DRONACHARYA
Group of Institutions

SIGNAL AND SYSTEM

LABORATORY MANUAL

B.Tech. Semester

Subject Code: BEC-453

Session: 2024-25, Even Semester

Name:	
Roll. No.:	
Group/Branch:	

DRONACHARYA GROUP OF INSTITUTIONS
DEPARTMENT OF ECZ
#27 KNOWLEDGE PARK 3

GREATER NOIDA

AFFILATED TO Dr. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

Table of Contents

Vision and Mission of the Institute

Vision:

"Instilling core human values and facilitating competence to address global challenges by providing Quality Technical Education."

Mission:

M1 - Enhancing technical expertise through innovative research and education, fostering creativity and excellence in problem-solving.

M2 - Cultivating a culture of ethical innovation and user-focused design, ensuring technological progress enhances the well-being of society.

M3 - Equipping individuals with the technical skills and ethical values to lead and innovate responsibly in an ever-evolving digital landscape

.

Vision and Mission of the Department

Vision:

"To achieve excellence in Electronics and Computer engineering through quality education, research contributing to the emerging technologies and innovation to serve industry and society"

Mission:

M1: To help students achieve their goals by recognizing, identifying, and to bring up their unique strengths through quality education and cutting-edge research training..

M2: To facilitate adequate exposure to the students through training in the state of-the art technologies.

M3: To imbibe ability in the students to solve real life problems as per need of the society through nurturing their skills, creative thinking, and research acumen.

Programme Educational Objectives (PEOs)

PEO1: To develop a strong foundation of engineering fundamentals to build successful careers maintaining high ethical standards.

PEO2: To prepare graduates for higher studies and research activities, facilitating a commitment to lifelong learning.

PEO3: Impart strong profession, ethical, social responsibility, integrity with environmental sensitivity.

Programme Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1: To analyses electronics systems applying principles of mathematics and engineering sciences, to develop innovative ethical solutions to complex engineering problems with team spirit and social commitment.

PSO2: To develop solution for real world problems based on principles of computer hardware, advanced software and simulation tools with a focus to devise indigenous, eco-friendly and energy efficient projects.

University Syllabus

1**. Introduction to MATLAB**
a. To define and use variables and functions in MATLAB.
b. To define and use Vectors and Matrices in MATLAB.
c. To study various MATLAB arithmetic operators and mathematical functions.
d. To create and use m-files.
2**. Basic plotting of signals**
a. To study various MATLAB commands for creating two and three dimensional plots.
b. Write a MATLAB program to plot the following continuous time and discrete time signals.
i. Step Function
ii. Impulse Function
iii. Exponential Function
iv. Ramp Function
v. Sine Function
3. **Time and Amplitude transformations**
Write a MATLAB program to perform amplitude-scaling, time-scaling and time shifting on a given signal.
4. **Convolution of given signals**
Write a MATLAB program to obtain linear convolution of the given sequences.
5. **Autocorrelation and Cross-correlation**
a. Write a MATLAB program to compute autocorrelation of a sequence x(n) and verify the property.
b. Write a MATLAB program to compute cross-correlation of sequences x(n) and y(n) and verify the property.
6. **Fourier Series and Gibbs Phenomenon**
a. To calculate Fourier series coefficients associated with Square Wave.
b. To Sum the first 10 terms and plot the Fourier series as a function of time.
c. To Sum the first 50 terms and plot the Fourier series as a function of time.
7. **Calculating transforms using MATLAB**
a. Calculate and plot Fourier transform of a given signal.
b. Calculate and plot Z-transform of a given signal.
8. **Impulse response and Step response of a given system**
a. Write a MATLAB program to find the impulse response and step response of a system form its difference equation.
b. Compute and plot the response of a given system to a given input.
9. **Pole-zero diagram and bode diagram**
a. Write a MATLAB program to find pole-zero diagram, bode diagram of a given system from the given system function.
b. Write a MATLAB program to find, bode diagram of a given system from the given system function.
10. **Frequency response of a system**
Write a MATLAB program to plot magnitude and phase response of a given system.
11. **Checking linearity/non-linearity of a system using SIMULINK**
a. Build a system that amplifies a sine wave by a factor of two.
b. Test the linearity of this system using SIMULINK.

Course Outcomes (COs)

Upon successful completion of the course, the students will be able to:

CO 1	Understand the basics operation of MATLAB
CO 2	Analysis the time domain and frequency domain signals.
CO 3	Implement the concept of Fourier series and Fourier transforms.
CO 4	Find the stability of system using pole-zero diagrams and bode diagram.
CO 5	Design frequency response of the system.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	3	-	-	3	2	-	-	-	1	1	-	2
CO 2	3	2	1	-	-	-	-	-	2	2	-	2
CO 3	2	2	2	-	2	-	-	-	1	1	-	1
CO 4	2	1	2	2	1	-	-	-	1	1	-	2
CO 5	1	2	3	2	1	-	-	-	1	1	-	2
Course Correlation mapping	**2.2**	**1.4**	**1.6**	**1.6**	**1.2**	**-**	**-**	**-**	**1.2**	**1.2**	**-**	**1.8**

CO-PSO Mapping

	PSO1	PSO2	PSO3
CO 1	2	3	2
CO 2	2	3	2
CO 3	2	3	2
CO 4	1	3	2
CO 5	2	3	2

Course Overview

This lab is designed to give you a quick way to become familiar with the MATLAB software by introducing you the basic features, commands, and functions. In this lab, you will discover that entering and solving complex numbers in MATLAB is as easy as entering and solving real numbers, especially with the help of MATLAB built-in complex functions. The lab is intended to be very interactive. You should have the required software running while you are reading the pages, and you should perform along with the examples. Upon completion, you should know how to start MATLAB, how to get HELP, how to assig variables in MATLAB and to perform the typical complex numbers operations.

List of Experiments mapped with COs

Si No.	Name of the Experiment	Course Outcome
1	Introduction to MATLAB a. To define and use variables and functions in MATLAB. b. To define and use Vectors and Matrices in MATLAB. c. To study various MATLAB arithmetic operators and mathematical functions. d. To create and use m-files.	CO 1
2	Basic plotting of signals a. To study various MATLAB commands for creating two and three dimensional plots. b. Write a MATLAB program to plot the following continuous time and discrete time signals. i. Step Function ii. Impulse Function iii. Exponential Function iv. Ramp Function v. Sine Function	CO 1
3	Time and Amplitude transformations Write a MATLAB program to perform amplitude-scaling, time-scaling and time shifting on a given signal.	CO 2
4	Convolution of given signals. Write a MATLAB program to obtain linear convolution of the given sequences.	CO 2
5	Autocorrelation and Cross-correlation a. Write a MATLAB program to compute autocorrelation of a sequence x(n) and verify the property. b. Write a MATLAB program to compute cross-correlation of sequences x(n) and y(n) and verify the property.	CO 3
6	Fourier Series and Gibbs Phenomenon a. To calculate Fourier series coefficients associated with Square Wave. b. To Sum the first 10 terms and plot the Fourier series as a function of time. c. To Sum the first 50 terms and plot the Fourier series as a function of time.	CO 2
7	Calculating transforms using MATLAB a. Calculate and plot Fourier transform of a given signal. b. Calculate and plot Z-transform of a given signal.	CO 4
8	Impulse response and Step response of a given system a. Write a MATLAB program to find the impulse response and step response of a system form its difference equation. b. Compute and plot the response of a given system to a given input.	CO 2
9	Pole-zero diagram and bode diagram a. Write a MATLAB program to find pole-zero diagram, bode diagram of a given system from the given system function. b. Write a MATLAB program to find, bode diagram of a given system from the given system function.	CO 3
10	Checking Linearity/Non-Linearity of a system using SIMULINK.	CO 2

DOs and DON’Ts

DOs

1. Login-on with your username and password.
2. Log off the computer every time when you leave the Lab.
3. Arrange your chair properly when you are leaving the lab.
4. Put your bags in the designated area.
5. Ask permission to print.

DON’Ts

1. Do not share your username and password.
2. Do not remove or disconnect cables or hardware parts.
3. Do not personalize the computer setting.
4. Do not run programs that continue to execute after you log off.
5. Do not download or install any programs, games or music on computer in Lab.
6. Personal Internet use chat room for Instant Messaging (IM) and Sites is strictly prohibited.
7. No Internet gaming activities allowed.
8. Tea, Coffee, Water & Eatables are not allowed in the Computer Lab.

General Safety Precautions

Precautions (In case of Injury or Electric Shock)

1. To break the victim with live electric source, use an insulator such as fire wood or plastic to break the contact. Do not touch the victim with bare hands to avoid the risk of electrifying yourself.
2. Unplug the risk of faulty equipment. If main circuit breaker is accessible, turn the circuit off.
3. If the victim is unconscious, start resuscitation immediately, use your hands to press the chest in and out to continue breathing function. Use mouth-to-mouth resuscitation if necessary.
4. Immediately call medical emergency and security. Remember! Time is critical; be best.

