**Department of Electrical Engineering and   
Computer Science**

**Faculty Member:** Dr. Salman Ghafoor  **Dated:** 12/09/2022

**Semester:** 5th **Section:** BEE 12C

**EE-232: Signals and Systems**

Lab 1: Introduction to MATLAB

Group Members

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **PLO4 – CLO3** | **PLO5 - CLO3** | **PLO8 -CLO4** | **PLO9 -CLO4** |
| **Name** | **Reg. No** | **Viva / Quiz / Lab Performance** | **Analysis of data in Lab Report** | **Modern Tool Usage** | **Ethics and Safety** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** |
| Danial Ahmad | 331388 |  |  |  |  |
| Muhammad Umer | 345834 |  |  |  |  |
| Syeda Fatima Zahra | 334379 |  |  |  |  |
|  |  |  |  |  |  |

**Table of Contents**

[2 Table of Figures 2](#_Toc114427835)

[3 Introduction to MATLAB 3](#_Toc114427836)

[3.1 Objectives 3](#_Toc114427837)

[3.2 Equipment 3](#_Toc114427838)

[3.3 Lab Instructions 3](#_Toc114427839)

[4 Lab Tasks 4](#_Toc114427840)

[4.1 Task 1 4](#_Toc114427841)

[4.1.1 Explain in words what the following MATLAB code will produce 4](#_Toc114427842)

[4.1.2 Consider the following definition of f; explain slicing operations on it 4](#_Toc114427843)

[4.1.3 Observe the result of the following assignments: 5](#_Toc114427844)

[4.2 Task 2 5](#_Toc114427845)

[4.2.1 Explain how the different values of cosine are stored in the vector h. 5](#_Toc114427846)

[4.2.2 What is h(1)? 5](#_Toc114427847)

[4.2.3 Is h(0) defined? 5](#_Toc114427848)

[4.3 Task 3 5](#_Toc114427849)

[4.3.1 Explain why it is necessary to write g(k+6). What happens if you use g(k) instead? 6](#_Toc114427850)

[4.3.2 Plotting in MATLAB 6](#_Toc114427851)

[4.3.3 Compare element wise .\* multiplication to matrix multiply. 6](#_Toc114427852)

[4.4 Task 4 7](#_Toc114427853)

[4.5 Task 5 7](#_Toc114427854)

[5 Conclusion 9](#_Toc114427855)

# Table of Figures

[Figure 4.3.2‑a: MATLAB Plot 6](#_Toc114350144)

[Figure 4.3.3‑a Plot of a Sinusoid 7](#_Toc114350145)

# Introduction to MATLAB

## Objectives

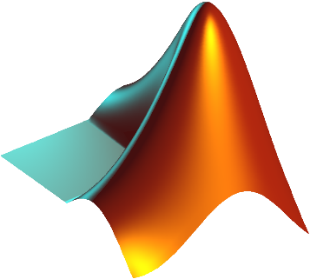
This Lab experiment has been designed to familiarize students with MATLAB and basic understanding of MATLAB commands

* How to use MATLAB interface
* Familiarization with Mathematical operators in MATLAB
* How to use MATLAB Help
* How to handle Matrices and Vectors in MATLAB
* How to make functions in MATLAB

## Equipment

Software

* *MATLAB*



## Lab Instructions

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

* Lab objectives
* MATLAB codes
* Results (Graphs/Tables) duly commented and discussed
* Conclusion

# Lab Tasks

## Task 1

### 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 ];

**Answer:** The above code will output new variables, namely, a, b, c, d, and e with values as defined by the colon notation. The colon notation’s general syntax is {start: step: stop}. If the step is not defined, it defaults to one. Following output is observed:

*a = 0 1 2 3 4 5 6*

*b = 2 6 10 14*

*c = 99 98 97 96 95 94 93 92 91 90 89 88*

*d = 2.0000 2.1111 2.2222 2.3333 2.4444 2.5556 2.6667 2.7778 2.8889 3.0000 3.1111 3.2222 3.3333 3.4444 3.5556 3.6667 3.7778 3.8889 4.0000*

The last line (assigning of e) is hidden because of the usage of the semicolon (which suppresses outputs when used as the end of a command/equation in command window)

### Consider the following definition of f; explain slicing operations on it

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

f(4:6)

size(f)

length(f)

f(2:2:length(f))

**Answer:** The first line instantiates a variable f as a row vector with indices occupying values as defined by simple commands such as zeros, linspace and ones. Following values are observed:

*0 0 0 0 0.2500 0.5000 0.7500 1.0000 1.0000 1.0000 1.0000 1.0000*

The second line slices the vector f from the index 4 to the index 6, meaning:

*0 0.2500 0.5000*

Third line displays the size of f as *row x columns*, hence, 1 x 2.

