Jawaharlal Nehru Engineering College

Laboratory Manual

Digital Signal Processing

For

Final Year (EEP) Students

Manual made by

Prof. J.R.Rana

FOREWORD

It is my great pleasure to present this laboratory manual for final year **ELECTRICAL ELECTRONIC & POWER** engineering students for the subject of Digital Signal Processing. Keeping in view the vast coverage required for visualization of concepts of Digital Signal Processing with simple language.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.

Faculty members are also advised that covering these aspects in initial stage itself, will greatly relive them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear.

H.O.D. (EEP)

LABORATORY MANUAL CONTENTS

This manual is intended for the final year students of **ELECTRICAL ELECTRONIC & POWER** engineering branch in the subject of Digital Signal Processing. This manual typically contains practical/Lab Sessions related to Digital Signal Processing covering various aspects related to the subject to enhance understanding.

Although, as per the syllabus, only descriptive treatment is prescribed, we have made the efforts to cover various aspects of electrical machine subject covering types of different Digital Signal Processing techniques, Transforms and applications will be complete in it to make it meaningful, elaborative understandable concepts and conceptual visualization.

Students are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions

Prof. J.R.Rana

SUBJECT INDEX

1. Do's and Don'ts

2. Lab exercise:

1) To represent basic signals (Unit Step, Unit Impulse, Ramp, Exponential, Sine and Cosine).

2) To develop the program for discrete convolution

3) To develop the program for discrete correlation.

4) To verify stability test.

5) To verify sampling theorem.

6) To design FIR Filter using window technique.

7) To design Digital IIR filter.

8) To design a program to compare direct realization values of digital IIR filter.

09) To design a program for computing inverse z transforms.

3. Conduction of Viva-Voce Examination

4. Evaluation and Marking Systems

1. DOs and DON'Ts:

DO's in Laboratory:

1. Use this software by using license key provided at the main server.
2. Any crash, invalid debugs may hang the software. In such cases, please wait for that program to respond otherwise this will directly delete the files if not saved properly.

DONT's in Laboratory:

1. Don't disturb the standard layout of the software.
2. Don't disturb the license settings.

Instructions for Laboratory Teachers:

1. Submission related to whatever lab work has been completed should be done during the next lab session. The immediate arrangements for printouts related to submission on the day of practical assignments.

2. Students should be taught for taking the observations /readings of Simulation output/script debugs under the observation of lab teacher.

3. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

2. Lab Exercise:

Practical 1 : (2 Hours):

TO REPRESENT BASIC SIGNALS (UNIT STEP, UNIT IMPULSE, RAMP, EXPONENTIAL, SINE AND COSINE).

AIM: To simulate basic signals (Unit Step, Unit Impulse, Ramp, Exponential, Sine and Cosine).

APPARATUS:

1) MATLAB Software
2) Digital Signal Processing Toolbox

THEORY:

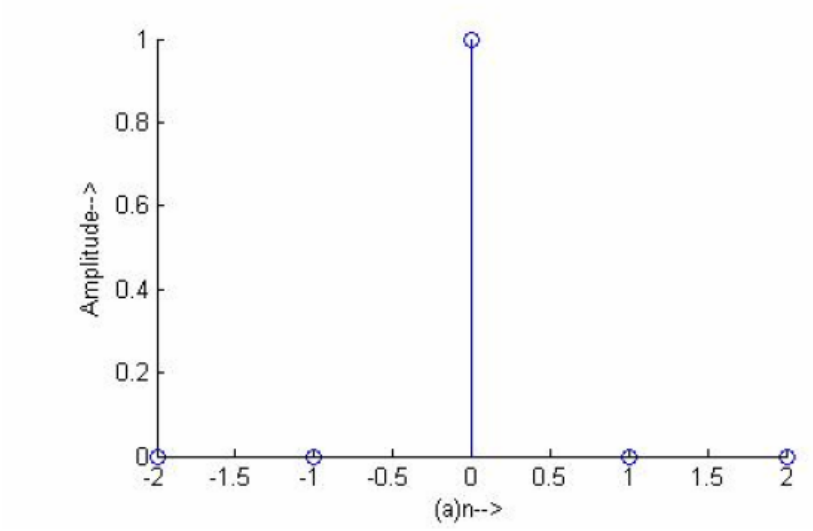
1. What are the basic signals in DSP and how they are mathematically represented?

MATLAB SCRIPTS FOR THE BASIC SEQUENCES:

Unit Impulse: -

```
% Program for the generation of unit impulse signal
clc;
clear all;
close all;
t = -2:1:2;
y=[zeros(1,2),ones(1,1),zeros(1,2)];
subplot (2,2,1);
stem (t,y);
ylabel ('Amplitude-->');
xlabel ('(a)n-->');
```

