z-Transform in MATLAB
12	To display the basic Signals and Systems Operations Using Simulink

Experiment Relevant to Learning Outcome

Learning Outcome	Relevant Experiment
To measure and display the response of various signals and systems using MATLAB design and simulation tool. (PLO5-P3)	Experiment 1-12

CLO-PLO Mapping

PLOs / CLOs	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	PLO11	PLO12	Cognitive Level	Psychomotor Level	Affective Level
CLO1	X												C3		
CLO2		X											C3		
CLO3			X										C4		
CLO4					X									P3	

Lab CLOs – Lab Experiment Mapping

Labs / CLOs	Lab 1	Lab 2	Lab 3	Lab 4	Lab 5	Lab 6	Lab 7	Lab 8	Lab 9	Lab 10	Lab 11	Lab 12
CLO4	P3	P3	P3	P2	P3	P3	P3	P3	P3	P3	P3	P2

Table of Contents

LAB # 1: To Identify the Continuous and Discrete Time Signals Using MATLAB

Objectives

After completing this lab, the student will be able to:

- ✓ Describe the basic operations and commands in MATLAB.
- ✓ Describe the steps involved in plotting the Continuous and Discrete time signals in MATLAB.

Pre Lab

Part I –Introduction to MATLAB

What is MATLAB?

MATLAB is a high-level technical computing language equipped with a user-friendly interface. Its name stems from the words ***MATrix*** and ***LABoratory*** as it is based on the use of matrices. MATLAB is an extremely powerful tool useful for scientists and engineers from various disciplines. For example, MATLAB can be used in a wide range of applications, such as telecommunications, signal and image processing, control, mathematics, financial modeling, bioengineering, aeronautics, and many more.

M-Files

In order to write many commands that are executed all together, the program must be written in a text editor. In this editor, one can type all the needed commands to form a program, save the program, and execute it any time he or she wants. The text files are called M-files due to their suffix *.m.

There are two categories of M-files: the Scripts and the Functions.

Scripts

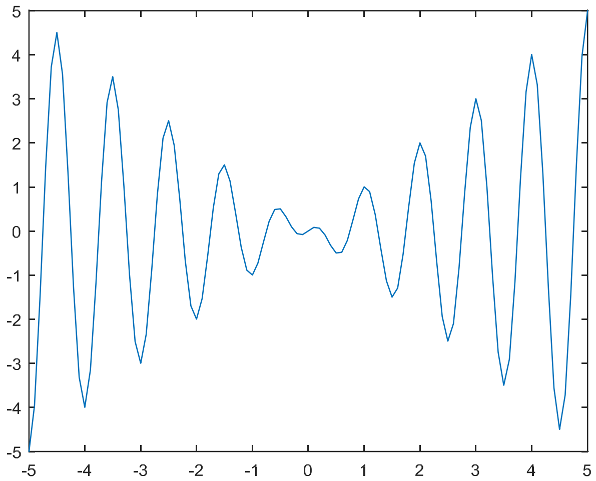
Scripts are the M-files with MATLAB commands. Their name must have a .m suffix. Scripts are suitable for solving the problems that require many commands. The advantage of the scripts is that they are implemented very easily.

Lab Tasks

Lab Task

Write a script file and execute.

```
% Program to  understand the use of script file
% f(t) = tcos(2πt), -5 ≤ t ≤ 5

t=-5:0.1:5;
f=t.*cos(2*pi*t);
plot(t,f)
```



Functions

Functions are also M-files, That is, are files with extension *.m* and must be saved in the *current Directory* of MATLAB. The difference between functions and scripts is that a function accepts one or more input arguments and returns one or more output arguments. To declare that an M-file is a function the first line of the m file must contain the syntax definition. More specifically, the first line of the M-file must be of the form *function [y1,y2,y3,…yn] = name(x1,x2,x3… xm).* The variable *y1,y2,…yn* are the outputs of the function while *x1,x2,…xm* are the input arguments. In case there is only one output, the square brackets are not necessary. The "*name*" specifies the name of the function. In order to execute a function, first the M-File is saved in *Current Directory*.

Lab Task

Write a function file and execute. Function should accepts as input two matrices and returns their sum and product.

```
% Program to  understand the use of a function file
% This function computes the sum and the product of 2 matrices

function [sm,pro]=oper(A,B)
sm=A+B;
pro=A*B;
end
```

```
% to test the function operation
% Enter the matric A
A= [ 2 3; 4 5]
% Enter Matrix B
B= [4 5; 5 6]
```

Useful Commands

Here we will learn and practice useful (when working with vectors and matrices) commands. As already discussed, the command sum returns the sum of the elements of a vector. The command cumsum returns a vector whose elements are the cumulative sum of the previous elements, the command prod is the product of the vector elements, while the command diff returns a vector in which each element is given by its subtraction with the previous element. The command max and min return the largest and smallest elements of the vector, respectively, as well as their index. The command sort sorts the vector elements in ascending (by default) or descending order. The command mean computes the mean value, while the command median returns the median value. All these commands are suitable also for matrices by slightly changing their syntax.

Commands	*Results/Comments*
`a = [4  2  7  0  6]`	`a = 4  2  7  0  6` %Definition of vector *a*
`s = sum(a)`	`s = 19` %Sum the elements of *a*
`c = cumsum(a)`	`c = 4  6  13  13  19` %Cumulative sum. The result is obtained as [4,4+2,4+2+7, 4+2+7+0, 4+2+7+0+6]
`p =  prod(a)`	`p = 0` %Product of all elements.
`d = diff(a)`	`d = -2  5  -7  6` % Difference between two consecutive elements i.e., d(1)=a(2)-a(1) ,etc.

`[m,i] = max(a)`	`m = 7`
	`i = 4`
	%The largest value is assigned to the variable *m*, and its index is assigned to variable *i*.

`[m,i] = min(a)`	`m = 0`
	`i = 3`
	%The smallest value is assigned to the variable *m*, and its index is assigned to variable *i*.

`max(a)`	`ans = 7` %If no output variable is specified, only the largest value is returned.
`mean(a)`	`ans = 3.8000` %Mean value of the elements.
`median(a)`	`ans = 4` %Median value of the vector.
`sort(a)`	`ans = 0  2  4  6  7` %Sorting in ascending order.
`sort(a,'descend')`	`ans = 7  6  4  2  0` %Sorting in descending order.

Special Forms of Matrices

The command ones(M,N) creates a matrix of size MxN with ones. Typing ones(M) returns a square matrix with ones. Similarly, the command zeros(M,N) creates a matrix of size MxN with zeros. The command rand(M,N) returns an MxN matrix with random elements. The command eye(M,N) defines an MxN matrix with ones in the main diagonal zeros everywhere. The command magic(M) creates a square matrix with the property that the sum of each row or column is equal.

Commands	Results/Comments

`ones(2,3)`	`ans = 1 1 1`
	`1 1 1`
	%Matrix of size 2x3 with ones.

`zeros(1,4)`	`ans = 0 0 0 0` %Matrix of size 1x4(or vector of length) with zeros.

`rand(3)`	`ans = 0.8600 0.9000 0.4600`
	`0.6000 0.1000 -0.3000`
	`0.4540 0.6023 0.2700`
	%Matrix of size 3x3 with random elements. If there is one input argument to the command, the obtained matrix is square.

`eye(4,2)`	`ans = 1 0`
	`0 1`
	`0 0`
	`0 0`
	%Magic of size 4x2 with ones in the main diagonal and zeros elsewhere.

`eye(3)`	`Ans = 1 0 0`
	`0 1 0`
	`0 0 1`
	%The identity matrix 1 of size 3x3.

`A = magic(3)`	`A = 8 1 6`
	`3 5 7`
	`4 9 2`
	%Magic matrix

Symbolic Variables

In MATLAB, a variable type is the symbolic variable (or object). A symbolic variable is defined by the command sym and syms. The use of symbolic variables allows the computation of limits, integrals, derivativesetc.

Part II- Plotting Signals in MATLAB

MATLAB is a very reliable and power full tool for plotting. A graph is constructed as a set of points in two or three dimensions. These points are typically connected with a solid line. Commonly a graph of a function is required. However in MATLAB a plot is done using the vectors and matrices not functions.

Plotting in Two Dimensions

Suppose that we want to plot a function $y(x)$, where x is an independent variable. The procedure to plot $y(x)$ is as follows:

1. Vector x is created. Such as $a \leq x \leq b$, where a and b are scalars.
2. The function $y(x)$ will be plotted over the interval $[a, b]$.
3. Create the vector y , which is of the same length as x. The two vectors have an equal number of elements.
4. The value of each element of y is the value of $y(x)$ calculated for each value of x.
5. Finally, the function $y(x)$ is plotted by typing ‘plot(x,y)’

Lab Task

Plot the function $y(x) = x^2, -2 \leq x \leq 2$

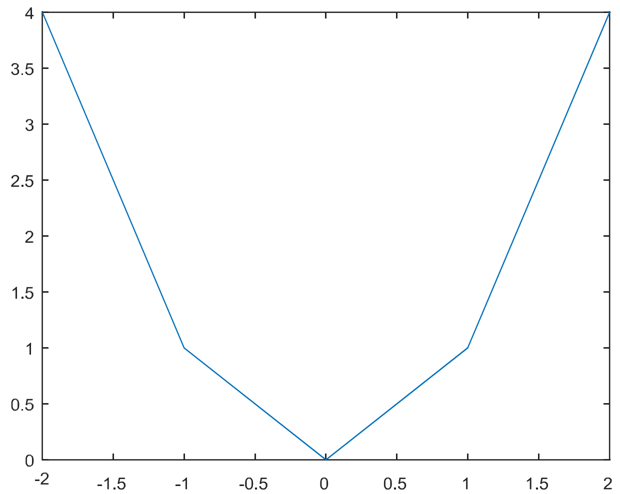
```
x=-2:2 % independent variable ,length of the plot
length (x)
y=x.^2 % Function
length (y)
plot(x,y)
```

x = -2 -1 0 1 2

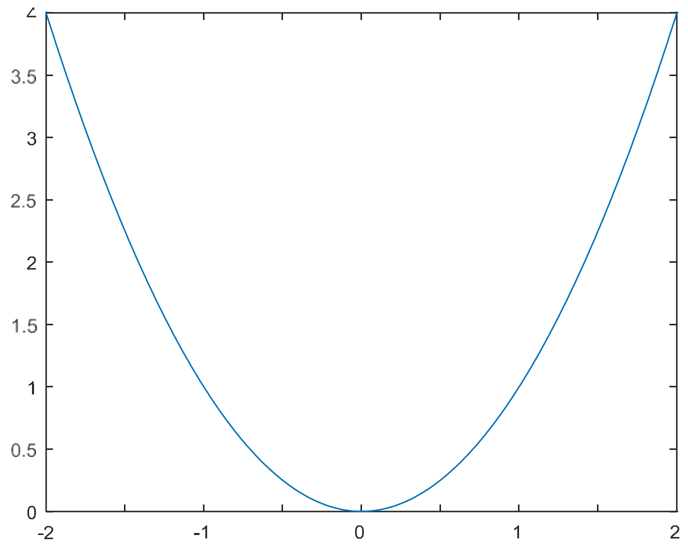
ans = 5

y= 4 1 0 1 4

ans = 5



Write down modified MATLAB code to plot exactly like below



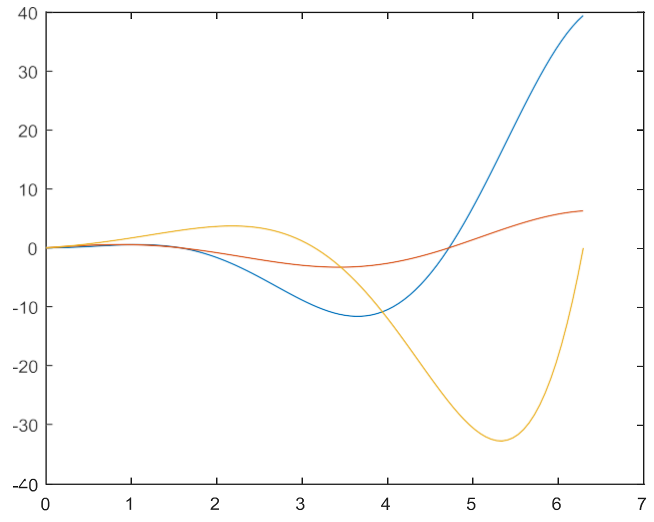
Plotting Several Function in One Figure

It is possible to plot more than one function in the name of figure by employing a different syntax of command Plot.

Lab Task

Plot the $y(x) = x^2 \cos(x)\,, g(x) = xcos(x), and\ f(x) = 2^x \sin(x)\,, \quad 0 \le x \le 2\pi$ in the same figure.

```
x = linspace(0,2*pi,100) % linspace could be used to create a vector.
x=0:pi/50:2*pi    % this is same value as above. Both method are correct
y = (x.^2).*cos(x);
g = x.*cos(x);
f = (2.^x).*sin(x);
plot(x,y,x,g,x,f)
```



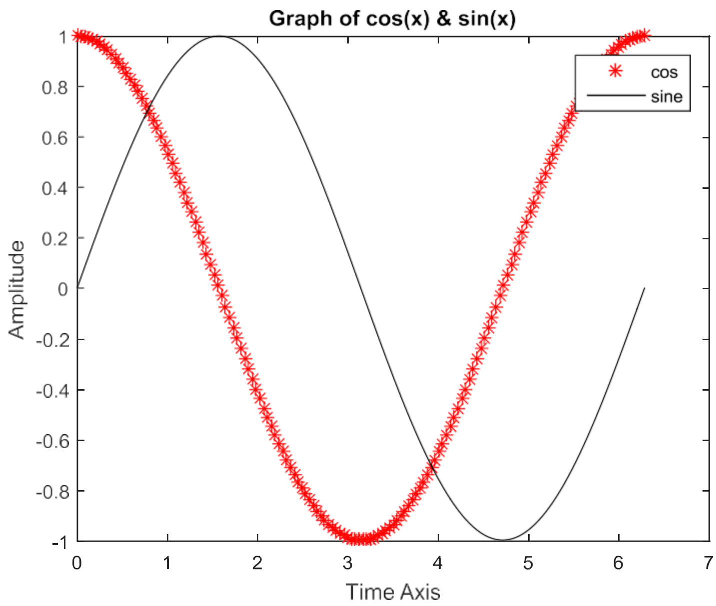
In the previous examples, the functions were plotted with predefined colors and line type (solid). It is possible to create a graph using colors, symbols used to draw the points, and type of line that connects the points of your choice. This is achieved by applying one more input argument to the command plot. The new argument is a series of special character given in single quotes. The available special characters are presented in the table below.

Symbols	*Color*	*Symbol*	*Point Type*	*Symbol*	*Line Type*
B	Blue	.	Point	-	Solid
G	Green	o	Circle	:	Dotted
R	Red	x	x-mark	-.	Dashdot
C	Cyan	+	Plus	_	Dashed
M	Magenta	*	Star		
Y	Yellow	s	Square		
K	Black	d	Diamond		
W	White	<,>	Triangle		
		P	Pentagram		
		h	Hexagram		

Formatting a Figure

The command grid on adds lines to the graph, while the command grid off removes the grid lines. Simply typing grid is switch between the two modes. Text besides the *x-axis* and *y-axis* can be added using the commands xlabel and ylabel, respectively. A graph title is inserted by the Command title.

```
% Program to  understand formatting a plot: axis labeling, title and legend %

x = linspace(0,2*pi,150);
plot(x,cos(x),'r*',x,sin(x),'k')
xlabel('Time Axis')
ylabel('Amplitude')
title('Graph of cos(x) & sin(x)')
legend('cos','sine')
```



Plotting in Different Figures

Up to this point, all plots were made in single figure named *Figure 1*. By typing at the command prompt figure, a new figure with name *Figure 2* appears without closing the old figures. The command subplot(m,n,p) or subplot(mnp) splits the figure window into $m \times n$small subfigures and makes the *pth* subfigure active.

Lab Task

Plot the functions that were plotted in last lab task in the following two ways.

1. In two different figures
2. In a figure but separately using subplot.

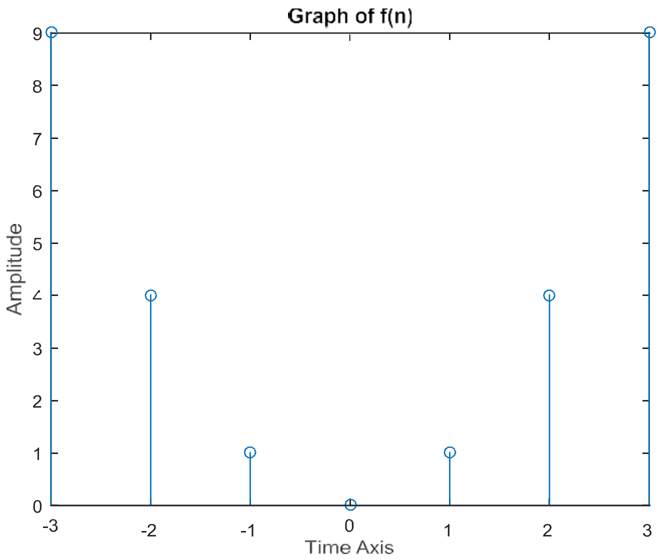
Plotting the Continuous Time and Discrete Time Functions

A discrete time function is a function of the form $f[n], n \in z, where\ z\ denotes\ the\ set\ of\ integers.$ In this case the appropriate command for plotting the function is 'stem(n,f)'

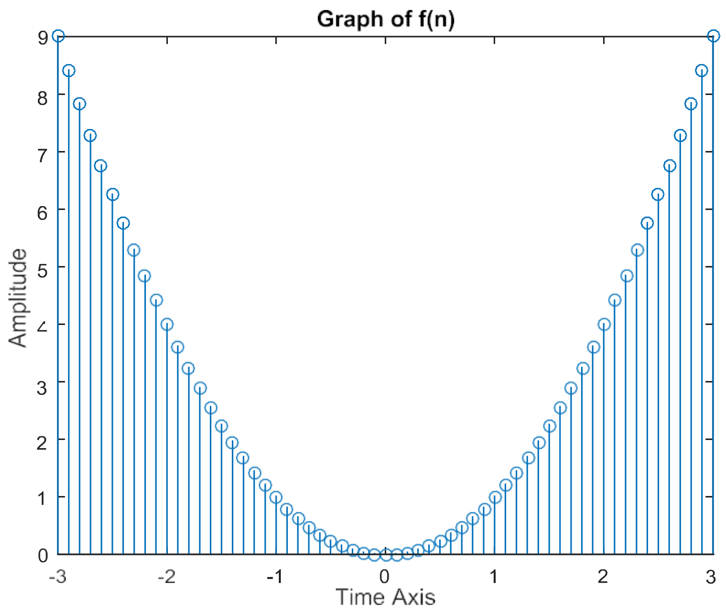
Lab Task

Plot the discrete function $f[n] = n^2, where -2 \le n \le 2.$

```
n = -3:3
f= n.^2
stem(n,f)
xlabel('Time Axis')
ylabel('Amplitude')
title('Graph of f(n)')
```



Modify the above lab task to create figure like below.



