Guru Gobind Singh Indraprastha University

University School of Automation & Robotics

Principles of Communication Systems Lab (ARI353)

Practical File

Name	Sujal Singh
Enrollment Number	04119051723
Batch	IIOT–B1–23

Index

S. No.	Date	Question	Remarks
1.		To study Amplitude Modulation using MATLAB.	
2.		To study the working Of DBSC modulator and demodulator.	
3.		To study and generate single side band (SSB) using MATLAB.	
4.		To study and generate frequency modulation and demodulation using MATLAB.	
5.		To visualise Frequency Modulation and Phase Modulation using MATLAB.	
6.		To verify Sampling Theorem and its reconstruction using MATLAB.	
7.		To study and generate Pulse Amplitude Modulation using MATLAB.	
8.		To study Pulse Width modulation and demodulation technique using MATLAB.	
9.		To study Pulse Position Modulation and Demodulation technique using MATLAB.	
10.		To generate modulate and demodulate amplitude shift key (ASK signal) using MATLAB.	

Experiment–1

Aim:

To study the amplitude modulation using MATLAB

Description:

MATLAB software with essential configurations.

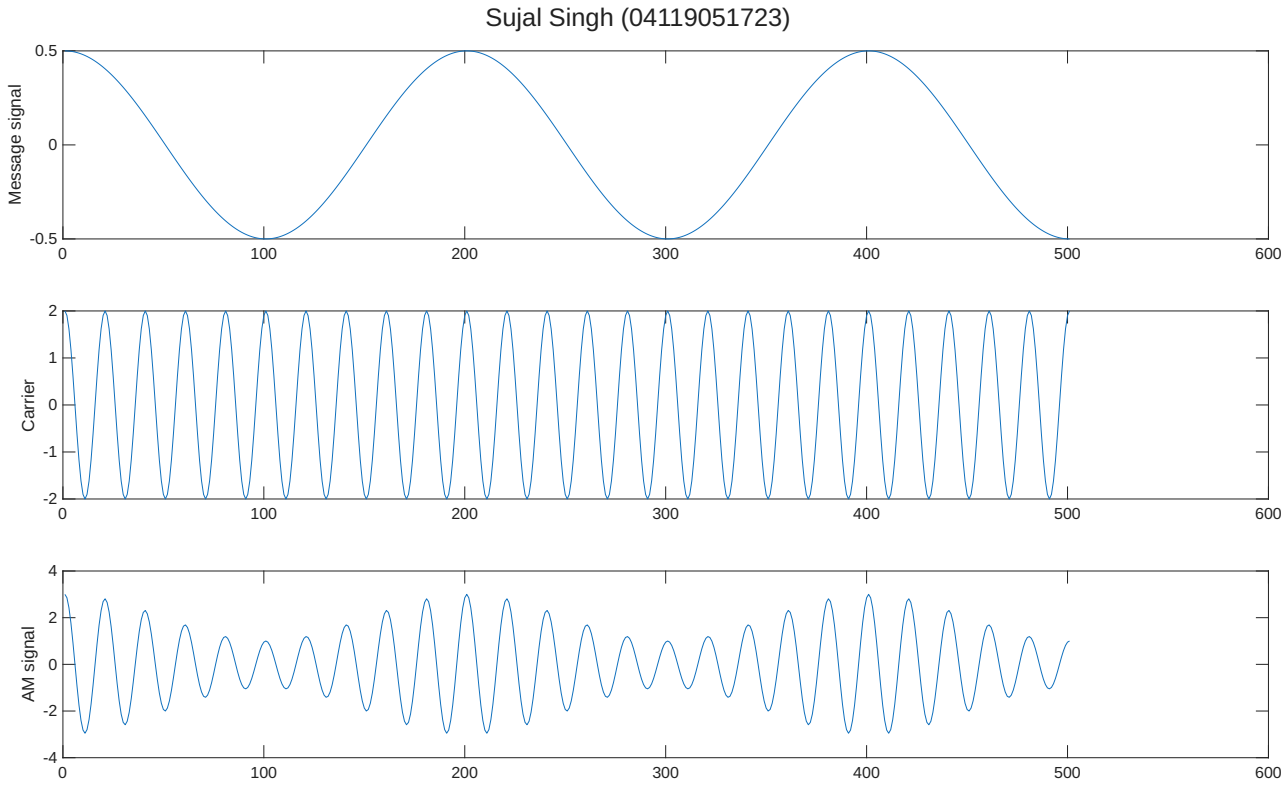
Theory:

Amplitude modulation involves altering the amplitude of a carrier wave while maintaining a constant carrier frequency, thereby encoding original information signals. This process entails transmitting wave signals by modulating their amplitudes. Widely known as AM, it serves as a prevalent method for transmitting data via radio carrier waves, primarily utilized in electronic communication systems.

Code:

```
clc;
clear all;
close all;
Ac = 2;         % carrier amplitude
fc = 0.5;       % carrier frequency
Am = 0.5;       % message signal amplitude
fm = 0.05;      % message signal frequency
Fs = 100;       % sampling rate/frequency
ka = 1;         % Amplitude Sensitivity
t = [0:0.1:50];              % time range in steps of 0.1
ct = Ac * cos(2 * pi * fc * t);    % carrier signal
mt = Am * cos(2 * pi * fm * t);    % message signal
AM = ct .* (1 + ka * mt);          % amplitude modulated signal
subplot(3,1,1);           % Message signal
plot(mt);
ylabel('Message signal');
subplot(3,1,2);           % Carrier signal
plot(ct);
ylabel('Carrier');
subplot(3,1,3);           % AM signal
plot(AM);
ylabel('AM signal')
sgtitle('Sujal Singh (04119051723)')
```

Output:


Sujal Singh (04119051723)
Message signal
Carrier
AM signal
0.5
0
-0.5
2
1
0
-1
-2
4
2
0
-2
-4
0
100
200
300
400
500
600

Experiment–2

Aim:

To study the working of DBSC modulator and demodulator

Description:

MATLAB software with essential configurations.

