UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lecture-wise Plan

Subject Name: **Teaching and Research Methodology** Subject Code- MVLSI301
Year: 2ndYear Semester: Third

Module Number	Topics	Number of Lectures
1	**Instruction**	**2L**
	Introduction to content, Elements of instruction, Learning objectives,	1
	Roles of the teacher and the learner in instruction	1
2	**Teaching and Learning**	**4L**
	Application of theories of learning to teaching and learning, Sequence of learning and Strategies of learning,	2
	Teaching methods, their merits and demerits,	1
	Use of ICT in teaching & learning, Classroom management, Individual differences.	1
3.	**Planning for teaching and learning**	**3L**
	Understanding the syllabus, Preparation of a scheme of work,	2
	Lesson plan preparation, Micro teaching	1
4.	**Assessment and Evaluation**	**4L**
	Define measurement, assessment, test, evaluation, Purpose of assessment and evaluation,	2
	Types of tests, Grading and reporting the results assessment.	1
	Evaluating teaching and learning	1
5.	**Definition and explanation of research**	**4L**
	Types and Paradigms of Research, History and Philosophy of Research (esp. Philosophical evolution, pathways to major discoveries & inventions),	2
	Research Process decision, planning, conducting, Classification of Research Methods;	2
	Reflective Thinking, Scientific Thinking.	1
6.	**Research problem formulation:**	**11L**
	Literature review- need, objective, principles, sources, functions & its documentation,	2
	Problem formulation esp. sources, considerations & steps, Criteria of a good research problem, Defining and evaluating the research problem,	2
	Variables esp. types & conversion of concepts to variables. Research design esp. Causality, algorithmic, quantitative and qualitative designs,	2
	Various types of designs. Characteristics of a good research design, problems and issues in research design;	3

	Hypotheses: Construction, testing, types, errors; Design of experiments especially classification of designs and types of errors.	2
7.	**Problem solving:**	**5L**
	Understanding the problem- unknowns, data & conditions, conditions - satisfiability, sufficiency, redundancy & contradiction,	1
	Separation of parts of the problem and conditions, notations; devising a plan- connection between data and unknown, similar/related problems, reuse of previous solutions, rephrasing/transforming the problem, solving partial or related problem,	2
	Transforming data and unknowns; carrying out the plan- esp. correctness of each step in multiple ways;	1
	Evaluation of solution and method- checking correctness of solution, different derivations, utility of the solution	1
8.	**Data & Reports:**	**5L**
	Infrastructural setups for research; Methods of data collection esp. validity and reliability, Sampling; Data processing and Visualization espicially Classification;	2
	Ethical issues espicially. bias, Misuse of statistical methods, Common fallacies in reasoning. Research Funding & Intellectual Property;	1
	Research reports: Research Proposal & Report writing esp. Study objectives, study design, problems and limitations;	1
	Prototype micro- project report implementing a major part of all the above (compulsory assignment)	1
Total Number Of Hours = 38		

Faculty In-Charge HOD, CSE Dept.

UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR
Lab Manual

Title of Course: Advanced Digital Signal Processing lab
Course Code: MVLSI381
L-T-P scheme: 0-0-3 **Course Credit: 2**

Objectives:
The main objective of this course is to introduce the architecture of DSP processor for developing real-time applications. In this course students, will learn about the computational building blocks and the basic architectural features of DSP. They will learn about programmable digital signal processors and implementation details of DSP algorithms like digital filters, including basic adaptive filters and FfTs. They will also be introduced to CODEC programming and interfacing codec and DSP as well as several real-world applications of DSP processors.

