Namal University, Mianwali

Department of Electrical Engineering

EE 345 (L) – Digital Signal Processing (Lab)

Lab – 1

MATLAB Review: Basic Operations and Signals

Student Name	Student ID
Fahim Ur Rehman Shah	NIM-BSEE-2021-24

Instructor: Zulaikha Kiran

Lab Engineer: Faizan Ahmad

Version 1.0 – Muhammad Usman – 15/04/2021
Version 1.1 – Zulaikha Kiran – 05/04/2022
Version 1.2 – Husnain, Saiqa Dilawaiz – 06/04/2023
Version 1.3 – Faizan Ahmad – 26/02/2024

Introduction

The purpose of this lab is to enable the student's basic review of MATLAB operations and functions, and matrix operations using MATLAB.

Course Learning Outcomes

CLO1: Develop algorithms to perform signal processing techniques on digital signals using MATLAB and DSP Kit DSK6713

CLO3: Deliver a report/lab notes/presentation/viva, effectively communicating the design and analysis of the given problem

Equipment

- Software
    - MATLAB

Instructions

1. This is an individual lab. You will perform the tasks individually and submit a report.
2. Some of these tasks are for practice purposes only while others (marked as 'Exercise') have to be answered in the report.
3. When asked to display an image/ graph in the exercise, either save it as jpeg or take a screenshot, in order to insert it in the report.
4. The report should be submitted on the given template, including:
    a. Code (copy and pasted, NOT a screenshot)
    b. Output values (from command window, can be a screenshot)
    c. Output figure/graph (as instructed in 3)
    d. Explanation where required
5. The report should be properly formatted, with easy to read code and easy to see figures.
6. Plagiarism or any hint thereof will be dealt with strictly. Any incident where plagiarism is caught, both (or all) students involved will be given zero marks, regardless of who copied whom. Multiple such incidents will result in disciplinary action being taken.

Objectives:

- How to get familiar with the interface of MATLAB
- Basic arithmetic operators and commands
- MATLAB variables
- Matrix manipulation
- Signal Plotting

Tools used:

- MATLAB

Basic Arithmetic Operations:

Operations	Symbol	Example
Addition	+	3 + 22
Subtraction	-	54.4 – 16.5
Multiplication	*	3.14 * 6
Division	/	10/2

Variables Naming Rule:

Variable names can contain up to 63 characters.
Punctuation characters are not allowed, because many of them have special meanings in MATLAB.

MATLAB Special Variables

ans	Default variable name for results
pi	Value of π
inf	Infinity
NaN	Not a number e.g. 0/0
realmin	The smallest usable positive real number
realmax	The largest usable positive real number

Types of Variables

Type	Examples
Integer	1362,-5656
Real	12.33,-56.3
Complex	X=12.2 – 3.2j (j = sqrt(-1))

Complex numbers in MATLAB are represented in rectangular form.
To separate real & imaginary part
H = real(X)
K= imag(X)
Conversion between polar & rectangular
C1= 1-2i
Magnitude: mag_c1 = abs(C1)
Angle: angle_c1 = angle(C1)
Note that angle is in radians

MATLAB Matrices

MATLAB treats all variables as matrices. For our purposes a matrix can be thought of as an array, in fact, that is how it is stored.
Vectors are special forms of matrices and contain only one row OR one column.
Scalars are matrices with only one row AND one column.
A matrix can be created in MATLAB as follows (note the commas AND semicolons):
» matrix = [1 , 2 , 3 ; 4 , 5 ,6 ; 7 , 8 , 9]

Row Vector

A matrix with only one row is called a row vector. A row vector can be created in MATLAB as follows (note the commas):

```
» rowvec = [12 ,  14 , 63]
        rowvec =
              12    14    63
```

Row vector can also defined in a following way:

```
  rowvec = 2 : 2 : 10;
      rowvec =
           2    4    6    8    10
```

Column Vector

A matrix with only one column is called a column vector. A column vector can be created in MATLAB as follows (note the semicolons):

```
» colvec = [13 ; 45 ; -2]
   colvec =

         13
         45
         -2
```

