

# LABORATORY RECORD COMMUNICATION ENGINEERING LAB (EC 2094) AUTUMN 2020

Debagnik Kar (1804373) (ETC-06)

SCHOOL OF ELECTRONICS ENGINEERING, KIIT Deemed to be University

## Index Page

Experiment No.	Aim of the Experiment	Date of Experiment	Date of Submission	Page No.	Faculty Remarks
01	Generation and detection of Amplitude Modulation and Demodulation, DSBSC and SSB-SC modulation.	03/08/2020	17/08/2020	2-8	
02	Study of Frequency modulation and Demodulation Techniques.	10/08/2020	17/08/2020	9-11	
03	Generation and detection of PAM, PWM and PPM techniques	17/08/2020	29/08/2020	12-19	
04	Study of Pulse Code Modulation (PCM) and demodulation. Multiplexing of signal using Time Division Multiplexing (TDM) technique.				
05	Generation and detection of Delta modulation Technique				
06	(i) Study of different Data formatting techniques.  (ii) Generation and Detection of Amplitude Shift Keying (ASK)				
07	(i) Generation and Detection of Frequency Shift Keying (FSK).  (ii) Generation and Detection of Binary Phase Shift Keying (BPSK)				
08	Generation and Detection of Quadrature Phase Shift Keying (QPSK).				
09	Open Ended Experiment-1				
10	Open Ended Experiment-2				

<b>Experiment Number</b>	01
<b>Date of Experiment</b>	03/08/2020
<b>Date of Submission</b>	17/08/2020
<b>Name of the student</b>	Debagnik Kar
<b>Roll Number</b>	1804373
<b>Section</b>	ETC-06

### **Aim of The Experiment :-**

Generation and detection of Amplitude Modulation and Demodulation, DSBSC and SSB-SC modulation.

### **Equipment / Software Required:-**

- MATLAB R2018a

### **Theory:**

Amplitude modulation: AM is a modulating technique which is generally used to transmit Radio signals over a carrier signal, where the amplitude of the modulated changes in response of the signal.

Mathematically,

If the carrier signal is,

$$c(t) = A_c \cdot \cos(2\pi \cdot f_c \cdot t)$$

And the message signal is,

$$m(t) = A_m \cdot \cos(2\pi \cdot f_m \cdot t)$$

Then the equation of the modulated signal is,

$$s(t) = [A_c + \cos(2\pi \cdot f_m \cdot t)] \cdot \cos(2\pi \cdot f_c \cdot t)$$

If we solve this equation further, we get,

$$s(t) = A_c \left[ 1 + \frac{A_m}{A_c} \cdot \cos(2\pi \cdot f_m \cdot t) \right] \cdot \cos(2\pi \cdot f_c \cdot t)$$

$$s(t) = A_c [1 + \mu \cdot \cos(2\pi \cdot f_m \cdot t)] \cdot \cos(2\pi \cdot f_c \cdot t), \quad \text{Where, } \mu = \frac{A_m}{A_c}$$

Modulating Index( $\mu$ ), also known as modulation depth, of a modulation scheme describes by how much the modulated variable of the carrier signal varies around its unmodulated level. It is defined differently in each modulation scheme.

**DSBSC:** The transmission of a signal, which contains a carrier along with two sidebands can be termed as Double Sideband Full Carrier system or simply DSBSC.

Mathematically,

If the carrier Signal is,

$$c(t) = Ac. \cos(2. \pi. fc. t)$$

And the message signal is,

$$m(t) = Am. \cos (2. \pi. fm. t)$$

Then the modulating signal will be,

$$s(t) = m(t).c(t)$$

or,  $s(t) = Am. Ac. \cos(2. \pi. fm. t) . \cos(2. \pi. fc. t)$

**SSBSC:** The process of suppressing one of the sidebands along with the carrier and transmitting a single sideband is called as Single Sideband Suppressed Carrier system or simply SSBSC.

Mathematically,

If the carrier Signal is,

$$c(t) = Ac. \cos(2. \pi. fc. t)$$

And the message signal is,

$$m(t) = Am. \cos (2. \pi. fm. t)$$

Then the modulating signal will be,

$$s(t) = \frac{Am. Ac}{2} . \cos[2. \pi. (fc + fm).t] \text{ for the upper Sideband}$$

$$s(t) = \frac{Am. Ac}{2} . \cos[2. \pi. (fc - fm).t] \text{ For the lower Sideband}$$

