**Department of Electrical Engineering**

|  |  |
| --- | --- |
|  |  |
| **Course/Section: BEE-8D** | **Semester: 4th** |
|  |  |

**EE-232 Signals and Systems**

**Lab #1 Introduction to Matlab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Reg. no.** | **Report Marks / 10** | **Viva Marks / 5** | **Total/15** |
| **Muhammad Haris Rehman** | **182797** |  |  |  |

before coming to the lab and soft copy of Pre-lab session should be deposited on LMS prior to start of the lab for necessary evaluation. The lab report should be submitted on LMS separate

# Familiarize yourself with MATLAB

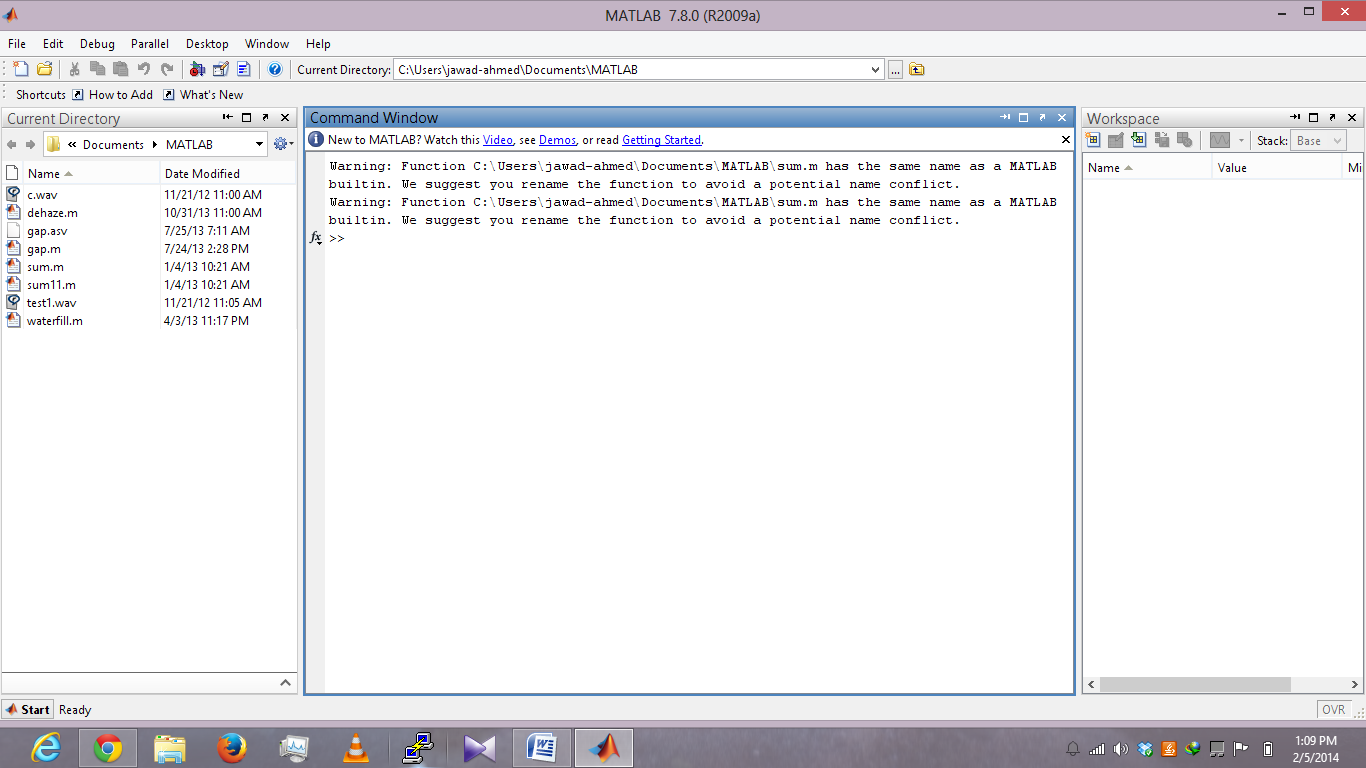
Let us introduce MATLAB in windows environment. Front-end (Graphical user interface) of MATLAB is very easy to use and is user friendly. Further it is compatible with standard windows applications and supports

* File operations
* Multiple windows view.
* help, demos and example (Check whether this component is installed)
* Wizard to create a GUI (graphical user interface)
* Wizard to profile code
* Toolboxes for different components like communication, control systems, data acquisition, curve fitting, fuzzy logic, neural network etc. (These components may or may not be available, depending upon the installation)

When MATLAB application is started, it looks like figure 1.