Extracting a column vector

A column vector can be extracted from a matrix. As an example we create a matrix below:

```
» matrix=[1,2,3;4,5,6;7,8,9]
matrix =
1   2   3
4   5   6
7   8   9
```

Here we extract the column 2 of matrix and make a column vector: here colon denote all rows and 2 denote second column

```
» col_two =matrix(: , 2)
col_two =  2
           5
           8
```

Extracting a row vector

A row vector can be extracted from a matrix. As an example we create a matrix below:

```
» matrix=[1,2,3;4,5,6;7,8,9]

matrix =
   1   2   3
   4   5   6
   7   8   9
```

Here we extract row 2 of matrix and make a row vector.

here colon denote all column and 2 denote second row.

```
» rowvec =matrix(2, : )
  rowvec =
          4    5    6
```

```
  matrix(2:3,1:2)
   4   5
   7   8
```

Concatenation

New matrices may be formed out of old ones

Suppose we have:

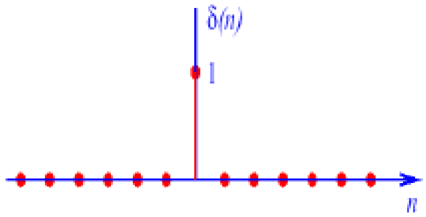
```
    a = [1 2; 3 4]
  a = [1 2 ;3 4]
```

Input	output
[a , a, a]	ans = 1 2 1 2 1 2 3 4 3 4 3 4

The Unit Delta (Impulse) function:

The unit sample, denoted by S(n), is defined by and plays the same role in discrete-time signal processing that the unit Impulse plays in continuous-time signal processing.

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

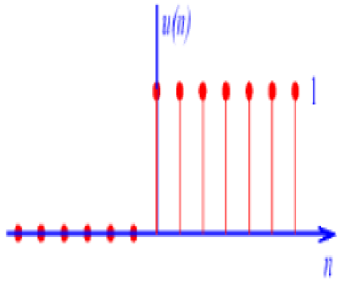



```
n=-5:5;
size1=size(n); %size of x-axis
impulse=zeros(size1);%generates a zero vector
index=find(n==0);%returns the index for which the condition is
true
impulse(index)=1;
stem(n,impulse)
axis([-6 6 0 1.2])
```

The Unit step function:

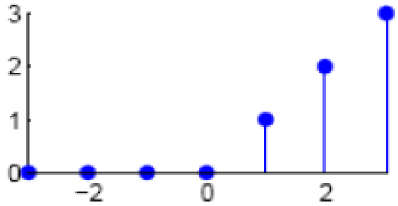
The unit step, denoted by u(n), is defined by

$$u(n) = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$$



The Unit Ramp Function:

$$r(n) = \begin{cases} n & 0 \leq n \leq \infty \\ 0 & \text{elsewhere} \end{cases}$$



Lab Exercise

Question 1.1

Perform the given commands

- Let a=[1 4 3;4 2 6 ;7 8 9]
- Determinant of matrix a. det(a) = ?
- Inverse of matrix a. inv(a) = ?
- Transpose of matrix a. a' = ?
- A row vector containing the minimum element from each column of a. min(a) = ?
- The smallest element in matrix a. min(min(a)) = ?
- A row vector containing the maximum element from each column of a. max(a) = ?
- The max element from matrix a. max(max(a)) = ?
- Bitwise calculate the square of each element of matrix a. a.^2 = ?
- Treats the columns of 'a' as vectors, returning a row vector of the sums of each column. sum (a) = ?
- Sum of all the elements in the matrix. sum(sum(a)) = ?
- Give the number or rows and the number of columns of matrix a. size (a) = ?
- Let a =[4 5 6] , find the number of elements in row vector. length(a) = ?
- Explain a(1:2,1:2), a(2:1) and a(3).

