Name: _____ Roll No: _____

Score: _____ Signature of the Lab Tutor: _____ Date: _____

DEPARTMENT OF ELECTRONIC & TELECOMMUNICATION ENGINEERING
MEHRAN UNIVERSITY OF ENGINEERING & TECHNOLOGY, JAMSHORO
Feedback Control Systems (1st Term, Third Year, 03TL)
Lab Practice # 01

An Introduction to MATLAB

What is MATLAB

The name MATLAB is short for MATrix LABoratory. It is a commercial software package whose main function is to perform calculations on matrices, row vectors and column vectors. It is widely used in both industry and academic institutions. It possesses many of the features of a high level, numerically oriented programming language, but in addition has a large collection of built-in functions for performing matrix and vector operations in a way that is very simple for the user. For example, to find the determinant of a matrix A one need only enter:

```
det(A)
```

MATLAB commands can be entered one line at a time, or the user can write programs of MATLAB code and define his or her own functions. In this session, we shall only use the one-line-at-a-time "interpretive" mode for simplicity, but the regular user will find that the power of the package is considerably extended by writing his or her own programs and functions.

Entering and quitting MATLAB

To enter MATLAB, simply double click on the MATLAB icon. To leave MATLAB and return to the PC's operating system
Simply type: quit

Creating and manipulating matrices and vectors

MATLAB works on matrices and vectors that may be real or complex. Variables do not have to be "declared". Results can be displayed in a variety of different formats.
Let us enter a row vector into the MATLAB workspace. Type in:

```
v = [2  4  7  5]
```

This creates a variable v whose current value is a row vector with four elements as shown. After pressing "return" the value of v will have been echoed back to you. To suppress the echo, one uses a semi-colon following the command line. Thus

```
w = [1  3  8  9];
```

Creates another row vector w, but does not echo. To check that w has appeared in the MATLAB workspace, type,

```
who
```

Which will display all the variables in the MATLAB workspace, and to check the value of w simply type w
Operations on row vectors can be best illustrated by some simple exercises.

Exercise#1: Investigate the effect of the following commands:

(a) v(2) (b) sum = v + w (c) diff = v – w (d) vw = [v w] (e) vw(2: 6) (f) v'

One way to generate a column vector is to take the transpose of a row vector, as you will have found in exercise 1(f). Column vectors can be typed in directly in one of two ways. For example, to enter the command vector

$$z = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

you can type

```
z = [1; 1; 0; 0]; or
z = [1
     1
     0
     0];
```

Exercise#2: Investigate the effect of the following commands

(i) z' (b) z*v (c) [v; w] (d) v*z (e) [z; v'] (f) z + v'

One way of creating matrices is by multiplying row and column vectors as seen in the above exercises. There are various ways of entering matrices. For example, to enter the matrix

$$M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

the most obvious ways are to type

```
M = [1  2; 3   4];     or
M = [1  2
3              4];
```

but there are more perverse ways such as

```
M = [[1  3]'[2  4]'];
```

Exercise#3:Investigate the effect of the following commands:

(a) N = inv(M) (b) M*N (c) I = eye(2) (d) M + I (e) M*z(1:2) (f) v(3:4)*M
(g) M(1,1) (h) M(1:2,1:2) (i) M(:,1) (j) M(2,:)

Built-in Functions and MATLAB help

You have already encountered some built-in functions. In addition to standard arithmetic operators, such as "+" and "*", you have met inv(A) that generates the inverse of A, and

eye(n) that generates the n-by-n identity matrix. MATLAB has a very large number of built-in functions. To see the full list of those variables on your machine type
help
To get information on a specific function ("inv" for example) type
help inv
The help command is very useful. Most of the MATLAB built-in functions have very descriptive names and are easily remembered or guessed.

Exercise#4: Use the help command to find out about the following built-in functions and make up your own simple examples:
1. ones 2. zeros 3. det 4. Linspace 5. Logspace

1. Graphs

The results of signal processing computations in MATLAB are huge matrices containing, for example, frequency responses, which one would like to display in graphical form. We will go through a simple example of generating data, plotting the graph, putting labels on the axes etc.
We will plot the graph of
y = sin(t);
for t going from 0 to 20 seconds. First we need to generate a vector t containing the values of time that we want to use. Type:
t=linspace(0,20,100);
which makes t into a row vector of 100 values from 0 to 20 inclusive. Then type
ys = sin(t);
To plot the set of points type:
Plot(t,ys)
The basic graph can be prettied up using the following self explanatory sequence of commands:
xlabel('Time in seconds')
ylabel('sin(t)')
title('your Roll No.')
grid

More than one set of points can be plotted on the same axes, and more than one graph can be displayed on the screen at once.
The following exercise leads you though this:

Exercise#5:

1. Generate a vector yc containing the values of cos(t) for the same time range as before.
2. From the "help" entry for "plot" find out how to get both ys and yc plotted against t on the same axis.
3. From the "help" entry for "subplot" find out how to plot ys and yc plotted on two separate graphs, one above the other.