EXPT. NO. 6 NAME: Page No.: 1 Date: 23 Sept

EXPERIMENT - 6

[U19CS012]

ASK, FSK and PSK

AIM : To study Amplitude shift keying (A.S.K.), Frequency shift keying (F.S.K.) and phase shift keying (P.S.K) modulation Technique and verify waveforms.

APPARATUS : MATLAB

THEORY: 1.> Modulation : Modulation is a process, by which some characteristics of a carrier wave is varied in accordance with a modulating (message) signal.

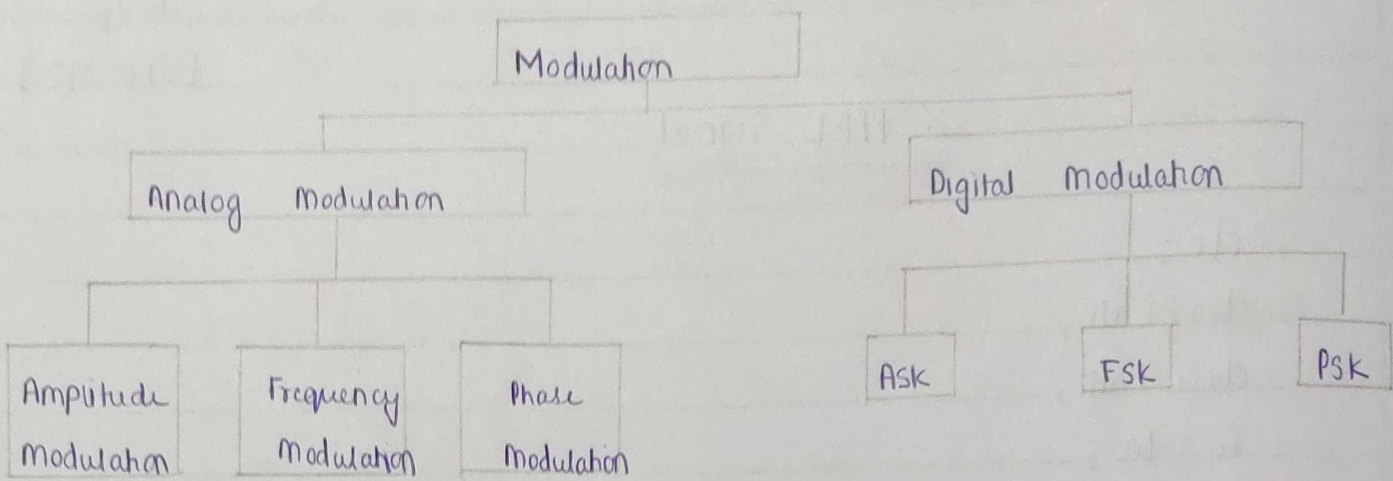
Digital modulation: It is a special kind of modulation, where the message signal is digital in nature and the carrier wave is analog (sinusodial) in nature.

The ASK, FSK and PSK are analogous to AM, FM and PM respectively. The difference is that in digital modulation techniques (ASK, FSK and PSK) the modulation signal is digital in nature. while in AM, FM and PM modulating signal is analog in nature.

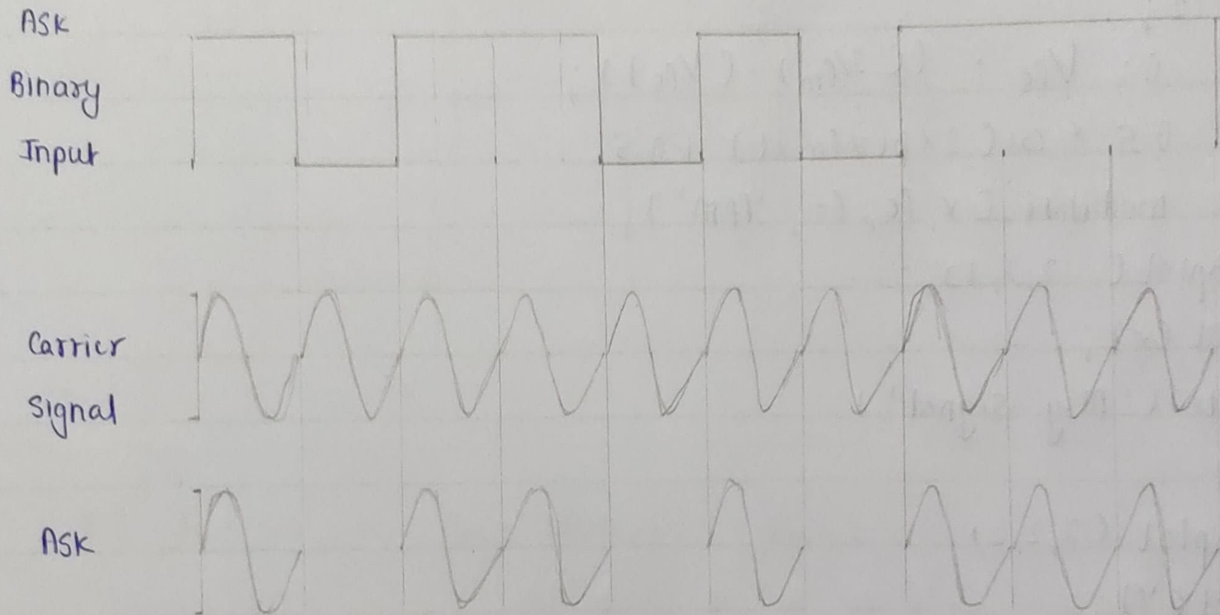
2.> ASK (Amplitude Shift keying)

In ASK, the amplitude of the carrier wave is changed (switched) according to the digital input signal (modulating signal)

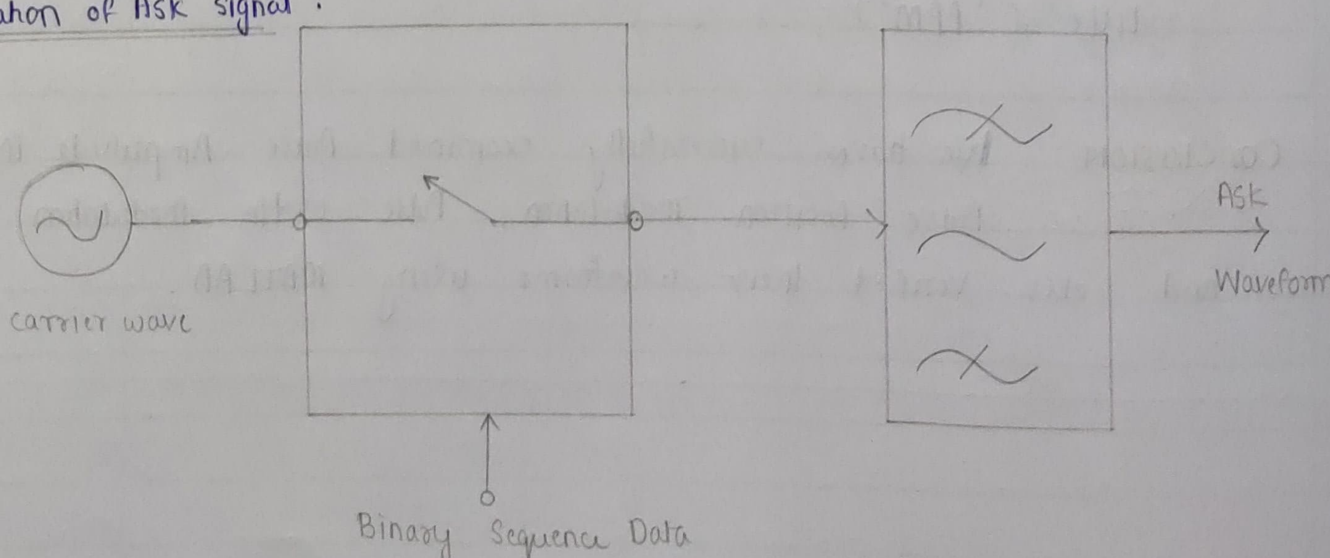
Teacher's Signature:



ASK



Generation of ASK signal :



Application of ASK : 1.) Wireless Base station
2.) Low Frequency RF Application
3.) ~~Instru~~ Industrial Network Devices

3.> FSK (Frequency Shift keying)

→ If the frequency of sinosoidal carrier wave is varied (switched) depending on the digital input signal, then it is known as the frequency shift keying.

→ Application of FSK :-

1. High Frequency Radio Transmission

4.> PSK (Phase Shift keying)

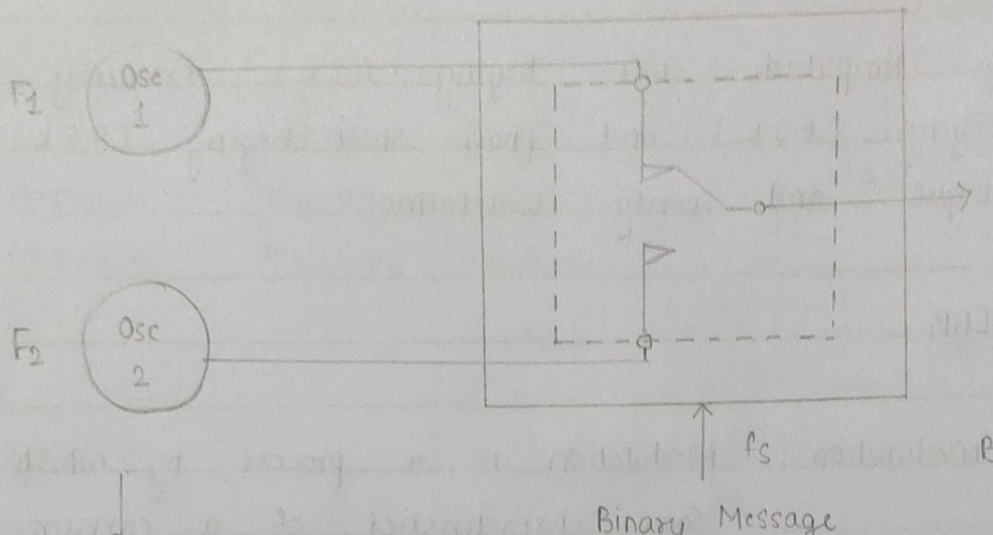
In PSK, phase of the carrier wave (analog in nature) is switched as per the input digital signal.

Application of PSK :

1. It is widely used for wireless LANs, RFID and Bluetooth communication.

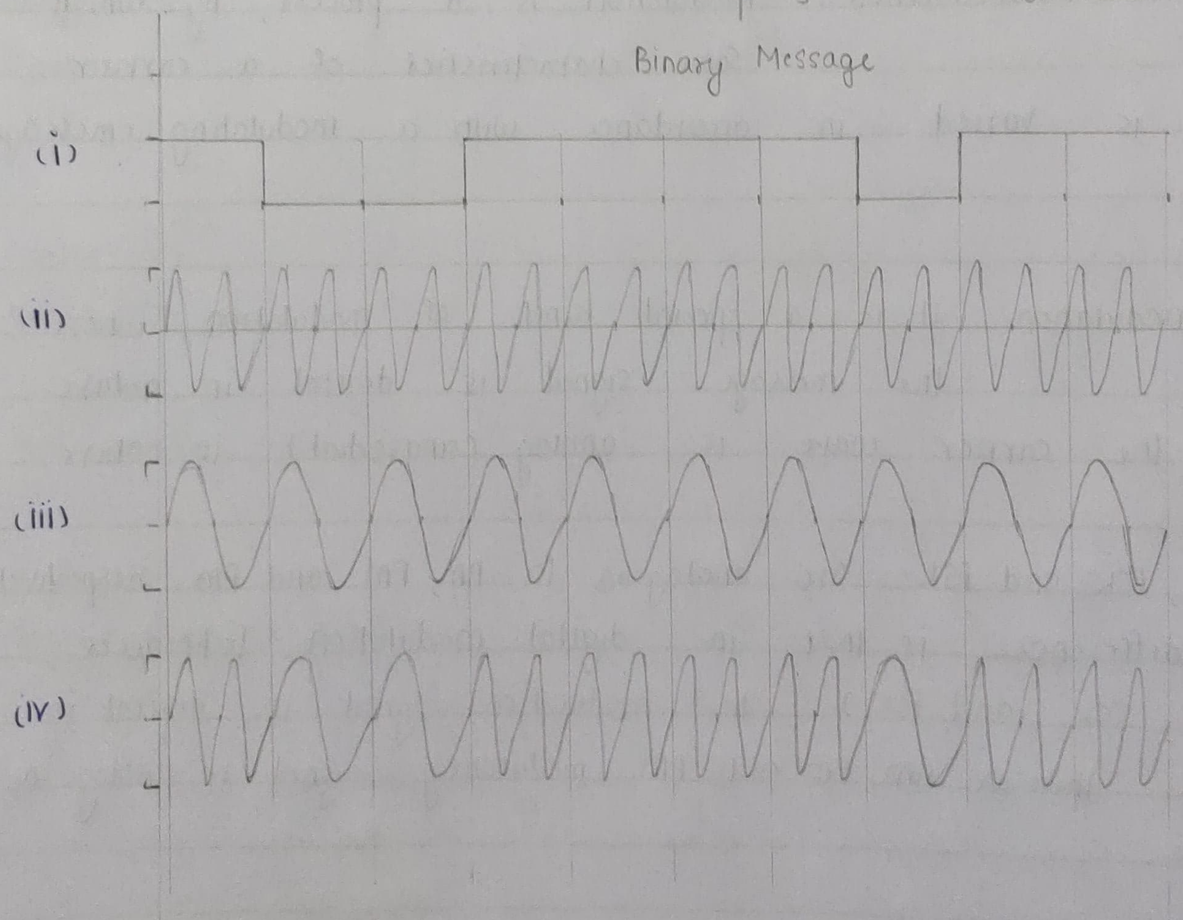
FSK

Generation of FSK



$F_1 \cong F_2$

Bit rate $F_s \ll F_1$



(i) Digital Bitstream

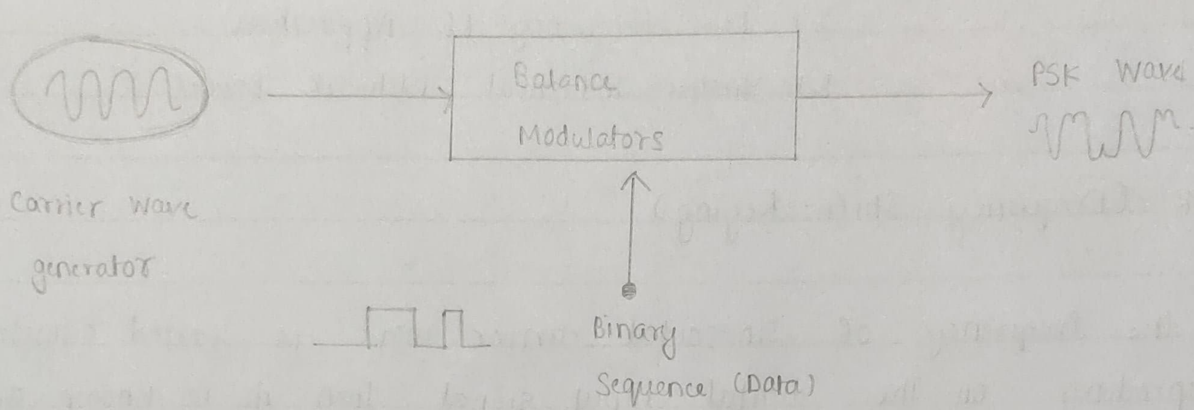
(ii) High frequency carrier wave

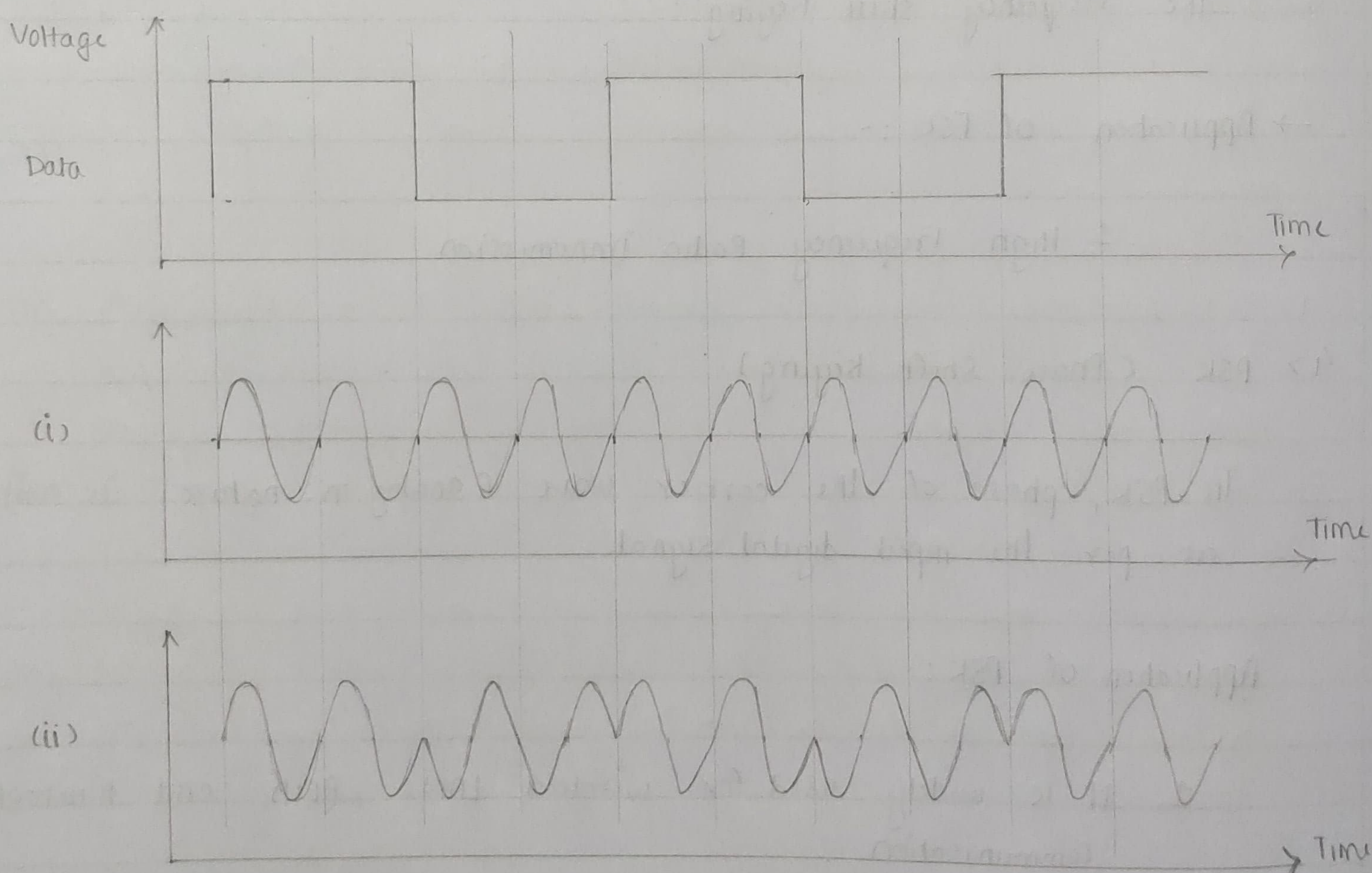
(iii) Low frequency carrier wave

(IV) FSK modulated wave

PSK

Generation of PSK





(i) Carrier Frequency Before modulation

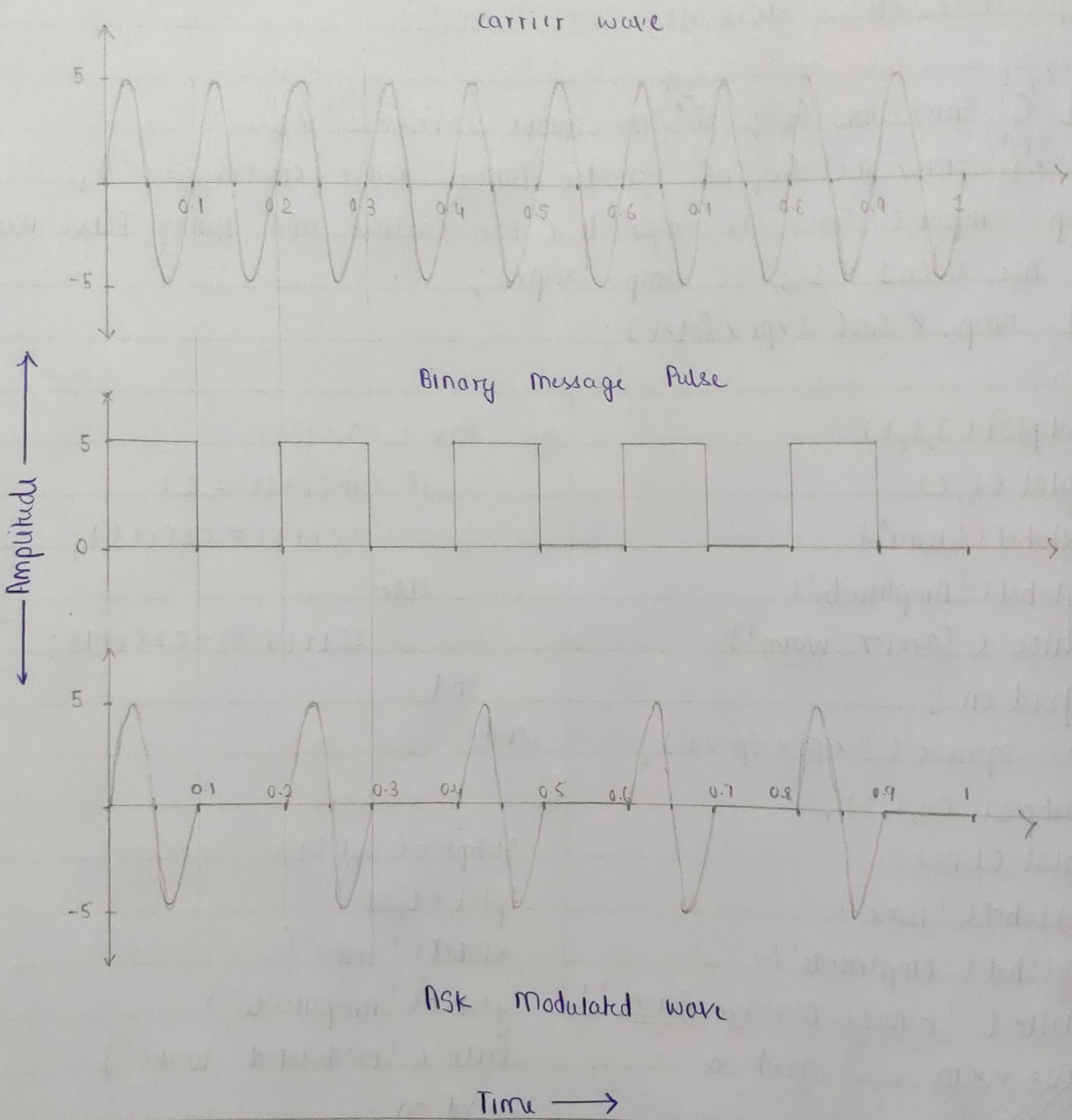
(ii) Carrier Frequency After modulation

MATLAB CODE : %% ASK

```
clc; close all; clear all; % for deleting all variables from memory
fc = input ('Enter the freq of Sine wave carrier:');
fp = input ('Enter the freq of Periodic Binary pulse (message):');
amp = input('Enter the amplitude ( For Carrier & Binary Pulse message):');

t = 0: 0.001 : 1;      % for setting the sampling interval
c = amp. * sin ( 2*pi* fc *t);      % for Generating carrier sine wave

subplot (3,1,1)     % For Plotting Carrier wave
plot (t,c)
xlabel ('Time')
ylabel ('Amplitude')
title ('carrier wave')
m = amp/2 . * square (2*pi* fp *t) + (amp/2);   % Square wave msg
subplot ( 3,1,2)     % Plotting Square Binary Pulse
plot ( t,m)
xlabel ('Time')
ylabel ('Amplitude')
title ('Binary Message Pulses')
w = c*m       % The Shift Keyed wave
subplot (3,1,3)     % for plotting ASK wave
plot ( t,w)
xlabel ('Time')
ylabel ( 'Amplitude')
title ('Amplitude Shift keyed Signal')
```

