Control Systems Laboratory (EE 3321) — Experiment 1

MATLAB Structure and Use

I. Overview of Experimental Procedure

This first laboratory experiment is an introduction to the use of MATLAB software. The basic user-computer interfaces, data entry techniques, operations, functions, mathematical expressions, and programming techniques are reviewed. The student is asked to perform the various tasks required to use MATLAB effectively in the performance of engineering analysis and design.

Succinct information concerning the basic MATLAB concepts required is presented within the text of this experiment manual document. The MATLAB User's Guide and Reference Guide should be used to extend the student's breadth and depth of understanding.

In this experiment, the student is given information about sources containing examples of MATLAB operations and functions, which should be reproduced by the student to gain hands-on familiarity and confidence with MATLAB. Further understanding of these methods and procedures is then promoted using additional, more complex examples of the student's own design.

II. Basic MATLAB Structure and Script Management

Once you have logged in to your account, Click on the MATLAB icon on the desktop. (MATLAB is also available through the Start Menu.) Once loaded, the the Command Window, Workspace, and Current Folder windows are presented. Using the Current Folder window, navigate to the location that will be used for this laboratory. MATLAB codes will be written in the form of a Script, or .m file. These are files of statements that can be executed by MATLAB when the file name is typed in the Command Window. They are created and edited in the Editor/Debugger window. One of the benefits of using .m files is that you need not retype commands over and over in the Command Window every time you make a mistake.

The following code should be present at the beginning of each MATLAB Script. It is a comment block that designates student and laboratory identification, followed by code that is used to clear the Workspace, close all figure windows, and clear the Command Window.

```
%{
    Student Name
    Missouri University of Science and Technology
    Department of Electrical and Computer Engineering
    Control Systems Laboratory (EE 3321)
    Experiment No.
    Due Date
%}
clear all; close all; clc;
```

Single-line comments may be written easily by preceded commented text with a **%** symbol. Section headers are created using a **%%** symbol. (Note that a space must be included after the symbol — following text designates the section title.)

III. MATLAB Informational Sources

The Discrete-Time Linear Systems Laboratory has an *Appendix* for MATLAB use. Students should be familiar with the practices and techniques of all of Chapter 1 (Sections 1.0 through 1.11). Reproduction of these sections will aid in the completion of this experiment and subsequent experiments involving MATLAB.

The text, *MATLAB for Control Engineers*, by Katsuhiko Ogata is also a great reference, and contains information for both MATLAB coding and control system analysis and design.

IV. MATLAB Function Help

MATLAB has an online help facility that may be invoked whenever the need arises. The command **help** will display a list of predefined functions and operators for which online help is available. The following command line will give information on the purpose function of the **roots** function (used as an example).

```
help roots
```

V. Experimental Procedure

1. Use the MATLAB command **roots()** to find the roots of the polynomial $2x^2+4x+10$. Verify your answer by using the quadratic formula to manually compute the roots.
2. Use the MATLAB command **conv()** to find the product of the polynomials $3x^2+7x+15$ and $4x^3+2x^2+1$.
3. Enter the character string: `x = 'We learn to use MATLAB in EE 3321 Laboratory';` Set the variable **y** equal to the character string **MATLAB Laboratory** by selecting the desired words and spaces from the variable x. (Hint: Since the two desired character strings are not consecutive, using the form [A B] to define y will be useful.)
4. Find the matrix **a-bc²+2d`** when the matrices **a**, **b**, **c**, and **d** are:

```
a = [  1.5    3.3                 b = [  0.5    0.3
       6.0    -4.5                       -0.1   0.2
       -2.5   0.7];                      0.4    -0.3];

c = [  1      2                   d = [  3.1    1.4    -0.3
       1      2];                        -0.5   1.6    0.1];
```

5. We define the element of the matrix **A** in row **i** and column **j** to be $\mathbf{A_{ij}}$. Use the above matrices and array operations to find $\mathbf{b_{ij}-c_{ij}(d_{ij})^4}$ for i = 1 and j = 2.
6. Create a function .m file, called **sumsin.m** that sums two sinusoids. The inputs should be **t**, f_1, and f_2; the outputs, $\mathbf{s_1} = \sin(2\pi f_1 t)$, $\mathbf{s_2} = \sin(2\pi f_2 t)$, and $\mathbf{s_3} = s_1 + s_2$.
7. Observe the characteristics of the three functions **round**, **floor**, and **ceil** on

```
x = [-3.6  -2.5  -1.4  -1  0  1.4  2.5  3.6];
```

8. Use **for** statements to find the values of $x(t) = 3\cos(2\pi ft+0.1)$ for t = 0, 0.1, 0.2, 0.3, and 0.4 s when f = 10, 15, and 20 Hz. Use one set of statements to compute the values for all three frequencies and store the results in a two-dimensional array. (Hint: Use two nested for loops, and a double index.) What is the value of x(t) for the parameters f = 15 Hz and t = 0.3 s?
9. Use the **while** statement to find the largest value of t for which $e^{1.2}\cos(\omega t)$ and t^3 are both less than 10. Make the computation for ω = 35, 40, and 45, and find your answers to the nearest 0.01 s.

10. Create a script that uses the sum of sinusoids function created earlier in the experiment, and plot the output signals. Use t = 0:0.01:10, $f_1 = 0.2$, $f_2 = 0.425$.

```
[s1 s2 s3] = sumsin(t, f1, f2);
```

Plot all three sinusoids on the **same axis**. **Label** the axis, and **title** the plot with your initials, experiment number, and a brief plot description. Create a **legend**, or plot label. In a new figure window, plot all three sinusoids on **separate axes**, but in the same window. **Title** each axis appropriately.

VI. Conclusion

Answer the following questions:

1. Have you used MATLAB before this experiment? If so, what MATLAB operations and/or concepts were new to you? Are any of the concepts from this experiment confusing?
2. How would you change this experiment? Would you add/remove any specific procedures or topics?