ENGR 253 LAB #1 - MATLAB Introduction

Objective
Introduction to using MATLAB with focus on Signal Processing.

Resources
- Course Lecture Material
- MATLAB or GNU Octave Development Environment

Background

- MATLAB vs. GNU Octave
  For this class you can use GNU Octave which is the most popular alternative to MATLAB and has had an active development for over three decades. GNU Octave support is available for Windows, IOS and Linux. Octave is also the most compatible alternative to MATLAB. For more information and to download GNU Octave visit www.gnu.org/software/octave .

  A good GNU Octave tutorial is available at:
  https://www.tutorialspoint.com/matlab/matlab_gnu_octave.htm

  For demo portion of this lab, you can use the help menu in Octave or MATLAB but it is more effective to search Google for "MATH Examples" or "GNU Octave Examples". Here are a couple of links:

  * https://www.mathworks.com/help/examples.html
  * https://octave.org/doc/v4.2.1/Simple-Examples.html

- MATLAB Overview

  MATLAB system consists of five main parts:

  - Development Environment
    It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

  - Mathematical Function Library
    A collection of computational algorithms ranging from functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

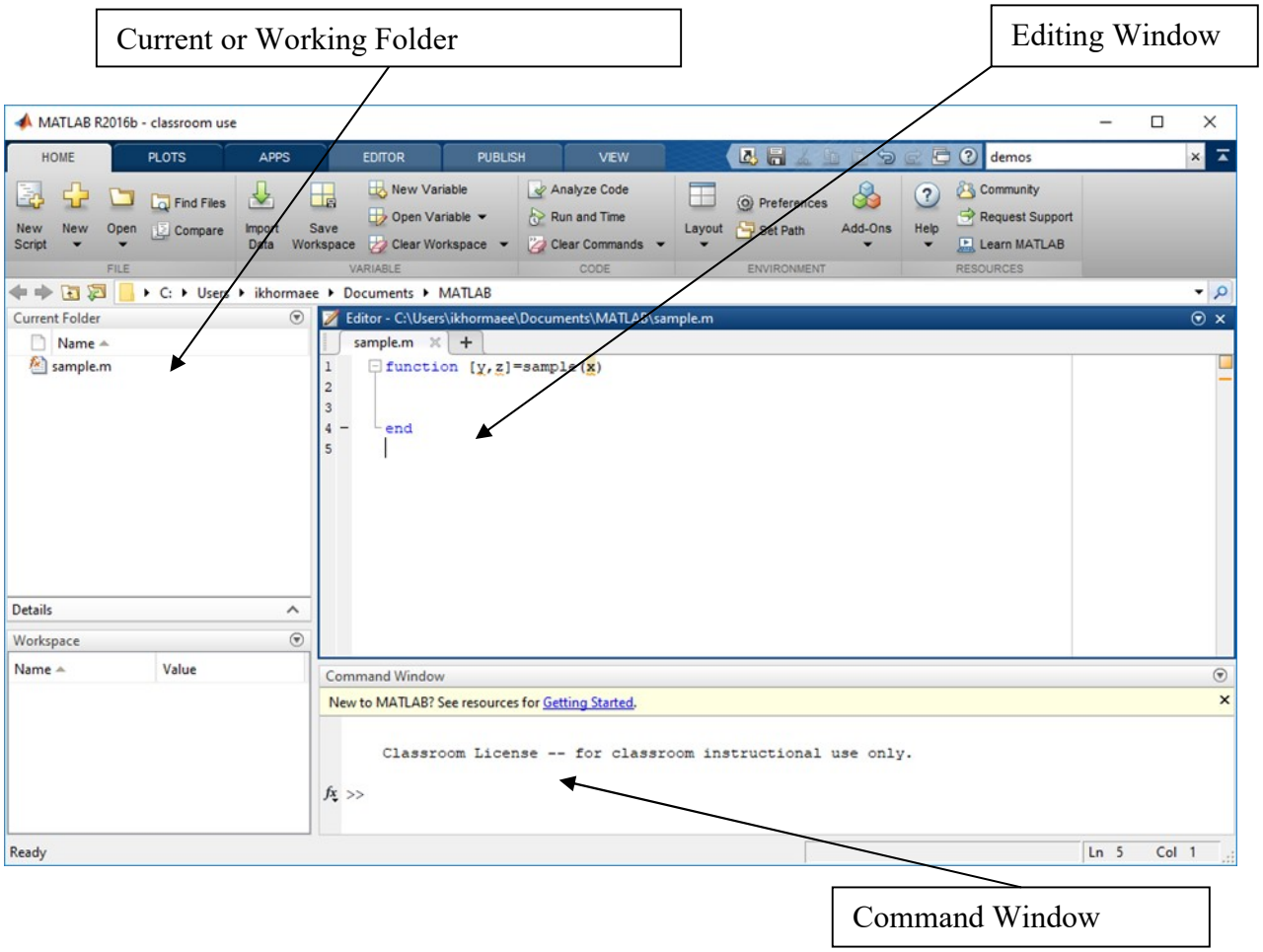
  - MATLAB Language
    A high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

  - Graphics
    MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

  - Application Program Interface (API)
    This is library enables programs written in C and FORTRAN to interact with MATLAB.