## **Code:-**

<<<File: GenerateAM.m Comment: This code will amplitude modulate and then demodulate a sine wave.>>>

```
%generating user defined AM Signals
%Written By Debagnik Kar 1804373

clc;
clear all;
close all;

t = linspace(0,1,1000) %Time of 1 secs divided by 1000 times
%Carrier wave
fc = input('Enter fc = ');
ac = input('Enter ac = ');
xc= cos(2*pi*fc*t)

%Message signal
fm = input('Enter fm = ');
am = input('enter am = ');

xm = cos(2*pi*fm*t);

%AMplitude modulation

y = [ac + am*xm].*xc;

%plot AM
subplot(4,1,1)
plot(t,xc);
xlabel("Time -->")
ylabel("Amplitude -->")
title("Carrier Wave")
subplot(4,1,2)
plot(t,xm)
xlabel("Time -->")
ylabel("Amplitude -->")
title("Message Wave")
subplot(4,1,3)
plot(t,y)
xlabel("Time -->")
ylabel("Amplitude -->")
title("Modulated Wave")

%if else statement

mu = am/ac;

if mu==1
```

```

        disp('Critical modulation');
elseif mu>1
    disp('Over modulated signal');
elseif mu<1
    disp('under modulated signal');
end

```

```

%Demodutating The wave
dm = y.^2;
[b,a] = butter(10,0.1);
xd = filter(b,a,dm);

subplot(4,1,4)
plot(t,xd);
xlabel("Time -->")
ylabel("amplitude")
title("Demodulated Wave")

```

<<<File:DSBSC.m Comment: This code will generate a DSBSC Modulated signal>>>

```

%DSBSC Generation
%Written by Debagnik Kar 1804373
clc;
clear all;
close all;
t = linspace(0,4,1000);

fc = input('Enter the carrier frequency: ');
ac = input('Enter the carrier amplitude: ');
fm = input('Enter the message frequency: ');
am = input('Enter the message amplitude: ');

y = am*cos(2*pi*fm*t) %message signal
z = ac*cos(2*pi*fc*t) %carrier signal

w = ((am*ac)/2).*(cos(2*pi*(fc+fm)*t)+cos(2*pi*(fc-fm)*t))
%DSBSC Modulation
subplot(3,1,1)
plot(t,z)
xlabel("time -->")
ylabel("magnitide -->")
title("Carrier Signal")
subplot(3,1,2)
plot(t,y)
xlabel("time -->")
ylabel("magnitide -->")
title("Message Signal")
subplot(3,1,3)
plot(t,w)
xlabel("time -->")
ylabel("magnitide -->")