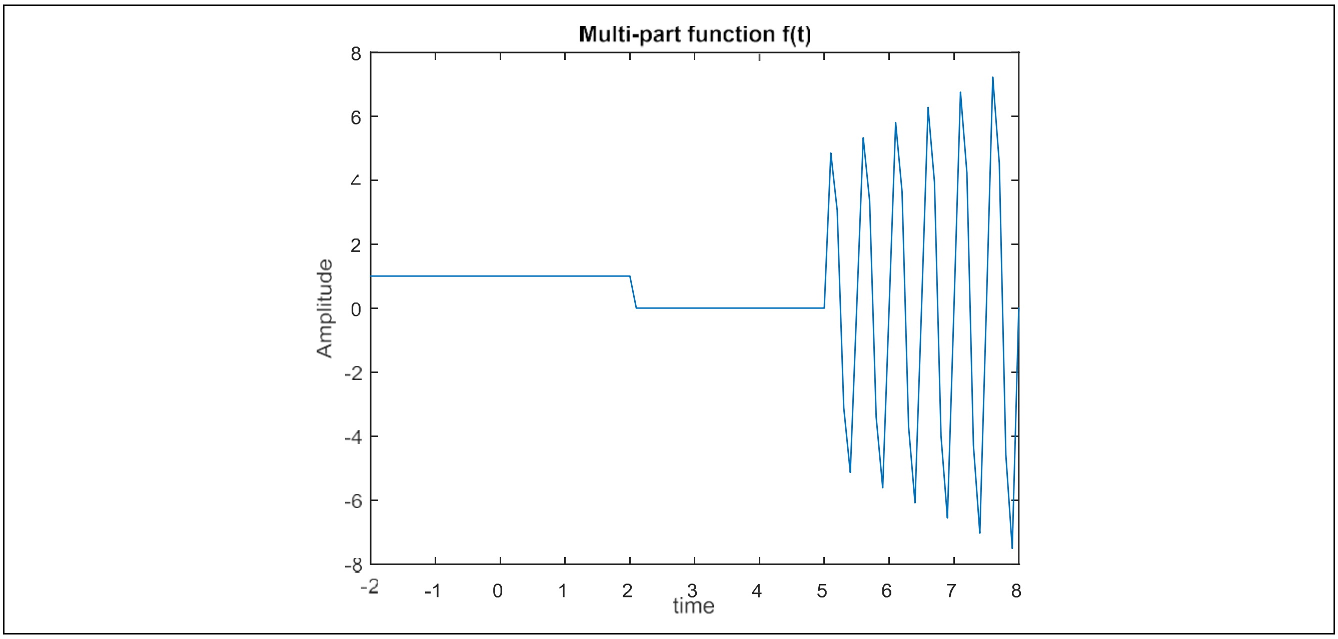
Now we will discuss the way of defining and plotting functions with more than one part.

Lab Task

Plot the following function

$$f(t) = \begin{cases} 1, & -2 \le t \le 2 \\ 0 & 2 < t < 5 \\ t\sin(4\pi t) & 5 \le t \le 8 \end{cases}$$

```
% Program to  understand piecewise functions plotting

t1=-2:.1:2;
t2=2.1:.1:4.9;
t3=5:.1:8;
f1=ones(size(t1));
f2=zeros(size(t2));
f3=t3.*sin(4*pi*t3);
t=[t1 t2 t3];
f=[f1 f2 f3];
plot(t,f)
title('Multi-part function f(t)')
xlabel( 'time')
ylabel( 'Amplitude')
```


Multi-part function f(t)
Amplitude
time
8
6
4
2
0
-2
-4
-6
-8
-2
-1
0
1
2
3
4
5
6
7
8

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:____________________________________Date: __________________________

LAB # 2: To Describe the Periodic and Aperiodic Continuous and Discrete Time Signals Using Elementary Signals in MATLAB

Objectives

After completing this lab, the student will be able to:

- ✓ Describe and show the basic Continuous and Discrete time signals (sinusoidal and complex exponential) by using elementary signals (unit step and unit impulse).

Pre Lab

Part I- Characterization of Signal by Variable Type

Continuous Time Signals

A signal is called continuous-time (or analog) signals if the independent (time) is defined in continuous interval. For $1-D$ signals, time domain of signal is continuous interval of the axis. In other words, for continuous-time signals the independent variable t is continuous. Moreover, the dependent value that usually denotes the amplitude of the signal is also continuous variable. An example of such a signal is speech as a function of time. An analog Signal is expressed by a function $x(t)$, where t takes real values.

Discrete Time Signals

A signal is called a discrete-time signal if the independent variable (time) is defined in a discrete interval (e.g., the set of integer number), while the dependent variable is defined in a continuous set of values. In the following example, the discrete-time signal $y[n] = \cos[n]$ is plotted. Note that when referring to discrete time the variable *n* is typically used to represent the time.

A discrete time signal x[n] is usually obtained by sampling a continuous-time signal $x(t)$ at a constant rate. Suppose Ts is the sampling period, that is every T_s we sample the value of $x(t)$. Suppose also that $n \in \mathbb{Z}$ *i-e.* $n = 0, \pm 1, \pm 2, \pm 3, \dots \dots$ The sequence of the sample is derived from the continuous time signal $x(t)$.

Digital Signals

Digital signals are the signals that both independent and dependent variables take values from a discrete set. In the following example, the signal $y[n] = \cos[n]$ is again plotted, but we use the command round to limit the set of values that $y[n]$ can take, That is, $y[n]$ can be $-1, 0 \; or \; 1$.

Part II - Basic Continuous Time Signals

We present the basic continuous Time Signals along with the way they are implemented and plotted in MATLAB.

Sinusoidal Signal

The first basic category presented is that of sinusoidal signal. This type of signal is of the form

$$x(t) = A\cos(\omega t + \theta)$$

Where

$$\omega = \; angular\ frequency, given\ in\ rad/s,$$
$$A = the\ amplitude\ of\ the\ signal, and$$
$$\theta = the\ phase (in\ radians).$$

Sinusoidal signals are periodic signals with fundamental period T given by

$$T = 2\pi/\,\omega.$$

Finally, a useful quantity is the frequency **f** given in Hertz. Frequency f is defined by

$$f = \omega/2\pi.$$

Complex Exponential Signals

Another signal highly associated with sinusoidal signal is the complex exponential signal. $A\,e^{j\Omega t+\theta}$ is also periodic with fundamental period given by $T = \frac{2\pi}{\omega}$. According to derived straightforward from Euler's Formula

$$A\,e^{j\Omega t+\theta} = A\,(\cos(\Omega t+\theta) + j\sin(\Omega t+\theta)).$$

Exponential Signals

Exponential signals are signals of the form

$$x(t) = Ae^{bt}.$$

If $b > 0$, $x(t)$ is an increasing function while

if $b < 0$, $x(t)$ is a decreasing function.

At $t = 0$ the signal takes the value

$$x(0) = A\ as\ e^{bt} = 1.$$

Unit Step Function

Another basic signal is unit step function u(t). The unit step function is given by

$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

i.e. it is not defined at t=0. In the following example three different methods of defining and plotting the unit step function are presented. `

The MATLAB command that generates the unit step function is the command heaviside(t).

Unit Impulse or Dirac Delta Function

The Dirac delta $\delta(t)$, strictly speaking, is not a function but is defined through its properties. The main property is

$$\int_{-\infty}^{\infty} f(t)\delta(t) = f(0),$$

Where $f(.)$ is an arbitrary function. Suppose that $f(t) = 1$, t$\in(-\infty,\infty)$. Then

$$\int_{-\infty}^{\infty} \delta(t) = 1.$$

For special reasons, $\delta(t)$ can be loosely defined as a function that is infinite at t=0 and zero elsewhere. This is the way that $\delta(t)$ is implemented from the MATLAB programmers. The mathematical expression is

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases}$$

An alternative definition for the Dirac function that is usually applicable when dealing with discrete-time signal is given now. In this case, we refer to $\delta(t)$ as the *Delta* or the *Kronecher* function. The mathematical definition of delta function is

$$\delta(t) = \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases}$$

Part III – Basic Discrete Time Signals

Now we will discuss the discrete time signals. Discrete time signals are the sequences. A sequence $x[n].n \in (-\infty\ \infty)$ consists of infinite (real or complex) elements or samples. DT signals are treated in the same way as the continuous time signals. However there are two main differences.

1. The definition of the time. For the discrete time signals, time is defined by the step 1.
2. The graph of the discrete time signals is obtained by the stem command. Stem is similar to plot but is suitable for discrete time signals.

Unit Impulse Sequence

The unit impulse sequence is the counter part of the dirac delta function when dealing with discrete time signals. It is also known as Kronecker delta. The mathematical expression of the $\delta[n]$ is :

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

In general, the unit impulse sequence is given by,

$$\delta[n - n_0] = \begin{cases} 1 & n = n_0 \\ 0 & n \neq n_0 \end{cases}$$

Unit Step Sequence

The unit step sequence is the counterpart of the unit step function when dealing with discrete time signals. The mathematical expression is

$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

In general case, the unit step sequence is given by

$$u[n - n_0] = \begin{cases} 1 & n \geq n_0 \\ 0 & n < n_0 \end{cases}$$

The command of the heaviside is not applicable to the unit step sequence.

Real Exponential Sequence

The mathematical expression of a real valued exponential sequence is $x[n] = a^n, a \in R,$ the sequence $x[n]$ is ascending if |a|>1, and descending if $|a| < 1$.

Sinusoidal Sequence

The sinusoidal sequence is defined by the expression of the form

$$x[n] = A\cos(\omega n + \phi) \qquad or \qquad x[n] = A\sin(\omega n + \phi)$$

Where

$A = Amplitude$
$\varphi = phase$
$\omega = frequency$

$$\frac{2\pi}{\omega} = \frac{2}{3}\pi$$

Pre Lab Tasks

1. Define and plot the signal $x(t) = \cos(\pi t)$.
2. Define and plot the real and the imaginary part of the signal $y(t) = e^{3jt}$, in the time of two periods.
3. **Rectangular Pulse Function:** The rectangular pulse function $pT(t)$ is rectangular pulse with unit amplitude and duration T. it is defined in terms of the unit step function $u(t)$ as

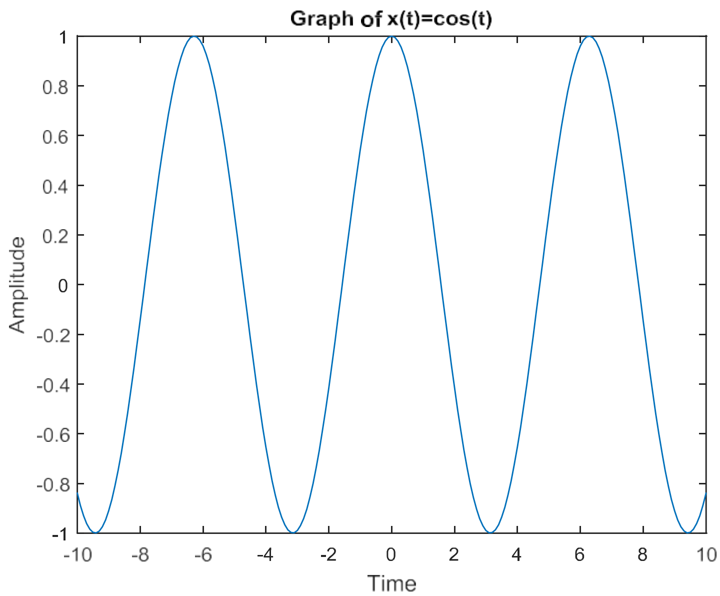
$$pT(t) = u\left(t + \frac{T}{2}\right) - u\left(t - \frac{T}{2}\right) = \begin{cases} 1, & -\frac{T}{2} \leq t \leq \frac{T}{2} \\ 0, & elsewhere \end{cases}$$

Plot the rectangular pulse function for the $T = \frac{Roll\ Number}{2}$

Lab Tasks

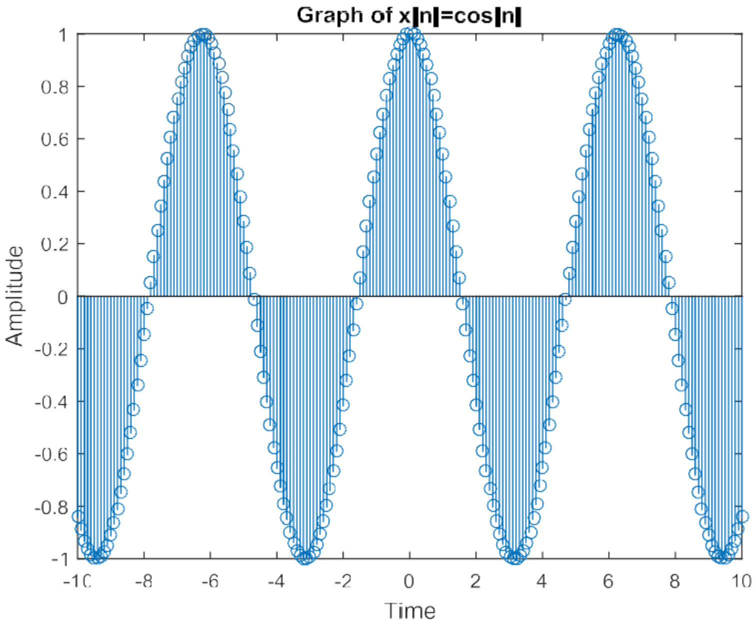
Lab Task

Plot the continuous time signal $x(t) = \cos(t)\ over\ the\ interval - Roll\ Number \leq t \leq Roll\ Number$.
Your plotted signal should look like as shown in the figure on next page



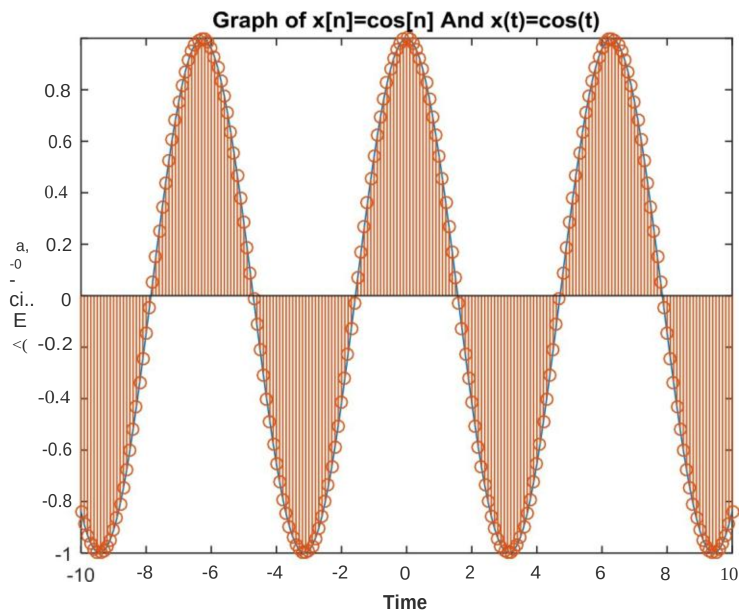
Lab Task

Plot the discrete time signal $x[n] = \cos[n]\ \ over\ the\ the\ interval - Roll\ Number\ \leq n \leq Roll\ Number.$



Lab Task

Modify the above lab tasks to plot the CT and DT signals together.



Lab Task

Plot the digital signal $x[n] = \cos(n), \;\; for - ROll\ Number \;\leq n \leq Roll\ Number$

Lab Task

Plot the signal $x(t) = 3\cos(3\pi t + \pi/3)$ in four periods.

Lab Task

Plot the following two signals together in the same plot.

$$x_1(t) = \cos(t) \quad and \quad x_2(t) = \sin(t + \frac{\pi}{2})$$

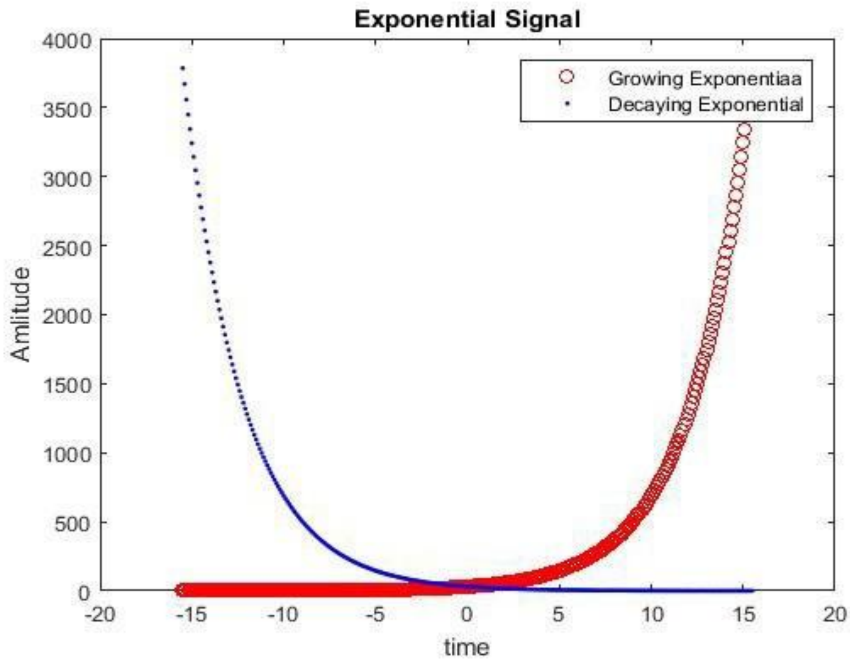
both are equal

Lab Task

Plot the Signal $x(t) = Ae^{\beta_1 t}$ and $y(t) = Ae^{\beta_2 t}$ in the time interval -T$\leq t \leq T$.

