Sri Venkateswara College of Engineering and Technology (Autonomous)

Chittoor

Department of Electronics and Communication Engineering

SIGNALS AND SYSTEMS LAB MANUAL

(14AEC09)

(II B.Tech -I Semester ECE)

ROLL NO: ______________________________

NAME: ________________________________

CLASS: _______________________________

List of The Experiments

S.No	Date	Name of The Experiment	Remarks/ Signature
1		BASIC OPERATIONS ON MATRICES.	
2.a		GENERATION OF CONTINUOUS TIME SIGNALS.	
2.b		GENERATION OF DISCRETE TIME SIGNALS.	
2.c		GENERATION OF PERIODIC CONTINUOUS AND DISCRETE TIME SIGNALS.	
3		OPERATIONS ON SIGNALS AND SEQUENCES.	
4.a		FINDING EVEN AND ODD PARTS OF A SIGNAL.	
4.b		FINDING EVEN AND ODD PARTS OF A DISCRETE TIME SIGNAL(SEQUENCE).	
5		CONVOLUTION OF TWO SEQUENCES.	
6		AUTO-CORRELATION & CROSS-CORRELATION BETWEEN SIGNALS.	
7.a		VERYFICATION OF LINEARITY PROPERTY OF A GIVEN SYSTEM.	
7.b		VERYFICATION OF TIME-INVARIANCE PROPERTY OF A GIVEN SYSTEM.	
8		COMPUTATION OF UNIT SAMPLE,UNIT STEP AND SINUSOIDAL RESPONSES OF THE GIVEN LTI SYSTEM AND VERIFYING ITS PHYSICAL REALIZABILITY AND STABILITY PROPERTIES.	
9		GIBBS PHENOMENON.	
10.a		FINDING FOURIER TRANSFORMS AND INVERSE FOURIER TRANSFORM.	
10.b		FINDING MAGNITUDE AND PHASE SPECTRUM OF FOURIER TRANSFORMS.	
11		WAVE SYNTHESIS USING LAPLACE TRANSFORM.	
12.a		FINDING AND LOCATING ZEROS AND POLES IN S- PLANE.	
12.b		FINDING AND LOCATING ZEROS AND POLES IN Z- PLANE.	
13		GENERATION OF GAUSSIAN NOISE COMPUTATION OF ITS MEAN,M.S VALUE,PSD.	
14		SAMPLING THEOREM VERIFICATION .	
15		REMOVAL OF NOISE BY AUTO CORRELATION/CROSS CORRELATIONIN A GIVEN SIGNAL CORRUPTED BY NOISE.	

Experiment No-1

BASIC OPERATIONS ON MATRICES

AIM : To write a MATLAB program to perform some basic operation on matrices such as addition, subtraction, multiplication ,right division ,left division ,inverse etc.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
close all;
clear all;
    A=[1 1 -2;2 -1 1;3 1 -1]
    B=[1 1 1;2 5 7;2 1 -1]
    MATRIXADDITION=A+B
    MATRIXSUBTRACTION=A-B
    MATRIXMULTIPLICATION=A*B
    ELEMENTWISEMULTIPLICATION=A.*B
    RIGHTDIVISION=A/B
    LEFTDIVISION=A\B
    ELEMENTWISERIGHTDIVISION=A./B
    ELEMENTWISELEFTDIVISION=A.\B
    INVERSE=inv(A)
    EXPONENTOFMATRIX=A^2
    ELEMENTWISEEXPONENTOFMATRIX=A.^B
    TRANSPOSE=A'
    ARRAYTRANSPOSE=A.'
```

RESULT : Thus,the MATLAB program of performing addition, subtraction, multiplication,right division ,left division ,inverse etc . was successfully completed using MATLAB software.

Experiment No-2a

GENERATION OF CONTINUOUS TIME SIGNALS

AIM : To write a MATLAB Program to generate continuous time signals like unit impulse, unit step, unit ramp, exponential signal and sinusoidal signals.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
clear all;
close all;
t=-10:0.01:10;
L=length(t);
for i=1:L
   % to generate a continuous time impulse function
   if t(i)==0
     x1(i)=1;
   else
      x1(i)=0;
   end;
   % to generate a continuous time unit step signal
   if t(i)>=0
     x2(i)=1;
      % to generate a continuous time ramp signal
     x3(i)=t(i);
   else
      x2(i)=0;
      x3(i)=0;
   end;
end;
% to generate a continuous time exponential signal
```

```
a=0.85;
x4=a.^t;
figure;
subplot(3,2,1);
plot(t,x1);
grid on;
xlabel('continuous time t ---->');
ylabel('amplitude---->');
title('Continuous time unit impulse signal');
subplot(3,2,2);
plot(t,x2);
grid on;
xlabel('continuous time t ---->');
ylabel('amplitude---->');
title('Unit step signal')
subplot(3,2,3);
plot(t,x3);
grid on;
xlabel('continuous time t ---->');
ylabel('amplitude---->');
title(' Unit ramp signal');
subplot(3,2,4);
plot(t,x4);xlabel('continuous time t ---->');
grid on;
ylabel('amplitude---->');
title('continuous time exponential signal');
% to generate  a continuous time signum function
 a=sign(t);
 subplot(3,2,[5,6]);
 plot(t,a);grid on;
 xlabel('continuous time t ------>');
 ylabel('amplitude ---->');
 title('continuous time signum function');
 figure;
  % to generate a continuous time sinc function
 t=-10:.1:10;
 Wt=sinc(t);
 plot(t,Wt);
 grid on;
 xlabel('continuous time t ------->');
 ylabel('amplitude ---->');
 title('continuous time sinc function');
```

<u>RESULT</u> : Thus, the MATLAB Program of the generation of continuous time signals like unit impulse, unit step,unit ramp, exponential signal and sinusoidal signals was successfully executed using MATLAB software.

Experiment No-2b

GENERATION OF DISCRETE TIME SIGNALS

AIM : To write a MATLAB Program to generate discrete time signals like unit step, saw tooth, triangular, Sinusoidal, ramp, and sinc function.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
clear all;
close all;
n=-10:1:10;
L=length(n);
for i=1:L
   % to generate a discrete time impulse function
   if n(i)==0
     x1(i)=1;
   else
      x1(i)=0;
   end;
   % to generate a discrete time unit step signal
   if n(i)>=0
     x2(i)=1;
      % to generate a discrete time ramp signal
     x3(i)=n(i);
   else
      x2(i)=0;
      x3(i)=0;
   end;
end;
% to generate exponential sequence
a=0.85;
```

```
x4=a.^n;

figure;
subplot(3,2,1);
stem(n,x1);
xlabel('discrete time n ---->');
ylabel('amplitude---->');
title('Discrete time unit impulse signal');
subplot(3,2,2);
stem(n,x2);xlabel('discrete time n ---->');
ylabel('amplitude---->');
title('Unit step sequence')
subplot(3,2,3);
stem(n,x3);
xlabel('discrete time n ---->');
ylabel('amplitude---->');
title(' Unit ramp sequence');
subplot(3,2,4);
stem(n,x4);xlabel('discrete time n ---->');
ylabel('amplitude---->');
title('discrete time exponential signal');
% to generate  a discrete time signum function
 a=sign(n);
 subplot(3,2,[5,6]);
 stem(n,a);
 xlabel('discrete time n ------>');
 ylabel('amplitude ---->');
 title('discrete time signum function');
   % to generate a discrete time sinc function
 Ts=.2;
n=-30:1:30;
t=n*Ts
Wn=sinc(t);
figure;
 stem(n,Wn);
 xlabel('discrete time n ------->');
 ylabel('amplitude ---->');
 title('discrete time sinc function');
```

RESULT : Thus, the MATLAB Program of the generation of discrete time signals like unit step, saw tooth, triangular, sinusoidal, ramp and sinc functions were successfully executed using MATLAB software.

Experiment No-2c

GENERATION OF PERIODIC CONTINUOUS AND DISCRETE TIME SIGNALS

AIM : To write a MATLAB Program to generate periodic continuous and discrete time signals like square wave, triangular wave ,and saw tooth signal.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
clear all;
close all;
t=-10:0.01:10;
n=-10:1:10;
duty=50;
figure(1);
subplot(2,1,1);
Xt=square(t,duty);
plot(t,Xt);
title('continuous time square wave');
xlabel('continuous time t --->');
ylabel('X(t)');
subplot(2,1,2);
Xn=square(n,duty);
stem(n,Xn);
title('discrete time square wave');
xlabel('discrete time n --->');
ylabel('X(n)');
%  generation of triangular wave
figure(2);
subplot(2,1,1);
f=.1;
Yt=sin(2*pi*f*t);
plot(t,Yt);
```

```
title('continuous time sine wave');
xlabel('continuous time t -->');
ylabel('Y(t)');
subplot(2,1,2);
xn=sin(2*pi*.1*n);
stem(n,xn);
title('discrete time sine wave');
xlabel('discrete time n -->');
ylabel('Y(n)');
%   generation of saw tooth signal
width=1.0;
Zt=sawtooth(t,width);
figure(3);
subplot(2,1,1);
plot(t,Zt);
title('continuous time sawtooth signal');
xlabel('continuous time t -->');
ylabel('Z(t)');
subplot(2,1,2);
Zn=sawtooth(n,width);
stem(n,Zn);
title('discrete time sawtooth signal');
xlabel('discrete time n -->');
ylabel('Z(n)');
```

RESULT : Thus, the MATLAB program of the generation of continuous time signals like unit step, sawtooth, triangular, sinusoidal, ramp and sinc functions were successfully executed using MATLAB software.

Experiment No-03
OPERATIONS ON SIGNALS AND SEQUENCES

AIM : To write a MATLAB program to perform various operations on signals and sequences such as addition, multiplication, scaling, shifting and folding, computation of energy and average power.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
t=0:0.025:1;
A=1;f1=1;f2=2;
s1=A*sin(2*pi*f1*t);
s2=A*sin(2*pi*f2*t);
fprintf('\n 1.operations on continuous time signals \n');
fprintf('\n 2.operations on discrete time signals \n');
ch=input('\n \n enter your choice :');
switch ch
   case 1
        figure
      subplot(5,2,1);
        plot(t,s1);
        grid;
        title('original signal with frequency 1');
        xlabel('time t');
        ylabel('amplitude');
        subplot(5,2,2);
        plot(t,s2);
         grid;
        title('original signal with frequency2');
        xlabel('time t');
        ylabel('amplitude');
        subplot(5,2,3);
        plot(t,s1+s2);
```

```
grid;
xlabel('time t');
ylabel('amplitude');
title('sum of 2 signals');
subplot(5,2,4);
plot(t,s1.*s2);
grid;
xlabel('time t');
ylabel('amplitude');
title('multiplication of s1 and s2');
p=t+0.2;
subplot(5,2,5);
plot(p,s1);
axis([0 1.3 -1 1]);
 grid;
 xlabel('time t');
ylabel('amplitude');
title('time delayed signal');
p=t-0.2;
subplot(5,2,6);
plot(p,s1);
axis([-0.3 1 -1 1]);
grid;
xlabel('time t');
ylabel('amplitude');
title('time advanced signal');
p=t/2;
subplot(5,2,7);
plot(p,s1);
grid;
%axis([0 10 -1 1]);
xlabel('time t');
ylabel('amplitude');
title('compressed signal');
p=2*t;
subplot(5,2,8);
plot(p,s1);
%axis([0 10 -1 1]);
 grid;
xlabel('time t');
ylabel('amplitude');
title('expanded signal');
p=-1*t;
subplot(5,2,9);
plot(p,s1);
grid;
 ylabel('amplitude');
title('reflected signal');
```

```
        subplot(5,2,10);
        plot(t, 3*s1);
        axis([0 1 -4 4]);
        grid;
        xlabel('time');
        ylabel('amplitude');
        title('amplitude scaled signal');
        %sx=s1.^2;sx
        %syms sx t;
        %zx=int(sx,t);zx

case 2
n=-10:1:10;
l=length(n);
f1=0.1;
f2=0.125;
x1=sin(2*pi*f1*n);
x2=sin(2*pi*f2*n);
figure;
subplot(4,2,1);
stem(n,x1);
title('discrete sine wave x1(n) with time period=10');
grid on;
subplot(4,2,2);
stem(n,x2);
title(' discrete sine wave x2(n) with time period=8');
grid on;
x3=x1+x2;
subplot(4,2,3);
stem(n,x3);
title('sum of two discrete time signals');
grid on;
x4=x1.*x2;
subplot(4,2,4);
stem(n,x4);
title('multiplication of two discrete time signals');
grid on;
no=2;
x5=sin(2*pi*f1*(n-no));
subplot(4,2,5);
stem(n,x5);
title(' time shifted signal of x1(n)');
grid on;x6=fliplr(x1);
subplot(4,2,6);
stem(n,x6);
title(' time folded signal of x1(n)');
grid on;
x7=x2.*2;
```

```
subplot(4,2,[7 8]);
stem(n,x7);
title(' amplitude scaled discrete time signal of x2(n) ');
grid on;
e=sum(abs(x7).^2);e
l=length(n);
p=e/l;
p
end
```

RESULT : Thus, the MATLAB program of performing various operations on signals And sequence were successfully executed using MATLAB software.

.

Experiment No-4a

EVEN AND ODD PARTS OF A CONTINUOUS TIME SIGNAL

AIM : To write a MATLAB program to find the even and odd parts of a signal.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
  clear all;
   close all;
   t=-5:0.001:5;
   A=0.8;
 x1=A.^(t);
 x2=A.^(-t);
 if(x2==x1)
     disp('The given signal is even signal');
 else
     if(x2==(-x1))
     disp('The given signal is odd signal');
     else
     disp('The given signal is neither even nor odd');
     end
 end
 xe=(x1+x2)/2;
 xo=(x1-x2)/2;
 subplot(2,2,1);
 plot(t,x1);
 xlabel('time  t ---->');
 ylabel('x(t)');
 title('original signal x(t)');
 subplot(2,2,2);
 plot(t,x2);
 xlabel('time t ---->');
 ylabel('amplitude');
```

```
title('time reflected signal x(-t)');
subplot(2,2,3);
plot(t,xe);
xlabel('time t ---->');
ylabel('amplitude');
title('even part of a signal x(t)');
subplot(2,2,4);
plot(t,xo);
xlabel('time t ---->');
ylabel('amplitude');
title('odd part of a signal x(t)');
figure;
plot(t,xe+xo);
xlabel('time t -----> ');
ylabel('x(t)');
title('reconstructed original signal');
%real part of signal x1
xr=real(x1);
xi=imag(x1);
figure;
subplot(5,1,1);
plot(t,xr);
xlabel('time t -----> ');
ylabel('xr(t)');
title('real part of exponential signal');
grid on;
subplot(5,1,2);
plot(t,xi);
xlabel('time t -----> ');
ylabel('xi(t)');
title('imaginary part of exponential signal');
grid on;
f=2;
x3=exp(j*2*pi*f*t);
subplot(5,1,3);
plot(t,x3);
xlabel('time  ----->');
ylabel('x3(t)');
title('complex exponetial signal');
grid on;
x4=real(x3);
subplot(5,1,4);
plot(t,x4);
xlabel('time  ----->');
ylabel('x4(t)');
title('real part of complex signal');
grid on;
x5=imag(x3);
```

```
subplot(5,1,5);
plot(t,x5);
xlabel('time  ----->');
ylabel('x5(t)');
title('imaginary part of complex signal');
grid on;
```

RESULT : Thus, the MATLAB program of finding even and odd parts of signals was successfully executed using MATLAB software.

.

Experiment No-4b

EVEN AND ODD PARTS OF A DISCRETE TIME SIGNAL(SEQUENCE)

AIM : To write a MATLAB program to find the even and odd parts of a sequence.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
  clear all;
   close all;
   n=-10:1:10
   A=0.8;
 x1=A.^(n);
 x2=A.^(-n);
 %n1=-n
 if(x2==x1)
     disp('The given signal is even signal');
 else
     if(x2==(-x1))
     disp('The given signal is odd signal');
     else
     disp('The given signal is neither even nor odd');
     end
 end
 xe=(x1+x2)/2;
 xo=(x1-x2)/2;
 subplot(2,2,1);
 stem(n,x1);
 xlabel('discrete time n  ---->');
 ylabel('x(n)');
 title('original signal x(n)');
 subplot(2,2,2);
 stem(n,x2);
 xlabel('discrete time n  ---->');
 ylabel('amplitude');
```

```
title('time reflected signal x(-n)');
subplot(2,2,3);
stem(n,xe);
xlabel('discrete time n  ---->');
ylabel('amplitude');
title('even part of a signal x(n)');
subplot(2,2,4);
stem(n,xo);
xlabel('discrete time n  ---->');
ylabel('amplitude');
title('odd part of a signal x(n)');
figure;
stem(n,xe+xo);
xlabel('discrete time n  ---->');
ylabel('x(n)');
title('reconstructed original signal');
%real part of signal x1
xr=real(x1);
xi=imag(x1);
figure;
subplot(5,1,1);
stem(n,xr);
xlabel('discrete time n  ---->');
ylabel('xr(n)');
title('real part of exponential signal');
grid on;
subplot(5,1,2);
stem(n,xi);
xlabel('discrete time n  ---->');
ylabel('xi(t)');
title('imaginary part of exponential signal');
grid on;
f=.1;
x3=exp(j*2*pi*f*n);
subplot(5,1,3);
stem(n,x3);
xlabel('discrete time n  ---->');
ylabel('x3(n)');
title('complex exponetial signal');
grid on;
x4=real(x3);
subplot(5,1,4);
stem(n,x4);
xlabel('time  ----->');
ylabel('x4(n)');
title('real part of complex signal');
grid on;
x5=imag(x3);
```

```
subplot(5,1,5);
stem(n,x5);
xlabel('discrete time n  ---->');
ylabel('x5(n)');
title('imaginary part of complex signal');
grid on;
```

RESULT : Thus, the MATLAB Program of finding even and odd parts of signals was successfully executed using MATLAB software.

.

Experiment No-5
CONVOLUTION OF TWO SEQUENCES

AIM : To write a MATLAB program to find the convolution of two sequences.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
%clear all;
n1=input('enter the initial time value of i/p sequence');
x=input('enter the i/p sequence ');
n2=input('enter the initial time value of impulse response');
h=input('enter the impulse response');
sx=length(x);
sh=length(h);
nx=n1:sx+n1-1;
nh=n2:sh+n2-1;
ny=n1+n2:1:sx+sh+n1+n2-2;
y=conv(x,h);y
figure(1);
subplot(2,2,1);
stem(nx,x);
xlabel('discrete time n');
ylabel('x(n)');
title('input signal');
subplot(2,2,2);
stem(nh,h);
xlabel('discrete time n');
ylabel('h(n)');
title('impulse response');
subplot(2,2,[3,4]);
stem(ny,y);
xlabel('discrete time n');
```

```
ylabel('y(n)');
title('linear convolution');
t=0:0.001:.1;
xt=sin(2*pi*50*t);
figure(2);
subplot(3,1,1);
plot(t,xt);
xlabel('time');
ylabel('xt');
yt=cos(2*pi*50*t);
subplot(3,1,2);
plot(t,yt);
xlabel('time');
ylabel('yt');
zt=conv(xt,yt);
subplot(3,1,3);
plot(zt);
```

RESULT : Thus, the MATLAB Program of finding the convolution between two signals was successfully executed using MATLAB software.

.

OUTPUT:-

```
enter the initial time value of i/p sequence0
enter the i/p sequence [1 2 4 3]
enter the initial time value of impulse response-1
enter the impulse response[4 2 3 1]
y =
    4   10   23   27   20   13
```

Experiment No-06
AUTO-CORRELATION & CROSS-CORRELATION BETWEEN SIGNALS

AIM : To write a MATLAB program to compute autocorrelation and cross correlation between signals.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc; clear all; close all;
t=0:0.01:1;
f1=3;
x1=sin(2*pi*f1*t);
figure;
subplot(2,1,1);
plot(t,x1);
title('sine wave');
xlabel('time ---->');
 ylabel('amplitude---->');
grid;
[rxx lag1]=xcorr(x1);
subplot(2,1,2);
plot(lag1,rxx);
grid;
title('auto-correlation function of sine wave');
figure;
subplot(2,2,1);
plot(t,x1);
title('sine wave x1');
xlabel('time ---->');
ylabel('amplitude---->');
grid;
 f2=2;
```

```
x2=sin(2*pi*f2*t);
subplot(2,2,2);
plot(t,x2);
title('sine wave x2');
xlabel('time ---->');,ylabel('amplitude---->');
grid;
[cxx lag2]=xcorr(x1,x2);
subplot(2,2,[3,4]);
plot(lag2,cxx);
grid;
title('cross-correlation function of sine wave');
```

RESULT : Thus the MATLAB Program of computing auto correlation and cross correlation between signals was successfully executed using MATLAB software.

.

OUTPUT:

Experiment No-7(a)
LINEAR SYSTEM OR NON-LINEAR SYSTEM

AIM : To write a MATLAB program to verify the given system is linear or non-linear.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc; clear all; close all;
x1=input('enter the x1[n] sequence=');    %  [0 2 4 6]
x2=input('enter the x2[n] sequence=');    % [3 5 -2 -5]
if length(x1)~=length(x2)
disp(' length of x2 must be equal to the length of x1');
   return;
end;
h=input('enter the h[n] sequence=');% [-1 0 -3 -1 2 1]
a=input('enter the constant a= ');  % 2
b=input('enter the constant b= ');  %  3
y01=conv(a*x1,h);
y02=conv(b*x2,h);
y1=y01+y02;
x=a*x1+b*x2;
y2=conv(x,h);
L=length(x1)+length(h)-1;
n=0:L-1;
subplot(2,1,1);
stem(n,y1);
label('n --->'); label('amp ---->');
title('sum of the individual response');
subplot(2,1,2);
stem(n,y2);
xlabel('n --->'); ylabel('amp ---->');
title('total response');
if y1==y2
disp('the system is a Linear system');
```

```
else
   disp('the system is a non-linear system');
end;
```

RESULT : Thus, the MATLAB Program of verifying the system is linear ornon linear was successfully executed using MATLAB software.

Experiment No-07(b)

TIME-INVARIANT OR TIME-VARIANT SYSTEM

AIM : To write a MATLAB program to verify the given system is Time –invariant or Time–variant system.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc; clear all; close all;
x=input('enter the sequence x[n]='); %[0 2 3 1 -2 7 3]
h=input('enter the sequence h[n]='); %[4 -5 -11 -3 7 2 6 8 -15]
d=input('enter the positive number for delay d=');  % 5
xdn=[zeros(1,d),x];           % delayed input
yn=conv(xdn,h);               % output for delayed input
y=conv(x,h);                 % actual output
ydn=[zeros(1,d),y];           % delayed output
figure;
subplot(2,1,1);
stem(0:length(x)-1,x);
xlabel('n ---->'),ylabel('amp --->');
title('the sequence x[n] ');
subplot(2,1,2);
stem(0:length(xdn)-1,xdn);
xlabel('n ---->'),ylabel('amp --->');
title('the delayed sequence of x[n] ');
figure;
subplot(2,1,1);
stem(0:length(yn)-1,yn);
xlabel('n ---->'),ylabel('amp --->');
title('the response of the system to the delayed sequence of x[n] ');
subplot(2,1,2);
stem(0:length(ydn)-1,ydn);
xlabel('n ---->'),ylabel('amp --->');
title('the delayed output sequence ');
if yn==ydn
```

```
    disp('the given system is a Time-invarient system');
else
    disp('the given system is a Time-varient system');
end;
```

INPUT SEQUENCE:

Enter the sequence x[n] = [0 2 3 1 -2 7 3]
Enter the sequence h[n] = [4 -5 -11 -3 7 2 6 8 -15]
Enter the positive number for delay d=5
The given system is a Time-invariant system

OUTPUT:

<u>RESULT</u> : Thus, the MATLAB Program of verifying the system is Time –invariant or Time–variant System was successfully executed using MATLAB software.

Experiment No-08

COMPUTATION OF UNIT SAMPLE,UNIT STEP AND SINUSOIDAL RESPONSES OF THE GIVEN LTI SYSTEM AND VERIFYING ITS PHYSICAL REALIZABILITY AND STABILITY PROPERTIES

AIM : To write a MATLAB program to find the impulse response& step response of the LTI system governed by the transfer function H(s) .

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
clear all;
close all;
syms s complex;
H=1/(s^2+4*s+3);
disp('Impulse response of the system h(t) is');
h=ilaplace(H);
simplify(h);
disp(h);
Y=1/(s*(s^2+4*s+3));
disp('Step response of the system is');
y=ilaplace(Y);
simplify(y);
disp(y);
t=0:0.1:20;
h1=subs(h,t);
subplot(2,1,1);
plot(t,h1);
xlabel('time');
ylabel('h(t)');
title('Impulse response of the system');
y1=subs(y,t);
subplot(2,1,2);
plot(t,y1);
xlabel('time');
ylabel('x(t)');
title('step response of the system');
```

RESULT : Thus, the MATLAB program of the generation of impulse response & step response of the LTI system was successfully executed using MATLAB software.

Experiment No-9

GIBBS PHENOMENON

AIM : To write a MATLAB program to construct the periodic square wave represented by its Fourier Series by considering only 3,9,59 terms.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
clear all;
 close all;
N=input('enter the no. of signals to reconstruct=');
n_har=input('enter the no. of harmonics in each signal=');
t=-1:0.001:1;
omega_0=2*pi;
for k=1:N
    x=0.5;
    for n=1:2:n_har(k)
        b_n=2/(n*pi);
        x=x+b_n*sin(n*omega_0*t);
    end
    subplot(N,1,k);
    plot(t,x);
    xlabel('time--->');
    ylabel('amp---->');
    axis([-1 1 -0.5 1.5]);
    text(0.55,1.0,['no.of har=',num2str(n_har(k))]);
end
```

RESULT : Thus, the MATLAB program of Gibbs Phenomenon was successfully verified using MATLAB software.

Experiment No-10(a)

FOURIER TRANSFORMS AND INVERSE FOURIER TRANSFORMS

AIM : To write a MATLAB program to find the Fourier transform and inverse Fourier transforms of given functions.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

To find Fourier transform

```
clc; clear all; close all;
syms t s;syms w real;
syms A real;syms o real;syms b float;
 f=dirac(t);
F=fourier(f);
disp('the fourier transform of dirac(t) =');
disp(F);
 f1=A*heaviside(t);
F1=fourier(f1);
disp('the fourier transform of A =');
disp(F1);
f2=A*exp(-t)*heaviside(t);
F2=fourier(f2);
disp('the fourier transform of exp(-t) =');
disp(F2);
 f3=A*t*exp(-b*t)*heaviside(t);
F3=fourier(f3);
disp('the fourier transform of A*t*exp(-b*t)*u(t) =');
disp(F3);
 f4=sin(o*t);
F4=fourier(f4);
disp('the fourier transform of sin(o*t) =');
disp(F4);
```

To find inverse Fourier transforms of Given functions.

OUTPUT:-

RESULT : Thus the MATLAB program of finding the Fourier transform and inverse Fourier transform of given functions was successfully executed using MATLAB software.

Experiment no-10(b)

MAGNITUDE AND PHASE SPECTRUM OF FOURIER TRANSFORMS

AIM : To write a MATLAB program to find Fourier transform of the given signal and to plot its magnitude and phase spectrum.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc; clear all; close all;
syms t s ;
syms w float;
f=3*exp(-t)*heaviside(t);          % given function
F=fourier(f);                     % to find Fourier Transform
disp('the fourier transform of 3*exp(-t)*u(t) =');
disp(F);                   % to display the result in the command window
w=-2*pi:pi/50:2*pi;
F1=subs(F,w);             % substitute w in F function
Fmag=abs(F1);              % to find magnitude
Fphas=angle(F1);         % to find phase
subplot(2,1,1);
plot(w,Fmag);
xlabel('w ---->');
ylabel('Magnitude --->');
title('Magnitude spectrum');
grid;
subplot(2,1,2);
plot(w,Fphas);
xlabel('w ---->');
ylabel('Phase in radians--->');
title('Phase spectrum');
grid;
```

OUTPUT:-

RESULT : Thus the MATLAB program of finding Fourier transform and ploting magnitude and Phase spectrums were successfully completed using MATLAB software.

Experiment No-11

LAPLACE TRANSFORM

AIM : To write a MATLAB program to plot the time domain and its frequency domain of a given function using Laplace Transform .

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
syms t s;
f=1.5-2.5*t*exp(-2*t)+1.25*exp(-3*t);
a=simplify(f);
disp('The given time domain function is = ')
pretty(a);
figure(1);
ezplot(f);
F=laplace(f,t,s);
disp('The obtained frequency domain function is = ')
pretty(F);
figure(2)
ezplot(F);
figure(2);
f=ilaplace(F);
simplify(f);
disp('The synthesis function is = ')
pretty(f);
figure(3);
ezplot(f);
```

RESULT : Thus,the MATLAB program of plotting the time domain and its frequency domain was successfully executed using MATLAB software.

Experiment No-12(a)
ZEROS AND POLES IN S- PLANE

AIM : To Write a MATLAB program to find the poles,zeros and to plot pole-zero map in S-Plane .

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc; clear all; close all;
num=input('enter the numerator polynomial vector\n');    % [1 -2 1]
den=input('enter the denominator polynomial vector\n');    % [1 6 11 6]
H=tf(num,den)
[p z]=pzmap(H);
disp('zeros are at ');
disp(z);
disp('poles are at ');
disp(p);
pzmap(H);
if max(real(p))>=0
   disp(' All the poles do not lie in the left half of S-plane ');
   disp(' the given LTI system is not a stable system ');
else
   disp('All the poles lie in the left half of S-plane ');
   disp(' the given LTI system is a stable system ');
end;
```

RESULT : Thus, the MATLAB program of finding and plotting pole-zero map in S-plane was successfully executed using MATLAB software.

Experiment No-12(b)
ZEROS AND POLES IN Z- PLANE

AIM : To Write a MATLAB program to find the poles,zeros and to plot pole-zero map in Z-Plane .