```

```

title("DSBSC Signal")

<<<File: SSBSC.m Comment: Generates an upper sideband, a lower side band SSBSC>>>

% Generation of SSB-SC Signal
% Written by Debagnik Kar
clear all
close all
clc

fc = input('Enter the frequency of Carrier: ');
ac = input('Enter the amplitude of Carrier: ');
fm = input('Enter the frequency of Message: ');
am = input('Enter the amplitude of Message: ');

t = linspace(0,1,1000)

m = am*cos(2*pi*fm*t) %Message signal
c = ac*cos(2*pi*fc*t) %carrier Signal

Susb = ((am*ac)/2).*cos(2*pi*(fc+fm)*t) %upper Sideband
Slsb = ((am*ac)/2).*cos(2*pi*(fc-fm)*t) %lower Sideband

subplot(4,1,1)
plot(t,c,'r')
xlabel("Time -->")
ylabel("Amplitude-->")
title("Carrier Wave")

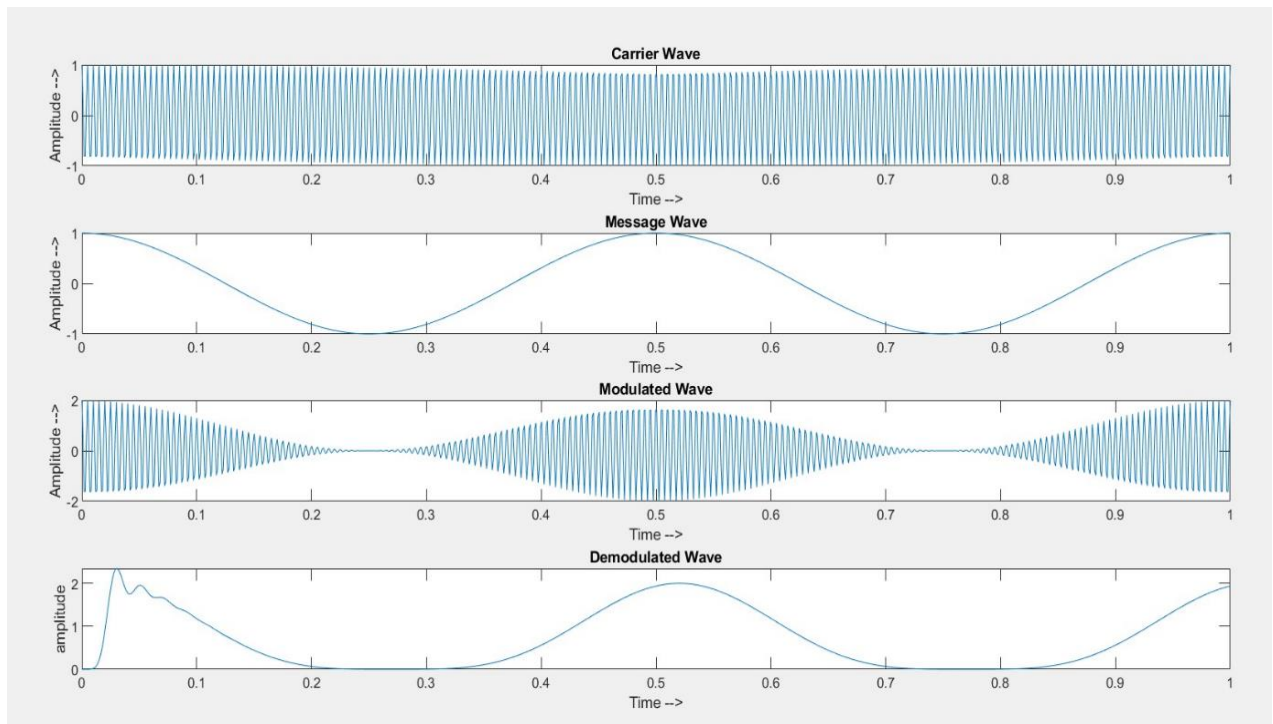
subplot(4,1,2)
plot(t,m,'g')
xlabel("Time -->")
ylabel("Amplitude-->")
title("Message Wave")

subplot(4,1,3)
plot(t,Susb,'b')
xlabel("Time -->")
ylabel("Amplitude-->")
title("Upper Sideband SSB-SC Signal")

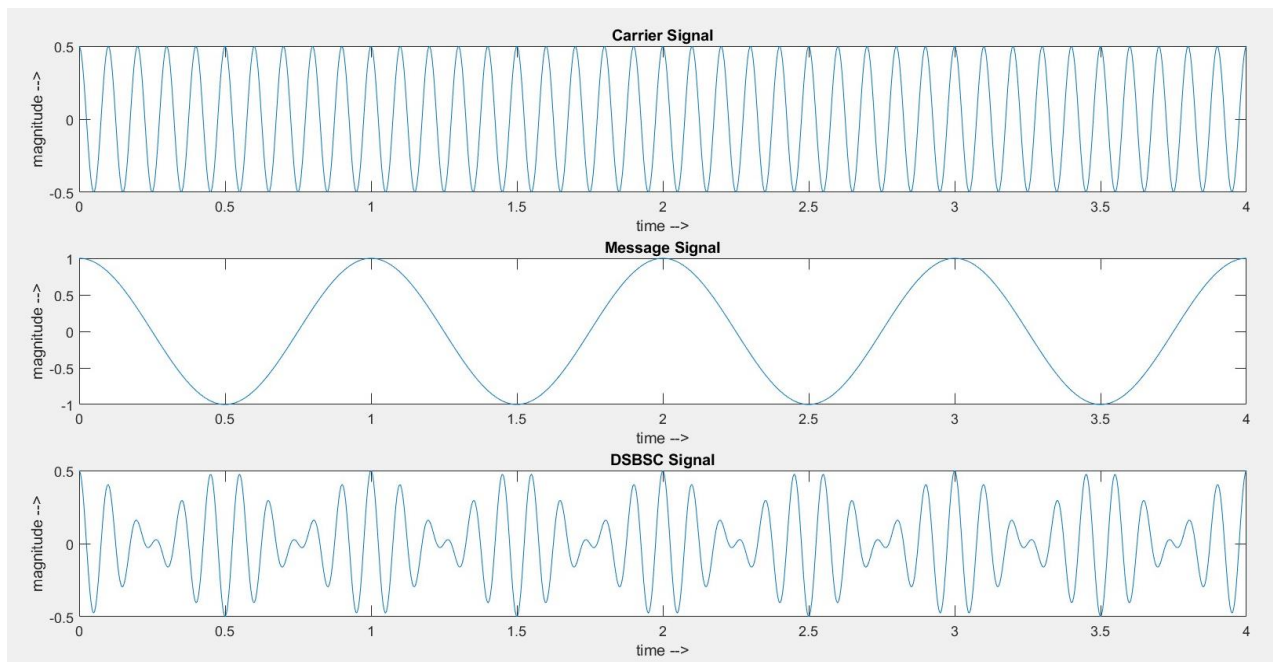
subplot(4,1,4)
plot(t,Slsb,'k')
xlabel("Time -->")
ylabel("Amplitude-->")
title("Lower Sideband SSB-SC Signal")

