Experiment No. - 1

GENERATION OF BASIC SIGNALS USING MATLAB

(a). Program for the generation of unit step function and delayed step function with a delay of 5.

```
% unit step function and its delay
%version.
clear all;
n= -10:20;
u=[zeros(1,10),ones(1,21)];
stem(n,u);
axis([-10 20 0 1.2])
title('Unit step sequence');
xlabel('time index n ---->');
ylabel('Amplitude ----->');
%unit delayed step function
n= -10:20;
m=input('\n Enter the delay=');
ud=[zeros(1,10+m),ones(1,21-m)];
stem(n,ud)
axis([-10 20 0 1.2])
title('Unit delayed step sequence');
xlabel('time index n------>');
ylabel('Amplitude------->');
```

Input:

```
Enter the delay=5
```

(b). Program for the generation of unit sample sequence and delayed unit sample sequence with a delay of 5.

```
% unit sample sequence
clc;
close all;
clc
clear all;
n=-10:20;
u=[zeros(1,10),1,zeros(1,20)];
stem(n,u);
title('Unit sample sequence');
xlabel('time index n');
ylabel('Amplitude');
axis([-10 20 0 1.2])
%unit delayed function
m=input('\n Enter the delay=');
```

```
ud=[zeros(1,10+m),1,zeros(1,20-m)];
stem(n,ud);
axis([-20 20 0 2])
title('Unit delayed sample by m');
xlabel('time index n------>');
ylabel('Amplitude------>');
```

Input:

```
Enter the delay=5
```

(c). Program for the generation of unit ramp signal.

```
% Program for unit ramp signal.
clc;
close all;
clear all;
n= 0:10;
u= [(n)<=10];
x=n.*u;
stem(n,x);
title('Plot of x(n)=n*u(n)');
axis([0 10.5 0 10.5])
xlabel('time----->');
ylabel('Amplitude---->');
```

(d). Program for the generation of exponential signal.

```
% Program for the generation of exponential signal.
clc;
close all;
clear all;
n=100;
h=1/n;
t=0:h:1;
a=exp(-2*t);
plot(t,a);
title('Graph of y=exp(-2*t)');
%raising exp(2*t)
xlabel('x-axis');
ylabel('y-axis');
```

e). Plot a graph of y=sin3πx for 0≤ x ≤ 1. Also give the title and label the x & y axes of the graph.

```
%Program to generate sine curve
clc;
close all;
clear all;
n=100;
h=1/n;
x=0:h:1;
y=sin(3*pi*x);
plot(x,y);
title('Graph of y= sin(3pix)');
xlabel('x-axis');
ylabel('y-axis');
```

(f). Plot a graph of y1=cos3πx. Also give the title and label the x & y axes of the graph.

```
% Program to generate cosine curve
clc;
close all;
clear all;
n=100;
h=1/n;
x=0:h:1;
y1=cos(3*pi*x);
plot(x,y1);
title('Graph of y= cos(3pix)');
xlabel('x-axis');
ylabel('y-axis');
```

(g). Plot a graph of sin(nπx) on the interval -1≤x ≤1 for n=1,2,……8 using for loop.[Note use subplot command to plot the 8 graphs in one window].

```
% Program to generate sin(nπx) curve
clc;
close all;
clear all;
m=100;
h=1/m;
x=-1:h:1;
for n=1:8
subplot(4,2,n)
   y=sin(n*pi*x);
plot(x,y);
title('Graph of sin(nπx)');
```

```
xlabel('x-axis');
ylabel('y-axis');
end
```

h). Plot an exponentially decaying sine wave A= $e^{-2t}\sin(2\pi 5t)$, -1≤t ≤1. Also demonstrate the use of 'subplot' and 'figure' commands.

