LAB SESSION 1

INTRODUCTION TO MATLAB

WHAT IS MATLAB?

MATLAB stands for **MAT**RIX **LAB**ORATORY. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation

- Algorithm development

- Modeling, simulation, and prototyping

- Data analysis, exploration, and visualization

- Scientific and engineering graphics

- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN.

MATLAB features a family of application-specific solutions called toolboxes. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

MATLAB TUTORIAL 1

This tutorial covers basic MATLAB commands that are used in introductory signals and systems analysis. An easy way to learn MATLAB is to sit down at a computer and follow along with the examples given in this tutorial and the examples given in the textbook. The topics covered in this tutorial are:

1. MATLAB Basics

    A. Definition of Variables

2. Plotting

1. MATLAB BASICS

MATLAB is started by clicking the mouse on the appropriate icon and is ended by typing exit or by using the menu option. After each MATLAB command, the "return" or "enter" key must be depressed.

A. DEFINITION OF VARIABLES

There are several predefined variables which can be used at any time, in the same manner as user- defined variables:

i	sqrt(-1)
j	sqrt(-1)
pi	3.1416...

For example,

```
y= 2*(1+4*j)
yields: y=  2.0000 + 8.0000i
```

There are also a number of predefined functions that can be used when defining a variable. Some common functions that are used in this text are:

abs	magnitude of a number (absolute value for real numbers)
angle	angle of a complex number, in radians
cos	cosine function, assumes argument is in radians
sin	sine function, assumes argument is in radians
exp	exponential function

For example, with y defined as above,

```
                    c = abs(y)
yields: c = 8.2462
                    c = angle(y)
yields: c =  1.3258
```

With a=0 as defined previously,

```
                    c = cos(a)
yields: c =   1
                    c = exp(1)
yields: c =   2.7183
```

2. PLOTTING

Commands covered:

plot

xlabel

ylabel

title

axis

stem

subplot

The command most often used for plotting is ‘plot’, which creates linear plots of vectors and matrices; plot(t,y) plots the vector t on the x-axis versus vector y on the y-axis.

To label your axes and give the plot a title, type;

```
xlabel('time (sec)')
ylabel('step response')
title('My Plot')
```

For discrete-time signals, use the command ‘stem’ which plots each point with a small open circle and a straight line. To plot y[k] versus k, type

```
stem(k,y)
```

To plot more than one graph on the screen, use the command subplot(mnp) which partitions the screen into an mxn grid where p determines the position of the particular graph counting the upper left corner as p=1. For example,

```
subplot(211),
subplot(212),
```

This has 2 rows, one column, and one plot in each row.

Task 1:

Find the Area of Triangle by using MATALB

Area=$\sqrt{s(s-a)(s-b)(s-c)}$

Where s = (a+b+c)/2.

Note: Write the MATLAB code for Task 1 in Comments and Discussion Section.

COMMENTS & DISCUSSION

____

____

____

____

____