Learning Outcomes:
1. Understand the architecture and building blocks of digital signal processor.
2. Analyze and process signals using DSP Processor.
3. Implementing FIR, IIR and basic adaptive filters to suit specific requirements for specific applications.
4. Learn codec programming and interfacing it with DSP.
5. Understand the applications of DSP processors
6. Designing and implementing a small application using DSP processor

Course Contents:
Exercises that must be done in this course are listed below:
Experiment 1: - Generate continuous and Discrete signal
Experiment 2: - Graphical representation of unit step signal
Experiment 3: - Graphical representation of unit sample signal
Experiment 4: - Graphical representation of unit ramp signal
Experiment 5: - Graphical representation of exponential signal
Experiment 6: - Graphical representation of exponential increasing- decreasing signal
Experiment 7: - Graphical representation of even signal
Experiment 8: - Graphical representation of odd signal
Experiment 9:- Determine whether given signal is periodic or not
Experiment 10: - Convolution of given sequences
Experiment 11: - Cross correlation of given sequences
Experiment 12: - Plot Magnitude and Phase Response
Experiment 13: - Impulse Response of a given System
Experiment 14: -Z Transform of the Sequence a given sequence
Experiment 15: - Inverse Z Transform of the Sequence a given sequence
Experiment 16: - DFT and IDFT of a Sequence
Experiment 17: - 8- point DFT of the Sequence
Experiment 18: - Circular convolution of following sequences

Text Book:

1. J. G. Proakis and D. G. Manolakis, "Digital Signal Processing", Prentice Hall India, 3rd edition, 1997, ISBN: 81-203-1129-9

Recommended Systems/Software Requirements:
1. SCILAB

Experiment 1: -Generate continuous and Discrete signal

AIM: - To write a scilab code to sketch the continuous time signal x (t) =2∗exp(-2 t) and also its discrete time equivalent signal with a sampling period T = 0 . 2 sec

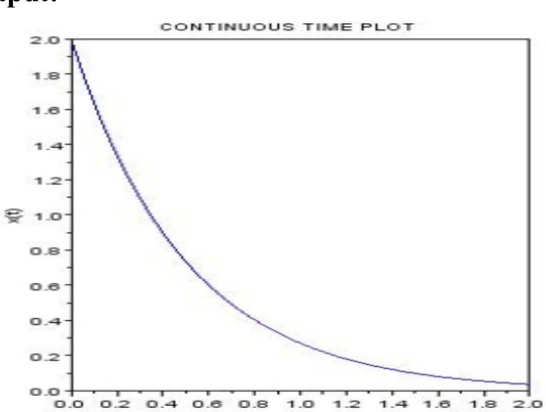
Algorithm: -

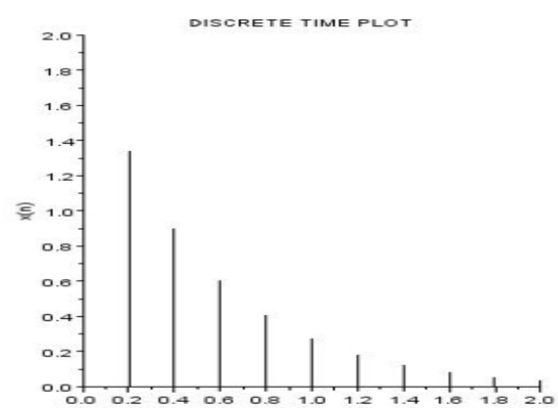
Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear;
clc ;
 close ;
 t =0:0.01:2;
 x1 =2* exp ( -2*t);
 subplot (1 ,2 ,1);
 plot (t,x1);
xlabel ( ' t ' );
ylabel ( ' x ( t ) ' );
 title ( 'CONTINUOUS TIME PLOT ' );
 n =0:0.2:2;
 x2 =2* exp ( -2*n);
subplot (1 ,2 ,2);
plot2d3 (n, x2);
xlabel ( 'n ' );
ylabel ( ' x (n) ' );
title ( 'DISCRETE TIME PLOT ' );
```

Output: -





UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

Lab Manual

Experiment 2: - Graphical representation of unit step signal

AIM: - To write the Scilab code to find the unit step signal and sketch the output wave form.

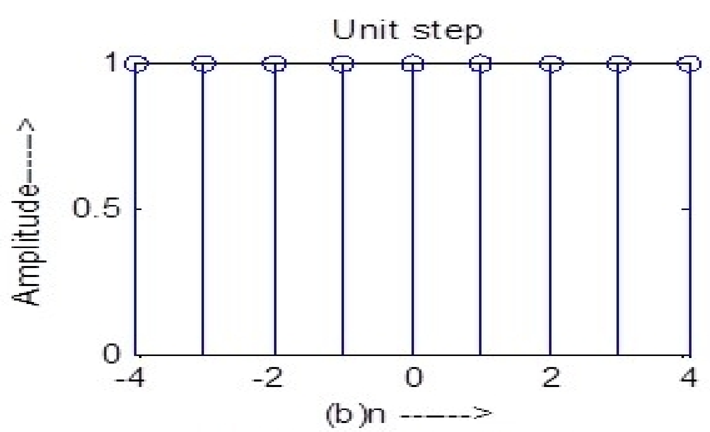
Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear;
clc;
close;
L = 4; // Upper limit
n = -L: L;
x = [ zeros (1, L), ones (1, L +1)];
a= gca ();
a. thickness = 2;
a. y_location = "middle ";
plot2d3 (‘gnn‘, n,x)
xtitle(‘Graphical Representation of Unit Step Signal', ' n ', ' x [ n] ' );
```