Ⓐ ASK :

3.) $F_c = 10$ Hz $F_p = 5$ Hz Amp = 5



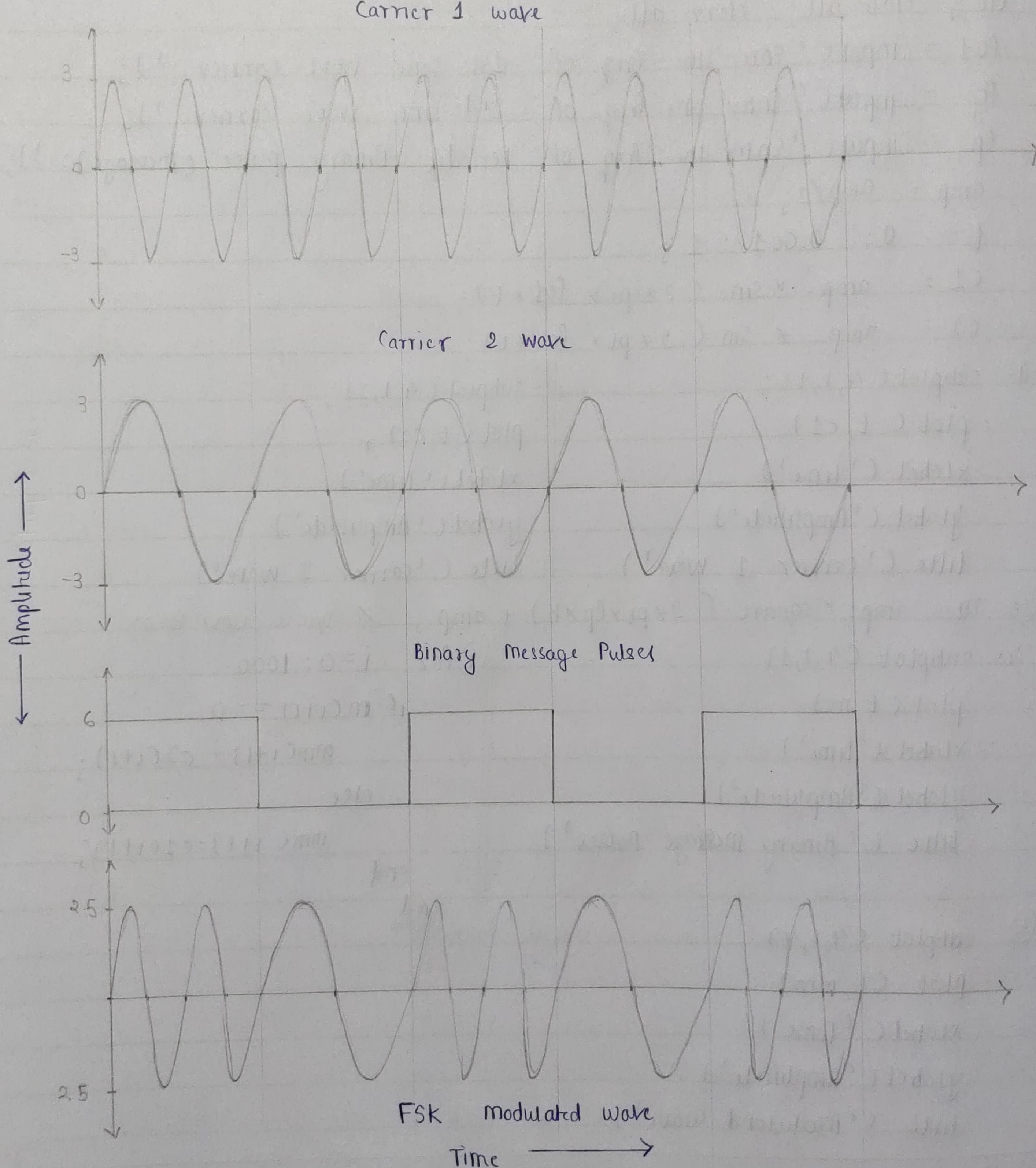
%% FSK

```
clc; close all; clear all;
fc1 = input('Enter the freq of 1st sine wave carrier:');
fc2 = input('Enter the freq of 2nd sine wave carrier:');
fp = input('Enter the freq of Periodic Binary pulse (message):');
amp = amp/2;
t= 0: 0.001 : 1;
c1 = amp.*sin (2*pi*fc1*t);
c2 = amp.*sin (2*pi*fc2*t);
① subplot(4,1,1);
plot ( t,c1)
xlabel ('Time')
ylabel ('Amplitude')
title ('carrier 1 wave')
② subplot(4,1,2);
plot(t,c2);
xlabel ('Time')
ylabel ('Amplitude')
title ('carrier 2 wave')
③ m= amp.*square (2*pi*fp*t) + amp;   % square wave form
subplot (4,1,3)
plot(t,m)
xlabel ('Time')
ylabel ('Amplitude')
title ('Binary message Pulses')
④ for i=0: 1000
    if m(i+1) ==0
        mm(i+1)= c2(i+1);
    else
        mm(i+1)=c1(i+1);
    end
end
⑤ subplot (4,1,4)
plot (t,mm)
xlabel ('Time')
ylabel ('Amplitude')
title ('Modulated wave')
```

(B)

FSK :

2.) $F_{c1} = 30$ $F_{c2} = 10$ $F_p = 5$ $amp = 6$



%% PSK

```
clc; close all; clear all;

fc1 = input('Enter the freq of 1st sine wave carrier :');
fp = input('Enter the freq of Periodic Binary Pulse (message) :');
amp = input('Enter the amplitude (For Carrier and Binary Pulse Message)');
t = 0: 0.001 : 1;        amp = amp/2;
c1 = amp. * sin(2*pi*fc1*t);
```

```
① subplot(3,1,1)
plot(t,c)
xlabel('Time')
ylabel('Amplitude')
title('Carrier wave')
grid on;
m = square(2*pi*fp*t);
② subplot(3,1,2),
plot(t,m)
xlabel('Time')
ylabel('Amplitude')
title('Binary Message Pulses')
w = c*m        grid on
subplot(3,1,3)
```

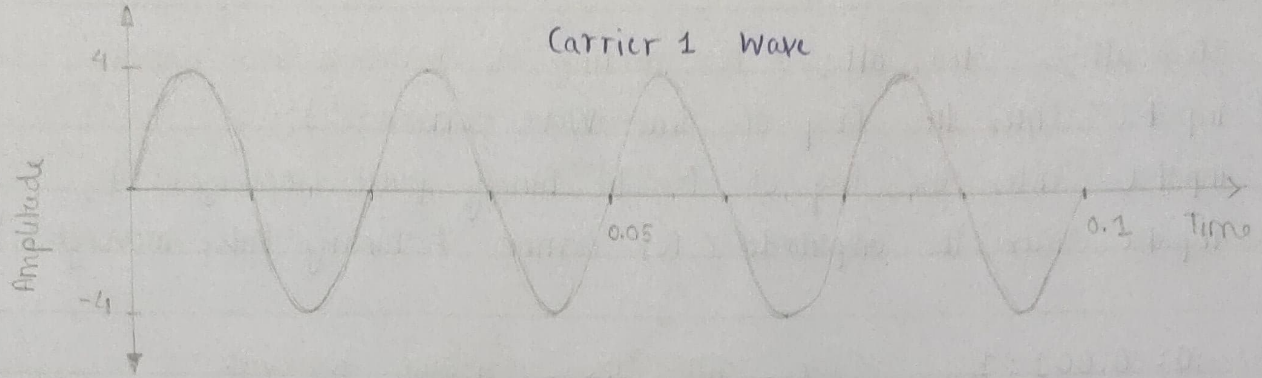
```
③ for i=0:1000
    if (m(i+1) == 1)
        s(i+1) = c1(i+1);
    else
        s(i+1) = -c1(i+1);
    end
end
④
subplot(3,1,3)
plot(t,s)
xlabel('Time')
ylabel('Amplitude')
title('modulated wave')
grid on.
```

Waveforms using MATLAB

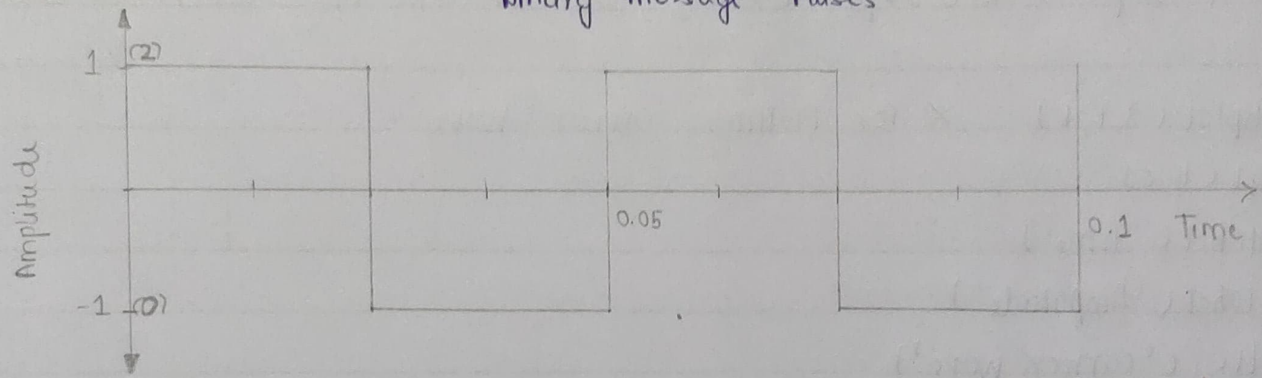
PSK :

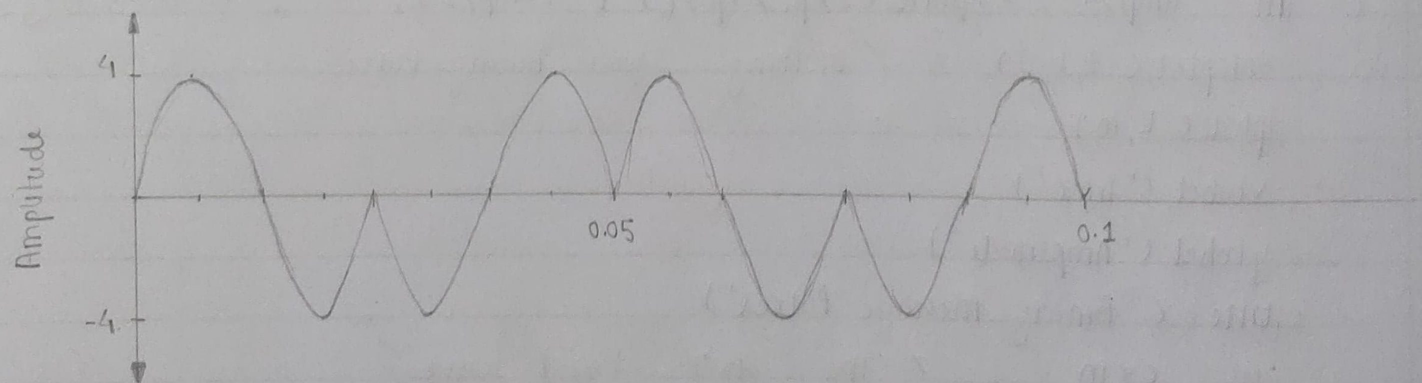
1.) $F_{c1} = 40\,Hz$ $F_p = 20\,Hz$ $Amp = 8\,V$

Carrier 1 Wave



Binary message Pulses





Modulated Wave

EXPT. NO. | NAME: | Page No.: 12 | Date: 23 Sept

[U19CS012]

CONCLUSION :

1) We have successfully studied ASK, PSK and FSK modulation technique and verified their waveforms using MATLAB. We also observed the schematic diagrams for ASK, FSK & PSK.

Teacher's Signature:

EXPERIMENT 7 : [U19CS012]

EFFECT OF AWGN on AM and FM

AWGN : Additive white Gaussian noise

AIM: To study the transmission of Amplitude modulated (AM) and Frequency modulated (FM) signal under the Additive Gaussian Noise channel.

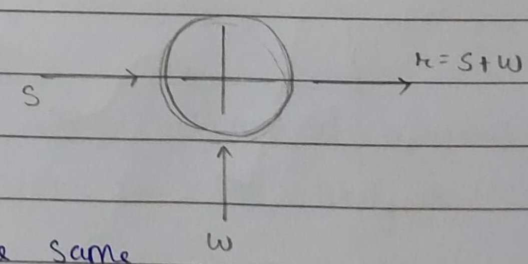
APPARATUS:- MATLAB

THEORY: 1.> Additive White Gaussian Noise (AWGN)

A Basic Noise model used to mimic the effect of many random processes that occur in nature. Channel produces Additive white Gaussian Noise. (AWGN)

(a) Additive - The received signal equals the transmit signal plus some noise, where the noise is statistically independent of the signal.

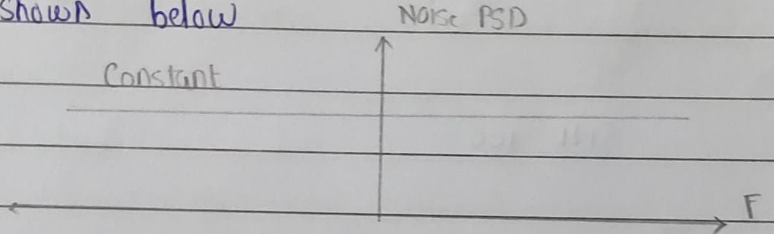
$$r(t) = s(t) + w(t)$$



(b) White : It refers that the noise has the same power distribution at every frequency OR it has uniform power across the frequency band for the information system.

- It is an analogy to the color white which has uniform emissions at all frequencies in the visible spectrum. If we focussed a beam of light for each color on the visible spectrum onto a single spot, that combination would result in a beam of white light.

- As a consequence, the Power Spectral Density (PSD) of white noise is constant for all frequencies ranging from $-\infty$ to $+\infty$ as shown below



ⓒ Gaussian - Gaussian distribution, or a normal distribution, has an average of zero in the time domain, and is represented as a bell shaped curve.

- The probability distribution of the noise samples is Gaussian with a zero mean.

- The values close to zero have a higher chance of occurance while the values far away from zero are less likely to appear.

- In reality, the ideal flat spectrum from $-\infty$ to $+\infty$ is true for frequencies of interest in wireless communication (few KHz - Hundred GHz) but not for higher frequencies.

2.) Signal to Noise Ratio

- The SNR or S/N is a measure used in science and engineering that compares the level of a desired signal to the level of background noise.

- It is defined as the ratio of signal power to the noise power, often expressed in decibels. A ratio higher than 1:1 (greater than 0 dB) indicates more signal to that noise.

- SNR, bandwidth and channel capacity of a communication channel are connected by the Shannon-Hartley theorem

$$SNR_{dB} = 10 \log_{10}\left(\frac{P_{signal}}{P_{noise}}\right)$$

Shannon-Hartley Theorem

It states that the channel capacity (bits per second) OR information rate of data that can be communicated at low error data using an average received signal power through communication channel subject to additive white Gaussian noise (AWGN) of power.

$$C = B \log_2\left(1 + \frac{S}{N}\right), \quad B = \text{Bandwidth of channel in Hz}$$

- It is related to signal to noise (SNR) or carrier to noise (CNR) [Linear Power ratio]

- 5 dB - 10 dB ⇒ It is below the minimum level to establish a connection due to the noise level being nearly indistguishable from the desired signal (useful information)

- 25 dB - 40 dB = deemed to be good.

- 41 dB or higher = considered to be Excellent

3.> Mathematics of AM

- Let modulating signal be $e_m = E_m \sin(\omega_m t)$
  carrier signal be $e_c = E_c \sin(\omega_c t)$

$$\therefore E_{AM} = E_c + e_m$$
$$= E_c + E_m \sin(\omega_m t)$$

The instantaneous value of the amplitude modulated wave can be given as

$$e_{AM} = E_{AM} \sin(\theta)$$
$$= E_{AM} \sin(\omega_c t)$$
$$e_{AM} = (E_c + E_m \sin(\omega_m t)) \sin(\omega_c t)$$

This is an equation of AM wave.

4.> Mathematics of FM

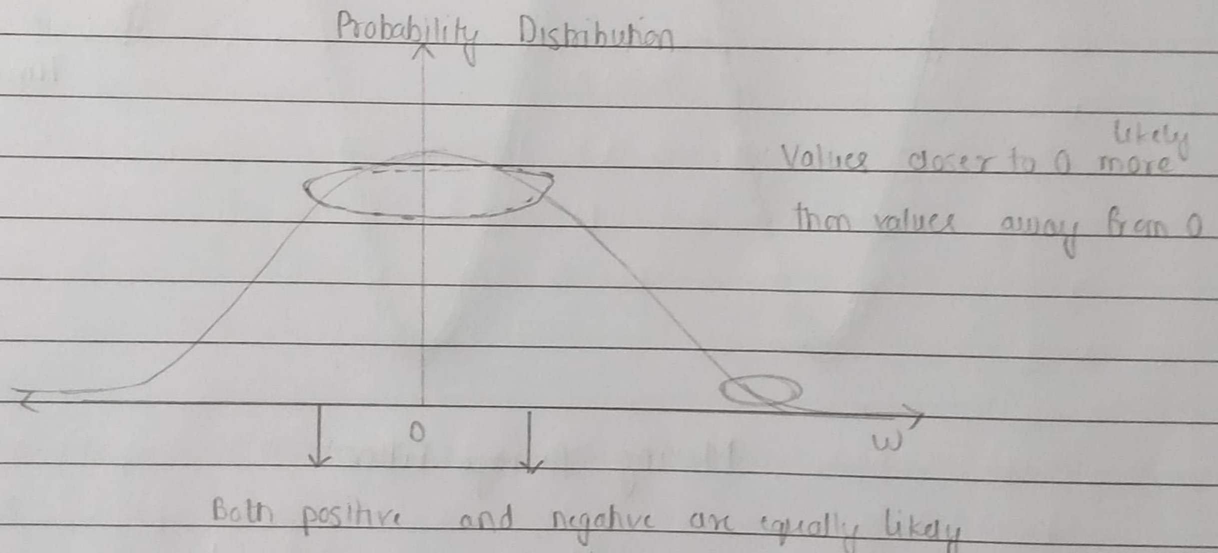
$$s(t) = E_s \sin(\omega_c t + m_s \sin(\omega_m t))$$

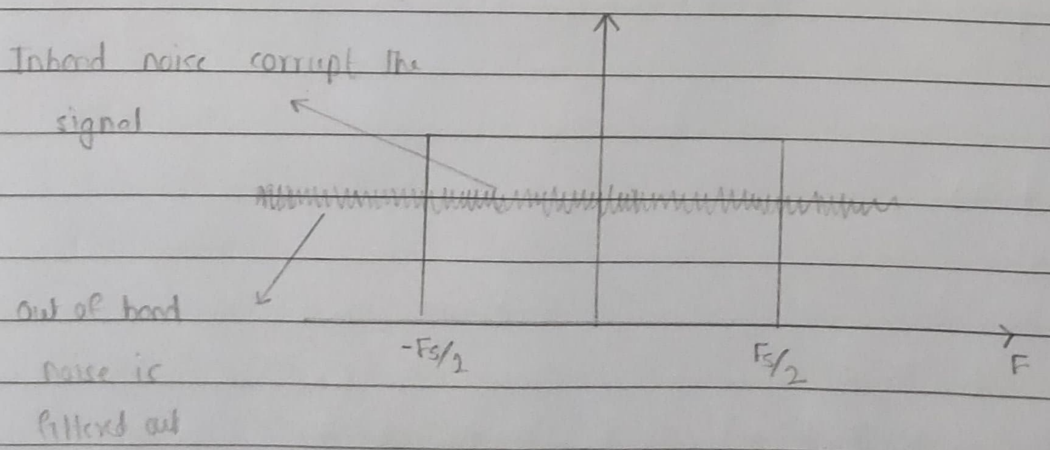
This is the expression for FM wave, m_f = modulation index

$$m_F = \frac{\text{Frequency deviation}}{\text{Modulating Frequency}} = \frac{\Delta f}{f_m}$$

- Frequency deviation Δf = represents the maximum departure of the instantaneous frequency $F_i(t)$ of the FM wave from the carrier frequency F_c.

⇒ Gaussion





EXPT. NO. NAME:

MATLAB CODE :

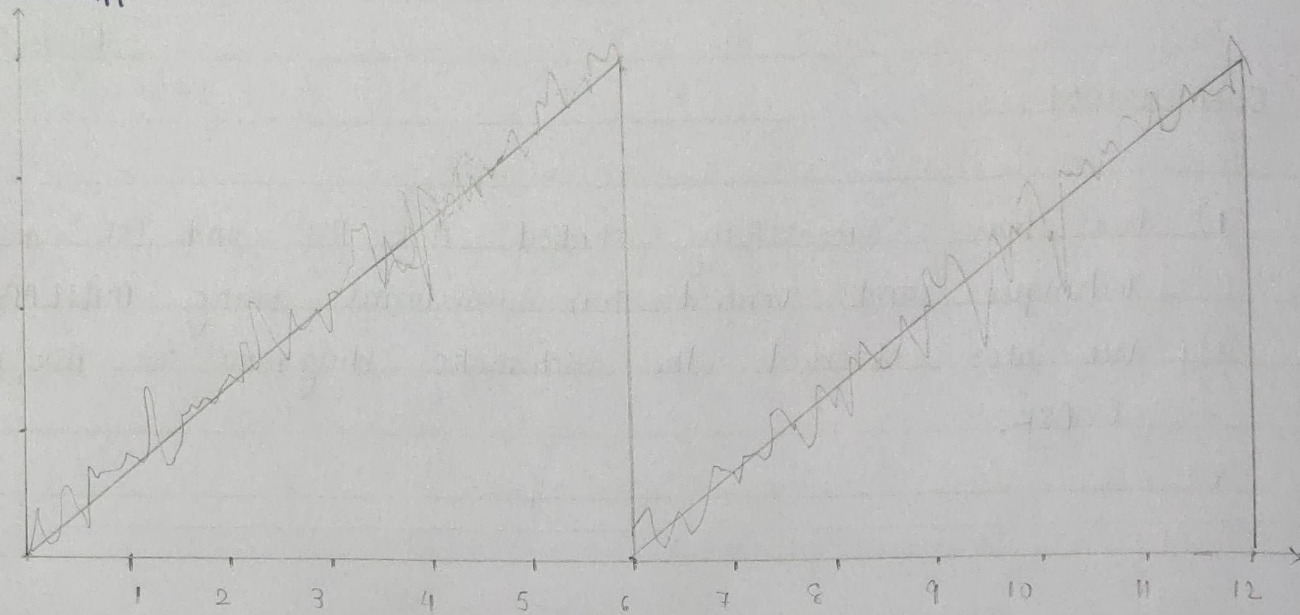
AWGN in different function

```
clc;
clear all;
t= 6:0.1 :10);
x= sawtooth (t);
y= awgn (x, 10, 'measured');
plot( t, [x,y]);
legend ('original signal', 'signal with AWGN');

clc;
clear all;
t= (0 : 0.1 : 10);
x = sin( t);
y = awgn( x, 10, 'measured' );
plot ( t, [x,y] );
legend( 'original signal', 'signal with AWGN' )

clc;
clear all;
t= (0: 0.1 : 10);
x = cos (t);
y= awgn (x, 10, 'measured');
plot ( t, [x,y]);
legend ('original signal', 'signal with AWGN');
```

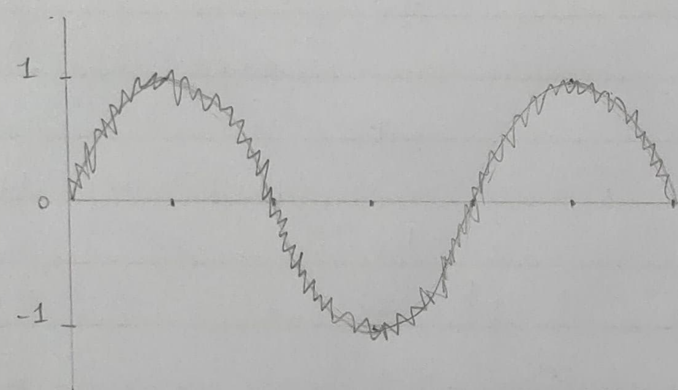
Teacher's Signature:

AWGN Effect on Different functions

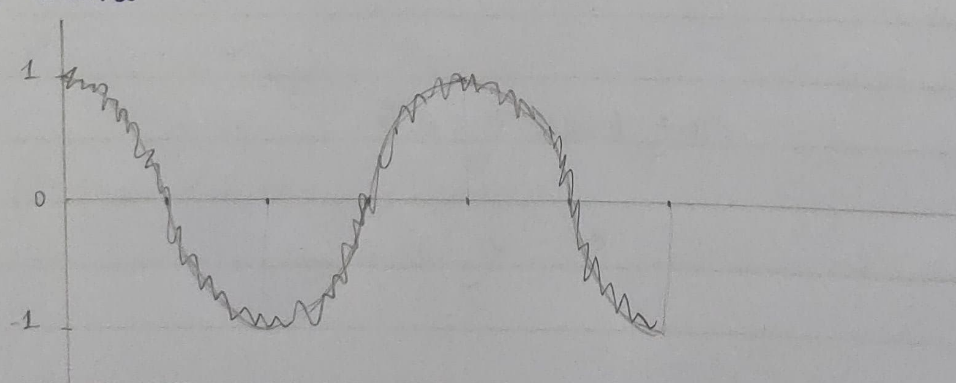
1) Sawtooth



2) Sine



3) Cosine



MATLAB CODE :

AWGN in AM

```
clc;
clear all;
t= 0:0.001:1;
vm= 5;
vc= 10;
fm= 2;
fc= 25;

m = vm * sin( 2*pi* fm *t);
c = vc * sin( 2* pi* fc*t);
amp = vc + vm * sin( 2*pi* fm * t);
am = amp. * sin (2*pi* fc *t);
y = awgn (am, 10, 'measured');
```

①

```
subplot (4,1,1);
plot (t,m);
xlabel ('time ')
ylabel (' amplitude)
```

②

```
subplot (4,1,2);
plot( t,c);
xlabel('time');
ylabel ('amplitude');
```

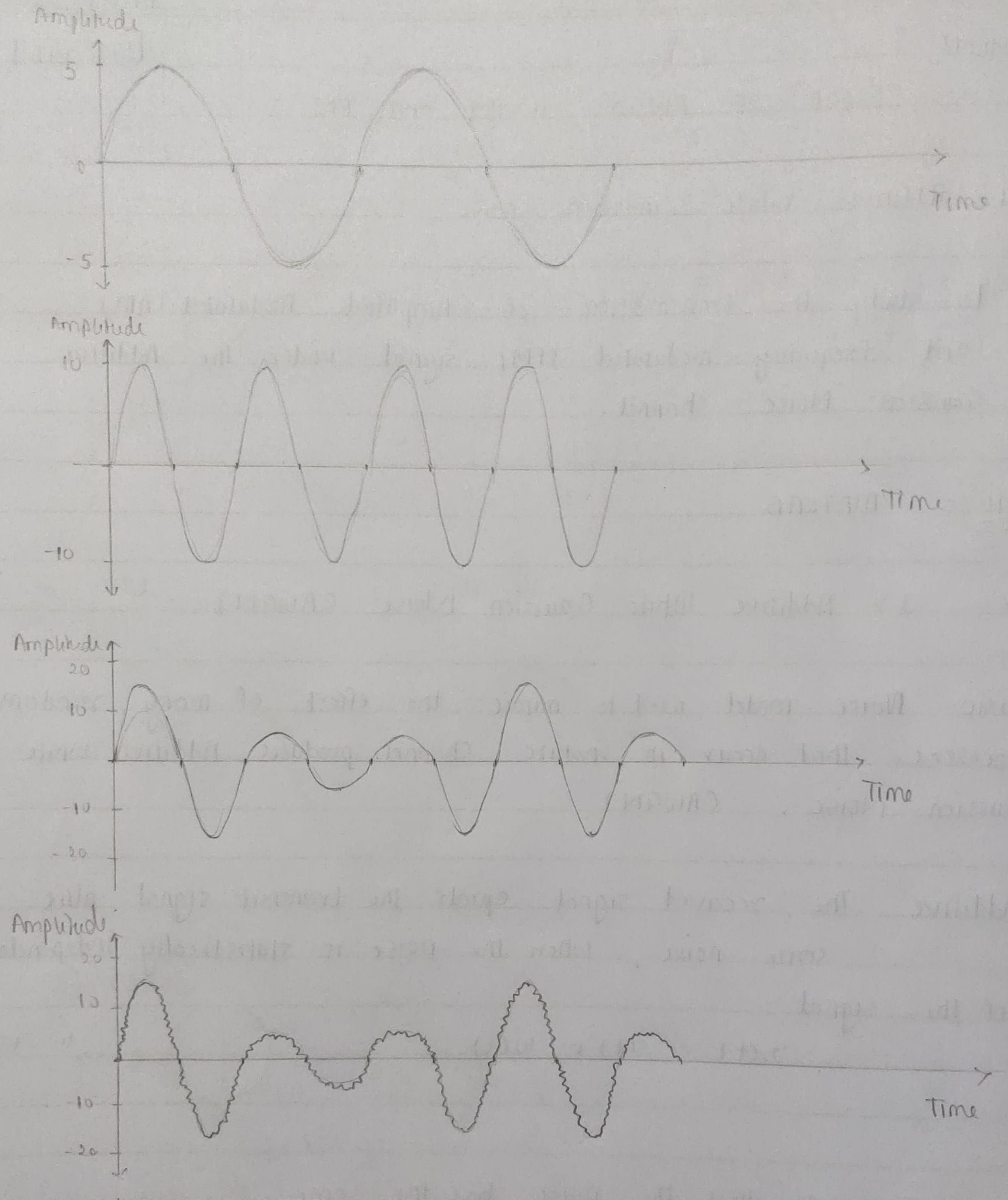
③

```
subplot ( 4,1,3);
plot (t, am);
xlabel ('time')
ylabel ( 'amplitude')
```

④

```
subplot ( 4,4,4);
plot ( t, y);
x label ('time')
y label ('amplitude')
title ( 'amplitude modulated signal with AWGN');
```

AM signal under AWGN

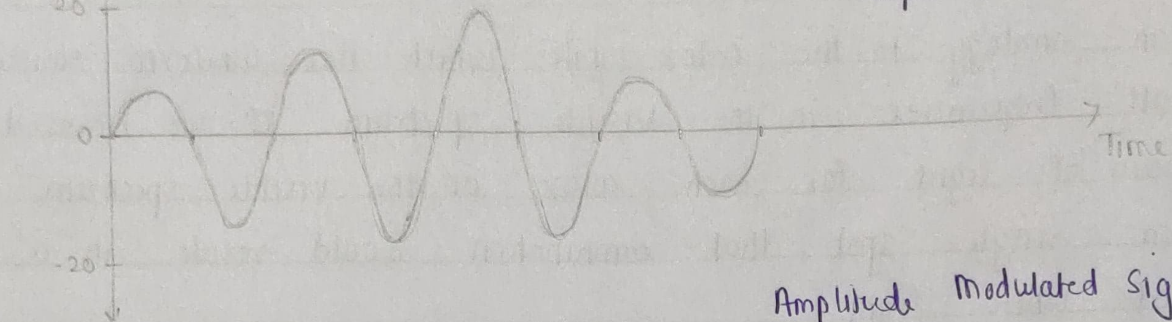


AWGN in different SNR

```
clc; clear all;
t = 0 : 0.001 : 1;
vm = 5;      fm = 2;
vc = 10;     fc = 25;
m = vm * sin( 2*pi* fm*t);
c = vc * sin( 2*pi* fc*t);
amp = vc + vm* sin( 2*pi* fm*t);
am = amp. * sin( 2*pi* fc*t);
y1 = awgn ( am, 10, 'measured');
y2 = awgn ( am, 100, 'measured');
y3 = awgn ( am, 1000, 'measured');

(1)
subplot (4,1,1);
plot (t, am)
xlabel ('time');
ylabel ('amplitude');
title ('amplitude modulated signal');

(2)
subplot(4,1,2);
plot( t, y1);
xlabel ( 'time' )
ylabel ( 'amplitude')
title (' AM signal with AWGN [snr 10]');

(3)
subplot (4,1,3);
plot (t, y2);
xlabel ( 'time');
ylabel ('amplitude');
title ('AM signal with AWGN [snr 100]');

(4)
subplot (4,1,4);
plot (t, y3);
xlabel ('time')
ylabel ('amplitude')
title (' AM signal with AWGN
        [snr 1000] ');
```

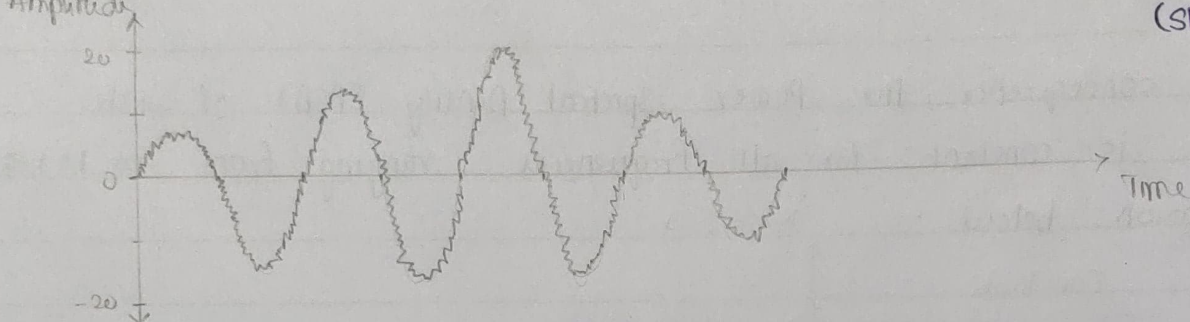
AM Signal with different SNR values

Amplitude Modulated signal

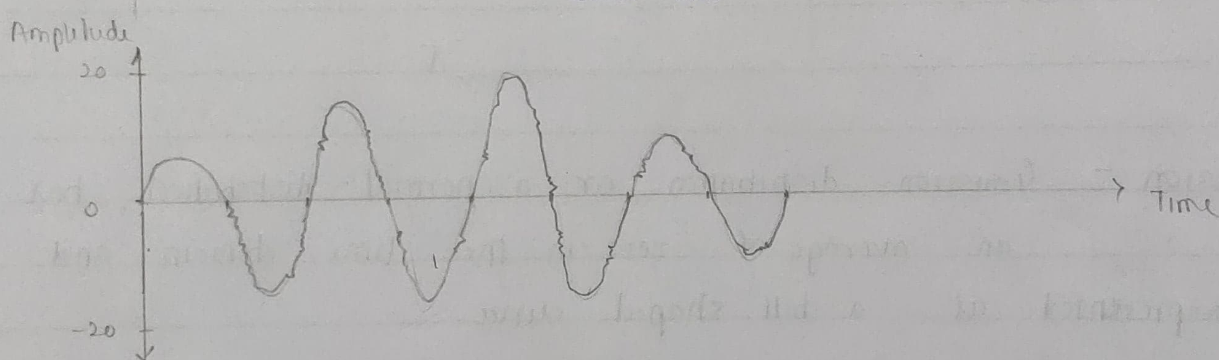


Amplitude Modulated Signal with (AWGN)

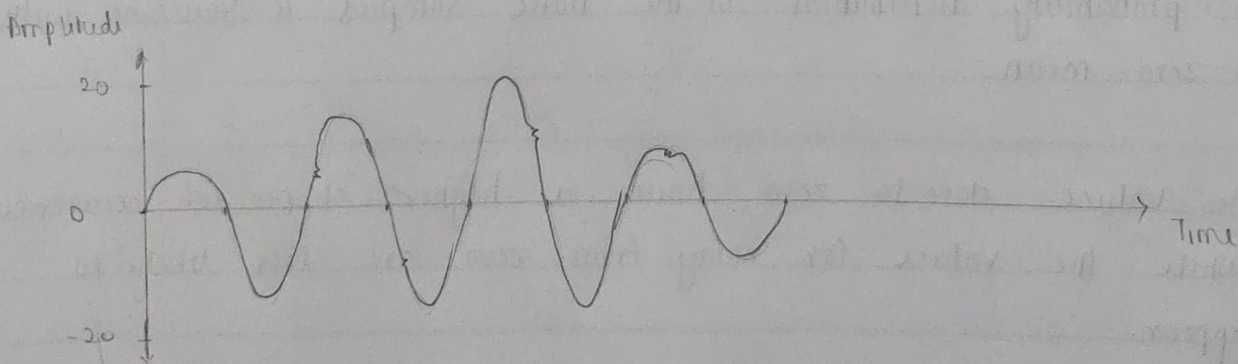
(SNR = 10)



SNR = 100



SNR = 1000



AWGN in FM

```
clc;
clear all;
t = 0: 0.001 : 1;
Vm = 5;
Vc = 5;
fm = 2;
fc = 25;
fd = 5;
msg = Vm* sin ( 2* pi* fm*t);
c = vc * sin( 2* pi*fc*t);
y = vc * sin ( 2* pi * fc*t + fd. *cos(2*pi* fm*t)) ;
z = awgn (y, 5, 'measured');
```

(1)

```
subplot( 4,1,1);
plot ( t, msg );
xlabel ('time');
ylabel ('Amplitude');
title ('message signal');
```

(2)

```
subplot (4,1,2);
Plot ( t, c );
xlabel ('time');
ylabel ('amplitude');
title ( 'carrier signal' );
```

(3)

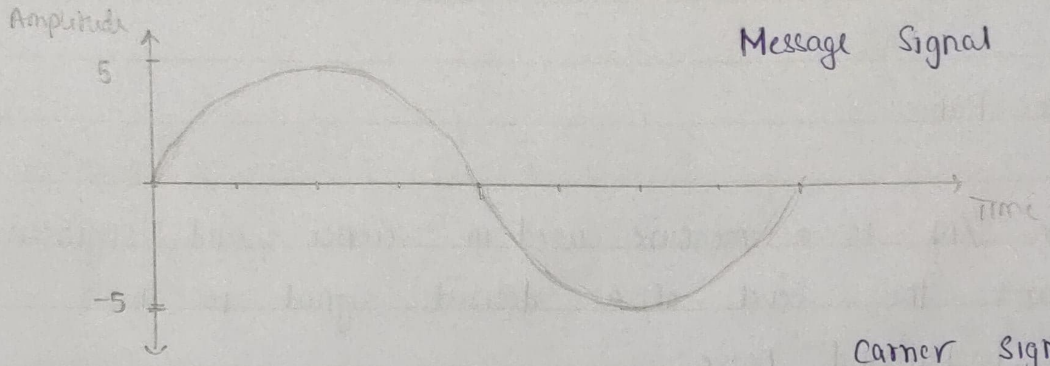
```
subplot (4,1,3);
plot (t, y) ;
xlabel ('time');
ylabel ('Amplitude');
title ('Frequency modulated signal')
```

(4)

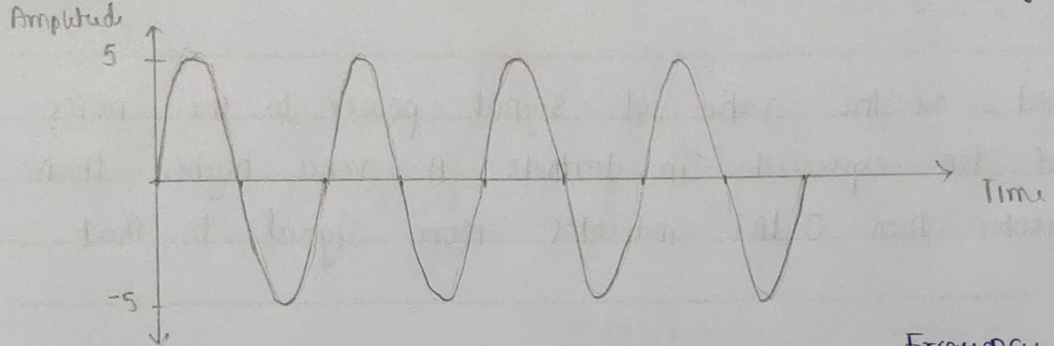
```
subplot (4,1,4);
plot ( t, z);
xlabel ( 'time' );
ylabel ( 'amplitude' );
title ( 'Frequency modulated signal with
AWGN' );
```

FM Signal under AWGN

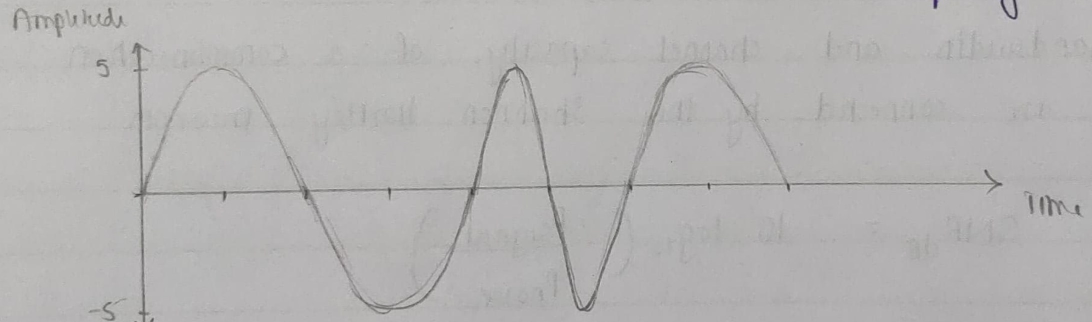
Message Signal



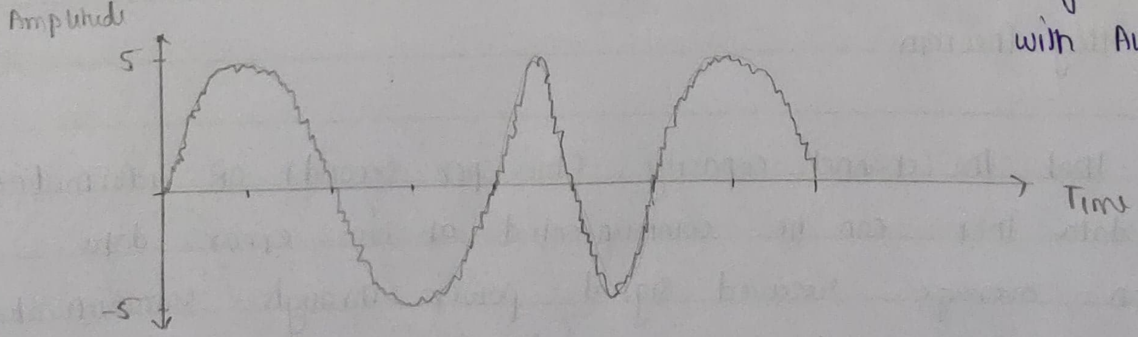
Carrier Signal



Frequency Modulated Signal



Frequency Modulated signal with AWGN



FM in Different SNR

```
clc; clear all;
t= 0: 0.001 :1;
vm = 10;        fm = 2;
vc = 5;         fc = 25;
fd = 10
m = vm * sin( 2*pi*fm * t);
c= vc* sin( 2*pi* fc*t);
amp= vc + vm*sin( 2* pi*fm*t);
y = vc*sin ( 2*pi *fc*t + fd.*cos(2*pi* fm*t) );
y1 =awgn ( y, 1, 'measured' );
y2 = awgn ( y, 10 ,'measured');
y3 = awgn ( y, 100, 'measured' );
```

①
```
subplot (4,1,1);
plot ( t, y);
xlabel ('time')
ylabel('amplitude')
title ('Frequency modulated signal')
```

②
```
subplot (4,1,2);
plot( t, y1);
xlabel ('time');
ylabel ('amplitude');
title (' snr 10');
```

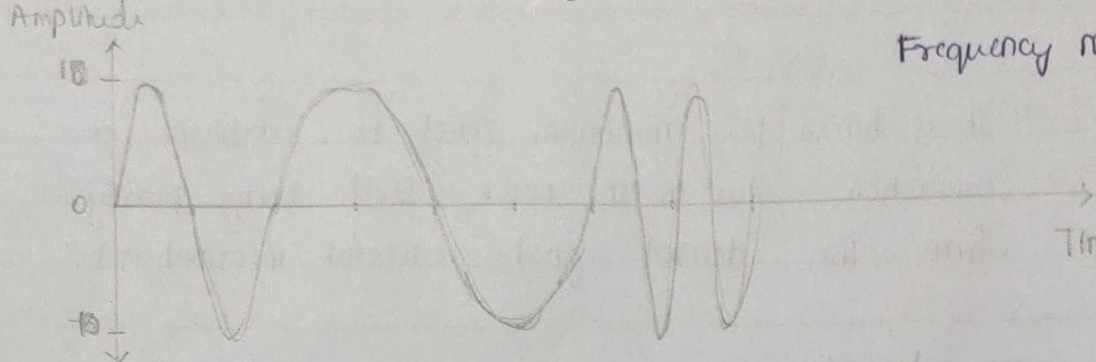
③
```
subplot ( 4,1,3);
plot ( t, y2);
xlabel (' time');
ylabel (' amplitude');
title (' AWGN [snr 100] FM');
```

④
```
subplot ( 4,1,4);
plot ( t, y3);
xlabel ('time');
ylabel ('amplitude');
title ('AWGN [snr 1000] FM');
```

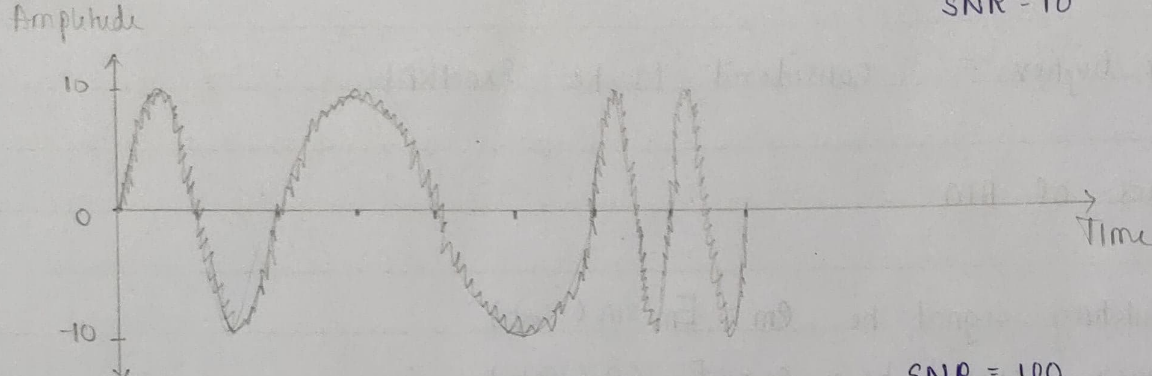
Teacher's Signature:

FM signal with different SNR

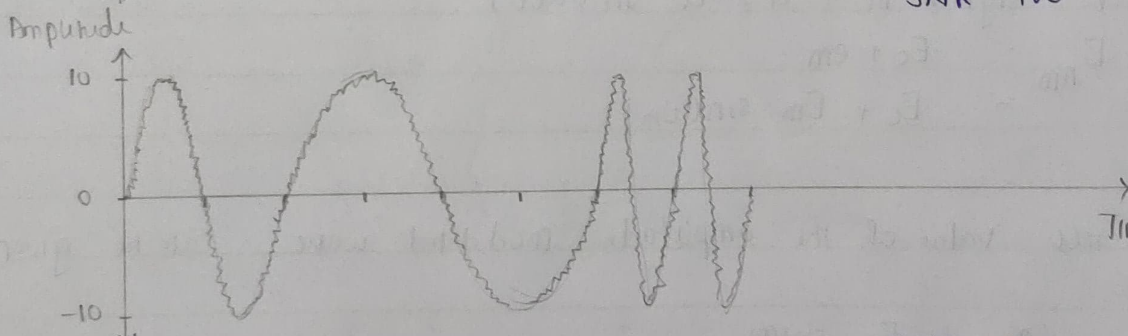
Frequency modulated Signal



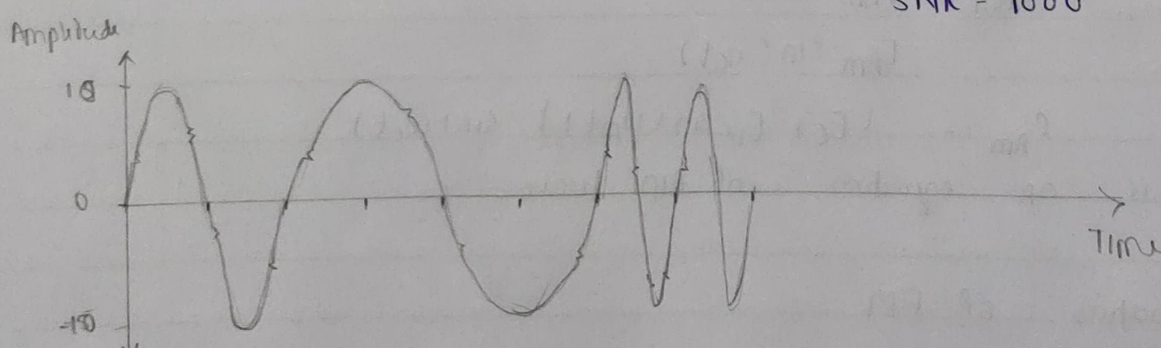
SNR = 10



SNR = 100



SNR = 1000



EXPT. NO. NAME: Page No.: 16 Date: 14 Oct

[U19CS012]