Where $A = Roll\ Number$

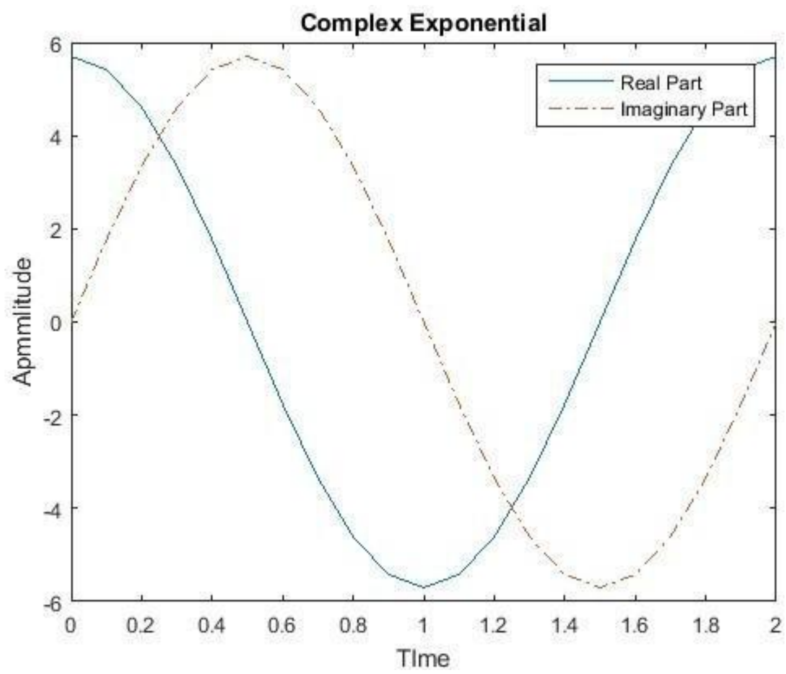
$$\beta_1 = \frac{A}{100} \quad , \beta_2 = -\frac{A}{100} \quad \text{and} \quad T = A/2$$



Lab Task

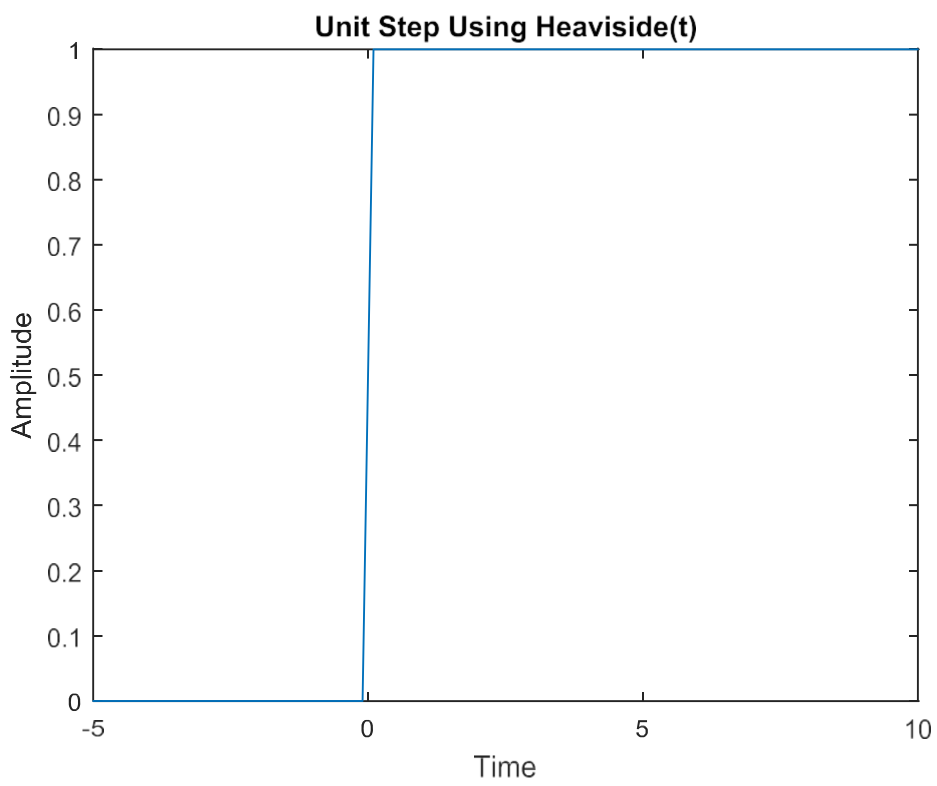
Plot the real and imaginary parts of signal $y(t) = e^{j\pi t + \frac{\pi}{3}}$ in the time of one period.

First of all find the period of the signal. $T = \frac{2\pi}{\omega} = \frac{2\pi}{\pi} = 2$



Lab Task

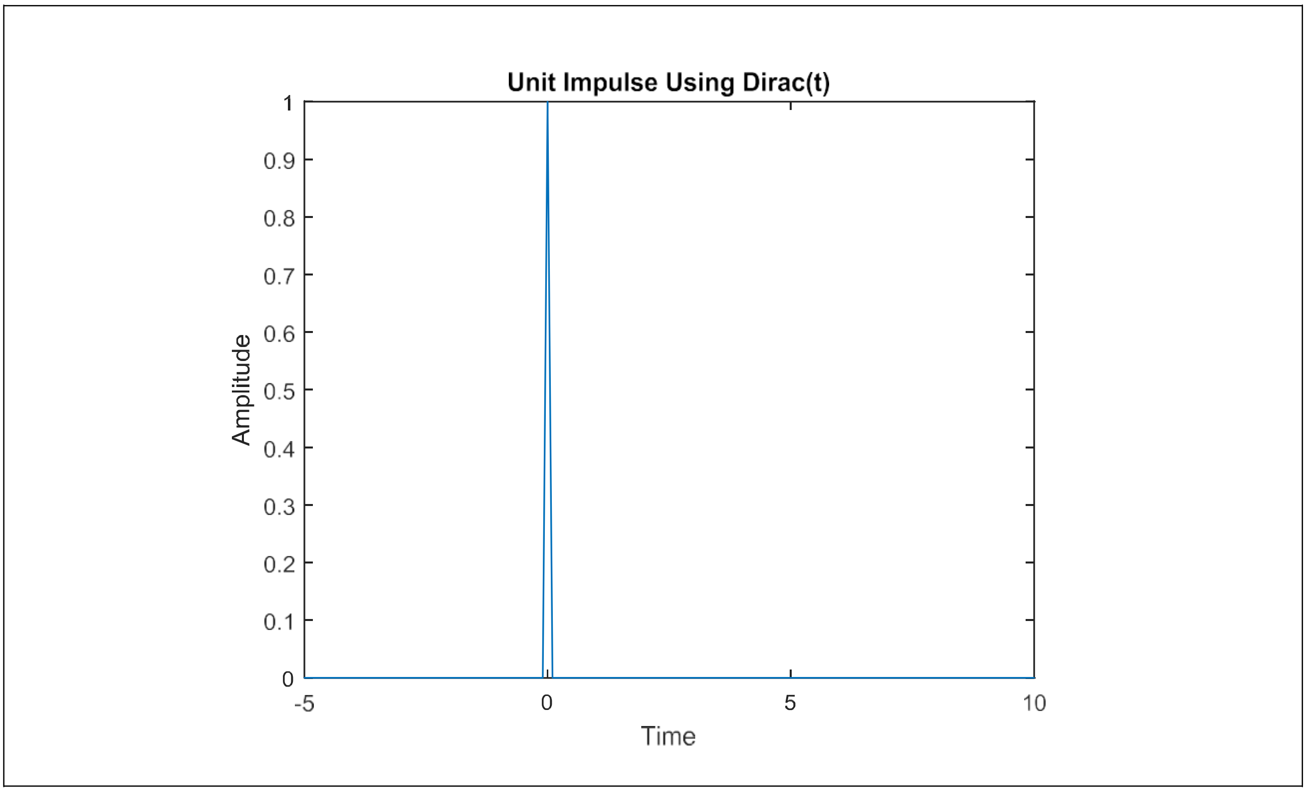
Implement the unit step function.



Plot the unit step function for $t_0 = 2$.

Lab Task

Plot the unit impulse function.


Unit Impulse Using Dirac(t)
Amplitude
Time
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-5
0
5
10

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:__________________________________Date: _______________________

LAB# 3: To Explain the Properties of Different Signal Types Using Signal Operations in MATLAB

Objectives

After completing this lab, the student will be able to:

- ✓ Display the periodic signals, casual signals, energy and power signals, even and odd signals, deterministic and stochastic signals using MATLAB.
- ✓ Show the results obtained after performing the composite operations on signals (time reversal, time scaling and time shifting) in MATLAB.

Pre Lab

Part I-Properties of Continuous Time Signals

Continuous Time Periodic Signals

A continuous-time signal $x(t)$ is periodic if there is a positive number T such that

$$x\,(t) = x(t+T) \quad \forall\, t$$

If T is the smallest positive value of T for which above equation is satisfied, then T is called the fundamental period of the signal. Moreover, if k $\in$ Z another expression that denotes the periodicity of a signal is

$$x(t) = \mathrm{x}(t+\ kT)\,, \forall t\ \in R$$

For example, a sinusoidal signal $x(t) = Acos(wt+\theta)$ is periodic with fundamental period $T = \frac{2\pi}{\omega}$ but $T = \frac{4\pi}{\omega}, T = \frac{6\pi}{\omega}, T = \frac{8\pi}{\omega}$, etc. are also periods of the signal $x(t)$.

Sum of Continuous Time Periodic Signals

Suppose that $x_1(t)\ and\ x_2(t)$ are periodic signals with periods $T_1\, and\, T_2$, respectively. In this chapter, the condition that must be fulfilled in order for the signal $x(t) = x_1(t)\ and\ x_2(t)$ to be also periodic is given. Moreover, the period of $x(t)$ is computed. Since $x_1(t)\ and\ x_2(t)$ are periodic, we get

$$x_1(t) = (t+mT_1) \qquad \text{and} \qquad x_2(t) = (t+kT_2)\,,\ k\ \in Z$$

Thus, $$x(t) = x_1(t) + x_2(t) = x_1(t+mT_1) + x_2(t+kT_2), k \in Z$$

Suppose that x(t) is indeed periodic with period T. Then, the following relationship holds:

$$x(t) = x(t+T) = x_1(t+T) + x_2(t+T)$$

Combining the equations we yields,

$$x_1(t+T) + x_2(t+T) = x_1(t+mT_1) + x_2(t+kT_2)$$

Equation is valid if

$$mT_1 = kT_2 = T \qquad\qquad where\ m,k \in Z$$

if m and k are prime numbers, then the period of the signal $x_1(t)$ and $x_2(t)$ is computed directly according to relation defined in above equation.

Discrete Time Periodic Signals

A discrete-time signal is periodic if there is a positive integer number N such that

$$x[n] = x[n+N], \qquad \forall n \in Z$$

A discrete-time signal x[n] is periodic if there are integer positive numbers $m,\ N,$ such that

$$w = \frac{2\pi m}{N}$$

Continuous Time and Discrete Time Casual Signal

A signal $x(t)$ or $x[n]$ is **casual** if it is zero for all negative values of time, i.e., $x\ (t) = 0,\ t < 0$. **Non-casual** signals are signals that have non zero values in both positive and negative time. Finally, a signal that is zero for all positive time is called **anti casual.**

Sum of Periodic Discrete-Time Periodic Signals

The sum *z*[*n*] = *x*[*n*] + *y*[*n*] of periodic signals *x*[*n*] with period *N*1, and *y*[*n*] with period *N*2 is periodic if the ratio of periods of the summands is rational—that is, Here *p* and *q* are integers not divisible by each other
If so, the period of $z[n]$ is $qN2 \ = \ pN1$

$$\frac{N_2}{N_1} = \frac{p}{q}$$

Even and Odd Signal / Sequence

A signal $x(t)$ is even (or has even symmetry or is an even function of t) if $x(-t) = x(t)$, $-\infty < t < \infty$ and the same hold for the sequence $x[n]$.

A signal $x(t)$ is odd (or has odd symmetry or is an odd function of t) if $x(-t) = -x(t)$, $-\infty < t < \infty$ and the same hold for the sequence $x[n]$.

Energy and Power Signals/ Sequence

An important classification of signals is between (finite) energy and (finite) power signals. The Energy E_x of a continuous time signal $x(t)$ is computed according to

$$E_x = \lim_{T\to\infty} \frac{1}{2\pi} \int_{-T}^{T} |x(t)|^2 dt = \int_{-\infty}^{\infty} |x(t)|^2 dt$$

Where $|x(t)|$ is the absolute value of $x(t)$. A signal $x(t)$ is an energy signal if its energy is definite,
i.e., $0 \le E_x < \infty$.

The Power P_x of a signal $x(t)$ is given by

$$P_x = \lim_{T\to\infty} \frac{1}{2\pi} \int_{-T}^{T} |x(t)|^2 dt$$

If $x(t)$ is periodic, the power is computed from the relationship.
Likewise the continuous time signals, discrete time signals can also be classified between (finite) energy and (finite) power signals. The Energy E_x of a discrete time signal $x[n]$ is computed according to

$$E_x = \sum_{-\infty} |x[n]|^2$$

Where $|x(t)|$ is the absolute value of $x[n]$. A signal $x[n]$ is an energy signal if its energy is definite, i.e, $0 \leq E_x < \infty$.

The Power P_x of a signal $x[n]$ is given by

$$P_x = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]^2$$

A signal $x[n]$ is called a power signal if $0 \leq P_x < \infty$ In order to calculate the energy or the power of a signal, recall that the command $limit(F, x, a)$ computes the limit of the function F when the symbolic variable x tends to a.

Part II- Signal Transformations

Transformations of the time Variable for Continuous-Time Signals

In many cases, there are signals related to each other with operations performed on the independent variable, namely, the time. In this section, we examine the basic operations that are performed on the independent variable.

Time Reversal or Reflection

The first operation discussed is the signal's reflection. A signal $y(t)$ is a reflection or a reflected version of $x(t)$ about the vertical axis if $y(t) = x(-t)$.

The operation of time reversal is actually an alternation of the signal values between negative and positive time. Assume that x is the vector that denotes the signal $x(t)$ in time t. The MATLAB statement that plots the reflected version of x(t) is plot(-t,x).

$$P_x = \frac{1}{T}\int_{t_o}^{t_{o+T}} |x(t)^2| \, c$$

A signal $x(t)$ is called a power signal if $0 \leq P_x < \infty$ In order to calculate the energy or the power of a signal, recall that the command $limit(F, x, a)$ computes the limit of the function F when the symbolic variable x tends to a.

Time Scaling

The second operation discussed is timing scaling. A signal $x_1(t)$ is compressed version of $x(t)$ if $x_1(t) = x(at)$, $a > 1$. The time compression practically means that the time duration of the signal is reduced by a factor a. On the other hand, a signal $x_2(t)$ is expanded version of $x(t)$ if $x_2(t) = x(at)$, $a < 1$. In this case, the time duration of the signal is increased by a factor 1/a.

In order to plot in MATLAB ~~a~~ time-scaled version of $x(t)$, namely, a signal of the form $y(t) = x(at)$, the statement employed is plot $(\frac{1}{a} * t, x)$. In contrast to what someone would expect the vector of time t must be multiplied by 1/a and not a.

Time Shifting

A third operation performed on time is one of time shifting. A signal $y(t)$ is a time-shifted version of $x(t)$ if $(t) = x(t - t_0)$, where t_0 is the time shift. If $t_0 > 0$, the signal $y(t) = x(t - t_0)$ is shifted by t_0 units to the right (i.e., toward +∞); while if $t_0 < 0$, the signal $y(t) = x(t - t_0)$ is shifted by t_0 units to the left. The time shifting is implemented in MATLAB in an opposite way to what may be expected. More specifically, in order to plot the signal $x(t - t_0)$ the corresponding MATLAB statement is $plot(t + t0, x)$.

Pre Lab Tasks

1. Consider the signal $x(t) = te^{-0.1t}\cos(t)\,, 0 \le t \le 20$. Plot
   a. The signal $x(t)$.
   b. Then even decomposition $x_e(t)$ of $x(t)$.
   c. The odd decomposition $x_o(t)$ of $x(t)$.
   d. The signal $y(t) = x_e(t) + x_o(t)$.
2. Suppose that $x(t) = \begin{cases} t & 0 \le t \le 2 \\ 4 - t & 2 \le t \le 4 \end{cases}$. Plot the signals $x(t)$, $x(-t)$, $x(t/2)$, $x(2 + 4t)$ and $x(-2 - 4t)$.

Lab Tasks

Lab Task

Verify that the signal $x(t) = \sin(t)$ is periodic.

31

Lab Task

Plot the signal $x(t) = \cos(t) + \sin(3t)$ in time of three periods.

T1=2pi/w
T=2w/1

T2=2pi/w
T=2w/3

mT1 = kT2 =T

Lab Task

Plot the causal, non-causal and anti-causal signal and sequence.

CAUSAL FOR POSITIVE
NON CAUSAL BOTH POSITIVE AND NEGATIVE
ANTI CAUSAL NEGATIVE

Lab Task

Find out if the signals $x(t) = t^2$ and $y(t) = t^3$ are even or odd.

