Name: Maddula Naga Teja				**Branch: AI & DS**
Lab Section: Sec - 1				**Id No: 2000080061**
Pre-lab Session work (15M)	**In-Lab Session work (15M)**	**Post Lab session work (10M)**	**Viva (10M)**	**Total Marks 50M**
Remarks:				
Date:	**Signature of the Instructor**		**Marks awarded**	

Signal Processing-Lab-(20AD 3107)

Lab-1: Introduction to Matlab

Lab Report

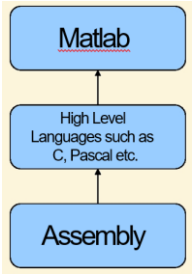
Introduction: MATLAB is an environment for performing calculations and simulations of a variety of types. MATLAB, short for matrix laboratory, was developed and founded MathWorks by Cleve Moler, a professor of mathematics and computer science. In the earliest version of MATLAB, there were about 80 commands, and they allowed for matrix calculations. Now there are thousands of commands, and there are many, many types of calculations that one can perform. During the course of this lab, the student will learn how to make calculations using MATLAB and will learn a little about simulating systems using the simulation tools provided by MATLAB.

Objectives:

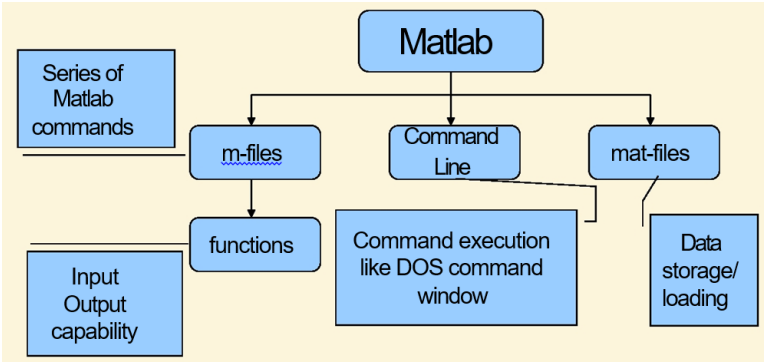
1. To familiar with the Matlab environment.
2. Write the simple Matlab code to perform certain operations.

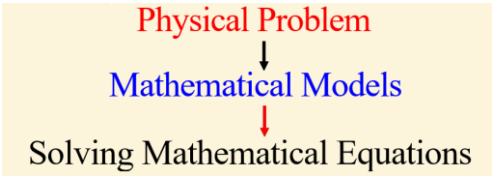
Requirements: Digital Computer with MATLAB software.

MATLAB (short for MATrix LABoratory) is a powerful computing environment that handles anything from simple arithmetic to advanced data analysis. At its core is the matrix as its basic data type. Combined with extensive maths and graphics functions, complicated calculations can be carried out by specifying only a few simple instructions. MATLAB can be used to do anything your scientific calculator can do, and more. The installations in the computer labs include tool boxes that include functions for electrical engineering related tasks, such as signal processing and system analysis. The installations in the computer labs include tool boxes that include functions for electrical engineering related tasks, such as signal processing and system analysis. You can plot your data in a multitude of visualizations as well. Matlab is basically a high level language which has many specialized toolboxes for making things easier for us



Matlab is too broad for our purposes in this course. The features we are going to require is





Computations can be carried out in one of three ways:

1. Directly from the command line,
2. by writing a script that carries out predefined instructions, or
3. by writing your own functions.

Writing your own functions is much like programming in other languages, except that you have the full resources of MATLAB's functions at your disposal, making for very compact code. The MATLAB-6 and above environment has several windows at your disposal.

Command Window: The main window is the command window, where you will type all your commands, for small programs.

1. **Definition of Vectors & Matrices:** MATLAB is based on matrix and vector algebra; even scalars are treated as 1x1 matrices. Therefore, vector and matrix operations are as simple as common calculator operations. Vectors can be defined in two ways.

