B.Sc VI semester(NEP)
Electronics
Signals &Systems Practicals
Course Code:036ELE012
Paper-I

Write and execute following program using MATLAB/OCTAVE/SCILAB, etc.

1. Generate and plot unit impluse, unit step, ramp, real sequences.
2. Generate and plot sinusoidal, cosinusoidal and periodic sequences.
3. Generate even & odd components of a sequence
4. Perform amplitude scaling, time scaling, folding and time-shifting operations on signals.
5. Perform Up sampling and down sampling operation on a given sequence.
6. Perform addition, subtraction and multiplication operation on signals.
7. Find the linear convolution of two finite duration sequences.
8. Find the cross-correlation of two finite duration sequences.
9. Evaluate & plot auto-correlation of a sequence
10. Verify the sampling theorem.

1. Generate and plot unit impulse, unit step, ramp, real sequences.

AIM: Generate various signals and sequences (Periodic and aperiodic), such as Unit Impulse, Unit Step, ramp, sinusoidal, cosinusoidal, square,and triangular signal.

Software Required: Matlab software

PROGRAM:

```
% Generation of signals and sequences
 clc;
clear all;
close all;
```

generation of unit impulse signal

```
 t1=-1:0.01:1
y1=(t1==0);
subplot(2,2,1);
 plot(t1,y1);
 xlabel('time');
 ylabel('amplitude');
title('unit impulse signal');
```

generation of impulse sequence

```
 subplot(2,2,2);
stem(t1,y1);
xlabel('n');
ylabel('amplitude');
title('unit impulse sequence');
```

generation of unit step signal

```
t2=-10:1:10;
y2=(t2>=0);
subplot(2,2,3);
plot(t2,y2);
xlabel('time');
ylabel('amplitude');
title('unit step signal');
```

generation of unit step sequence

```
subplot(2,2,4);
stem(t2,y2);
xlabel('n');
ylabel('amplitude');
title('unit step sequence');
```

generation of ramp signal

```
y7=t; figure;
subplot(2,2,1);
plot(t,y7);
xlabel('time');
ylabel('amplitude');
title('ramp signal');
```

generation of ramp sequence

```
subplot(2,2,2);
stem(t,y7);
xlabel('n');
ylabel('amplitude');
title('ramp sequence');
```

Result: Various signals & sequences generated using Matlab software.

2. Generate and plot sinusoidal, cosinusoidal and periodic sequences.

AIM: Generate various signals and sequences (Periodic and aperiodic), such as sinusoidal, cosinusoidal, square,and triangular signal.

Software Required: Matlab software

PROGRAM:

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

generation of square wave signal

```
t=0:0.002:0.1;
y3=square(2*pi*50*t);
figure;
subplot(2,2,1);
plot(t,y3);
axis([0 0.1 -2 2]);
xlabel('time');
ylabel('amplitude');
title('square wave signal');
```

generation of square wave sequence

```
subplot(2,2,2);
stem(t,y3);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
title('square wave sequence');
```

generation of sawtooth signal

```
y4=sawtooth(2*pi*50*t);
subplot(2,2,3);
plot(t,y4);
axis([0 0.1 -2 2]);
xlabel('time');
ylabel('amplitude');
title('sawtooth wave signal');
```

generation of sawtooth sequence

```
subplot(2,2,4);
stem(t,y4);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
title('sawtooth wave sequence');
```

generation of triangular wave signal

```
 y5=sawtooth(2*pi*50*t,.5);
 figure;
subplot(2,2,1);
plot(t,y5);
axis([0 0.1 -2 2]);
xlabel('time');
ylabel('amplitude');
title(' triangular wave signal');
```

generation of triangular wave sequence

```
subplot(2,2,2);
 stem(t,y5);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
 title('triangular wave sequence');
```

generation of sinsoidal wave signal

```
y6=sin(2*pi*40*t);
subplot(2,2,3);
plot(t,y6);
axis([0 0.1 -2 2]);
 xlabel('time');
ylabel('amplitude');
title(' sinsoidal wave signal');
```

generation of sin wave sequence

```
subplot(2,2,4);
stem(t,y6);
axis([0 0.1 -2 2]);
xlabel('n');
ylabel('amplitude');
title('sin wave sequence');
```

Result: Various signals & sequences generated using Matlab software.

3. Generate even & odd components of a sequence also find real and imaginary parts of signal.

AIM: Finding even and odd part of the signal and sequence and also find real and imaginary parts of signal.