Precautions (In case of Fire)

1. Turn the equipment off. If power switch is not immediately accessible, take plug off.
2. If fire continues, try to curb the fire, if possible, by using the fire extinguisher or by covering it with a heavy cloth if possible isolate the burning equipment from the other surrounding equipment.
3. Sound the fire alarm by activating the nearest alarm switch located in the hallway.
4. Call security and emergency department immediately:

Emergency : 201 (Reception)

Security : 231 (Gate No.1)

Guidelines to students for report preparation

All students are required to maintain a record of the experiments conducted by them. Guidelines for its preparation are as follows: -

1) All files must contain a title page followed by an index page. ***The files will not be signed by the faculty without an entry in the index page.***

2) Student's Name, Roll number and date of conduction of experiment must be written on all pages.

3) For each experiment, the record must contain the following

(i) Aim/Objective of the experiment
(ii) Pre-experiment work (as given by the faculty)
(iii) Lab assignment questions and their solutions
(iv) Test Cases (if applicable to the course)
(v) Results/ output

Note:

1. Students must bring their lab record along with them whenever they come for the lab.

2. Students must ensure that their lab record is regularly evaluated.

Lab Assessment Criteria

An estimated 10 lab classes are conducted in a semester for each lab course. These lab classes are assessed continuously. Each lab experiment is evaluated based on 5 assessment criteria as shown in following table. Assessed performance in each experiment is used to compute CO attainment as well as internal marks in the lab course.

Grading Criteria	**Exemplary (4)**	**Competent (3)**	**Needs Improvement (2)**	**Poor (1)**
AC1: Designing experiments	The student chooses the problems to explore.	The student chooses the problems but does not set an appropriate goal for how to explore them.	The student fails to define the problem adequately.	The student does not identify the problem.
AC2: Collecting data through observation and/or experimentation	Develops a clear procedure for investigating the problem	Observations are completed with necessary theoretical calculations and proper identification of required components.	Observations are completed with necessary theoretical calculations but without proper understanding. Obtain the correct values for only a few components after calculations. Followed the given experimental procedures but obtained results with some errors.	Observations are incomplete. Lacks the appropriate knowledge of the lab procedures.
AC3: Interpreting data	Decides what data and observations are to be collected and verified	Can decide what data and observations are to be collected but lacks the knowledge to verify	Student decides what data to gather but not sufficient	Student has no knowledge of what data and observations are to be collected
AC4: Drawing conclusions	Interprets and analyses the data in order to propose viable conclusions and solutions	Incomplete analysis of data hence the quality of conclusions drawn is not up to the mark	Cannot analyse the data or observations for any kind of conclusions.	Lacks the required knowledge to propose viable conclusions and solutions
AC5: Lab record assessment	Well-organized and confident presentation of record & ability to correlate the theoretical concepts with the concerned lab results with appropriate reasons.	Presentation of record is acceptable	Presentation of record lacks clarity and organization	No efforts were exhibited

LAB EXPERIMENTS

LAB EXPERIMENT 1

OBJECT:

Introduction to MATLAB.

(i) To define and use variables and functions in MATLAB.
(ii) To define and use vectors, matrices and polynomials in MATLAB.
(iii) To study various MATLAB arithmetic operators and mathematic function.
(iv) To create and use m-files.

SOFTWARE USED: MATLAB 7.9

THEORY: MATLAB is a programming language developed by Math Works. It started out as a matrix programming language where linear algebra programming was simple. MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming.

MATLAB is developed by Math Works.

It allows matrix manipulations; plotting of functions and data; implementation of algorithms; creation of user interfaces; interfacing with programs written in other languages, including C, C++, Java, and FORTRAN; analyze data; develop algorithms; and create models and applications.

It has numerous built-in commands and math functions that help you in mathematical calculations, generating plots, and performing numerical methods.

In MATLAB environment, every variable is an array or matrix.
Variables can be defined in MATLAB in the following ways:

```
x = 3          % defining x and initializing it with a value
x = sqrt(16)        % defining x and initializing it with an expression
```

The clear command deletes all (or the specified) variable(s) from the memory.

```
clear x     % it will delete x, won't display anything
clear          % it will delete all variables in the workspace
          % peacefully and unobtrusively
```

A vector is a one-dimensional array of numbers. MATLAB allows creating two types of vectors:

- Row vectors
- Column vectors

Row vectors are created by enclosing the set of elements in square brackets, using space or comma to delimit the elements.
For example,

```
r = [7 8 9 10 11]
```

Column vectors are created by enclosing the set of elements in square brackets, using semicolon(;) to delimit the elements.

```
c = [7; 8; 9; 10; 11]
```

Creating Matrices

A matrix is a two-dimensional array of numbers.

In MATLAB, a matrix is created by entering each row as a sequence of space or comma separated elements, and end of a row is demarcated by a semicolon. For example a 3-by-3 matrix is created as:

```
m = [1 2 3; 4 5 6; 7 8 9]
```

MATLAB ENVIRONMENT:

- Command Window
- Command History
- Workspace
- Current Directory
- Figure Window

Command Window:
Whenever MATLAB is invoked, the main window called command window is activated. The command window displays the command prompt'>>' and a cursor where commands are entered and are executed instantaneously.

Command History Window:
Command history window consists of a list of all the commands that are entered at the command window. These commands remain in the list until they are deleted. Any command may be executed by selecting and double clicking it with the mouse.

Workspace:
A workspace is a collection of all the variables that have been generated so far in the current MATLAB session and shows their data type and size. All the commands executed from Command Window and all the script files executed from the Command Window share common workspace, so they can share all the variables.

Current Directory:
The Current Directory window contains all the files and folders present in the Current Directory. To run any file, it must either be in the Current Directory or on the search path.

Edit Window:
An Edit Window is used to create a new program file, or to modify existing files. In this window, programs can be written, edited and saved. The programs written using the MATLAB editor are automatically assigned an extension (.m) by the editor and are known as M- files.

Figure Window:
A Figure Window is a separate window with default white background and is used to display MATLAB graphics. The results of all the graphic commands executed are displayed in the figure window.
MATLAB allows two different types of arithmetic operations −

- Matrix arithmetic operations
- Array arithmetic operations

Matrix arithmetic operations are same as defined in linear algebra. Array operations are executed element by element, both on one dimensional and multi-dimensional array.
The matrix operators and arrays operators are differentiated by the period (.) symbol. However, as the addition and subtraction operation is same for matrices and arrays, the operator is same for both cases.
The following table gives brief description of the operators −

Operator	**Description**
+	Addition or unary plus. A+B adds the values stored in variables A and B. A and B must have the same size, unless one is a scalar. A scalar can be added to a matrix of any size.
-	Subtraction or unary minus. A-B subtracts the value of B from A. A and B must have the same size, unless one is a scalar. A scalar can be subtracted from a matrix of any size.
*	Matrix multiplication. C = A*B is the linear algebraic product of the matrices A and B. More precisely, $C(i,j) = \sum_{k=1}^{n} A(i,k)B(k,j)$
/	Slash or matrix right division. B/A is roughly the same as B*inv(A). More precisely, B/A = (A'\B')'.
./	Array right division. A./B is the matrix with elements A(i,j)/B(i,j). A and B must have the same size, unless one of them is a scalar.
\	Backslash or matrix left division. If A is a square matrix, A\B is roughly the same as inv (A)*B, except it is computed in a different way. If A is an n-by-n matrix and B is a column vector with n components, or a matrix with several such columns, then X = A\B is the solution to the equation *AX* = *B*. A warning messageis displayed if A is badly scaled or nearly singular.
.\	Array left division. A.\B is the matrix with elements B(i,j)/A(i,j). A and B must have the same size, unless one of them is a scalar.
.^	Array power. A. ^B is the matrix with elements A (i,j) to the B(i,j) power. A andB must have the same size, unless one of them is a scalar.

^	Matrix power. X^p is X to the power p, if p is a scalar. If p is an integer, the power is computed by repeated squaring. If the integer is negative, X is inverted first. For other values of p, the calculation involves eigenvalues and eigenvectors, such that if [V,D] = eig(X), then X^p = V*D.^p/V.
'	Matrix transpose. A' is the linear algebraic transpose of A. For complex matrices, this is the complex conjugate transpose.
.'	Array transpose. A' is the array transpose of A. For complex matrices, this does not involve conjugation.