Method1: The first method is used for arbitrary elements:

```
>> v = [1 3 5 7]; %creates a 1x4 vector with elements 1,3, 5 and 7.
```

* Note that commas could have been used in place of spaces to separate the elements. that is

```
>> v = [1,3,5,7];
```

• Additional elements can be added to the vector:

```
>> v(5) = 8
```
% yields the vector v = [1 3 5 7 8].

Previously defined vectors can be used to define a new vector. For example, with ' v ' defined above

```
>> a = [9 10];
>> b = [v a]
```
% creates the vector b = [1 3 5 7 8 9 10].

Method2: The second method is used for creating vectors with equally spaced elements:

```
t = 0: 0.1:10
```
% creates a 1x101 vector with the elements 0, .1, .2, .3,...,10. Note that the middle number defines the increment.

If only two numbers are given, then the increment is set to a default of 1: For example.

```
k = 0:10    % creates a 1x11 vector with the elements 0, 1, 2, ..., 10.
```

2. Matrices: Matrices are defined by entering the elements row by row

```
>> M = [1 2 4; 3 6 8; 2 6 5]   % creates a 3X3 matrix
>> M'  % Transpose of matrix  M
```

*The inverse of M is M^{-1} denoted in Matlab as M^-1

```
>> M^-1
>> M(2,3)          % displays the element of second row and third column
```

3. Arithmetic Operations: When applying addition, subtraction, multiplication and division between a scalar (that is a single number) and a vector we use +, -, *, and / respectively.

```
% Let
>> a = [1 2 3;4 5 6;7 8 9]
>> b = [9 8 7;6 5 4;3 2 1]
% Then
>> a + b  % addition of two matrices
>> a - b  % Subtraction of two matrices
>> a*b    %  Multiplication
>> a+2    %  addition of a constant to matrix 'a '
>> a .*b  % Element wise multiplication
```

+	addition
−	subtraction
*	multiplication
/	right division (a / b means (a÷b)
\	left division (a \ b means (b÷a)
^	exponent

4. Relations and logical operations:

==	equal	&	AND
~=	not equal	|	OR
<	less than	~	NOT
<=	less than or equal	pi	3.14159265…
>	greater than	j	imaginary unit, $\sqrt{-1}$
>=	greater than or equal	i	same as j

5. Special Matrices

```
M = [ ] ;          % null matrix:
M = zeros(n,m);    % n x m matrix of zeros:
M = ones(n,m);     % n x m matrix of ones:
M = eye(n);        % n x n identity matrix:
```

6. Help, Document and Demos : Matlab provides excellent tutorials that are accessible by typing >> demo

The Basic matrix operations tutorial under the Matrices tutorial, the Image Processing and Signal Processing tutorial under Toolboxes are highly recommended.

To get information on a particular function of Matlab, we type help function_name

```
>> help fft      % help for Fast Fourier Transform
>> help mean     % help for mean or average value
```

Comments in Matlab must be proceeded by % .

To define a function in Matlab, we first create a function with the name of the function. We then define the function in the file. For example, the Matlab documentation recommends stat.m written as below for calculating mean and standard deviation of a signal ' x '.

```
function [mean,stdev] = stat(x)
% This comment is printed out with help stat
n = length(x); % assumes that x is a vector.
mean = sum(x) / n; % sum adds up all elements.
stdev = sqrt(sum((x - mean).^2)/n);
```

7. Viewing Matlab Matrices as one Dimensional Arrays : Convert multi-dimensional to a one-dimensional array by simply using array_name (:) **.** For example as previously defined matrix a=[1 2 3;4 5 6;7 8 9]

```
>> a(:)    % results 147258369
```

Note that the elements in the one-dimensional array are arranged column by column, **not** row by row. This is the same convention that is used for two dimensional arrays in **Fortran**, and this will also be the convention that we will adopt for representing two dimensional arrays in C++.