E O

Lab Task

Decompose the unit step sequence into even and odd parts over the interval $-Roll\ Number \leq n \leq Roll\ Number$ and verify your result by constructing unit step signal from the even and odd parts.

Lab Task

Verify that $x(t) = u(t)$ is a power-type signal while $x(t) = u(t) - u(t-1)$ is an energy-type signal.

Lab Task

Compute the energy and the power of the signal $x(t) = 2cos(\pi t)$ in one period time.

Lab Task

Suppose that $x(t) = te^{-t}, -1 \leq t \leq 3$. Plot the signal $x(-t)$.

Lab Task

Consider again the continuous-time signal $x(t) = te^{-t}, -1 \leq t \leq 3$. We will plot the signal $x_1 = x(2t)$, which is a time compression of $x(t)$ by a factor a=2; and the signal $x_2 = x(0.5t)$, which is a time expansion of $x(t)$ by a factor 1/a=2.

Lab Task

The signal $x(t) = te^{-t}$, $-1 \leq t \leq 3$ is again considered. We will plot the signals $x_1(t) = x(t-2)$, that is, a shifted version of $x(t)$ by two units to the right (here $t_0 = 2$) and $x_2(t) = x(t+3) = x(t-(-3))$, that is, a shifted version of $x(t)$ by 3 units to the left (here $t_0 = -3$.)

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:__________________________________Date: _______________________

LAB # 4: To Explain the Properties of the System Using I/O Relationship in MATLAB

Objectives

After completing this lab, the student will be able to:

- ✓ Show the properties (causality, linearity, stability and time invariance,) of different systems using MATLAB.

Pre-Lab

Properties of System

In this section, we introduce the basic system properties. An illustrative example accompanies each property. Of course, an example does not prove a property. However, the illustrated examples are carefully selected in order to correspond with the introduced property. The following properties apply to both continuous-time and discrete-time systems.

Causal and Non-Causal Systems

A system is causal if the system output $y(t_0)$ at time $t = t_0$ does not depend on values of the input $x(t)$ for $t > t_0$. In other words, for any input signal $x(t)$, the corresponding output $y(t)$ depends only on the present and past values of $x(t)$. So, if the input to a causal system is zero for $t < t_0$ the output of this system is also zero for $t = t_0$. Correspondingly, a discrete-time system is causal if its output $y[n_0]$ at time $n = n_0$ depends only on the values of the input signal $x[n]$ for $n \leq n_0$. All natural system is causal. However, in engineering there are many non causal systems. For example off-line data processing is a non causal system.

Static (Memory Less) and Dynamic (With Memory) System

A system is static or memory less if for any input signal $x(t)$ or $x[n]$ the corresponding output $y(t)$ or $y[n]$ depends only on the value of the input signal at the same time. A non-static system is called dynamic or dynamical.

Linear and Non-Linear System

Let $y(t)$ denote the response of a system S to an input signal $x(t)$, that is, $y(t) = S\{x(t)\}$. System S is a linear if for any input signals $x_1(t)$ and $x_2(t)$ and any scalars a_1 and a_2 the following relationship holds:

$$S\{a_1x_1(t) + a_2x_2(t)\} = a_1S\{x_1(t)\} + a_2S\{x_2(t)\}$$

In other words, the response of a linear system to an input that is a linear combination of two signals is the linear combination of the responses of the system to each one of these signals. The linearity property is generalized for any number of input signals, and this is often referred to as the principle of superposition. The linearity property is a combination of two other properties the additive property and the homogenous property. A system S satisfies the additive property if for any input signals$x_1(t)$ and $x_2(t)$.

$$S\{x_1(t) + x_2(t)\} = S\{x_1(t)\} + S\{x_2(t)\}$$

While the homogeneity property implies that for any scalar a and any input signal $x(t)$,

$$S\{ax(t)\} = aS\{x(t)\}$$

Time-Invariance

A system is time invariant, if a time shift in the input signal results in the same time shift in the output signal. In other words, if $y(t)$ is the response of a time invariant system to an input signal $x(t)$, then the system response to the input signal $x(t - t_0)$ is $y(t - t_0)$. The mathematical expression is

$$y(t - t_0) = S\{x(t - t_0)\}.$$

Equivalently, a discrete-time system is time or (more appropriately) shift invariant if,

$$y[n - n_0] = S\{x[n - n_0]\}.$$

From the above equations, we conclude that if a system is time invariant, the amplitude of the output signal is the same independent of the time instance the input is applied. The difference is a time shift in the output signal. A non-time – invariant system is called time – varying or time - variant system.

Stability

Stability is a very important system property. The practical meaning of a stable system is that for a small applied input the system response is also small (does not diverge). A more formal definition is that a system is stable or bounded- input bounded-output (BIBO) stable if the system response to any bounded-input signal is a bounded-output signal. The mathematical expression is as follows. Suppose that a positive number $M < \infty$ exists, such that $|y(t)| \leq N$. A non-stable system is called unstable.

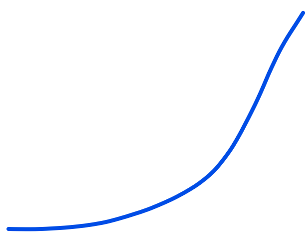
Pre lab Tasks

1. Find out if the system described by the I/O relationship $y[n] = x[2n]$ is static or dynamic?
2. Determine if the system with I/O relation $y(t) = 3x(t) + 2cos(\pi t/3)$ is linear and time invariant.
3. Determine if the discrete-time systems with I/O relationship $y[n] = x[n] + x[n-1]$ is stable.

Lab Tasks

Lab Task

Using the input signal $x(t) = u(t) - u(t-1)$ find out if the system described by the I/O relationship $y(t) = 3x(t)$ and $y(t) = x(t-1)$ are static or dynamic.



Lab Task

Determine if the discrete-time system described by the I/O relationship $y[n] = x^2$ and $y[n] = x[\frac{n}{2}]$ are static or dynamic. Use the input signal $x[n]$ = [0 1 2 3 4], $-1 \leq n \leq 3$.

y[n] = x[n/2]
y[4] = x[4/2]
y[4] = x[2]

Lab Task

Let $x_1(t) = u(t) - u(t-2)$ be input signals to the systems described by the I/O relationship $y(t) = 2x(t)$ and $y(t) = x^2(t)$. Determine if the linearity property holds for these two systems.

in this we have assumed the x2 and a,b

Lab Task

Determine if the linearity property holds for the discrete-time system described by the I/O relationship $y[n] = 2^{x[n]}$ and $y[n] = nx[n]$. Consider the input signal $x_1[n] = 0.8^n$, $0 \leq n \leq 5$ $0 \leq n \leq 5$ and $x_2[n] = \cos[n]$, $0 \leq n \leq 5$.

The linearity property holds for a discrete-time system if

$$S\{a_1 x_1[n] + a_2 x_2[n]\} = a_1 S\{x_1[n]\} + a_2 S\{x_2[n]\}$$

5

Lab Task

Suppose that the response of a system S to an input signal $x(t)$ is $y(t) = te^{-t}x(t)$. Determine if this system is time invariant by using the input signal $x(t) = u(t) - u(t-5)$.

Lab Task

Consider a system described by the I/O relationship $y(t) = 1 - 2x(t-1)$. Determine if this is a time invariant system by using the input signal $x(t) = \cos(t)\,[u(t) - u(t-10)]$.

6

Lab Task

Suppose that an input signal $x(t) = \cos(2\pi t)$ is applied to two systems described by the I/O relationship $y_1[t] = x^2[t]$ and $y_2(t) = tx(t)$. Determine if these two system as stable.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:_________________________________Date: ______________________

LAB # 5: To Sketch the Response of Linear Time Invariant Systems by Performing Convolution Using MATLAB

Objectives

After completing this lab, the student will be able to:

- ✓ Construct and display the convolution and deconvolution of Continuous and Discrete time signals using MATLAB.

Pre Lab

Computation of Continuous-Time Convolution

The impulse response of a linear time-invariant system completely specifies the system. More specifically, if the impulse response of a system is known one can compute the system output for any input signal. We now present on of most important topics in signals and systems theory.

The response (or output) of a system to any input signal is computed by the convolution of the input signal with the impulse response of the system.

Suppose that $y(t)$ denotes the output of the system, $x(t)$ is the input signal, and $h(t)$ is the impulse response of the system. The mathematical expression of the convolution relationship is

$$y(t) = x(t) * h(t)$$

Where, the symbol * denotes convolution, and it must not be confused with the multiplication symbol. The calculation of the convolution between two signals involves the computation of an integral. The complete form of is

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

By alternation variables τ and t, becomes

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

The Command conv

The computational process followed in the previous lab is the analytical way of deriving the convolution between two signals. In MATLAB, the command $conv$ allows the direct computation of the convolution between two signals.

De-convolution of Continuous-Time Signals

Suppose that the impulse response of a system $h(t)$ and the output of a system $y(t)$ are available and we want to compute the input signal $x(t)$ that was applied to the system in order to generate the output $y(t)$.

This process is called deconvolution and is implemented in MATLAB using the command deconv. The syntax is $x = deconv(y, h)$ where x is the input vector, h is impulse response vector, and y is the system response vector. The deconvoution process is also useful for determining the impulse response of a system if the input and output signals are known. In this case, the command is $h = deconv(y, x)$.

Remarks

The commutatively property is not valid for the $deconv$ command; hence, the output signal must be the first input argument of the command $deconv$.

Computation of Discrete-Time Convolution

The process that has to be followed to analytically derive the convolution sum is similar to the one followed in the continuous-time case. First, the input and impulse response signals are plotted in the k-axis. Then one of the two signals is reversed about the amplitude axis, and its reflection is shifted form $-\infty, +\infty$ by changing appropriately the value of n. the output of the system is computed from the overlapping values of $x[n]$ and $h[n-k]$ according to the convolution sum $y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$.

Discrete-time Convolution

In this section, several examples on computing the convolution between two discrete-time signals by using the command conv are given. There are two basic principles that should be considered when commutating the convolution between two discrete-time signals $x[n]$ and $h[n]$.

The Two Principles of Convolution
1. Suppose that the length of vector *x* is *N* and the length of vector h is *M.* The outcome of the command $y = conv(x, h)$ is a vector y of length $M + N - 1$. In other words, $length(y) = length(x) + length(h) - 1$.   2. If the nonzero values of $x[n]$ are in the interval $[a_x, b_x]$ and the nonzero values of $h[n]$ are in the interval $[a_h, b_h]$, then the nonzero values of the output $y[n]$ are in the interval $[a_x + a_h, b_x + b_h]$

De-convolution of Discrete-Time Signals

Suppose that the impulse response of a system $h[n]$ and the output of a system $y[n]$ are available and we want to compute the input signal $x[n]$ that was applied to the system in order to generate the output $y[n]$. This process is called deconvolution and is implemented in MATLAB using the command deconv. The syntax is $x = deconv(y, h)$, where x is the input vector, h is impulse response vector, and y is the system response vector. The deconvoution process is also useful for determining the impulse response of a system if the input and output signals are known. In this case, the command is $h = deconv(y, x)$.

Remarks

The commutatively property is not valid for the deconv command; hence, the output signal must be the first input argument of the command deconv.

Pre Lab Tasks

1. A system is described by the impulse response $h(t) = t, 0 \le t \le 10$. Compute and plot the response of the system to the input signal $x(t) = 0.8^t, 0 \le t \le 10$.
2. A system is described by the impulse response $h(t) = e^{-2t}u(t-1)$. Compute and plot the response of the system to the input signal $x(t) = u(t) - u(t-2)$.
3. Compute the input of system for which you have computed the system response in the above task and verify the answer.
4. Compute the convolution between the complex sequence $x = [3 + 2j, 1 + j, 4 + 6j]$ and $h = [1 - 2j, j, 3 - 2j, 2]$.

5. Suppose that a discrete-time system is described by the impulse response $h(t) = 0.8^t, 0 \leq t \leq 10$. Compute and plot the response of the system to the input signal.

$x[n] = \sqrt{\frac{1}{n}}, 1 \leq n \leq 5$

Lab Tasks

Lab Task

A linear time-invariant system is described by the impulse response

$$h(t) = \begin{cases} 1-t & 0 \leq t \leq 1 \\ 0 & elsewhere. \end{cases}$$

Calculate the response of the system to the input signal

$$x(t) = \begin{cases} 1, & 0 \leq t \leq 2 \\ 0 & elsewehere. \end{cases}$$

convulation is commutative
deconv isnt commutative

Lab Task

We will consider the same signals used in the previous example. Therefore, the problem is to compute the impulse response $h(t)$ of a system when the response of the system to the input $x(t)$ is the signal $y(t)$, which is depicted in the previous figure.

Lab Task

Suppose that a linear time-invariant (LTI) system is described by the impulse response $h(t) = e^{-t}u(t)$. Compute the response of the system to the input signal .

$$x(t) = \begin{cases} 0.6, & -1 < t < 0.5 \\ 0.3, & 0.5 < t < 3 \\ 0, & t < -1 \; and \; t > 3 \end{cases}$$

Lab Task

Suppose that the impulse response of a system is

$$h[n] = \begin{cases} n, & -5 < n \leq 5 \\ 0, & elsewhere \end{cases}$$

Compute the response of the system to the input signal

$$x[n] = \begin{cases} 1, & 0 \leq n \leq 4 \\ 0, & elsewhere \end{cases}$$

Lab Task

Write a function that computes and plots the convolution of two sequences. The function must accept as arguments the two signals and the time intervals in which the sequence are defined. Execute your function for

$x[n] = n^2, -2 \leq n \leq 2 \;\; and \;\; h[n] = \frac{1}{n+2}, -1 \leq n \leq 3.$

Lab Task

Compute the impulse response of system for which you have computed the system response in the above task and verify the answer.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:________________________________Date: ______________________

LAB # 6: To Sketch the Line Spectrum of Periodic Signals Using Properties of Fourier Series Coefficients in MATLAB

Objectives

After completing this lab, the student will be able to:

- ✓ Construct the complex exponential fourier series coefficients.
- ✓ Display the line spectrum of periodic signals.

Pre-Lab

Fourier series of Periodic Signals

Periodic Signals

We define a continuous-time signal $x(t)$ to be periodic if there is a positive

$$x\,(t+T) = x(t) \quad \forall\, t$$

The fundamental period T_0 of $x(t)$ is the smallest positive value of *T* is satisfied, and $1/T_0 = f_0$ is referred to as the fundamental frequency.

Two basic examples of periodic signals are the real sinusoidal signal:

$$x(t) = \cos(\omega_0 t + \emptyset)$$

And the complex exponential signal:

$$x(t) = e^{j\omega_0 t}$$

Where $\omega_0 = 2\pi f_0$ is called the *fundamental angular frequency.*

Complex Exponential Fourier series Representation

The complex exponential Fourier series representation of a periodic signal $x(t)$ with fundamental period T_0 is given by:

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jkw_0 t}$$

Where a_k are known as the complex Fourier coefficient is given by:

$$a_k = \frac{1}{T}\int_T x(t) e^{jkw_0 t}$$

Plotting the Line Spectra

The graph of magnitudes $|a_k|$ versus frequency is known as magnitude spectrum, while the graph of the angles of a_k versus frequency is known as phase spectrum. They magnitude spectrum and phase spectrum are collectively known as “Line Spectra”.

Trigonometric Fourier series Representation

The trigonometric Fourier series representation of a periodic signal $x(t)$ with fundamental period $T0$is given by:

$$x(t) = a_0 + \sum_{k=1}^{\infty}(a_k coskw_0 t + b_k sinkw_0 t)$$

Where ak, bk are Fourier coefficients given by:

$$a_k = \frac{2}{T_0}\int_{T_0} x(t)coskw_0 t\, dt$$

$$b_k = \frac{2}{T_0}\int_{T_0} x(t)sinkw_0 t\, dt$$

Properties of Fourier series

The Fourier series coefficients derived through the analysis equation determine completely a periodic signal $x(t)$. Hence, one can say that the complex exponential coefficients ak and the signal $x(t)$ are Fourier series pair. This relationship is denoted by $x(t) \leftrightarrow ak$. Some properties of Fourier series are presented in this section and MATLAB programmers verify them.

Suppose that the complex exponential Fourier series coefficients of the periodic signal $x(t)$ and $y(t)$ are denoted by a_k and b_k respectively, or in other words $x(t)\leftrightarrow a_k$ and $y(t)\leftrightarrow b_k$. Moreover let z_1 and z_2 denote two complex numbers. Then,

$$z_1 x(t) + z_2 y(t) \leftrightarrow z_1 a_k + z_2 b_k$$

Time Shifting

A shift in time of the periodic signal results on a phase change of the Fourier series coefficients. So if

$$x(t) \leftrightarrow a_k$$

The exact relationship is:

$$x\,(t - t_1) \leftrightarrow e^{-jk\omega_0 t_1}.\ a_k$$

Time Reversal

This property states that if $x(t) \leftrightarrow a_k$then $x(-t) \leftrightarrow a_{-k}$.

Signal Multiplication

The Fourier series coefficients of the product of two signals equal the convolution of the Fourier series coefficients of each signal. Suppose that $x(t) \leftrightarrow ak$ and $y(t) \leftrightarrow bk$ then,

$$x(t)y(t) \leftrightarrow a_k * b_k$$

Where ‘ * ’ denotes (discrete time) convolution.

Parseval’s Identity

The Parseval’s Identity states that the average power of a periodic signal is $x(t)$ with period equals the sum of the squares of the complex exponential Fourier series coefficients.

Mathematically,

$$P_x = \frac{1}{T}\int_T |x(t)|^2 dt = \sum_{k=-\infty}^{\infty} |a_k|^2$$

Pre Lab Tasks

1. The periodic signal $x(t)$ is defined in one period as $x(t) = t\,e^{-t},\ 0 \le t \le 6$. Plot the approximate signals in 4 periods in time using 81 terms of the complex exponential and trigonometric forms of Fourier series. For comparison reasons, plot the original signal $x(t)$ over the same time interval.
2. The periodic signal $x(t)$ in one period is given by:

$$x(t) = \begin{cases} 1, & 0 \le t \le 1 \\ 0 & 1 \le t \le 2 \end{cases}$$

Plot in one period the approximate signals using 41 and 201 terms of the complex exponential Fourier series. Furthermore, each time plot the complex exponential coefficients.