```

## Output/Graph:-

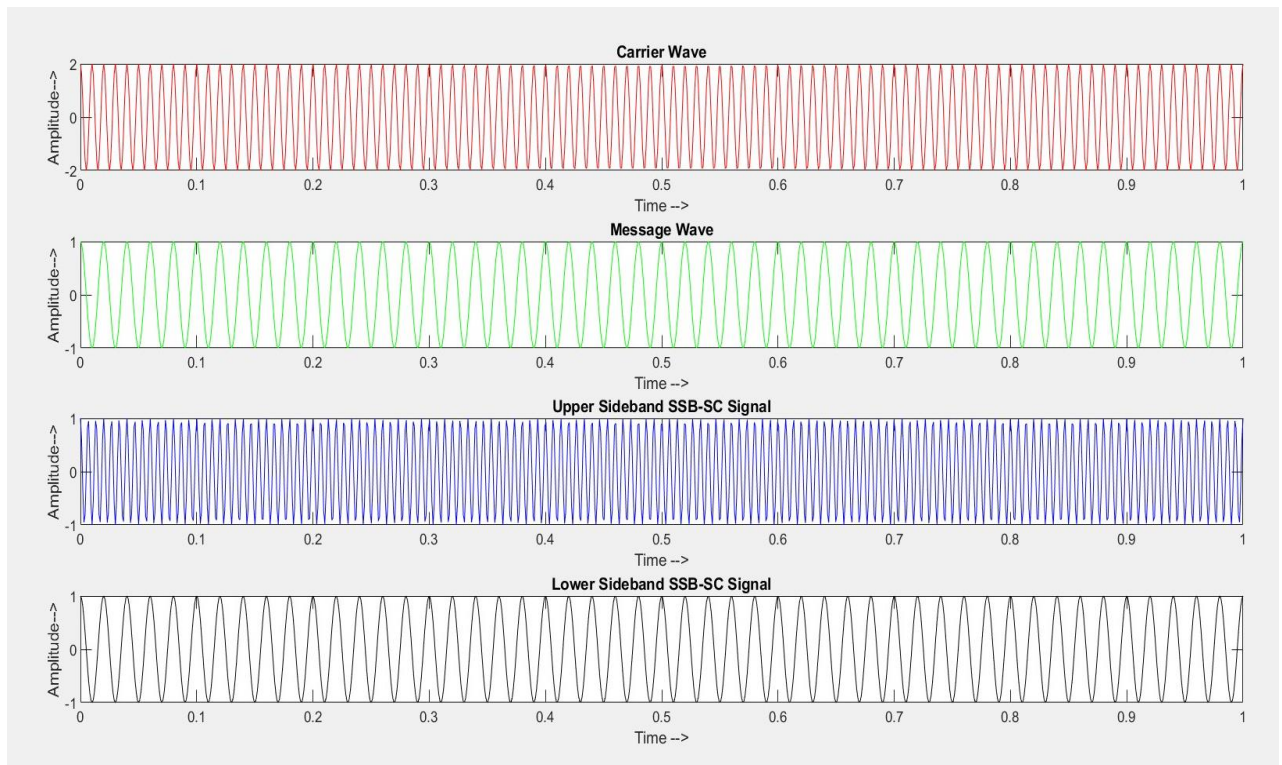


*Fig 1.1 : Amplitude modulating and Demodulating a Sine wave of 2Hz*



*Fig 1.2: Generation of DSBSC Signal*





*Fig 1.3: Generation of SSBSC Signal*

### **Discussion or Inference of the experiment**

This experiment taught me about different kind of amplitude modulation and demodulation techniques that is practiced in analog radio communication technology. It also helped me visualize the difference between the different techniques of transmission

### **Conclusion:-**

The simulation of experiment is done successfully using MATLAB Software.

<b>Experiment Number</b>	02
<b>Date of Experiment</b>	10/08/2020
<b>Date of Submission</b>	17/08/2020
<b>Name of the student</b>	Debagnik Kar
<b>Roll Number</b>	1804373
<b>Section</b>	ETC-06

### **Aim of The Experiment :-**

Study of Frequency modulation and Demodulation Techniques.

### **Equipment / Software Required:-**

- MATLAB R2018a

### **Theory**

Frequency Modulation (FM) is a form of modulation in which changes in the carrier wave frequency correspond directly to changes in the baseband signal.