Figure 1: MATLAB Windows

**Menu Current Directory Workspace window with variable details**



**Command Window**

|  |  |
| --- | --- |
| **WINDOW** | **PURPOSE** |
| Command Window | To enter commands |
| Command History Window | To see previous commands |
| Workspace window | Provide information about the variables that are used |
| Current Directory Window | Shows the files in current directory |
| Launch pad window (Toolbox Menu) | Provide access to tools and demos |

Table 1: Description of common windows

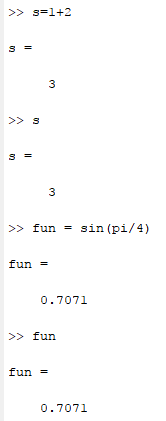
There are other windows of MATLAB, which are equally important. Table 2 describes them in brief and will be introduced later.

|  |  |
| --- | --- |
| **WINDOW** | **PURPOSE** |
| Help window | Provides Help information |
| Editor window | Text editor to write programs |
| Figure window | Output of Graphical commands |

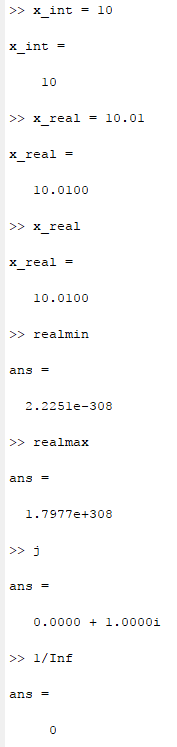
Table 2: Description of other windows

## Pre-Lab Tasks

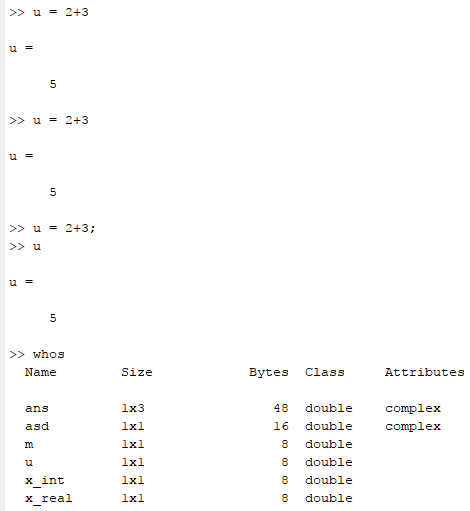
### The Command window



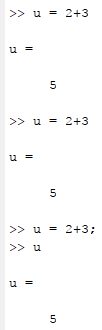
### Numbers in MATLAB



### Workspace in MATLAB



### MATLAB HELP

****

## Lab Tasks

### Lab Task 1:

(a) Make sure that you understand the **colon** notation. In particular, explain in words what the following MATLAB code will produce

a = 0 : 6

b = 2 : 4 : 17

c = 99 : -1 : 88

d = 2 : (1/9) : 4

e = pi \* [ 0:0.1:2 ];

(b) Extracting and/or inserting numbers into a vector is very easy to do. Consider the following definition of f:

f = [ zeros(1,3), linspace(0,1,5), ones(1,4) ]

f(4:6)

size(f)

length(f)

f(2:2:length(f))

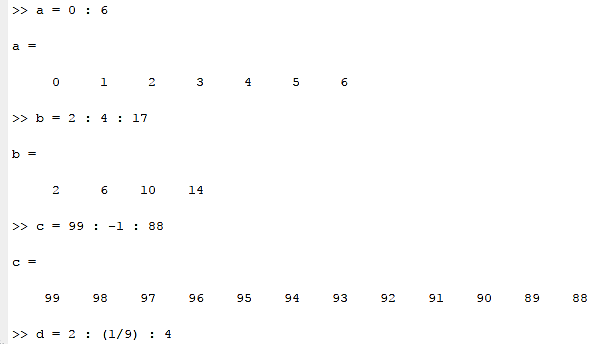
Explain the results echoed from the last four lines of the above code.