Lab Tasks

Lab Task

Using the complex exponential Fourier series representation, calculate the Fourier Series Coefficients of the periodic signal $x(t)$ shown below. Approximate $x(t)$ by using 5, 11, 21 and 61 terms of FS coefficients a_k, and compare the results. $x(t)$ In one period is given by:

$$x(t) = \begin{cases} 1, & 0 \le t \le 1 \\ 0 & 1 \le t \le 2 \end{cases}$$

Lab Task

The line spectra of the above example are computed below. The line spectra are plotted for $-5 \le k \le 5$ and $-40 \le k \le 40$.

Lab Task

We repeat the previous example for trigonometric Fourier series representation and compute the Fourier series coefficients $a_0,\ b_k,\ and\ c_k$ for various numbers of trigonometric terms used.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:________________________________Date: ______________________

LAB # 7: To Reproduce the Continuous Time Fourier Transform (CTFT) Using MATLAB Functions

Objectives

After completing this lab, the student will be able to:

- ✓ Trace the output response of CTFT in MATLAB.

Pre Lab

Introduction to Fourier Transform

Mathematical Definition

As stated that, the Fourier transform expresses a signal (or function) $x(t)$ in the (cyclic) frequency domain; that is, the signal is described by a function $X(\omega)$. The Fourier transform is denoted by the symbol $F\{.\}$; that is one can write

$$X(\omega) = F\{x(t)\}$$

In other words, the Fourier transform of a signal $x(t)$ is a signal $X(\omega)$. An alternative way of writing is

$$x(t) \xrightarrow{F} X(\omega)$$

The mathematical expression of Fourier transform is

$$X(\omega) = F\{x(t)\} = \int_{-\infty}^{\infty} x(t).e^{-j\omega t}dt$$

It is clear that $X(\omega)$ is a complex function of ω. In case of the Fourier transform of $x(t)$ has to be express in the frequency domain f, then substituting ω by $2\pi f$ yields

$$X(f) = F\{x(t)\} = \int_{-\infty}^{\infty} x(t).e^{-j2\pi ft}\,dt$$

In order to return from frequency domain to back time domain the *inverse Fourier transform* is applied. The inverse Fourier transform is denoted by the symbol $F^{-1}\{.\}$; that is, one can write

$$x(t) = F^{-1}\{X(\omega)\}$$

Or alternatively

$$X(\omega) \xrightarrow{F^{-1}} x(t)$$

The mathematical expression of the inverse Fourier transform is

$$x(t) = F^{-1}\{X(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega).e^{j\omega t}d\omega$$

Using f instead of ω

$$x(t) = F^{-1}\{X(f)\} = \int_{-\infty}^{\infty} X(f).e^{j2\pi ft}df$$

The cyclic frequency is measured in rad/s, while f is measured in Hertz. The Fourier transform of a signal is called (frequency) *spectrum.*

The Command fourier and ifourier

The computation of the integral is not always a trivial matter. However, in MATLAB® there is a possibility to compute is directly the Fourier transform $X(\omega)$ of a signal $x(t)$ by using command fourier. Correspondingly, the inverse Fourier transform is computed by using the command ifourier. Before executing these two commands, time t and frequency ω must be declared as symbolic variables. Symbolic variable is declared by using the command syms.

Fourier Transform Pairs

In this section, the most common Fourier transform pairs are represented. The computation of the Fourier and inverse Fourier transform require the computation of the integrals and sometime can be quite hard. This is the reason that already computed Fourier transform pairs are used to compute the Fourier or the inverse Fourier transform of a function. Thus, the computational procedure is to express a complicated function of interest in term of function with known Fourier (or inverse Fourier) transform and then based on the properties of Fourier transform to compute the Fourier transform of the complicated function. In the following table the common pairs are given below as:

Time Domain	Frequency Domain	Commands	Results/Comments
$x(t)$	$X(\omega)$	`syms t w w0 t0`	
$\delta(t)$	1	`x=dirac(t); fourier(x,w)`	`ans = 1`
1	$2\pi\delta(\omega)$	`fourier(1,w)`	`ans=2*pi*dirac(w)`
$u(t)$	$\frac{1}{j\omega}+\pi\delta(\omega)$	`X=1/(j*w)+pi*dirac(w); ifourier(X,t)`	`ans=(pi+pi*(2* heaviside (t)-1))(2*pi)`
$\delta(t-t_0)$	$e^{-j\omega t_0}$	`x=dirac(t-t0); fourier(x,w)`	`ans=1/exp(t0*w*i)`
$e^{j\omega_0 t}$	$2\pi\delta(\omega-\omega_0)$	`X=2*pi*dirac(w-w0); ifourier(X,t)`	`ans =exp(t*w0*i)`
$\cos(\omega_0 t)$	$\pi\delta(\omega-\omega_0)+\pi\delta(\omega+\omega_0)$	`X=pi*(dirac(w-w0)+dirac(w+w0)); ifourier(X,t)`	`ans=(1/exp(t*w0*i))/2 + exp(t*w0*i)/2`
$\sin(\omega_0 t)$	$(\pi/j)\delta(\omega-\omega_0)-(\pi/j)\delta(\omega-\omega_0)$	`X=(pi/j)* (dirac(w-w0)-dirac(w+w0)); x= ifourier(X,t)`	`ans=((1/exp(t*w0*i))*i)/2 - (exp(t*w0*i)*i)/2`
$e^{-at}u(t)$, $Re(a)>0$	$1/(j\omega+a)$	`a=8; x=exp(-a*t)*heaviside(t); X= fourier(x,w)`	`ans=1/(8 + w*i)`
$te^{-at}u(t)$, $Re(a)>0$	$1/(j\omega+a)^2$	`x=t*exp(-a*t)*heaviside(t); fourier(x,w)`	`ans =1/(8 + w*i)^2`
$\left(\frac{t^{n-1}}{(n-1)!}\right)e^{-at}u(t)$, $Re(a)>0$	$1/(j\omega+a)^n$	`n=4; X=1/(j*w+a)^n; ifourier(X,t)`	`ans=(t^3*heaviside(t))/(6*exp(8*t))`

Properties of Fourier Transform

In this section, we present the main properties of the Fourier transform. Each property is verified by an appropriate example.

Linearity

If $X_1(\omega) = F\{x_1(t)\}$ and $X_2(\omega) = F\{x_2(t)\}$ then for any scalar a_1 and a_2 the following property stands:

$$F\{a_1x_1(t) + a_2x_2(t)\} = a_1X_1(\omega) + a_2X_2(\omega)$$

Time Shifting

If $X(\omega) = F\{x(t)\}$ then for any t_0

$$F\{x(t - t_0)\} = e^{-j\omega t_0}X(\omega)$$

Frequency Shifting

If $X(\omega) = F\{x(t)\}$ then for any ω_0 ,

$$F\{e^{j\omega_0 t}x(t)\} = X(\omega - \omega_0)$$

Scaling in Time and Frequency

If $X(\omega) = F\{x(t)\}$ then for any scalar $b > 0$

$$F\{x(bt)\} = \frac{1}{|b|}X(\omega/b)$$

and

$$F\left\{\frac{1}{|b|}x\left(\frac{t}{b}\right)\right\} = X(b\omega)$$

The natural meaning of this property is that if the signal is expanded in time $(b < 1)$, then its spectrum is compressed to lower frequencies. On the other hand, a time compression of a signal $(b > 1)$ results in an expansion of the signal spectrum to higher frequencies.

Time Reversal

If $x(\omega) = F\{x(t)\}$, then

$$F\{x(-t)\} = X(-\Omega)$$

Differentiation in Time and Frequency

If $x(\omega) = F\{x(t)\}$, then

$$F\left\{\frac{dx(t)}{dt}\right\} = j\omega X(\omega)$$

Integration

If $x(\omega) = F\{x(t)\}$, then

$$F\left\{\int_{-\infty}^{t} x(\tau)\,d\tau\right\} = \frac{1}{j\omega}X(\omega) + \pi X(0)\delta(\omega)$$

Duality

If $x(\omega) = F\{x(t)\}$, then

$$F\{X(t)\} = 2\pi x(-\omega)$$

Pre Lab Task

1. Compute and plot the Fourier transform of the following signals:

   a. $x_1(t) = e^{5t}u(-t)$
   b. $x_2(t) = te^{-5t}u(t)$
   c. $x(t) = \begin{cases} 1 & 0 \leq t < 1 \\ 0 & 1 \leq t < 2 \\ 1 & 2 \leq t < 3 \end{cases}$

2. Compute and plot the convolution between the signals $x(t) = e^{-t^2}$ and $h(t) = e^{-t}u(t) + e^{t}u(-t)$.

Lab Tasks

Lab Task

Compute the Fourier transform of the function $x(t) = e^{-t^2}$

Lab Task

Compute the inverse Fourier transform of the function $X(\omega) = 1/(1 + j\omega)$.

Lab Task

Compute the Fourier transform (or spectra) of the signal $e^{-t}u(t)$ and compare with result of previous example.

Lab Task

Plot the Fourier transform of the continuous-time signal $x(t) = \frac{\sin(\pi t)}{\pi t} = sinc(t)$

Lab Task

Suppose that a signal $x(t)$ is given by $x(t) = te^{-3t}$. Compute the Fourier transform $X(\omega)$ of signal $x(t)$ and plot for $-20 \leq \omega \leq 20\ rad/sec$:

a) The magnitude of $X(\omega)$
b) The angle of $X(\omega)$
c) The real part of $X(\omega)$
d) The imaginary par of $X(\omega)$

Lab Task

Compute the convolution between the signals $h(t) = e^{-t}u(t)$ and $x(t) = e^{-t}\cos(2\pi t)u(t)$.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:_________________________________Date: ______________________

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:_________________________________Date: ______________________

LAB # 8: To Trace the Response of Continuous Time Signals Using Laplace Transform in MATLAB

Objectives

After completing this lab, the student will be able to:

- ✓ Explain the output of Laplace Transform using MATLAB.

Pre Lab

Laplace and Inverse Laplace Transform

Mathematical Definition

A continuous-time signal described in the time domain t is a signal given by a function $f(t)$. As previously mentioned, the Laplace transform expresses a signal in the complex frequency domain s (or s domain); that is, a signal is described by a function $F(s)$ Laplace transform is denoted by the symbol $L\{.\}$; that is, one can write

$$F(s) = L\{f(t)\}$$

In other words, the Laplace transform of a function $f(t)$ is a function $F(s)$. An alternative way of writing is

$$f(t) \xrightarrow{L} F(s)$$

There are two available forms of Laplace transform. The first is the two-sided (or bilateral) Laplace transform where the Laplace transforms F(s) of a function $f(t)$ is given by:

$$F(s) = L\{f(t)\} = \int_{-\infty}^{\infty} f(t)e^{-st}\, dt.$$

Variable s is a complex-valued number, and thus can be written as $s = a + j\omega$. This relationship reveals the association between the Laplace transform and the Fourier transform. More specially, substituting S by $j\omega$ yields the Fourier transform $X(\omega)$ of $x(t)$. Setting the lower limit of the integral to zero yields the one-sided (or unilateral) Laplace transform, which is described by the relationship

$$F(s) = L\{f(t)\} = \int_{0}^{\infty} f(t)e^{-st}\, dt.$$

The use of the one-sided Laplace transform is better suited for the purpose of this lab, as the signals considered are usually casual signals. In order to return from the s-domain back to the time domain, the inverse Laplace transform is applied. The inverse Laplace transform is denoted by the symbol $L^{-1}\{.\}$, that is, one can write

$$f(t) = L^{-1}\{F(s)\},$$

Or alternatively

$$F(s) \xrightarrow{L^{-1}} f(t)$$

The mathematically expression that describes the inverse Laplace transform of a function $F(s)$ is

$$f(t) = L^{-1}\{F(s)\} = \frac{1}{2\pi}\int_{\sigma-j\infty}^{\sigma+j\infty} F(s)e^{st}\, dt$$

Command Laplace and ilaplace

The computation of the integrals can sometimes be quite hard. This is the reason that already computed Laplace transform pairs are used to compute the Laplace transform and the inverse Laplace transform of a function (or signal). However, in MATLAB®, the Laplace transform $F(s)$ of a function $f(t)$ is easily computed by executing the command laplace. Moreover, the inverse Laplace transform of a function $F(s)$ is obtained by executing the command ilaplace. Before using these two commands, the declaration of complex frequency s and of time t as symbolic variables is necessary. Recall that in order to define a symbolic variable the command syms is used. Finally, note that the command laplace computes the unilateral Laplace transform of a function.

Functions $f(t) = e^{-t}$and $F(s) = 1/(1+s)$ are a Laplace transform pair. In other words, the (unilateral) Laplace transform of $f(t) = e^{-t}$ is $F(s) = 1/(1+s)$, while the inverse Laplace transform of $F(s) = 1/(1+s)$is $f(t) = e^{-t}$. A Laplace transform pair is denoted by $f(t) \to F(s)$. In our case, the Laplace transform pair is $e^{-t} \to 1/(1+s)$. In section 10.4, the most common Laplace transform pairs are presented. At the moment, let us introduce alternative syntaxes of the laplace and ilaplace commands. The most effective and in the same time simple syntax of the command laplace is ilaplace(f,s). In this way, the Laplace transform of the signal $f(t)$ is optimal and makes possible the computation of the Laplace transform in every case. Such a case is when the Laplace transform of a constant function has to be computed.

Finally an alternative available syntax for the Laplace transform computation is laplace(f,t,s);that is, the function f is transferred from t to s, while the inverse Laplace transform is computed using the command ilaplace(F,s,t); that is, the function F is transferred from s to t.

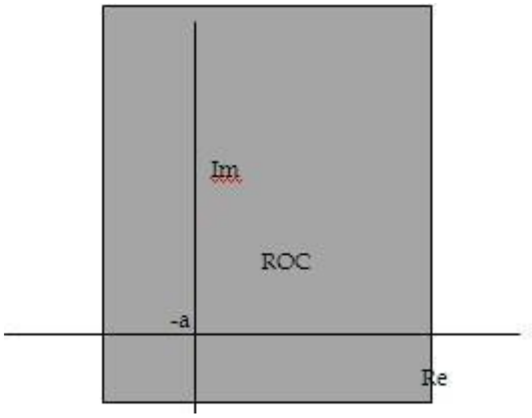
Region of Convergence

The Laplace transform of a function does not always exists as the integral does not always converge. Consider, for example, the signal $x(t) = e^{-at}$u(t),$a \in R$. The Laplace transform $X(s)$ of $x(t)$ is given by

$$X(s) = L\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-st}\,dt == \int_{-\infty}^{\infty} e^{-at}u(t)e^{-st}\,dt = \int_{0}^{\infty} e^{-(a+s)t}\,dt = \frac{1}{s+a}\left(e^{-(a+s)t}|_{t\to\infty} - 1\right).$$

Recall that s is a complex variable, hence the term $e^{-(a+s)t}|_{t\to\infty}$ does not become infinite as long as $Re\{s+a\}> 0$. Hence the Laplace transform of$x(t) = e^{-at}u(t), a \in R$ converges as $Re\{s\}> -a$. The area that corresponds to the values s such that $Re\{s\}> -a$ is called region of convergence ROC. The ROC of the Laplace transform of $x(t) = e^{-at}u(t)$ is depicted in figure 10.1.

Hence, for $\{s\}> -a$, the Laplace transform of $x(t) = e^{-at}u(t), a \in R$ is $X(s) = \frac{1}{s+a}$. In this lab, due to the fact that we focus on the MATLAB implementation, referring to the ROC of the Laplace transform of a signal is omitted for simplicity. However, for clarity reasons the ROC of the Laplace transform of a signal must be specified. For example, the Laplace transform of $x(t) = e^{-at}u(t), a \in R$ is $X(s) = \frac{1}{s+a}$ with ROC $Re\{s\}> -a$. In other words, the signal $x_1(t) = e^{-at}u(t), a \in R$ and $x_2(t) = -e^{-at}u(-t), a \in R$ have the same Laplace transform but with different ROC.



Region of convergence of the Laplace transformof $x(t) = e^{-at}u(t)$

Laplace Transform Pairs

In this section, the most common Laplace transform pairs are illustrated. The computation of Laplace transform or the inverse Laplace transform requires the computation of the integrals and it can be quite hard. This is the reason that already computed Laplace transform pairs are used to compute the Laplace transform or the inverse Laplace transform of a function. Thus, the computational procedure is to express a complicated function of interest in terms of function with known Laplace(or inverse Laplace) transform, and then based on the properties of Laplace transform to compute the Laplace (or inverse Laplace) transform of the complicated function. In the table below, the most common Laplace transform pairs are given. The illustrated pairs are confirmed by using the commands laplace and ilplace. Recall that command laplace computes the unilateral Laplace transform ($t \geq 0$). Thus, the unit step function (implemented in MATLAB with the command heaviside) can be omitted.