Theory:

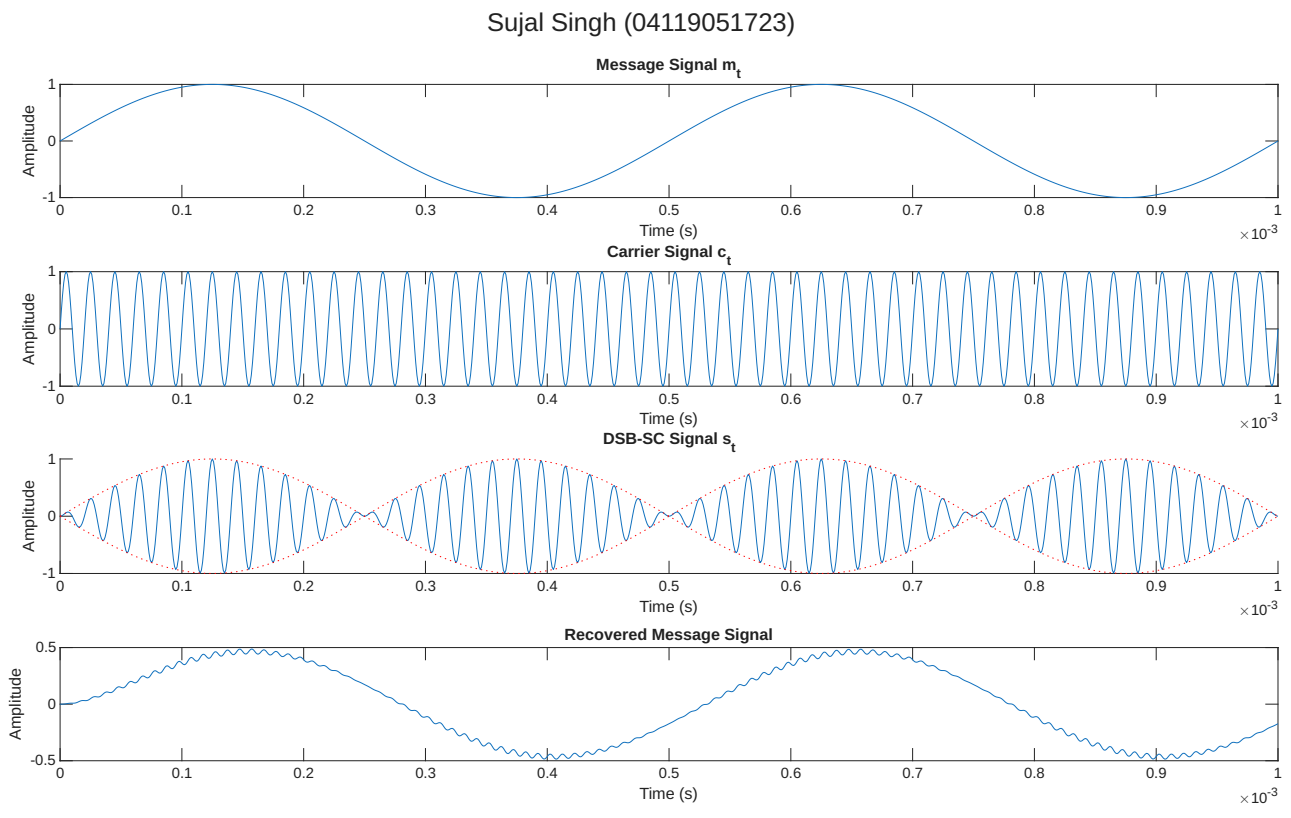
DBSC modulation involves modulating a carrier signal with both upper and lower sidebands containing the message signal, followed by the suppression of one sideband and the carrier itself. At the receiver, the carrier is recovered, and the appropriate sideband is selected for demodulation. This technique optimizes bandwidth usage and spectral efficiency in communication systems, reducing interference and improving signal quality. Overall, DBSC modulation and demodulation offer a streamlined approach to transmitting information effectively while conserving valuable frequency spectrum resources.

Code:

```
%program for dsbsc modulation and demodulation
close all;
clear all;
clc;

t = 0:0.000001:0.001;
Vm = 1;
Vc = 1;
fm = 2000;
fc = 50000;

m_t = Vm * sin(2 * pi * fm * t);
subplot(4,1,1);
plot(t, m_t);
title('Message Signal m_t');
xlabel('Time (s)');
ylabel('Amplitude');

c_t = Vc * sin(2 * pi * fc * t);
subplot(4,1,2);
plot(t, c_t);
title('Carrier Signal c_t');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(4,1,3);
s_t = m_t .* c_t;
hold on;
```

```
29 plot(t, s_t);
30 plot(t,m_t,'r:');
31 plot(t,-m_t,'r:');
32 hold off;
33 title('DSB-SC Signal s_t');
34 xlabel('Time (s)');
35 ylabel('Amplitude');
36
37 r_t = s_t .* c_t;
38
39 [b, a] = butter(1, 0.01);
40 mr = filter(b, a, r_t);
41 subplot(4,1,4);
42 plot(t, mr);
43 title('Recovered Message Signal');
44 xlabel('Time (s)');
45 ylabel('Amplitude');
46 sgtitle('Sujal Singh (04119051723)')
```

Output:



Experiment–3

Aim:

To study and generate single side band (SSB) using MATLAB

Description:

MATLAB software with essential configurations.

Theory:

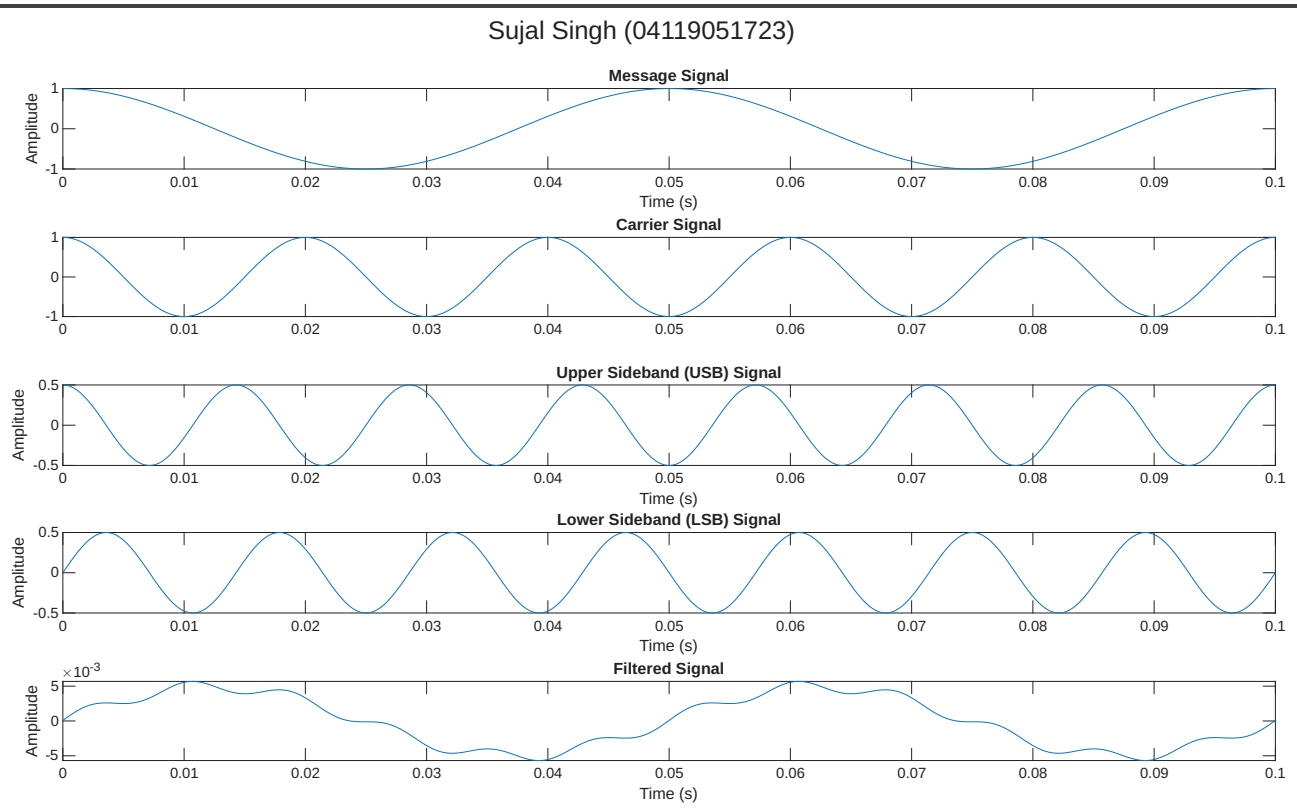
Single Sideband Suppressed Carrier (SSBSC) modulation is a technique where only one sideband, along with the carrier, is transmitted while the other sideband and carrier are suppressed. This method effectively reduces bandwidth usage compared to double-sideband modulation, making it more spectrally efficient. At the receiver, the carrier is recovered, and the single sideband is demodulated to retrieve the original message signal. SSBSC modulation is commonly used in radio communications, especially for long-distance transmission, where bandwidth conservation is crucial. Overall, SSBSC modulation offers improved spectral efficiency and reduced interference, enhancing the performance of communication systems.

Code:

```
% to generate ssb using phase methood and detection of ssb signal using
% matlab
close all
clear all
clc
fs = 8000;
fm =20;
fc = 50;
Am = 1;
Ac =1;
t=[0:0.1*fs]/fs;
subplot(5,1,1);
m1 = Am * cos(2 * pi * fm * t);
plot(t, m1);
title('Message Signal');
xlabel('Time (s)');
ylabel('Amplitude');
m2 = Am*sin(2*pi*fm*t);
subplot(5,1,2)
c1=Ac*cos(2*pi*fc*t);
plot(t,c1);
title('Carrier Signal');
c2=Ac*sin(2*pi*fc*t);
subplot(5,1,3)
S_usb = 0.5*m1.*c1-0.5*m2.*c2;
plot(t, S_usb);
title('Upper Sideband (USB) Signal');
```

```
xlabel('Time (s)');
ylabel('Amplitude');
subplot(5,1,4);
S_lsb = 0.5*m1.*c2 + 0.5*m2.*c1;
plot(t, S_lsb);
title('Lower Sideband (LSB) Signal');
xlabel('Time (s)');
ylabel('Amplitude');
r=S_usb.*c1;
subplot(5,1,5)
[b a] = butter(1,0.0001);
mr=filter(b,a,r);
plot(t,mr);
title('Filtered Signal');
xlabel('Time (s)');
ylabel('Amplitude');
sgtitle('Sujal Singh (04119051723)')
```

Output:



Experiment–4

Aim:

To study and generate frequency modulation and demodulation using MATLAB.

Description:

MATLAB software with essential configurations.

Theory:

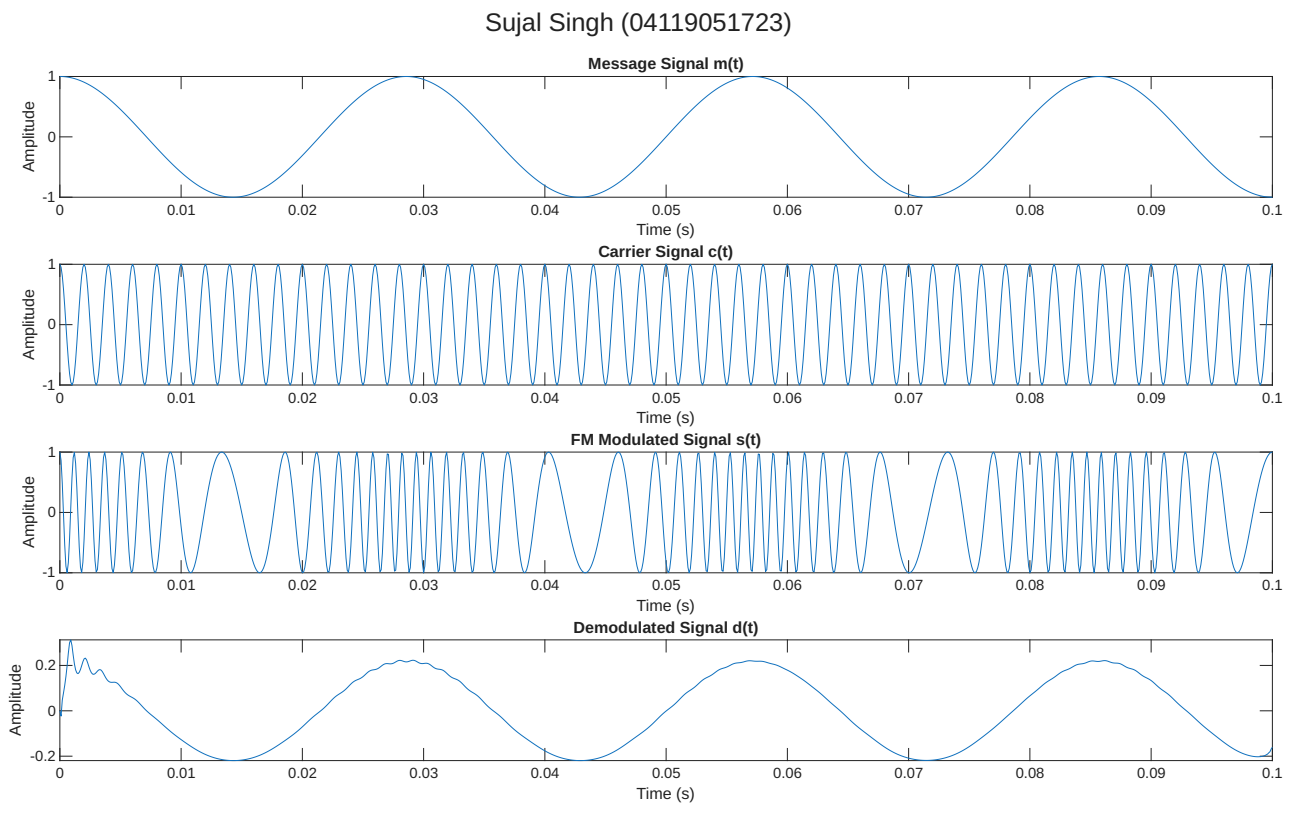
Frequency modulation is a technique or a process of encoding information on a particular signal (analogue or digital) by varying the carrier wave frequency in accordance with the frequency of the modulating signal. If we observe the graph, we can notice that the frequency of a carrier increases when the amplitude of the input signal is increased. Here, the carrier frequency is maximum when the input signal is at its highest. Also, the frequency of a carrier decreases if the amplitude of the modulating signal goes down. What it means is that the carrier frequency is minimum when the input signal is at its lowest. On the other line of spectrum, after the successful modulation recovery of the message signal is another job. When there is modulation, usually , we need to successfully demodulate it and, at the same time, recover the original signal. In such cases, an FM demodulator, also known as an FM discriminator or FM detector, is used. While there are several types of FM demodulators, the main functionality of these devices is to convert the frequency variations of the input signal into amplitude variations of the output signal. The demodulators are used along with an audio amplifier or possibly a digital interface.

Code:

```
%fm modulation and demodulation
close all
clear all
clc
fs=10000;
Am=1;
Ac=1;
fm=35;
fc=500;
B=10;
t=(0:.1*fs)/fs;
wc=2*fc*pi;
wm=2*fm*pi;
m_t=Am*cos(wm*t);
subplot(4,1,1)
plot(t,m_t);
title('Message Signal m(t)');
xlabel('Time (s)');
ylabel('Amplitude');
c_t=Ac*cos(wc*t);
subplot(4,1,2)
```

```
plot(t, c_t);
title('Carrier Signal c(t)');
xlabel('Time (s)');
ylabel('Amplitude');
s_t=Ac*cos(wc*t + B*sin(wm*t));
subplot(4,1,3)
plot(t, s_t);
title('FM Modulated Signal s(t)');
xlabel('Time (s)');
ylabel('Amplitude');
d=demod(s_t,fc,fs,'fm');
subplot(4,1,4)
plot(t, d);
title('Demodulated Signal d(t)');
xlabel('Time (s)');
ylabel('Amplitude');
sgtitle('Sujal Singh (04119051723)')
```

Output:



Experiment–5

Aim:

To visualise Frequency Modulation and Phase Modulation using MATLAB.

Description:

MATLAB software with essential configurations.

Theory:

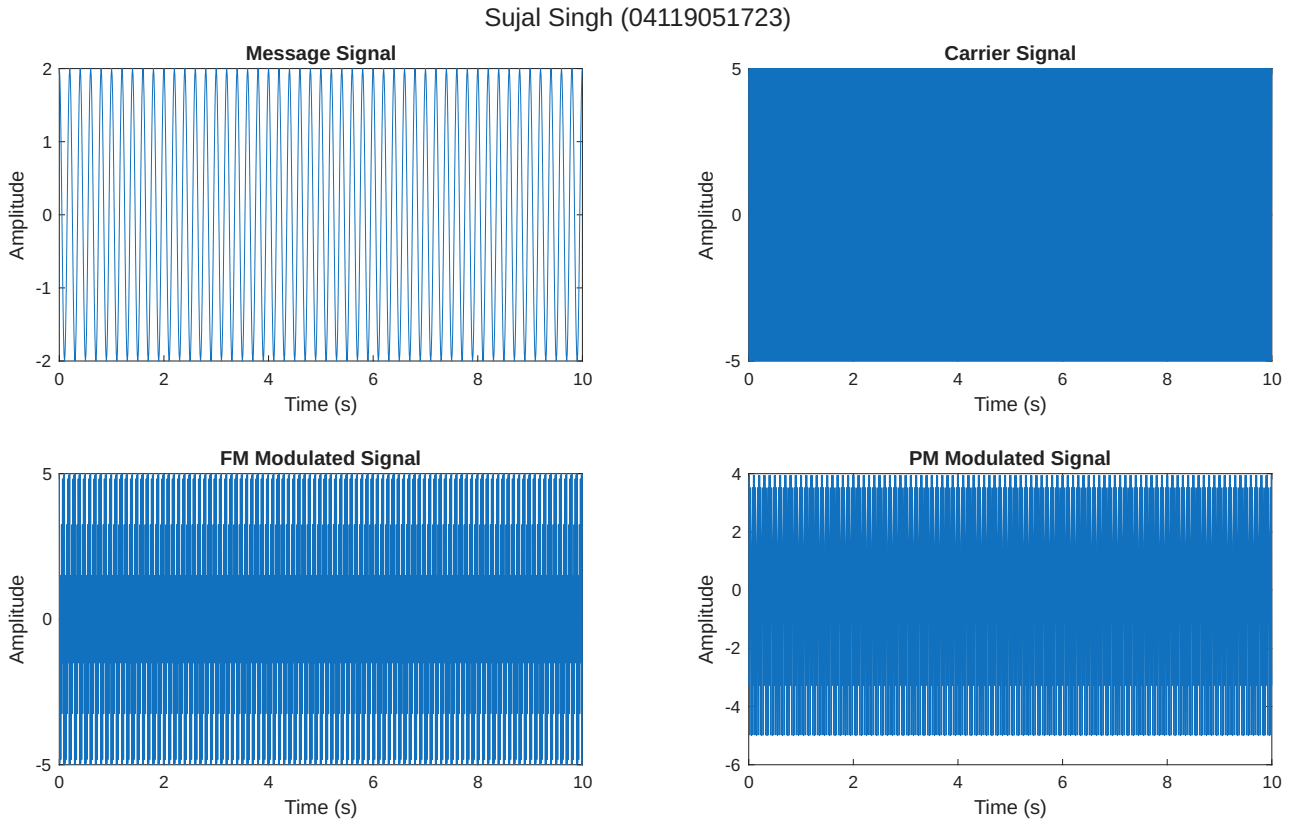
Frequency Modulation (FM) and Phase Modulation (PM) are two important forms of angle modulation, where the angle (phase) of the carrier signal is varied according to the message signal, while the amplitude remains constant. Both techniques offer advantages such as better noise immunity and efficient use of power, making them widely used in radio broadcasting, telecommunications, and digital communication systems.

