ENGR 253 LAB #1 - MATLAB Introduction

Objective
Understanding and hands on experience with MATLAB with focus on Signal Processing.

Resources
- Signals & Systems textbook by Oppenheim and Willsky
- Windows running MatLab release 14 or later
- USB hard disk or other removable drives (note Lab computer data is lost after reboot)
- Course Lecture Material

Background

❖ 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.

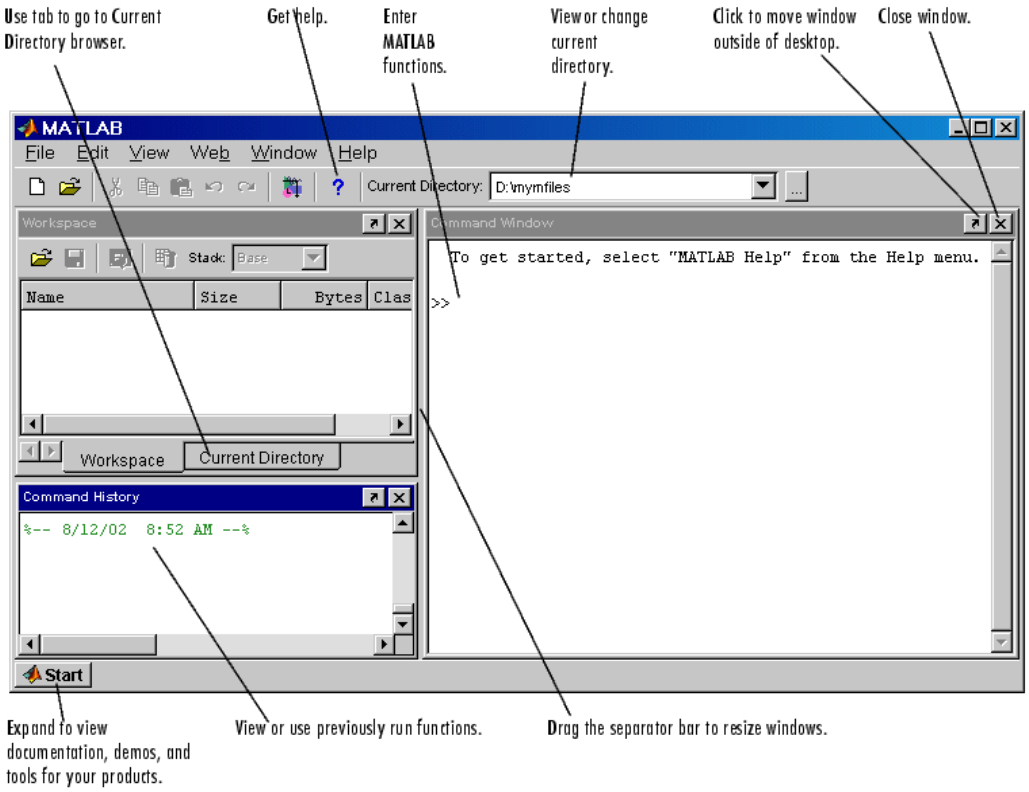
- Graphics
  MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes 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.

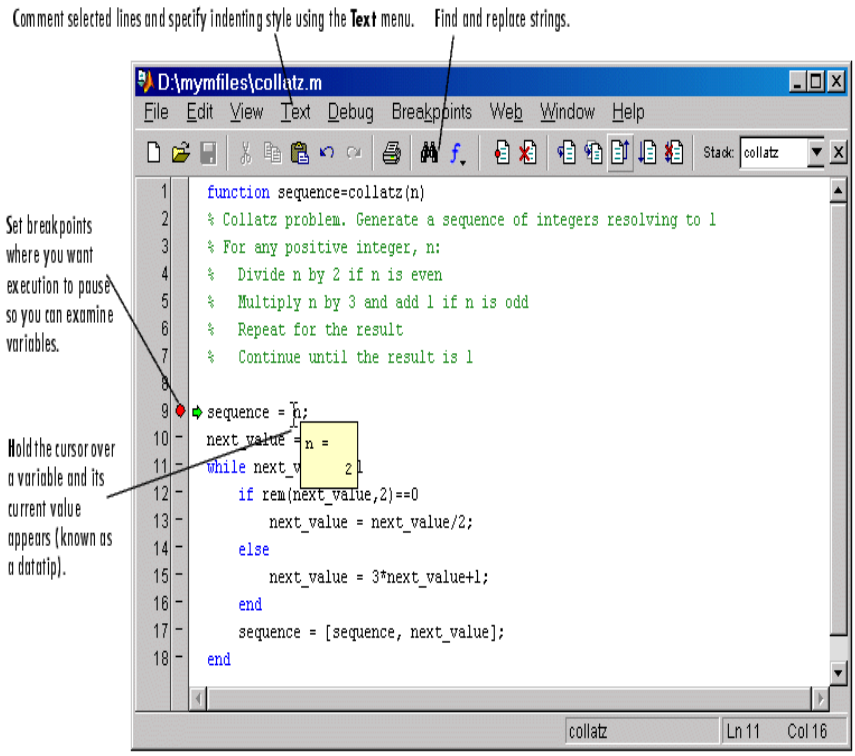
❖ To start MATLAB go MS Windows' Start menu and select the following:

<Start><MATLAB><MATLAB>

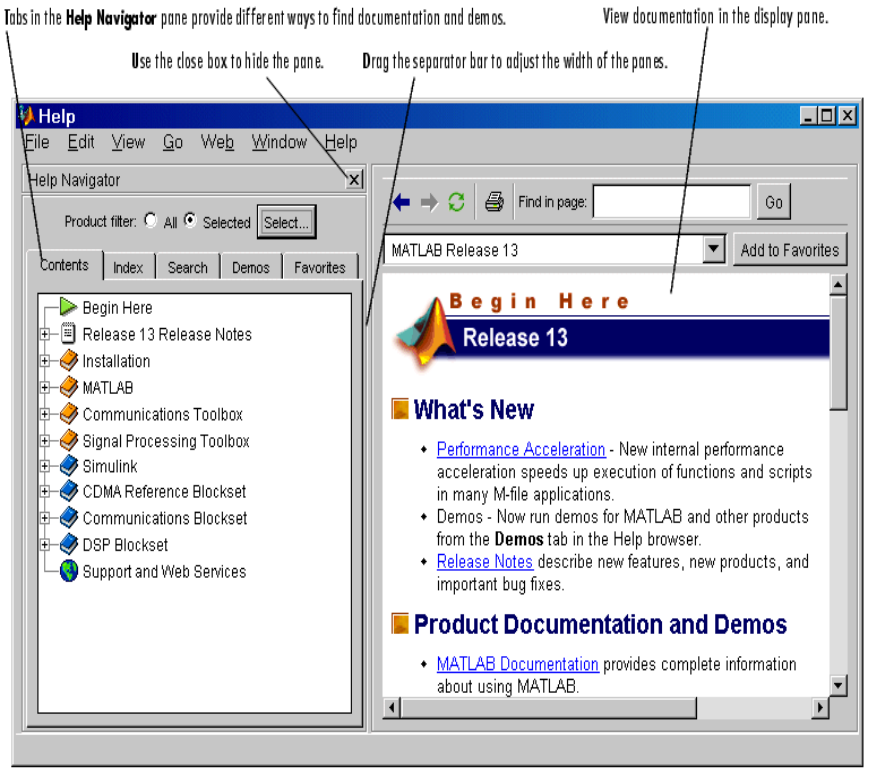
❖ Components of MATLAB Development Environment



❖ Editor/Debugger window
In Development Environment, right click in Workspace window to create a new test.m file. Once you have a file now by double clicking on it you will have access to Editor/Debugger window:


Comment selected lines and specify indenting style using the Text menu.
Find and replace strings.
Set breakpoints where you want execution to pause so you can examine variables.
Hold the cursor over a variable and its current value appears (known as a datatip).
D:\mymfiles\collatz.m
File Edit View Text Debug Breakpoints Web Window Help
Stack: collatz
function sequence=collatz(n)
% Collatz problem. Generate a sequence of integers resolving to 1
% For any positive integer, n:
% Divide n by 2 if n is even
% Multiply n by 3 and add 1 if n is odd
% Repeat for the result
% Continue until the result is 1
sequence = n;
next_value = n
while next_v
n = 21
if rem(next_value,2)==0
next_value = next_value/2;
else
next_value = 3*next_value+1;
end
sequence = [sequence, next_value];
end
collatz
Ln 11 Col 16

❖ Help Section
From MATLAB Development Environment select <help><MATLAB help>



❖ What can be done with MATLAB?
From HELP Section select the demo tab and review the various MATLAB Demos.
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 *commandName*	Display help on commands
%	At the beginning of line indicate comment (Not executed)
;	At the end of line causes the command results to not be displayed
n=[-100:100]	-100<= n =<100
x=expression	Assign the expression to x
zeros(r,c)	Number of rows and columns of zero to be created
stem(n,x2)	Discrete plot of function x2 with index n
A(2,3)	Return the element at 2nd row and 3rd column of Array
A=[11, 12, 13; 21, 22, 23}	Create and initialize to 2x3 array: 11 12 13 21 22 23
t = [-5:0.1:5}	Produces a set of indices from -5 to 5 with .1 steps
Plot(t, x)	Plots x as a continuous sign over index t range in general you want to use plot for continuous-time and stem for discrete

	time.
title('title_string') xlable('x_axis_name_string') ylabel('y_axis_name_string')	Title and label axis
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)
exp(expression)	Return exponential value
sin(expression) cos(expression)	Basic trig. functionality
M file script	You can type commands directly into M file (file_name.m). you can run the script by simply typing its name without ".m"

M file function "name.m"	Function [y,z]= sample(x)
	%You can create a function that accepts inputs x and return output y,z:
	y=x/2;
	z=x*2;
	
	you can call this function from the command window by typing its name
	>> sample(2)
	y=
	1
	z=
	4

for Loop	for Repeat 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 execute 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

Experiment #1

Run the Matlab demo by using the "Help > Demo Menu". Document the top three most relevant demos to the course (Signal and Systems). For each demo, include:

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

Experiment #2

Select one of the three demos from experiment #1 with instructor's pre-approval. For the selected demo include:

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

Decribe the components used in the demo and the functionality of the demo.

Report Requirements

Reports must be prepared individually even if the experiments are performed as a team. 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.