DIGITAL SIGNAL PROCESSING LABORATORY MANUAL

B.TECH (IV YEAR – I SEM) – R16

Department of Electronics and Communications

SIR C.R.REDDY COLLEGE OF ENGINEERING

(Approved by AICTE, New Delhi & permanently affiliated to JNTUK, Kakinada)
Ranked as “A” grade college by Govt. of Andhra Pradesh, “AA” grade by NPTEL
ELURU- 534007, WEST GODHAVARI DIST, ANDHRA PRADESH, INDIA
Phone no.:08812-230840, Visit us at http:// www.sircrrengg.ac.in.

Course Name: Digital Signal Processing Laboratory **Course code: C418.**

List of the Experiments / programs

To Student has to perform at least FOUR Experiments in each part

PART-1(SIGNALS)

1) Generation of discrete time signals for discrete signals
2) To verify the Linear Convolution
a) Using MATLAB
b) Using Code Composer Studio (CCS)
3) To verify the Circular Convolution for discrete signals
a) Using MATLAB
b) Using Code Composer Studio (CCS)
4) To find the addition of Sinusoidal Signals
5) To verify Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform(IDFT)
a) Using MATLAB
b) Using Code Composer Studio (CCS)
6) Transfer Function Stability Analysis: using pole-zero plot, bode plot, Nyquist plot, z-plane plot.

PART-2 (FILTERS)

7) Frequency Response of IIR low pass Butterworth Filter
8) Frequency Response of IIR high pass Butterworth Filter
9) Frequency Response of IIR low pass Chebyshev Filter
10) Frequency Response of IIR high pass Chebyshev Filter
11) Frequency Response of FIR low pass Filter using Rectangle Window
12) Frequency Response of FIR low pass Filter using Triangle Window

PART – 3(IMAGE PROCESSING)

13) An image processing in a false contouring system
14) To generate the histogram equalization to the image
15) To verify the Normalized Cross Correlation to the addition of noise and removal of noise using filters to an image.
16) Compute the edge of an image using spatial filters.
17) Perform the image motion blur and calculate PSNR to the noise image and also noise free image.
18) To verify the PSNR to the Second order Decomposition of Discrete Wavelet transforms and to the reconstructed image using inverse Discrete Wavelet transform

ELECTRONICS & COMMUNICATION ENGINEERING

Vision of the Department

To be a premier department in the region that nurtures individuals for acquisition of knowledge and skills with research orientation which suits the local and global needs of industry and society in the field of Electronics and Communication Engineering.

Mission of the Department

M1. To impart quality education and encourage research with an emphasis on application and innovation

M2. To cater the emerging societal needs through all-round development of students

M3. To enable individuals face the global competition

M4. To inculcate the importance of ethical and moral values in students

Program Educational Objectives (PEOs)

PEO No.	PEO Description
PEO 1	Develop a solid foundation in mathematics and sciences to solve electronics and communication engineering problems
PEO 2	Apply core engineering concepts to develop novel products and solutions demanded by modern industry
PEO 3	Pursue higher studies, research & development in science & technology and keep abreast of latest technological developments.
PEO 4	Inculcate professional and ethical attitude ,effective communication skills ,Team spirit & leadership qualities
PEO 5	Contribute to the needs of the society in solving technical problems using electronics communications engineering principles, tools and practices

Program Outcomes:

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, research literature, and analyse complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, society, 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 modelling 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 society 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 multi-disciplinary 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 multi-disciplinary 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.

PART-1(SIGNALS)

GENERATION OF DISCRETE-TIME SIGNALS

AIM:

Write a MATLAB program for the generation of unit impulse, unit step, ramp, exponential and Sinusoidal sequences.

PROGRAM:

```
%program for the generation of unit impulse sequence [δ (n)]

K=input ('enter the length of sequence of K :');
n= - K: 1: K;
x= [zeros (1, K), ones (1, 1), zeros (1, K)];
subplot (5, 1, 1);
stem (n, x);
xlabel('time index');
ylabel ('s(n)');
title ('unit impulse');

 %program for the generation of unit step sequence [u (n)]

L=input ('enter length of sequence of L :');
n= - L: 1: L;
x= [zeros (1, L), ones (1, L+1)];
subplot (5,1,2);
stem (n,x);
xlabel('time index');
ylabel('u(n)');
```

```
title ('unit step');
```

%program for the generation of exponential sequence x (n) =c (a) ^{n}

```
M=input ('enter sequence length of M :');
n=0: M;
B=input ('enter value of B :');
d=B;
C=input ('enter a value of C :');
a=C;
x=d*(a. ^n);
subplot (5,1,3);
stem (n,x);
xlabel('time index');
ylabel('e(n)');
```

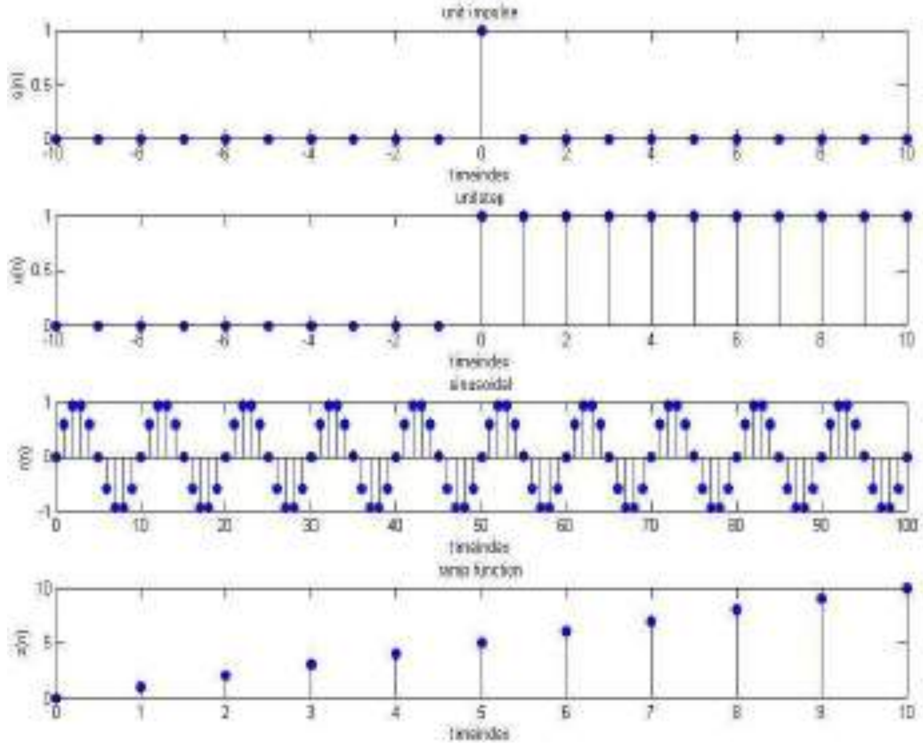
```
title ('exponential');

%program for the generation of sinusoidal sequence

N=input ('enter the length of N :');
n=0: N;
x=sin (2*pi/10*n);
subplot (5,1,4);
stem (n,x);
xlabel('time index');
ylabel('r(n)');
title ('sinusoidal');


%program for the generation of ramp sequence

O=input ('enter sequence length of O:');
l=0: O;
x=l;
subplot (5,1,5);
stem (l,x);
xlabel('time index');
ylabel('z(n)');
title ('ramp function');
```

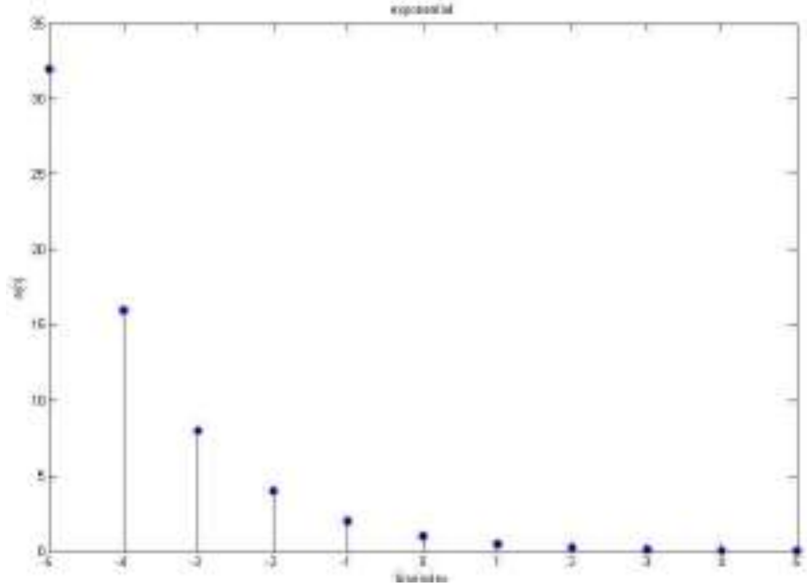
RESULT:

Given sequences are generated and plotted using MATLAB.