Question 1.2

1. Generate a 6x6 matrix A
2. Generate a 6x1 matrix z
3. Solve linear system of equations Ax=z
4. Compute determinant of matrix A
5. Extract a 4x4 matrix from the 6x6 matrix and 4x1 matrixes from 6x1

Question 1.3

1) $\frac{35.7*(64-7^4)}{45+5^3}$

2) $\frac{3^7\log(76)}{7^3+564} + \sqrt[3]{910}$

3) Solve

$$\cos^2\left(\frac{5\pi}{6}\right)\sin\left(\frac{7\pi}{8}\right)^2 + \frac{\tan\left(\frac{\pi}{6}\ln 8\right)}{\sqrt{7}}$$

4) Create the matrix shown below by using the vector notation for creating vectors with constant spacing and/or the linspace command when entering the rows.

$$B = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 & 25 \\ 72 & 66 & 60 & 54 & 48 & 42 & 36 & 30 & 24 \\ 0 & 0.125 & 0.250 & 0.375 & 0.500 & 0.625 & 0.750 & 0.875 & 1.000 \end{bmatrix}$$

Question 1.4

1) Write a MATLAB code that generates Delta (Impulse) Function.

- Use the plotting function stem to make the graphs
- Replace the stem command in the above code with the plot command and run the code again. How does this change the plot? And why?

2) Write a MATLAB code that generate Unit Step function:

- Use the plotting function stem to make the graphs
- Replace the stem command in the above code with the plot command and run the code again. How does this change the plot?

3) Write a MATLAB code that generate Unit Ramp function

- Use the plotting function plot to make the graphs
- Replace the plot command in the above code with the stem command and run the code again. How does this change the plot?

Question 1.1

```
% Create the matrix a
a = [1 4 3; 4 2 6; 7 8 9]; % Creates a 3x3 matrix a

% Calculate determinant, inverse, and transpose
determinant_a = det(a); % Calculates determinant and stores in determinant_a
inverse_a = inv(a);    % Calculates inverse and stores in inverse_a
transpose_a = a';     % Calculates transpose and stores in transpose_a

% Find row vectors of minimum and maximum elements, and smallest and largest
elements
min_elements = min(a);         % Finds minimum in each column, stores in
min_elements
max_elements = max(a);         % Finds maximum in each column, stores in
max_elements
smallest_element = min(min(a));  % Finds smallest element overall, stores in
smallest_element
largest_element = max(max(a));  % Finds largest element overall, stores in
largest_element

% Calculate element-wise square, sum of each column, and sum of all elements
squared_a = a.^2;        % Performs element-wise square, stores in squared_a
column_sums = sum(a);     % Calculates sum of each column, stores in
column_sums
total_sum = sum(sum(a));  % Calculates sum of all elements, stores in total_sum

% Get matrix dimensions
[rows, columns] = size(a); % Gets number of rows and columns, stores in rows
and columns

% Create a row vector
b = [4 5 6];

% Display results
disp('Determinant of a:');
```

```
Determinant of a:
```

```
disp(determinant_a);
```

```
   48.0000
```

```
disp('Inverse of a:');
```

```
Inverse of a:
```

```
disp(inverse_a);
```

```
   -0.6250   -0.2500    0.3750
```

```
    0.1250   -0.2500    0.1250
    0.3750    0.4167   -0.2917
```

```
disp('Transpose of a:');
```

```
Transpose of a:
```

```
disp(transpose_a);
```

```
     1     4     7
     4     2     8
     3     6     9
```

```
disp('Minimum elements in each column:');
```

```
Minimum elements in each column:
```

```
disp(min_elements);
```

```
     1     2     3
```

```
disp('Smallest element in the matrix:');
```

```
Smallest element in the matrix:
```

```
disp(smallest_element);
```

```
     1
```

```
disp('Maximum elements in each column:');
```

```
Maximum elements in each column:
```

```
disp(max_elements);
```

```
     7     8     9
```

```
disp('Largest element in the matrix:');
```

```
Largest element in the matrix:
```

```
disp(largest_element);
```

```
     9
```

```
disp('Element-wise square of a:');
```

```
Element-wise square of a:
```

```
disp(squared_a);
```

```
     1    16     9
    16     4    36
    49    64    81
```

```
disp('Sum of each column:');
```

```
Sum of each column:
```

```
disp(column_sums);
```

```
    12    14    18
```

```
disp('Sum of all elements:');
```

```
Sum of all elements:
```

```
disp(total_sum);
```

```
    44
```

```
disp('Number of rows:');
```

```
Number of rows:
```

```
disp(rows);
```

```
     3
```

```
disp('Number of columns:');
```

```
Number of columns:
```

```
disp(columns);
```

```
     3
```

```
% Explain sub-matrix operations
disp('Explanation of sub-matrix operations:');
```

```
Explanation of sub-matrix operations:
```

```
a(1:2, 1:2)
```

```
ans = 2×2
     1     4
     4     2
```

```
disp('- a(1:2, 1:2): Selects elements from rows 1 and 2, and columns 1 and 2.');
```

```
- a(1:2, 1:2): Selects elements from rows 1 and 2, and columns 1 and 2.
```

```
a(1:2)
```

```
ans = 1×2
     1     4
```

```
disp('- a(1:2): Selects rows the range of element starting from 1st row 1st column 1st element to 2nd element');
```

```
- a(1:2): Selects rows the range of element starting from 1st row 1st column 1st element to 2nd element
```

```
a(3) % This line remains the same
```

```
ans = 7
```

```
disp('- a(3): Selects the third element (which is the entire third row in this case).');
```

```
- a(3): Selects the third element (which is the entire third row in this case).
```