8. Working with Memory in Matlab: To keep track of how memory is allocated, we use command 'whos'

```
b=[34 54 56];
a=[2 3 56];
whos
```

9. Control Structures:

If Statement Syntax	Some dummy examples

if (Condition_1)
Matlab Commands
elseif (Condition_2)
Matlab Commands
elseif (Condition_3)
Matlab Commands
else
Matlab Commands
end
	if ((a>3) & (b==5))
	Some Matlab commands
	end
	if (a<3)
	Some Matlab commands
	elseif (b~=5)
	Some Matlab commands
	end
	if (a<3)
	Some Matlab commands
	end

For loop syntax	Some dummy examples
for i=Index_Array Matlab Commands end	for j =1:3:200 Some Matlab commands end for m = 13: -0.2:-21 Some Matlab commands end for k =[0.1 0.3 -13 12 7] Some Matlab commands end
While Loop Syntax while (condition) Matlab Commands end	while ((a>3) & (b==5)) Some Matlab commands end

10. Generation and plotting of basic signals:

```
>> Plot([3 4 5 3 2 2])
```

To make a graph of y = sin(t) on the interval t = 0 to t = 10, we do the following:

```
>> t = 0:0.3:10;
>> y = sin(t);
>> plot(t,y) ;
```

Another example:

```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x,y); xlabel('x = 0:2\pi'); ylabel('Sine of x');
title('Plot of the Sine Function');
```

Multiple Graphs :

```
t = 0:pi/100:2*pi;
y1=sin(t);
y2=sin(t+pi/2); plot(t,y1,t,y2); grid on
```

Multiple Plots

```
t = 0:pi/100:2*pi;
```

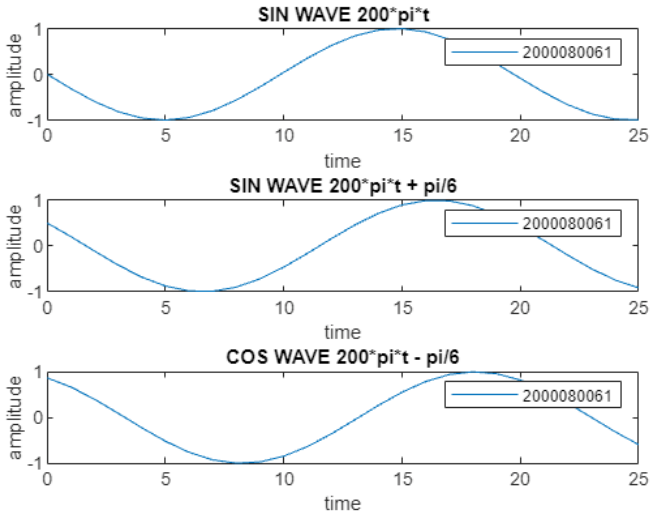
```
y1=sin(t);
y2=sin(t+pi/2);
subplot(2,2,1); plot(t,y1);
subplot(2,2,2); plot(t,y2);
```

Exercise

1) Develop Matlab codes, simulate, and plot the following signals.

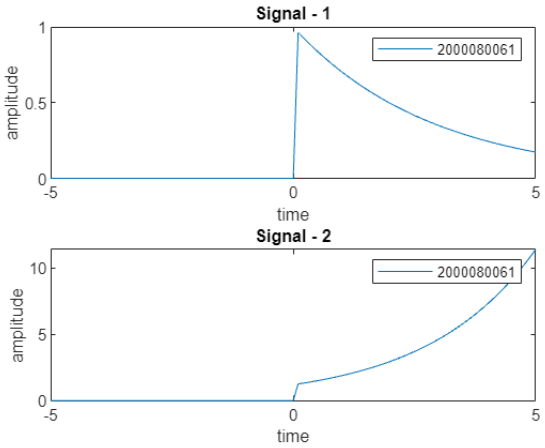
(a) $\sin(200\pi t)$ (b) $\sin(200\pi t + \frac{\pi}{6})$ (c) $\cos(200\pi t - \frac{\pi}{6})$

```
clc;
clear all;
close all;
t=0:1:25;
x=sin(200*3.14*t);
y=sin(200*3.14*t + 3.14/6);
z=cos(200*3.14*t - 3.14/6);

subplot(3,1,1);
plot(t,x);
title('SIN WAVE 200*pi*t');

xlabel('time');
ylabel('amplitude');
legend('2000080061');

subplot(3,1,2);
plot(t,y);
title('SIN WAVE 200*pi*t + pi/6');

xlabel('time');
ylabel('amplitude');
legend('2000080061');

subplot(3,1,3);
plot(t,z);
title('COS WAVE 200*pi*t - pi/6');

xlabel('time');
ylabel('amplitude');
legend('2000080061');
```