Mathematically,

If the Carrier signal is

$$c(t) = \cos(2.\pi. f_c. t)$$

And the message signal is

$$m(t) = \sin(2.\pi. f_m. t)$$

Then the modulating signal will be,

$$s(t) = \cos[2.\pi. f_c. t - \{\mu. \sin(2.\pi. f_m. t)\}]$$

$$\text{Where } \mu = \frac{A_m}{A_c}$$

### **Code:-**

<<<File:FMModDemod.m Comment: This code generates a FM signal and Demodulates it>>>

```
%Generating a FM Signal and Demodulating it
%Written by Debagnik Kar 1804373

clc;
clear all;
close all;

fc=input('Enter the carrier signal: ');
fm=input('Enter the message signal: ');
mu=input('Modulation index ');

t=linspace(0,1,1000);

c=cos(2*pi*fc*t);%carrier signal
m=sin(2*pi*fm*t);%message signal

subplot(4,1,1);
plot(t,c,'r'); %plotting the carrier signal
ylabel('amplitude');
xlabel('time');
title('Carrier signal');

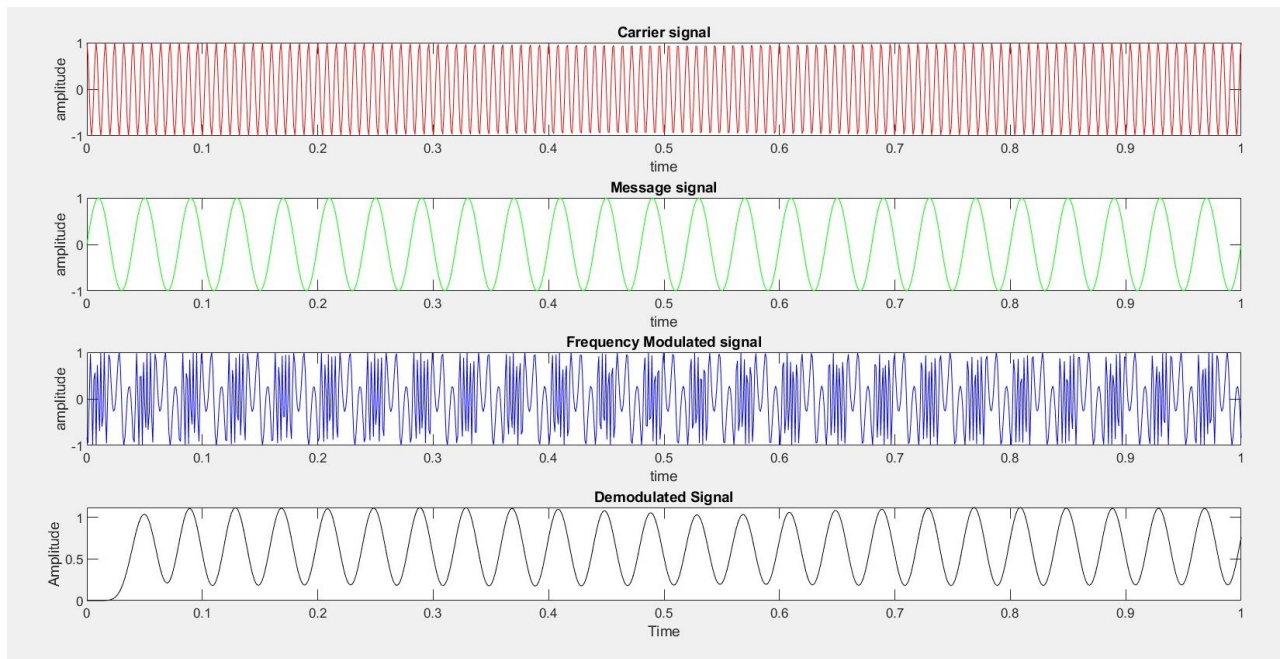
subplot(4,1,2);
plot(t,m,'g'); %plotting the message signal
ylabel('amplitude');
xlabel('time');
title('Message signal');

y=cos(2*pi*fc*t-(mu*cos(2*pi*fm*t))); % FM Generation

subplot(4,1,3);
plot(t,y,'b'); % Plotting the FM Generation
ylabel('amplitude');
xlabel('time');
title('Frequency Modulated signal');

%FM Demodulation
dem = diff(y);
dem = [0,dem];
rect_dem = abs(dem)
b,a]=butter(10,0.06);
rec = filter(b,a,rect_dem);
subplot(4,1,4)
plot(t,rec,'k')
xlabel('Time')
ylabel('Amplitude')
title('Demodulated Signal')
```

### **Output/Graph:-**



*Fig 2.1: Generation of FM Signal and demodulating it.*

### **Discussion or Inference of the experiment**

This experiment taught me techniques of radio transmission that are currently being practiced. It also taught me the reasons that FM is better than AM in radio signal transmissions. I also learnt to demodulate a FM signals.

### **Conclusion:-**

Simulation of experiment is done successfully

<b>Experiment Number</b>	03
<b>Date of Experiment</b>	24/08/2020
<b>Date of Submission</b>	29/08/2020
<b>Name of the student</b>	Debagnik Kar
<b>Roll Number</b>	1804373
<b>Section</b>	ETC - 06

### **Aim of The Experiment :-**

Generation and detection of PAM, PWM and PPM techniques

### **Equipment / Software Required:-**

MATLAB R2018a

### **Theory**

Pulse Amplitude Modulation (PAM): is an analog modulating scheme in which the amplitude of the pulse carrier waves varies proportionally to the instantaneous amplitude of the message signal. There are two types of PAM 1. Natural PAM 2. Flat top PAM.

1. Natural PAM: a signalled at Nyquist rate is reconstructed, by passing it through an efficient Low pass Frequency with exact cut-off frequency.
2. Flat top PAM: Although the natural PAM signal is passed through an LPF, it cannot recover the signal without distortion. Hence to avoid this noise, flat-top sampling is done. Flat-top sampling is the process in which sampled signal can be represented in pulses for which the amplitude of the signal cannot be changed with respect to the analog signal, to be sampled. The tops of amplitude remain flat. This process simplifies the circuit design.