VIVA QUESTIONS:

1. What is the use of stem command?
2. What is the use of xlabel command?
3. What is the use of ylabel command?
4. What is the use of subplot command?
5. What is the use of title command?


unit impulse
x(n)
1
0.5
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
timeindex
unit step
x(n)
1
0.5
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
timeindex
sinusoidal
x(n)
1
0
-1
0
10
20
30
40
50
60
70
80
90
100
timeindex
ramp function
x(n)
10
5
0
0
1
2
3
4
5
6
7
8
9
10
timeindex


exponential
x(n)
35
30
25
20
15
10
5
0
-5
-4
-3
-2
-1
0
1
2
3
4
5
timeindex

LINEAR CONVOLUTION OF SEQUENCES

a) Write a MATLAB program to find the linear convolution of two sequences x (n) and h (n)

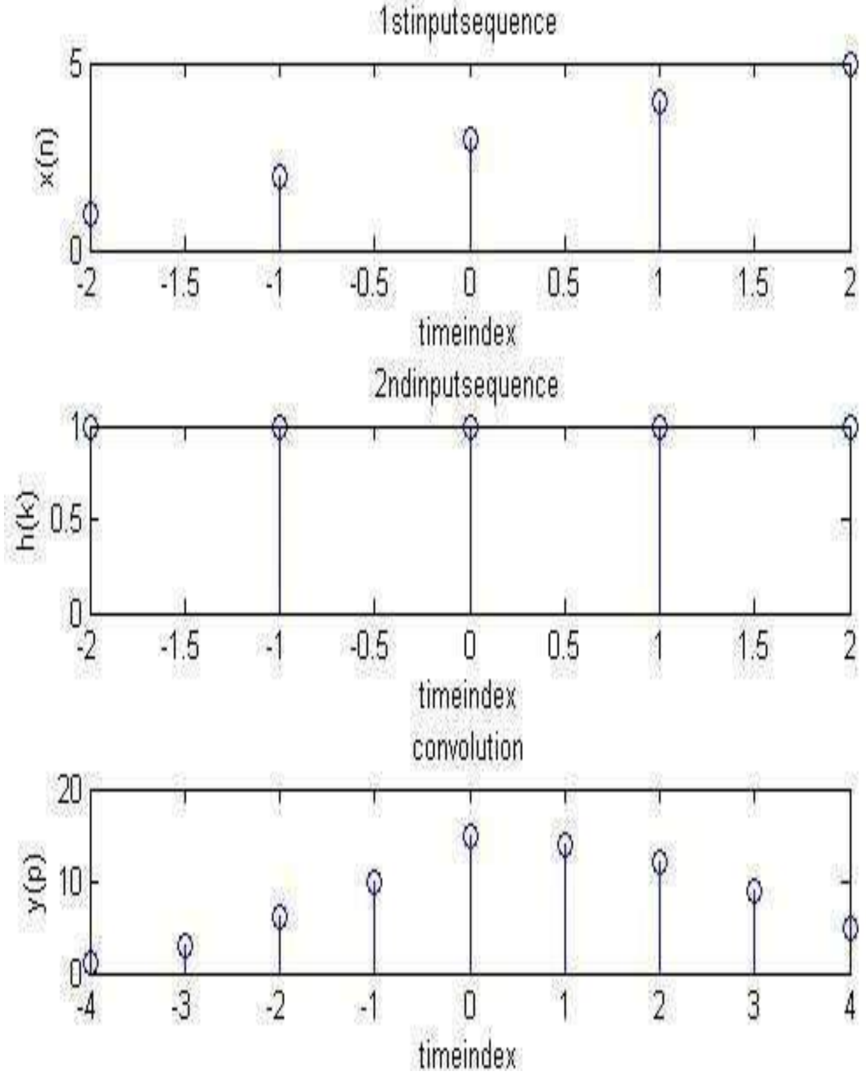
PROGRAM:

```
%program for the generation of first sequence [x (n)]
N=input ('enter length of the first sequence :');
n=-N: 1: N;
x=input ('enter the first sequence :');
subplot (3,1,1);
stem (n,x);
xlabel('time index');
ylabel('x(n)');
title ('1stinputsequence');

%program for the generation of second sequence [h (n)]
M=input ('enter length of the second sequence :');
k=-M: 1: M;
h=input ('enter the second sequence :');
subplot (3,1,2);
stem (k,h);
xlabel('time index');
ylabel('h(k)');
title ('2ndinputsequence');

%program for the generation of convolved sequence [y(n)=x(n)*h(n)]
p= (-N-M):1 :( M+N);
y=conv(x, h);
subplot (3,1,3);
stem (p,y);
xlabel('time index');
ylabel('y(p)');
title ('convolution');
```

RESULT: Linear convolution of two sequences x (n) and h (n) are determined using MATLAB.

VIVA QUESTIONS:

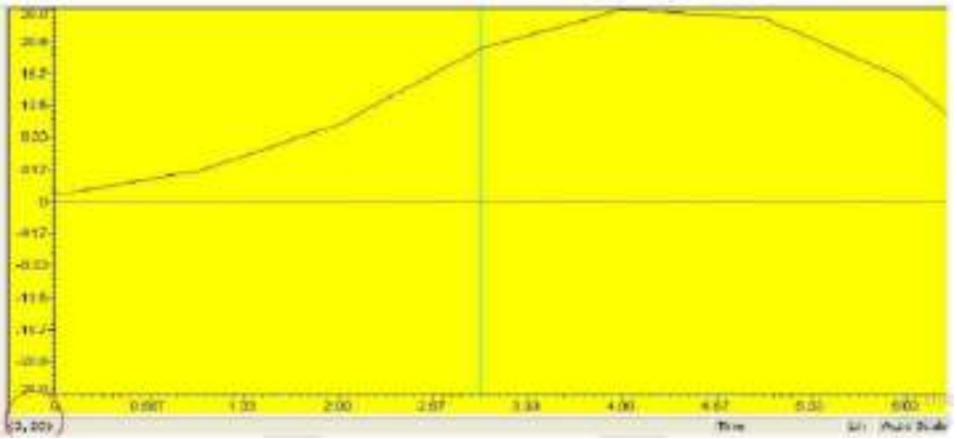
1. What is the use of input command?
2. What is the use of length command?
3. What is the use of ‘conv’ function?
4. What are the uses of MATLAB?
5. What are the different toolboxes in MATLAB?



b) Linear convolution using Code Composer Studio (CCS)

```
#include<stdio.h>
main()
{
int m=4;
int n=4;
int i=0,j;
int x[10]={1,2,3,4,0,0,0,0};
int h[10]={1,2,3,4,0,0,0,0};
int *y;
y=(int *)0x0000100;
for(i=0;i<m+n-1;i++)
{
y[i]=0;
for(j=0;j<=i;j++)
y[i]+=x[j]*h[i-j];
}
for(i=0;i<m+n-1;i++)
printf("%d\n",y[i]);
}
```

OUTPUT:



RESULT:

Linear convolution of two sequences x (n) and h (n) are determined using MATLAB and Code Composer Studio.

VIVA QUESTIONS:

1. What is the use of input command?
2. What is the use of length command?
3. What is the use of ‘conv’ function?
4. What are the uses of MATLAB?