Functions for Arithmetic Operations:-

Apart from the above-mentioned arithmetic operators, MATLAB provides the following commands/functions used for similar purpose −

Function	Description
uplus(a)	Unary plus; increments by the amount a
plus (a,b)	Plus; returns a + b
uminus(a)	Unary minus; decrements by the amount a
minus(a, b)	Minus; returns a - b
times(a, b)	Array multiply; returns a.*b
rdivide(a, b)	Right array division; returns a ./ b
ldivide(a, b)	Left array division; returns a.\ b
mrdivide(A, B)	Solve systems of linear equations $xA = B$ for x
mldivide(A, B)	Solve systems of linear equations $Ax = B$ for x
power(a, b)	Array power; returns a.^b
mpower(a, b)	Matrix power; returns a ^ b

PROGRAM:-

```
a = 10;
b = 20;
c = a + b
d = a - b
e = a * b
f = a / b
g = a \ b
x = 7;
y = 3;
z = x ^ y
```

The M Files

MATLAB allows writing two kinds of program files −

- **Scripts** − script files are program files with **.m extension**. In these files, you write series of commands, which you want to execute together. Scripts do not accept inputs and do not return any outputs. They operate on data in the workspace.

- **Functions** − functions files are also program files with **.m extension**. Functions can accept inputs and return outputs. Internal variables are local to the function.

You can use the MATLAB editor or any other text editor to create your **.m** files.

RESULTS:- All the operations have been performed a using MATLAB and results have been seen on the command window.

RELATED QUESTIONS:-

1. How to create matrix in MATLAB?
2. Name some electronics engineering fields where MATLAB can be used.
3. What is the function of MATLAB function clc?
4. What happens if we put a semicolon after a statement in a MATLAB program?
5. What are M –files?

LAB EXPERIMENT 2

OBJECT:-
i) Plot the basic signals (Impulse, Step function and Ramp function)
ii) To create 2-D and 3-D plots.

SOFTWARE USED:- MATLAB 7.9

PROCEDURE:-

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

THEORY:-

Impulse Function

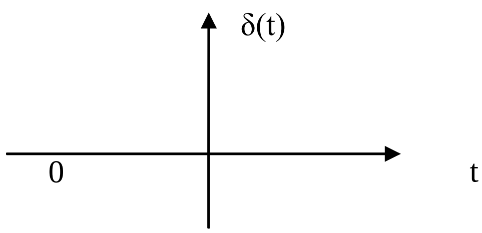
The impulse function is defined as

$\int_{-\infty}^{\infty} \delta(t)dt = 1$

and

δ(t = 0 for t ≠ 0)

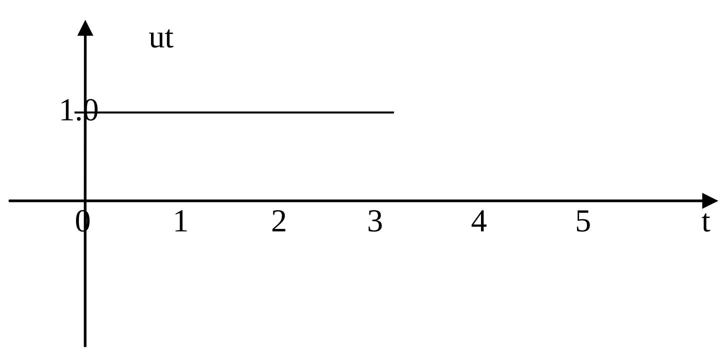
That is the impulse function has zero amplitude everywhere expect at t = 0.



Step Function

The unit step function is defined as

U(t) = 1 for t ≥ 0
= 0 for t > 0



Program:

```
t=(-2:0.01:10);
impulse = t==0;
unitstep = t>=0;
plot(t, impulse)
plot(t, unitstep)
ramp = t.*unitstep;
plot(t,ramp)
xlabel('Time')
ylabel('Amplitude')
title('impulse  function')
title('unit step function')
title('ramp function')
```

2-D plot

Define x as a vector of linearly spaced values between 0 and 2π. Use an increment of π/10 between the values. Define y as sine values of x.

Program:

```
clc
clear all
x=(0:pi/10:2*pi)
y=sin(x)
plot(x,y)
title('2D Plot')
xlabel('Time')
ylabel('Amplitude')
```

3-D plot

A three-dimensional plot may refer to

a) A graph or plot embedded into a three-dimensional space

b) The plot of a function of two variables, embedded into a three-dimensional space

Program:

```
clc
clear all
t=(-4:0.01:4)
x=t.^2
y=4*t
plot3(x,y,t)
grid on
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
title('3D Plot')
```

RESULTS:

Results have been seen on the command window.