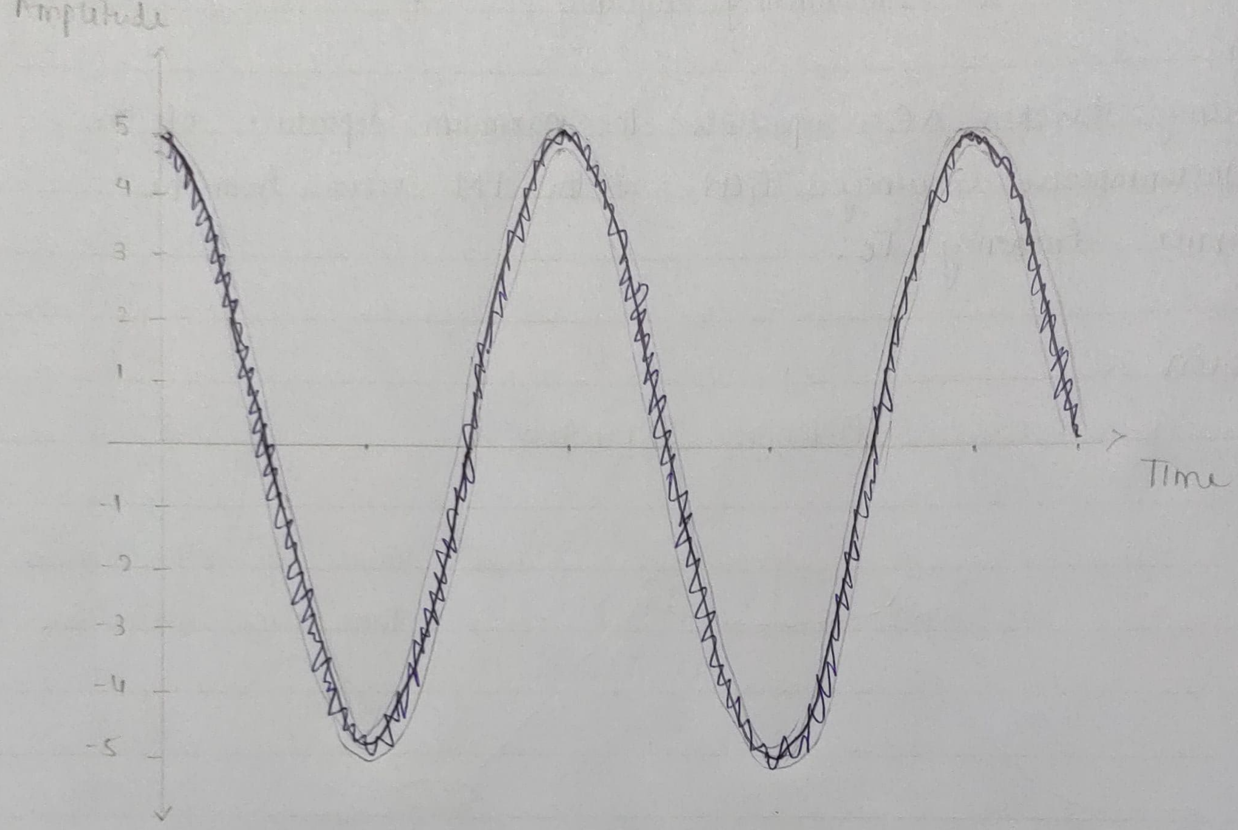
Moving Average Filter

```
clear all;    clc;
close all;
fs = 500000;
fm = 10000;
t = 1:200;
x = 5* cos( 2*pi* (fm/fs)*t );
z = awgn(x,5);
plot (x, 'g', 'linewidth', 1.5);
hold on;
plot (z); hold on;
for i: 1:194;
    y(i) = ( z(i) + z(i+1) + z(i+2) + z(i+3) + z(i+4) + z(i+6))/6;
end
plot (y, 'r', 'linewidth', 1.5);
legend( 'Actual', 'Noisy', 'Filtered');
title ( 'moving Average Filter', 'fontsize', 12);
xlabel ( '--> time in 2us');
ylabel ( '--> volts');
```

> CONCLUSION: We have successfully studied the effect of AWGN on transmission of Amplitude modulation (AM) and Frequency Modulation (FM).

Teacher's Signature:

Use of moving average filter to retrieve the signal by averaging the noise fluctuation



Moving Average Filter

— = Actual — = Noisy - Filtered

EXPERIMENT - 8 [U19CS012]

SINGLE SIDE BAND (SSB-SC) MODULATION SCHEME

AIM: Write and simulate a program for single side-band (SSB) modulation scheme. Draw the message / carrier waveforms and resultant modulated signal in time domain and frequency domain. Show input / output using matlab code in virtual mode.

APPARTUS: MATLAB

THEORY:

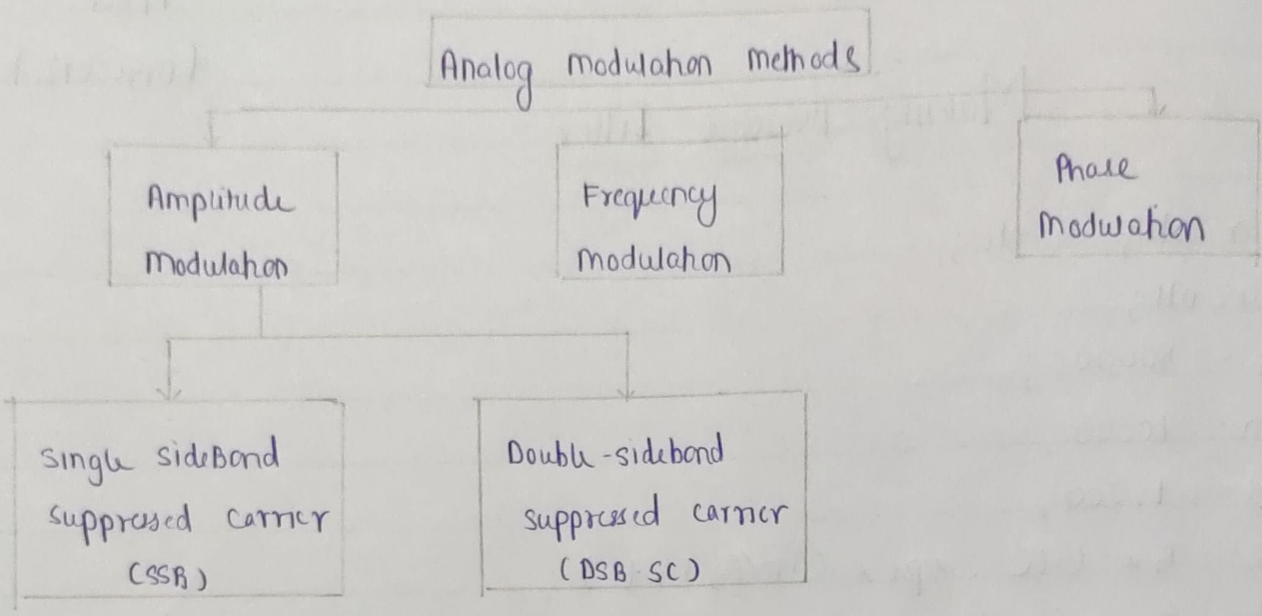
1.) Modulation: Modulation is a process by which some characteristics of a carrier wave is varied in accordance with a modulating (message) signal.

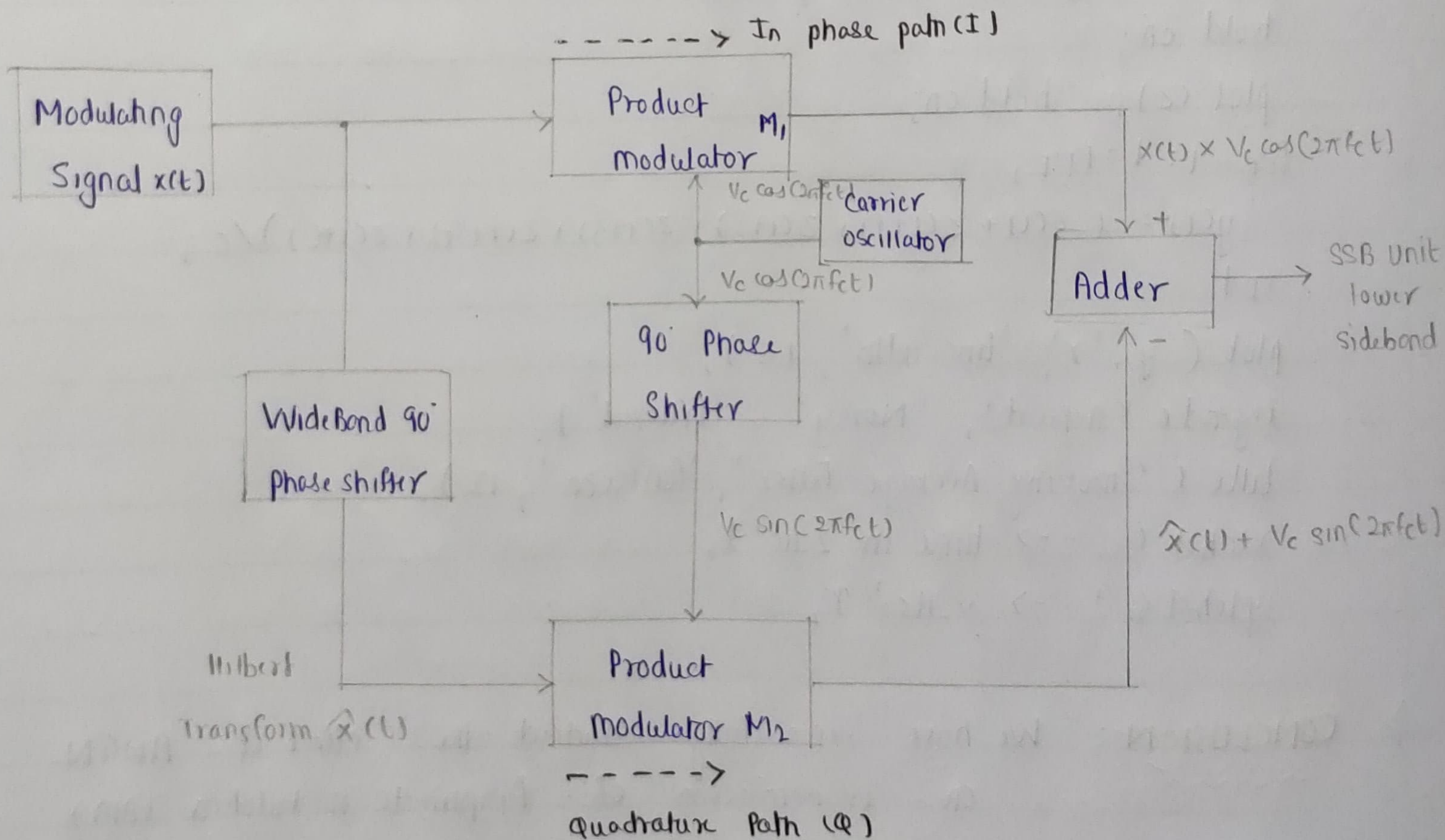
2) Analog modulation: It is a kind of modulation, where the message signal and the carrier wave both are analog in Nature.

3.) Single Side Band (SSB-SC) modulation:

- SSB-SC is a type of Amplitude Modulation
- In conventional A.M, we have two side band and the one carrier wave (no information is contained by the carrier)
- In SSB-SC modulation, only one side band is transmitted because

Teacher's Signature:





Generation of SSB-SC : By Hilbert Transform method both USB (upper side band) and LSB (lower side band) having the same information.

- Therefore, the transmission bandwidth is reduced to half and also required less power compare to other method of A.M.

4.) There are two methods of generation of SSB-SC :

1. Frequency discrimination method
2. Hilbert transform method or Phase discrimination method

5.) Applications:

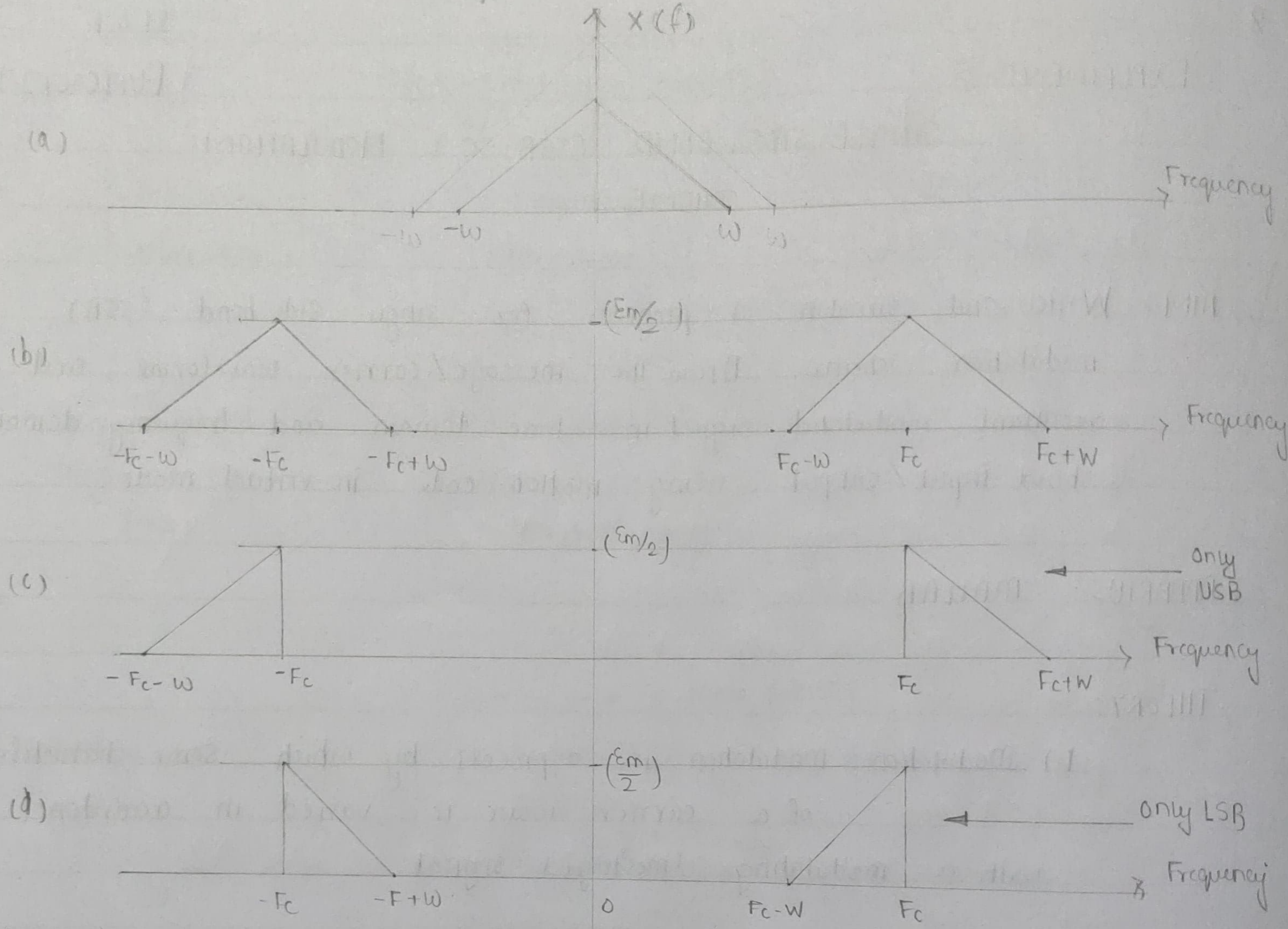
→ In point-to-point communications

→ Radar Communication

→ Where the power saving and low bandwidth requirements are important

→ In many voice applications.

Fig 4 Exp-8



(a) Spectrum of message Signal

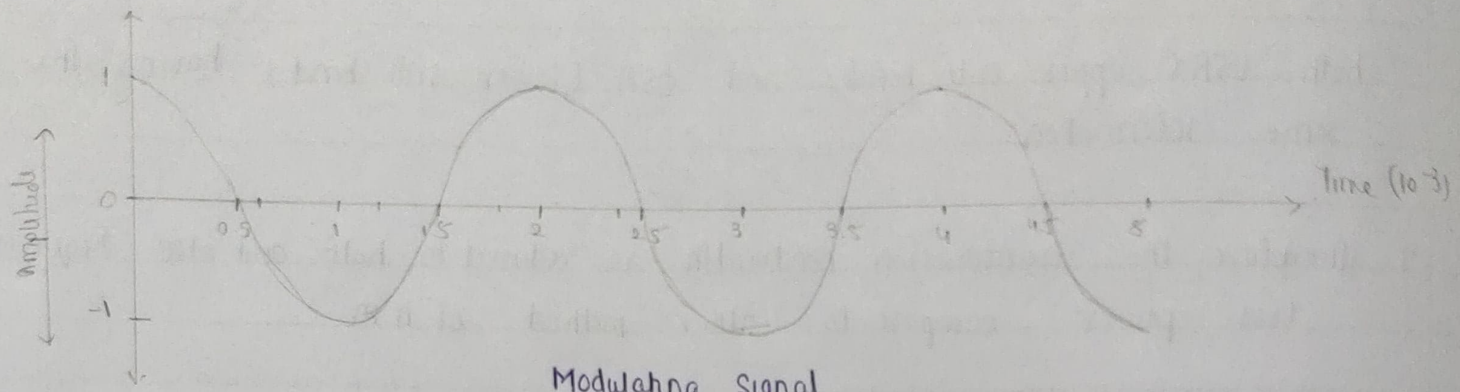
(b) Spectrum of DSB- SC wave

(c) spectrum of DSB- SC with only USB Transmission

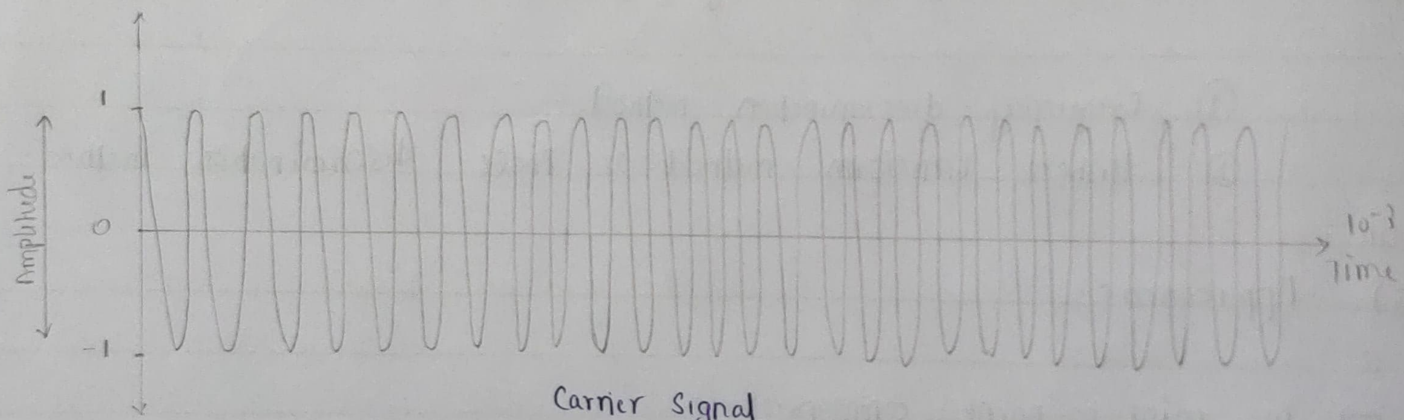
(d) Spectrum of DSB-SC with only LSB Transmission

MATLAB Code :

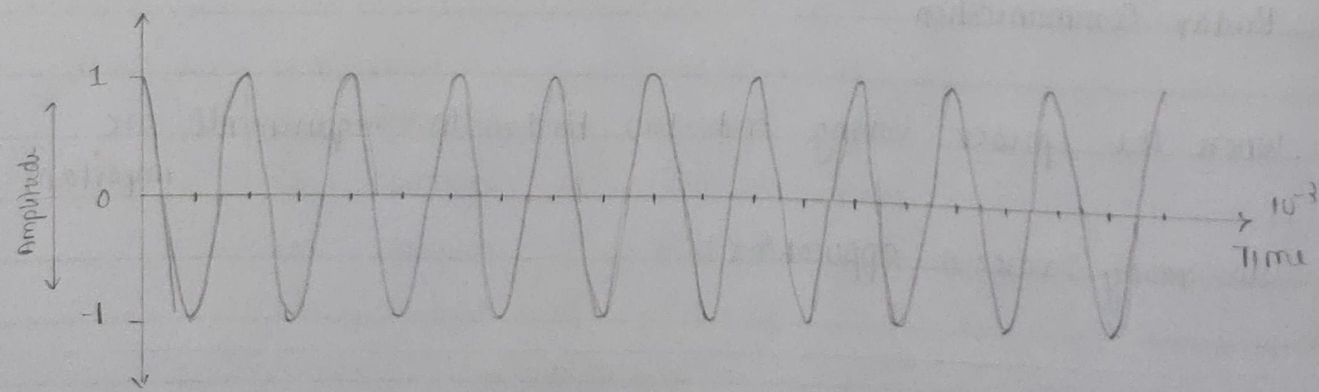
```
clc;
clear all;
close all;
am = 1 ;                          % amplitude of modulating signal
ac = 1 ;                          % amplitude of carrier signal
fm = 500;                         % modulating signal frequency
fc = 5000;                        % carrier frequency
fs = 100000;                      % sampling frequency
ts = 1/fs;                        % sampling interval
N = 10000;                        % no. of samples
t = (-N/2 : 1: (N/2-1))*ts;       % time interval
m = am * cos(2*pi* fm*t);         % modulating signal
mh = am * sin(2*pi* fm*t);        % hilbert transformation of message signal
c = ac * cos(2*pi* fc*t);         % carrier signal
ch = ac * sin(2*pi* fc*t);        % hilbert transform of carrier signal
st = m.*c - mh.*ch;               % SSB-SC signal

% time domain of all signals
subplot (3,2,1)
plot( t,m, 'red', 'linewidth', 1.5 );
axis ( [0 0.005 -2.5 2.5] );
xlabel ('time');
ylabel ('amplitude');
title ('modulating signal');
grid on;
```