3. CIRCULAR CONVOLUTION OF SEQUENCES

a) **AIM:** Write a MATLAB program to find the circular convolution of two sequences x (n) and h (n)

PROGRAM:

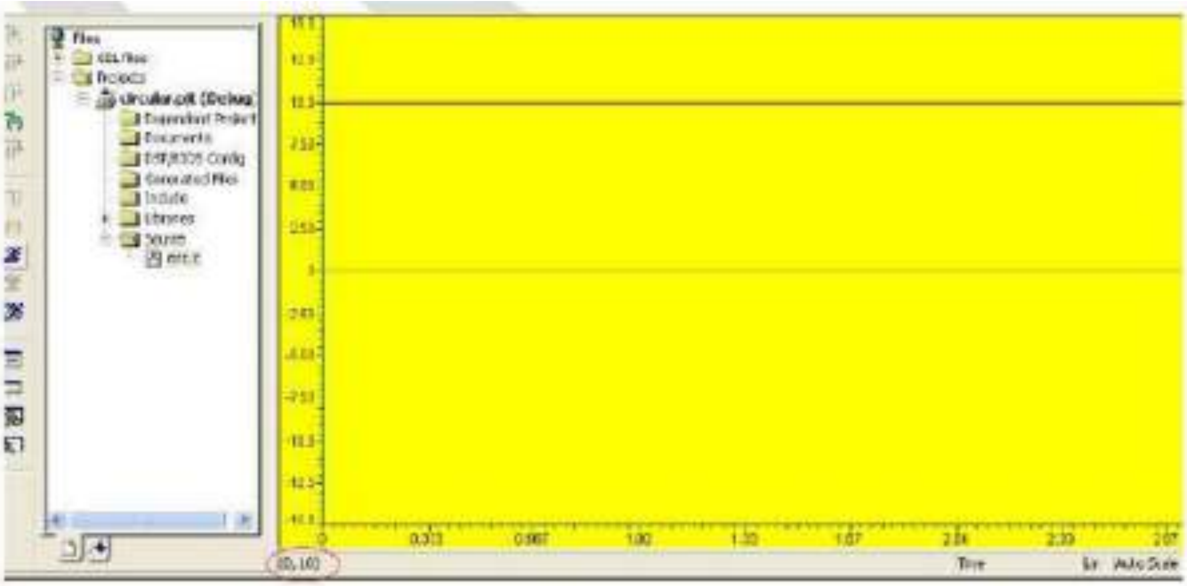
```
%Circular Convolution
clc;

clear all;

close all;

disp('circular convolution program');

x=input('enter i/p x(n):');

m=length(x);

h=input('enter i/p sequence h(n)');

n=length(h);

subplot(2,2,1), stem(x);

title('i/p sequencce x(n)is:');

xlabel('---->n');

ylabel('---->x(n)');grid;

subplot(2,2,2), stem(h);

title('i/p sequencce h(n)is:');

xlabel('---->n');

ylabel('---->h(n)');grid;

disp('circular convolution of x(n) & h(n) is y(n):');

if(m-n~=0)

if(m>n)

h=[h,zeros(1,m-n)];

n=m;

end

x=[x,zeros(1,n-m)];

m=n;end
```

```
y=zeros(1,n);
y(1)=0;
a(1)=h(1);
for j=2:n
a(j)=h(n-j+2);
end
%ciruclar conv
for i=1:n
y(1)=y(1)+x(i)*a(i);
end
for k=2:n
y(k)=0;
% circular shift
for j=2:n
x2(j)=a(j-1);
end
x2(1)=a(n);
for i=1:n
if(i<n+1)
a(i)=x2(i);
y(k)=y(k)+x(i)*a(i);
end
end
end
disp(y);
subplot(2,2,[3,4]),stem(y);
title('Circular convolution of x(n) & h(n) is:');
xlabel('---->n');
ylabel('---->y(n)');
grid;
```

b) Circular Convolution for discrete signals using Code Composer Studio (CCS)

```
#include<stdio.h>
intm,n,x[30],h[30],y[30],i,j,temp[30],k,x2[30],a[30];
void main()
{
int *y;
y=(int *)0x0000100;
printf(" enter the length of the first sequence\n");
scanf("%d",&m);
printf(" enter the length of the second sequence\n");
scanf("%d",&n);
printf(" enter the first sequence\n");
for(i=0;i<m;i++)
scanf("%d",&x[i]);
printf(" enter the second sequence\n");
for(j=0;j<n;j++)
scanf("%d",&h[j]);
if(m-n!=0) /*If length of both sequences are not equal*/
{
if(m>n) /* Pad the smaller sequence with zero*/
{
for(i=n;i<m;i++)
h[i]=0;
n=m;
}
for(i=m;i<n;i++)
x[i]=0;
m=n;
}
y[0]=0;
a[0]=h[0];
for(j=1;j<n;j++) /*folding h(n) to h(-n)*/
a[j]=h[n-j];
/*Circular convolution*/
for(i=0;i<n;i++)
y[0]+=x[i]*a[i];
for(k=1;k<n;k++)
{
y[k]=0;
/*circular shift*/
for(j=1;j<n;j++)
x2[j]=a[j-1];
x2[0]=a[n-1];
for(i=0;i<n;i++)
{
a[i]=x2[i];
y[k]+=x[i]*x2[i];
}}
/*displaying the result*/
printf(" the circular convolution is\n");
for(i=0;i<n;i++)
printf("%d ",y[i]);
}
```

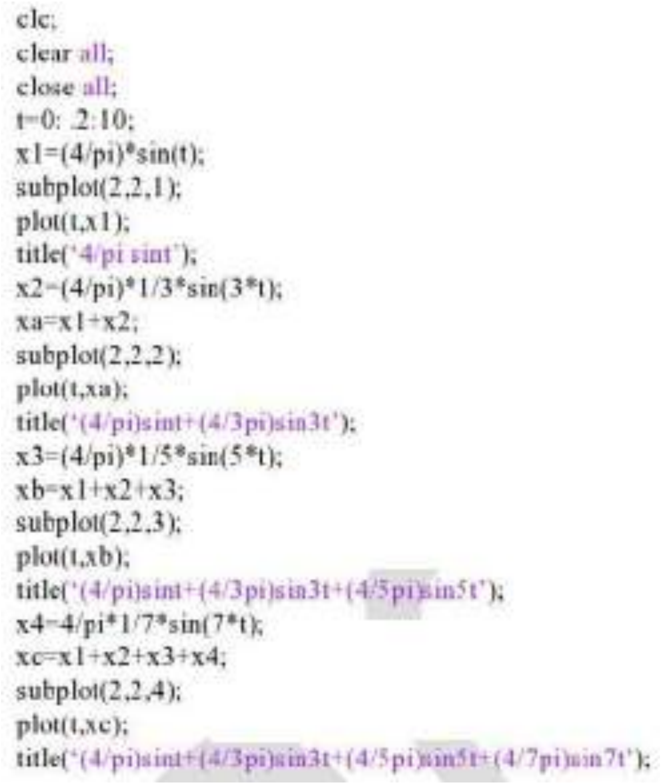
OUTPUT:



4. ADDITION OF SINUSIODAL SIGNALS

Aim: To Find the addition of Sinusoidal Signals

Program:



```
clc;
clear all;
close all;
t=0: .2:10;
x1=(4/pi)*sin(t);
subplot(2,2,1);
plot(t,x1);
title(‘4/pi sint’);
x2=(4/pi)*1/3*sin(3*t);
xa=x1+x2;
subplot(2,2,2);
plot(t,xa);
title(‘(4/pi)sint+(4/3pi)sin3t’);
x3=(4/pi)*1/5*sin(5*t);
xb=x1+x2+x3;
subplot(2,2,3);
plot(t,xb);
title(‘(4/pi)sint+(4/3pi)sin3t+(4/5pi)sin5t’);
x4=4/pi*1/7*sin(7*t);
xc=x1+x2+x3+x4;
subplot(2,2,4);
plot(t,xc);
title(‘(4/pi)sint+(4/3pi)sin3t+(4/5pi)sin5t+(4/7pi)sin7t’);
```

Output:

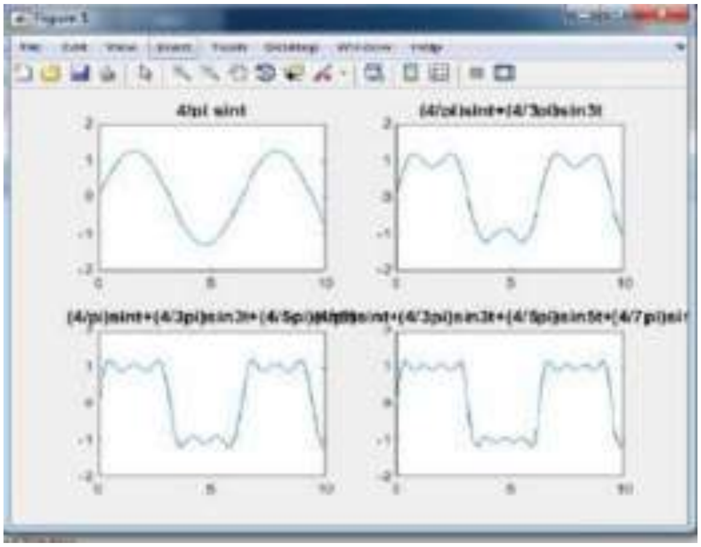


Figure 1
4/pi sint
(4/pi)sint+(4/3pi)sin3t
(4/pi)sint+(4/3pi)sin3t+(4/5pi)sin5t
(4/pi)sint+(4/3pi)sin3t+(4/5pi)sin5t+(4/7pi)si
2
1
0
-1
-2
0
5
10

Result:

5. To Verify Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT)

PROGRAM:

```
clc;
close all;
clear all;
xn=input('Enter the sequence x(n)'); %Get the sequence from user
ln=length(xn); %find the length of the sequence
xk=zeros(1,ln); %initialize an array of same size as that of input sequence
ixk=zeros(1,ln); %initialize an array of same size as that of input sequence
%DFT of the sequence
%-------------------------------------------------
for k=0:ln-1
for n=0:ln-1
xk(k+1)=xk(k+1)+(xn(n+1)*exp((i)*2*pi*k*n/ln));
end
end
%--------------------------------------------------
%Plotting input sequence
%-------------------------------------------------
t=0:ln-1;
subplot(221);
stem(t,xn);
ylabel ('Amplitude');
xlabel ('Time Index');
title('Input Sequence');
%----------------------------------------------------------
magnitude =abs(xk); % Find the magnitudes of individual DFT points
% plot the magnitude response
%-------------------------------------------------------
t=0:ln-1;
subplot(222);
stem(t,magnitude);
ylabel ('Amplitude');
xlabel ('K');
title('Magnitude Response');
%-------------------------------------------------------
phase=angle(xk); % Find the phases of individual DFT points % plot the magnitude
sequence
%-------------------------------------------------------
t=0:ln-1;
```

```
subplot(223);
stem(t,phase);
ylabel ('Phase');
xlabel ('K');
title ('Phase Response');
%-----------------------------------------------------------
%IDFT of the sequence
%-----------------------------------------------------------
for n=0:ln-1
for k=0:ln-1
ixk(n+1)=ixk(n+1)+(xk(k+1)*exp(i*2*pi*k*n/ln));
end
end
ixk=ixk./ln;
%-----------------------------------------------------------
%code block to plot the input sequence
%-----------------------------------------------------------
t=0:ln-1;
subplot(224);
stem(t,ixk);
ylabel ('Amplitude');
xlabel ('Time Index');
title ('IDFT sequence');
%-----------------------------------------------------
```

Output:

```
Xn=[1 2 3 4 5]
Xk = 15,-2.50+3.44i,-2.50+0.81i,-2.49-0.81i,-2.49-3.44i
```

RESULT:

VIVA QUESTIONS:

1. What is the use of ‘dftmtx’ function?
2. What is the use of ‘dct’ function?
3. What is the use of ‘fft’ function?
4. What is the use of ‘fft2’ function?
5. What is the use of ‘fftshift’function?

PART-2(FILTERS)

7). FREQUENCY RESPONSE OF IIR LOW PASS BUTTERWORTH FILTER

AIM:

Write a MATLAB program to design a Butterworth low pass digital filter for the given specifications and plot the Frequency response.

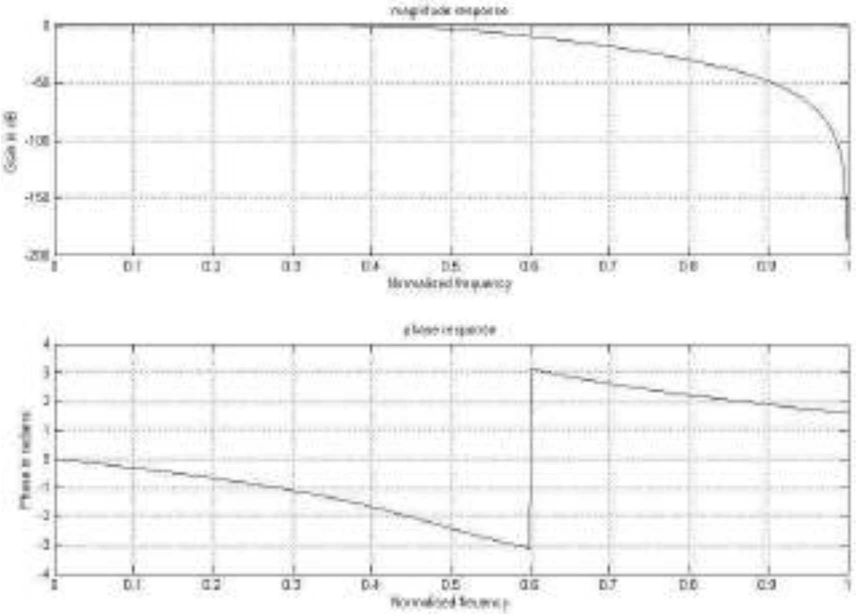
```
% To design a Butterworth low pass filter for the given specifications

clear all;
alphap=4;          % Pass band attenuation in dB
alphas=30;         % Stop band attenuation in dB
fp=400;             % pass band frequency in Hz
fs=800;            % stop band frequency in Hz
F=2000;             % Sampling frequency in Hz
omp=2*fp/F; oms=2*fs/F;
% To find cutoff frequency and order of the filter

[n,wn]=buttord(omp,oms,alphap,alphas)
% system function of the filter
[b,a]=butter (n, wn)
w=0:.01:pi;
[h,om]=freqz (b,a,w,'whole');
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);plot(om/pi,m);grid;
ylabel('Gain in dB');
xlabel('Normalised frequency');
title('magnitude response');
subplot(2,1,2);plot (om/pi,an);grid
ylabel('Phase in radians');
xlabel('Normalised frequency');
title('phase response');
```

RESULT:

Butterworth low pass digital filter is designed with the given specifications and obtained the frequency response using MATLAB.

VIVA QUESTION:

1. What is the use of ‘buttord’ function?
2. What is the use of ‘butter’ function?
3. What is the use of ‘freqz’ function?
4. What is the use of ‘residuez’ function?
5. What is the use of ‘ zplane’ function?



8). FREQUENCY RESPONSE OF IIR HIGH PASS BUTTERWORTH FILTER

AIM:

Write a MATLAB program to design a Butterworth high pass digital filter for the given specifications and plot the Frequency response.