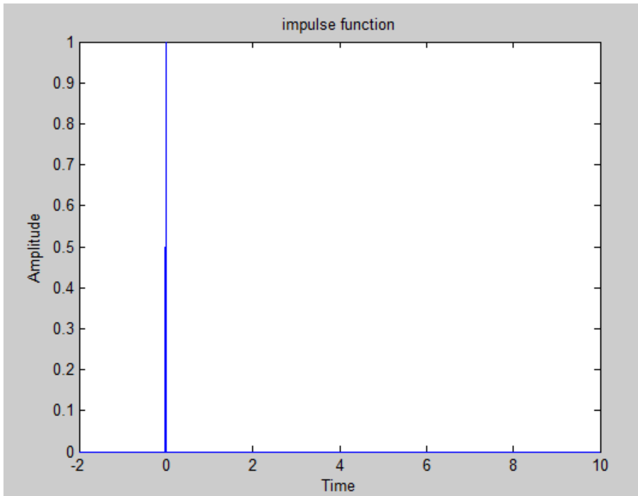


Figure-1

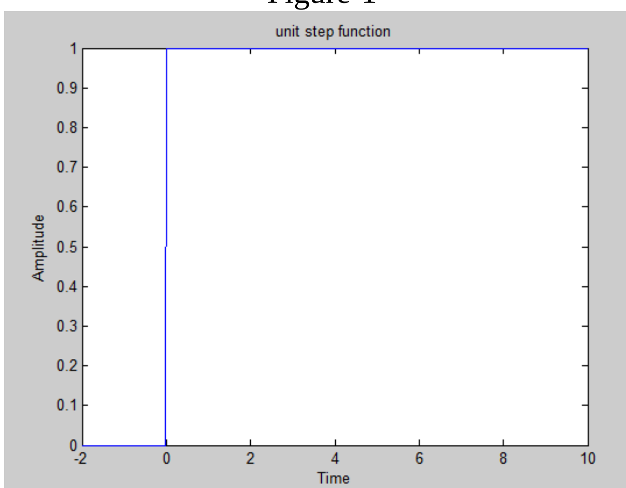


Figure-2

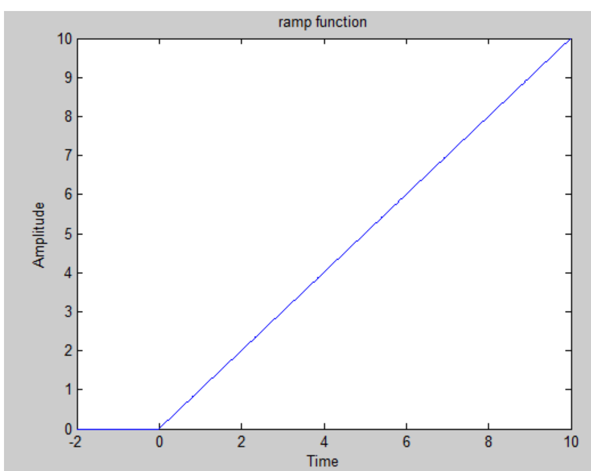


Figure-3

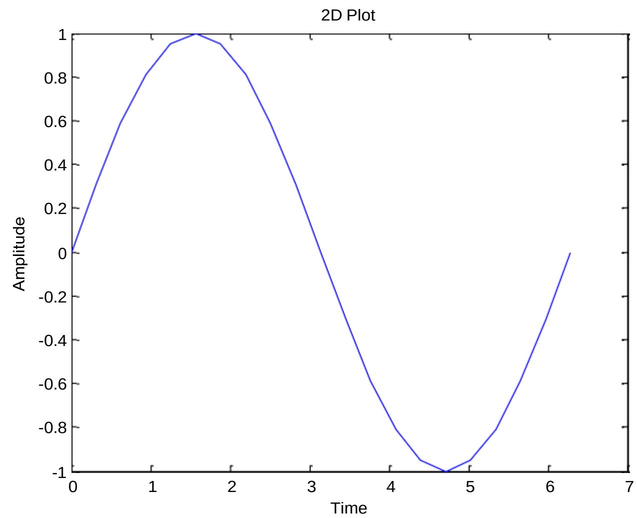


Figure-4

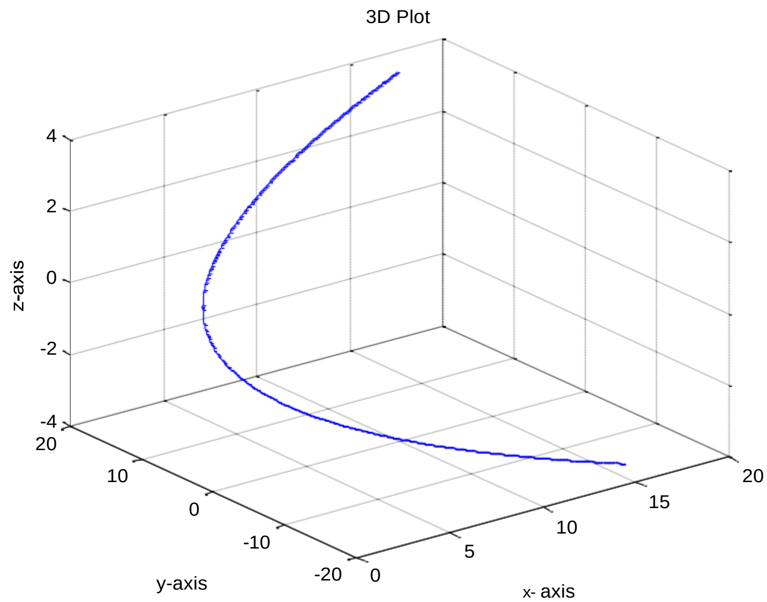


Figure-5

<u>PRECAUTIONS:-</u>

1) Program must be written carefully to avoid errors.

2) Programs can never be saved as standard function name.

3) Functions in MATLAB are case sensitive so commands must be written in proper format.

RELATED QUESTIONS:-

1. What is the type of program files that MATLAB allows to write?
2. How to plot a graph in MATLAB?
3. What is MATLAB used for?
4. How to run MATLAB code?
5. How to comment in MATLAB?

LAB EXPERIMENT 3

OBJECT: Write a MATLAB program to obtain linear convolution of the given sequences.

SOFTWARE USED: MATLAB 7.9

PROCEDURE:-

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

THEORY:

Convolution is a mathematical operation used to express the relation between input and output of an LTI system. It relates input, output and impulse response of an LTI system as

$$y(t)=x(t)*h(t)$$

Where y (t) = output of LTI

x (t) = input of LTI

h (t) = impulse response of LTI

There are two types of convolutions:

a) Continuous Convolution

$$\begin{aligned} y(t) &= x(t) * h(t) \\ &= \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \end{aligned}$$

b) Discrete Convolution

$$\begin{aligned} y(n) &= x(n) * h(n) \\ &= \Sigma_{k=-\infty}^{\infty} x(k)h(n-k) \end{aligned}$$

PROGRAM:

```
clc;
clear all;
close;
disp('enter the length of the first sequence m=');
m=input('');
disp('enter the first sequence x[m]=');
for i=1:m
   x(i)=input('');
end
disp('enter the length of the second sequence n=');
n=input('');
disp('enter the second sequence h[n]=');
for j=1:n
   h(j)=input('');
end
y=conv(x,h);
figure;
subplot(3,1,1);
stem(x);
ylabel ('amplitude---->');
xlabel('n--- >');
title('x(n) Vs n');
subplot(3,1,2);
stem(h);
ylabel('amplitude --->');
xlabel('n--- >');
title('h(n) Vs n');
subplot(3,1,3);
stem(y);
ylabel('amplitude --->');
xlabel('n--- >');
title('y(n) Vs n');
disp('linear convolution of x[m] and h[n] is y');
```

INPUT:--

Enter the length of the first sequence m=

6

Enter the length of first sequence x[m]=

1

2

3

4

5

6

Enter the length of the second sequence n=

6

Enter the length of second sequence h[n]=

1

2

3

4

5

6

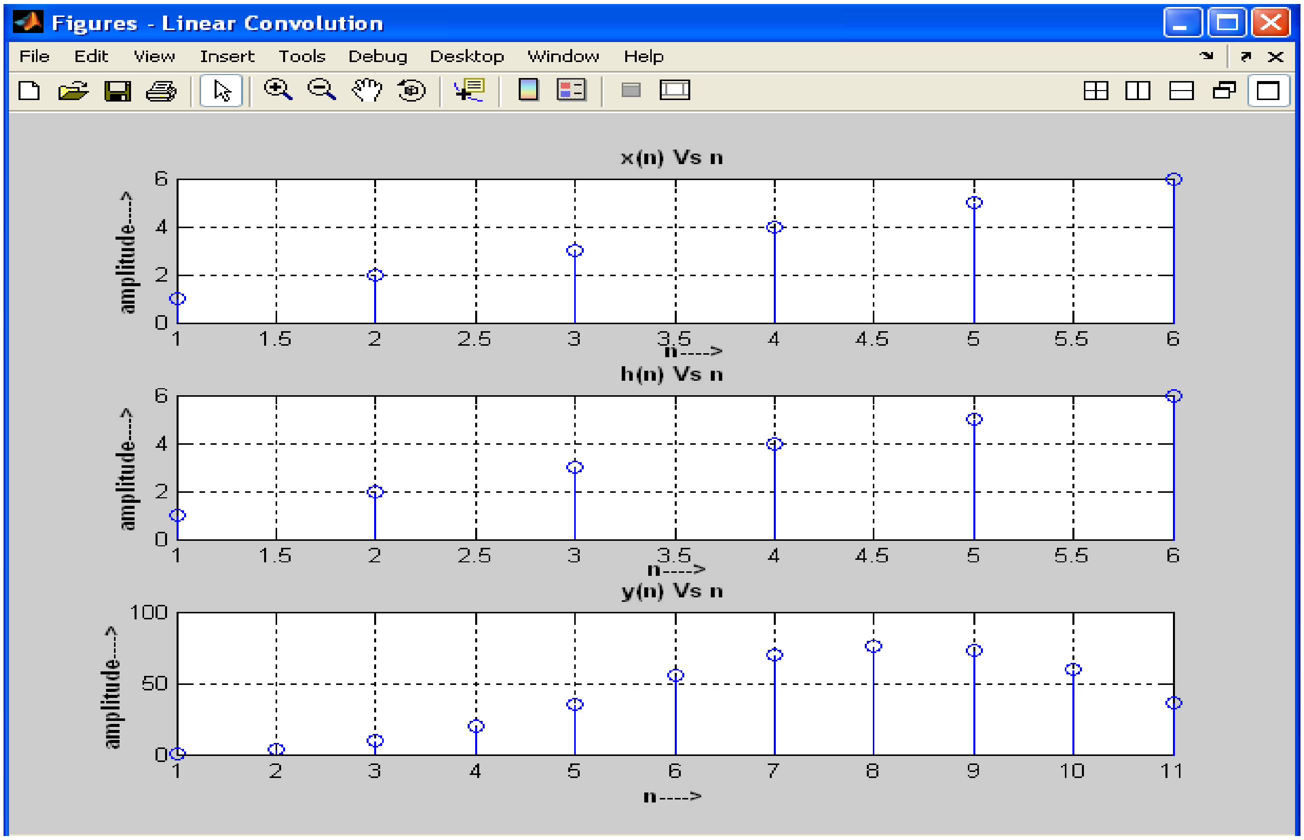
OUTPUT:-

Linear convolution of x[m] and h[n] is y=

1 4 10 20 35 56 70 76 73 60 36

RESULTS: - Thus the program for linear convolution is written using MATLAB and verified

.



PRECAUTIONS:

1) Program must be written carefully to avoid errors.

2) Programs can never be saved as standard function name.

3) Functions in MATLAB are case sensitive so commands must be written in proper format.

RELATED QUESTIONS:

1. What is the MATLAB working environment?
2. What is the function of MATLAB function clear all?
3. What function can be used in MATLAB to calculate the linear convolution of two sequences?
4. State mathematical formula for convolution of 2 discrete time signals X_1 (n) & X_2 (n)

LAB EXPERIMENT 4

OBJECT:- Write a MATLAB program to perform amplitude-scaling, time-scaling andtime-shifting on a given signal.

SOFTWARE USED:-MATLAB 7.9

PROCEDURE:-

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

THEORY:

Amplitude Scaling

In the amplitude scaling, a signal y(t) resulting from a signal x(t), on amplitude scaling by a real value k becomes:

$$y(t)=k\,x(t)$$

Program:

```
clc
clear all
t=(-3:0.01:7)
unitstep=t>=0
u1=3*unitstep
ramp = t.*unitstep
r1=2*ramp
plot(t,unitstep)
hold on
plot(t,ramp)
hold on
plot(t,u1)
hold on
plot(t,r1)
title('amplitude-scaling')
xlabel('Time')
ylabel('Amplitude')
```

Time Shifting

The time shifting of the signal x(t) is done by replacing t by $t-t_0$. Thus $x(t-t_0)$ represents x(t) time shifting by t_0. If t_0 is positive, the shift is to the right, i.e. $x(t- t_0)$ represents the

signal x(t) delayed by t_0 and if t_0 is negative, the shift is to the left, i.e., $x(t+ t_0)$ represents the signal x(t) advanced by t_0.

