University of Washington
Department of Electrical Engineering
EE 235 Lab 1 Part One:
Introduction to Matlab

In this lab, you will work through a series of exercises to get you started in Matlab ("Matrix Laboratory"). You will learn how to use Matlab variables, perform basic operations, and write your own script files.

Note:

- Pre-labs should be done <u>individually</u>, and turned in at the beginning of the lab. Each student will be checked off by the TA during the lab section.
- All in-lab exercises and the lab report should be completed in groups of 2-3.
- No lab report this week – lab report will be with part 2 next week

<u>What Is Expected From You In Lab 1 Part One</u>

□ Completion of the Prelab exercise: show to the TA at beginning of lab (2 points)

□ Completion of 2 in-lab check offs with TA (2 points)

<u>PRE-LAB</u>

Read the *GettingStarted* tutorial and familiarize yourself with the Matlab user interface. This material will be needed to answer the prelab questions and in the lab.

1. Set up the Remote Desktop Connection. Create a new folder by right-clicking in CURRENT FOLDER window and selecting New Folder. Call this folder **Lab 1.** Double-click the folder **Lab 1** and then press "Select Folder". You will see that your "Current Folder" now shows you are in the **Lab 1** directory that you specified. Create a script template for future use: Write your name, lab section, and the date in your EDITOR window, save the file as `template.m`. *Make a screenshot showing the Matlab interface with the specified folder and script template.* Exit Matlab and then restart it. Make sure you can find this file again, both through Matlab and using a folder browser outside of Matlab.
2. While we will be working on paper with continuous-time signals, we need to use discrete-time representations of those signals in lab.
   a. Give an equation defining sampling time T_S in terms of sampling frequency F_S
   b. How many time samples are in a vector created by the Matlab command t=0:0.5:2 ?

<u>IN-LAB</u>

For the following exercises, refer to the *GettingStarted* tutorial, specifically Sections **Matlab Variables** and **Basic Operations** for Exercise 1 and Section **Creating a Signal that is a Function of Time** for Exercise 2. Ask your TA if something isn't clear.

1. ***<u>Lab Exercise #1</u>***: *M-files*
   We will write code using M-files which have the extension *.m. M-files can be used as scripts that contain a sequence of commands that are executed exactly as if they were manually typed into the Matlab COMMAND window.
   a) Open up a new script by doing the following: Go to Home tab -> New Script

b) Save your script: Go to Editor tab -> Save. Then, save file as **Ex1.m**

c) On the Matlab script window on the upper center window, first copy or type in the following lines of code and comments (which are done in Matlab using the % symbol):

```
% FILE: Ex1.m
% NAME: [FILL IN NAME HERE]
% DESCRIPTION: Matlab Variables and Basic Operations

% Clear all variables and close all windows
clear all; close all;

% PART A

% PART B

% PART C

% PART D
```

d) Under PART A:
   - i) Write the line of code to create a 3 x 1 column vector called **y1** with values 4, 6, and 2. Suppress the output using a semicolon at the end of the line of code.
   - ii) Write the line of code to a 3 x 3 matrix called **z**, where the first row has all ones, the second row has values 3, 6, and 9, and the third row has all zeros. Again, suppress the output by using a semicolon at the end of the line of code.
   - iii) Test your current script by pressing <F5> or by going to Editor -> Run. You should see your two variables **y1** and **z** on your WORKSPACE window.

e) Under PART B:
   - i) Write the line of code to extract the value 6 in vector **y1**, and store that value in variable **c**. Display the output so we can see the value of **c** on the COMMAND window. Therefore, omit the semicolon at the end of the line of code.
   - ii) Using the colon operator, write the line of code to extract the last two elements in vector **y1** and store them in variable **c1**. Display the output by omitting the semicolon.
   - iii) Test your script so far by pressing <F5> or by going to Editor -> Run. You should see the variables **y1**, **z**, **c,** and **c1** on your WORKPSACE window. You should also see the values of **c** and **c1** displayed on the COMMAND window.

f) Under PART C:
   - i) Write the line of code to access the value 9 in matrix **z**, and store that value in variable **d**. Display the output on the COMMAND window.
   - ii) Using the colon operator, change the values of matrix **z** so that the last two columns in the 3rd row are -1. Display the output on the COMMAND window.
   - iii) Test your script so far by pressing <F5> or by going to Editor -> Run. You should see the variables **y1**, **z**, **c**, **c1**, and **d** on your workspace window. You should also see the values of **c**, **c1**, **d**, and **z** displayed on the COMMAND window.

g) Under PART D:

i) Write the line of code to get the number of columns in vector **y1**, and store that value in variable **e**. For this part, display the output on the COMMAND window.

ii) Test your script so far by pressing <F5> or by going to Editor -> Run.

CHECKOFF #1: Demonstrate your script for **Ex1.m** with your Lab TA**.**

2. ***Lab Exercise #2***: *Defining signals*

a) Open up a new script and save it as **Ex2.m**

b) Copy or type in the following for **Ex2.m**:

```
% FILE: Ex2.m
% NAME: [FILL IN NAME HERE]
% DESCRIPTION: More Matlab Variables and Basic Operations

% Clear all variables and close all windows
clear all;
close all;

% PART A

% PART B

% PART C

% PART D
```

c) For PART A: Write the lines of code to define the following two signals **x(t) = 0.5t** and **y(t) = t².** Both signals will be defined over the range **0 ≤ t ≤ 3** with a sampling frequency of 2.

d) For PART B: Write the line of code to define a new signal **z(t)**, where **z(t) = x(t) – 2y(t)**.

e) For PART C: Write the line of code to access **z(t)** at t = 2, and store the result in variable **w1**. Display the output on the COMMAND window.

f) For PART D: Write the lines of code to extract values of **z(t)** from **0 ≤ t ≤ 1.5**, and store the values in variable **w2**. Display the output on the COMMAND window.

g) Run and test your script.

CHECKOFF #2: Demonstrate your script for **Ex2.m** with your Lab TA**.**