Modulating Signal



Carrier Signal



Modulated Signal

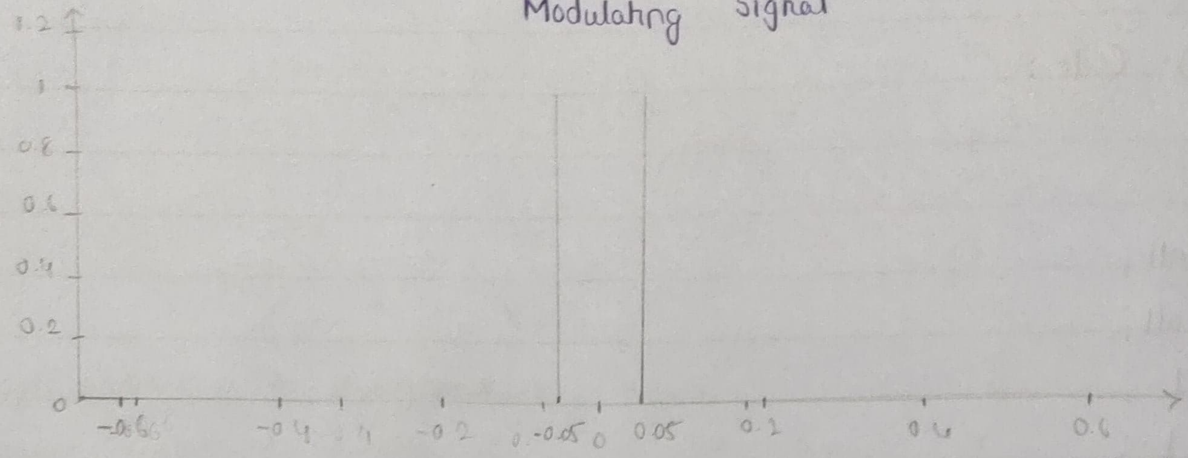
```
subplot (3,2,3)
plot ( t,c, 'black', 'line width', 1.5 );
axis ( [ 0 0.005 -2.5 2.5 ] );
xlabel ('time');
ylabel ('amplitude');
title ('carrier signal');
grid on;
```

```
subplot(3,2,5)
plot ( t,st, 'blue', 'Linewidth', 1.5 );
axis ( [ 0 0.005 -2.5 2.5 ])
xlabel ('time');
ylabel ('amplitude');
title ('modulated signal');
grid on;
```

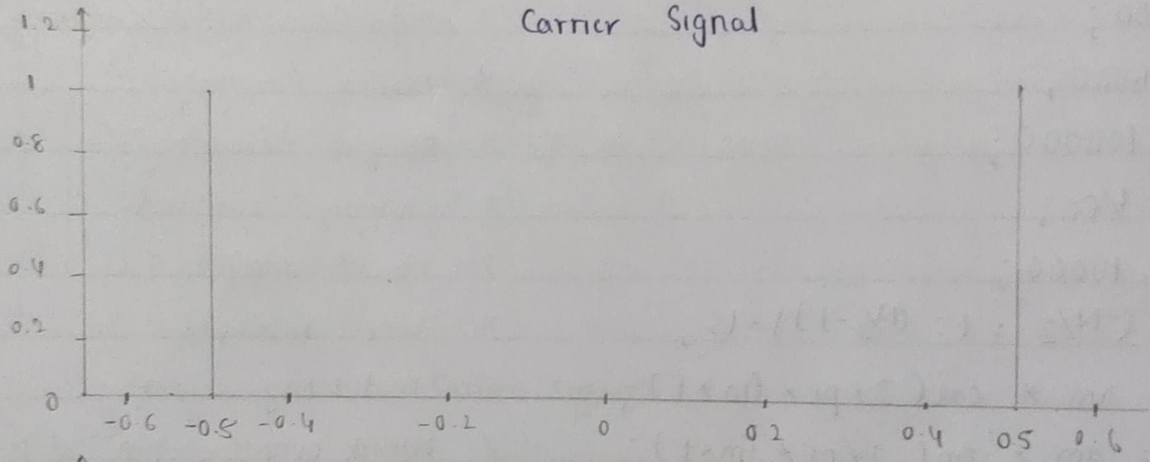
```
% spectrum of all signals
f = ( -N/2 : 1: (N/2 -1)) * fs/N;
M = abs ( (2/N) * fftshift ( fft(m)) );
C = abs ( (2/N) * fftshift ( fft (c)) );
SF = abs ( (2/N) * fftshift ( fft (st)) );
subplot (3,2,2);
plot (f, M/max(M), 'red', 'linewidth', 1.5 );
axis ( [ -2*fc 2*fc -0.1 1.1 ]);
xlabel ('frequency');
ylabel ('amplitude');
title ('modulating signal');
grid on;
```

```
subplot (3,2,4);
plot( f, C/max(C), 'black', 'linewidth', 1.5);
axis ( [ -2*fc 2*fc -0.1 1.1 ];
xlabel ('frequency');
ylabel ( ~~carrier~~ 'amplitude');
title ('carrier signal'); grid on;
```

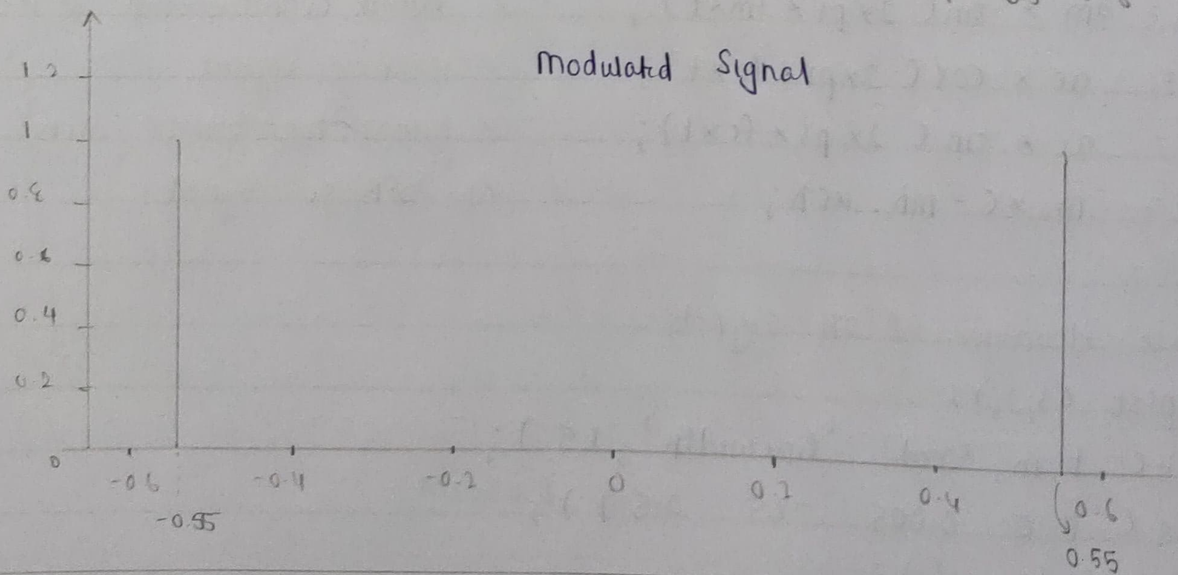
Modulating Signal



Carrier Signal



Modulated Signal



[U19CS012]

```
subplot (3,2,6);
plot ( f, SF/ max (SF) , 'blue', 'linewidth' , 1.5);
axis ( [ -2*fc   2*fc   -0.1  1.1]);
xlabel ( 'frequency');
ylabel ( 'amplitude');
title ( 'modulating signal');
grid on;
```

CONCLUSION : We successfully observed single side Band (SSB-SC) modulation scheme and as we come to know that here low Bandwidth is required for transmission. Hence, we also save power.

—x—

EXPT. NO. 9 NAME: Page No.: 1 Date: 21 Oct

EXPERIMENT 9 [U19CS012]

QAM MODULATION AND DE MODULATION

> AIM: Study of 16 QAM Modulation and Demodulation technique with consetellation ~~dig~~ diagram and waveforms.

> APPARATUS: MATLAB

> THEORY: ① QAM: Quadrature Amplitude modulation or QAM is a form of modulation which is widely used for modulating data signals onto a carrier used for radio communications.

- QAM is a signal in which two carriers shifted in phase by 90° are modulated & the resultant output consists of both Amplitude and phase variations.

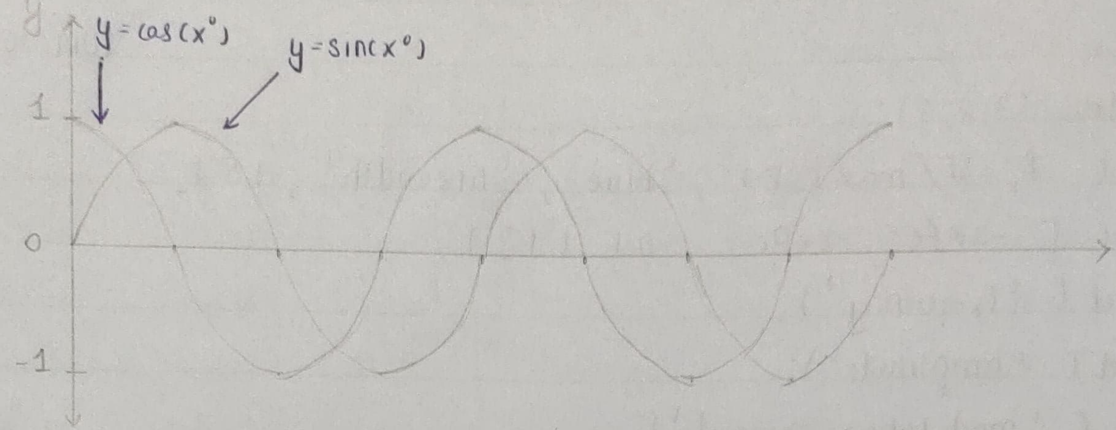
- hence it may also be considered as a mixture of amplitude and phase modulation. QAM is both an analog and digital modulation technique.

② main parameters to be considered while designing any communication system
  - Transmission Power
  - Transmission Bandwidth

- Although the SSB-SC systems are most power and bandwidth efficient but still their performance lags in the noisy environment.

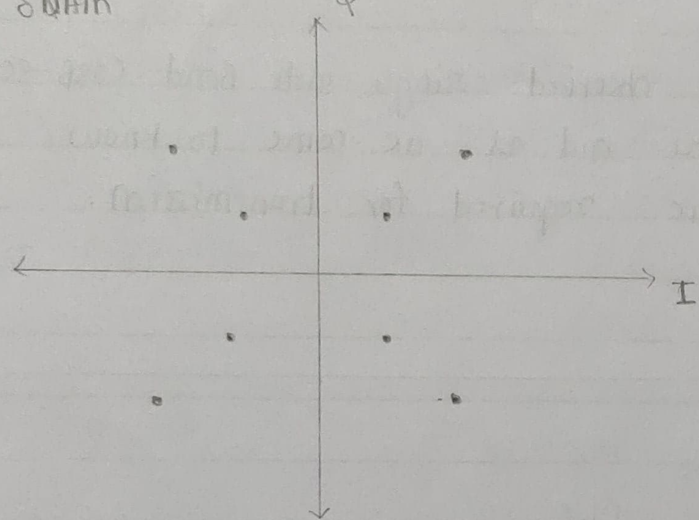
Teacher's Signature:

Quadrature = Sine wave + Cosine wave

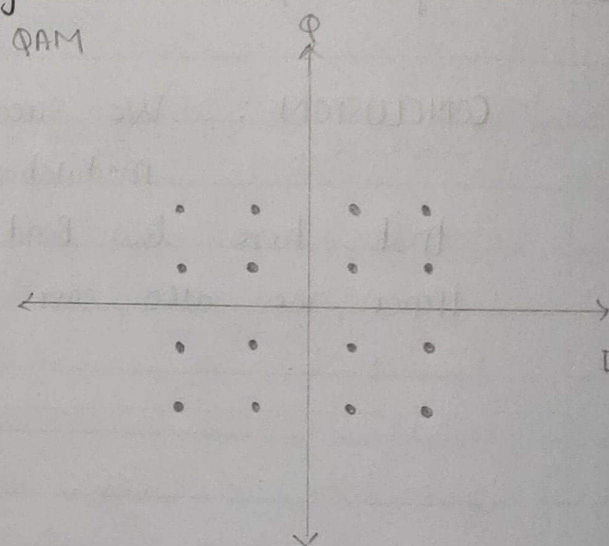


Types of QAM - constellation Diagram →

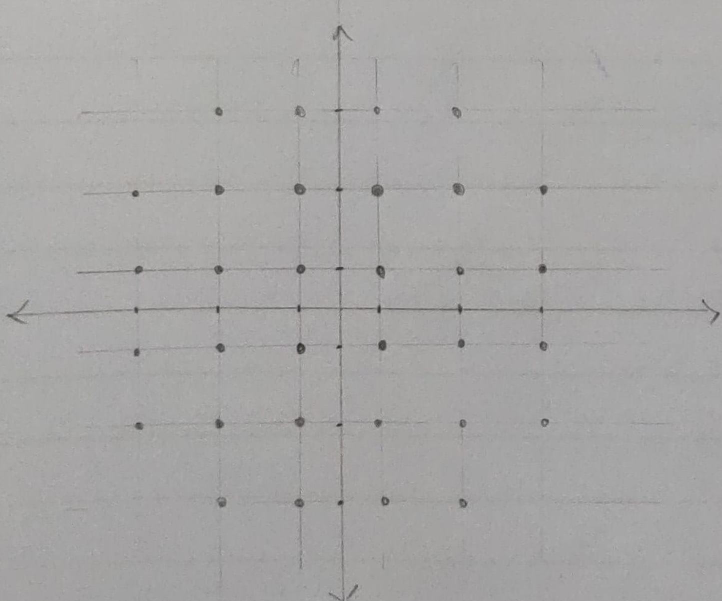
8 QAM



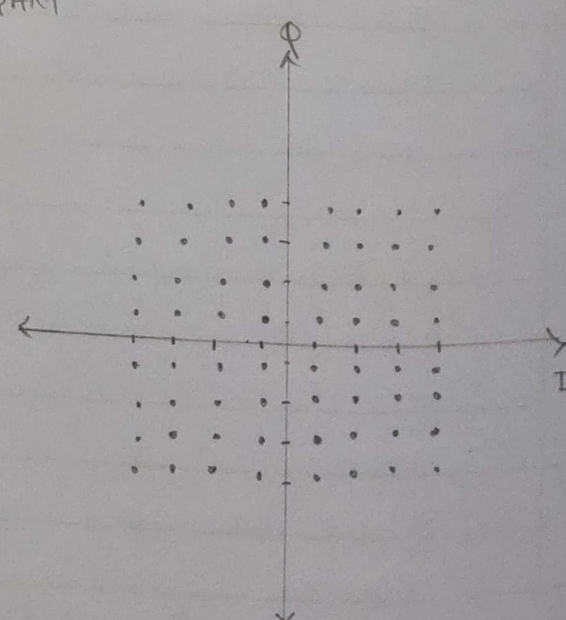
16 QAM



32 QAM



64 QAM



3.> Why QAM?

- The main aim is to save bandwidth. Two modulated signal occupies the same transmission channel.
- A motivation for the use of QAM comes from the fact that a straight amplitude modulated signal occupies twice the bandwidth of the modulating signal.
- This is very wasteful of the available frequency spectrum
- QAM places two independent double sideband suppressed carrier signals in the same spectrum.

4.) Types of QAM

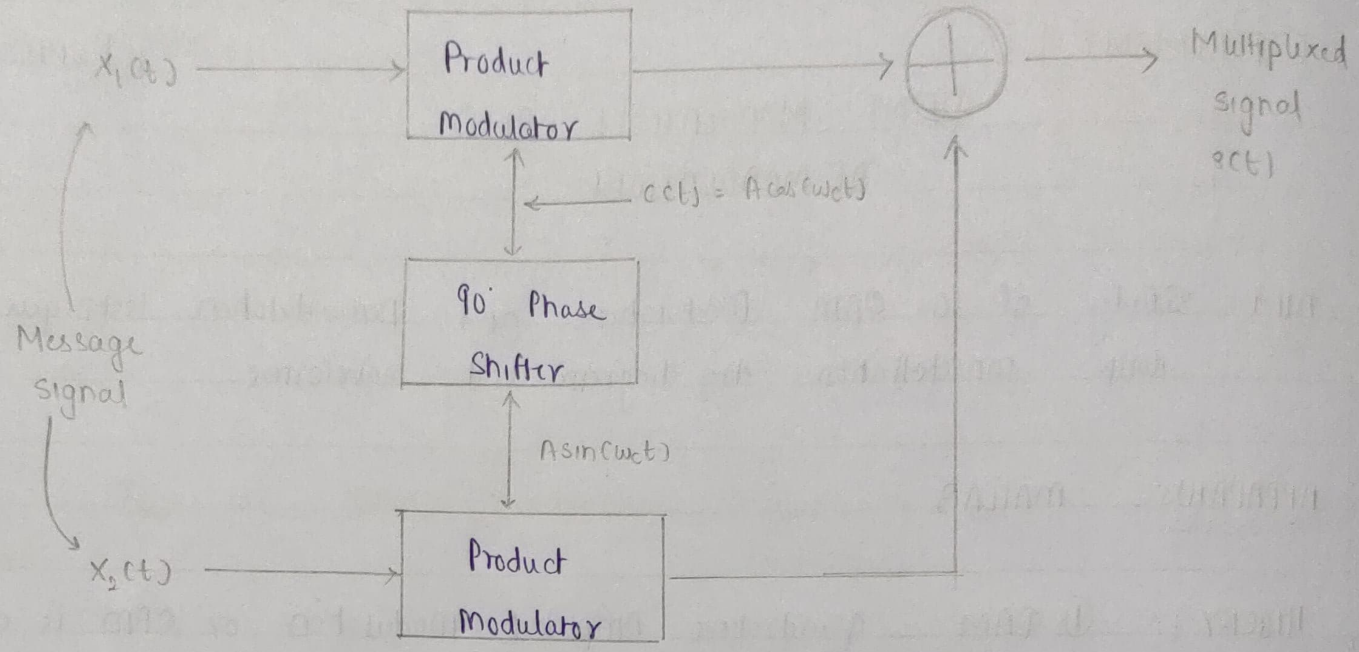
- A variety of forms of QAM are available which include:

  - 16 QAM
  - 32 QAM
  - 64 QAM
  - 128 QAM
  - 256 QAM

5.) QAM modulation

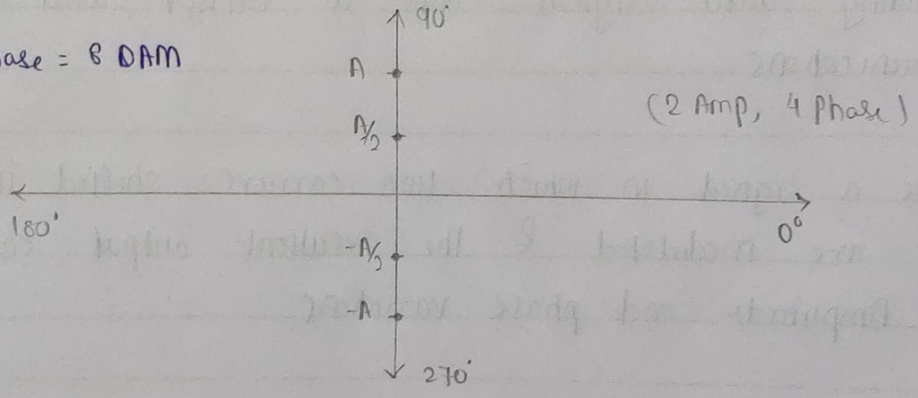
- QAM theory states that both Amplitude and phase changes within a QAM signal.
- The basic way in which a QAM signal can be generated is to generate two signals that are 90° out of phase with each other and them sum them.

QAM Modulation

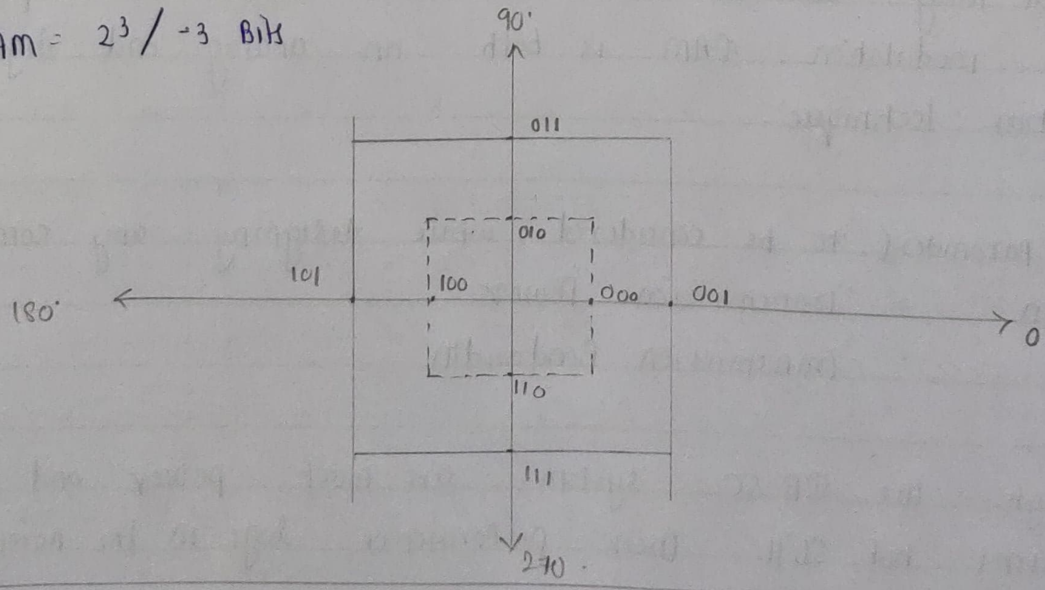


$$s(t) = x_1(t)\, A\cos(\omega_c t) + x_2(t)\, A\sin(\omega_c t)$$

2 Amp × 4 phase = 8 QAM



8 QAM = 2^3 / -3 Bits



- The I and Q signals can be represented by the equation below:

$$I = A\cos(\phi)$$
$$Q = A\sin(\phi)$$

- This signals will not overlap with each other because they are orthogonal.

- It is possible to transmit two DSB-SC signal with in bandwidth of "2fm"

- It provides bandwidth effeciency.

- Gives better performance than SSB and also improves data rate.

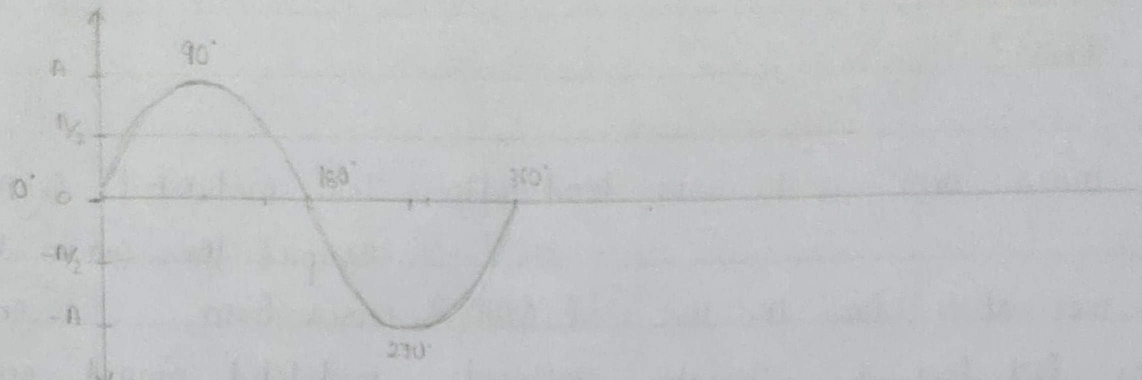
6.) QAM Demodulation

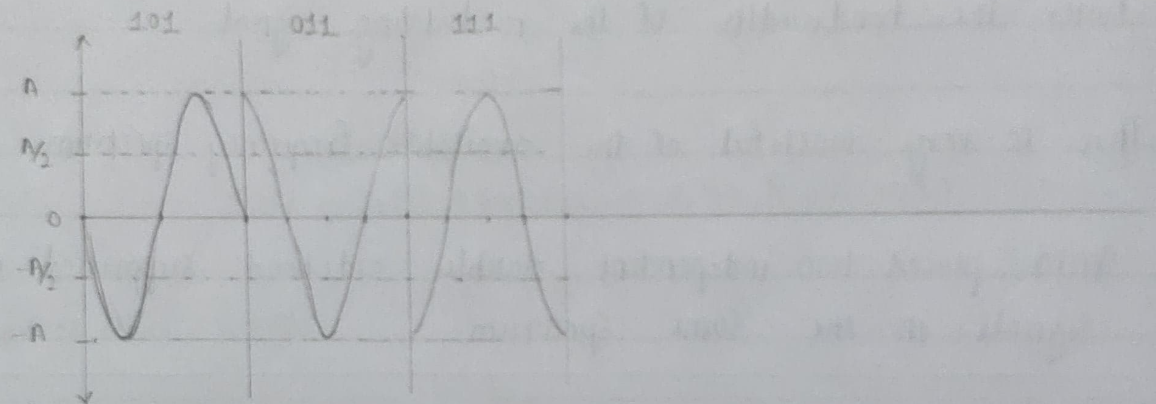
- The QAM demodulator is very much the reverse of the QAM modulator
- The signals enter the system, they are split and each side is applied to a mixer.