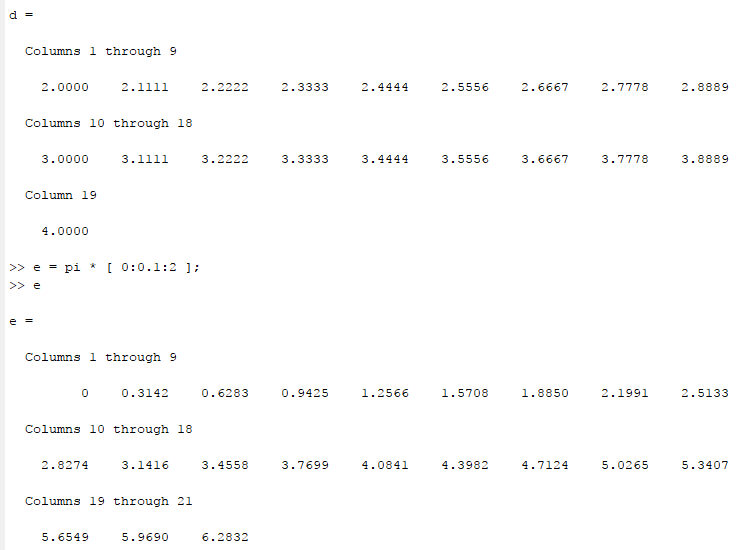
(c) Observe the result of the following assignments:

g = f; g(4:6) = pi\*(1:3)

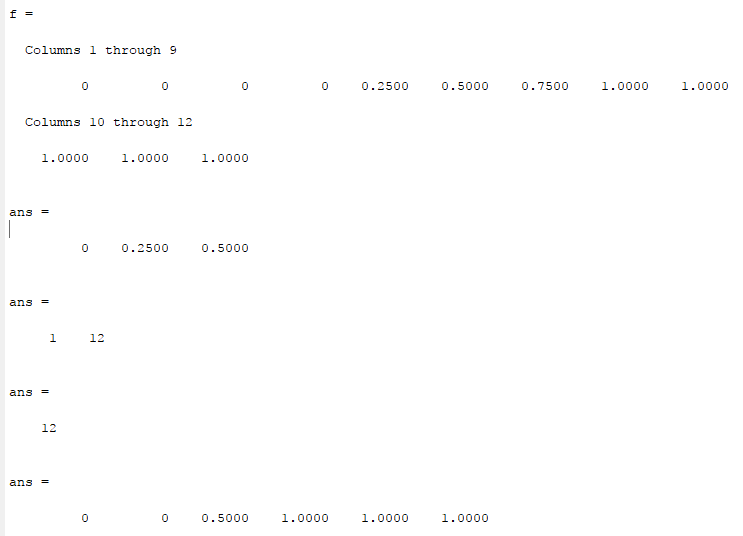
**Solutions of Task 1:**

a)



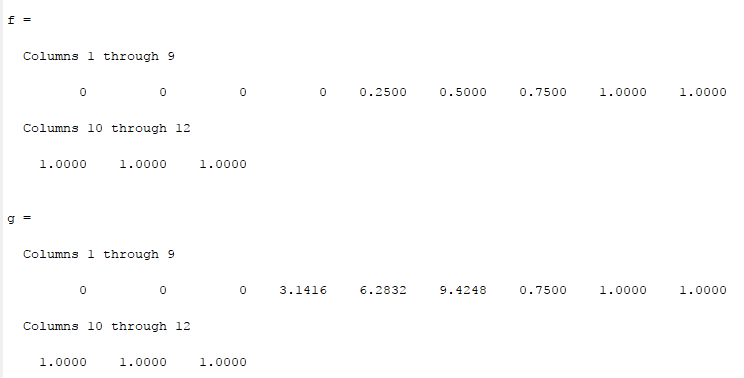


b)



**Ans:** size function gives the size of the vector i.e. 1 12, whereas the length function gives the length of vector i.e. 12. The last line of code prints the values at the even places of the vector i.e. starting from 2 upto the length of vector and incremented by 2

c)



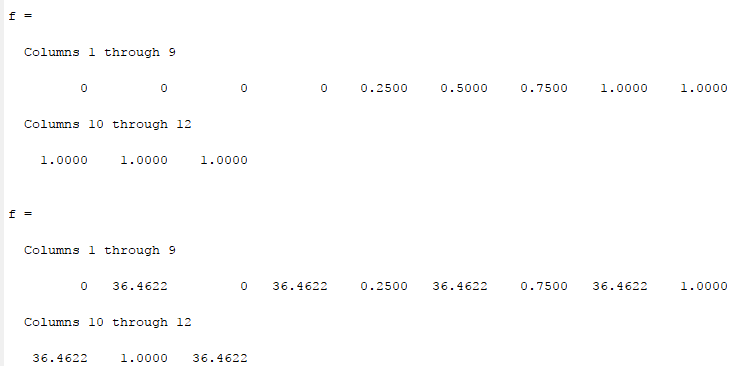
### Lab Task 2:

Now write a statement that will take the vector f defined in part (b) and replace the even indexed elements (i.e., f(2), f(4), etc) with the constant ‘ππ ‘ (pi raised to the power pi) (Try: finding help on ‘^’ operator or the function ‘power’). *Use a vector replacement, not a loop.* Experiment with vectors in MATLAB. Think of the vector as a set of numbers. Try the following:

h = cos( pi\*(0:11)/4 ) %<---comment: compute cosines

Explain how the different values of cosine are stored in the vector h. What is h(1)? Is h(0) defined?

**Solution of Lab Task 2:**



**Ans:** h(1) will be the first element of the vector, whereas h(0) doesn’t exist.

### Lab task 3:

Loops can be written in MATLAB, but they are NOT the most efficient way to get things done. It’s better to **always avoid loops** and use the colon notation instead. The following code has a loop that computes values of the cosine function. (The index of g() must start at 1.) Rewrite this computation without using the loop (follow the style in the previous part).

g = [ ]; %<--- initialize the g vector to be empty

for k=-5:5

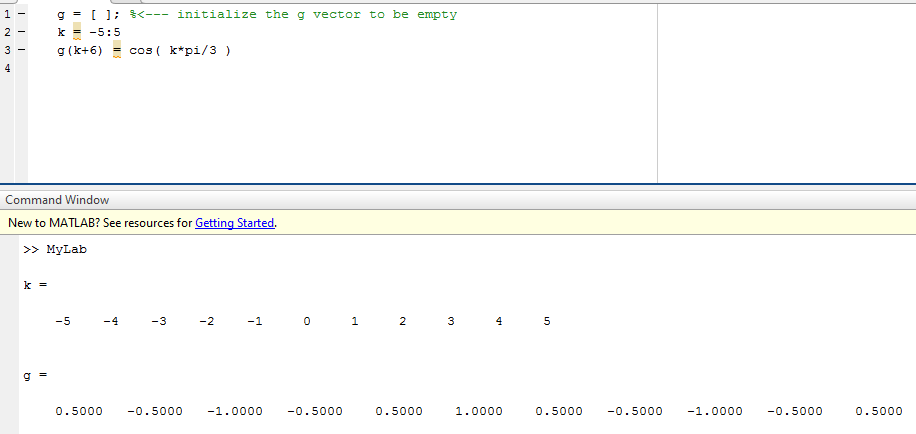
g(k+6) = cos( k\*pi/3 ) ;

end

g

Explain why it is necessary to write g(k+6). What happens if you use g(k) instead?

**Solutions to Lab Task 3:**



**Ans:** The indices must be the real positive numbers. If I use the g(k), it will be an error. So, it is necessary to use g(k+6).

**Plotting in MATLAB**

Plotting is easy in MATLAB for both real and complex numbers. The basic plot command will plot a vector y versus a vector x connecting successive points by straight lines. Try the following:

x = [-3 -1 0 1 3 ];

y = x.\*x - 3\*x;

plot( x, y )

z = x + y\*sqrt(-1)

plot( z ) %<---- complex values: plot imag vs. real.

Use help arith to learn how the operation x.\*x works when x is a vector; compare to matrix multiply.

Note: stem() command is used to plot discrete set of data.

### Lab task 4:

Go to File > New > M –file. MATLAB editor will open up. Enter the following code in the editor and then save the file as mylab1.m

clear all;

clc;

t = -1 : 0.01 : 1;

x = cos( 5\*pi\*t );

y = 1.4\*exp(j\*pi/2)\*exp(j\*5\*pi\*t);

plot( t, x, ’b-’, t, real(y), ’r--’ ), grid on

%<--- plot a sinusoid

title(’TEST PLOT of a SINUSOID’)

xlabel(’TIME (sec)’)