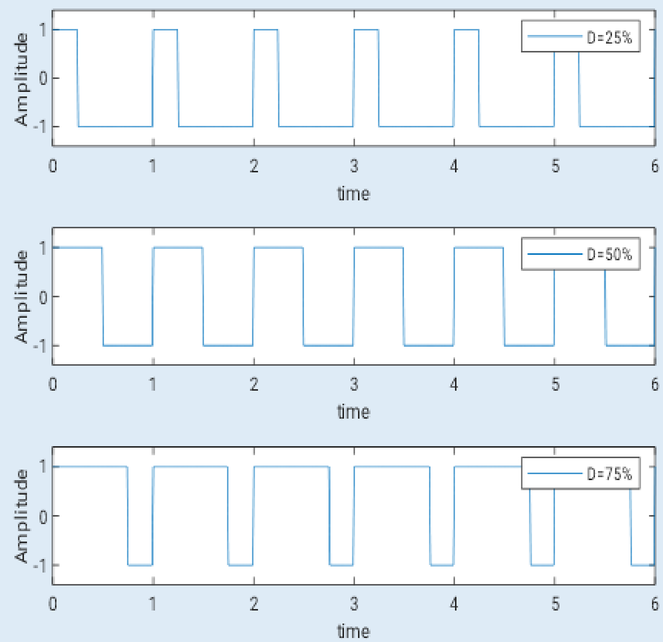
```
% Program to generate exponentially
%decaying sine wave
clc;
clear all;
close all;
n= 100;
h= 1/n;
t=0:h:1;
subplot(3,1,1)
a=exp(-2*t);
plot(t,a);
title('Graph of y= exp(-2*t)');
% raising exp(2*t)
xlabel('x-axis');
ylabel('y-axis');
subplot(3,1,2);
b=(sin(2*pi*5*t));
plot(t,b);
title('Graph of y= sin(2*pi*5*t)');
xlabel('x-axis');
ylabel('y-axis');
subplot(3,1,3);
A= a.*b;
plot(t,A);
title('Graph of y= exp(-2*t)sin(2*pi*5*t)');
xlabel('x-axis');
ylabel('y-axis');
```

i). Generate and plot a triangular wave.

```
% Generation of triangular wave
clc;
close all;
clear all;
t= 0:0.01:6;
x1=sawtooth(2*pi*t,0.5);
plot(t,x1);
title('triangular wave');
xlabel('Time');
ylabel('Amplitude');
```

j). Generation and plot square wave with duty cycle 25,50 and 75%.

```
%Program to generate square with various duty cycle
clc;
close all;
clear all;
t=0:0.01:6;
x1= square(2*pi*t,25);
subplot(3,1,1)
plot(t,x1);
axis([0 6 -1.4 1.4]);
legend('D=25%');
xlabel('time');
ylabel('Amplitude');
x2=square(2*pi*t,50);
subplot(3,1,2);
plot(t,x2);
axis([0 6 -1.4 1.4]);
legend('D=50%');
xlabel('time');
ylabel('Amplitude');
x3=square(2*pi*t,75);
subplot(3,1,3);
plot(t,x3);
axis([0 6 -1.4 1.4]);
legend('D=75%');
xlabel('time');
ylabel('Amplitude');
```



Conclusion:

Viva Questions:

1. Define sinusoidal signal
2. Define C.T.S
3. Define D.T.S.
4. Compare C.T.S & D.T.S
5. Define Stem, Plot, Plot3,fplot, ezplot, linspace, flyplr, grid,mesh and legend
6. Draw the C.T.S & D.T.S diagrams

Experiment No. 2

EVALUATION OF IMPULSE RESPONSE OF A SYSTEM

PROGRAM: IMPLULSE RESPONSE USING DECONV FUNCTION

```
clc;                                                  % clear screen
close all                                             % close all figure windows
clear all;                                            % clear work space
y = input('Output sequence y(n) of the system = ');   % enter the output sequence
x = input('Input sequence x(n) of the system = ');    % enter the input sequence
[h,r] = deconv(y,x);          % deconvolute output and input to get the impulse response
disp('Impulse response of the system is = ');
disp(h);                                              % display result
N= length(h);                                         % find the length of h
n = 0:1:N-1;                                          % define time axis
stem(n,h);                                            % plot the impulse response
xlabel('n');                                          % label x axis
ylabel('h(n)');                                       % label y axis
title('Impulse Response of the system');              % graph title
```

OUTPUT:

Output sequence y(n) of the system = [1 -3/4]

Input sequence x (n) of the system = [1 -3/4 1/8]

Impulse response of the system = [1 0 –0.125 -0.0937 -0.0546]