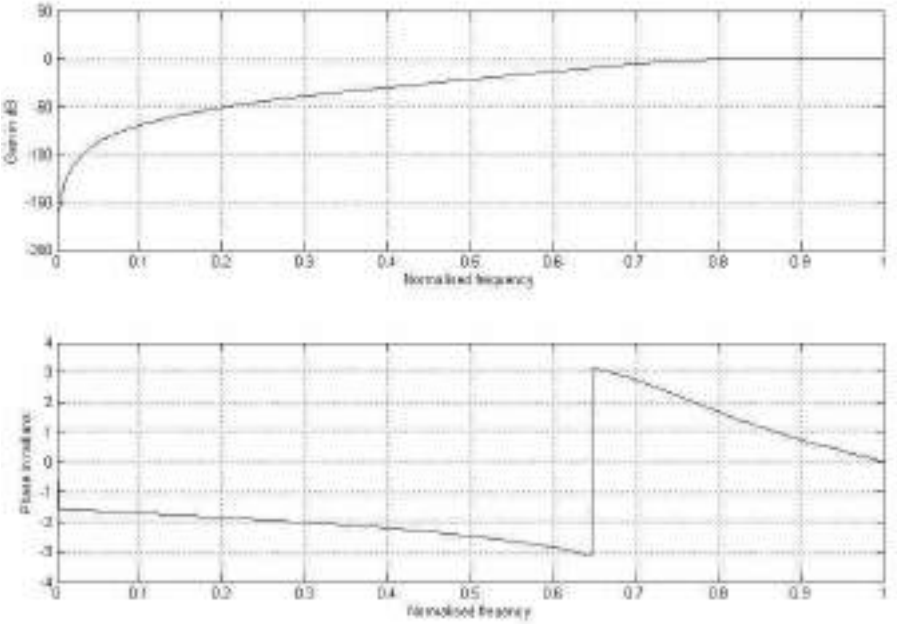
```
 % To design a Butterworth high pass filter for the given specifications
clear all;
alphap=4; %Pass band attenuation in dB
alphas=30; %Stop band attenuation in dB
fs=400; %pass band frequency in Hz
fp=800; %stop band frequency in Hz
F=2000; %Sampling frequency in Hz
omp=2*fp/F; oms=2*fs/F;
%To find cutoff frequency and order of the filter
[n,wn]=buttord(omp,oms,alphap,alphas)
% system function of the filter
[b,a]=butter(n,wn,'high')
w=0:.01:pi;
[h,om]=freqz(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);plot(om/pi,m);grid;
ylabel('Gain in dB');
xlabel('Normalised frequency');
subplot(2,1,2);plot(om/pi,an);grid;
ylabel('Phase in radians');
xlabel('Normalised frequency');
```

RESULT:

Butterworth High pass digital filter is designed with the given specifications
using MATLAB.

VIVA QUESTIONS:

1.What is the use of ‘real’ function?
2.What is the use of ‘freqs’ function?
3.What is the use of ‘poly’ function?
4.What is the use of ‘roots’ function?
5.What is the use of ‘ ceil’ function?



9). FREQUENCY RESPONSE OF IIR BAND PASS BUTTERWORTH FILTER

AIM:

Write a MATLAB program to design a Butterworth band pass digital filter
for the given specifications and plot the Frequency response.

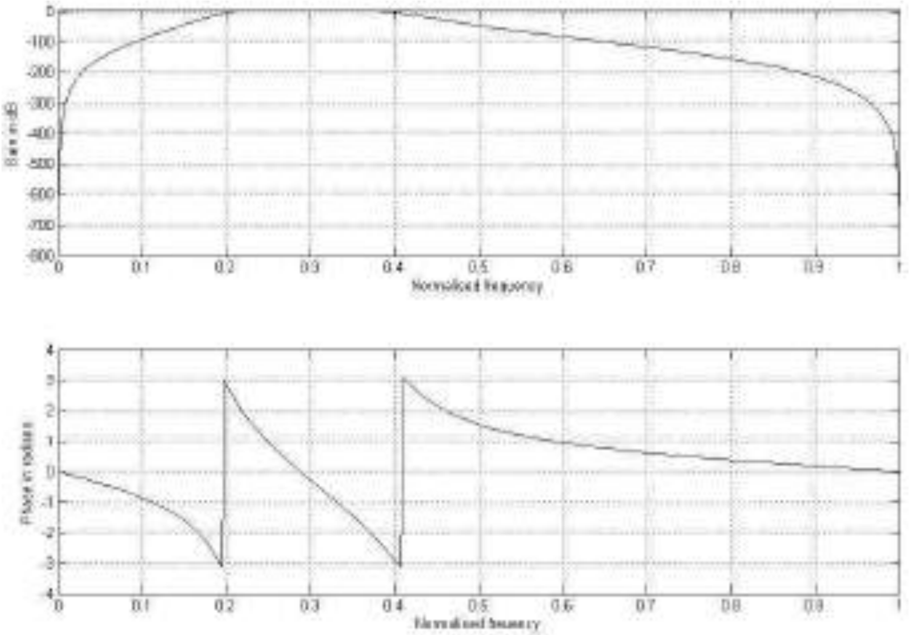
```
%To design a Butterworth Bandpass filter for the given specifications
clear all;
alphap=2; %Passband attenuation in dB
alphas=20; %Stopband attenuation in dB
wp=[.2*pi,.4*pi]; %passband frequency in radians
ws=[.1*pi,.5*pi]; %stopband frequency in  radians
%To find cutoff freency and order of the filter
[n,wn]=buttord(wp/pi,ws/pi,alphap,alphas)
% system function of the filter
[b,a]=butter(n,wn)
w=0:.01:pi;
[h,ph]=freqz(b,a,w);
m=20*log(abs(h));
an=angle(h);
subplot(2,1,1);plot(ph/pi,m);grid;
ylabel('Gain in dB');
xlabel('Normalised frequency');
subplot(2,1,2);plot (ph/pi,an);grid
ylabel('Phase in radians');
xlabel('Normalised freuency');
```

RESULT:

Butterworth Band pass digital filter is designed with the given
specifications using MATLAB.

VIVA QUESTION:

1. What is the use of ‘fliplr’ function?
2. What is the use of ‘dfs’ function?
3. What is the use of ‘dir2cas’ function?
4. What is the use of ‘cas2dir’ function?
5. What is the use of ‘dir2par’ function?



10). FREQUENCY RESPONSE OF IIR LOWPASS CHEBYSHEV FILTER

AIM:

Write a MATLAB program to design a Chebyshev lowpass digital filter for the given specifications and plot the Frequency response.

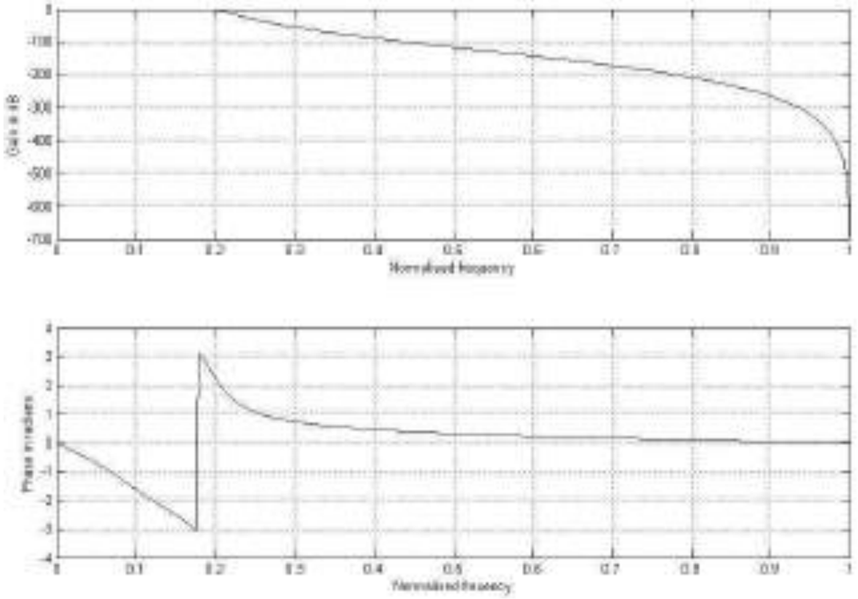
```
% To design a chebyshev type I lowpass filter for the given specifications
clear all;
alphap=1; %Passband attenuation in dB
alphas=15; %Stopband attenuation in dB
wp=.2*pi; %passband frequency in radians
ws=.3*pi; %stopband frequency in radians
%To find cutoff freency and order of the filter
[n,wn]=cheb1ord(wp/pi,ws/pi,alphap,alphas)
% system function of the filter
[b,a]=cheby1(n,alphap,wn)
w=0:.01:pi;
[h,ph]=freqz(b,a,w);
m=20*log(abs(h));
an=angle(h);
subplot(2,1,1);plot(ph/pi,m);grid;
ylabel('Gain in dB');xlabel('Normalised frequency');
subplot(2,1,2);plot(ph/pi,an);grid
ylabel('Phase in radians');xlabel('Normalised freuency');
```

RESULT:

chebyshev low pass digital filter is designed with the given specifications and obtained frequency response using MATLAB.