Pulse Width Modulation (PWM): is an analog modulating scheme in which the duration or width or time of the pulse carrier varies proportional to the instantaneous amplitude of the message signal.

The width of the pulse varies in this method, but the amplitude of the signal remains constant. Amplitude limiters are used to make the amplitude of the signal constant. These circuits clip off the amplitude, to a desired level and hence the noise is limited. It is mainly used in the semiconductor industries to transmit signals between microcontrollers or actuators.

Pulse Position Modulation: is an analog modulating scheme in which the amplitude and width of the pulses are kept constant, while the position of each pulse, with reference to the position of a reference pulse varies according to the instantaneous sampled value of the message signal.

The transmitter has to send synchronizing pulses (or simply sync pulses) to keep the transmitter and receiver in synchronism. These sync pulses help maintain the position of the pulses. The following figures explain the Pulse Position Modulation.

<b>PAM</b>	<b>PWM</b>	<b>PPM</b>
Amplitude is varied	Width is varied	Position is varied
Bandwidth depends on the width of the pulse	Bandwidth depends on the rise time of the pulse	Bandwidth depends on the rise time of the pulse
Instantaneous transmitter power varies with	Instantaneous transmitter power varies with the amplitude and the width of the pulse	Instantaneous transmitter power remains constant with the width of the pulses
System complexity is high	System complexity is low	System complexity is low
Noise interference is high	Noise interference is low	Noise interference is low
It is similar to amplitude modulation	It is similar to amplitude modulation	It is similar to phase modulation

*Table 3.1: Comparison of different types of modulations*

Demodulation: For demodulating all the signals. A Butterworth filter is used and the specifications of it are given below:

- PAM:
- PWM:

### Code:-

<<< File: PulseAmpMod.m Comment: Generated a PAM signal and then Demodulates it. Also asks the user if he/she wants to continue to generate a PPM Signal from the PWM signal.>>>

```
%PULSE AMPLITUDE MODULATION PAM
%Written by Debagnik Kar 1804373
clc
close all
clear all

t = 1:0.0001:2
f = input('Enter the value of frequency: ')

x=sawtooth(2*pi*f*t)
ts=0.02
%PULSE GENERATION
for k=1:length(t)
    if mod(t(1,k),ts)==0
        pulse(1,k)=1 %PULSE
    else
        pulse(1,k)=0;
    end
end
natural_pam = x.*pulse %Natural Pulse amplitude modulation

subplot(5,1,1)
plot(t,x,'r')
xlabel('Time -->')
ylabel('Amplitude -->')
title('Message')
subplot(5,1,2)
plot(t,pulse,'b')
xlabel('Time -->')
ylabel('Amplitude -->')
title('Pulse Signal')
subplot(5,1,3)
plot(t,natural_pam,'g')
xlabel('Time -->')
ylabel('Amplitude -->')
title('Natural PAM')

%FLATTOP PAM
FlatTop_pam =zeros(1,length(t));
k=1;

while k <length(t)
    if natural_pam(1,k)~=0
        FlatTop_pam(1,k:k+49)=natural_pam(1,k)*ones(1,50); % pulse
        duration is 50*0.001
        k=k+49;
    end
end
```

```

else FlatTop_pam(1,k)=0;
    k=k+1;
end
end

subplot(5,1,4)
plot(t,FlatTop_pam,'k')
xlabel('Time -->')
ylabel('Amplitude -->')
title('Flat top PAM')

%Demodulation
[b,a]=butter(7,0.004)
demod=filter(b,a,FlatTop_pam)
subplot(5,1,5)
plot(t,demod,'c')
xlabel('Time -->')
ylabel('Amplitude -->')
title('Demodulated PAM')

```

<<<File: PulseWidthMod.m Comment: Generation and demodulation of PWM>>>

```

%Generation of PWM
%Written by Debagnik Kar 1804373

clc
clear all
close all

fc = input('Carrier frequency in Hz: ') %user input prompt
fm = input('Message Frequency in Hz: ') %user input prompt

t = 0:0.001:1 % TIME

carrier = sawtooth(2*pi*fc*t)

subplot 411
plot(t,carrier)
title('Carrier wave')
xlabel('Time')
ylabel('Amplitude')

% Modulating waveform
message = cos(2*pi*fm*t)
subplot 412
plot(t,message)
title('Message wave')
xlabel('Time')
ylabel('Amplitude')

% PWM waveform generation