Output: -

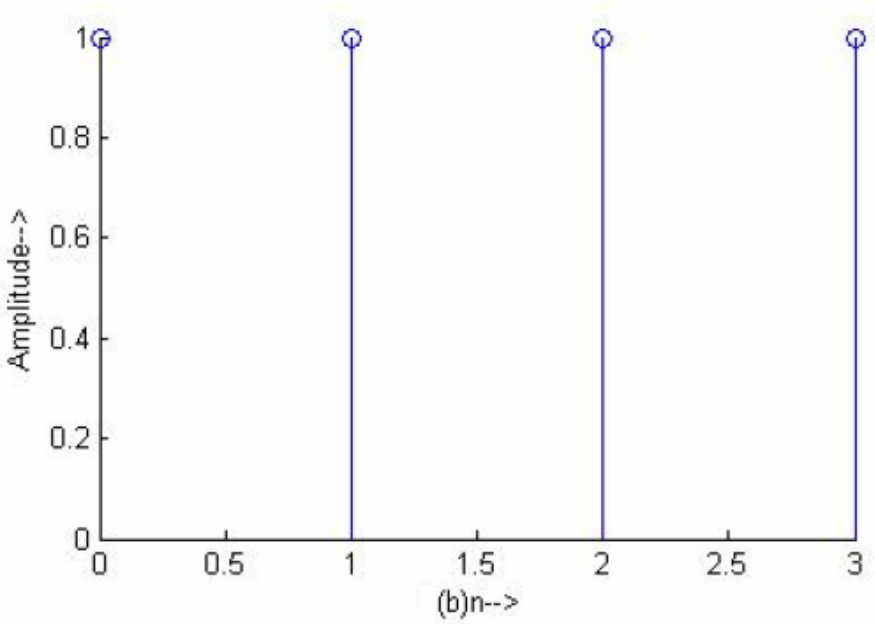


Unit Step Sequence: -

```
% Program for the generation of unit step sequence [u(n)-u(n-N)]
n=input ('Enter the N value');
t=0:1:n-1;
y1=ones(1,n);
subplot(2,2,2);
stem(t,y1);
ylabel('Amplitude-->');
xlabel('(b)n-->');
```

Output: -

Enter the N value4

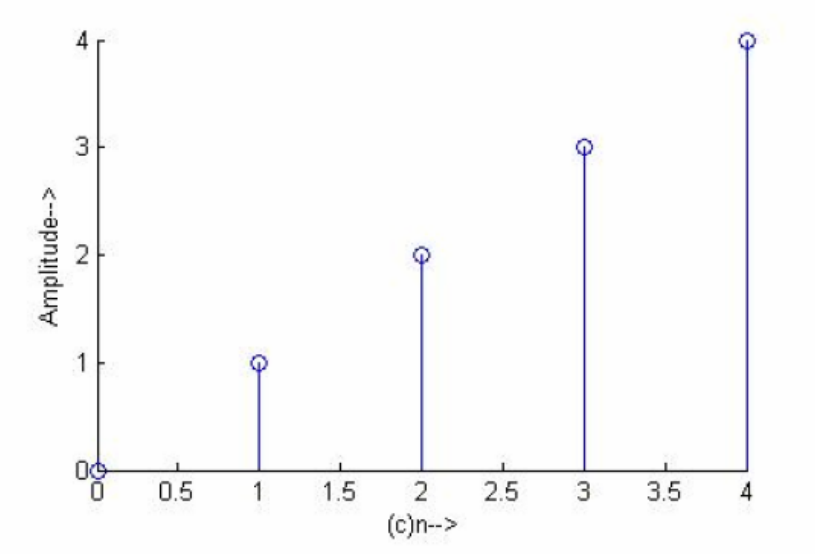


Ramp Sequence: -

```
% Program for the generation of ramp sequence
n1=input ('Enter the length of ramp sequence');
t=0:n1;
subplot(2,2,3);
stem(t,t);
ylabel('Amplitude-->');
xlabel('(c)n-->');
```

Output: -

Enter the length of rampsequence4

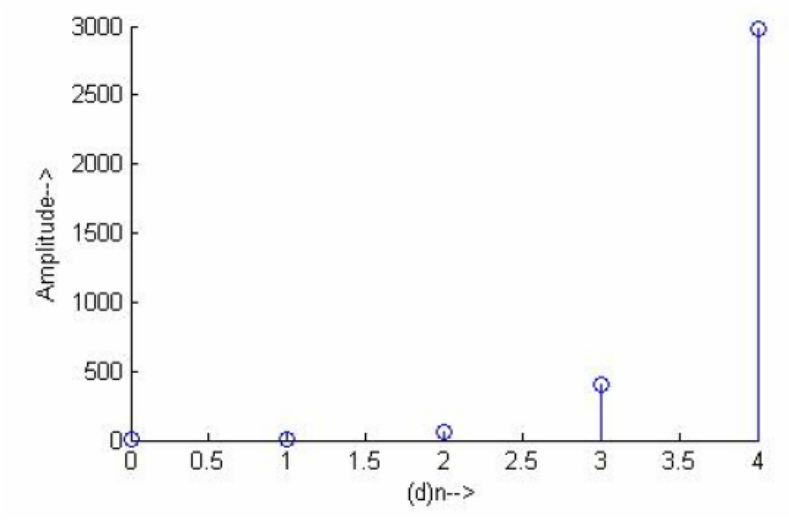


Exponential Sequence: -

```
% Program for the generation of exponential sequence
n2=input ('Enter the length of exponential sequence');
t=0:n2;
a=input('Enter the (a) value');
y2=exp(a*t);
subplot(2,2,4);
stem(t,y2);
ylabel('Amplitude-->');
xlabel('(d)n-->');
```

Output: -

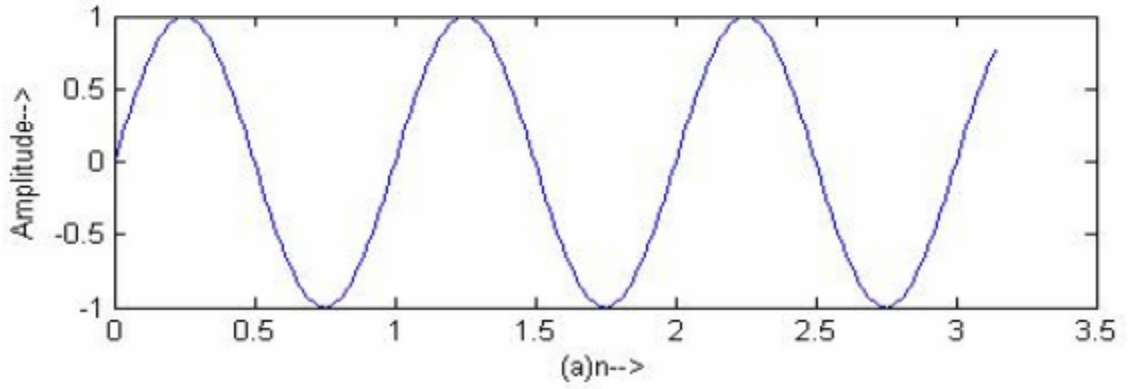
Enter the length of exponential sequence4
Enter the (a) value2