VIVA QUESTION:

1.What is the use of ‘cheb1ord’ function?
2.What is the use of ‘cheby1’ function?
3.What is the use of ‘cheby2’ function?
4.What is the use of ‘cheb2ord’ function?
5.What is the use of ‘chebiap’ function?



11). FREQUENCY RESPONSE OF IIR HIGH PASS CHEBYSHEV FILTER

AIM:

Write a MATLAB program to design a Chebyshev high pass digital filter for the given specifications and plot the Frequency response.

```
clc;
clear all;
close all;
ws=input('enter stop band freq');
wp=input('enter pass band freq');
rp=input('enter pass band ripple in db');
rs=input('enter stop band ripple in db');
r=input('enter peak to peak ripple in pass band in db');
[n,wn]=cheb1ord(wp,ws,rp,rs);
disp(n);
disp(wn);
[b,a] = cheby1(n,r,wn,'high');
disp('numerator polynomial');
disp(b);
disp('denominator polynomial');
disp(a);
w=0:0.1/pi:pi;
[h,om]=freqz(b,a,w);
m=abs(h);
gain=20*log10(m);
a=angle(h);
subplot(2,1,1)
plot(om,gain);grid;
title('frequency response')
xlabel('Normalized frequency');
ylabel('gain in db');
subplot(2,1,2)
plot(om,a);grid;
title('phase response');
```

```
xlabel('frequency');

ylabel('phase');
```

RESULT:

Chebyshev high pass digital filter is designed with the given specifications and obtained frequency response using MATLAB.

12). FREQUENCY RESPONSE OF IIR BAND PASS CHEBYSHEV FILTER

AIM:

Write a MATLAB program to design a Chebyshev band pass digital filter for the given specifications and plot the Frequency response.

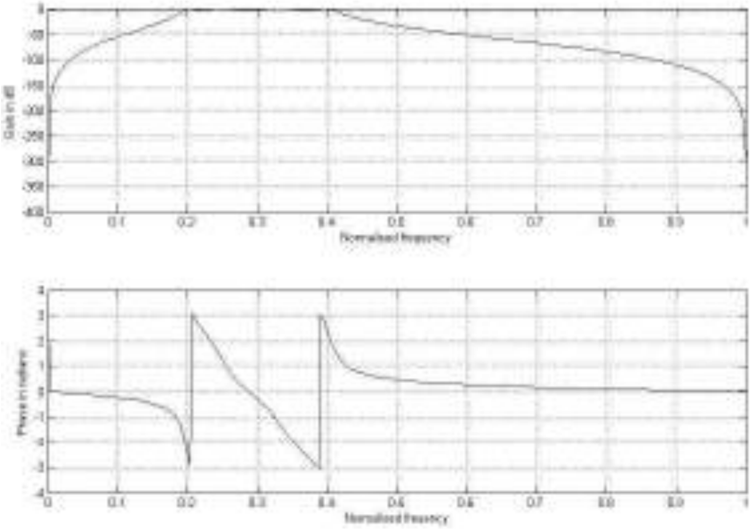
```
% To design a chebyshev type I band pass filter for the given
specifications
clear all;
alphap=2; %Passband attenuation in dB
alphas=20; %Stopband attenuation in dB
wp=[.2*pi,.4*pi]; %passband frequency in radians
ws=[.1*pi,.5*pi]; %stopband frequency in  radians
%To find cutoff freency and order of the filter
 [n,wn]= cheb1ord (wp/pi,ws/pi,alphap,alphas)
% system function of the filter
[b,a]=cheby1(n,alphap,wn)
w=0:.01:pi;
[h,ph]=freqz(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);plot(ph/pi,m);grid;
ylabel('Gain in dB');xlabel('Normalised frequency');
subplot(2,1,2);plot(ph/pi,an);grid;
ylabel('Phase in radians');xlabel('Normalised freuency');
```

RESULT:

chebyshev band pass digital filter is designed with the given specifications and obtained frequency response using MATLAB.

VIVA QUESTIONS:

1.What is the use of ‘cheby1(n,Rp,wn,’bandpass’)’ function?
2.What is the use of ‘cheby1(n,Rp,wn,’high’)’’ function?
3.What is the use of ‘cheby2ord’ function?
4.What is the use of ‘cheby2’ function?
5.What is the use of ‘ellipord’ function?



13). FREQUENCY RESPONSE OF FIR LOW PASS FILTER USING RECTANGULAR WINDOW

AIM:

Write a MATLAB program to design and obtain frequency response of a Low pass FIR filter using Rectangular and Hamming windows.

```
% To design a 25-tap lowpass filter with cutoff frequency of 0.5pi radians using
% rectangular and Hamming windows and plot their frequency response
clear all;
wc=.5*pi;

N=25;alpha=(N-1)/2;eps=.001;
n=0:1:N-1;

hd=sin(wc*(n-alpha+eps))./(pi*(n-alpha+eps));

wr=boxcar(N);%Rectangular window sequence

 hn=hd.*wr';%Filter coeffcients

w=0:.01:pi;

h=freqz(hn,1,w);

plot(w/pi,abs(h));
hold on

wh=hamming(N);%Hamming window sequence
hn=hd.*wh';%filter coefficients

w=0:.01:pi;

h=freqz(hn,1,w);
plot(w/pi,abs(h),'-.');grid;

xlabel('normalised frequency\omega/\pi');
ylabel('Magnitude');hold off
```

RESULT:

Low pass FIR filter is designed using Rectangular and Hamming windows and frequency response is obtained in MATLAB.

VIVA QUESTIONS:

1.What is the use of ‘boxcar’ function?
2.What is the use of ‘hamming’ function?
3.What is the use of ‘hanning’ function?
4.What is the use of ‘fir1’ function?
5.What is the use of ‘fir2’ function?

14). FREQUENCY RESPONSE OF FIR LOW PASS FILTER USING TRIANGLE WINDOW

AIM:

Write a MATLAB program to design and obtain frequency response of a high pass FIR filter using Triangle window.

```
clc;
clear all;
close all;
format long;
rp=input('enter pass band ripple in db');
rs=input('enter stop band ripple in db');
fs=input('enter stop band freq');
fp=input('enter pass band freq');
f=input('ente sampling freq');
wp=2*(fp/f);
ws=2*(fs/f);
num=-20*log10(sqrt(rp*rs))-13;
dem=14.6*(fs-fp)/f;
n=ceil(num/dem);
n1=n+1;
if(rem(n,2)~=0)
n1=n;
n=n-1;
end
y=triang(n1);
%lpf%
b=fir1(n,wp,y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,1);
```

```
plot(o/pi,m);
title('lpf');
ylabel('gain in db');
xlabel('normalized frequency');
%hpf%
b=fir1(n,wp,'high',y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,2);
plot(o/pi,m);
title('hpf');
ylabel('gain in db');
xlabel('normalized frequency');
%bpf%
wn=[wp,ws];
b=fir1(n,wn,'bandpass',y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,3);
plot(o/pi,m);
title('bpf');
ylabel('gain in db');
xlabel('normalized frequency');
%bsf%
b=fir1(n,wn,'stop',y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,4);
plot(o/pi,m);
title('bsf');
ylabel('gain in db');
```

```
xlabel('normalized frequency');
```

RESULT:

Low pass FIR filter is designed using Triagle window and frequency response is obtained in MATLAB.