```



```

p=find(carrier>=message)
    pwm(p) = -1
q=find(carrier<message)
    pwm(q) = 1

subplot 413
plot(t,pwm)
axis([0 1 0 2])
title('PWM waveform')
xlabel('Time')
ylabel('Amplitude')

[b, a] = butter(6,0.009)
demod = filter(b,a,pwm)
subplot 414
plot(t,demod,'r')
xlabel('Time -->')
ylabel('Amplitude -->')
title('Demodulated Wave')

prompt = input('Do you want to continue to PPM? (1 - yes/0 -
no): ')
if prompt == 1
    PulsePosMod(t,pwm)
elseif prompt == 0
    disp('Okay!')
    exit()
else
    disp('Wrong choice, Exiting')
    goto(46)
end

```

<<<File: PulsePosMod.m Comment: Extension of PulseWidthMod.m , It generates a PPM Signal from the PWM from the above signal>>>

```

%PPM Extension Funtion to PWM
%Written By Debagnik Kar

function PulsePosMod(t,pwm,f)
clc
figure(2)
dip=diff(pwm)
dip = [0,dip]
ppm=zeros(1,length(dip))
k=1
%Positive edge
while k<length(dip)
    if dip(1,k) == 2 % take -2 for negative edge triggering
        ppm(1,k:k+9)=ones(1,10) % I took to block
        k=k+10
    else

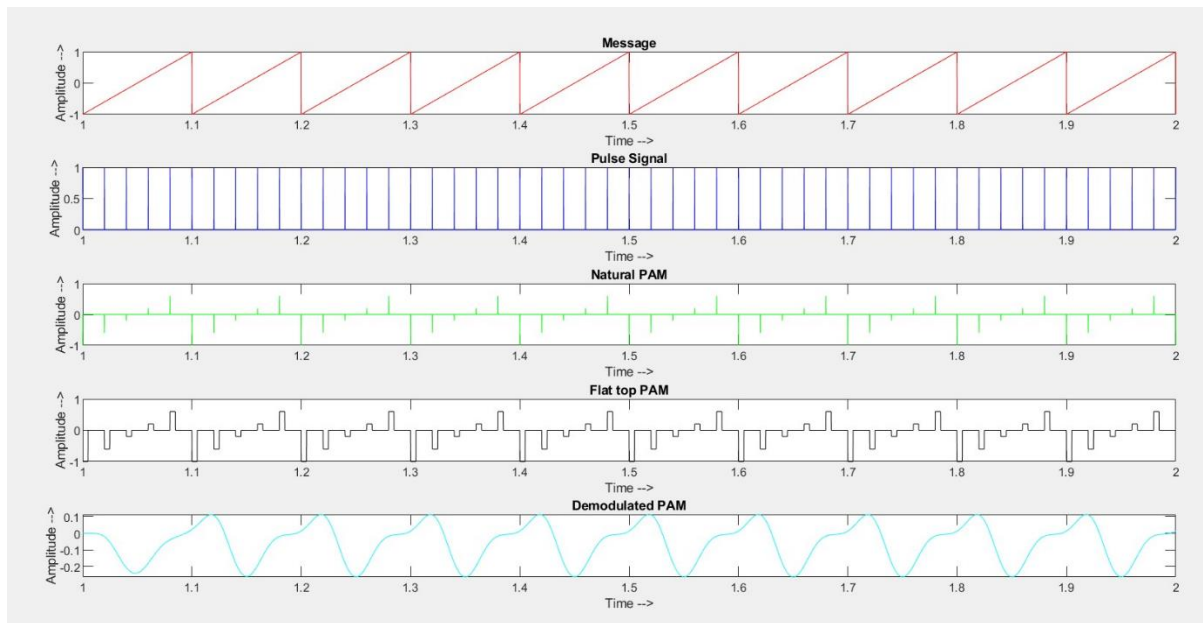
```

```

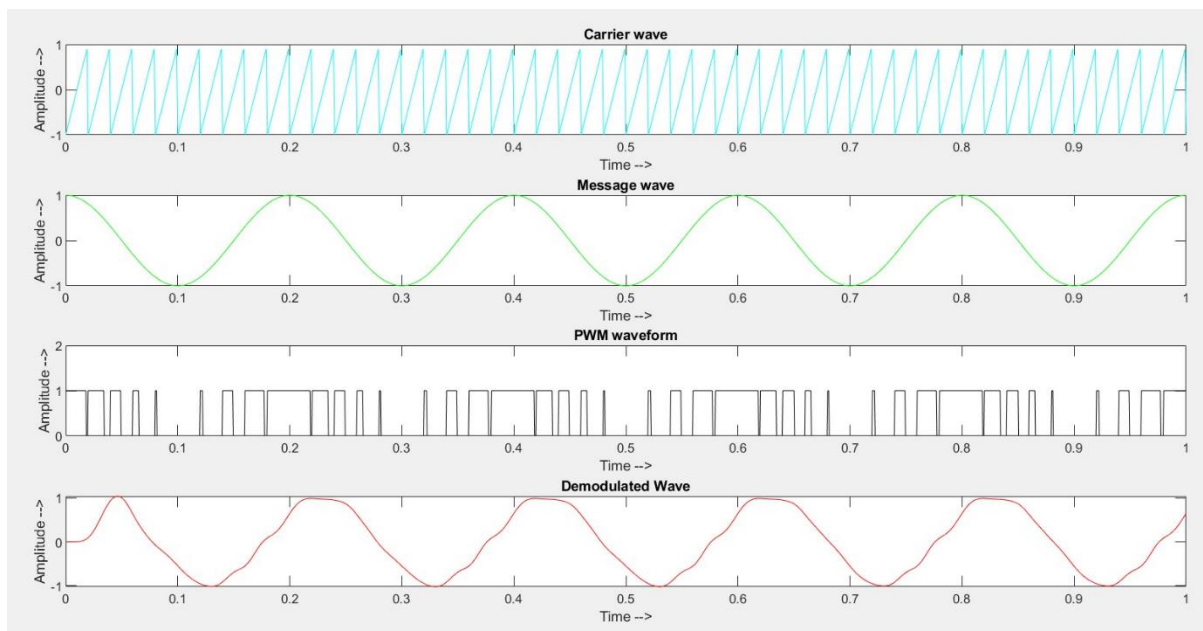
        k=k+1
    end
end
subplot 311
plot(t,cos(2*pi*f*t),'r')
xlabel('Time -->')
ylabel('Amplitude -->')
title('Message signal')
subplot 312
plot(t,pwm,'c')
axis([0 1 0 2])
xlabel('Time -->')
ylabel('Amplitude -->')
title('PWM Signal')
subplot 313
plot(t,ppm,'g')
axis([0 1 0 2])
xlabel('Time -->')
ylabel('Amplitude -->')
title('PPM Signal')
end