Program:

```
clc
clear all
t=(-1:0.01:10)
impulse=t==0
unitstep=t>=0
u1=t>=3
u2=t>=5
plot(t,unitstep,'r')
hold on
plot(t,u1,'g')
hold on
plot(t,u2,'m')
legend('unitstep','u1','u2')
title('Time-Shifting')
xlabel('Time')
ylabel('Amplitude')
```

Time scaling

```
clear all;
close all;
clc;
k=2;
x1=[1 2 3 4 5];
a=length(x1);
n=0:1:a-1;
subplot(3,1,1);
stem(x1);
xlabel('number of samples');
ylabel('amplitude');
title('input signal');
x2=k*x1;
subplot(3,1,2);
stem(x2);
xlabel('number of samples');
ylabel('amplitude');
title('amplified signal');
x3=x1/k;
subplot(3,1,3);
stem(x3);
xlabel('number of samples');
ylabel('amplitude');
title('attenuated signal');
```

```
display(x2);
display(x3);
```

RESULTS:-

Results have been seen on the command window.

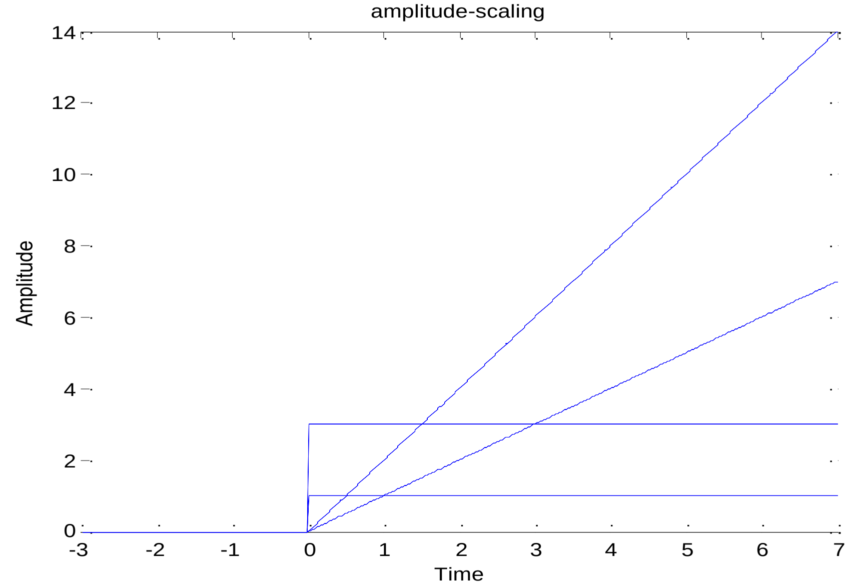


Figure-1

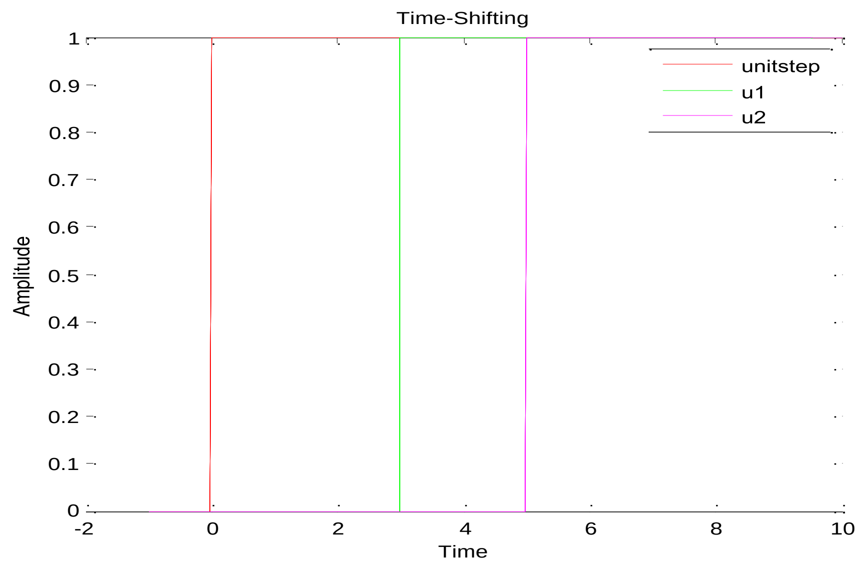


Figure-2

Time scaling

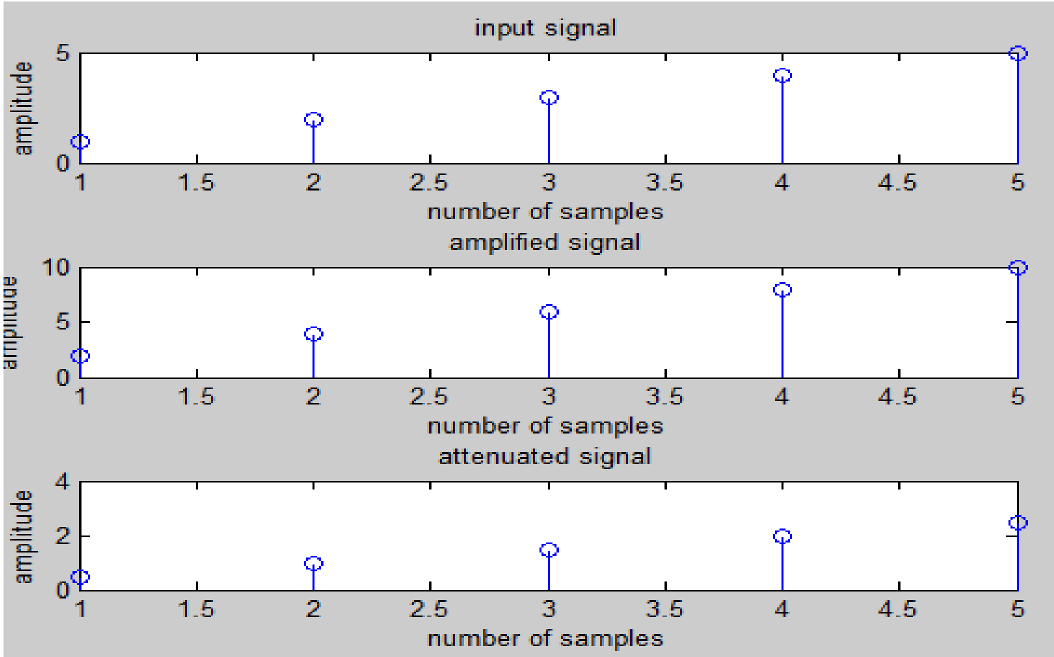


Figure-3

PRECAUTIONS:-

1) Program must be written carefully to avoid errors.

2) Programs can never be saved as standard function name.

3) Functions in MATLAB are case sensitive so commands must be written in proper format.

RELATED QUESTIONS

1. What operations can be performed on signal?
2. What functions can be used in MATLAB to label the axis in graph?
3. Differentiate between clc and clear all functions.
4. Functions in MATLAB are case sensitive or not.
5. How to plot a graph in MATLAB?

LAB EXPERIMENT 5

OBJECT:-Write a MATLAB program to compute autocorrelation and cross correlation of a sequence.

SOFTWARE USED:-MATLAB 7.9

PROCEDURE:-

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

THEORY:

Auto-correlation: Autocorrelation function gives the measure of similarity, match or coherence between a signal and its delayed replica. This means that autocorrelation function is a special from of a cross- correlation function. It is defined as the correlation of a signal with itself.

Consider a signal x(t).Then the autocorrelation function of this signal with its delayed version will be

$$R(\tau) = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)\, x(t-\tau)\, dt$$

The autocorrelation function is defined separately for energy signals and for power or periodic signals.

Cross-Correlation: The correlation or more precisely cross-correlation between two different waveforms or two signals is the measure of match or similarity or regularity between one signal and time delayed version of the other signal.