Code:

```
%to visualize the fm and phase modulation using matlab
clear all
clc
t=0:0.01:10;
Am=input('Enter Message Signal Amplitude(Am) = ');
fm=input('Enter Message Signal Frequency(fm) = ');
Ac=input('Enter Carrier Signal Amplitude(Ac) = ');
fc=input('Enter Carrier Signal Frequency(fc) = ');
kf=input('Frequency Sensitivity(kf) = ');
kp=input('Phase Sensitivity(kp) = ');
wm=2*pi*fm;
m=Am*cos(wm*t);
subplot(2,2,1)
plot(t, m);
hold on
title('Message Signal');
xlabel('Time (s)');
ylabel('Amplitude');
wc=2*pi*fc;
c=Ac*cos(wc*t);
subplot(2,2,2)
plot(t, c);
hold on
title('Carrier Signal');
xlabel('Time (s)');
ylabel('Amplitude');
s_fm = Ac*cos(wc*t + 2*pi*kf*cumsum(m));
subplot(2,2,3)
plot(t, s_fm);
hold on
```

```
title('FM Modulated Signal');
xlabel('Time (s)');
ylabel('Amplitude');
s_pm = Ac*cos(wc*t + kp*m);
subplot(2,2,4)
plot(t, s_pm);
hold on
title('PM Modulated Signal');
xlabel('Time (s)');
ylabel('Amplitude');
sgtitle('Sujal Singh (04119051723)')
```

Output:



Experiment–6

Aim:

To study sampling theorem and it's reconstruction using MATLAB software.

Description:

MATLAB SN Software with communication toolbox

Theory:

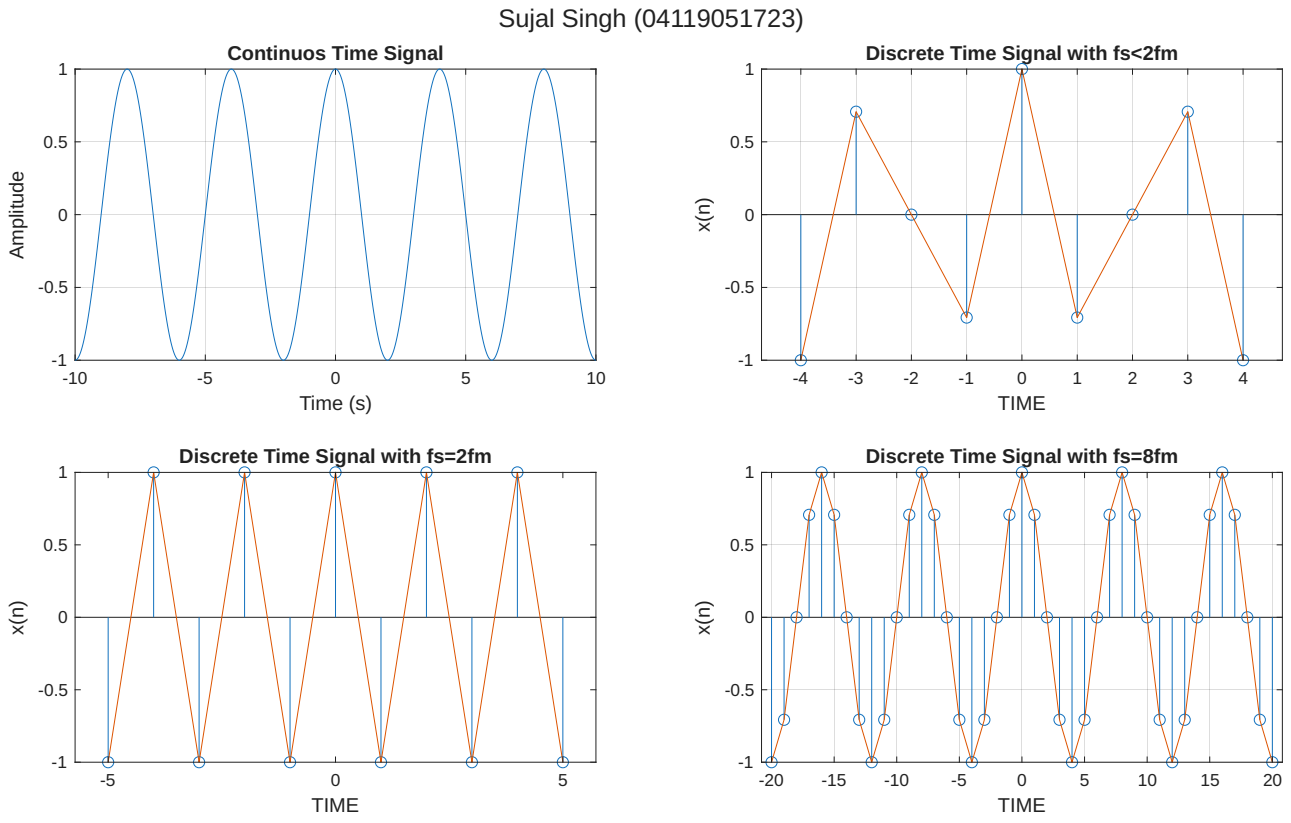
The Sampling Theorem, also known as the Nyquist-Shannon Sampling Theorem, states that to accurately reconstruct a continuous signal from its samples, the sampling frequency must be at least twice the highest frequency component present in the signal. In other words, if a continuous signal contains frequencies up to max f max , then it should be sampled at a frequency, f_s such that $f_s \geq 2 \times f_{\max} f_s \geq f_{\max}$ to avoid aliasing and loss of information.