Output: -



Experiment 3: - Graphical representation of unit sample signal

AIM: - To write the Scilab code to find the unit sample signal and sketch the output wave form.

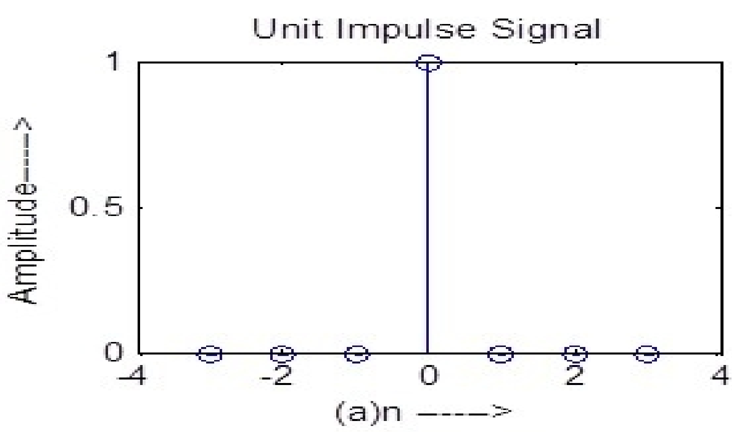
Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear;
clc;
close;
L = 4; // Upper limit
n = -L: L;
x = [ zeros (1,L) ,1, zeros (1,L)];
a= gca ();
a.thickness = 2;
a.y_location = " middle ";
plot2d3 ( ’ gnn ’ ,n,x)
xtitle ( ’ Graphical Representation of Unit Sample Sequence ’ , ’ n ’ , ’ x [ n ] ’ );
```

Output: -



Experiment 4: - Graphical representation of unit ramp signal

AIM: - To write the Scilab code to find the unit ramp signal and sketch the output wave form.

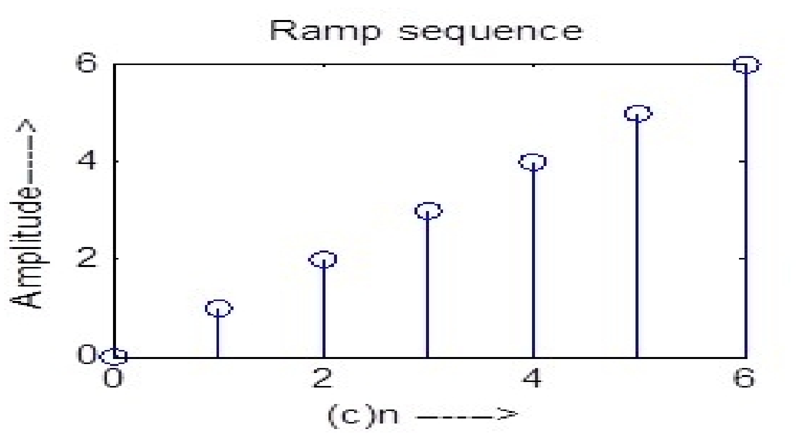
Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear;
clc;
close;
L = 4; // Upper limit
n = -L: L;
x = [ zeros (1, L) ,0: L];
a= gca ();
a.thickness = 2;
a.y_location = "middle ";
plot2d3 (‘gnn‘, n,x)
xtitle('Graphical representation of unit ramp signal, ' n ' , ' x [ n ] ' );
```

Output: -



Experiment 5: - Graphical representation of exponential signal

AIM: - To write the Scilab code to find the exponential signal and sketch the output wave form.