Fourth line displays the highest value of the size(f).

The last line displays the values of vector f from index 2 to the end of its length, while skipping an element in between.

*0 0 0.5000 1.0000 1.0000 1.0000*

### Observe the result of the following assignments:

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

**Answer:** The indices 4 to 6 of the vector g (which is a copy of f) are replaced with factors of .

*3.1416 6.2832 9.4248*

## 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.*

**Answer:** f(2:2:length(f)) = pi^pi will accomplish the desired task without the use of loops.

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.

**Answer:** Each element of (0:11), which is a simple row vector with default single steps, is multiplied with and divided by 4 before being passed to the cosine function to create a vector with shape the same as (0:11).

### What is h(1)?

**Answer:** h(1) is 1, as the first element of (0:11) is 0, which multiplied with anything yields zero. And the identity cos(0) = 1 results in the first element of h being 1.

### Is h(0) defined?

**Answer:** No; Indices in MATLAB start from 1, rather than 0 like in most programming languages, and will throw an error.

## 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. Rewrite this computation without using the loop:

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

for k=-5:5

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

end

g

**Answer:** g(1:11) = cos( ((-5:5) .\* pi) / 3)

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

**Answer:** g(k) will throw an error as k’s initial value starts from -5, and negative indices don’t exist in MATLAB. The value 6 comes from the fact that indices start from 1, rather than 0.

### 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



Figure ‑: MATLAB Plot

### Compare element wise .\* multiplication to matrix multiply.

**Answer:** Matrix multiply is the procedure of multiplying two matrices such that the resultant matrix shape is {B for B in AB \* BA where A and B represent the row x column of two matrices. The operator .\* on the other hand is element-wise multiplication, where each element of a matrix will multiply with the specified scalar, or an element with the same dimension as that of the element of the matrix to be multiplied.

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

## 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)’)



Figure ‑ Plot of a Sinusoid

Explain why the plot of real(y) is a sinusoid. What is its phase and amplitude?

**Answer:** In y = 1.4\*exp(j\*pi/2)\*exp(j\*5\*pi\*t), the initial two terms, 1.4\*exp(j\*pi/2), reduce to 1.4j and the latter term exp(j\*5\*pi\*t) expands to cos(5\*pi\*t) + j sin(5\*pi\*t).

Multiplication of the two results in y = 1.4j cos(5\*pi\*t) - 1.4sin(5\*pi\*t). The real part of which is:

Re(y) = - 1.4sin(5\*pi\*t) which belongs to the sinusoid family.

Re(y) can also be expressed as 1.4sin(5\*pi\*t + pi); from the identity sin(x + pi) = - sin(x).

Hence, the phase of Re(y) is pi and its amplitude is 1.4.

## 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.*

**Answer:** The implementation of sigadd is given below:

**Implementation:**

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

    % This function adds the indices of two inputs x1 and x2

    % Pad with zeros at the beginning if input index exceeds

    % the range of x1 / x2

    if n1(length(n1)) > length(x1)

        x1 = [*zeros*(1, n1(1) - 1), x1];

    end

    if n2(length(n2)) > length(x2)

        x2 = [*zeros*(1, n2(1) - 1), x2];

    end

    % If-Else conditions to pad and add proper indices

    if length(n2) > length(n1)

        x2\_pad = x2(1:n2(length(n2)));

        x1\_pad = [x1(1:n1(length(n1))), *zeros*(1, ...

            length(x2\_pad) - length(x1(1:n1(length(n1)))))];

    elseif length(n2) < length(n1)

        x1\_pad = x1(1:n1(length(n1)));

        x2\_pad = [x2(1:n2(length(n2))), *zeros*(1, ...

            length(x1\_pad) - length(x2(1:n2(length(n2)))))];

    else

        x1\_pad = x1;

        x2\_pad = x2;

    end

    % Post-proc, add the padded inputs

    y = x1\_pad + x2\_pad;

    n = length(y);

end

**Output:**

>> x1 = [1 2 3 4 5 6 7 8 9];

>> x2 = [2 4 6 8 10 12 14 16 18 20 22 24];

>> n1 = 3:11;

>> n2 = 1:12;

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

y = 4 7 10 13 16 19 22 25 28 31 24

n = 12

# Conclusion

In this lab, we have familiarized ourselves with MATLAB, standing for matrix laboratory. We implemented and solved different tasks related to vector indices, slicing, column operator as well as element-wise multiplication. We also observed that as a rule, MATLAB indices of any data and/or loop start at 1, rather than 0 as is usual in most programming languages. Lastly, we implemented a function sigadd() that adds the selected indices of two vectors by padding zeros to the larger slice, if an equal length was not established.