Sine Sequence: -

```
% Program for the generation of sine sequence
t=0:.01:pi;
y=sin(2*pi*t);
figure(2);
subplot(2,1,1);
plot(t,y);
ylabel('Amplitude-->');
xlabel('(a)n-->');
```

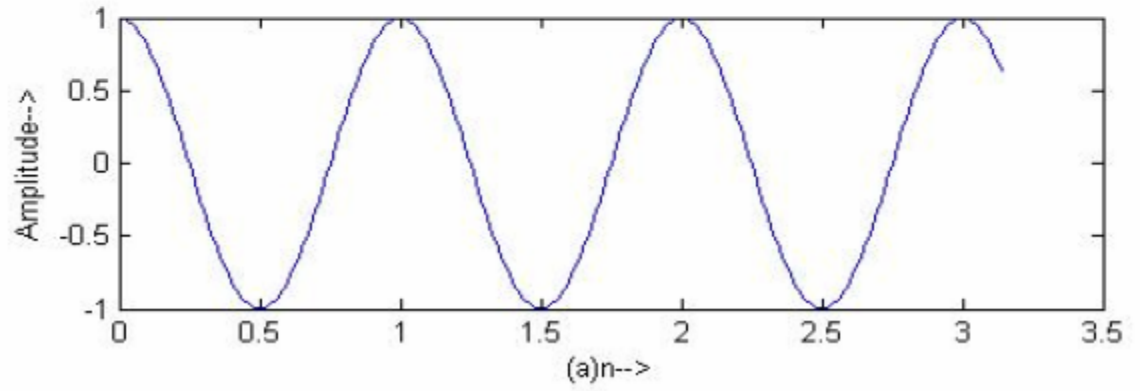
Output: -



Cosine Sequence: -

```
% Program for the generation of cosine sequence
t=0:.01:pi;
y=cos(2*pi*t);
figure(2);
subplot(2,1,1);
plot(t,y);
ylabel('Amplitude-->');
xlabel('(a)n-->');
```

Output: -



CONCLUSION:

Any type of the complex waveform can be represented in Digital Signal Processing mathematically as the combination of these basic signals, which helps for understanding the signal behavior for the analysis.

Practical 2 : (2 Hours):

TO DEVELOP THE PROGRAM FOR DISCRETE CONVOLUTION

AIM: To develop MATLAB program for evaluation of discrete convolution.

APPARATUS:

1) MATLAB Software
2) Digital Signal Processing Toolbox

THEORY:

1. What is convolution of signals?
2. How convolutions are represented?
2. Is it equivalent to the multiplication of two sequences?

MATLAB SCRIPT:

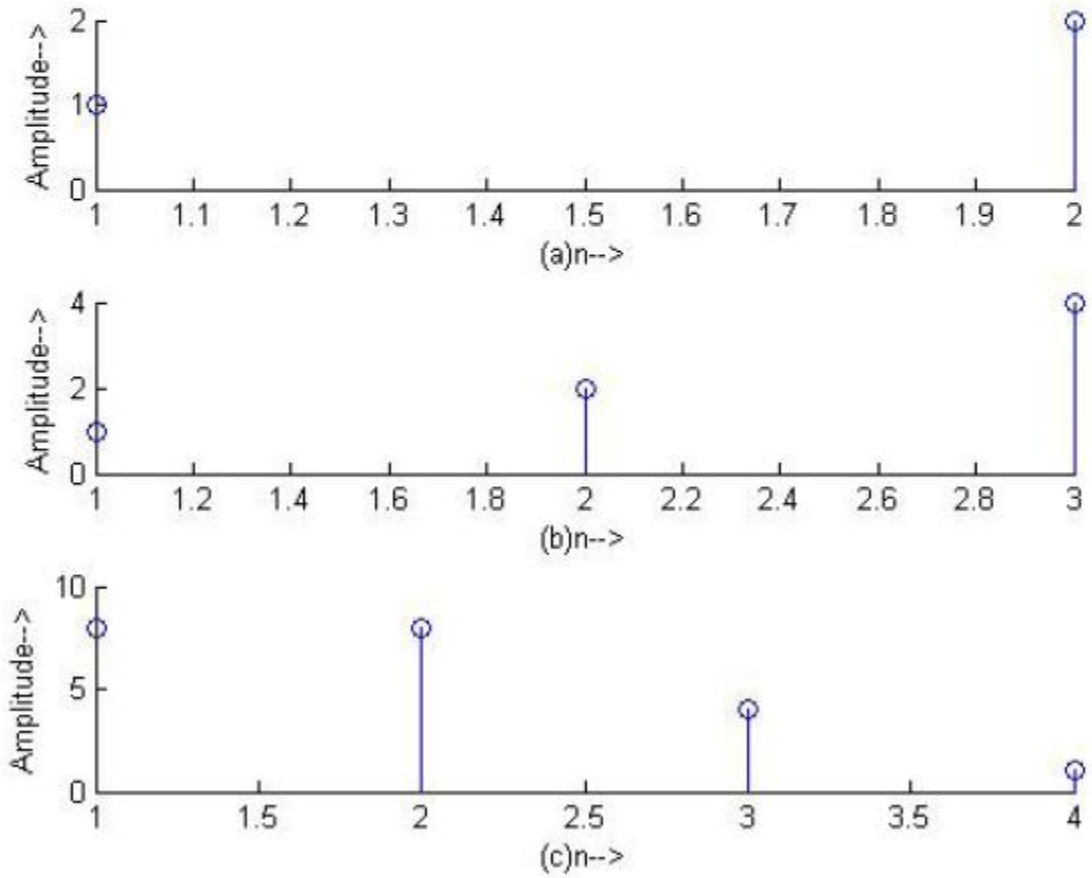
```
%Program for linear convolution of the sequence x= [1,2] and h= [1,2,4]
clc;
clear all;
close all;
x=input('enter the 1st sequence');
h=input('enter the 2nd sequence');
y=conv(x,h);
figure;
subplot(3,1,1);
stem(x);
ylabel('Amplitude-->');
xlabel('(a)n-->');
subplot(3,1,2)
stem(h);
ylabel('Amplitude-->');
xlabel('(b)n-->');
subplot(3,1,3)
stem(fliplr(y));
ylabel('Amplitude-->');
xlabel('(c)n-->');
disp('The resultant signal is');y
```

Output: -

```
enter the 1st sequence[1 2]
enter the 2nd sequence[1 2 4]
The resultant signal is

y =

    1    4    8    8
```



CONCLUSION:-

Discrete convolution of two sequences gives the product of two sequences. This property of signals is used in designing the DSP based filters.