7.) Bit error Rate (Received Bits)

- While higher order modulation rates are able to offer much faster data rates and higher levels of spectral effeciency for the radio communication system, this comes at a price.
- The higher order modulation schemes are considerable less relient to noise and interference.

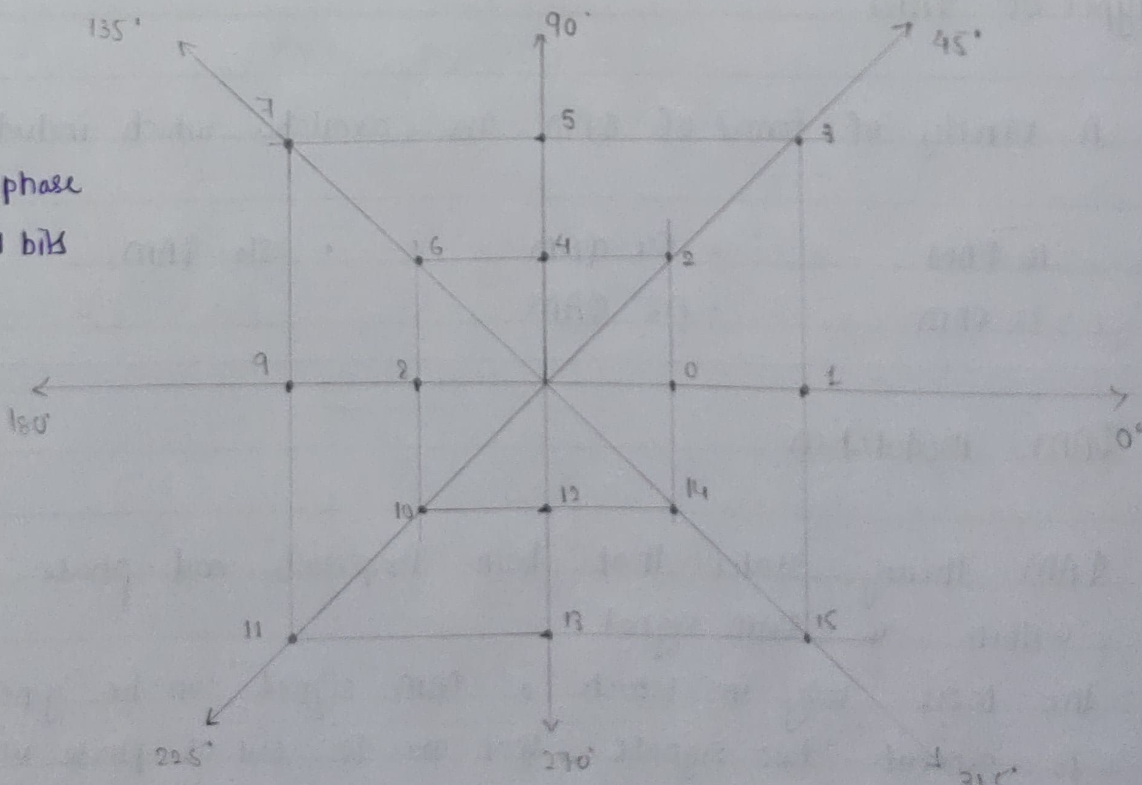
Phasor Diagram





16 QAM

2 Amp * 8 phase
$= 2^4$ / 4 bits



- Many radio communications systems now use dynamic adaptive modulation techniques. They sense the channel conditions and adapt the modulation scheme to obtain the highest data rate for the given conditions.

- M-QAM technique provides better bit error rate performance than M-PSK modulation techniques.

8.> Advantages

- The advantage of using QAM is that it is a higher order form of modulation. As a result, it is able to carry more bits of information per symbol

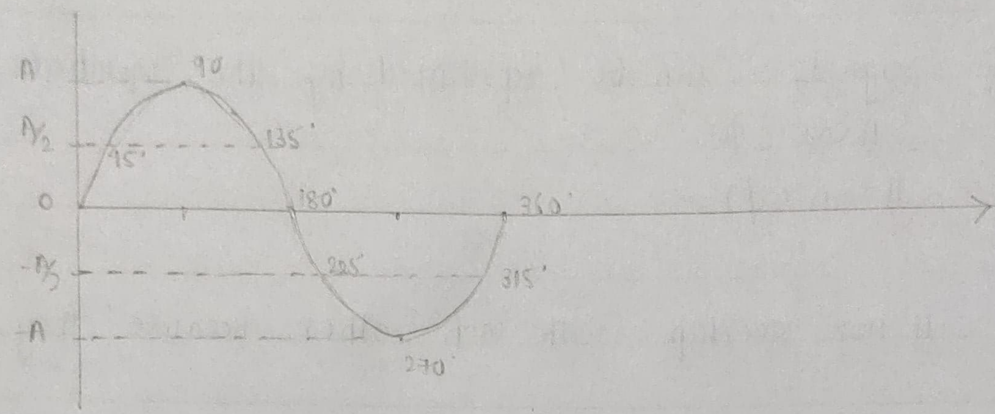
- By selecting a higher order format of QAM, the data rate of a link can be increased.

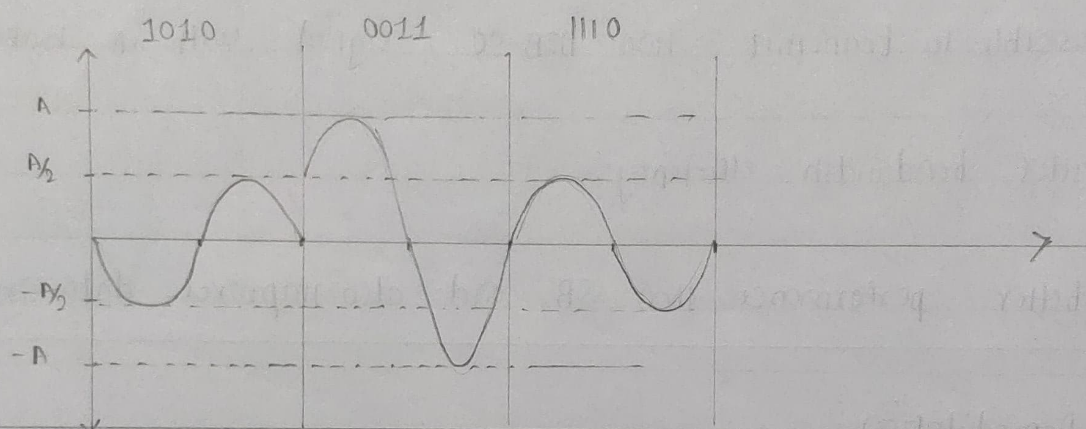
- Bit rate ~~of~~ is increased without increasing the bandwidth.

9.> Applications

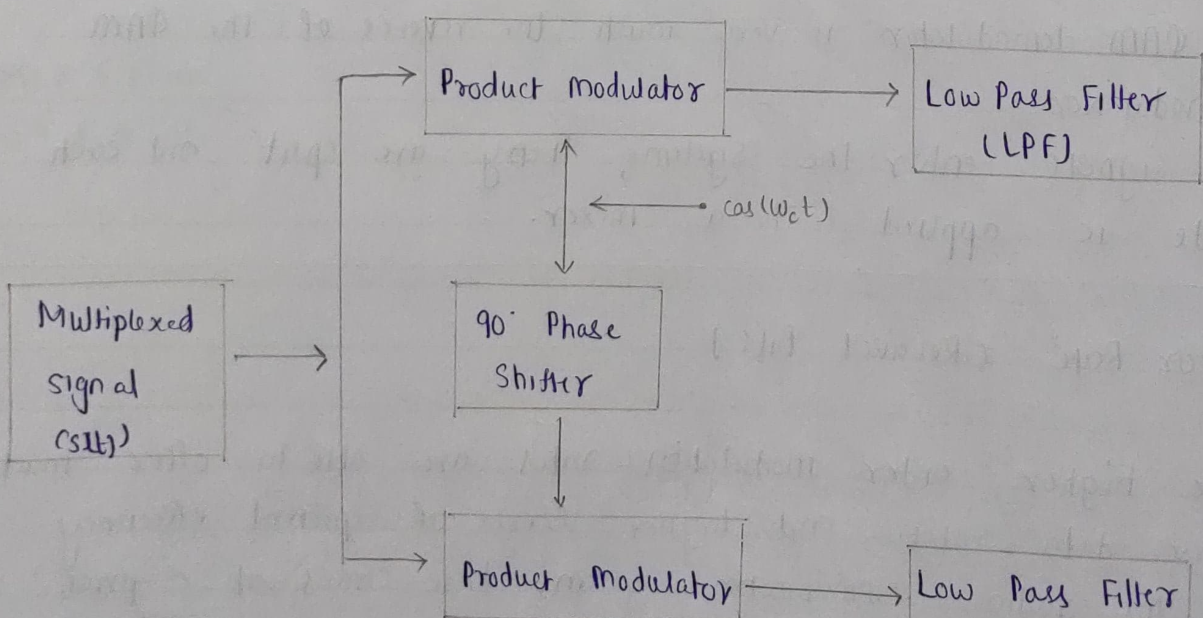
- Quadrature multiplexing is used in color television to multiplex the so-called chrominance signals which carry the information about colors.
- QAM scheme is used on telephone lines for data transmission
- Ultra high capacity Microwave Backhaul systems also use 1024-QAM.

Phasor Diagram





QAM Demodulation

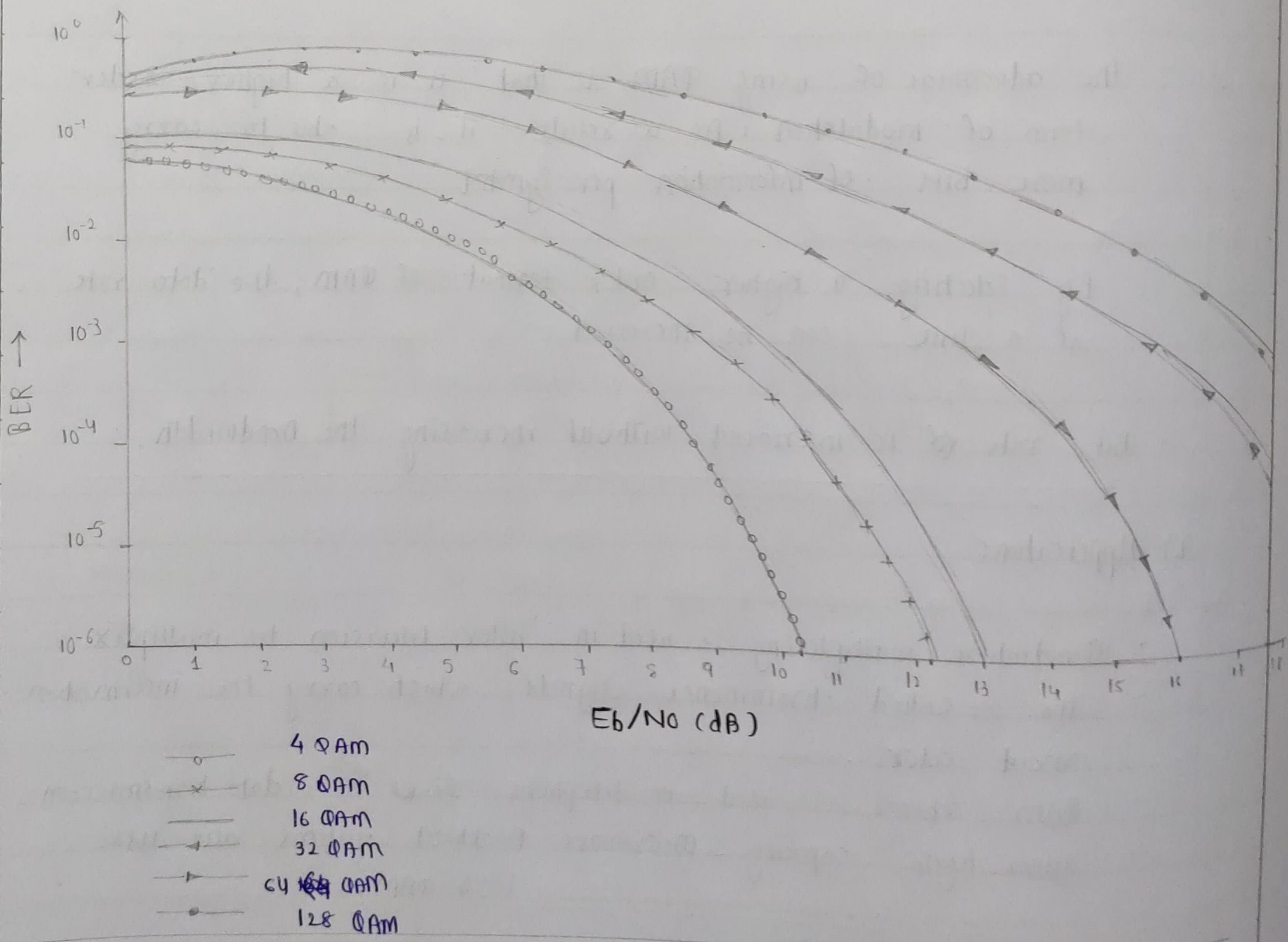


MATLAB Code:

```
clc;
clear all;
close all;
M = 16;
x = (0 : M-1);
y = qammod (x, M);
scatterplot (y);
% z = qamdemod (y, M, pi/4);
% scatterplot (z);
ber_1 = [];                    ((3 * log2(M) * EbNo )/2*(M-1))
for EbNOdB = 0:20;
        EbNO = 10 ^ (EbNOdB/10);
        ber = (1/ log2 (M)) * (2*(1- sqrt(1/m)) * erfc (csqrt(^)) )
        ber_1 = [ber_1 ber];
end
EbNOdB = 0:20;
figure
semilogy (EbNodB, ber_1 (1,:), 'ro-');
xlabel ('Eb/NO (dB)');
ylabel ('BER');
title ('BER of 16-QAM');
axis ([0 16 10^-6 10^0]);
grid on;
```

Modulation	Bit Per Symbol	Symbol Rate
BPSK	1	1 Bit Rate
QPSK	2	1/2 Bit Rate
8PSK	3	1/3 Bit Rate
16 PSK	4	1/4 bit Rate
32 PSK	5	1/5 bit Rate
64 PSK	6	1/6 bit Rate

"BER Values" using MATLAB



[U19CS012]

```
% BER comparision of various M-QAM
clc; clear; close all;
M = [4 8 16 64 128 256]
for i=1: length (M)
    ber_th1 = [];
for EbNOdB = 0:20
    EbNO = 10.^ (EbNodB/10);
    ber = (1/ log2 (M(i))) * (2* (1-sqrt(1/m(i)))) *
          erfc (sqrt ((3* log2(M(i)) * EbNO) / 2*(m(i)-1))))
    ber_thi = [ber_th1 ber];
end
    berth = [ber_th; ber_thi];
end
EbNodB = 0:20;
semilogy (EbNodB, ber_th (1,:), 'ro-'); hold on
semilogy (EbNodB, ber_th (2,:), 'g+-'); hold on
semilogy (EbNodB, ber_th (3,:), 'y.-'); hold on
semilogy (EbNodB, ber_th (4,:), 'k>-'); hold on
semilogy (EbNodB, ber_th (5,:), 'c<-'); hold on
semilogy (EbNodB, ber_th (6,:), 'mx-'); hold on

xlabel ('Eb/NO(dB)'); ylabel ('BER'); axis ([0 20 10^-6 10^0]);
```

» CONCLUSION: We successfully examined the 16-Quadrature Amplitude Modulation (16-QAM) and Demodulation scheme.

We also evaluated 16 BER values for different QAM using MATLAB.

-x-

Teacher's Signature:

EXPERIMENT 10 [U19CS012]

PULSE CODE MODULATION AND DEMODULATION

AIM: To demonstrate the pulse code modulation (PCM) and demodulation technique. To show the sampled, quantized / encoded and decoded time domain signal for different bit-codes.

SOFTWARE: MATLAB

THEORY: 1.> Pulse Code Modulation (PCM)

- PCM is a technique, which is used to convert an analog signal into digital signal.

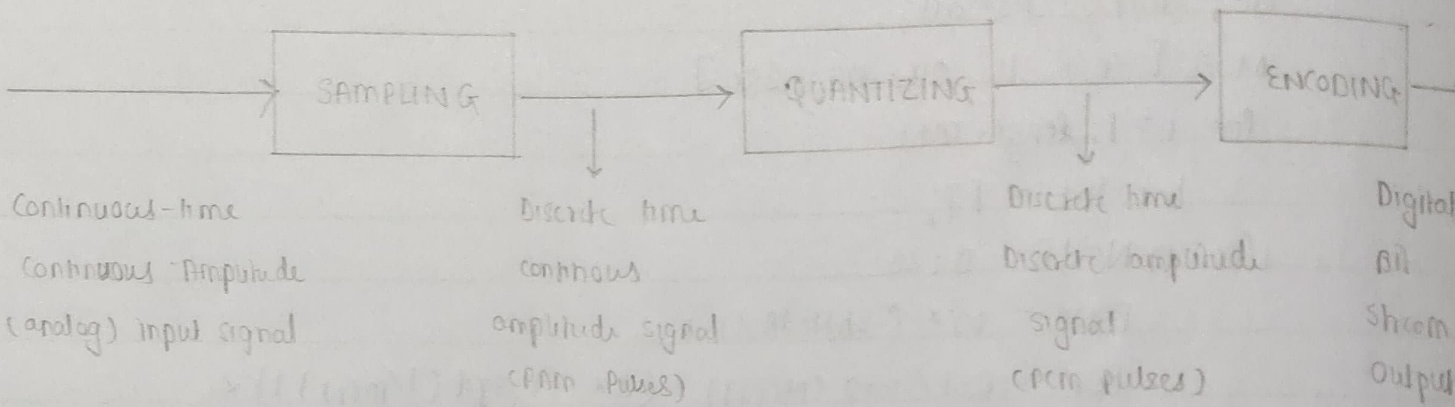
- PCM is a preferred method of communication within public switched telephone network (PSTN)

- A PCM stream is determined by two following steps:

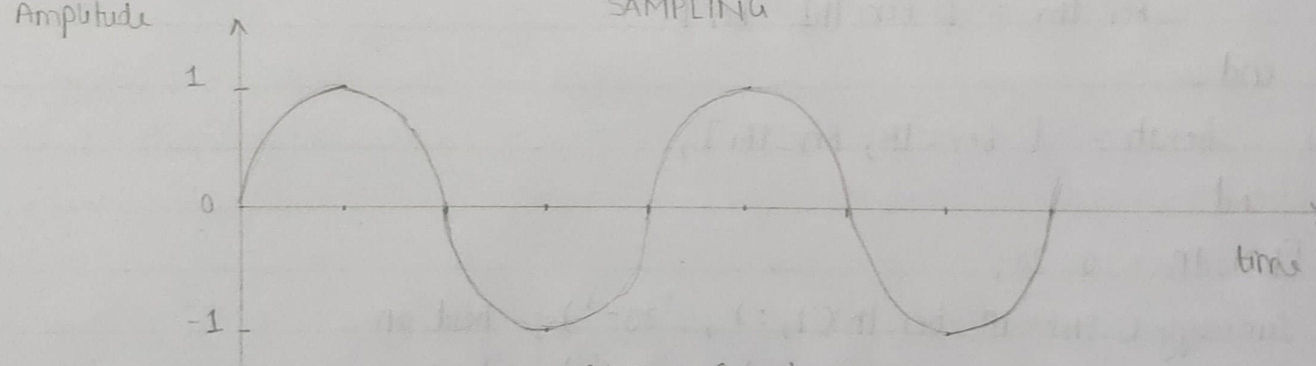
  a) Sampling Rate: which the number of times per second that samples are taken.

  b) Bit depth: which determines the number of possible digital values that can be used to represent each sample.

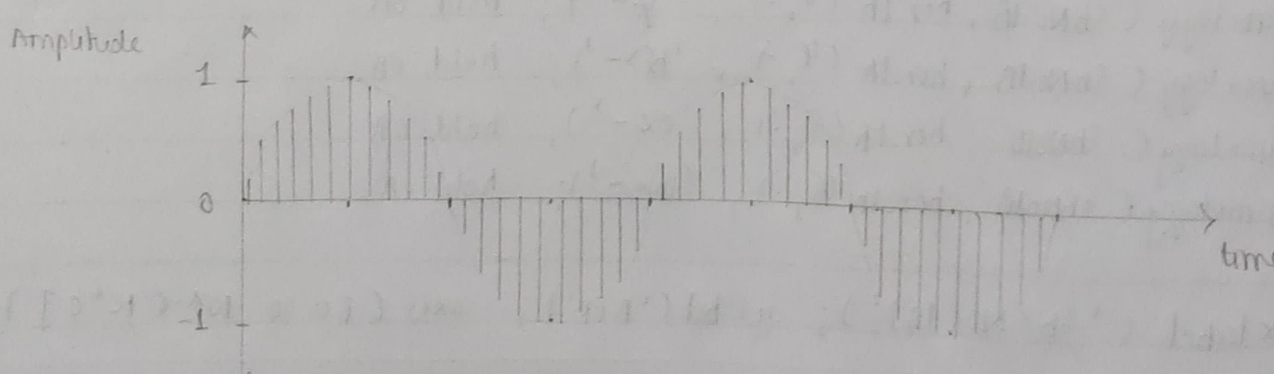
- Hence, the output of PCM resembles a binary sequence.



SAMPLING



Analog Signal



Discrete Time signal

2.) Reasons for Digital Transmission

- Less susceptible to interference cause by noise due to discrete level.

- Easy to detect errors due to discrete levels

- Easy to Encrypt (Higher Security)

- Simpler to store digital data

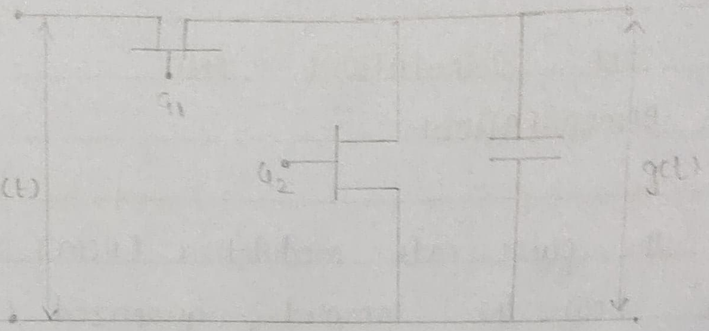
3.) Sampling

- Sampler extract samples of a continuous signal.
- Sampler produces samples that are equivalent to the instantaneous value of the continous signal at the specified various points.
- The sampling process generates Flat-top Pulse Amplitude Modulated (PAM) signal.

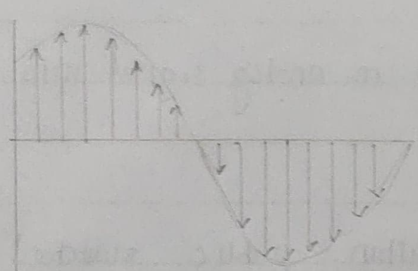
4.) Quantization

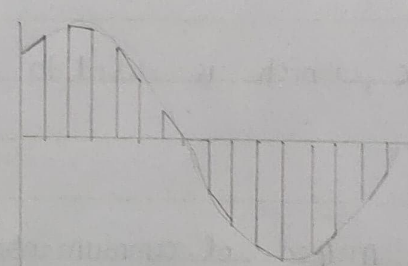
- Quantization is done by dividing the range of possible values of the analog samples into some different levels and assigning the center value of each level to any sample in the quantization interval.
- Quantization approximates the analog signal values with the nearest quantization values.