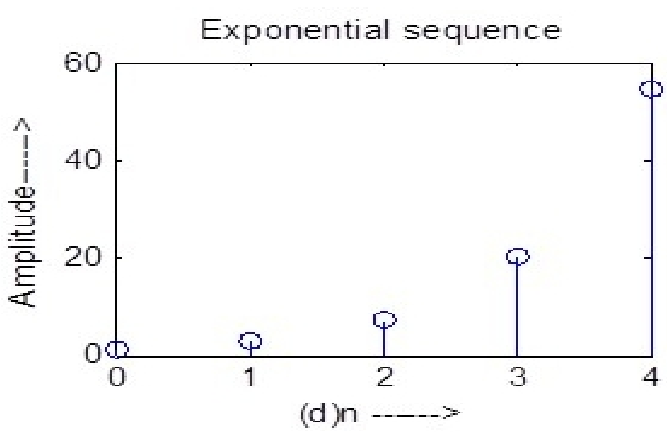
Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear;
clc ;
close ;
a =1.5;
n =1:10;
x = (a)^n;
a= gca ();
a.thickness = 2;
plot2d3 ( ' gnn ' ,n,x)
xtitle ( ' Graphical representation of exponential signal', ' n ' , ' x [ n ] ' );
```

Output: -



Experiment 6: - Graphical representation of exponential increasing- decreasing signal

AIM: - To write the Scilab code to find the exponential increasing- decreasing signal and sketch the output wave form.

Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear;
clc;
close;
a = -1.5;
n = 0:10;
x = (a)^n;
a= gca ();
a. thickness = 2;
a. x_location = " o r i g i n ";
a. y_location = " o r i g i n ";
plot2d3 ( ' gnn ' ,n,x)
xtitle ( ' Graphical representation of exponential increasing- decreasing signal', ' n ' , ' x [ n ] ' );
```

Experiment 7: - Graphical representation of even signal

AIM: - To write the Scilab code to find the even signal and sketch the output wave form.

Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear;
clc;
close;
n = -7:7;
x1 = [0 0 0 1 2 3 4];
x = [x1 ,5, x1(length (x1) : -1:1) ];
```

```
a= gca ();
a. thickness = 2;
a. y_location = " middle ";
plot2d3 ( ' gnn ' ,n,x)
xtitle ( ' Graphical representation of even signal', ' n ', ' x [ n ] ' );
```

Experiment 8: - Graphical representation of odd signal

AIM: - To write the Scilab code to find the odd signal and sketch the output wave form.

Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear ;
clc ;
close ;
n = -5:5;
x1 = [0 1 2 3 4 5];
x = [-x1($ : -1:2) ,x1 ];
a= gca ();
a. thickness = 2;
a. y_location = " middle ";
a. x_location = " middle "
plot2d3 ( ' gnn ' ,n,x)
xtitle ( ' Graphical representation of even signal', ' n ', ' x [ n ] ' );
```