Practical 3 : (2 Hours):

TO DEVELOP THE PROGRAM FOR DISCRETE CORRELATION

AIM: - To develop MATLAB program for evaluating discrete correlation.

APPARATUS:

1) MATLAB Software
2) Digital Signal Processing Toolbox

THEORY:-

1) What is discrete correlation?
2) What are the applications of Discrete Correlation?
3) How it can be implemented in the time domain analysis?

MATLAB SCRIPT:

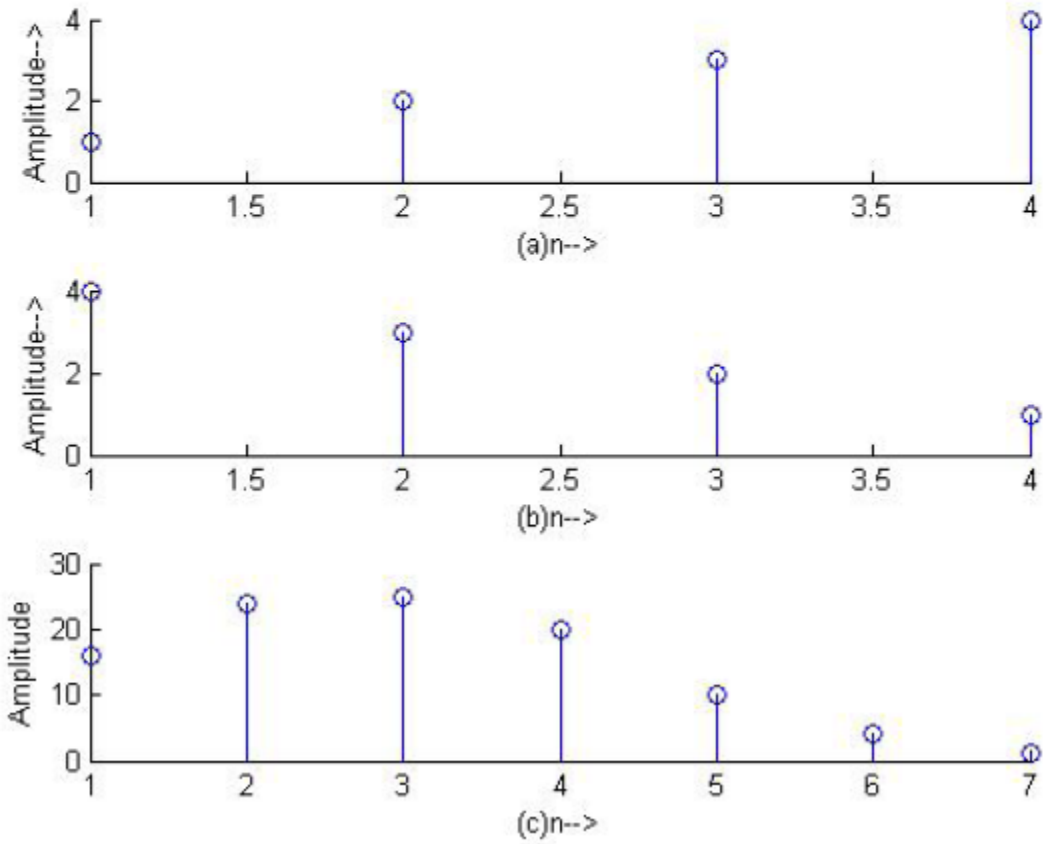
```
%Program for computing cross correlation of the sequences x=[1,2,3,4] and h=[4,3,2,1]
clc;
clear all;
close all;
x=input('enter the 1st sequence');
h=input('enter the 2nd sequence');
y=xcorr(x,h);
figure;
subplot(3,1,1);
stem(x);
ylabel('Amplitude-->');
xlabel('(a)n-->');
subplot(3,1,2)
stem(h);
ylabel('Amplitude-->');
xlabel('(b)n-->');
subplot(3,1,3)
stem(fliplr(y));
ylabel('Amplitude');
xlabel('(c)n-->');
disp('The resultant signal is');
fliplr(y)
```

OUTPUT:-

```
enter the 1st sequence[1 2 3 4]
enter the 2nd sequence[4 3 2 1]
The resultant signal is

ans =

   16.0000   24.0000   25.0000   20.0000   10.0000    4.0000    1.0000
```



CONCLUSION:

Cross correlation is the important property of DSP and it is a measure of similarity of two waveforms as a function of a time-lag applied to one of them.

Practical 4 : (2 Hours):

TO VERIFY STABILITY TEST

AIM: To verify Stability test in MATLAB.

APPARATUS:

1) MATLAB Software
2) Digital Signal Processing Toolbox

THEORY:

1. What is stability test?
2. What are the important constraints?
3. What are different conditions for system stability?

MATLAB SCRIPT:

```
%Program for stability test
clc;
clear all;
close all;
b=input('enter the denominator coefficients of the filter');
k=poly(b);
knew=fliplr(k);
s=all(abs(knew));
if(s==1)
disp('''stable system''');
else
disp('''Non-stable System''');
end
```

Output: -

```
enter the denominator coefficients of the filter[1 -1 .5]
'stable system'
```

CONCLUSION:-

Stability test is used to verify the stability of the system by putting the condition that a bounded input system remains stable and produces a bounded output (BIBO).

Practical 5 : (2 Hours):

TO VERIFY SAMPLING THEOREM

Aim:- To verify sampling theorem

APPARATUS:

1) MATLAB Software
2) Digital Signal Processing Toolbox

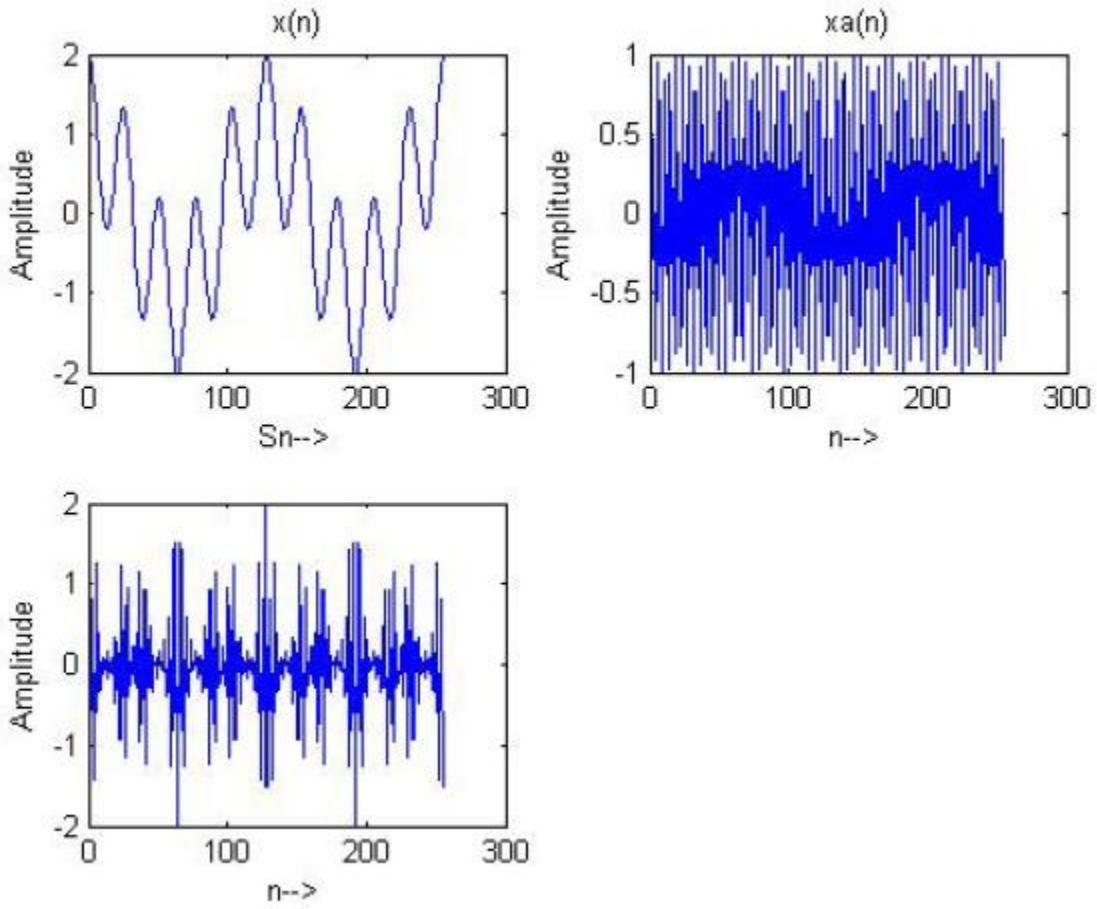
THEORY:-

1. What is sampling theorem?
2. Where it is applied?

MATLAB CODE:

```
%Program to understand sampling theorem
f1=1/128;
f2=5/128;
n=0:255;
fc=50/128;
x=cos(2*pi*f1*n)+cos(2*pi*f2*n);
xa=cos(2*pi*fc*n);
xamp=x.*xa;
subplot(2,2,1);
plot(n,x);
title('x(n)');
xlabel('Sn-->');
ylabel('Amplitude');
subplot(2,2,2);
plot(n,xa);
title('xa(n)');
xlabel('n-->');
ylabel('Amplitude');
subplot(2,2,3);
plot(n,xamp);
xlabel('n-->');
ylabel('Amplitude');
```

Output: -



CONCLUSION:-

We conclude that frequency of sampling must be greater than twice the highest frequency component of the signal otherwise the signal becomes unequally spaced.

Practical 6 : (2 Hours):

To design FIR Filter using window technique

Aim:- To design FIR Filter using window technique.

APPARATUS:

1) MATLAB Software
2) Digital Signal Processing Toolbox

Theory:-

1) What is FIR filter?
2) What is window technique?
3) How the filters can be designed by using window technique?