PART-3(IMAGE PROCESSING)

15. An image processing in a false contouring system

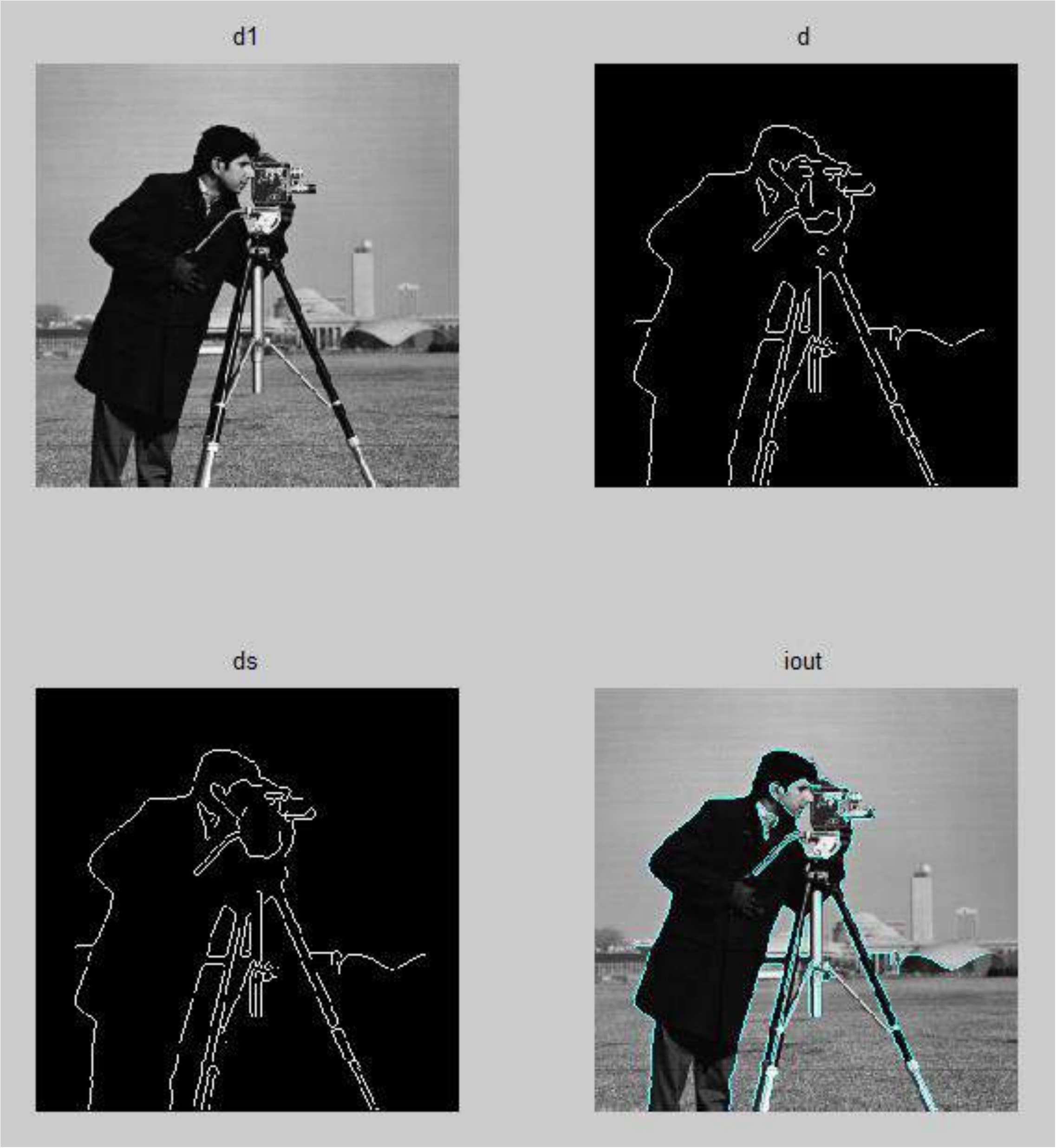
Aim: An image processing in a false contouring system

Software: MATLAB

Program:

```
d1 = double(imread('cameraman.tif'))./255;  % Load the image, scale from 0 to 1
subplot(2, 2, 1); imshow(d1); title('d1');  % Plot the original image
d = edge(d1, 'canny', .6);                  % Perform Canny edge detection
subplot(2, 2, 2); imshow(d); title('d');    % Plot the edges
ds = bwareaopen(d, 40);                     % Remove small edge objects
subplot(2, 2, 3); imshow(ds); title('ds');  % Plot the remaining edges
iout = d1;
BW = ds;
iout(:, :, 1) = iout;                       % Initialize red color plane
iout(:, :, 2) = iout(:, :, 1);              % Initialize green color plane
iout(:, :, 3) = iout(:, :, 1);              % Initialize blue color plane
iout(:, :, 2) = min(iout(:, :, 2) + BW, 1.0);   % Add edges to green color plane
iout(:, :, 3) = min(iout(:, :, 3) + BW, 1.0);   % Add edges to blue color plane
subplot(2, 2, 4); imshow(iout); title('iout');  % Plot the resulting image
```

output:


d1
d
ds
iout

16. To generate the histogram equalization to the image

Aim: To generate the histogram equalization to the image

Program:

```
clc;

clear all;

close all;

imgetfile;

u=imread(ans);

o=rgb2gray(u);

imshow(o);

figure;

imhist(o);

 i=histeq(o);

figure;

imshow(i);

figure;

imhist(i);
```

Result: In this experiment provides for thresholding an image and the evaluation of its histogram using Histogram equalization. This experiment illustrates the relationship among the intensities (gray levels) of an image and its histogram.