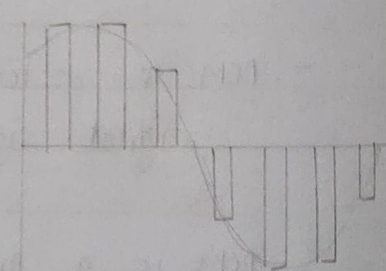
Flat-top PAM



Instantaneous Sampling	Natural Sampling	Flat-top Sampling
It is not practical method	This method is used practically	This is also used practically.
Sample rate = infinity	Sample rate satisfied Nyquist	Sample rate satisfied Nyquist

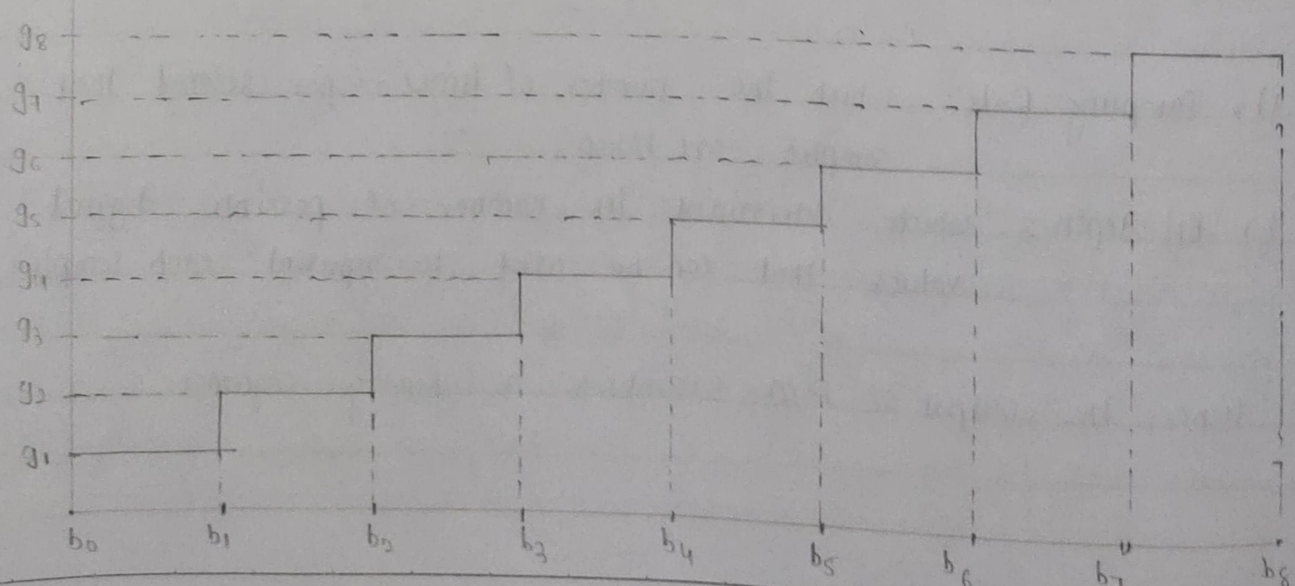






Uniformly Quantized Signal

A/D output = n bits per sample (quantization level $M = 2^m$)



EXPT. NO. | NAME: | Page No.: 5 | Date:

5) Pulse Code Modulation (PCM) is a method of converting an Analog signal into a digital signal. (A/D conversion)

- PCM produces a series of number or digits instead of pulse train.
- Each one of these digits, in binary code, represents the approximate amplitude of the signal sample at that instant.

6) Concluding Remark for PCM

- In PCM Transmitter, the signal $x(t)$ is first passed through the low pass filter of cut-off frequency F_m Hz.

- This low pass filter blocks all the Frequency components above F_m Hz. This means that now the signal $x(t)$ is band-limited to F_m Hz.

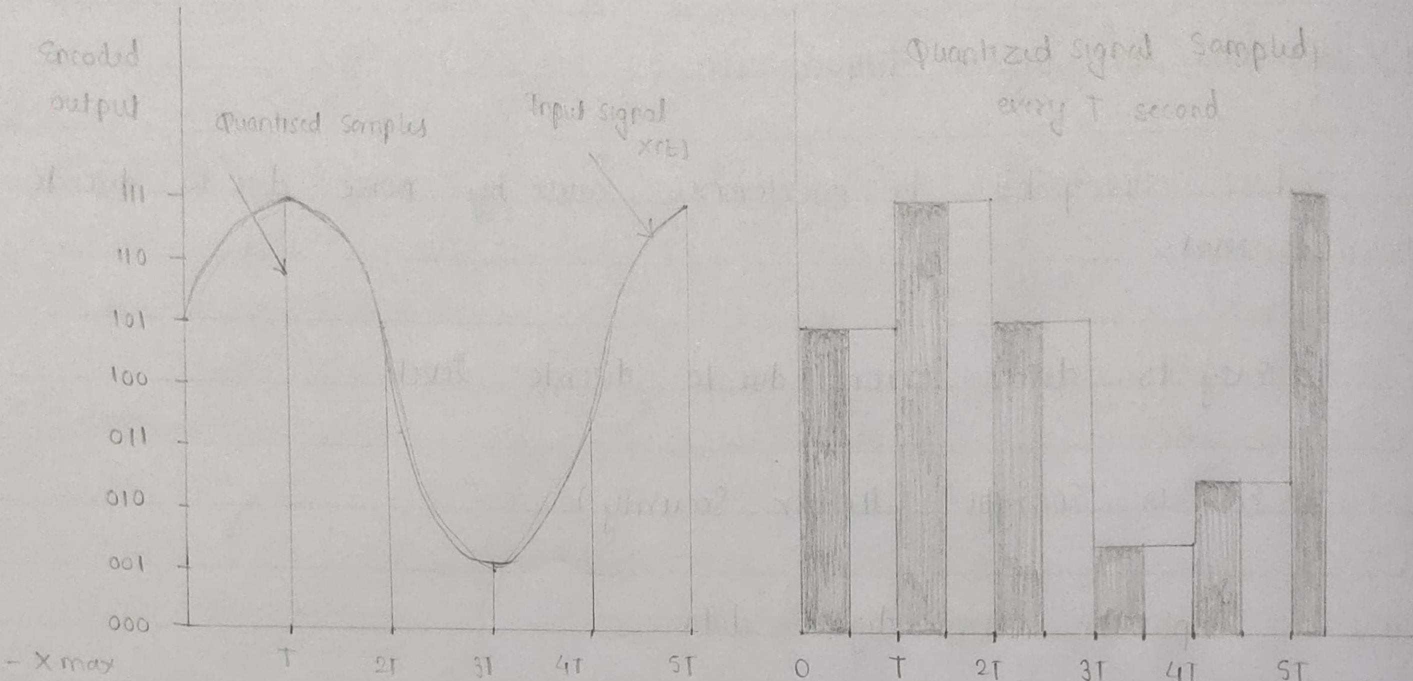
- The Sample and Hold circuit then samples this signal at the rate of F_s.

- Sampling Frequency F_s is selected sufficiently above Nyquist rate to avoid Aliasing.

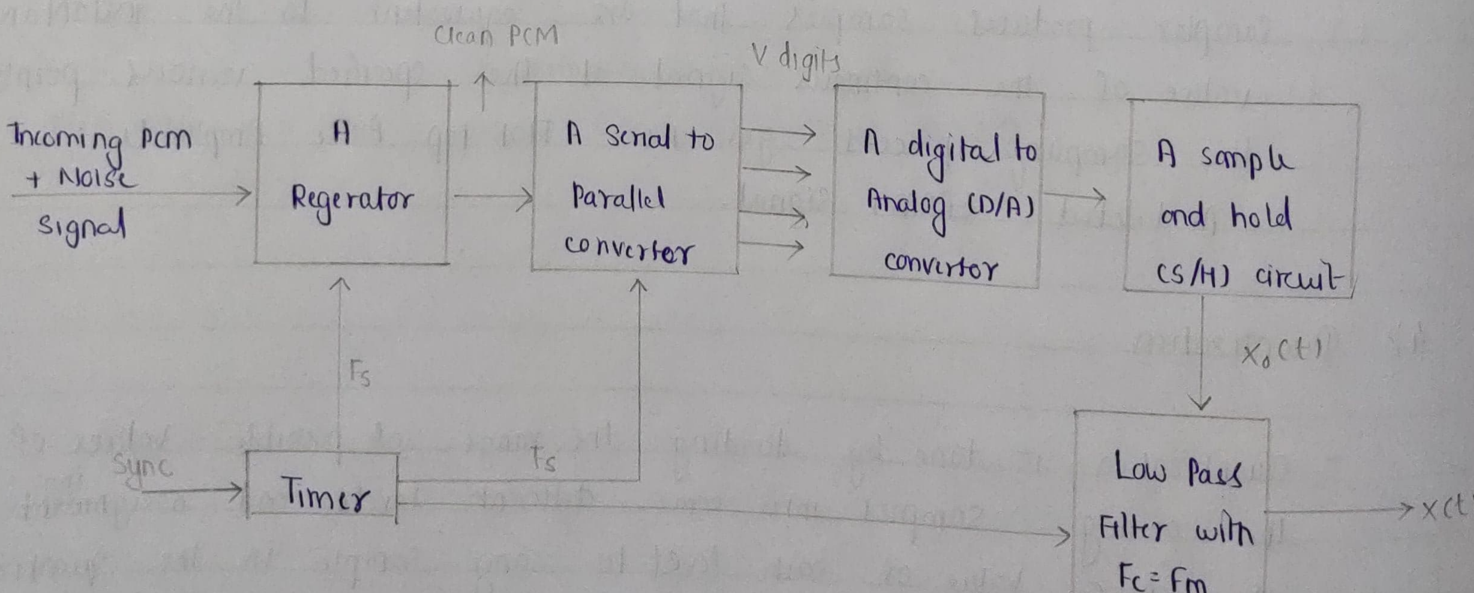
- The Output from sample and hold circuit is denoted by $x(nT_s)$

Transmitter

Figure : 1 Quantization of a sampled Analog signal



PCM Receiver



- This signal $x(nT_s)$ is discrete in time and continuous in Amplitude
- A q-level quantizer compares input $x(nT_s)$ with its fixed digital levels.
- Quantized signal is then encoded in PCM output using encoder.

7.) PCM Standards

- There are two standards of PCM ① The European standard
  ② The American standard
- They differ slightly in the detail of their working but the principles are the same.

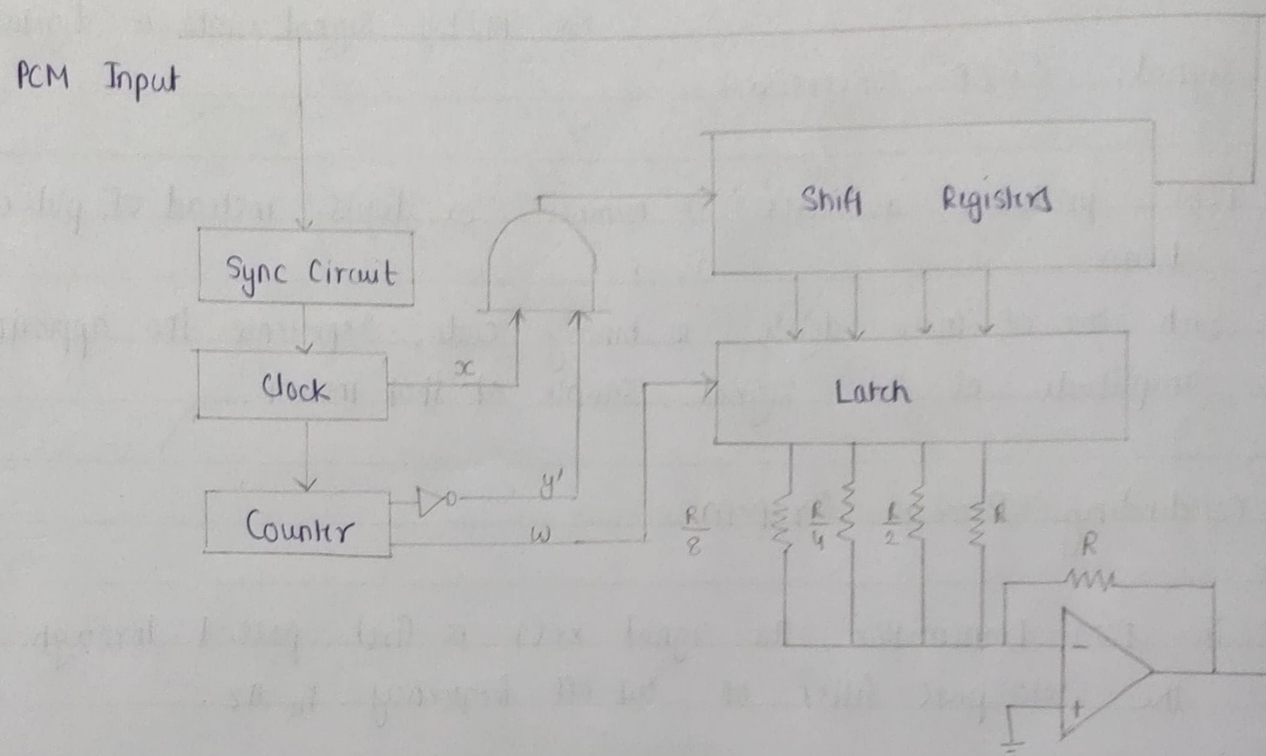
- European PCM = 30 channels
- North American PCM = 24 channels
- Japanese PCM = 24 channels

In India, we follow the European PCM of 30 channels system working.

8.) Applications

- In Compact Disks
- Digital Telephony
- Digital Audio Applications

PCM Demodulator



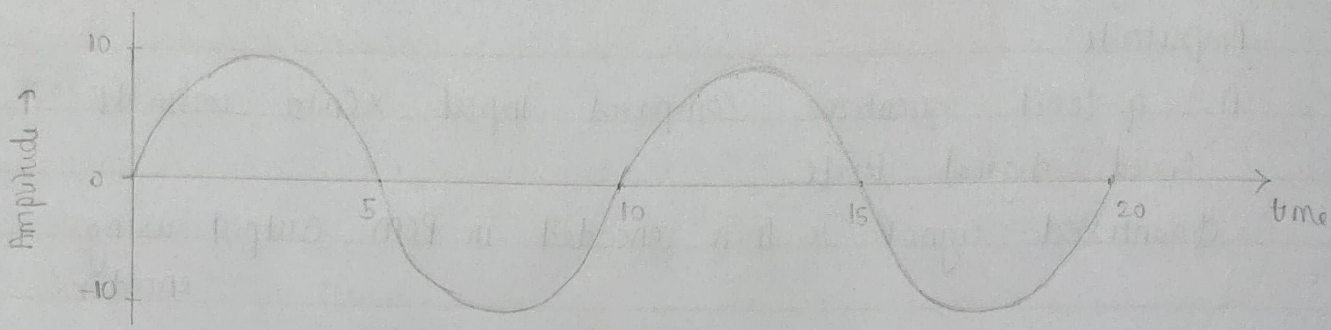
EXPT. NO. NAME: Page No.: 1 Date:

MATLAB CODE: ① Sampling

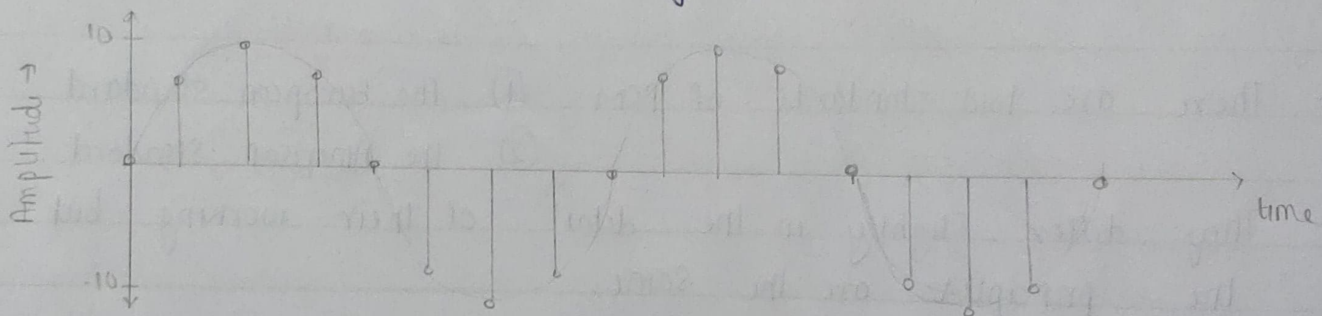
```
n = input('Enter n value for n-bit pcm system:');
n1 = input('Enter number of samples in a period:');
 L = 2^n;
%% Signal generation
% x = 0:1/100:4*pi;
% y = 8*sin(x)
% subplot(2,2,1);
% plot(x,y); grid on;
% SAMPLING OPERATION
 x = 0:2*pi/n1:4*pi;
 s = 8*sin(x);
 subplot(3,1,1);
 plot(s);
 title('Analog Signal');
 ylabel('Amplitude -->');
 xlabel('Time -->');
 subplot(3,1,2);
 stem(s);
 grid on;
 title('Sampled Signal');
 ylabel('Amplitude -->');
 xlabel('Time -->');
```

Teacher's Signature:

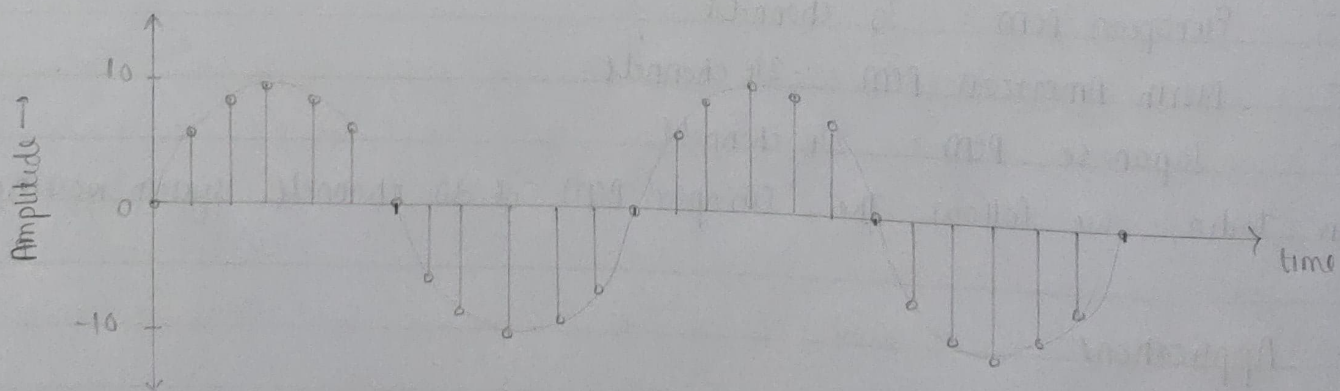
Outputs from MATLAB



Analog Signal ↑



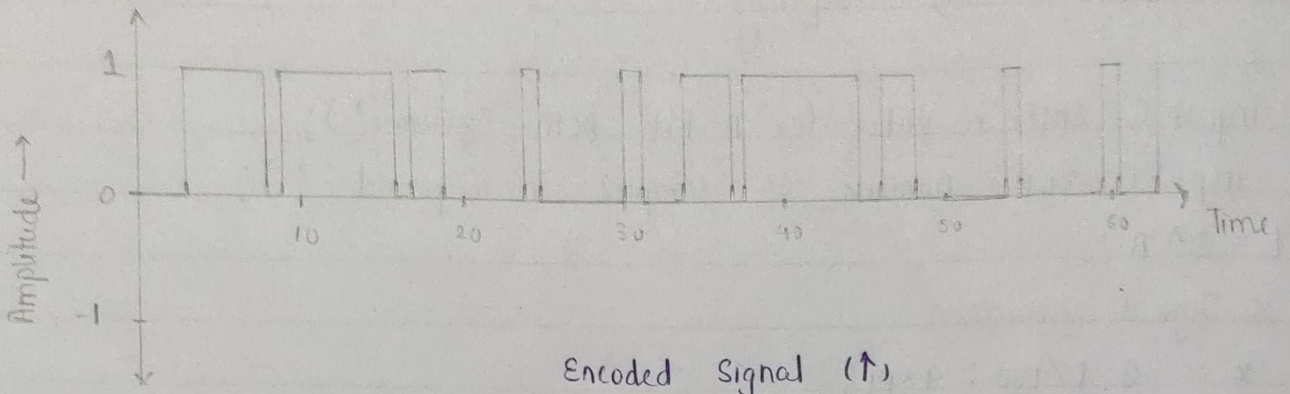
Sampled Signal (↑)



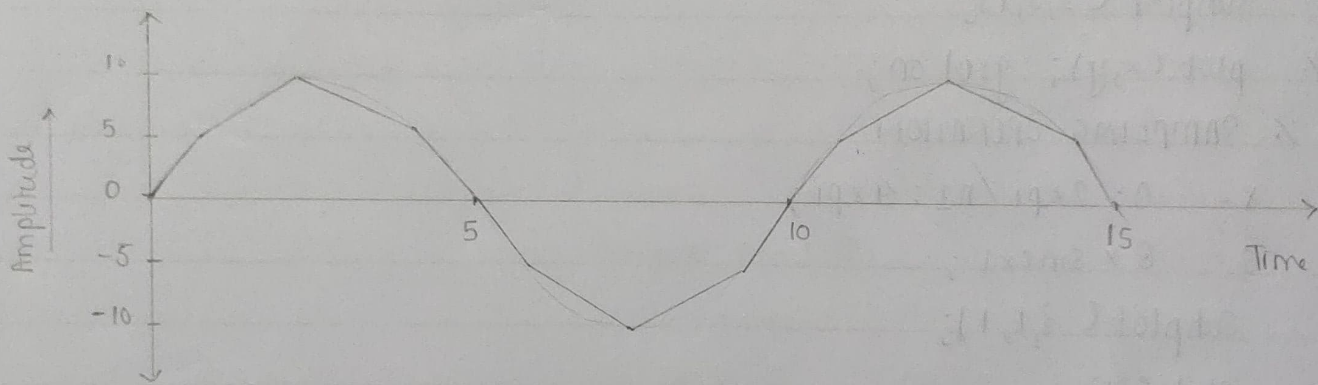
Quantized Signal

(2) Quantization Process

```
vmax = 8
vmin = -vmax;                          % level are between vmin & vmax with diff del
del = (vmax - vmin)/L;
part = vmin: del: vmax;
code = vmin - (del/2) : del: vmax + (del/2);
[ind,q] = quantiz( s, part, code);     % quantization Process

l1 = length(ind);
l2 = length(q);

for i=1:l1                             % to make index as binary decimal so started 0 to N
    if (ind(i)~=0)
        ind(i) = ind(i)-1;
    end
    i=i+1;
end

for i=1:l2                             % to make quantize value in between lvl
    if( q(i) == vmin - (del/2) )
        q(i) = vmin + (del/2);
    end
subplot(3,1,3);                        % Display Quantize Values
stem(q); grid on;
title('Quantized Signal');
ylabel('Amplitude -->');
xlabel('Time -->');
```



Encoded Signal (↑)



Demodulated Signal

(3) Encoding

```
% Encoding Process
figure
code = de2bi( ind, 'left-msb');        % convert decimal to binary
k=1;
for i=1: l1
    for j=1:n
        coded(k) = code(i,j);          % convert code matrix to coded row
        j=j+1;
        k=k+1;
    end
  i=i+1;
end
subplot(2,1,1); grid on;
stairs( coded);
axis ( [0 100 -2 3] );                 % Display the encoded signal
title( 'Encoded Signal');
ylabel( 'Amplitude -->');
xlabel( 'Time -->');
```

[U19CS012]

(4) Demodulation of PCM signal

```
quant = reshape ( coded, n, length(coded)/n );
index = bi2de ( quant', 'left-msb' );        % getback index in decimal
q = del * index + Vmin + (del/2) ;           % getback quantized values
subplot ( 2,1,2 );   grid on;
plot (q);
title ('Demodulated Signal');                % Plot Demodulated signal
ylabel ('Amplitude -->');
xlabel ('Time -->');
```

> CONCLUSION: (1) We successfully demonstrated the Pulse Code Modulation (PCM) and demodulation technique.

(2) We observed block diagrams for receiver and transmitter of PCM signals. In later stage we also observed demodulation circuit which consist of Shift Registers, Latch and opamp.

(3) In the last phase, we executed MATLAB code and observed sampling, Quantization, Encoding and Demodulation waves and drawn them.

—×—

Teacher's Signature:

EXPERIMENT - 11 [U19CS012]

DELTA MODULATION

AIM : To demonstrate the delta modulation (DM) and demodulation Technique.