Question 1.2

```
% Generate a 6x6 matrix A
A = rand(6); % Creates a 6x6 matrix with random elements

% Generate a 6x1 matrix z
z = rand(6, 1); % Creates a 6x1 matrix with random elements

% Solve linear system of equations Ax = z
x = A \ z; % Solves Ax = z and stores the solution in x

% Compute determinant of matrix A
det_A = det(A); % Calculates the determinant of A and stores it in det_A

% Extract a 4x4 matrix from the 6x6 matrix
A_sub = A(1:4, 1:4); % Extracts a 4x4 sub-matrix from the top-left corner

% Extract a 4x1 matrix from the 6x1 matrix
z_sub = z(1:4); % Extracts a 4x1 sub-matrix from the top 4 elements

% Display results
disp('Solution to Ax = z:');
```

```
Solution to Ax = z:
```

```
disp(x);
```

```
   -0.8261
    0.8344
   -0.0316
    0.0396
   -1.6430
    1.9476
```

```
disp('Determinant of A:');
```

```
Determinant of A:
```

```
disp(det_A);
```

```
   -0.0208
```

```
disp('Extracted 4x4 sub-matrix:');
```

```
Extracted 4x4 sub-matrix:
```

```
disp(A_sub);
```

```
    0.5752    0.0430    0.5470    0.3685
    0.0598    0.1690    0.2963    0.6256
    0.2348    0.6491    0.7447    0.7802
    0.3532    0.7317    0.1890    0.0811
```

```
disp('Extracted 4x1 sub-matrix:');
```

```
Extracted 4x1 sub-matrix:
```

```
disp(z_sub);
```

```
    0.3507
    0.9390
    0.8759
    0.5502
```

```
% Explanation of sub-matrix operations
disp('Explanation of sub-matrix operations:');
```

```
Explanation of sub-matrix operations:
```

```
disp('- A(1:4, 1:4): Selects elements from rows 1 to 4 and columns 1 to 4.');
```

```
- A(1:4, 1:4): Selects elements from rows 1 to 4 and columns 1 to 4.
```

```
disp('- z(1:4): Selects elements from rows 1 to 4.');
```

```
- z(1:4): Selects elements from rows 1 to 4.
```

Question 1.3

```
% Question 1.3

% 1) Calculating results
numerator1 = 35.7 * (64 - 7^4);
denominator1 = 45 + 5^3;
result1 = numerator1 / denominator1;

% 2) Calculating results
numerator2 = 3^7 * log(76);
denominator2 = 7^3 + 564;
result2 = (numerator2 / denominator2)+(nthroot(910,3));

% 3) Calculating results
angle = 5*pi/6;
result3 = cos(angle)^2 * sin(7*pi/8)^2 + tan(pi/6 * log(8)) / sqrt(7);

% 4) Creating matrix B with equal spacing in all rows
% Define starting and ending points for each row
start_values = [1, 72, 0];
end_values = [25, 24, 1];
num_elements = 9;

% Create row vectors using linspace
rows = [];
for i = 1:3
    rows = [rows; linspace(start_values(i), end_values(i), num_elements)];
end
% Combine rows into the matrix
B = rows;
% Display results
fprintf('1) %.4f\n', result1);
```

```
1) -490.7700
```

```
fprintf('2) %.4f\n', result2);
```

```
2) 20.1330
```

```
fprintf('3) %.4f\n', result3);
```

```
3) 1.4395
```

```
disp('4) ');
```

```
4)
```

```
disp(B); % Display the matrix
```