Experiment 9:-Determine whether given signal is periodic or not

AIM: - To write the Scilab code to find the given signal is periodic or not

Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear ;
clc ;
close ;
t =0:0.01:10;
x1=cos (2* %pi *t /3) ;
subplot (1 ,2 ,1);
plot (t,x1);
xlabel ( ’ t ’ );
ylabel ( ’ x ( t ) ’ );
title ( ’CONTINUOUS TIME PLOT ’ );
n =0:0.2:10;
x2= cos (2* %pi *n /3) ;
subplot (1 ,2 ,2);
plot2d3 (n,x2);
xlabel ( ’ n ’ );
ylabel ( ’ x ( n ) ’ );
title ( ’DISCRETE TIME PLOT ’ );
```

Experiment 10: -Convolution of given sequences

AIM: - To write the Scilab code to find the convolution of a given signal x (n) = [1 2 1 1] , h (n) = [1 -1 1 -1]and sketch the output wave form.

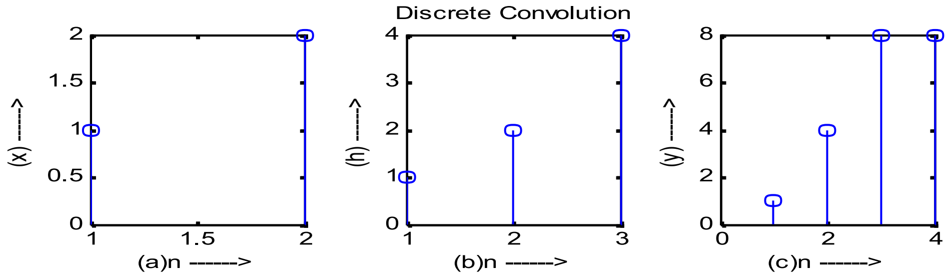
Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear;
clc;
close;
x = [1 2 1 1];
h =[1 -1 1 -1];
y= convol (x,h);
disp ( round (y))
```

Output: -



Experiment 11: - Cross correlation of given sequences

AIM: - To write the Scilab code to find the cross correlation of a given signal x (n) = [1 2 1 1] , h (n) = [1 1 2 1]and sketch the output wave form.

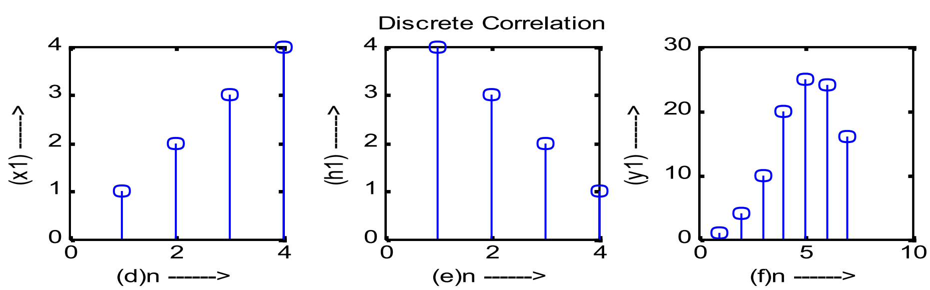
Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear;
clc;
close;
x = [1 2 1 1];
h = [1 1 2 1];
h1 = [1 2 1 1];
y= convol (x, h1);
11 disp (round (y));
```

Output:-



Experiment 12: - Plot Magnitude and Phase Response

AIM: - To write the Scilab code to find the magnitude and phase plot of a given system and sketch the output wave form.

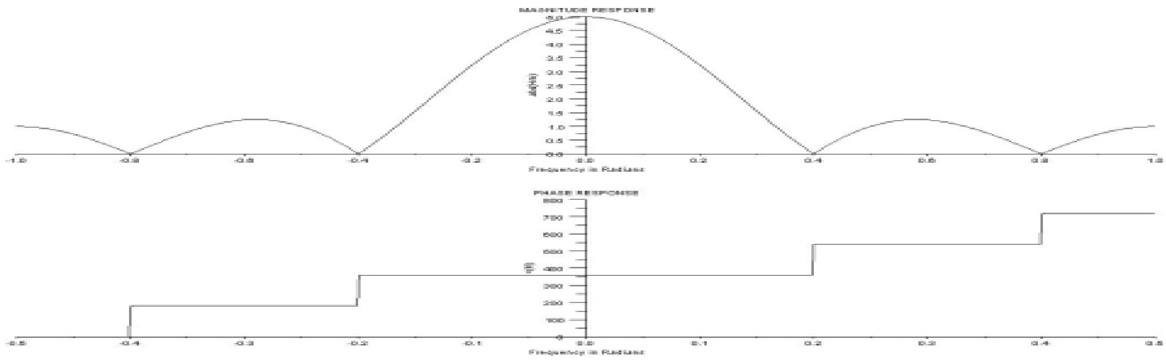
Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear ;
clc ;
close ;
w=- %pi :0.01: %pi ;
H =1+2* cos(w) +2* cos (2* w);
[ phase_H ,m]= phasemag (H);
Hm=abs(H);
a= gca ();
subplot (2 ,1 ,1);
a. y_location =" o r i g i n ";
plot2d (w/%pi ,Hm);
xlabel ( ' Frequency i n Radians ' )
ylabel ( ' abs (Hm) ' );
title ( 'MAGNITUDE RESPONSE ' );
subplot (2 ,1 ,2);
a= gca ();
a. x_location =" o r i g i n ";
a. y_location =" o r i g i n ";
plot2d (w /(2* %pi ),phase_H );
xlabel ( ' Frequency i n Radians ' );
ylabel ( ' <(H) ' );
title ( 'PHASE RESPONSE ' );
```