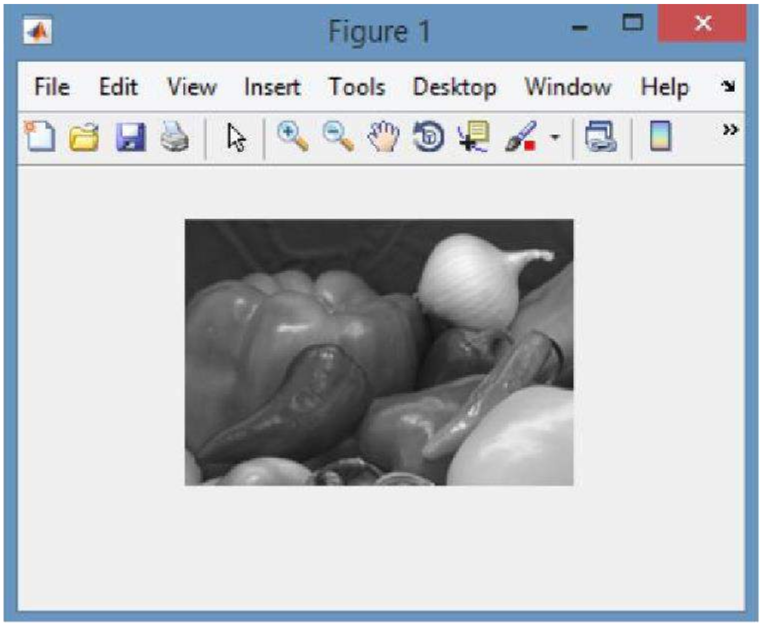


Fig 1.1: Gray Scale Image

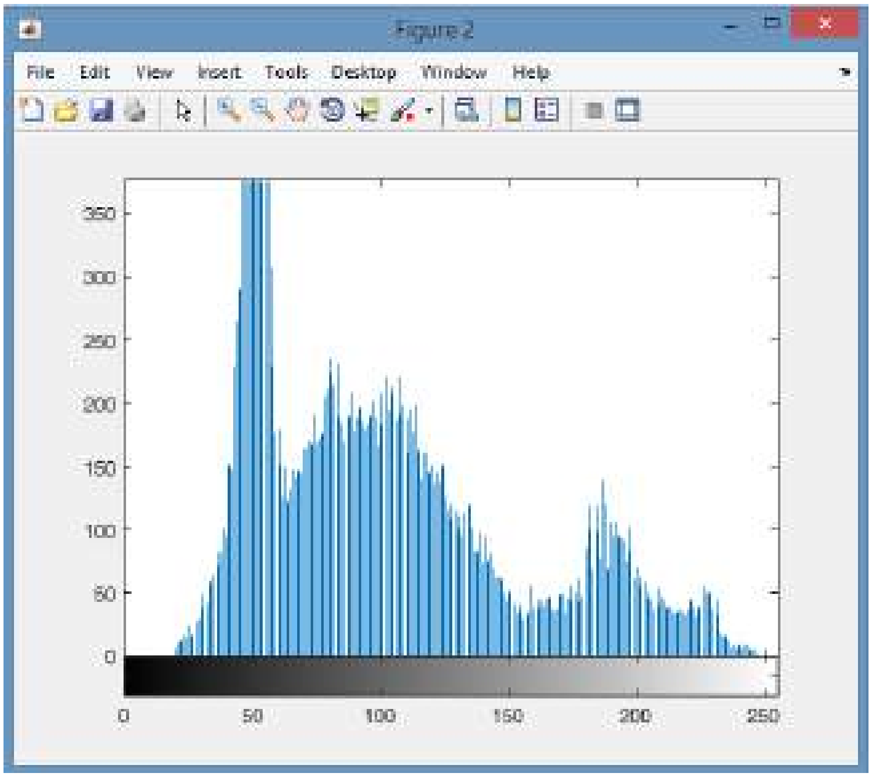


Fig 1.2: Histogram of Original Image

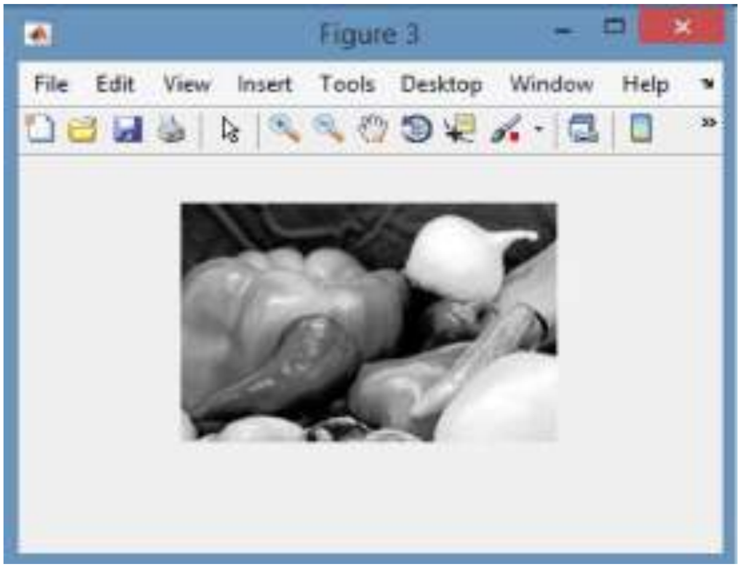


Fig 1.3: Image after Applying Histogram Equalization

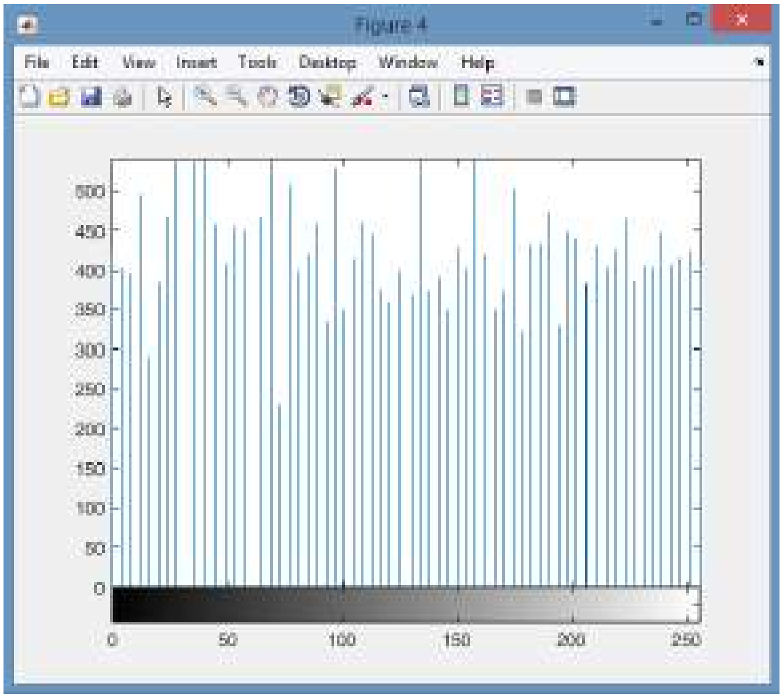


Fig 1.4: Histogram equalization

Viva questions:

1. What is histogram equalization in digital image processing?
2. Why histogram equalization is needed?
3. What happens if histogram equalization is applied twice?
4. Why histogram is used in image processing?
5. What is histogram equalization in Matlab?
6. What is entropy image processing?
7. Why is a histogram useful?
8. What is the advantage of histogram?
9. What is contrast in image processing?
10. How do you make a histogram graph?

17. To verify the normalized cross correlation to the addition of noise and removal of noise using filters to an image

```
A = rgb2gray(imread('onion.png'));
A_noise=imnoise(A,'gaussian');
%Prewitt_A_noise=edge(A_noise,'Prewitt');
imshow(A_noise);
%imshow(Prewitt_A_noise);
%aa=A+A_noise;
Filtered_snp=medfilt2(A_noise,[3 3]);
imshow(Filtered_snp);
aa1=normxcorr2(A_noise,Filtered_snp);
imshow(aa1)
```

18. Compute the Edge of an image using spatial filters

Spatial filtering term is the filtering operations that are performed directly on the pixels of an image. The process consists simply of moving the filter mask from point to point in an image. At each point let (x,y), the response of the filter at that point is calculated using a predefined relationship. The spatial filtering can be mainly of two types, Linear spatial filtering and Non-linear spatial filtering.

Aim: To compute the edge of an image using spatial filtering

```
clc;

clear;

x=imread('cameraman.TIF');

% --------filter coeff.-----------%

f1=[1 1 1;1 1 1;1 1 1];

f2=[1 0 -1;1 0 -1;1 0 -1];

f3=[1 1 1;0 0 0;-1 -1 -1];

% -------filtering-----------%

x1=x(1:256,1:256);

y1=conv2(x1,f1);

M1=max(max(y1));

y1=255.*(y1./M1);


y2=conv2(x1,f2);

M2=max(max(y2));

y2=255.*(y2./M2);


y3=conv2(x1,f3);

M3=max(max(y3));

y3=255.*(y2./M3);


subplot(2,2,1);imshow(x);title('IMAGE ')

subplot(2,2,2);imshow(uint8((y1)));title('f1- IMAGE ')

subplot(2,2,3);imshow(uint8((y2)));title('f2- IMAGE ')

subplot(2,2,4);imshow(uint8((y3)));title('f3-IMAGE ')
```

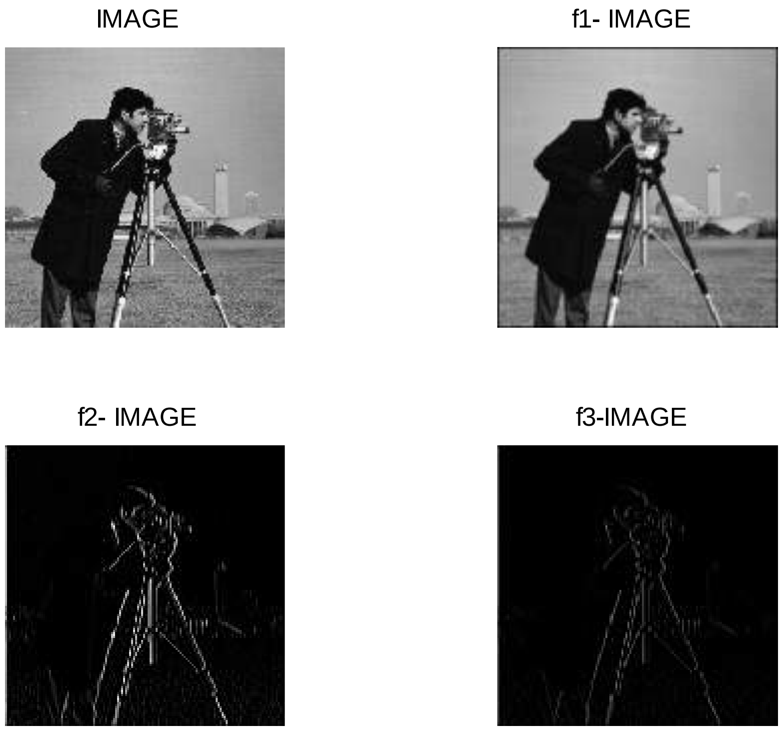


IMAGE
f1- IMAGE
f2- IMAGE
f3-IMAGE