SOFTWARE : MATLAB

THEORY : 1 Delta Modulation

- It is a technique used to convert analog-to-digital and digital-to-analog signal.

- In this modulation signal is sent in differential form, the data is encrypted/ transmitted in 1 bit.

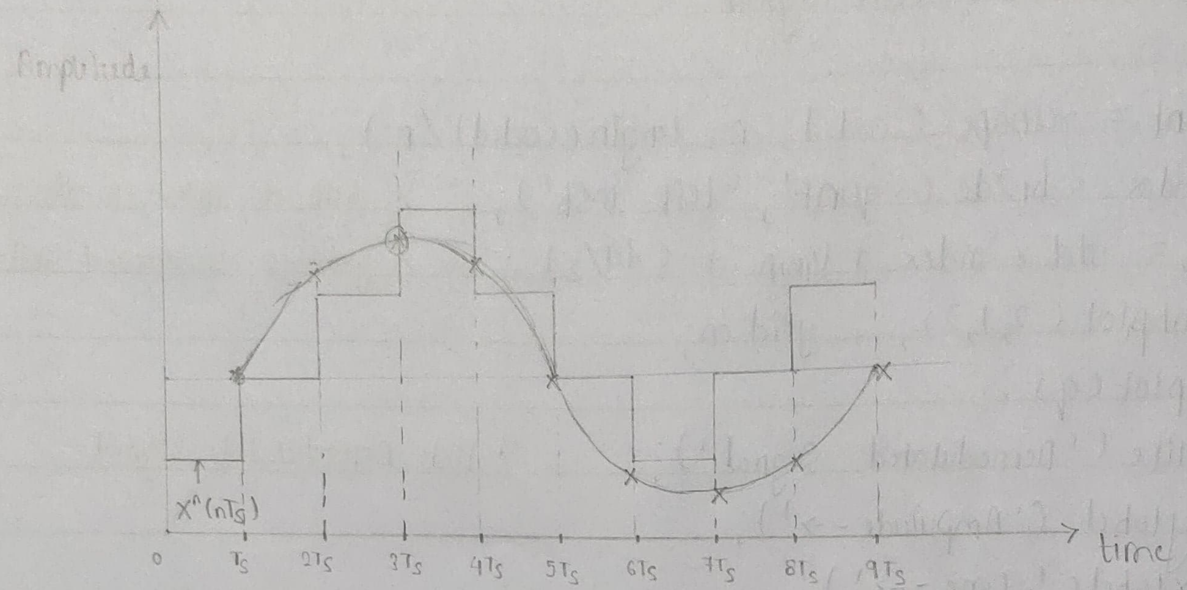
- The analog signal is approximated with series of segments and each segment is compared to original analog to determine the change in relative Amplitude.

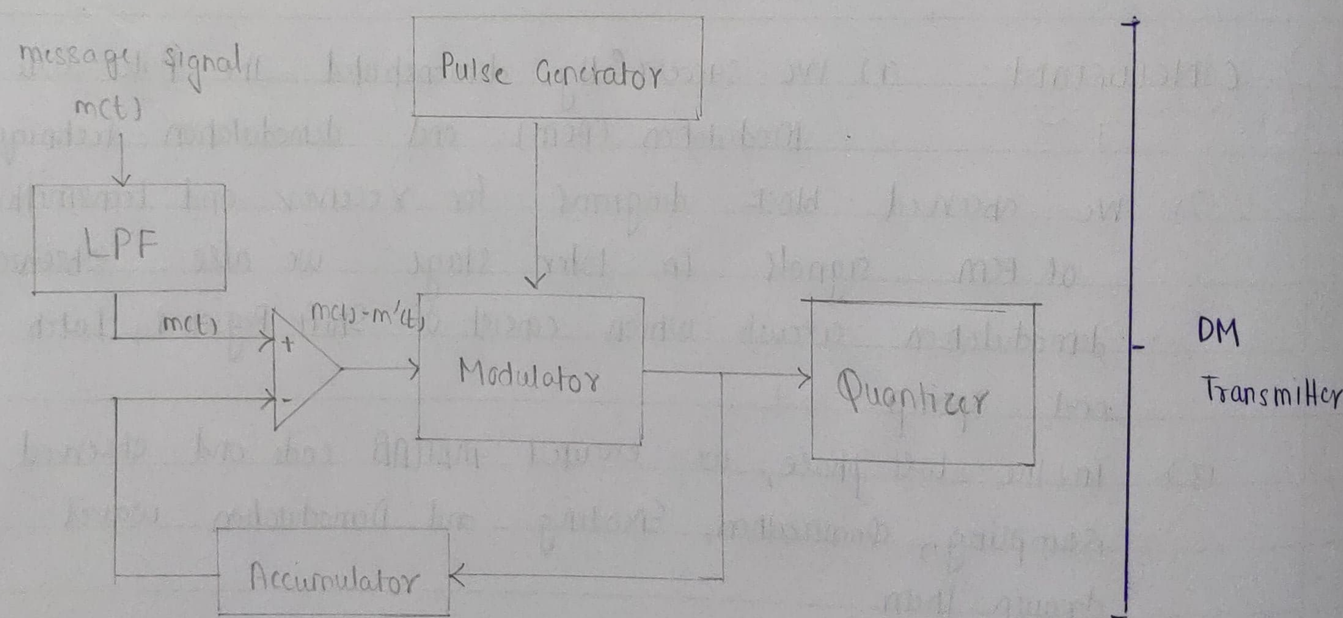
- Hence, only changes in information is sent and if no change occurs it remains on the same state.

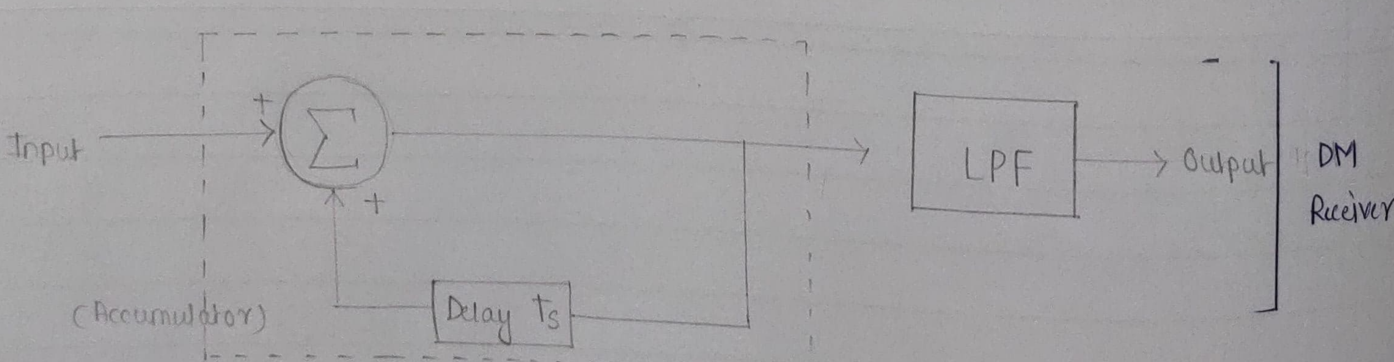
- This is the simplified form of Differential Pulse Code Modulation and also called as 1 bit (2 level) version of DPCM.

- It provides a staircase approximation of over-sampled base-band signal. Here, the difference between the present sample and

Teacher's Signature:


Amplitude
x^(nTs)
0
Ts
2Ts
3Ts
4Ts
5Ts
6Ts
7Ts
8Ts
9Ts
time


message signal
m(t)
LPF
m(t)
+
-
m(t)-m'(t)
Pulse Generator
Modulator
Quantizer
Accumulator
DM
Transmitter


Input
+
+
Σ
Delay Ts
(Accumulator)
LPF
Output
DM
Receiver

previous approximate sample is quantized into two levels ie Δ (delta).

- This is used for voice transmission.

2.) Operating Principle

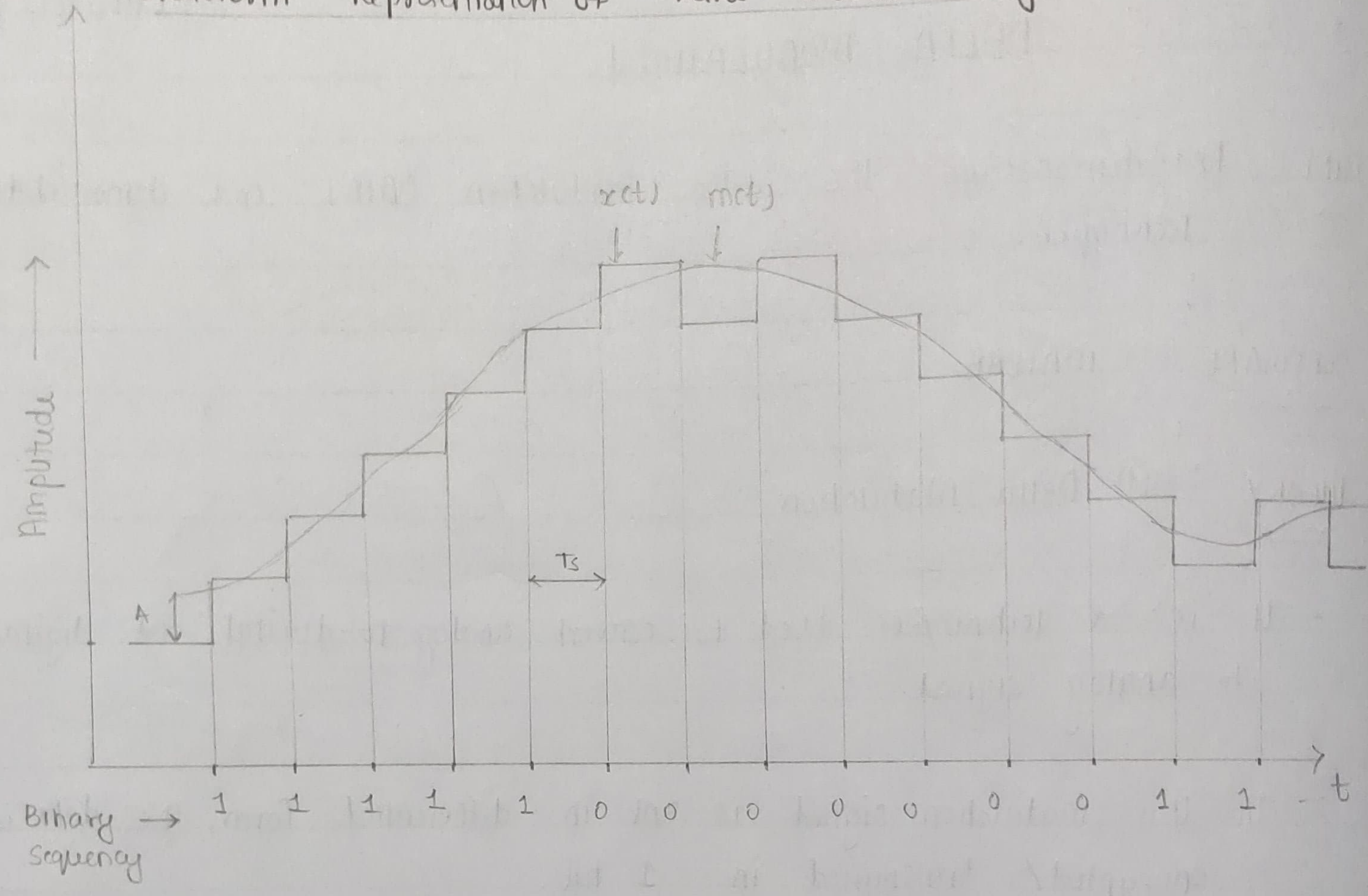
- The operating principle of DM is such that, a comparision between present and previously sampled value is performed, the difference of which decides the increment or decrement in the transmitted values.

- When the two sample values are compared, either we get difference having a positive polarity or negative polarity.

- If the difference polarity is positive, then the step of the signal denoted by Δ is increased by 1. As against in case when difference polarity is negative then step of the signal is decreased ie. reduction in Δ.

- When +Δ is noticed i.e. increases the step size, then 1 is transmitted, However, in case of -Δ ie decrease in step size, 0 is transmitted.

Waveform Representation of Delta Modulated Signal



3.> Advantages of Delta Modulation

- Due to transmission of 1 bit per sample, it permits low channel bandwidth as well as signaling rate.

- ADC is not required. Thus permits easy generation and detection.

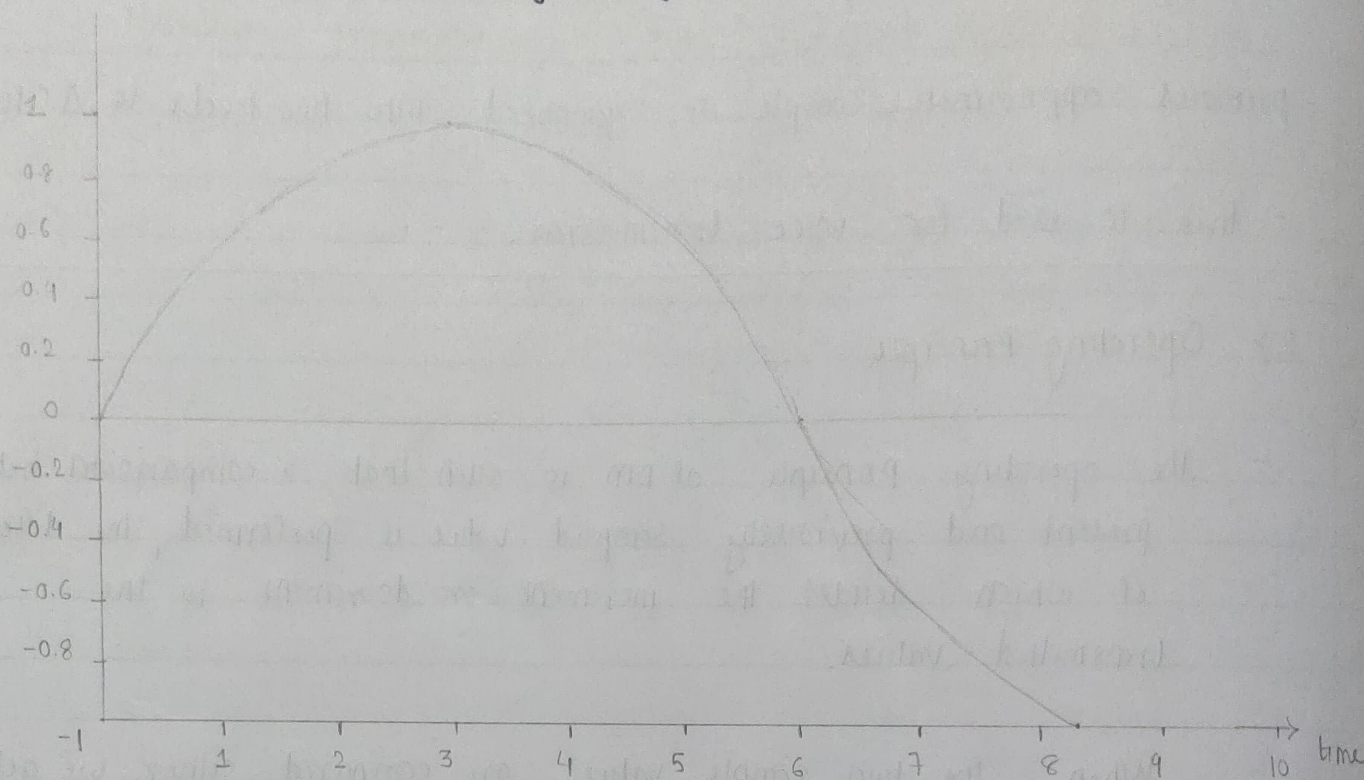
4.> Disadvantages of Delta Modulation

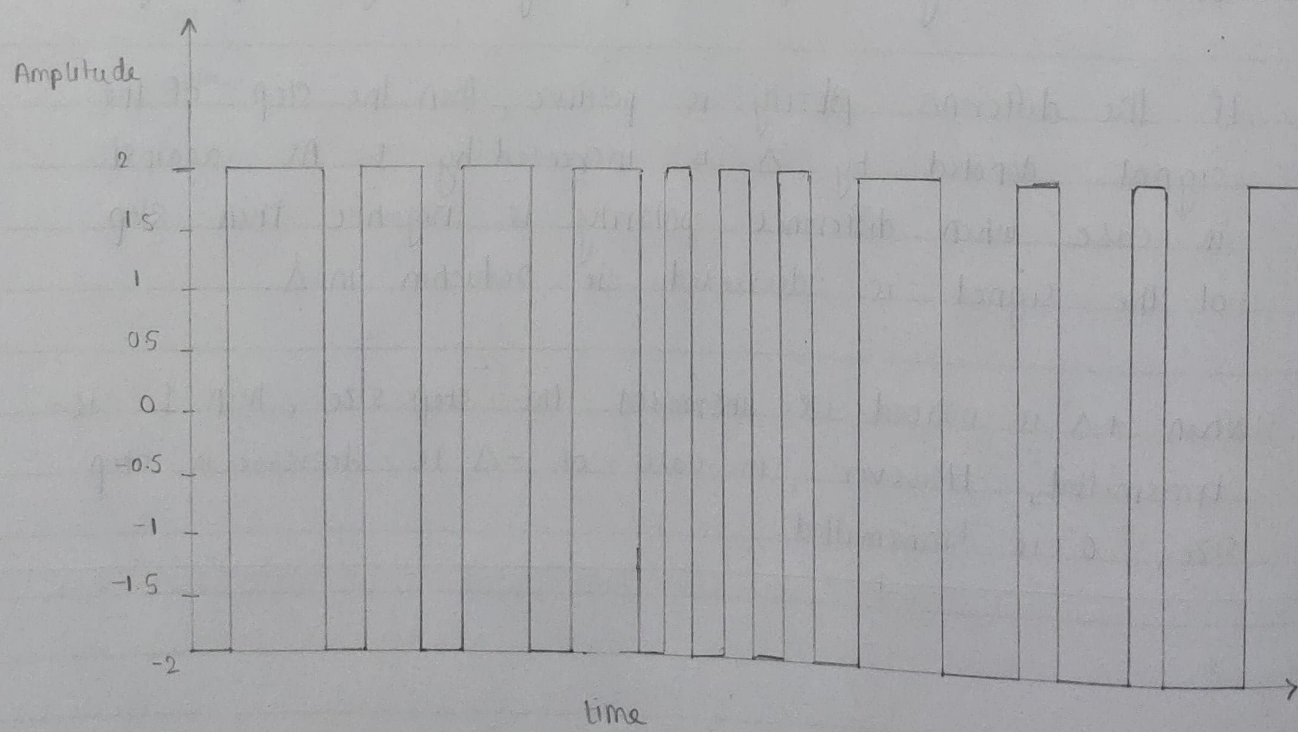
- Delta modulation leads to drawbacks such as slope overload distortion (when Δ is small) and granular noise (when Δ is large.

5.> Application of Delta Modulation

- It is widely used in radio communication devices and digital voice storage and voice transmission.

Original signal





DM output

EXPT. NO. NAME:

Page No.: 1

Date:

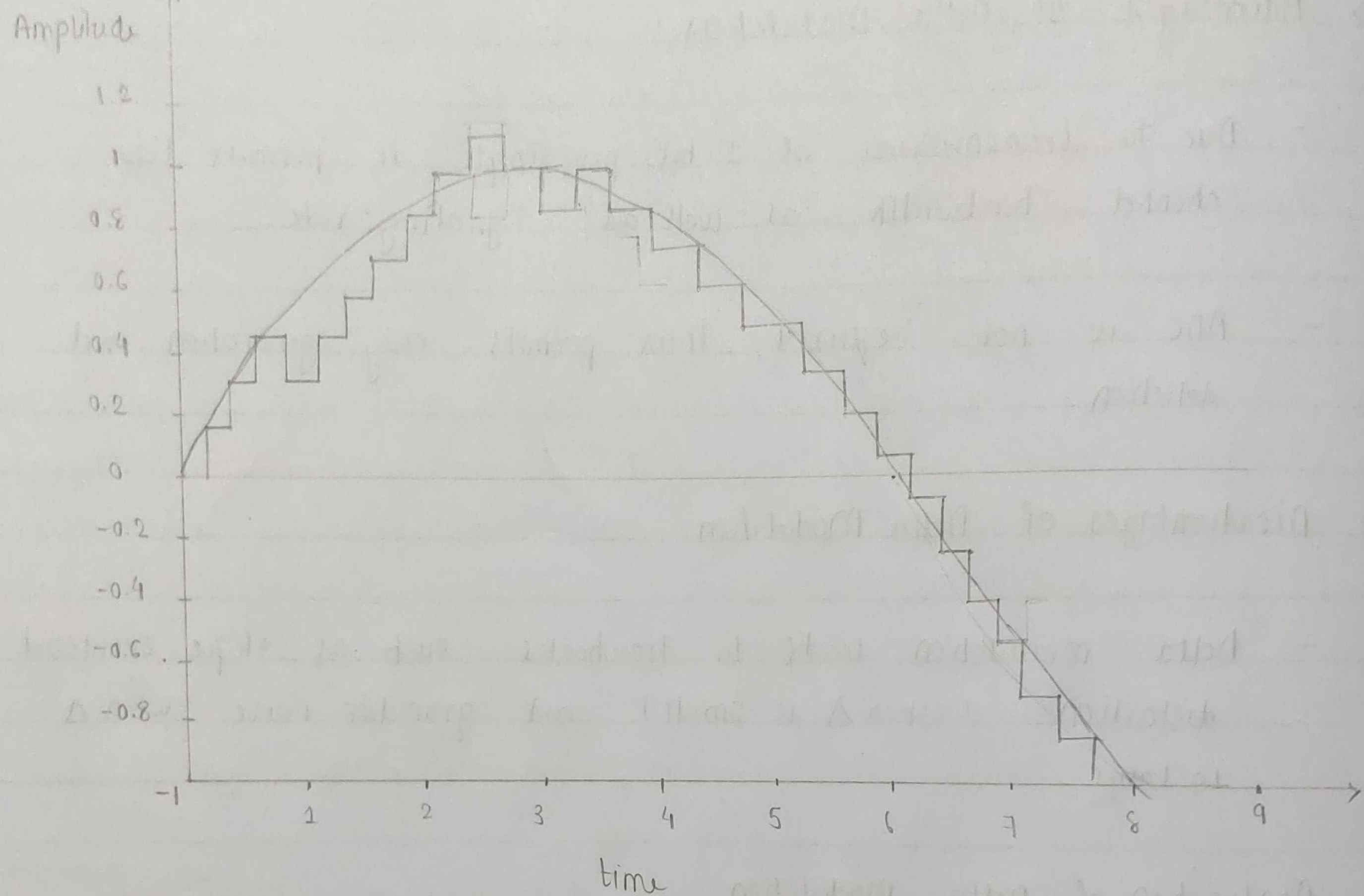
6) MATLAB CODE:

```
%% Delta Modulation (DM)
    predictor = [0 1];
    partition = [-1:.1:9];
    Step = 0.2
    partition= [0]
    codebook = [-1* Step step];
% DM    Quantizer
   t = [0: pi/20 : 2*pi];
   x = 1.1 * sin( 2*pi*0.1 *t); % Analog Signal
% Quantize x(t) using DPCM
   encoded_x = dpcmenco (x, Codebook, partition, predictor);
% Try to recover x from modulated signal
   decoded_x = dpcmdeco ( encoded_x, codebook, predictor);
% Plots
   figure
   plot(t,x);
   xlabel('time');
   title('original signal');
    figure
    Stairs (t, 10.* codebook (encodedx +1) .'g');
    Xlabel ('time');
    title ('DM output');
    figure
     plot (t,x)
    hold;
```

Teacher's Signature:

⑧

Received Signal



[U19CS012]

```
stairs( t, decoded x );
grid;
xlabel ('time');
title ('received signal');
```

CONCLUSION: We have successfully understood and demonstrand the delta modulation (DM) and demodulation technique. and also verified it with sampled, quantized/encoded and decoded time domain signal using MATLAB in virtual LAB mode.

END OF DCOM JOURNAL

SUBMITTED BY:

BHAGYA VINOD RAND [U19CS012]
[A-Div] [D-12]
C.S.E, SVNIT (2nd yr)