- To start MATLAB go MS Windows' Start menu and select MATLAB application from the menu or click on the desktop icon if present.

❖ Components of MATLAB Development Environment
"Arrangement of components may be different depending on the version"



❖ Editor Window
From the home tab click on the "New Script" to create a new script or MATLAB Program. It is important to save the file in your desired directory. Note that lab computers erase all use files when power is recycled.

❖ What can be done with MATLAB?
From home table select the menu item <help><Examples> and review the MATLAB Examples. It is important to become comfortable with online documents to answer questions and find information needed.

❖ Commands Quick Reference:

Command or Syntax	Description
Hints	➢ MATLAB is case sensitive which means "Stem" is different from "stem" ➢ Reserve words and name cannot be used as variable or function names. ➢ Help command displays command usage instruction ➢ % at the beginning of line indicate comment. The line is not executed. ➢ ; at the end of line prevents command results from being displayed
Arithmetic Operators	Operators used for matrix, complex and real + Add - Subtract * Multiply / Divide ^ Power .* Term-by-term operator (matrix only) ./ Term-by-term operator (matrix only) .^ Term-by-term operator (matrix only)
x=expression	Assign the expression to x
n=[-100:100]	Creates array n with 201 elements from -100 to 100
zeros(r,c)	Number of rows and columns of zeros to be created
ones(r,c)	Number of rows and columns of ones to be created
A=[11, 12, 13; 21, 22, 23]	Create and initialize to 2x3 array: 11 12 13 21 22 23
t = [-5:0.1:5]	Creates an array t with 101 elements from -5 to 5 with .1 steps
A(2,3)	Return the value of element at 2nd row and 3rd column of array A
plot(t, x)	Plots x as a continuous time signal over range of t. t and n must be arrays of the same size.
stem(n, x)	Plots x as a continuous time signal over range of t. t and n must be arrays of the same size.
title('title_string') xlabel('x_axis_name_string') ylabel('y_axis_name_string')	Adding a title to the graph Adding horizontal axis label Adding Vertical axis label
exp(v) pi ()	Returns value of e^v where e=2.71 Returns value of π or 3.14
sin(Av) cos(Av) tan(Av)	Returns the corresponding trig value for angle Av in radians.

m file script	You can type commands directly into script file (file_name.m) then run the script by simply typing its name without ".m" in the command window.

function "name.m"	% This function accepts inputs x and returns y,z
	function [y,z]= sample(x)
	  y=x/2;
	  z=x*2;
	end
	
	If the above function is saved in a m file script as sample.m then by typing the following in the command window:
	>> sample(2)
	
	The following output will be displayed:
	y=
	  1
	z=
	  4

for Loop	for loop repeats statements a specific number of times. The general form of a for statement is: for variable = expr, statement, ..., statement end Example: for I = 1:N, for J = 1:N, A(I,J) = 1/(I+J-1); end end

if Statement	if conditionally executes statements. The general form of the IF statement is:
	if expression
	statements
	elseif expression
	statements
	else
	statements
	end
	
	Example:
	if I == J
	A(I,J) = 2;
	elseif abs(I-J) == 1
	A(I,J) = -1;
	else
	A(I,J) = 0;
	End

disp(x)	Displays value of x that can be number of strings
x=input(prompt)	Displays the prompt and saves user's input in x.

Experiment #1

Use the <Help ><Examples> menu search through MATLAB examples. Document the top three most relevant examples to the course (Signal and Systems). For each example, include:

- Description of each example's functionality
- Reason for selecting the example as the top three with focus on relevance to the course topics

Experiment #2

Select one of the three examples from experiment #1. For the selected example, include:

- Description of example functionality
- Reason for selecting this example from the three listed in Experiment 1
- Components used in the example and brief description of each
- Code listing (if longer than 2 pages, only include the first 2 pages)
- Sample/Test Results

Describe the components used in the example and the functionality of the example.

Report Requirements

Lab and reports must be completed individually. All reports must be computer printed (Formulas and Diagrams may be hand drawn) and at minimum include:

For each Experiment

a) A clear problem statement: specifying items given and to be found.
b) Theory or process used.
c) Resulting circuits, calculation, tables, timing diagram, schematic and other relevant results.

For the report as a whole

a) Cover sheet with your name, class, lab, completion date and team members' names.
b) Lessons Learned from the experiments.
c) A new experiment and expected results, which provide additional opportunity to practice the concepts in this lab.