Output: -



Experiment 13: - Impulse Response of a given System

AIM: - To write the Scilab code to find the impulse response of a given system and sketch the output wave form.

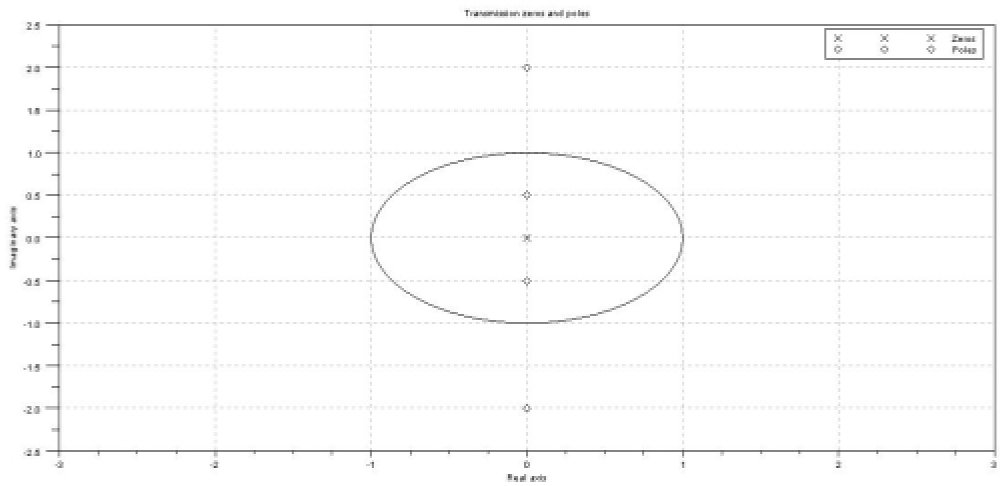
Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear;
clc;
close ;
z=%z;
a=z ^3+2*( z ^(2) ) -4*(z) +1;
b=z ^3;
h = ldiv (a,b ,4) ;
disp (h,"h(n)=");
```

Output:-



Experiment 14: -Z Transform of the Sequence a given sequence

AIM: - To write the Scilab code to calculate the z transform of a given sequence and sketch the output wave form.

Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear ;
clc ;
close ;
function [za ]= ztransfer ( sequence ,n)
z= poly (0, ' z ' , ' r ' )
za= sequence *(1/ z)^n'
endfunction
x1 =[2 -1 3 2 1 0 2 3 -1];
n = -4:4;
zz= ztransfer (x1 ,n);
disp (zz ,"Z-transform of sequence is : ");
disp ( 'ROC is the entire plane except z = 0 and z =%inf ' );
```

Experiment 15: - Inverse Z Transform of the Sequence a given sequence

AIM: - To write the Scilab code to calculate the inverse z transform of a given sequence and sketch the output wave form.

Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear;
clc ;
close ;
z=%z;
a =(2+2* z+z ^2) ;
b=z ^2;
h = ldiv (b,a ,6) ;
disp (h," First six values of h ( n )=");
```

Experiment 16: -DFT and IDFT of a Sequence

AIM: - To write the Scilab code to calculate the DFT of a given sequence x [n] = [1, 1 , 0 , 0] and IDFT of a Sequence Y[k] = [1 , 0 , 1 , 0] and sketch the output wave form.

Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

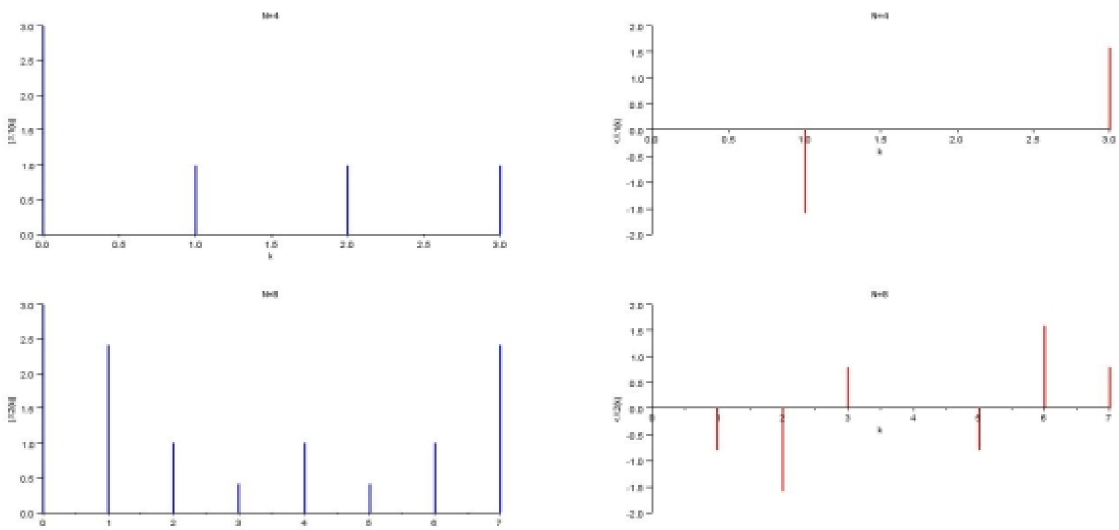
Source Code: -

x [n] = [1, 1 , 0 , 0] and IDFT of a Sequence Y[k] = [1 , 0 , 1 , 0]

```
clear ;
clc ;
close ;
```

```
x = [1 ,1 ,0 ,0];
//DFT Computation
X = fft (x , -1);
Y = [1 ,0 ,1 ,0];
//IDFT Computation
y = fft (Y , 1);
// Display sequence X[ k ] and y [ n ] in command window
disp (X,"X[ k]=");
disp (y,"y [ n]=");
```

Output: -



Experiment 17: - 8- point DFT of the Sequence

AIM: - To write the Scilab code to calculate the 8- point DFT of a given sequence x [n] = [1, 1 , 1 , 1 , 1 , 1 , 0 , 0]and sketch the output wave form.

Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear ;
clc ;
close ;
x = [1 ,1 ,1 ,1 ,1 ,1 ,0 ,0];
//DFT Computation
X = fft (x , -1);
// D i s p l a y s e q u e n c e X[ k ] i n command window
disp (X,"X[ k]= ");
```

Experiment 18: -Circular convolution of following sequences

AIM: - To write the Scilab code to calculate the circular convolution of a given sequence x [n] = [1 , 2 , 2 , 1 , 0] and Y[k]=exp (- j *4*pi *k /5) . X [k] and sketch the output wave form.

Algorithm: -

Step 1: - Start the program
Step 2: - Get the input for signal generation
Step 3: - Use the appropriate library function
Step 4: - Display the output and output wave form

Source Code: -

```
clear ;
clc ;
close ;
x=[1 ,2 ,2 ,1 ,0];
X= fft(x , -1);
k =0:1:4;
j= sqrt ( -1);
pi =22/7;
H= exp(-j *4* pi*k /5) ;
Y=H.*X;
//IDFT Computation
y= fft(Y ,1) ;
// D i s p l a y s e q u e n c e y [ n ] i n command window
disp ( round (y),"y [ n]= ");
// P l o t s
n =0:1:4;
a = gca ();
```

```
a. y_location = " o r i g i n ";
a. x_location = " o r i g i n ";
plot2d3 (n, round (y) ,5);
poly1 =a. children (1) . children (1) ;
poly1 . thickness =2;
xtitle ( ' P l o t o f s e q u e n c e y [ n ] ' , ' n ' , ' y [ n ] ' );
```