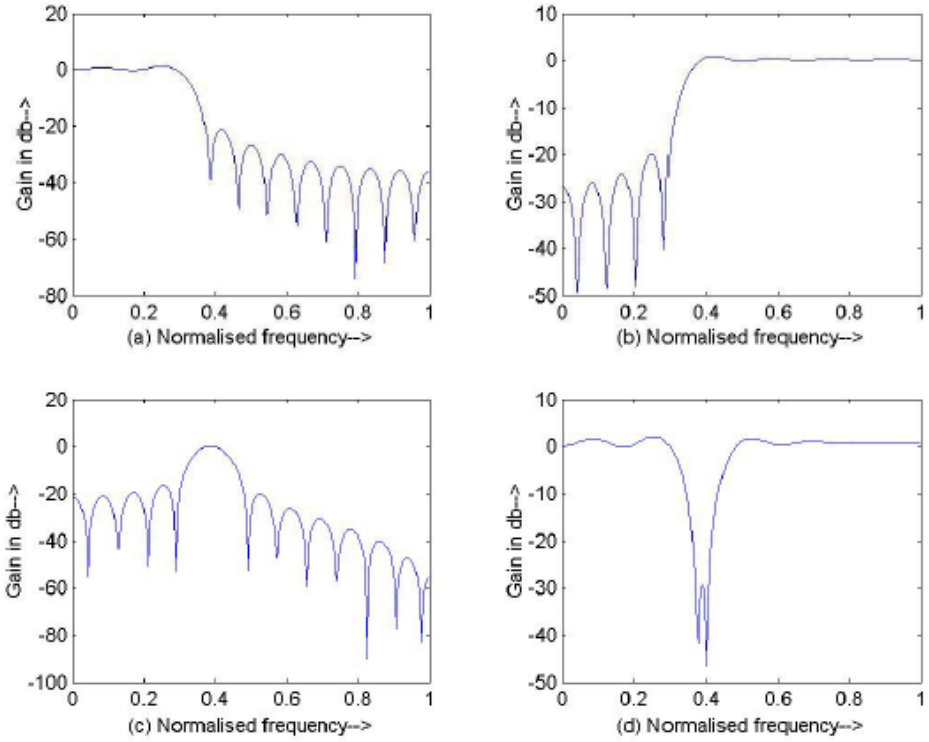
MATLAB SCRIPT:-

```
%Program for the design of FIR low pass, high pass, band pass, band stop using
%rectangular window
clc;
clear all;
close all;
rp=input('enter the passband ripple');
rs=input('enter the stopband ripple');
fp=input('enter the passband freq');
fs=input('enter the stopband freq');
f=input('enter the 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=boxcar(n1);
%low pass filter
b=fir1(n,wp,y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,1);
plot(o/pi,m);
```

```
ylabel('Gain in db-->');
xlabel('(a) Normalised frequency-->');
%high pass filter
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);
ylabel('Gain in db-->');
xlabel('(b) Normalised frequency-->');
%band pass filter
wn=[wp ws];
b=fir1(n,wn,y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,3);
plot(o/pi,m);
ylabel('Gain in db-->');
xlabel('(c) Normalised frequency-->');
%band stop filter
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);
ylabel('Gain in db-->');
xlabel('(d) Normalised frequency-->');
```

Output: -

enter the passband ripple0.04
enter the stopband ripple0.02
enter the passband freq1500
enter the stopband freq2000
enter the sampling freq9000



Conclusion:-

We conclude that, window technique helps in designing the filters.

Practical 7 : (2 Hours):

TO DESIGN DIGITAL IIR FILTER

AIM:- To design Digital IIR filter using MATLAB.

APPARATUS:

1) MATLAB Software
2) Digital Signal Processing Toolbox

THEORY:-

1) What is IIR filter?
2) How to design Digital IIR filter?

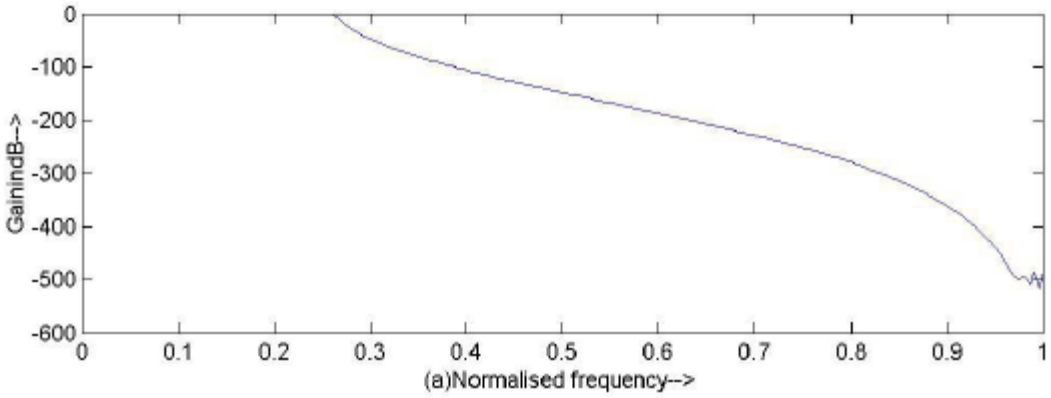
MATLAB SCRIPT:

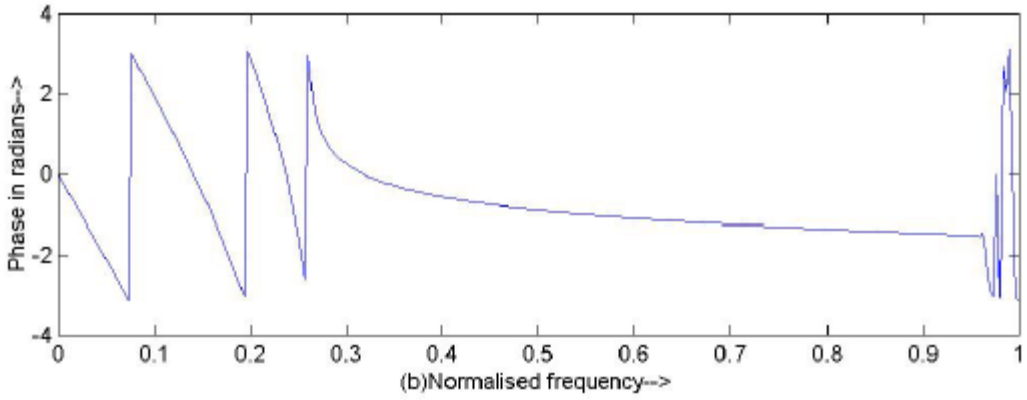
Low Pass: -

```
%program for the design of chebyshev lowpass filter
clc;
close all;
clear all;
format long
rs=input('enter the stopband ripple...');
rp=input('enter the passband ripple...');
ws=input('enter the stopband freq...');
wp=input('enter the passband freq...');
fs=input('enter the sampling freq...');
w1=2*wp/fs;
w2=2*ws/fs;
[n,wn]=cheb1ord(w1,w2,rp,rs);
[b,a]=cheby1(n,rp,wn);
w=0:0.01:pi;
[h,om]=freqz(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);
plot(om/pi,m);
ylabel('GainindB-->');
xlabel('(a)Normalised frequency-->');
subplot(2,1,2);
plot(om/pi,an)
xlabel('(b)Normalised frequency-->');
ylabel('Phase in radians-->');
```