Time Domain	s-Domain	Commands	Results/Comments
$x(t)$	$X(s)$	`syms s t a w`	-
$\delta(t)$	1	`laplace(dirac(t),s)`	`ans=1`
$u(t)$	$1/s$	`laplace(heaviside(t),s)`	`ans=1/s`

$\frac{d^n\delta(t)}{dt^n}$	s^n	`n=4`
		`x=diff(dirac(t),n);`
		`X= laplace(x,s)`
			`X=s^4`

$e^{-at}u(t)$	$1/(s+a)$	`x=exp(-a*t)`
		`*heaviside(t);`
		`laplace(x,s)`
			`ans=1/(s+a)`

$e^{+at}u(t)$	$1/(s-a)$	`ilaplace(1/(s-a),t)`	`ans=exp(a*t)`

$e^{jw_o t}u(t)$	$1/(s-jw_o)$	`laplace(exp(j*w*t),s)`	`ans=1/(s-i*w)`
$\cos(w_o t)\,u(t)$	$s/(s^2+w_o{}^2)$	`X= s/(s^2+w^2);` `x=ilaplace(X)`	`x=cos(w*t)`
$\sin(w_o t)\,u(t)$	$\frac{w_o}{(s^2+w_o{}^2)}$	`x=sin(w*t);` `X=laplace(x)`	`x=w/(s^2+w^2)`
$e^{-at}\cos(w_o t)\,u(t)$	$(s+a)/((s+a)^2+w_o{}^2)$	`x=exp(-a*t) *cos(w*t);` `laplace(x,s)`	`ans=(s+a)/((s+a)^2+w^2)`
$e^{-at}\sin(w_o t)\,u(t)$	$w_o/((s+a)^2+w_o{}^2)$	`X=w/((s+a)^2+w^2);` `x= ilaplace(X,t)`	`x=exp(-a*t)*sin(w*t)`
$t^n u(t)$	$n!/s^{n+1}$	`x=(t^5)*heaviside(t);` `laplace(x,t,s)`	`ans=120/s^6%n=5`

$e^{-at}t^n u(t)$	$\frac{n!}{(s+a)^{n+1}}$	`X=factorial(10)`
		`/(s+a)^11;`
		`ilaplace(X,t)`
			`ans=t^10*exp(-a*t)%n=10`

$\text{t}\cos(w_o t)\,u(t)$	$(s^2-w_o{}^2)/\,(s^2+w_o{}^2)^2$	`x=t*cos(w*t);` `laplace(x,t,s)`	`ans=1/(s^2+w^2)^2*(s^2-` `w^2)`
$(t^{n-1}/(n-1))\,e^{-at}u(t)$	$1/(s+a)^n$	`X=1/(s+a)^6;` `x= ilaplace(X,t)`	`x=1/120*t^5*exp(-a*t)%n=6`

Inverse Laplace Transform through Partial Fraction

Partial Fraction of a Rational Function

In the usual case, the Laplace Transform of a function is expressed as a rational function of s. that is, is given as a ratio of two polynomials of sThe mathematical expression is

$$X(s) = \frac{B(s)}{A(S)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_o}{a_n s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_o}$$

Where a_i and b_i are real scalars. First, we discuss the case $m < n$; that is, the case that the degree of polynomial $B(s)$ of the numerator is lower than the degree of the polynomial $A(s)$ of the denominator. Suppose that λ_i are the roots of $A(s)$. The following case is considered:

The roots are λ_i *distinct*; that is, every root appears only once. In this case, $A(s)$ is written in the factored form $A(s) = a_n \prod_{i=1}^{n} s - \lambda_i$, while $X(s)$ is written as

$$X(s) = \frac{c_1}{s-\lambda_1} + \frac{c_2}{s-\lambda_2} + \cdots + \frac{c_n}{s-\lambda_n}$$

The Command residue

We introduced a procedure that has to be followed in order to express a function from rational form to partial fraction form. Alternatively, this procedure can be implemented directly in MATALB by using the command residue. The syntax is $[R, P, K] = residue(B, A)$, where B is a vector containing the coefficients of the numerator polynomial and A is the vector of the coefficients of the denominator polynomial. Suppose that a signal $X(s)$ is written in the rational form.

$$X(s) = \frac{B(s)}{A(S)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_{1s} + b_o}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_{1s} + a_o}$$

By defining the vectors $B = [\, b_m, \dots, b_0]$ and $A = [\, a_m, \dots, a_0]$, and executing the command $[R, P, K] = residue(B, A)$, the signal $X(s)$ can be written in partial fraction form as

$$X(s) = \frac{r_1}{s-p_1} + \frac{r_2}{s-p_2} + \cdots + \frac{r_n}{s-p_n} + K,$$

Where

$R = [\, r_1, r_2, \dots, r_n]$

$P = [p_1, p_2, \dots, p_n]$

K is the residue of the division of the two polynomials. K is null when $m < n$.

Pre Lab Task

1. Compute the Laplace Transform of the signals:
   a. $x(t) = e^{-3t} + e^{3t}$
   b. $x(t) = 3\cos(2t) + \sin(3\pi t)$
   c. $x(t) = \delta(t) + \delta(t-1)$
2. Compute the inverse Laplace transform of the signal:

$$X(s) = 1 + \frac{1}{s} + \frac{1}{s^2} + \frac{1}{s^3} + \frac{1}{s^4} + \frac{1}{s^5}$$

3. Express in partial fraction form the signal:

$$X(s) = \frac{s^3 - 2s^2 - 5s + 6}{s^2 - s + 2}$$

Lab Tasks

Lab Task

Compute the Laplace transform of the function $f(t) = -1.25 + 3.5te^{-2t} + 1.25e^{-2t}$. Confirm your answer by evaluating the inverse Laplace transform of the result.

Lab Task

Compute the inverse Laplace transform of the function $Y(s) = \frac{1}{s} + \frac{2}{(s+4)} + \frac{1}{(s+5)}$.

Lab Task

Express in the partial fraction form of the signal

$$X(s) = \frac{s^2 + 5s + 4}{s^4 + 1}$$

Verify your result by computing the inverse Laplace transform from both forms (partial fraction and rational)

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:__________________________________Date: _______________________

LAB # 09: To Reproduce the Properties of Laplace Transform Using MATLAB Functions

Objectives

After completing this lab, the student will be able to:

- ✓ Follow the steps involved in verification of properties of Laplace Transform in MATLAB.

Pre Lab

Laplace Transform Properties and Theorems

In this section, we present the main properties of the Laplace transform. Each property is verified by an appropriate example

Linearity

Let $X_1(s) = L\{x_1(t)\}$ and $X_2(s) = L\{x_2(t)\}$. Then for any scalars a_1, a_2 the following property stands

$$L\{a_1x_1(t) + a_2x_2(t)\} = a_1X_1(s) + a_2X_2(s)$$

```
syms t s
x1=exp(-t);
x2=cos(t);
a1=3;
a2=4;
Le=a1*x1+a2*x2;
Left=laplace(Le,s)
X1=laplace(x1);
X2=laplace(x2);
Right=a1*X1+a2*X2
```

```
Left =
3/(s + 1) + (4*s)/(s^2 + 1)

Right =
3/(s + 1) + (4*s)/(s^2 + 1)
```

Time Shifting

If $X(s) = L\{x(t)u(t)\}$, then for any $t_0 > 0$

$$X(s) = L\{x(t - t_o)u(t - t_o)\} = e^{-st_o}X(s).$$

```
syms t s
t0=2;
%x(t)=cos(t)u(t)
Le=cos(t-t0)*heaviside(t-t0);
Left=simplify(laplace(Le))

X=laplace(cos(t),s);
Right=exp(-s*t0)*X

X=laplace(cos(t),s);
Right=exp(-s*t0)*X
```

```
Left =

(s*exp(-2*s))/(s^2 + 1)


Right =

(s*exp(-2*s))/(s^2 + 1))
```

Complex Frequency Shifting

If $X(s) = L\{x(t)u(t)\}$ then for any s_0,

$$L\{x(t)e^{s_o t}\} = X(s - s_o).$$

```
syms t s
x=t.^3;
s0=2;
f=x*exp(s0*t);
L=laplace(f,s)

X=laplace(x,s);
R=subs(X,s,s-2)
```

```
L =

6/(s - 2)^4


R =

6/(s - 2)^4
```

Time Scaling

If $X(s) = L\{x(t)u(t)\}$ then for any $b > 0$

$$L\{x(bt)\} = \frac{1}{b}X\left(\frac{s}{b}\right).$$

```
syms t s b

le=exp(-b*2*t);

L=laplace(le,s)

x=exp(-2*t);

X=laplace(x,s);

R=simplify((1/b)*subs(X,s,s/b))
```

```
L =

1/(2*b + s)


R =

1/(2*b + s)
```

Differentiation in s-domain

If $X(s) = L\{x(t)\}$ and $n = 1,2, \ldots$ then

$$L\{-t^n x(t)\} = \frac{d^n X(s)}{ds^n}.$$

```
syms t s
%n=5

x=exp(t);
f=((-t)^5)*x;
L=laplace(f,s)

x=laplace(x,s);
R=diff(X,5)
```

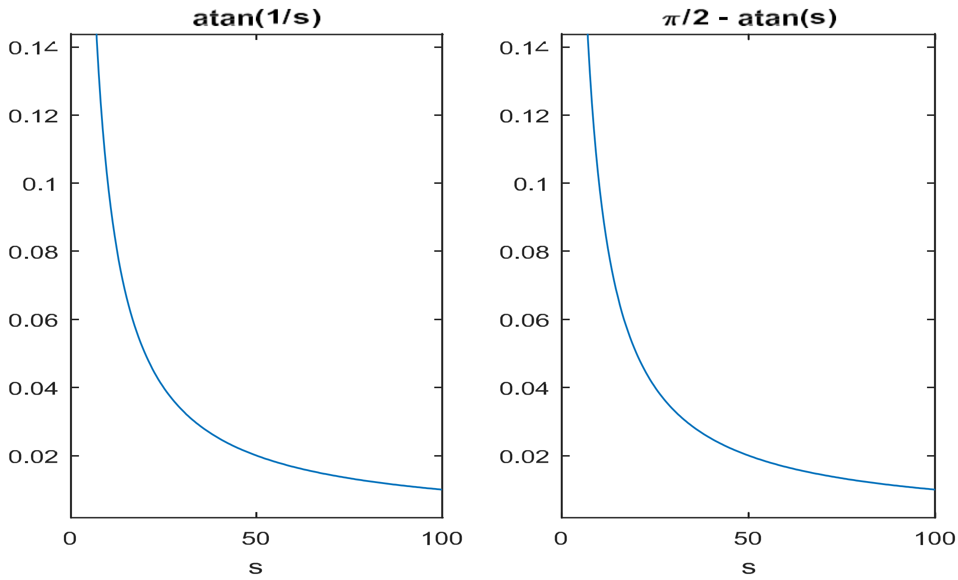
```
L =

-120/(s - 1)^6


R =

-120/(s + 2)^6
```

Integration in the Complex Frequency

If $X(s) = L\{x(t)\}$ then

$$L\left\{\frac{x(t)}{t}\right\} = \int_s^{\infty} X(u)du$$

```
syms s t u
x=sin(t);
L=laplace(x/t,s);
subplot(121)
ezplot(L,[0 100])

X=laplace(x,u);
R=int(X,u,s,inf);
subplot(122)
ezplot(R,[0 100])
```



Pre Lab Task

1. Compute the convolution between the signals $X_1(s) = \frac{1}{(s+1)^2}$ and $X_2(s) = \frac{2}{(s+1)^3}$.

Lab Tasks

Lab Task

Compute the convolution between the signals $x(t) = 5e^{-t}u(t)$ and $h(t) = te^{-t}u(t)$.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:__________________________________Date: _______________________

LAB # 10: To Trace the Response of Discrete Time Signals using Z-transform in MATLAB

Objectives

After completing this lab, the student will be able to:

- ✓ Manipulate and measure the output of Z-Transform using MATLAB.

Pre Lab

Z-Transform and Inverse Z-Transform

Mathematical Definition

As Laplace transformation is a more general transform as compared to Fourier Transform for continuous-time signals, z-Transform is a more general transform than discrete-time Fourier Transform when dealing with discrete-time signal. A discrete-time signal is defined in the discrete time domain n; that is, it is given by a function $f[n], n \in Z$. Z-Transform is denoted by the symbol $Z\{.\}$ and expresses a signal in z-domain, i.e., the signal is given by a function $F(z)$. The mathematical expression is

$$F(z) = Z\{f[n]\}$$

In other words, the z-transform of a function $f[n]$ is $F(z)$. The mathematical expression of the two sided (or-bilateral) z-transform is

$$F(z) = Z\{f[n]\} = \sum_{n=-\infty}^{\infty} f[n]z^{-n}$$

Where z is a complex variable. Setting the lower limit of the sum from minus infinity to zero yields the one-sided (or unilateral) z-transform whose mathematical expression is

$$F(z) = Z\{f[n]\} = \sum_{n=-\infty}^{\infty} f[n]z^{-n}$$

In order to return from the z-domain back to the discrete-time domain, the inverse z-transform is applied. The inverse z-transform is denoted by the symbol $Z^{-1}\{.\}$; that is, one can write

$$f[n] = Z^{-1}\{F(z)\}$$

The mathematical expression of the inverse z-transform is

$$x[n] = \frac{1}{2\pi j} \oint X(z)z^{n-1}\, dz.$$

The z-transform of a sequence is easily computed.

Commands ztrans and iztrans

In MATLAB, the z-transform $F(z)$ of a sequence $f[n]$ is computed easily by using the command ztrans. Moreover, the inverse z-transform of a function $F(z)$ is computed by using the command iztrans. Before using these two commands, the declaration of the complex variable z and of the discrete time n as symbolic variables is necessary. Recall that in order to define a symbolic variable, the command syms is used. The commands ztrans and iztrans are used exactly in same way as the commands laplace and ilaplace are used to compute the Laplace and inverse Laplace transforms of a function. Finally, we note that the command ztrans computes the unilateral z-transform.

The function $f[n] = 2^n u[n]$ and $F(z) = \frac{z}{z-2}$ are a z-transform pair. In other words, the z-transform of earlier one is later one and inverse z-transform of later one is earlier one. A z-transform pair is denoted by $x[n] \leftrightarrow X(z)$. At the moment, we discuss alternative syntaxes of the commands z-trans and iztrans. The most effective, and in the same time simple, syntax of the command ztrans is ztrans(f,z). In this way, the z-transform of a sequence $f[n]$ is expressed with the second input argument (here is z) as the independent variable. Using this syntax is optimal and makes possible the computation of the z-transform in every case. Such a case is when z-transform of a constant function has to be computed.

Region of Convergence

The z-transform of a sequence does not always exist. The region of convergence (ROC) of the z-transform of a sequence $x[n]$ is the range of z for which the z-transform of $x[n]$ is not infinite. For example, the z-transform $X(z)$ of the sequence $x[n] = 0.8^n u[n]$ is computed as $X(z) = \sum_{n=0}^{\infty} 0.8^n z^{-n} = \sum_{n=0}^{\infty} (\frac{0.8}{z})^n$. Thus $X(z)$ converges; that is, $X(z) < \infty$ if $\left|\frac{0.8}{z}\right| < 1 \Rightarrow |z| > 0.8$.

Therefore, the ROC of the z-transform $X(z)$ of the signal $x[n] = 0.8^n u[n]$ is $|z| > 0.8$.

Z- Transform Pairs

In this section, the most common z-transform pairs are presented. In the general case, the z-transform or the inverse z-transform of a function cannot be easily computed directly. This is the reason that already computed z-transform pairs are used to compute the z-transform or the inverse z-transform of a complicated function. Thus, the computational procedure is to express a complicated function of interest in terms of functions with known z (or inverse z) transform and then based on the properties of z-transform to compute the z (or inverse z) transform of the complicated function. In the table below, the most common z-transform pairs are given. The illustrated pairs are confirmed by using the commands ztrans and iztrans. It should be noted that since the command ztrans computes the one sided z-transform ($n \geq 0$), the unit step sequence $u[n]$ can be omitted from the signal definition.

Discrete-Time Domain	z-Domain	Commands	Result
X[n]	X(z)	`syms n z a w`	
$\delta[n]$	1	`f=dirac(n);` `ztrans(f,z)`	`Ans=dirac(0)`
$u[n]$	z/(z-1)	`f=heaviside(n)` `ztrans(f,z)`	`ans=z/(z-1)`
$n.u[n]$	z/(z-1)^2	`ztrans(n,z)`	`ans=z/(z-1)^2`
$a^n u[n]$	z/(z-a)	`F=z/(z-a);` `f=iztrans(F,n)`	`f=a^n`

$na^n u[n]$	$\frac{az}{(z-a)^2}$	`f=n*a^n;` `ztrans(f,z)`	`ans=z*a/(-z+a)^2`
$cos(w0n)u[n]$	$z^2 - \frac{zcos(w0)}{z^2-2zcos(wo)+1}$	`f=cos(w*n)` `ztrans(f,z)`	ans=$z^2 - \frac{zcos(w0)}{z^2-2zcos(wo)+1}$
$sin(w0n)u[n]$	$\frac{zsin(wo)}{z^2-2zcos(w0)+1}$	`f=sin(w*n);` `ztrans(f,z)`	ans= $\frac{zsin(wo)}{z^2-2zcos(w0)+1}$
$a^n \cos(won)\, u[n]$	$z^2 - \frac{azcos(w0)}{z^2-2azcos(wo)+a^2}$	`f=(a^n)*cos(w*n)` `ztrans(f,z)` `simplify(ans)`	ans=$z^2 - \frac{azcos(w0)}{z^2-2azcos(wo)+a^2}$
$a^n sin(wo)u[n$	$\frac{azsin(wo)}{z^2-2azcos(wo)+a^2}$	`f=(a^n)*sin(w*n);` `ztrans(f,z)` `simplify(ans)`	ans= $\frac{azsin(wo)}{z^2-2azcos(wo)+a^2}$

Inverse Z-Transform through Partial Fraction

Partial Fraction Expansion of a Rational Function

The z-transform of a sequence is usually expressed as a rational function of z, i.e., it is written as a ratio of two polynomials of *z*. The mathematical expression is

$$X(z) = \frac{B(z)}{A(z)} = \left(\frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}\right.$$

Where a_i, b_i are real numbers. In order to express the function $X(z)$ in a partial fraction form, we follow a similar approach to that followed in the Laplace transform chapter.

A signal in rational form can be expressed in partial fraction form by using the command residue. The command residue is used to confirm the result of the previous example.

Pre Lab Tasks

1. Compute the z-transform of the signal $x[n] = u[n] + nu[n] + n^2u[n]$.
2. Compute the z-transform of the signals $x[n] = u[n]$ and $x[n] = u[n-1]$.
3. Express in partial fraction form the signal:

$$X(z) = \frac{12 - 38z^{-1} + 11z^{-2} + 3z^{-3} + 54z^{-4}}{1 - 5z^{-1} + 6z^{-2}}$$

Lab Tasks

Lab Task

Compute the z-transform of the sequence $f[n] = [-3,5,6,7,8], -2 \leq n \leq 2$.