PROGRAM: IMPLULSE RESPONSE USING IMPZ FUNCTION

```
%Program for impulse response using IMPZ function
clc;                                                % clear screen
close all;                                          % close all figure windows
clear all;                                          % clear workspace
y=input('Output sequence y(n) of the system = ');% enter the coefficients of y terms
x=input('Input sequence x(n) of the system = '); % enter the coefficient of x terms
N=input('Enter the length of impulse response =');% define the length of the response
h=impz(y,x,N);                                       % calculate the impulse response
disp('Impulse Response of system h(n)');     % display the values of impulse response
```

```
disp(h);                                          % graphical plot
n=0:1:N-1;                                        % define x axis
stem(n,h);                                        % plot the impulse response
xlabel('n');                                      % label x axis
ylabel('h(n)');                                   % label y axis
title('Impulse Response');                        % graph title
```

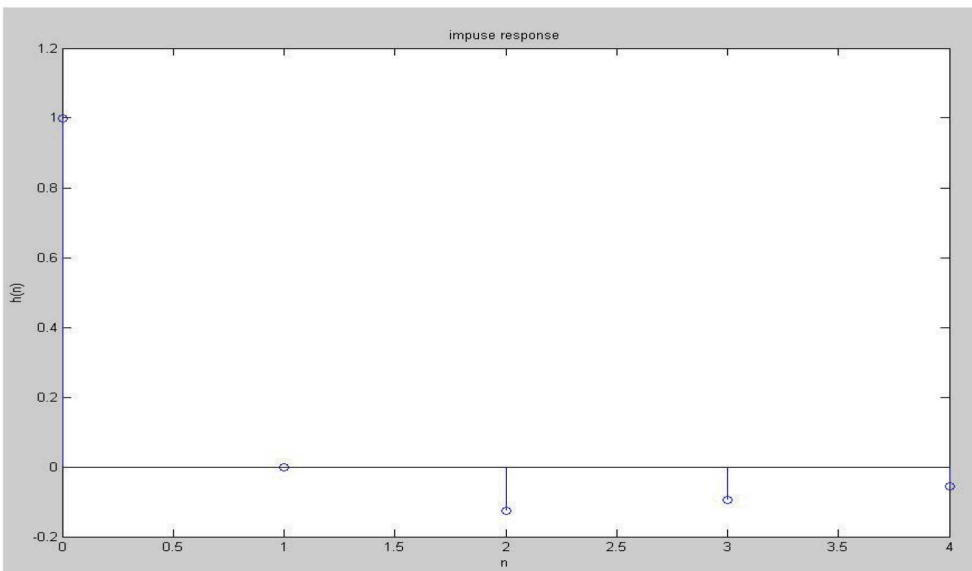
OUTPUT:

Output sequence y(n) of the system = [1 -3/4]

Input sequence x(n) of the system = [1 -3/4 1/8]

Enter the length of impulse response =5

Impulse Response of system h(n)= 1.0000   0   -0.1250   -0.0938   -0.0547



PROGRAM: IMPLULSE RESPONSE USING FILTER FUNCTION

```
%Program for impulse response using filter function

clc;                                         % clear screen
close all;                                   % close all the figure windows
clear all;                                   % clear the work space
y=input('Enter the co-efficient of y = ');   % define the coefficient of y terms
x=input('Enter the co-efficient of x = ');   % define the coefficient of x terms
N=input('Enter the length of impulse sequence =');% define the length of the response
xi=[1,zeros(1,N-1)];                          % define the input signal
h=filter(x,y,xi);                             % calculate the impulse response
disp( 'Impulse Response of the system is = ');% display the response
```

```
disp(h);                                       % graphical display
n=0:1:N-1;                                     % define the time axis
stem(n,h);                                     % plot the impulse response
xlabel('n');                                   % label x axis
ylabel('h(n)');                                % label y axis
title('Impulse Response of the system');      % graph title
```

OUTPUT:

```
Enter the co-efficient of y = [1 -3/4 1/8]
Enter the co-efficient of x = [1 -3/4]
Enter the length of impulse sequence = 5
Impulse Response of the system is =
    1.0000         0   -0.1250   -0.0938   -0.0547
```

VIVA QUESTIONS WITH ANSWERS

1. What are the various methods available to determine the response of LTI systems?
2. What is impulse response and what is its significance?
3. Define the transfer function of an LTI system
4. What is BIBO stability? What is the condition to be satisfied foe stability?
5. What do you mean by real time signal? Give example.