Now go to Command Window and type

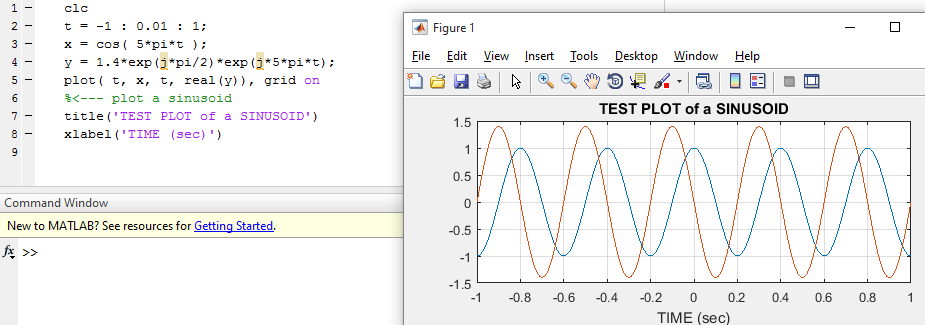
mylab1 %<---will run the commands in the file

type mylab1 %<---will type out the contents of

% mylab1.m to the screen

Explain why the plot of real(y) is a sinusoid. What is its phase and amplitude? Make a calculation of the phase from a time-shift measured on the plot.

**Solutions To Lab Task 4:**



**Functions in MATLAB**

User-defined functions (i.e., those no pre-programmed in Matlab) can be defined by using an m file. These files are simply text files whose name ends with the suffix .m. Those m files that a user may create are typically stored in the work directory. This directory is the default working directory of a MATLAB installation.

Functions defined by m files start with the line

function [assignment\_variable(s)] = function\_name(arguments)

Here, function is a required MATLAB particle to identify the file as a function, assignment\_variable(s) is an optional dummy variable, function\_name is the name given to the function, and arguments are optional values passed on to the function from the main MATLAB interface or from within another function.

### Lab task 5:

Create a function “sigadd” to add two sequences ‘x1’ and ‘x2’.

Function [y,n]=sigadd(x1,n1,x2,n2)

Where ‘x1’ and ‘x2’ are two sequences and ‘n1’ and ‘n2’ are their respective indices vectors. Add values of ‘x1’ and ‘x2’ at corresponding indices, pad zeros if length of two sequences are not same.

Suppose x1= [1 2 3 4 5 6 7 8 9] with index n1= 3:11 and x2= [2 4 6 8 10 12 14 16 18 20 22 24] with index n2=1:12. Here you can observe that the length of both the signals is not same and the indexes of both the signals are not starting from the same point. So you have to pad zeros before adding both the sequences so that the output y will have the index values starting from 1 up to 12.

Hint: You may need the loops and if else checks. Loops syntax is already discussed above and syntax of if else is given below.

**Syntax of if/elseif/else**

Execute statements if condition is true

if *expression*

*statements*

elseif *expression*

*statements*

else

*statements*

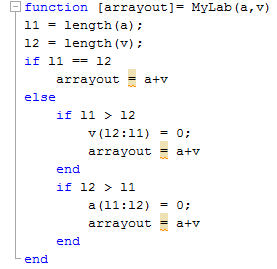
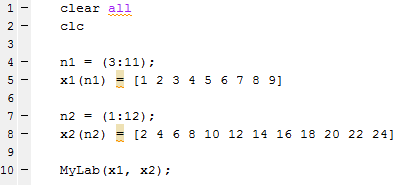
end

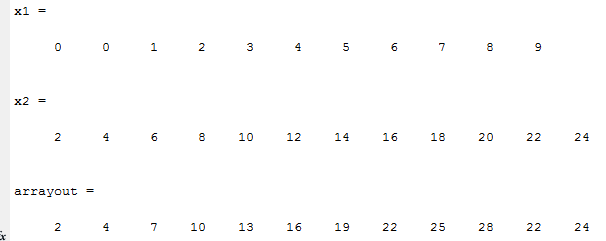
if *expression*, *statements*, end evaluates an expression, and executes a group of statements when the expression is true.

elseif and else are optional, and execute statements only when previous expressions in the if block are false. An if block can include multiple elseif statements.

An evaluated expression is true when the result is nonempty and contains all nonzero elements (logical or real numeric). Otherwise, the expression is false.

**Solutions to Lab Task 5:**

****

****

**Best Resource or learning MATLAB: For getting familiar with MATLAB operations, matrices, arrays, loops and plots etc., go to the link given below which contain official documentation on mathworks with name “Getting Started with MATLAB”** www.mathworks.com/help/pdf\_doc/**matlab**/**getstart**.pdf‎