Let us consider two general complex signals $x_1(t)$ and $x_2(t)$ is defined as

$$R_{12}(\tau) = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} x_1(t)\, x_2(t-\tau)\, dt$$

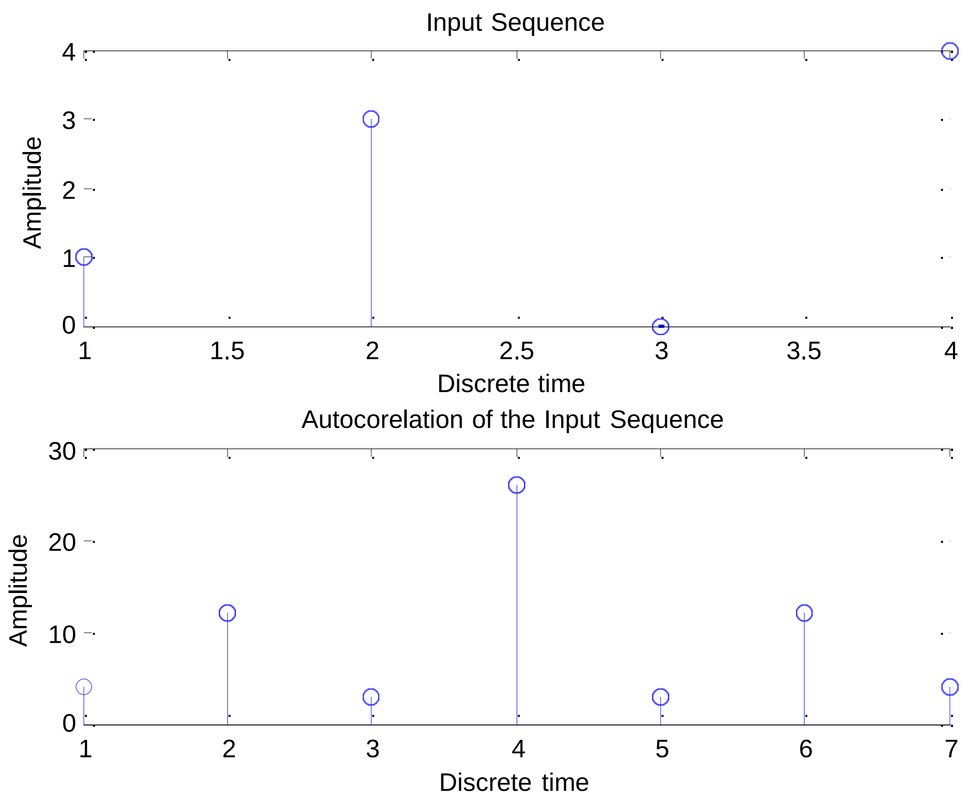
PROGRAM:

```
stem(x1)
xlabel('Discrete time')
ylabel('Amplitude')
title('Input Sequence')
subplot(2,1,2)
```

```
stem(y)
title('Autocorelation of the Input Sequence')
xlabel('Discrete time')
xlabel('Discrete time')
```

RESULTS:

Results have been seen on the command window.



Cross-correlation

```
r = xcorr(x,y)
r = xcorr(x)
r = xcorr(___,maxlag)
r = xcorr(___,scaleopt)
[r,lags] = xcorr(___)
```

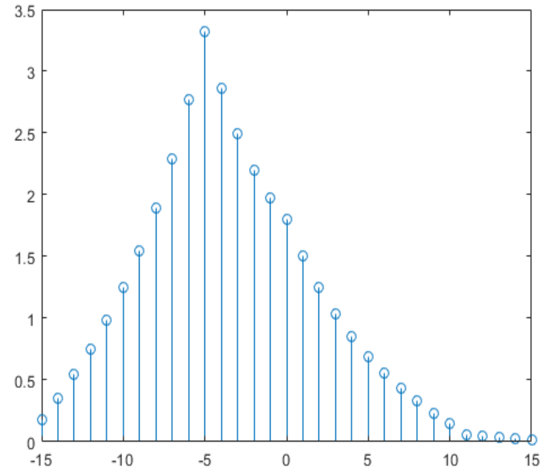
example

Cross Correlation of two vectors

Create a vector x and a vector y that is equal to x shifted by 5 elements to the right. Compute and plot the estimated cross-correlation of x and y. The largest spike occurs at the lag value when the elements of x and y match exactly (-5).

```
n = 0:15;
```

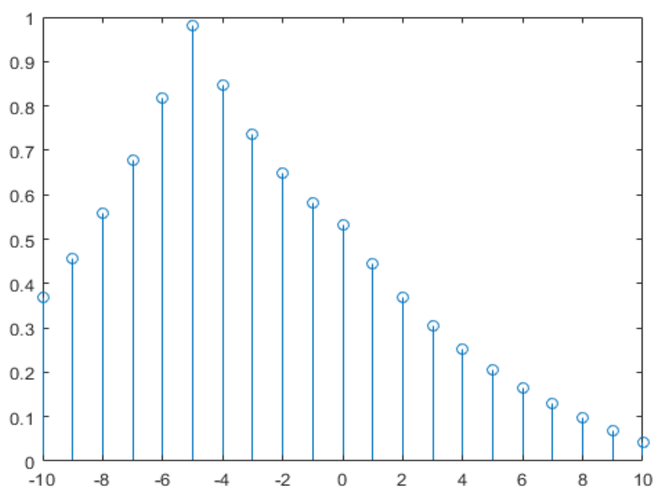
```
x = 0.84.^n;
y = circshift(x,5);
[c,lags] = xcorr(x,y);
stem(lags,c)
```



Normalized Cross-Correlation

Compute and plot the normalized cross-correlation of vectors x and y with unity peak, and specify a maximum lag of 10.

```
n = 0:15;
x = 0.84.^n;
y = circshift(x,5);
[c,lags] = xcorr(x,y,10,'normalized');
stem(lags,c)
```



PRECAUTIONS:

1) Program must be written carefully to avoid errors.

2) Programs can never be saved as standard function name.

3) Functions in MATLAB are case sensitive so commands must be written in proper format.

RELATED QUESTIONS:

1. Differentiate between auto correlation and cross correlation.
2. What is the use of stem function in MATLAB?
3. What do you mean by M-file in MATLAB?
4. What is the use of title function in MATLAB?
5. What is the relation between cross correlation and auto correlati

LAB EXPERIMENT 6

OBJECT: Write a MATLAB program to generate Fourier series of a Square Wave.

SOFTWARE USED: MATLAB 7.9

PROCEDURE:-

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

THEORY:

Let's assume we have a square wave with following characteristics:
Period=2ms
Peak−to−Peak Value=2 V
Average Value=0 V
So, we can express it as:

$$x(t) = \frac{4}{\pi}\sum_{n=1}^{\infty}\frac{1}{(2n-1)}\sin[(2n-1)2\pi f_o t] \ \cdots \ (1)$$

$$and\ assume$$

$$f_0 = 500Hz$$

If g(t) is given by

$$g(t) = \frac{4}{\pi}\sum_{n=1}^{12}\frac{1}{(2n-1)}\sin[(2n-1)2\pi f_o t] \ \cdots \ (2)$$

Now, a Mat lab code can be written for g(t) between 0 and 4ms with an interval of 0.05 ms to demonstrate that g(t) is a decent approximation of original function x(t).

PROGRAM:

```
% Fourier Series Expansion for Square Wave
%% Parameters as mentioned in text
f = 500;                          % Frequency
C = 4/pi;                         % Constant Value
dt = 5.0e-05;                     % Interval between teo time steps
tpts = (4.0e-3/5.0e-5) + 1;       % Total points & quot;(final point-initial point)/Interval+1%
for n = 1: 12                     % Values we are considering to approximate Fourier Series
instead of infinity as given in original function x(t)
```

```
for m = 1: tpts                        % Here, we'll consider all & quot;t& quot; points to cover &
quot;from 0 to 4ms interval & quot;
s1(n,m) = (4/pi)*(1/(2*n - 1))*sin((2*n - 1)*2*pi*f*dt*(m-1));          % Approximate Fourier
Series g(t)
end
end
for m = 1:tpts
a1 = s1(:,m);                          % VERY IMPORTANT ! Here, we are assigning a1 for each
single column (total 81)
a2(m) = sum(a1);                       % Here, we are summing up the whole column to one single
value (adding all 12 values in one column)
end
                                       % Now, we have a row vector &quot;a2&quot; with total values
&quot;81&quot;
f1 = a2';                              % Here, we have final values &quot;f1&quot; (total 81 points) as
transpose of a2 computed above
t = 0.0:5.0e-5:4.0e-3;                 % it's already given in text (0 to 4ms with interval of 0.05 ms)
plot(t,f1)
xlabel('Time, s')
ylabel('Amplitude, V')
title('Fourier Series Expansion')
```

<u>RESULTS:</u>

Results have been seen on the figure window.

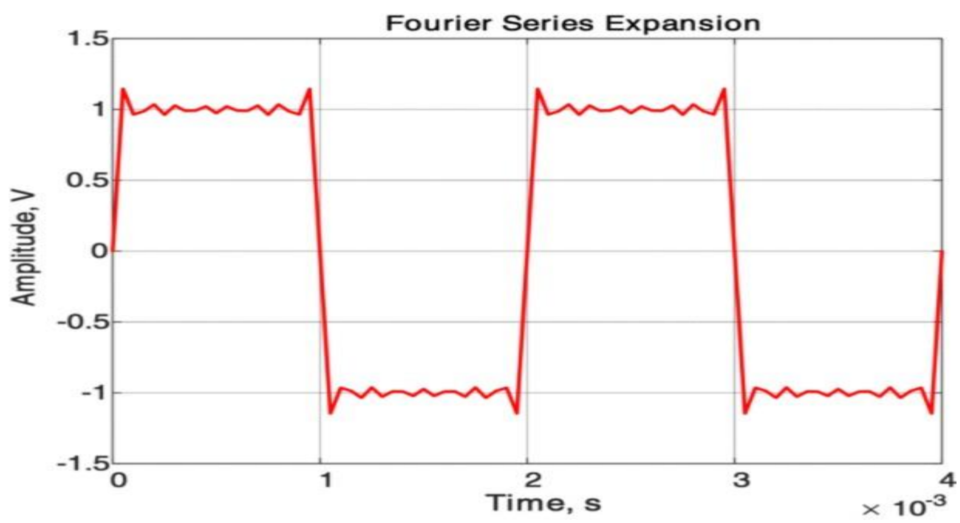


Fig. 1: Fourier Series of Square Wave

PRECAUTIONS:

1) Program must be written carefully to avoid errors.

2) Programs can never be saved as standard function name.

3) Functions in MATLAB are case sensitive so commands must be written in proper format.

RELATED QUESTIONS:

1. State Dirichlet's conditions for a function to be expanded as a Fourier series.
2. What is Fourier series?
3. What are the two types of Fourier series?
4. How is a trigonometric Fourier series represented?
5. How to comment multiple lines in MATLAB?

LAB EXPERIMENT 7

OBJECT:
Write a MATLAB program to Calculate and plot using MATLAB Fourier Transform and Z-Transform of a given signal.

SOFTWARE USED: MATLAB 7.9

PROCEDURE:-

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

THEORY:

Z transform:-

The (one-sided) z-transform of a function $f(n)$ is defined as

$$Z[f](z) = \sum_{n=0}^{\infty} f(n) z^{-n}.$$

The notation $Z[f]$ refers to the z-transform of f at z. Let R be a positive number so that the function $g(z)$ is analytic on and outside the circle $|z| = R$. Then the inverse z-transform (IZT) of g at n is defined as

$$Z^{-1}[g](n) = \frac{1}{2\pi i} \oint_{|z|=R} g(z) z^{n-1} dz,\ n = 1,2,...$$

The notation $Z^{-1}[f]$ means the IZT of f at n. The Symbolic Math Toolbox commands `ztrans` and `iztrans` apply the z-transform and IZT to symbolic expressions, respectively. See `ztrans` and `iztrans` for tables showing various mathematical representations of the z-transform and inverse z-transform and their Symbolic Math Toolbox counterparts.

The z-transform is often used to solve difference equations. In particular, consider the famous "Rabbit Problem." That is, suppose that rabbits reproduce only on odd birthdays (1, 3, 5, 7, ...). If $p(n)$ is the rabbit population at year n, then p obeys the difference equation

$p(n+2) = p(n+1) + p(n)$, $p(0) = 1$, $p(1) = 2$.

Fourier Transform:-

The Fourier transform of the function *f* is traditionally denoted by adding a circumflex: $\hat{f}$. There are several common conventions for defining the Fourier transform of an integral function $f : \mathbb{R} \to \mathbb{C}$. Here we will use the following definition:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \xi}\, dx,$$

for any real number ξ.

When the independent variable *x* represents *time* (with SI unit of seconds), the transform variable ξ represents frequency (in hertz). Under suitable conditions, *f* is determined by $\hat{f}$ via the **inverse transform**:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{2\pi i \xi x}\, d\xi,$$

f or any real number *x*.

PROGRAM:

Fourier transform of exp (-2*x^2)

```
clc
clear all
syms x
f=exp(-2*x^2)        %our function
ezplot(f,[-2,2])     % plot of our function
FT=fourier(f)        %Fourier transform
ezplot(FT)
```

z transform of $x(n) = \frac{1}{4^n} u(n)$

```
clc
clear all
syms z n
ztrans(1/4^n)
```

Inverse z transform of $X(z) = \frac{2z}{2z-1}$

```
clc
clear all
syms z n
```

iztrans(2*z/(2*z-1))

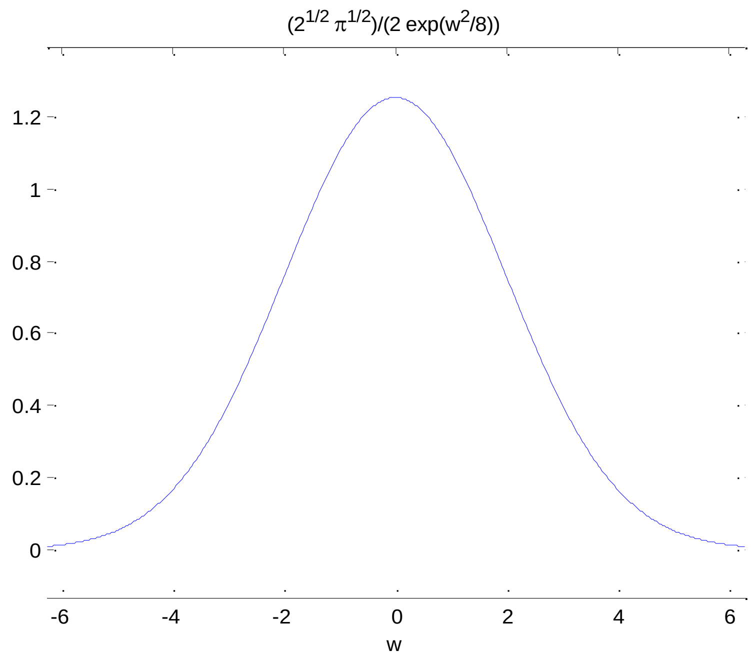


Figure-1

RESULTS:

Z- Transform of $x(n) = ¼^n\, u(n)$

$x(z = z/(z^{-1/4})$

Inverse Z- Tranforms of $x(z) = 2z/2z^{-1}$

$X(n) = (1/2)^n$

PRECAUTIONS:

1) Program must be written carefully to avoid errors.

2) Programs can never be saved as standard function name.

3) Functions in MATLAB are case sensitive so commands must be written in proper format.

RELATED QUESTIONS:

1. What are functions used in MATLAB to find z transform and Fourier transform?
2. Find Z Transform of X(n) = nu(n).
3. Find the inverse Fourier transform of f(t)=1.
4. What is the set of all values of z for which X(z) attains a finite value?
5. The Z-Transform X(z) of a discrete time signal x(n) is defined as_____________

LAB EXPERIMENT 8

OBJECT: Write a MATLAB program to find the impulse response and step response of a system from its difference equation.

SOFTWARE USED: MATLAB 7.9

PROCEDURE:-

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

THEORY:

Step Response

Step Response is the output from a system excited with a step input. The output step response lasts for infinite time.

Impulse Response

Impulse Response is the output from a system excited with a unit impulse input. The output impulse response lasts for infinite time.

PROGRAM:

Step Response: Determine the first 50 values of the step response of a system with a known impulse response h[n]=0.5n *u[n].

```
clc
clear all
h=0.5.^[0:49]
x=ones(1,50)
y=conv(x,h)
stem([0:49],y(1:50))
```

Impulse Response: Determine the first 50 values of the step response of a system with a known impulse response h[n] =0.5n *u[n]. Here define delta as a vector containing 21 elements. Element number 11 is one the rest are zeros.)

```
clc
clear all
delta=[ zeros(1,10), 1, zeros(1,10)]
h=0.5.^[0:49]
y=conv(delta,h)
stem([0:49],y(1:50))
```

RESULTS:

Results have been seen on the command window.

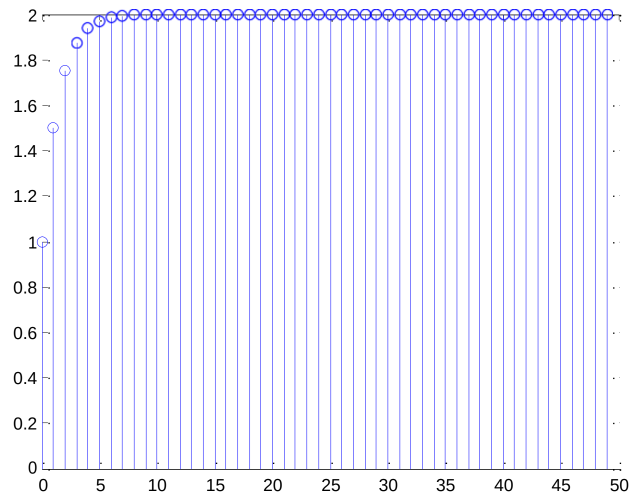


Figure-1: Impulse Response

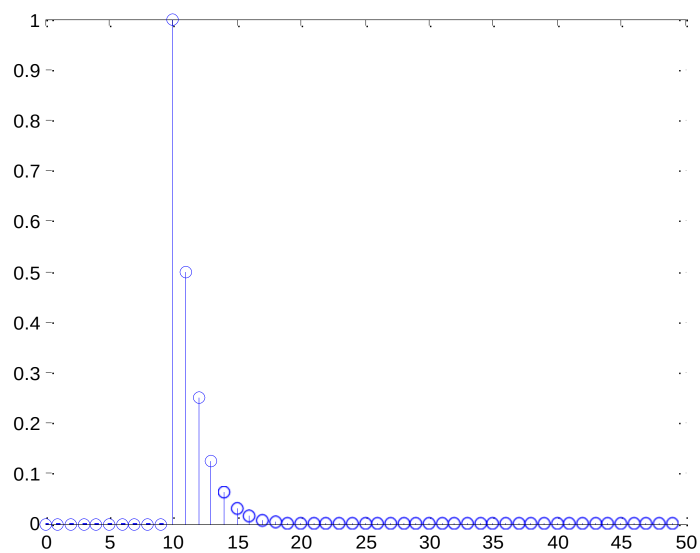


Figure-2: Step Response

PRECAUTIONS:

1) Program must be written carefully to avoid errors.

2) Programs can never be saved as standard function name.

3) Functions in MATLAB are case sensitive so commands must be written in proper format.

RELATED QUESTIONS:

1. What is the relation between the step response and impulse response?
2. What is the use of conv function in MATLAB?

3. What will be the step response of the impulse response of a ramp function?

4. To save a file, what should be the extension of file?

5. In MATLAB, the impulse response of the step response of a system is_____to the step response of the impulse response of the system.

LAB EXPERIMENT 9

OBJECT: Write a MATLAB program to plot magnitude and phase response of a given system.

SOFTWARE USED: MATLAB 7.9

PROCEDURE:-

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

THEORY:

If X(jΩ) is Fourier Transform of continuous time signal x(t), then

$$X(j\Omega) = \int_{-\infty}^{\infty} x(t)\ e^{-j\Omega t}\,dt \quad \text{for all } \Omega$$

and

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\Omega)\, e^{j\Omega t}\, d\Omega \quad \text{for all } t$$

In general, X (jΩ) is a complex valued function of Ω. Therefore we can write X(jΩ) as

$$X(j\Omega) = X_R(j\Omega) + j\,X_I(j\Omega)$$

Where

$X_R(j\Omega)$ is real part of X(jΩ)

and

$X_I(j\Omega)$ is imaginary part of X(jΩ)

The magnitude of X(jΩ)is given by

$$\underline{X(j\Omega)} = \sqrt{X_R(j\Omega)^2 + X_I(j\Omega)^2}$$

and the phase of X(jΩ) is given by

$$X(j\Omega) = \tan^{-1}\frac{X_I(j\Omega)}{X_R(j\Omega)}$$

The plot between | X (jΩ) | vs Ω is known as amplitude spectrum and the plot between X (jΩ) vs Ω is known as phase spectrum.

PROGRAM:

```
clc

clear all

t = 0:1/100:10-1/100;

x = sin(2*pi*15*t) + sin(2*pi*40*t);

y = fft(x);

m = abs(y);

p = unwrap(angle(y));

f = (0:length(y)-1)'*100/length(y);

subplot(2,1,1),  plot(f,m)

ylabel('Abs. Magnitude')

grid on

hold on;

subplot(2,1,2), plot(f,p*180/pi)

ylabel('Phase [Degrees]')

grid on

xlabel('Frequency [Hertz]')
```

RESULTS:

Results have been seen on the command window.

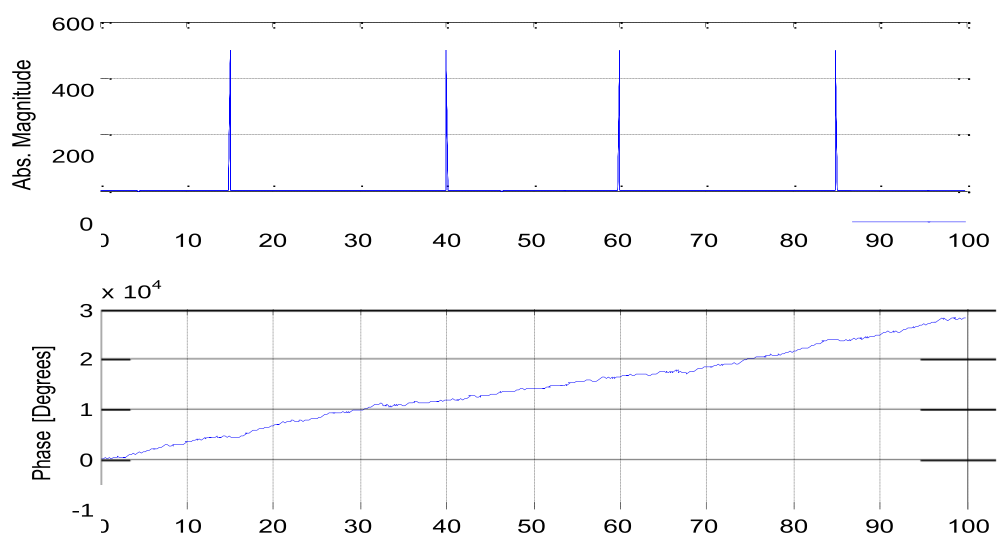


Figure-1

PRECAUTIONS:

1) Program must be written carefully to avoid errors.

2) Programs can never be saved as standard function name.

3) Functions in MATLAB are case sensitive so commands must be written in proper format.

RELATED QUESTIONS:

1. How do you find the magnitude and phase of a frequency response?
2. Differentiate between plot and subplot functions
3. What do you understand by grid on and hold on functions in MATLAB?
4. Is it possible to save a program with a standard function name?
5. Explain MATLAB API (Application Program Interface)

LAB EXPERIMENT 10

OBJECT: Checking Linearity/Non-Linearity of a system using SIMULINK.

SOFTWARE USED: MATLAB 7.9

PROCEDURE:-

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

THEORY:

Linear and nor-linear systems

A system that obeys the superposition principle is said to be a linear systems. Superposition principle states that the response to a weighted sum of input signals be equal to the corresponding weighted sum of the outputs of the systems to each of the individual input signals. If an arbitrary input x_1**(t)** produces an output $\mathbf{y_1(t)}$ and an arbitrary input x_2**(t)** produces an output $\mathbf{y_2(t)}$ then the system is linear if the weighted sum of inputs **a**x_1**(t) + b**x_2**(t)** produces an output $\mathbf{ay_1(t)+by_2(t)}$, where a and b are constants.

That is,

$$T[ax_1(t) + bx_2(t)] = aT[x_1(t)] + bT[x_2(t)]$$

Similarly for a discrete-time linear system

$$T[ax_1(n) + bx_2(n)] = aT[x_1(n)] + bT[x_2(n)]$$

PROGRAM:

```
clc
clear all
a = 2
n = 3
S1 = @(x) a*x.^n    %// anonymous function 1 describing your system
x1 = 1:4            %// test input signal
x2 = 11:14          %// test input signal. Same length
r1 = S1(x1)+S1(x2)
r2 = S1(x1+x2)
S2 = @(x) a*x        %// anonymous function 2 describing your system
r1 = S2(x1)+S2(x2)
r2 = S2(x1+x2)
```

RESULTS:

Results have been seen on the command window.

```
a =
   2
```

```
n =
     3
S1 =
    @(x)a*x.^n

x1 =

     1     2     3     4
x2 =
    11    12    13    14
r1 =
        2664        3472        4448        5616
r2 =
        3456        5488        8192       11664
S2 =
    @(x)a*x
r1 =
    24    28    32    36
r2 =
    24    28    32    36
```

PRECAUTIONS:

1) Program must be written carefully to avoid errors.

2) Programs can never be saved as standard function name.

3) Functions in MATLAB case sensitive so commands must be written in proper format.

RELATED QUESTIONS:

1. What is the difference between linear and nonlinear system?
2. What is the following type of system called? y[n] = x[n] + y[n-1].
3. Determine the nature of the given system: y(t)=x(sint).
4. What does MATLAB consist o

This lab manual has been updated by

Dr. Pallavi Verma

(pallavi.verma@gnindia.dronacharya.info)

Crosschecked By
HOD ECZ

Please spare some time to provide your valuable feedback.