Experiment No. 3

COMPUTATION OF N – POINT DFT AND TO PLOT THE MAGNITUDE AND PHASE SPECTRUM

Program:

```
%without fft for dft %
clc;
clear all;
close all;
x=input('enter the 1st value');
N=length(x);
for k=0:N-1
y(k+1)=0;
for n=0:N-1
y(k+1)=y(k+1)+x(n+1)*exp(-i*2*pi*n*k/N);
end
```

```
end
y
mag=abs(y)
phase=angle(y)
subplot(2,2,1)
n=0:1:length(x)-1;
stem(n,x)
title('the values of x');
subplot(2,2,2)
n1=0:1:length(y)-1;
stem(n1,y)
title('output of y');
subplot(2,2,3)
n2=0:1:length(y)-1;
stem(n2,mag)
title('magnitude plot');
subplot(2,2,4)
n3=0:1:length(y)-1
stem(n3,phase)
title('phase plot');
```

Output:

```
enter the 1st value[1 2 3 4]
N = 4
y = 10.0000     -2.0000 + 2.0000i    -2.0000 - 0.0000i        -2.0000 - 2.0000i
mag = 10.0000    2.8284    2.0000    2.8284
phase = 0    2.3562   -3.1416   -2.3562
n3 = 0     1     2     3
```

Experiment No. 4

TO PERFORM LINEAR CONVOLUTION OF GIVEN SEQUENCES

Program:

```
%linear convolution %
clc;
clear all;
close all;
x=input('enter the 1st value');
h=input('enter the 2nd value');
y=conv(x,h);
```

```
disp('the o/p of linear convolution is');
disp(y)
subplot(2,2,1);
n=0:1:length(x)-1;
stem(n,x)
title('the 1st o/p value');
xlabel('time index');
ylabel('ampitude');
subplot(2,2,2);
n2=0:1:length(h)-1;
stem(n2,h)
title('the 2nd o/p value');
xlabel('time index');
ylabel('ampitude');
subplot(2,2,[3 4]);
n3=0:1:length(y)-1;
stem(n3,y)
title('the linear convolution o/p values ');
xlabel('time index');
ylabel('ampitude');
```

Output:

```
enter the 1st value[1 2 3]
enter the 2nd value[1 2 3]
the o/p of linear convolution is
    1     4    10    12     9
```

Experiment No. 5

LINEAR AND CIRCULAR CONVOLUTION BY DFT AND IDFT METHOD.

Program:

(i) Linear convolution using DFT and IDFT:

```
clc;
clear all;
close all;
x=input('enter the 1st value');
h=input('enter the 2nd value');
N=length(x)+length(h)-1
```

```
Xk=fft(x,N)
Hk=fft(h,N)
Yk=Xk.*Hk
yn=ifft(Yk,N)
yr=real(yn);
subplot(2,2,1)
n=0:1:length(x)-1;
stem(n,x)
title('the values of x');
subplot(2,2,2)
n1=0:1:length(h)-1;
stem(n1,h)
title(' values of h');
subplot(2,2,[3 4])
n2=0:1:N-1;
stem(n2,yr)
title('real of ifft');
```

Output:

```
enter the 1st value[1 2 3]
enter the 2nd value[1 1]

N =4
Xk =6.0000          -2.0000 - 2.0000i   2.0000          -2.0000 + 2.0000i
Hk =2.0000           1.0000 - 1.0000i        0           1.0000 + 1.0000i
Yk =12    -4     0    -4
yn = 1     3     5     3
```

(ii)Circular convolution using DFT and IDFT:

```
clc;
clear all;
close all;
x=input('enter the 1st value');
h=input('enter the 2nd value');
N1=length(x);
N2=length(h);
N=max(N1,N2);
Xk=fft(x,N)
Hk=fft(h,N)
Yk=Xk.*Hk
```

```
yn=ifft(Yk,N)
yr=real(yn)
subplot(2,2,1)
n=0:1:length(x)-1;
stem(n,x)
title('the values of x');
subplot(2,2,2)
stem(n,h)
title(' values of h');
subplot(2,2,[3 4])
stem(n,yr)
title('real of ifft');
disp(yr);
```

Output:

```
enter the 1st value[1 2 3 4]
enter the 2nd value[1 1 1 1]

Xk =10.0000          -2.0000 + 2.0000i         -2.0000           -2.0000 - 2.0000i
Hk =4     0     0     0
Yk =40     0     0     0
yn =10    10    10    10
yr =10    10    10    10
```