Output :-

```
enter the stopband ripple...45
enter the passband ripple...0.2
enter the stopband freq...1500
enter the passband freq...1300
enter the sampling freq...10000
```



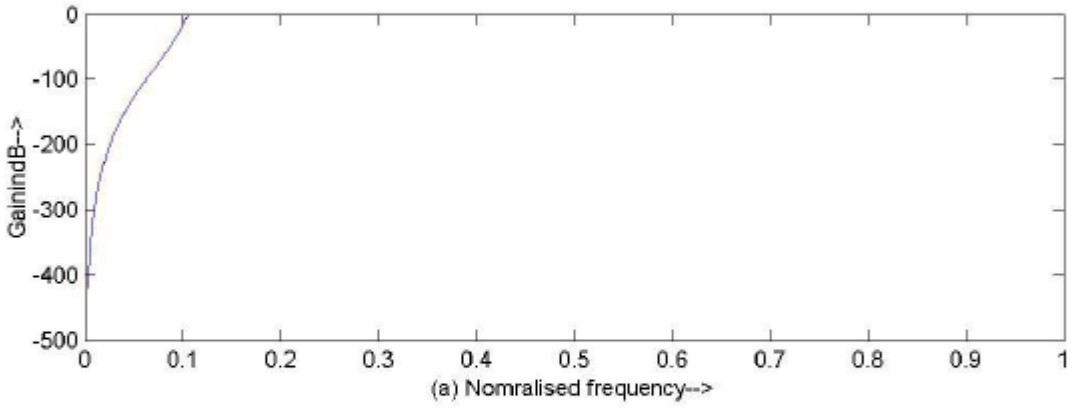


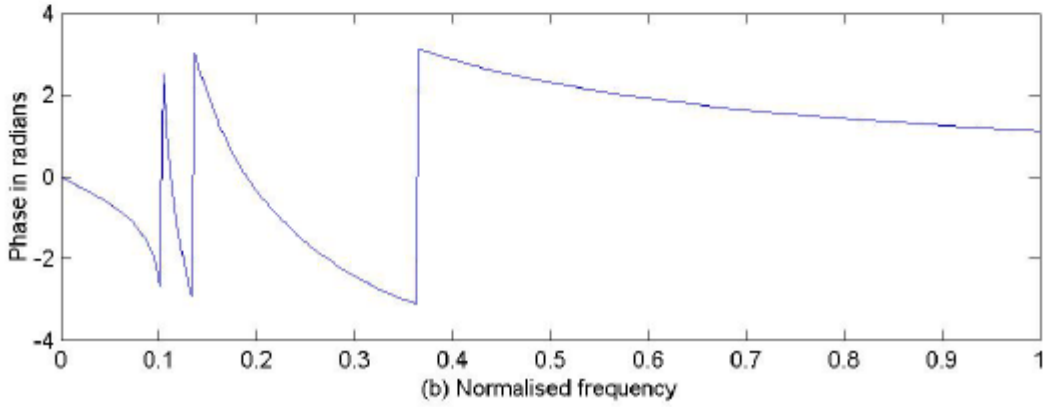
High Pass: -

```
%program for the design of chebyshev high pass filter
clc;
close all;
clear all;
format long
rp=input('enter the passband ripple...');
rs=input('enter the stopband ripple...');
wp=input('enter the passband freq.....');
ws=input('enter the stopband freq.....');
fs=input('ente the sampling freq.....');
w1=2*wp/fs;
w2=2*ws/fs;
[n,wn]=cheb1ord(w1,w2,rp,rs,'s');
[b,a]=cheby1(n,rp,wn,'high','s');
w=0:0.01:pi;
[h,om]=freqs(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);
plot(om/pi,m);
ylabel('GainindB-->');
xlabel('(a) Nomralised frequency-->');
subplot(2,1,2);
plot(om/pi,an);
xlabel('(b) Normalised frequency')
ylabel('Phase in radians')
```

Output: -

enter the passband ripple...0.3
enter the stopband ripple...60
enter the passband freq.....1500
enter the stopband freq.....2000
ente the sampling freq.....9000