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc; clear all; close all;
num=input('enter the numerator polynomial vector \n'); %[1 0 0]
den=input('enter the denominator polynomial vector \n');%[1 1 0.16]
H=filt(num,den)
z=zero(H);
disp('the zeros are at ');
disp(z);
[r p k]=residuez(num,den);
disp('the poles are at ');
disp(p);
zplane(num,den);
title('Pole-Zero map in the Z-plane');
if max(abs(p))>=1
   disp('all the poles do not lie with in the unit circle');
   disp('hence the system is not stable');
else
   disp('all the poles lie with in the unit circle');
   disp('hence the system is stable');
end;
```

RESULT : Thus, the MATLAB program of finding and plotting pole-zero map in Z-plane was successfully executed using MATLAB software.

Experiment No-13
GAUSSIAN NOISE

AIM : To write a MATLAB program to generate a Gaussian noise and to compute its Mean, Mean Square Value, Skew, Kurtosis, PSD, Probability Distribution function.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc; clear all; close all;
t=-10:0.01:10;
L=length(t);
n=randn(1,L);
subplot(2,1,1);
plot(t,n);
xlabel('t --->'),ylabel('amp ---->');
title('normal random function');
nmean=mean(n);
disp('mean=');disp(nmean);
nmeansquare=sum(n.^2)/length(n);
disp('mean square=');disp(nmeansquare);
nstd=std(n);
disp('std=');disp(nstd);
nvar=var(n);
disp('var=');disp(nvar);
nskew=skewness(n);
disp('skew=');disp(nskew);
nkurt=kurtosis(n);
disp('kurt=');disp(nkurt);
p=normpdf(n,nmean,nstd);
subplot(2,1,2);
stem(n,p)
```

RESULT : Thus, the MATLAB Program of generation of Gaussian noise and computation of its Mean, Mean Square Value,Skew, Kurtosis, PSD, Probability Distribution function was successfully executed using MATLAB software.

Experiment No-14
SAMPLING THEOREM

AIM : To write a MATLAB Program to verify the sampling theorem.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
close all;
clear all;
f1=3;
f2=23;
t=-0.4:0.0001:0.4;
x=cos(2*pi*f1*t)+cos(2*pi*f2*t);
figure(1);
plot(t,x,'-.r');
xlabel('time-----');
ylabel('amp---');
title('The original signal');
%case 1: (fs<2fm)
fs1=1.4*f2;
ts1=1/fs1;
n1=-0.4:ts1:0.4;
xs1=cos(2*pi*f1*n1)+cos(2*pi*f2*n1);

figure(2);
stem(n1,xs1);
hold on;
plot(t,x,'-.r');
hold off;
legend('fs<2fm');
%case 2: (fs=2fm)
fs2=2*f2;
ts2=1/fs2;
n2=-0.4:ts2:0.4;
```

```
xs2=cos(2*pi*f1*n2)+cos(2*pi*f2*n2);
figure(3);
stem(n2,xs2);
hold on;
plot(t,x,'-.r');
hold off;
legend('fs=2fm');
%case 3: (fs>2fm)
fs3=7*f2;
ts3=1/fs3;
n3=-0.4:ts3:0.4;
xs3=cos(2*pi*f1*n3)+cos(2*pi*f2*n3);
figure(4);
stem(n3,xs3);
hold on;
plot(t,x,'-.r');
hold off;
legend('fs>2fm');
```

<u>RESULT</u> : Thus, the MATLAB program of verification of sampling theorem was successfully executed using MATLAB software.

Experiment No-15
AUTO-CORRELATION/CROSS-CORRELATION

AIM : To write a MATLAB program to detect the periodic signal in the presence of noise by using Auto correlation and Cross Correlation method.

SOFTWARE REQURIED :

MATLAB R2006 (7.3 Version).

PROCEDURE :

- Open MATLAB Software
- Open new M-file
- Type the program
- Save in current directory
- Run the program
- For the output see command window\ Figure window.

PROGRAM :

```
clc;
clear all;
close all;
t=0:0.01:10;
s=cos(2*pi*3*t)+sin(2*pi*5*t);  % periodic signal
figure;
subplot(2,1,1);
plot(t,s);
axis([0 10 -2 2]);
xlabel(' t ---->'),ylabel(' amp ----> ');
title('the periodic signal');
L=length(t);
n=randn(1,L);  % noise signal
subplot(2,1,2);
plot(t,n);
xlabel(' t ---->'),ylabel(' amp ----> ');
title('the noise signal');
L=length(t);
f=s+n;  % received signal
figure;
subplot(2,1,1);
plot(t,f);
xlabel(' t ---->'),ylabel(' amp ----> ');
title('the received signal');
rxx=xcorr(f,s,200);
```

```
subplot(2,1,2);
plot(rxx);
title('the Correlator output');
```

RESULT : Thus, the MATLAB Program of detecting the periodic signal masked by noise using Auto Correlation &Cross Correlation method was performed successfully using MATLAB software .