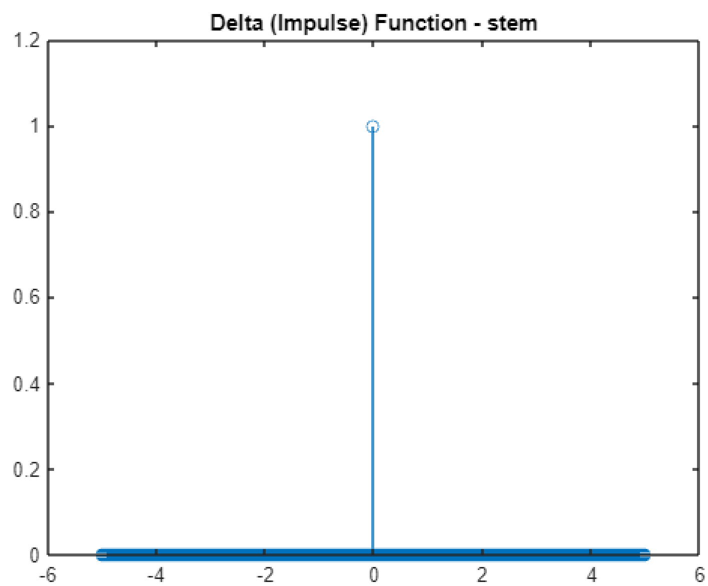
```
    1.0000    4.0000    7.0000   10.0000   13.0000   16.0000   19.0000   22.0000   25.0000
   72.0000   66.0000   60.0000   54.0000   48.0000   42.0000   36.0000   30.0000   24.0000
         0    0.1250    0.2500    0.3750    0.5000    0.6250    0.7500    0.8750    1.0000
```

Question 1.4

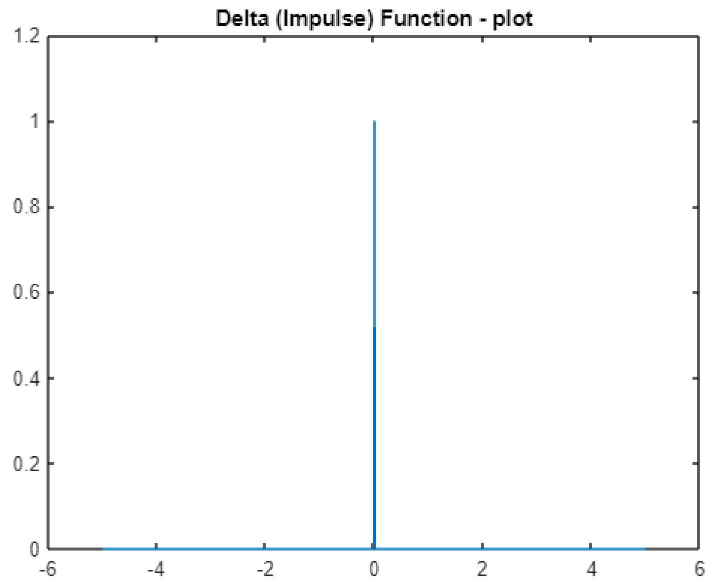
1) Write a MATLAB code that generates Delta (Impulse) Function.

- **Use the plotting function stem to make the graphs**
- **Replace the stem command in the above code with the plot command and run the code again. How does this change the plot? And why?**

```
t = -5:0.01:5;  % Time vector
impulse = t == 0;   % Impulse function (non-zero only at t=0)
% Plot with stem
figure;
stem(t, impulse);
axis([-6 6 0 1.2])
title('Delta (Impulse) Function - stem');
```




```
% Plot with plot
figure;
plot(t, impulse);
axis([-6 6 0 1.2])
title('Delta (Impulse) Function - plot');
```