Code:

```
%to study the sampling theorem and it's reconstruction
close all;
clear all;
clc
t=-10:.01:10;
T=4;
fm = 1/T;
x=cos(2*pi*fm*t);
subplot(2,2,1);
plot(t,x);
title('Continuos Time Signal');
xlabel('Time (s)');
ylabel('Amplitude');
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;
% Generate discrete time signals for fs = 2fm and fs = 8fm
n2 = -5:1:5;
x2 = cos(2*pi*fm*n2/fs2);
```

```
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);
n3=-20:1:20;
x3 = cos(2*pi*fm*n3/fs3);
subplot(2,2,4);
stem(n3,x3);
xlabel('TIME'); ylabel('x(n)');
title('Discrete Time Signal with fs=8fm');
hold on;
subplot(2,2,4);
plot(n3,x3);
grid;
sgtitle('Sujal Singh (04119051723)')
```

Output:



Experiment–7

Aim:

To study Pulse Amplitude Modulation using MATLAB.

Description:

MATLAB software with essential configurations.

Theory:

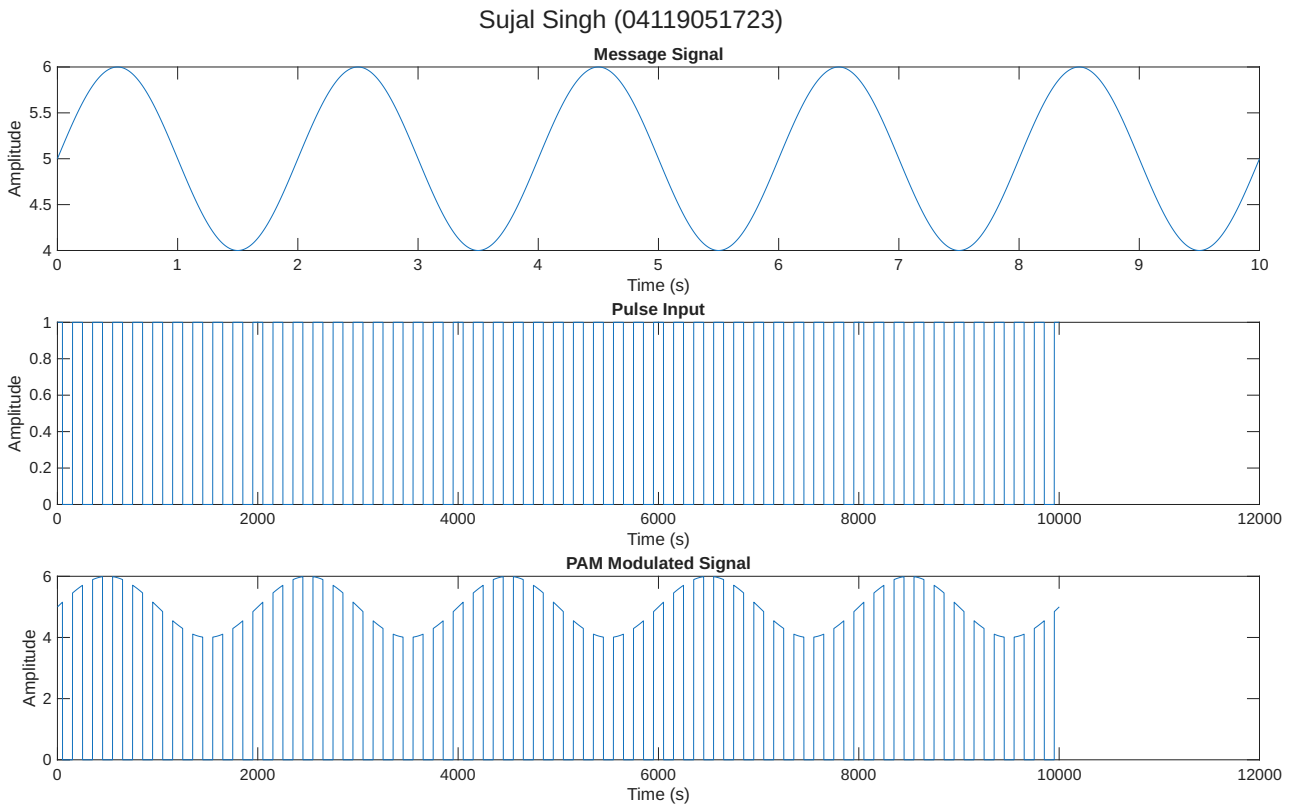
Pulse Amplitude Modulation (PAM) is a form of signal modulation where the amplitude of a series of pulses is varied in accordance with the amplitude of a modulating signal. In simpler terms, it's a method used to encode analog signals into digital signals by varying the amplitude of the pulses.