```

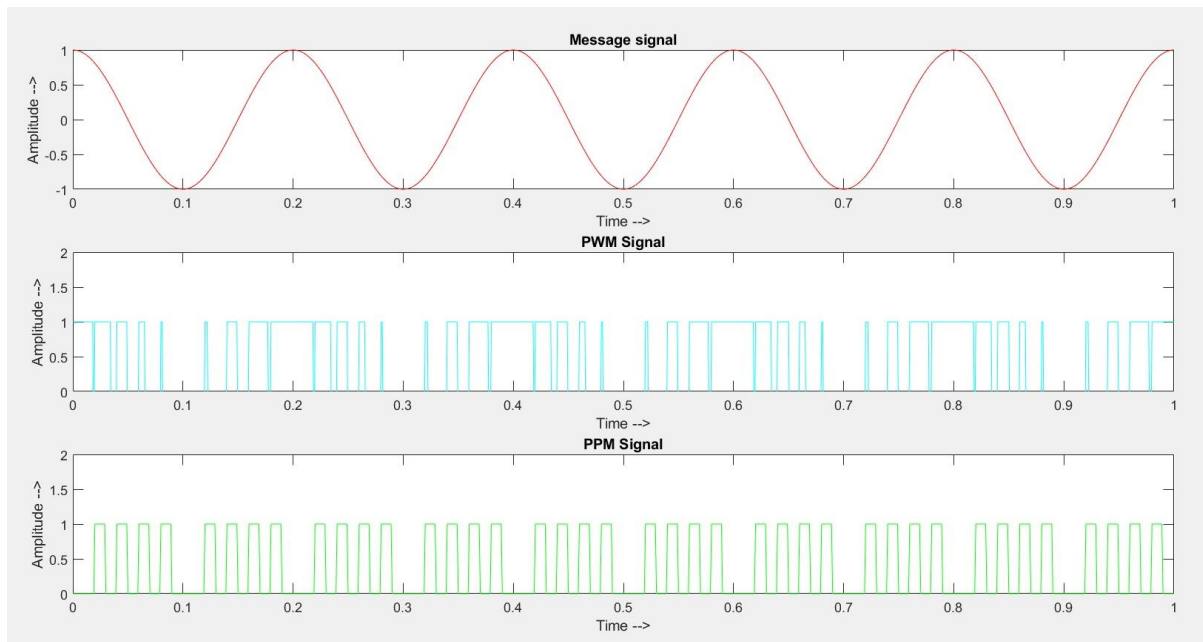
## Output/Graph:-



*Fig 3.1: Modulation and Demodulation of Natural PAM and Flat-top PAM.*



*Fig 3.2: Modulation and Demodulation of PWM Signal.*



*Fig 3.3: Modulation of PPM.*

### **Discussion or Inference of the experiment**

This Experiment taught me different pulse modulation techniques such as PAM, PWM, PPM. I also realized advantages of Pulse Position Modulation, over Pulse Amplitude Modulation or Pulse Width Modulation.

By understanding, Pulse position modulation has low noise interference when compared to PAM because amplitude and width of the pulses are made constant during modulation. Noise removal and separation is also very easy in pulse position modulation. I also learned about differentiator and inverter circuit, and its use during generation of PPM signal, through PWM signal.

### **Conclusion:-**

The simulation of PAM, PWM and PAM was done successfully using MATLAB Software.