Experiment No. 5

TO PERFORM CIRCULAR CONVOLUTION OF GIVEN SEQUENCES USING (A) THE CONVOLUTION SUMMATION FORMULA (B) THE MATRIX METHOD AND (C) LINEAR CONVOLUTION FROM CIRCULAR CONVOLUTION WITH ZERO PADDING

a) Circular convolution using convolution summation formula:

Program:

```
%circular convolution%
clc;
close all
clear all
x=input('enter the 1st value');
```

```
h=input('enter the 2nd value');
N1=length(x);
N2=length(h);
N=max(N1,N2);
for n=0:1:N-1
y(n+1)=0;
for M=0:1:N-1
i=n-M;
if(i<0)
i=i+N;
end
y(n+1)=y(n+1)+x(M+1)*h(i+1);
end
end
disp(y)
n=0:1:N-1;
stem(n,y)
title('circular convolution plot');
xlabel('time');
ylabel('amplitude');
```

Output:

```
enter the 1st value[1 2 2 1]
enter the 2nd value[4 3 2 1]
   13    15    17    15
```

b) Circular convolution using matrix method:

1st Method

```
%matrix convolution method%
clc;
clear all;
close all;
x=input('enter the 1st sequence');
h=input('enter the 2nd sequence');
y=h*x;
disp('the circular conv o/p is');
disp(y);
subplot(3,1,1)
n=0:1:length(x)-1;
stem(n,x);
title('input of x');
xlabel('time');
```

```
ylabel('amplitude');
subplot(3,1,2)
n1=0:1:length(h)-1;
stem(n1,h);
title('input of h');
xlabel('time');
ylabel('amplitude');
subplot(3,1,3)
n2=0:1:length(y)-1;
stem(n2,y);
title('output of y');
xlabel('time');
ylabel('amplitude');
```

Output:

```
enter the 1st sequence[1;2;3]
enter the 2nd sequence[1 3 2;2 1 3;3 2 1]

the circular conv o/p is
    13
    13
    10
```

2nd Method

```
%circular matrix mutlipication%
clc;
clear all;
close all;
x=input('enter the 1st sequence');
h=input('enter the 2nd sequence');
a=h'
A=circshift(a,1);
B=circshift(a,2);
C=[a A B]
y=C*x';
disp('the circular conv o/p is');
disp(y);
subplot(3,1,1)
n=0:1:length(x)-1;
stem(n,x);
title('input of x');
xlabel('time');
```

```
ylabel('amplitude');
subplot(3,1,2)
n1=0:1:length(h)-1;
stem(n1,h);
title('input of h');
xlabel('time');
ylabel('amplitude');
subplot(3,1,3)
n2=0:1:length(y)-1;
stem(n2,y);
title('input of y');
xlabel('time');
ylabel('amplitude');
```

<u>Output:</u>

<u>Plot:</u>

```
enter the 1st sequence[1 2 3]
enter the 2nd sequence[1 2 3]

a =

     1
     2
     3

C =

     1     3     2
     2     1     3
     3     2     1

the circular conv o/p is
    13
    13
    10
```

c) Linear convolution from circular convolution with zero padding

<u>Program:</u>

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

```
x=input('enter the 1st value');
h=input('enter the 2nd value');
N=length(x)+length(h)-1;
X=[x,zeros(1,N-length(x))];
H=[h,zeros(1,N-length(h))];
for n=0:N-1
    Y(n+1)=0;
    for M=0:N-1
        i=n-M;
        if(i<0)
            i=i+N;
        end
        Y(n+1)=Y(n+1)+X(M+1)*H(i+1);
    end
end
disp(Y)
stem(Y)
title('the output values');
xlabel('time');
ylabel('amplitude');
```

Output:

```
enter the 1st value[1 2 3 1]
enter the 2nd value[1 1 1]
     1     3     6     6     4     1
```

Experiment No. 7

VERIFICATION OF SAMPLING THEOREM BOTH IN TIME AND FREQUENCY DOMAINS

Program:

```
clc;                 % clears the command window
clear all;           % clears the variables declared
t=0:0.001:0.1;
f1=input ('Enter the input frequency1 = ');
f2=input ('Enter the input frequency2 = ');
y=cos(2*pi*f1*t)+cos(2*pi*f2*t);
f3=max(f1,f2);
 % under sampling
fs=f3;               %fs = sampling frequency
ts=1/fs;
tx=0:ts:0.1;
```