EXP 4: To Perform Sampling of a signal

```
clc; close all; clear all;

t=-10:0.01:10;
T=8;
fm=1/T;

x=cos(2*pi*fm*t);
fs1=1.2*fm;
fs2=2*fm;
fs3=8*fm;
n1=-4:1:4;
% Code for under sampling (fs<2fm)
xn1=cos(2*pi*n1*fm/fs1);
subplot(221)
plot(t,x);
xlabel('time in seconds');
ylabel('x(t)');
title('continous time signal');
subplot(222)
stem(n1,xn1);
hold on;
plot(n1,xn1);
xlabel('n');
ylabel('x(n)');
title('discrete time signal with fs<2fm');

% Code for critical sampling (fs=2fm)
n2=-5:1:5;
xn2=cos(2*pi*n2*fm/fs2);
subplot(223)
stem(n2,xn2);
hold on;
plot(n2,xn2);
xlabel('n');
ylabel('x(n)');
title('discrete time signal with fs=2fm');

% Code for over sampling (fs>2fm)
n3=-20:1:20;
xn3=cos(2*pi*n3*fm/fs3);
subplot(224)
stem(n3,xn3);
hold on;
plot(n3,xn3);
xlabel('n');
ylabel('x(n)');
title('discrete time signal with fs>2fm');
```

Exp 5: To Perform Amplitude Shift Keying.

```
clear all;
close all;
%GENERATE CARRIER SIGNAL
Tb=1; fc=10;
t=0:Tb/100:1;
c=sqrt(2/Tb)*sin(2*pi*fc*t);
%generate message signal
N=8;
m=rand(1,N);
t1=0;t2=Tb
for i=1:N
t=[t1:.01:t2]
if m(i)>0.5
m(i)=1;
m_s=ones(1,length(t));
else
m(i)=0;
m_s=zeros(1,length(t));
end
message(i,:)=m_s;
%product of carrier and message
ask_sig(i,:)=c.*m_s;
t1=t1+(Tb+.01);
t2=t2+(Tb+.01);
%plot the message and ASK signal
subplot(4,1,2);
axis([0 N -2 2]);
plot(t,message(i,:),'r');
title('message signal');xlabel('t--->');ylabel('m(t)');grid on
hold on
subplot(4,1,4);plot(t,ask_sig(i,:));
title('ASK signal');xlabel('t--->');ylabel('s(t)');grid on
hold on
end
hold off
%Plot the carrier signal and input binary data
subplot(4,1,3);plot(t,c);
title('carrier signal');xlabel('t--->');ylabel('c(t)');grid on
subplot(4,1,1);stem(m);
title('binary data bits');xlabel('n--->');
```

Exp 6: To Perform Phase Shift Keying.

```
clc; clear all; close all;

b = input('Enter the Bit stream \n '); %b = [0 1 0 1 1 1 0];
n = length(b);
t = 0:0.01:n;
x = 1:1:(n+1)*100;

% Conversion of Binary input to NRZ signal
for i = 1:n
    if (b(i) == 0)
        b_p(i) = -1;
    else
        b_p(i) = 1;
    end
    for j = i:.1:i+1
        bw(x(i*100:(i+1)*100)) = b_p(i);
    end
end

bw = bw(100:end);
sint = sin(2*pi*t);
st = bw.*sint;

subplot(3,1,1)
plot(t,bw)
grid on ; axis([0 n -2 +2])
xlabel('time in seconds');
ylabel('Amplitude');
title('NRZ Binary Input Signal');

subplot(3,1,2)
plot(t,sint)
grid on ; axis([0 n -2 +2])
xlabel('time in seconds');
ylabel('Amplitude');
title('Carrier Signal');

subplot(3,1,3)
plot(t,st)
grid on ; axis([0 n -2 +2])
xlabel('time in seconds');
ylabel('Amplitude');
title('PSK Signal');
```

Exp 7:To Perform Frequency Shift Keying

```
clc; clear all; close all;
%GENERATE CARRIER SIGNAL
Tb=1; fc1=2; fc2=5;
t=0:(Tb/100):Tb;
c1=sqrt(2/Tb)*sin(2*pi*fc1*t);
c2=sqrt(2/Tb)*sin(2*pi*fc2*t);

%generate message signal
N=8;
% m=rand(1,N);
m = input('Enter the 8-Bit stream \n '); %b = [0 1 0 1 0 1 1 0];
t1=0;t2=Tb
for i=1:N
    t=[t1:(Tb/100):t2];
    if m(i)>0.5
        m(i)=1;
        m_s=ones(1,length(t));
        invm_s=zeros(1,length(t));
    else
        m(i)=0;
        m_s=zeros(1,length(t));
        invm_s=ones(1,length(t));
    end
    message(i,:)=m_s;

    % Multiplier
    fsk_sig1(i,:)=c1.*m_s;
    fsk_sig2(i,:)=c2.*invm_s;
    fsk=fsk_sig1+fsk_sig2;

    %plotting the message signal and the modulated signal
    subplot(3,2,2);axis([0 N -2 2]);plot(t,message(i,:),'r');
    title('message signal');
    xlabel('t---->');ylabel('m(t)');grid on;hold on;

    subplot(3,2,5);plot(t,fsk(i,:));
    title('FSK signal');
    xlabel('t---->');ylabel('s(t)');grid on;hold on;

    subplot(3,2,1);stem(m);
    title('binary data');xlabel('n---->'); ylabel('b(n)');grid on;hold
on;

    subplot(3,2,3);plot(t,fsk_sig1(i,:));
    title('fsk signal-1');
    xlabel('t---->');ylabel('c1(t)');grid on;hold on;

    subplot(3,2,4);plot(t,fsk_sig2(i,:));
    title('carrier signal-2');
    xlabel('t---->');ylabel('c2(t)');grid on;hold on;
```

```
    t1=t1+(Tb+.01); t2=t2+(Tb+.01);
end
hold off
```