Lab Task

Compute the z-transform of the discrete-time signal $x[n] = n^2u[n]$.

Lab Task

Confirm your result of task 2, by computing the inverse z-transform of your outcome.

Lab Task

Find the inverse z-transform of $X(z) = \frac{2z+3}{z^2+5z+6}$

a. When $X(z)$ is in the rational form.
b. When $X(z)$ is in partial fraction form.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:_________________________________Date: ______________________

LAB # 11: To reproduce the properties of Z-Transform in MATLAB

Objectives

After completing this lab, the student will be able to:

- ✓ Follow the steps involved in verification of properties of Z-Transform in MATLAB.

Pre Lab

Z-Transform Properties and Theorems

In this section, the main properties of the z-transform are introduced and verified through appropriate examples.

Linearity

If $X_1(z) = Z\{x_1[n]\}$ and $X_2(z) = Z\{x_2[n]\}$, then for any scalar a_1, a_2.

$$Z\{a_1x_1[n] + a_2x_2[n]\} = a_1X_1(z) + a_2X_2(z)$$

```
syms n z

x1=n^2;

x2=2^n;

a1=3;

a2=4;
Le=a1*x1+a2*x2;

Left=ztrans(Le,z)

X1=ztrans(x1,z);

X2=ztrans(x2,z);

Right=a1*X1+a2*X2
```

```
Left =

(4*z)/(z - 2) + (3*z*(z + 1))/(z - 1)^3


Right =

(4*z)/(z - 2) + (3*z*(z + 1))/(z - 1)^3
```

Right Shift

If $X(z) = Z\{x[n]\}$ and $x[n]$ is casual, i.e., $x[n] = 0, n < 0$, then for any positive integer m,

$$Z\{x[n-m]u[n-m]\} = z^{-m}X(z)$$

```
syms n z
m=2;
x1=3^(n-m)*heaviside(n-m);
Left=simplify(ztrans(x1,z))

x=3^n*heaviside(n);
X=ztrans(x,z);
Right=simplify((z^(-m))*X)
```

```
Left =

(3/(z - 3) + 1/2)/z^2


Right =

(3/(z - 3) + 1/2)/z^2
```

Right Shift

If $X(z) = Z\{x[n]\}$, then for any positive integer m,

$$Z\{x[n-1]\} = z^{-1}X(z) + x[-1]$$
$$Z\{x[n-2]\} = z^{-2}X(z) + x[-2] + z^{-1}x[-1]$$
$$\vdots$$
$$Z\{x[n-m]\} = z^{-m}X(z) + x[-m] + z^{-1}x[-m+1] + \cdots + z^{-m+1}x[-1].$$

```
% x[n] = 0.8n ,-3 ? n ? 3
n=-3:3;
x=0.8.^n;
xminus1=x(3);
xminus2=x(2);
xminus3=x(1);

syms z n
xn2=0.8^(n-2);
Left=simplify(ztrans(xn2,z))



x=0.8.^n;
X=ztrans(x,z);
Right=simplify((z^-2)*X+xminus2+(z^-1)*xminus1)
```

```
Left =

(25*z)/(16*(z - 4/5))


Right =

(125*z)/(16*(5*z - 4))
```

Left shift

If $X(z) = Z\{x[n]\}$, then for any positive integer m,

$$Z\{x[n+1]\} = zX(z) - x[0]z$$
$$Z\{x[n+2]\} = z^2X(z) - x[0]z^2 - x[1]z$$

.

$$z\{x[n+m]\} = z^mX(z) - x[0]z^m - x[1]z^{m-1} - \cdots - x[m-1]z.$$

```
x0=0.8^(0);
x1=0.8^(1);

syms n z
xn2=0.8^(n+2);
Left=simplify(ztrans(xn2,z))

x=0.8^n;
X=ztrans(x,z);
Right=simplify((z^2)*X-(z^2)*x0-x1*z)
```

```
Left =

(16*z)/(25*(z - 4/5))


Right =

(16*z)/(5*(5*z - 4))
```

Scaling in the z-Domain

If $X(z) = Z\{x[n]\}$ then for any scalar a,

$$Z\{a^{-n}x[n]\} = X(az)$$

And

$$Z\{a^n x[n]\} = X(\frac{z}{a})$$

An important special case is when $a = e^{j\omega 0}$, for which $Z\{e^{j\omega 0 n}x[n]\} = X(e^{-j\omega 0}z)$.

```
syms n z a w0
x=4^n;
X=ztrans(x,z);

%case 1
Right=subs(X,z,a*z)
L=a^(-n)*x;
Left=simplify(ztrans(L,z))

%case 2
Right=simplify(subs(x,z,z/a))
L=a^n*x;
Left=simplify(ztrans(L,z))

%case 3
Right=subs(X,z,exp(-j*w0)*z);
L=exp(j*w0*n)*x;
Left=simplify(ztrans(L,z));
error=simplify(Left-Right)
```

```
Right =

(a*z)/(a*z - 4)


Left =

(a*z)/(a*z - 4)


Right =

2^(2*n)


Left =

-z/(4*a - z)


error = 0
```

Time Reversal

If $X(z) = Z\{x[n]\}$ then

$$Z\{x[-n]\} = X(z^{-1})$$

```
syms z
x=[1 2 3 4];
n=[0 1 2 3];
X=sum(x.*(z.^-n));
Right=subs(X,z,z^-1)
nrev=[-3 -2 -1 0];
xrev=[4 3 2 1];
Left=sum(xrev.*(z.^nrev))
```

```
Right =

4*z^3 + 3*z^2 + 2*z + 1


Left =

2/z + 3/z^2 + 4/z^3 + 1
```

Differentiation in the z-domain

If $X(z) = Z\{x[n]\}$ then

$$Z\{n.\,x[n]\} = -z\,\frac{dX(z)}{dz}$$

```
syms n z
x=0.9^n;
Left=ztrans(n*x,z)

X=ztrans(x,z);
d=diff(X,z);
Right=simplify(-z*d)
```

```
Left =

(90*z)/(10*z - 9)^2

Right =

(90*z)/(10*z - 9)^2
```

Summation

If $X(z) = Z\{x[n]\}$ and $x[n]$ is a casual discrete-time signal, then

$$Z\left\{\sum_{i=0}^{n} x[i]\right\} = \frac{z}{z-1}X(z)$$

```
syms n z
x=n^2;
s=symsum(x,n,0,n);
Left=simplify(ztrans(s,z))
X=ztrans(x,z);
Right=(z/(z-1))*X
```

```
Left =

(z^2*(z + 1))/(z - 1)^4


Right =

(z^2*(z + 1))/(z - 1)^4
```

Convolution in the time domain

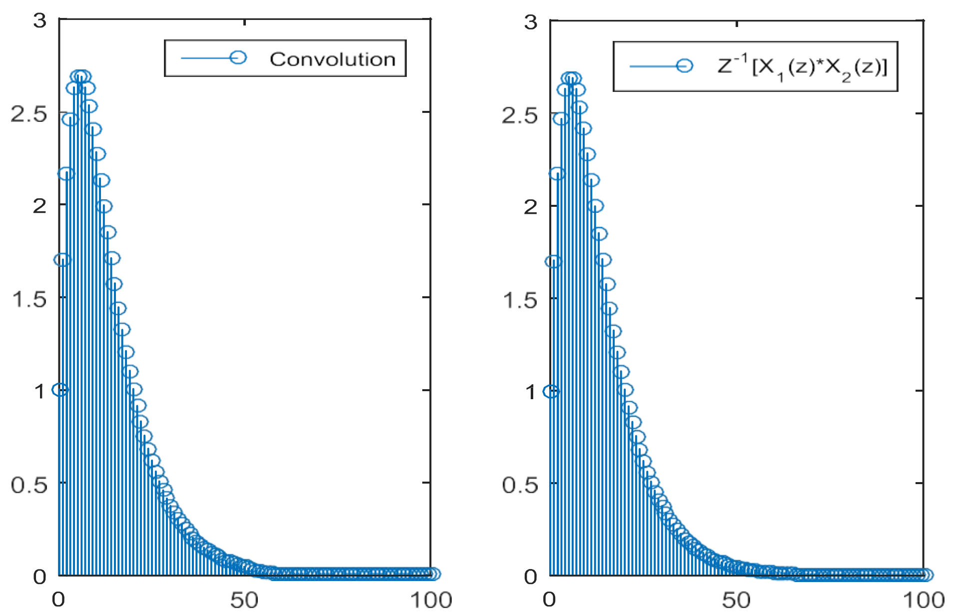
If $X_1(z) = Z\{x_1[n]\}$ and $X_2(z) = Z\{x_2[n]\}$, then

$$Z\{x_1[n] * x_2[n]\} = X_1(z)X_2(z)$$

Applying inverse z-transform to both sides yields

$$x_1[n] * x_2[n] = Z^{-1}\{X_1(z)X_2(z)\}$$

```
n=0:50;
x1=0.9.^n;
x2=0.8.^n;
y=conv(x1,x2);

subplot(121)
stem(0:100,y)
legend('Convolution');

syms n z
x1=0.9.^n;
x2=0.8.^n;

X1=ztrans(x1,z);
X2=ztrans(x2,z);

Right=iztrans(X1*X2);
n=0:100;
Right=subs(Right,n);

subplot(122)
stem(0:100,Right)
legend('Z^-^1[X_1(z)*X_2(z)]')
```



Lab Task:

Compute the convolution between the signals,

$$X_1(z) = \frac{z}{z-0.9} \text{ and } X_2(z) = \frac{z}{z+6}.$$

LAB # 12: To display the basic Signals & Systems Operations Using Simulink

Objectives

After completing this lab, the student will be able to:

- ✓ Describe the steps involved in performing the basic Signal operations in Simulink
- ✓ Describe the steps involved in performing the basic System operations in Simulink

Pre Lab

Introduction to Simulink

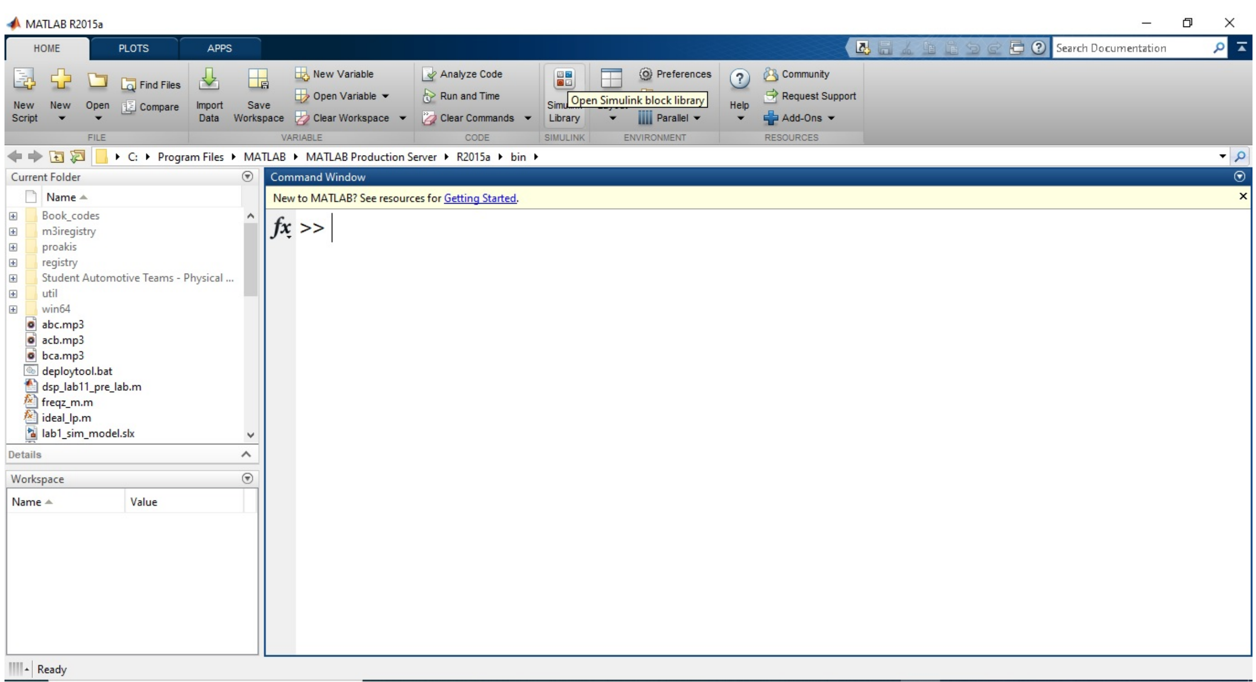
Simulink is a graphical extension to MATLAB for the modeling and simulation of systems. In Simulink, systems are drawn on screen as block diagrams. Many elements of block diagrams are available (such as transfer functions, summing junctions, etc.), as well as virtual input devices (such as function generators) and output devices (such as oscilloscopes). Simulink is integrated with MATLAB and data can be easily transferred between the programs. In this tutorial, we will introduce the basics of using Simulink to model and simulate a system.Simulink files are known as models and they have file extensions .mdl or slx.

Starting Simulink

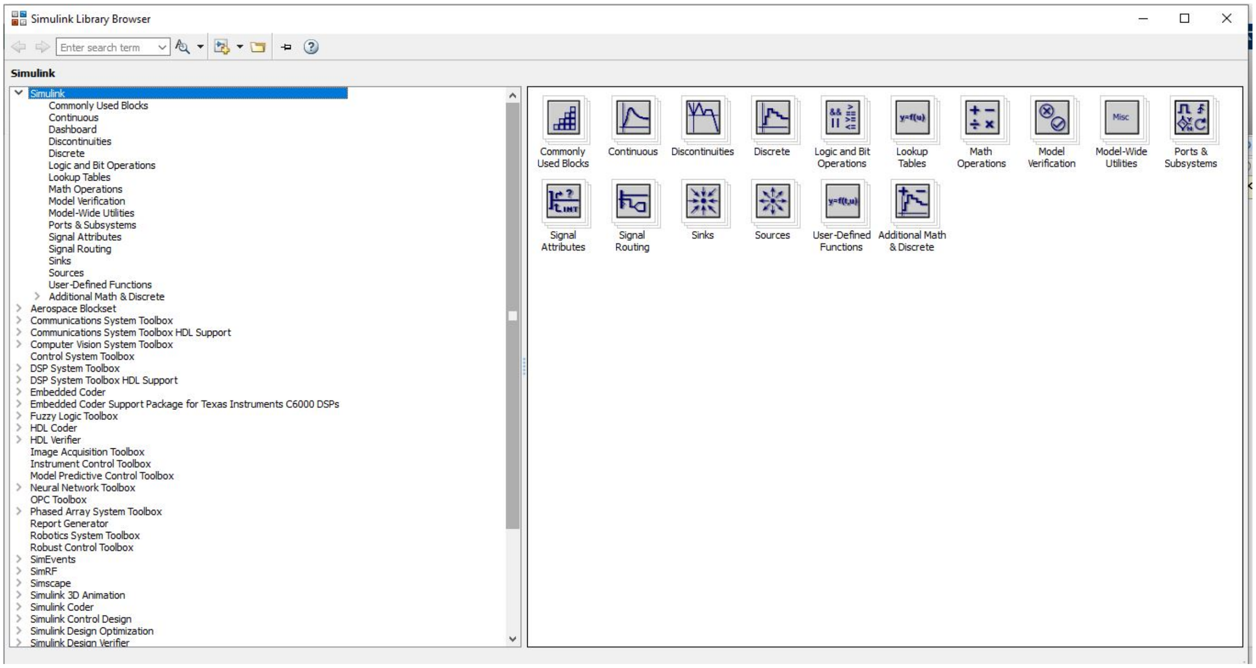
Simulink is started from the MATLAB command prompt by entering the following command:

```
>>simulink
```

Alternatively, you can click on the "Simulink Library Browser" button at the top of the MATLAB command window as shown below:

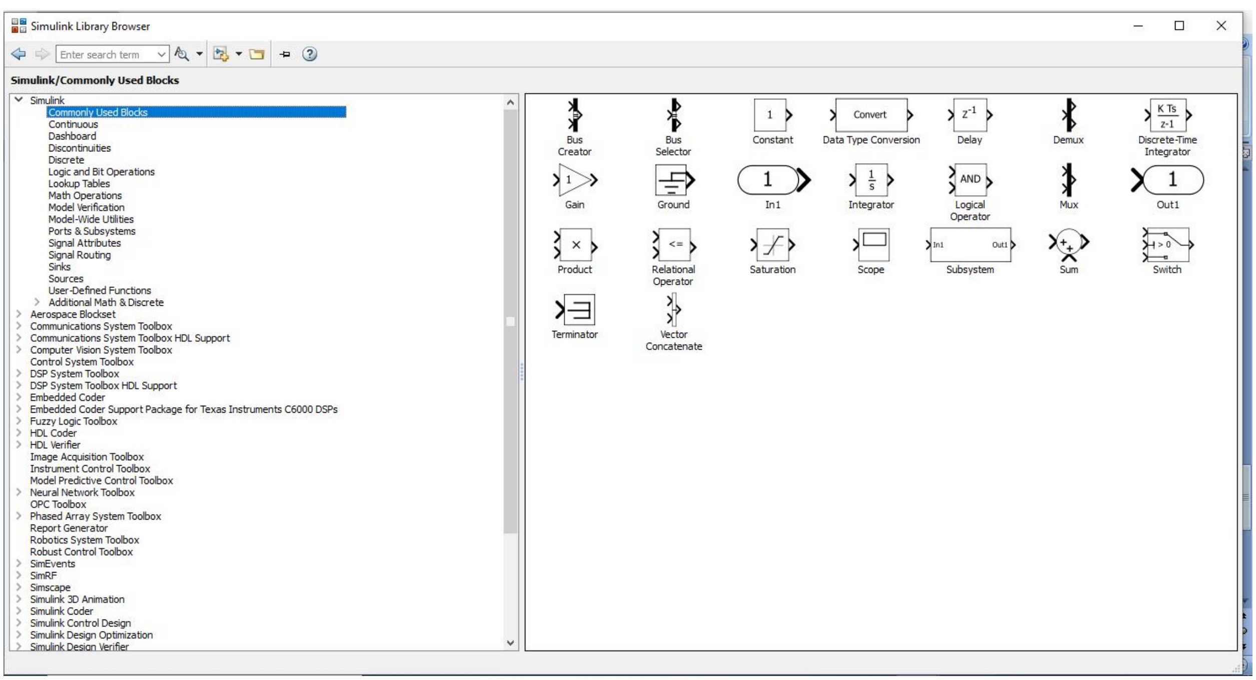


The Simulink Library Browser window should now appear on the screen. Most of the blocks needed for modeling basic systems can be found in the subfolders of the main "Simulink" folder (opened by clicking on the ">" button in front of "Simulink"). Once the "Simulink" folder has been opened, the Library Browser window should look like:

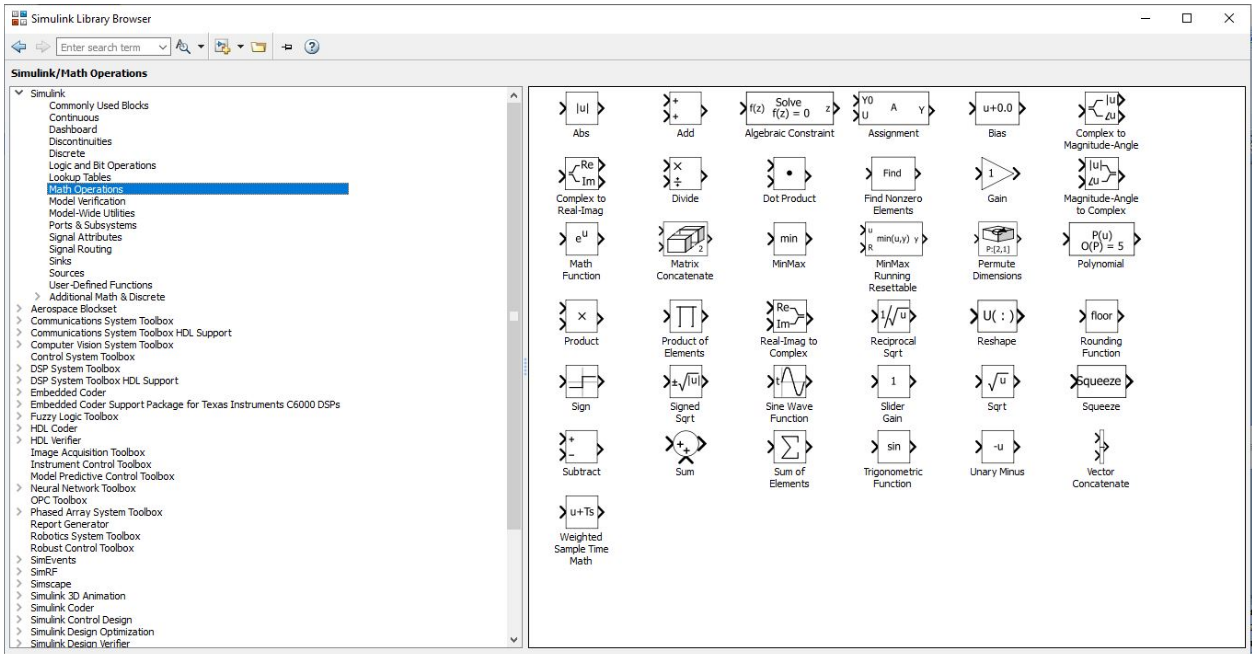


Some important libraries of “Simulink” are:

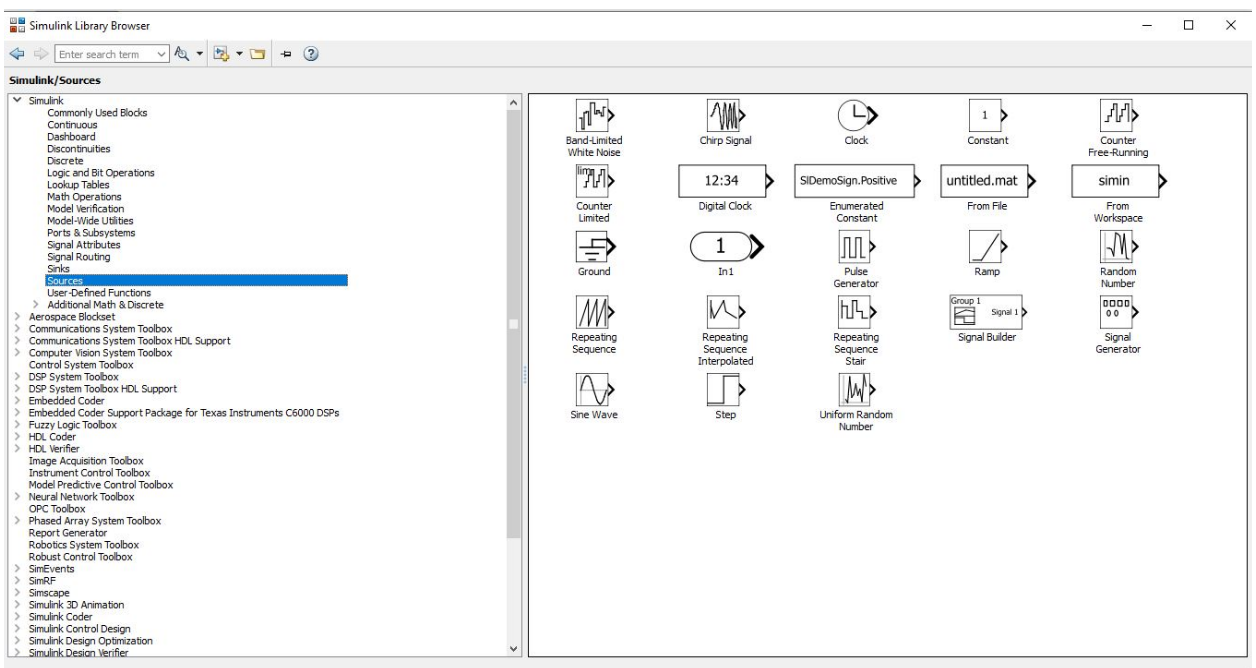
Commonly Used Blocks



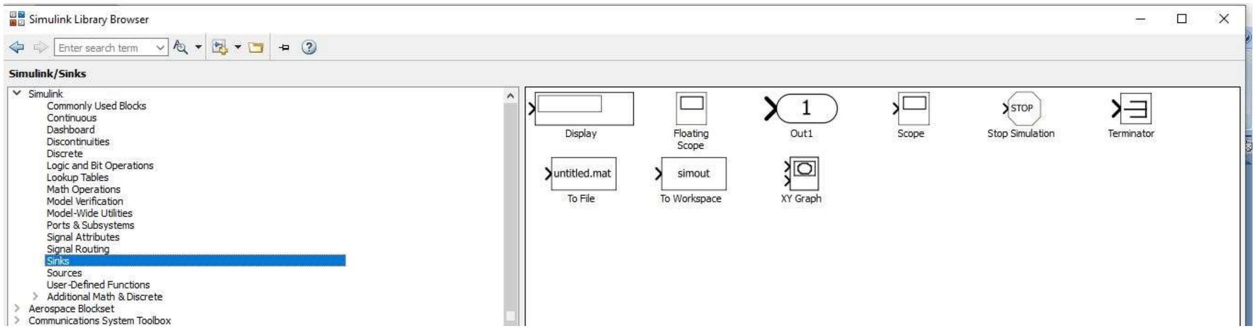
Math Operations:



Sources

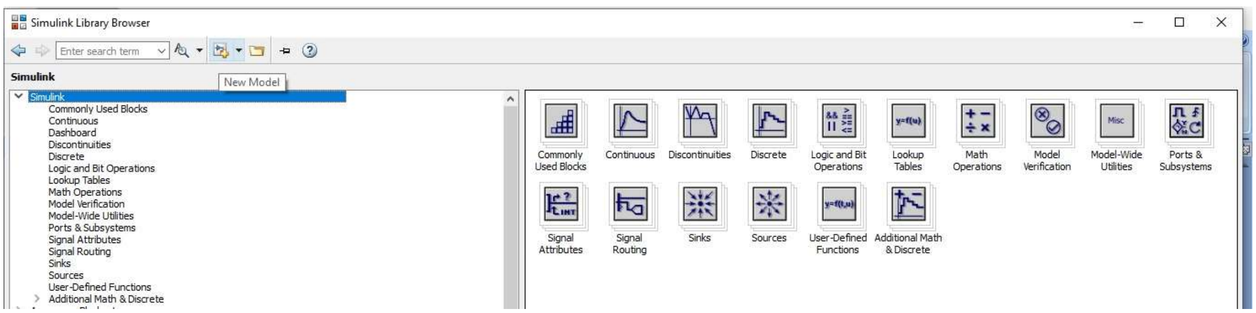


Sinks



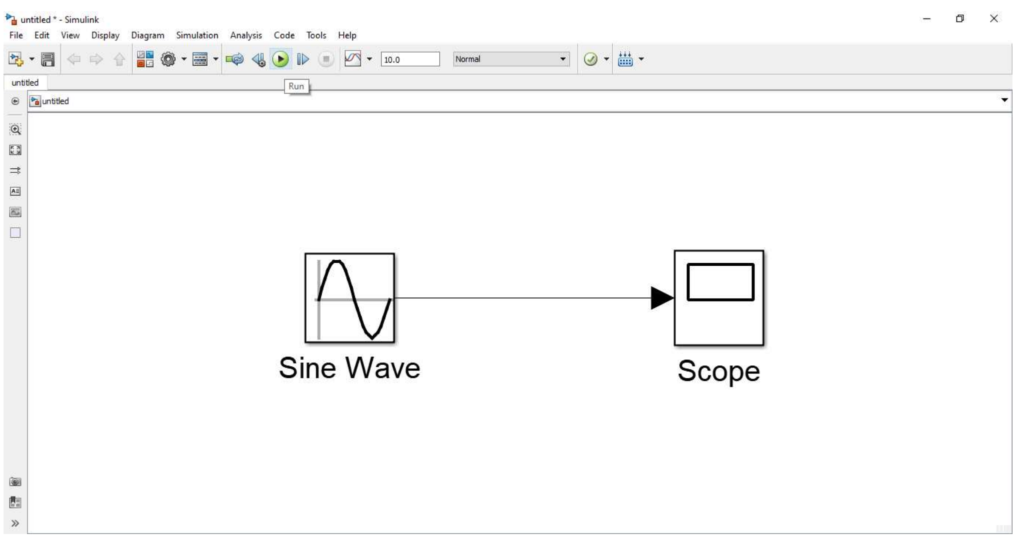
Creating a new Model:

In order to create a new model, click on “new model” button as shown below:

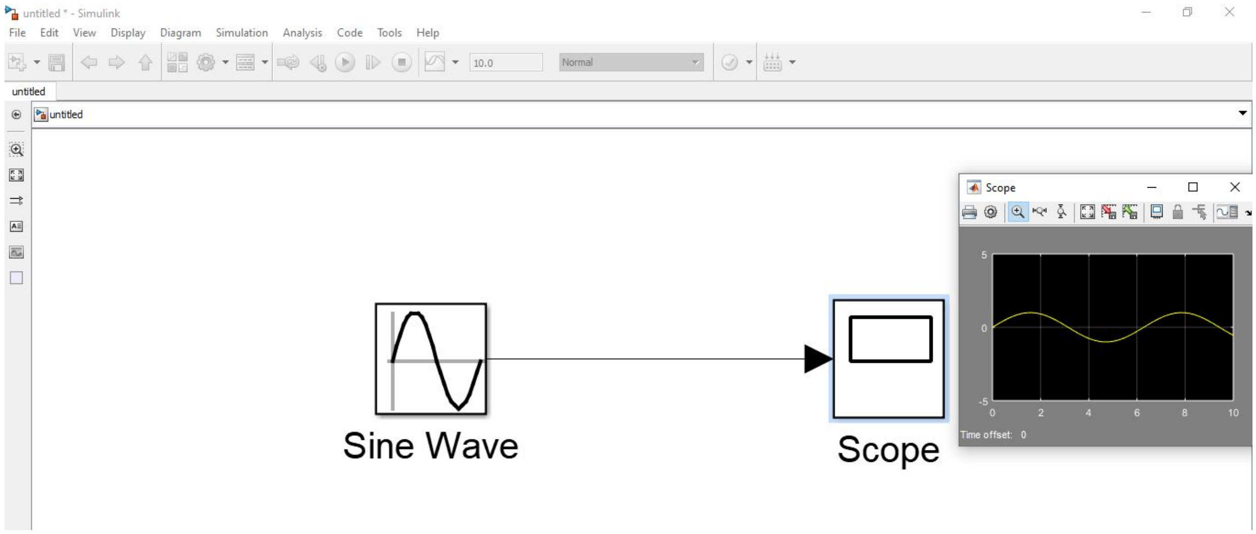


Creating a simple model:

Click the sources block and from there drag the sine wave block into the newly created model. Click the sinks block and from there drag the scope into the newly created model. Connect the sine wave block with the scope block and then press the green colour Run button at top as shown below:



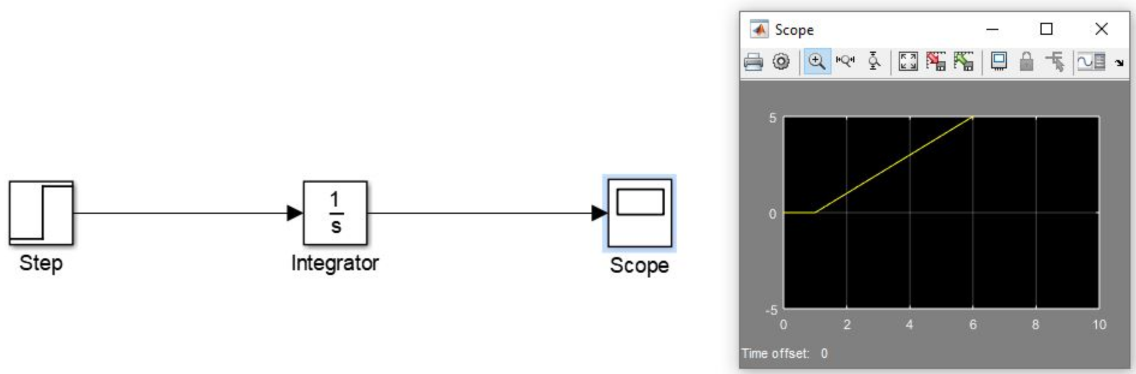
Double click on the scope to see the response, as shown below:



Similary we can use step signal or any other source instead of Sine wave.

Creating Model of an integrator system

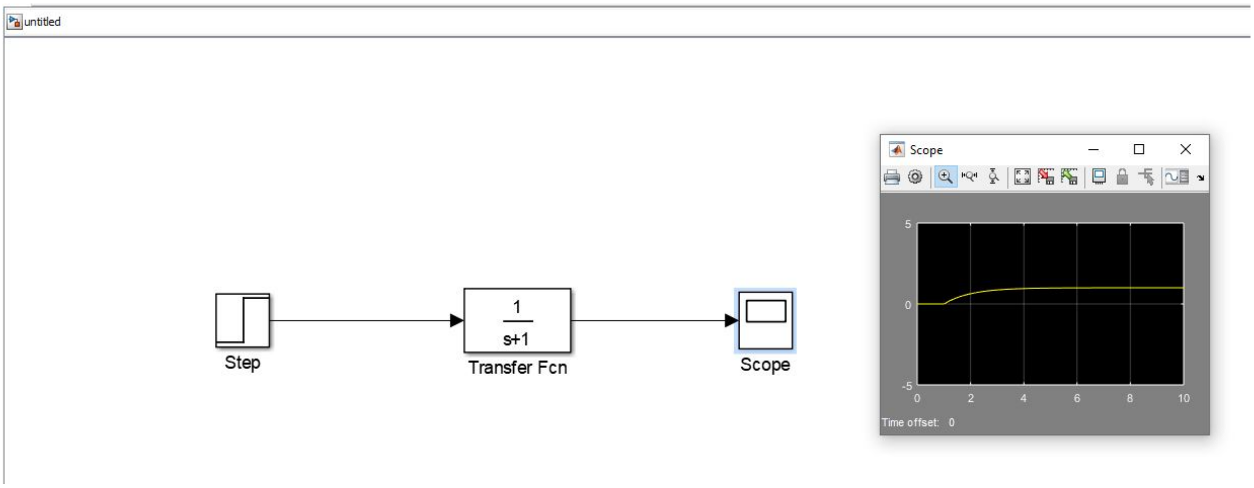
Create a new Model and then go to commonly used blocks . Click on integrator and add it to newly created model. Also add source and sink as discussed in last task. We will use step signal as source this time . We know that integral of Unit step is a ramp signal. We have also shown it in our model below:



Similarly we can also use signal generator as source . So we can also use square,sawtooth waves etc

Modelling of Linear systems

In simulink , we model linear systems by transfer function block as shown in model below where we are giving unit step input to a linear system and observing the response on scope. Transfer function blocks are available in "continuous" library of simulink



Lab Task 1: Modelling of low pass filter using Simulink

Lets consider the scenario where we have a low pass filter and its transfer function is given below:

$$H(s)=K\ \frac{1}{\tau s+1}$$

Where K is gain and it is equal to 1. 'τ' is time constant and it is equal to 1/f where f is cut-off frequency and it is equal to 30 rad/sec.

Model this low pass filter using simulink.

Lab Task 2: Modelling of High pass filter using Simulink

Lets consider the scenario where we have a high pass filter and its transfer function is given below:

$$H(s)=K\ \frac{s}{s+1/\tau}$$

Where K is gain and it is equal to 1. 'τ' is time constant and it is equal to 1/f where f is cut-off frequency and it is equal to 30 rad/sec.

Model this high pass filter using simulink.

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature:___________________________________Date: ________________________