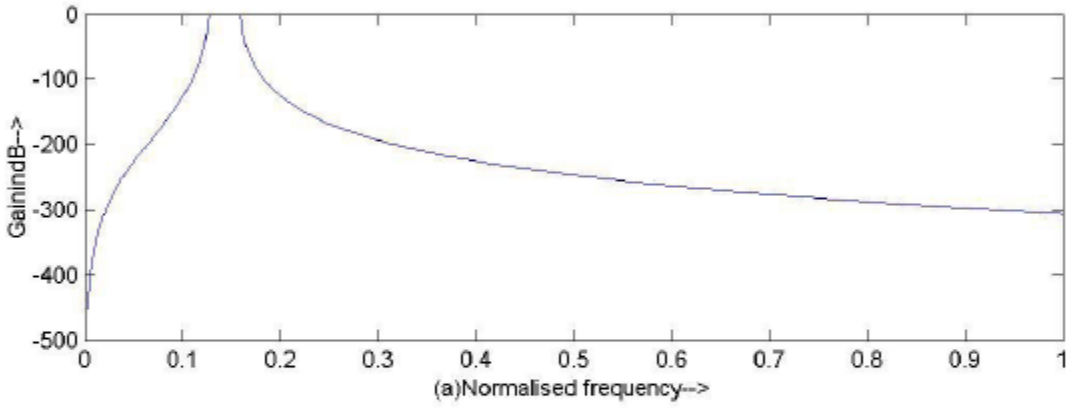


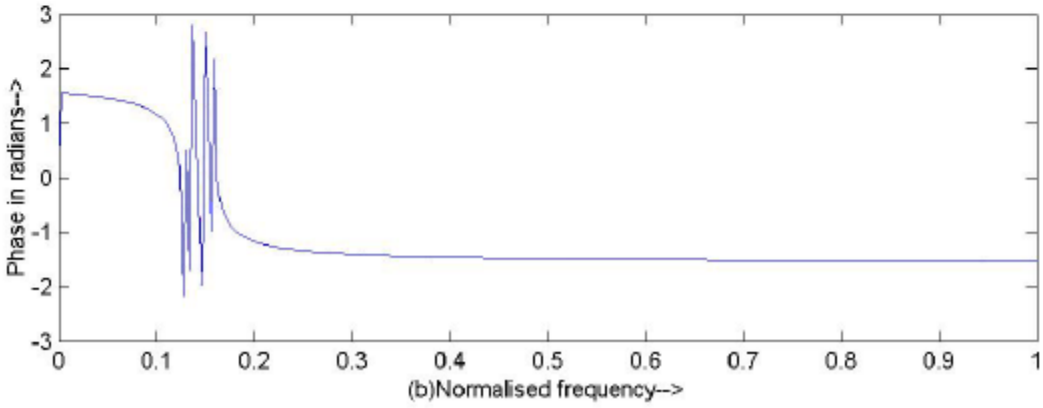
Band Pass: -

```
%program for the design of chebyshev bandpass filter
clc;
close all;
clear all;
format long
rp=input('enter the passband ripple...');
rs=input('enter the stopband ripple...');
wp=input('enter the passband freq...');
ws=input('enter the stopband freq...');
fs=input('enter the sampling freq...');
w1=2*wp/fs;
w2=2*ws/fs;
[n]=cheb1ord(w1,w2,rp,rs,'s');
wn=[w1 w2];
[b,a]=cheby1(n,rp,wn,'bandpass','s');
w=0:0.01:pi;
[h,om]=freqs(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);
plot(om/pi,m);
ylabel('GainindB-->');
xlabel('(a)Normalised frequency-->');
subplot(2,1,2);
plot(om/pi,an)
xlabel('(b)Normalised frequency-->');
ylabel('Phase in radians-->');
```

Output: -

enter the passband ripple...0.4
enter the stopband ripple...35
enter the passband freq...2000
enter the stopband freq...2500
enter the sampling freq...10000





Conclusion:-

We conclude that Digital IIR filters play important role in DSP technology.

Practical 8 : (2 Hours):

TO DESIGN A PROGRAM TO COMPARE DIRECT REALIZATION VALUES OF DIGITAL IIR FILTER

Aim:- To design a program to compare direct realization values of digital IIR filter

APPARATUS:

1) MATLAB Software
2) Digital Signal Processing Toolbox

Theory:-

1) What is Direct realization?
2) What is Digital IIR filter?
3) Mathematically how to compare direct realization values of digital IIR filter?

MATLAB SCRIPT:

```
%program for computing direct realisaion values of IIR digital filter
function y=direct(typ,b,a,x);
x=input('enter the input sequence=');
b=input('enter the numberator polynomials');
a=input('enter the denominator polynomials=');
typ=input('type of realisation=');
p=length(a)-1;
q=length(b)-1;
pq=max(p,q);
a=a(2:p+1);
u=zeros(1,pq);%u is the internal state;
if(typ==1)
for i=1:length(x),
unew=x(i)-sum(u(1:p).*a);
u=[unew,u];
y(i)=sum(u(1:q+1).*b);
u=u(1:pq)
end
elseif(typ==2)
   for i=1:length(x)
      y(i)=b(1)*x(i)+u(1);
      u=u((2:pq),0);
      u(1:q)=u(1:q)+b(2:q+1)*x(i);
      u(1:p)=u(1:p)-a*y(i);
   end
end
```

Conclusion:-

We conclude that realization of digital IIR filters can be effectively done by using MATLAB.

Practical 8 : (2 Hours):

TO DESIGN A PROGRAM FOR COMPUTING INVERSE Z TRANSFORMS

AIM:- To design a program for computing inverse z transforms

APPARATUS:

1) MATLAB Software
2) Digital Signal Processing Toolbox

THEORY:-

1) How inverse Z transform is computed?

MATLAB SCRIPT:

```
function x=invz(b,a,N);
b=input ('enter the numerator polynomials=');
a=input ('enter the denominator polynomials=');
N=input ('enter the number of points to be computed=');
[c,A,alpha]=tf2pf(b,a);
X=zeros (1,N);
x(1:length(c))=c;
for k=1:length(A),
x=x1A(k)*(alpha(k)).^(0:N-1);
end
x=real(x);
```

CONCLUSION:

We conclude that Inverse Z transform for a rational transfer function can be computed from the coefficients of polynomials of numerator and denominator of a transfer function.

3. Conduction of Viva-Voce Examinations:

Teacher should oral exams of the students with full preparation. Normally, the objective questions with guess are to be avoided. To make it meaningful, the questions should be such that depth of the students in the subject is tested Oral examinations are to be conducted in co-cordial environment amongst the teachers taking the examination. Teachers taking such examinations should not have ill thoughts about each other and courtesies should be offered to each other in case of difference of opinion, which should be critically suppressed in front of the students.

4. Evaluation and marking system:

Basic honesty in the evaluation and marking system is absolutely essential and in the process impartial nature of the evaluator is required in the examination system to become popular amongst the students. It is a primary responsibility of the teacher that right students who are really putting up lot of hard work with right kind of intelligence are correctly awarded.

The marking patterns should be justifiable to the students without any ambiguity and teacher should see that students are faced with unjust circumstances.