2) Consider an exponential signal $x(t) = k\,e^{-at}u(t)$ for the cases (a) $k = 1$, and $a = 0.35$ (b) $k = 1.2$ and $a = -0.45$. Develop Matlab codes, simulate, and plot the results.

```
clc;
clear all;
close all;
t=-5:0.1:5;
u = @(t) + (t>0);
x = @(t) (1*exp(-0.35*t));
y = @(t) (1.2*exp(0.45*t));

s1 = @(t) x(t).*u(t);
s2 = @(t) y(t).*u(t);

subplot(2,1,1);
plot(t,s1(t));
title('Signal - 1');

xlabel('time');
ylabel('amplitude');
legend('2000080061');

subplot(2,1,2);
plot(t,s2(t));
title('Signal - 2');

xlabel('time');
ylabel('amplitude');
legend('2000080061');
```



3) Explain the following

a) What is the purpose of command '**`clear all`**' ?

- It clears memory in the work space

b) What is the purpose of command '**`close all`**' ?

- It closes all figure windows opened in the MATLAB

c) What is the purpose of command '**`clc`**' ?

- It clears the command windows

d) What is the purpose of command '**`clf`**' ?

- Deletes all children of the current figure that have visible handles

e) What is the purpose of symbol '**`;`**' ?. If we remove this symbol what will happen ?

- It is used to terminate the line. If we not use that symbol ';' then it thinks that program has ended.

f) What is the purpose of symbol '**:**' ? Explain the meaning of the following (i) 1:10, (ii) 1:1:10 (iii) 1:0.1:10.

- ':' symbol is used for scaling from one point to another point. There are 3 parts in the syntax. 1st is starting index. Middle is scale. Last is ending index.
- 1:10 means 1 to 10, 1:1:10 mean 1 to 10 with scale of 1, 1:0.1:10 means 1 to 10 with the scale of 0.1

g) What is the meaning of '**pi**'? What is its value ?

- "pi" returns the floating-point number nearest to the value of π.

h) What is the purpose of command 'sub**plot**' ? Differentiate from '**plot**' command

- "subplot" is used to plot more than one graph in single figure window.
- "plot" is used to generate continuous signal graph.

i) What is the purpose of command '**figure**' ?

- "figure" creates a new figure window using default property values.

j) What is the meaning of letters '**r**', '**g**', '**b**', '**m**', and '**c**' in the '**plot**' command?

- "r" specifies the red color of the plot. "g" specifies the green color of the plot. "b" specifies the blue color of the plot. "m" specifies the magenta color of the plot. "c" specifies the cyan color of the plot.

k) To plot the curve in black color, what is the letter in the plot command to be used ?

- 'k' is used for black color, by including it in the plot code we can see the result.

l) What is the meaning of ‘‘LineWidth’, 1.5 ‘ in the ‘**`plot`**’ command? What is the meaning of numeric value 1.5 ? If this value changes, what will happen ?

- Line width is to change the width of the signal curve. The numerical is unit for the line width.

m) What are the purpose of commands ‘xlabel’ and ‘ylabel’ ?

- Those labels is used to denote the x and y axis in the graph.

n) What is the purpose of command ‘axis’ ?

- “axis” adds the specified title to the axes.

o) What is the purpose of command ‘**`title`**’ ?

- It is used to give the title / heading for the graph

p) What is the purpose of command ‘ **`legend`**’ ? If we remove this command, what will happen ?

- It is used to give name to the signal curve in the graph, If we remove the command nothing affects to the signal in the graph

q) What is the purpose of command ‘**`grid on`**’? If we remove this command, what will happen ?

- It gives grid lines to the graph to understand the structure of the signal and curve of it clearly.

r) Differentiate between the symbols ‘**`*`**’ and ‘**`.*`**’**.**

- ”*” is matrix multiplication while “. *” is elementwise multiplication.