Code:

```
%pulse amplitude modulation
%1. to study the pulse amp and demod
%2. to study pulse width
%3. to generate pulse positiob=n
% Pulse Amplitude Modulation (PAM)
close all; clear; clc;
t = 0:1/1e3:10;
d = 0:1/5:10;
x = 5 + sin(2*pi*(1/4)*2*t);
figure;
subplot(3,1,1);
plot(t, x);
title('Message Signal');
xlabel('Time (s)');
ylabel('Amplitude');

% Generate pulse input
y = pulstran(t, d, 'rectpuls', 0.1);
subplot(3,1,2);
plot(y);
title('Pulse Input');
xlabel('Time (s)');
ylabel('Amplitude');

% PAM modulation
z = x.*y;
subplot(3,1,3);
plot(z);
title('PAM Modulated Signal');
xlabel('Time (s)');
ylabel('Amplitude');
```

```
32 sgtitle('Sujal Singh (04119051723)')
```

Output:



Experiment–8

Aim:

To study Pulse Width modulation and demodulation technique using MATLAB software.

Description:

MATLAB software with essential configurations.

Theory:

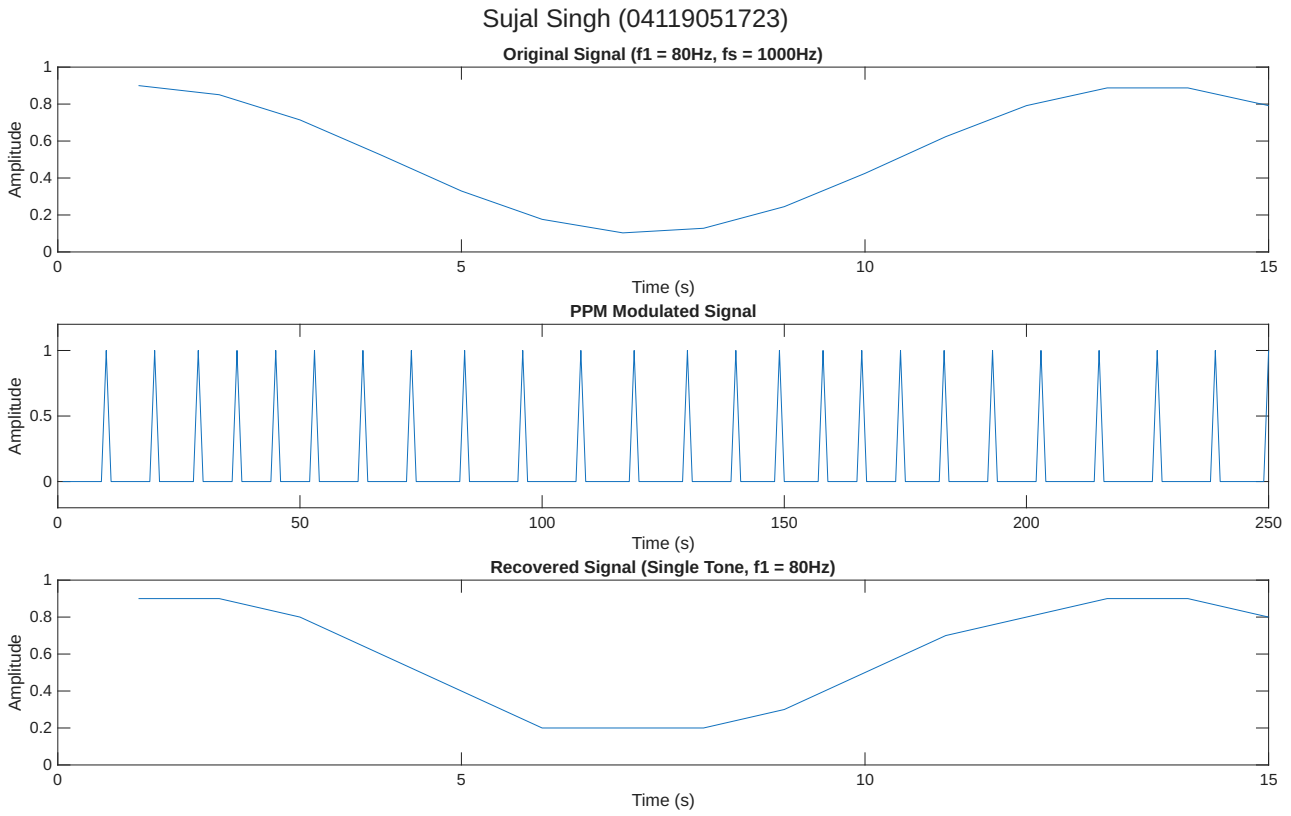
Pulse Width Modulation (PWM) is a pulse modulation technique in which the width (duration) of each pulse is varied in proportion to the instantaneous amplitude of the message signal, while the pulse amplitude and position remain constant. The modulation conveys information through the changing duty cycle of the pulses. PWM is simple to generate and widely used in communication systems, motor speed control, and power electronics.

Code:

```
%pulse width modulation and demodulation
close all; clear; clc;

fc = 1000;          % Carrier frequency
fs = 10000;         % Sampling frequency
f1 = 200;           % Message frequency

t = 0:1/fs:((2/f1)-(1/fs));
x1 = 0.4 * cos(2*pi*f1*t) + 0.5;

% Modulation
y1 = modulate(x1, fc, fs, 'pwm');

subplot(3,1,1);
plot(x1);
axis([0 50 0 1]);
title('Original Signal (f1 = 200Hz, fs = 10000Hz)');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(3,1,2);
plot(y1);
axis([0 50 -0.2 1.2]);
title('PWM Modulated Signal');
xlabel('Time (s)');
ylabel('Amplitude');

% Demodulation
x1_recover = demod(y1, fc, fs, 'pwm');
subplot(3,1,3);
```

```
plot(x1_recover);
axis([0 50 0 1]);
title('Recovered Signal (PWM Demodulated)');
xlabel('Time (s)');
ylabel('Amplitude');
sgtitle('Sujal Singh (04119051723)')
```

Output:



Experiment–9

Aim:

To study Pulse Position Modulation and Demodulation technique using MATLAB software.

Description:

MATLAB software with essential configurations.