Program:

```
clc
close all;
clear all;
%Even and odd parts of a signal
t=0:.001:4*pi;
x=sin(t)+cos(t);
subplot(2,2,1);
 plot(t,x)
xlabel('t');
ylabel('amplitude')
title('input signal')
y=sin(-t)+cos(-t);          % y(t)=x(-t)
subplot(2,2,2)
plot(t,y)
xlabel('t');
ylabel('amplitude')
title('input signal with t= -t')
even=(x+y)/2;
subplot(2,2,3)
plot(t,even)
xlabel('t');
ylabel('amplitude')
title('even part of the signal')
```

```
odd=(x-y)/2;
subplot(2,2,4)
 plot(t,odd)
xlabel('t');
ylabel('amplitude');
title('odd part of the signal');
% Even and odd parts of a sequence
 x1=[0,2,-3,5,-2,-1,6];
 n=-3:3;
y1= fliplr(x1);
%y1(n)=x1(-n)
 figure;
subplot(2,2,1);
stem(n,x1);
xlabel('n');
ylabel('amplitude');
 title('input sequence');
subplot(2,2,2);
stem(n,y1);
xlabel('n');
ylabel('amplitude');
title('input sequence with n= -n');
even1=.5*(x1+y1);
 odd1=.5*(x1-y1);
% plotting even and odd parts of the sequence
subplot(2,2,3);
stem(n,even1);
xlabel('n');
ylabel('amplitude');
```

```
title('even part of sequence');
subplot(2,2,4);
stem(n,odd1);
xlabel('n');
ylabel('amplitude');
title('odd part of sequence');
% plotting real and imginary parts of the signal
x2=sin(t)+j*cos(t);
figure;
subplot(3,1,1);
plot(t,x2);
xlabel('t');
 ylabel('amplitude');
title('input signal');
 subplot(3,1,2)
plot(t,real(x2));
xlabel('time');
ylabel('amplitude');
title('real part of signal');
subplot(3,1,3)
 plot(t,imag(x2));
xlabel('time');
ylabel('amplitude');
title('imaginary part of siganl')
```

Result: Various signals & sequences generated using Matlab software.

4. Perform amplitude scaling, time scaling, folding and time-shifting operations on signals.

AIM: Perform amplitude scaling, time scaling, folding and time-shifting operations on signals.

Program:

```
clc;
clear all;
 close all;
generating two input signals
 t=0:.01:1;
x1=sin(2*pi*4*t);
x2=sin(2*pi*8*t);
subplot(2,2,1);
plot(t,x1);
xlabel('time');
ylabel('amplitude');
 title('input signal 1');
 subplot(2,2,2);
plot(t,x2);
xlabel('time');
ylabel('amplitude');
title('input signal 2');
% scaling of a signal1
 A=2;
y3=A*x1;
figure;
subplot(2,2,1);
 plot(t,x1);
```

```
xlabel('time');
 ylabel('amplitude');
 title('input signal')
subplot(2,2,2);
 plot(t,y3);
xlabel('time');
ylabel('amplitude');
title('amplified input signal');
 % folding of a signal1
 h=length(x1);
 nx=0:h-1;
subplot(2,2,3);
plot(nx,x1);
xlabel('nx');
ylabel('amplitude');
title('input signal')
 y4=fliplr(x1);
nf=-fliplr(nx);
subplot(2,2,4);
plot(nf,y4);
xlabel('nf');
ylabel('amplitude');
title('folded signal');
%shifting of a signal 1 figure;
subplot(3,1,1);
 plot(t,x1);
xlabel('time t');
 ylabel('amplitude');
 title('input signal');
```

```
 subplot(3,1,2);
plot(t+2,x1);
xlabel('t+2');
ylabel('amplitude');
title('right shifted signal');
 subplot(3,1,3);
plot(t-2,x1);
xlabel('t-2');
ylabel('amplitude');
 title('left shifted signal');
```

Result: Various signals & sequences generated using Matlab software.

5. Perform addition, subtraction and multiplication operation on signals.

AIM: Perform addition, subtraction and multiplication operation on signals.

Program:

```
clc;
clear all;
 close all;
generating two input signals
 t=0:.01:1;
x1=sin(2*pi*4*t);
x2=sin(2*pi*8*t);
subplot(2,2,1);
plot(t,x1);
xlabel('time');
ylabel('amplitude');
 title('input signal 1');
 subplot(2,2,2);
plot(t,x2);
xlabel('time');
ylabel('amplitude');
title('input signal 2');
% addition of signals
y1=x1+x2;
subplot(2,2,3);
 plot(t,y1);
xlabel('time');
ylabel('amplitude');
 title('addition of two signals');
```

```
% multiplication of signals
y2=x1.*x2;
subplot(2,2,4);
plot(t,y2);
xlabel('time');
ylabel('amplitude');
title('multiplication of two signals');
%operations on sequences
n1=1:1:9;
s1=[1 2 3 0 5 8 0 2 4];
figure;
subplot(2,2,1);
stem(n1,s1);
xlabel('n1');
ylabel('amplitude');
title('input sequence1');
s2=[1 1 2 4 6 0 5 3 6];
subplot(2,2,2);
stem(n1,s2);
xlabel('n2');
ylabel('amplitude');
title('input sequence2');
% addition of sequences
s3=s1+s2;
subplot(2,2,3);
stem(n1,s3);
xlabel('n1');
ylabel('amplitude');
title('sum of two sequences');
```

```
% multiplication of sequences
s4=s1.*s2;
subplot(2,2,4);
 stem(n1,s4);
xlabel('n1');
ylabel('amplitude');
title('product of two sequences');
```

Result: Various signals & sequences generated using Matlab software.

6. Write the program for convolution between two signals and also between two sequences.

Aim: Write the program for convolution between two signals and also between two sequences**.**

Program:

```
clc;
close all;
clear all;
 %program for convolution of two sequences
x=input('enter input sequence: ');
 h=input('enter impulse response: ');
y=conv(x,h);
subplot(3,1,1);
stem(x);
xlabel('n');
 ylabel('x(n)');
 title('input sequence')
subplot(3,1,2);
stem(h);
xlabel('n');
ylabel('h(n)');
 title('impulse response sequence')
subplot(3,1,3);
stem(y);
xlabel('n');
 ylabel('y(n)');
 title('linear convolution')
 disp('linear convolution y=');
disp(y)
```

```
%program for convolution of two signals
t=0:0.1:10;
x1=sin(2*pi*t);
h1=cos(2*pi*t);
y1=conv(x1,h1);
figure;
subplot(3,1,1);
plot(x1);
xlabel('t');
ylabel('x(t)');
title('input signal')
subplot(3,1,2);
plot(h1);
xlabel('t');
ylabel('h(t)');
title('impulse response')
subplot(3,1,3);
plot(y1);
xlabel('n');
ylabel('y(n)');
title('linear convolution');
```

RESULT: convolution between signals and sequences is computed.

7. To compute Auto correlation and Cross correlation between signals and sequences.

Aim: To compute Auto correlation and Cross correlation between signals and sequences.

Program:

```
clc;
close all;
clear all;
two input sequences
x=input('enter input sequence');
h=input('enter the impulse suquence');
subplot(2,2,1);
 stem(x);
xlabel('n');
ylabel('x(n)');
 title('input sequence');
subplot(2,2,2);
stem(h);
 xlabel('n');
ylabel('h(n)');
title('impulse sequence');
cross correlation between two sequences
y=xcorr(x,h);
subplot(2,2,3);
 stem(y);
xlabel('n');
ylabel('y(n)');
```

```
title(' cross correlation between two sequences ');
```

auto correlation of input sequence

```
 z=xcorr(x,x);
subplot(2,2,4);
stem(z);
xlabel('n');
 ylabel('z(n)');
title('auto correlation of input sequence');
```

***cross correlation between two signals**

generating two input signals

```
 t=0:0.2:10;
x1=3*exp(-2*t);
 h1=exp(t);
figure;
subplot(2,2,1);
 plot(t,x1);
 xlabel('t');
ylabel('x1(t)');
title('input signal');
subplot(2,2,2);
 plot(t,h1);
xlabel('t');
ylabel('h1(t)');
title('impulse signal');
```

```
% cross correlation
subplot(2,2,3);
z1=xcorr(x1,h1);
plot(z1);
xlabel('t');
ylabel('z1(t)');
 title('cross correlation ');

% auto correlation
subplot(2,2,4);
z2=xcorr(x1,x1);
plot(z2);
xlabel('t');
ylabel('z2(t)');
 title('auto correlation ');
```

8. Verify the sampling theorem.

AIM: Verify the sampling theorem.

Program:

```
clc;
clear all;
 close all;
t=-10:.01:10;
T=4;
fm=1/T;
 x=cos(2*pi*fm*t);
subplot(2,2,1);
 plot(t,x);
xlabel('time');
 ylabel('x(t)');
title('continous time signal');
 grid;
n1=-4:1:4;
fs1=1.6*fm;
fs2=2*fm;
fs3=8*fm;
 x1=cos(2*pi*fm/fs1*n1);
subplot(2,2,2);
stem(n1,x1);
 xlabel('time');
ylabel('x(n)');
title('discrete time signal with fs<2fm')
hold on;
subplot(2,2,2);
 plot(n1,x1);
```

```
grid;
n2=-5:1:5;
 x2=cos(2*pi*fm/fs2*n2);
subplot(2,2,3);
stem(n2,x2);
xlabel('time');
ylabel('x(n)');
 title('discrete time signal with fs=2fm');
hold on;
subplot(2,2,3);
plot(n2,x2)
grid;
 n3=-20:1:20;
x3=cos(2*pi*fm/fs3*n3);
subplot(2,2,4);
stem(n3,x3);
 xlabel('time');
ylabel('x(n)');
title('discrete time signal with fs>2fm')
hold on;
 subplot(2,2,4);
 plot(n3,x3)
grid;
```