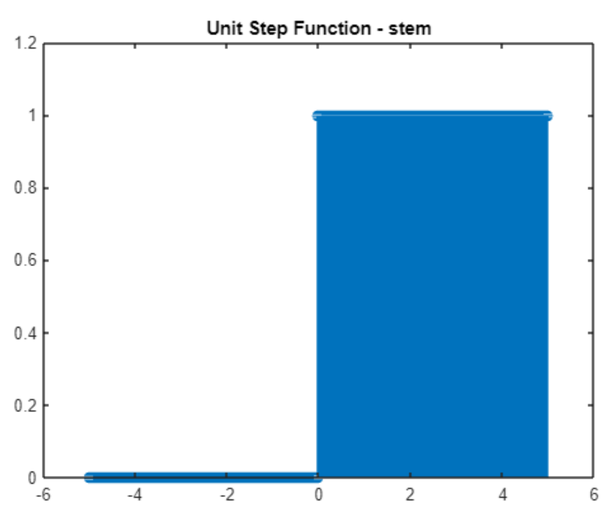


2) Write a MATLAB code that generate Unit Step function:

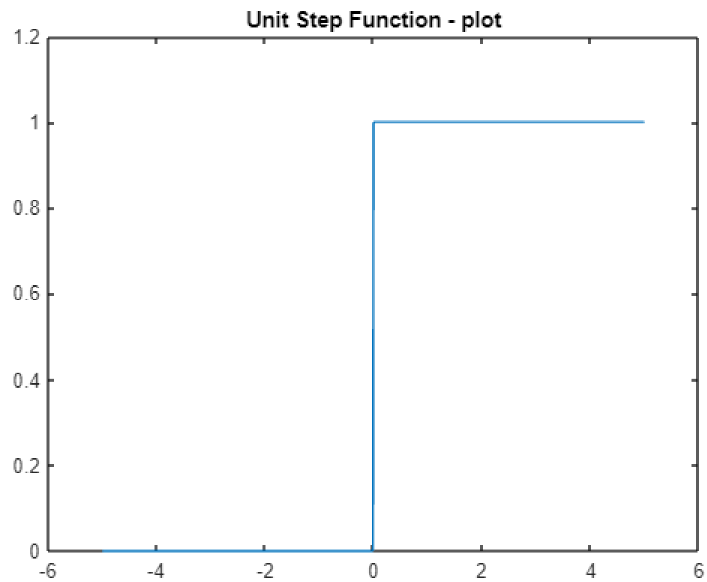
- **Use the plotting function stem to make the graphs**
- **Replace the stem command in the above code with the plot command and run the code again. How does this change the plot?**

```
t = -5:0.01:5;
step = t >= 0;   % Unit step function

% Plot with stem
figure;
stem(t, step);
axis([-6 6 0 1.2])
title('Unit Step Function - stem');
```



```
% Plot with plot
figure;
plot(t, step);
axis([-6 6 0 1.2])
title('Unit Step Function - plot');
```

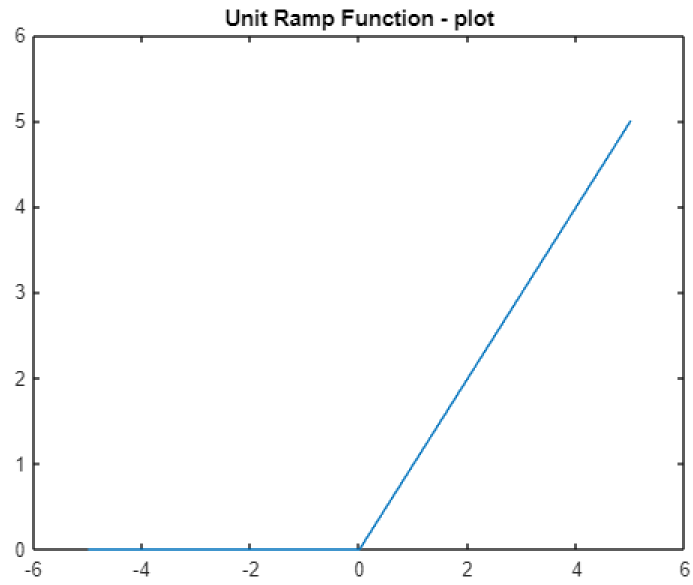


3) Write a MATLAB code that generate Unit Ramp function

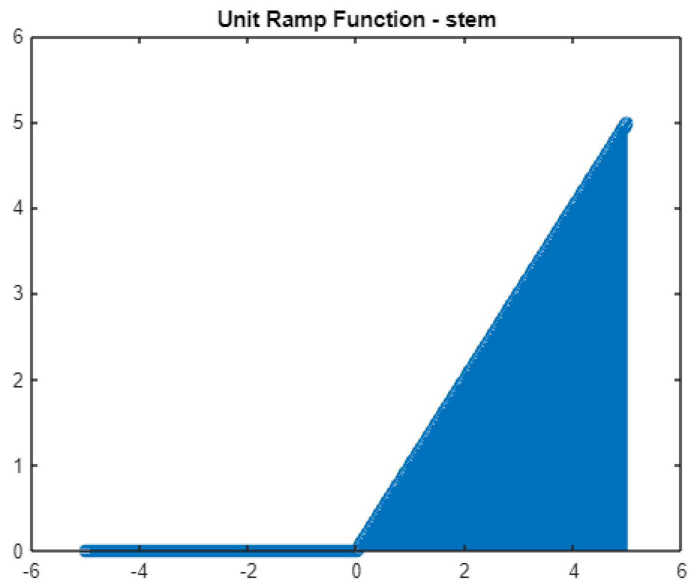
- **Use the plotting function plot to make the graphs**
- **Replace the plot command in the above code with the stem command and run the code again. How does this change the plot?**

```
t = -5:0.01:5;
ramp = t .* (t >= 0);  % Unit ramp function

% Plot with plot
figure;
plot(t, ramp);
axis([-6 6 0 6])
title('Unit Ramp Function - plot');
```




```
% Plot with stem
figure;
stem(t, ramp);
axis([-6 6 0 6])
title('Unit Ramp Function - stem');
```



Conclusion:

This MATLAB lab covered various functionalities. We explored creating, manipulating, and analyzing matrices using built-in functions and vector notation. We learned how to solve linear systems of equations and extract specific sections of matrices. Additionally, the lab covered performing mathematical computations with numerical values, functions, and roots. We also generated matrices with consistent spacing within rows or columns and visualized different functions using both stem and plot commands, observing their distinct graphical representations. Overall, this lab provided a hands-on experience with essential MATLAB functionalities for various applications in engineering, science, and mathematics.

Evaluation Rubric

- **Method of Evaluation**: In-lab marking by instructors, Report submitted by students
- **Measured Learning Outcomes**:

CLO1: Develop algorithms to perform signal processing techniques on digital signals using MATLAB and DSP Kit DSK6713

CLO3: Deliver a report/lab notes/presentation/viva, effectively communicating the design and analysis of the given problem

	Excellent 10	Good 9-7	Satisfactory 6-4	Unsatisfactory 3-1	Poor 0	Marks Obtained
Tasks (CLO1-P3)	All tasks completed correctly. Correct code with proper comments.	Most tasks completed correctly.	Some tasks completed correctly.	Most tasks incomplete or incorrect.	All tasks incomplete or incorrect.	
Output (CLO1-P3)	Output correctly shown with all Figures/Plots displayed as required and properly labelled	Most Output/Figures/Plots displayed with proper labels	Some Output/Figures/Plots displayed with proper labels OR Most Output/Figures/Plots displayed but without proper labels	Most of the required Output/Figures/Plots not displayed	Output/Figures/Plots not displayed	
Answers (CLO1-P3)	Meaningful answers to all questions. Answers show the understanding of the student.	Meaningful answers to most questions.	Some correct/ meaningful answers with some irrelevant ones	Answers not understandable/ not relevant to questions	Does not write any Answer	
Report (CLO3-A3)	Report submitted with proper grammar and punctuation with proper conclusions drawn and good formatting	Report submitted with proper conclusions drawn with good formatting but some grammar mistakes OR proper grammar but not very good formatting	Some correct/ meaningful conclusions. Some parts of the document not properly formatted or some grammar mistakes	Conclusions not based on results. Bad formatting with no proper grammar/punctuation	Report not submitted	
Total						