Theory:

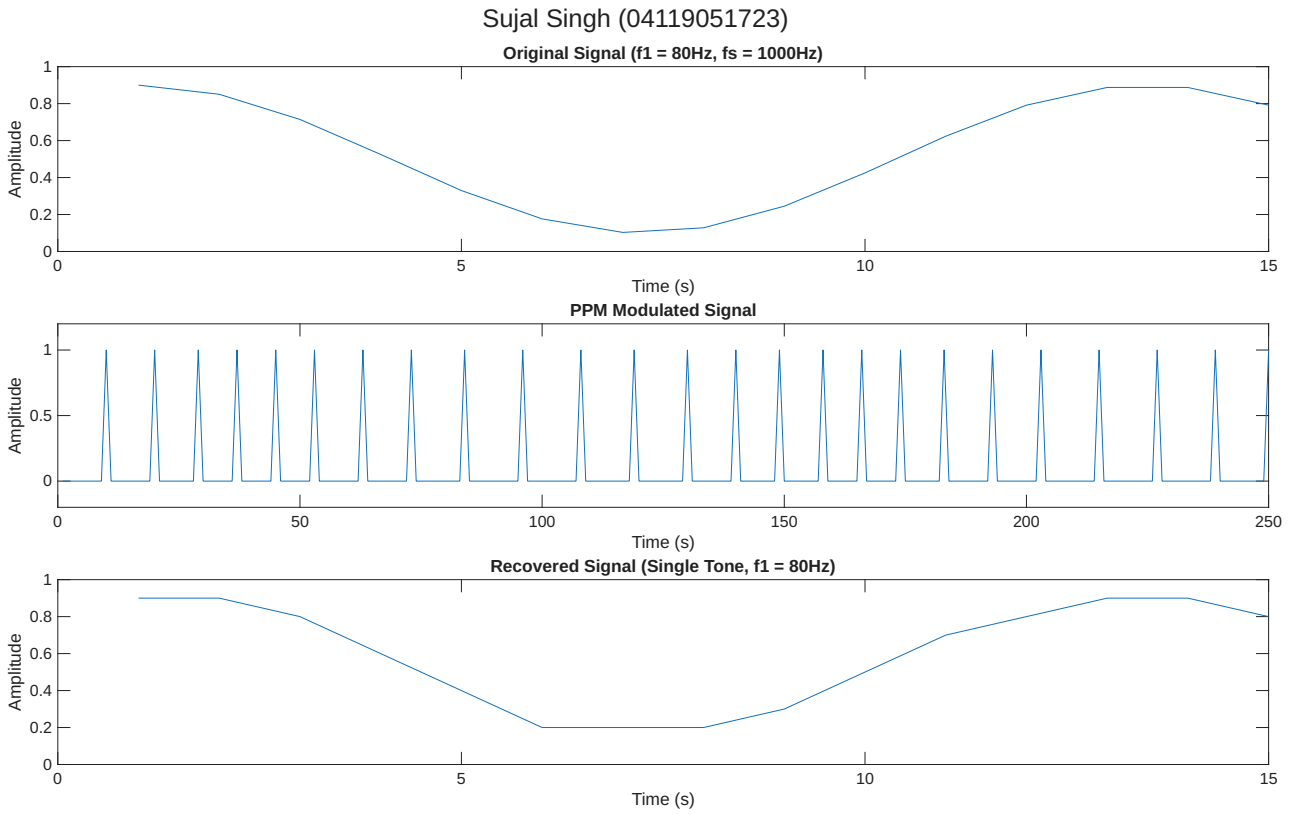
Pulse Position Modulation (PPM) is a pulse modulation technique in which the position of each pulse within a time slot is varied in accordance with the instantaneous amplitude of the message signal, while the pulse width and amplitude remain constant. The information is conveyed by the time shift of the pulse from its reference position. PPM offers better noise immunity compared to other pulse modulation methods but requires accurate synchronization at the receiver.

Code:

```
% Pulse Position Modulation (PPM)
close all; clear; clc;

fc = 100;          % Carrier frequency
fs = 1000;         % Sampling frequency
f1 = 80;           % Message frequency

t = 0:1/fs:((2/f1)-(1/fs));
x1 = 0.4 * cos(2*pi*f1*t) + 0.5;

% Modulation
y1 = modulate(x1, fc, fs, 'ppm');

subplot(3,1,1);
plot(x1);
axis([0 15 0 1]);
title('Original Signal (f1 = 80Hz, fs = 1000Hz)');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(3,1,2);
plot(y1);
axis([0 250 -0.2 1.2]);
title('PPM Modulated Signal');
xlabel('Time (s)');
ylabel('Amplitude');

% Demodulation
x1_recover = demod(y1, fc, fs, 'ppm');
```

```
subplot(3,1,3);
plot(x1_recover);
axis([0 15 0 1]);
title('Recovered Signal (Single Tone, f1 = 80Hz)');
xlabel('Time (s)');
ylabel('Amplitude');
sgtitle('Sujal Singh (04119051723)')
```

Output:



Experiment–10

Aim:

To generate modulate and demodulate amplitude shift key (ASK signal) using MATLAB.

Description:

MATLAB software with essential configurations.

Theory:

Pulse Amplitude Modulation (PAM) is a digital modulation technique where the amplitude of a series of pulses is varied in accordance with the amplitude of an analog signal being transmitted. In PAM, each sample of the analog signal is represented by a corresponding pulse amplitude. The process involves sampling the analog signal at regular intervals and quantizing each sample to a discrete amplitude level. These quantized samples then modulate a train of pulses, where the amplitude of each pulse corresponds to the amplitude of the quantized sample.

Code:

```
%to generate modulate and demodulate amplitude shift key(ASK signal) using
%matlab essential software with communication tool box
clc;
clear all;
close all;
Tb = 1;
fc = 10;
t=0:Tb/100:1;
c=sqrt(2/Tb)*sin(2*pi*fc*t);
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;
    ask_sig(i,:)=c.*m_s;
    t1 = t1+(Tb + 0.1);
    t2 = t2 + (Tb+0.1);
    subplot(5,1,2);
    axis([0 N -2 2]);
    plot(t,message(i,:),'r');
```

```
    hold on
    subplot(5,1,4);
    plot(t,ask_sig(i,:));
    hold on
end
hold off
subplot(5,1,3);
plot(t,c);
sgtitle('Sujal Singh (04119051723)')
```

Output: