NAME: Siddharth Bose

REGISTRATION NO.: 19BEC1278
EXP. NO: 1
DATE : 11/01/2022

AMPLITUDE MODULATION AND DEMODULATION

AIM: Write a MATLAB program to execute and display the output for Amplitude modulation and demodulation.

SOFTWARE REQUIRED: MATLAB

THEORY:

Amplitude modulation is a process by which the wave signal is transmitted by modulating the amplitude of the signal. The instantaneous value of the carrier amplitude changes in accordance with the amplitude and frequency variations of the modulating signal. The carrier signal frequency is greater than the modulation signal frequency.

If modulation signal $\mathbf{m(t)=A_m cos(2\pi f_m t)}$ and carrier signal $\mathbf{c(t)=A_c cos(2\pi f_c t)}$, then modulated signal $\mathbf{s(t)=[A_c+A_m cos(2\pi f_m t)]\ cos(2\pi f_c t)}$ where,

A_m: Amplitude of modulation signal
f_m: Frequency of modulation signal
A_c: Amplitude of carrier signal
f_c: Frequency of message signal

In amplitude modulation, it is particularly important that the peak value of the modulating signal be less than the peak value of the carrier. The relation between the amplitude of modulation signal and amplitude of carrier signal is given by

$\mathbf{m=A_m/A_c}$

where m is known as the modulation index.

The process of extracting the message signal from the modulated wave is known as demodulation. The circuit, which demodulates the modulated wave is known as the demodulator.

ALGORITHM:

Step 1: Define the values for Am (message signal amplitude), fm (message signal frequency), Ac (carrier signal amplitude), fc (carrier signal frequency) and m (modulation index).

Step 2: Use the equation

- m=Am*sin(2*pi*fm*t) to define the message signal
- C=Ac*sin(2*pi*fc*t) to define the carrier signal
- s=Ac*(1+ma*m).*sin(2*pi*fc*t) to define the modulated signal

Step 3: Use the stem() command to obtain the spectrum of modulated signal and the amdemod() command to obtain the demodulated signal.

Step 4: Plot the message signal, carrier signal, modulated signal, spectrum of modulated signal and the demodulated signal in a single window using the subplot() command.

Step 5: Finally, save and click on Run to obtain the output graphs.

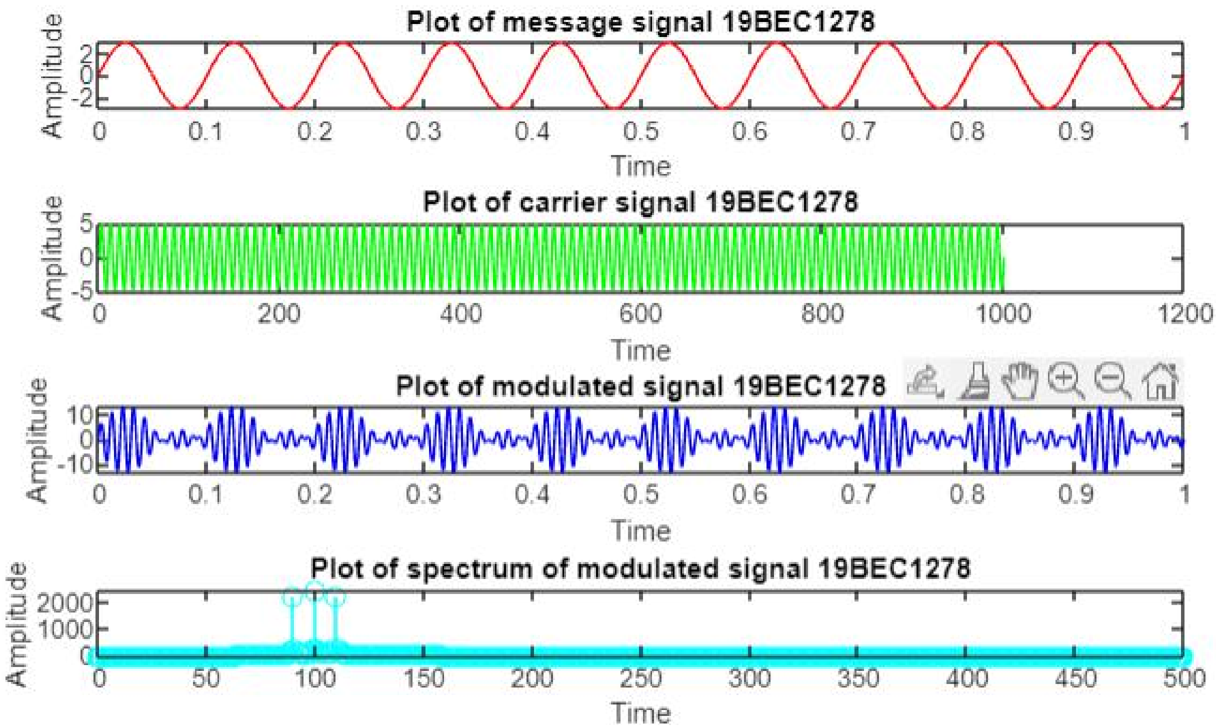
MATLAB CODE:

```
%% AM modulation and frequency spectrum
%message frequency=10Hz
%carrier frequency as 100Hz
clc;
clear;
close all;
Am=3;
Ac=5;
ma=Am/Ac;
fm=10;
fc=100;
Ts=1/1000
```

```
t=0:1/1000:1;

m=Am*sin(2*pi*fm*t);

subplot(5,1,1)

%plot(t,m)

plot(t,m,'r')

xlabel('Time')

ylabel('Amplitude')

title('Plot of message signal 19BEC1278')


%message spectrum

N=length(t);

y=fft(m,N);

z=y(1:floor(N/2)+1);

k=0:floor(N/2);

% figure,stem(k,abs(z))


%carrier signal

C=Ac*sin(2*pi*fc*t)

subplot(5,1,2)

% figure,plot(t,C)

plot(C,'g')

xlabel('Time')

ylabel('Amplitude')

title('Plot of carrier signal 19BEC1278')


%modulated signal

s=Ac*(1+ma*m).*sin(2*pi*fc*t); %s=Ac(1+mam(t)*sin2pifct)

subplot(5,1,3)

plot(t,s,'b')
```

```
%plot(msg)
xlabel('Time')
ylabel('Amplitude')
title('Plot of modulated signal 19BEC1278')
%figure,plot(t,s)

%spectrum of modulated signal
y1=fft(s,N);
z1=y1(1:floor(N/2)+1);
k1=0:floor(N/2);
subplot(5,1,4)
%figure,plot(t,s)
%plot(msg,'b')
stem(k1,abs(z1),'c')
xlabel('Time')
ylabel('Amplitude')
title('Plot of spectrum of modulated signal 19BEC1278')

%%Amplitude demodulation using inbuilt commandFs=1000;
Z=amdemod(s,fc,Fs);
subplot(5,1,5)
plot(t,Z,'k')
xlabel('Time')
ylabel('Amplitude')
title('Plot of demodulated signal using inbuilt matlab command 19BEC1278')
```

OUTPUT:



INFERENCE:

The plot of AM modulated signal was generated and it is observed that the modulated wave has an envelope which is the same as the of message signal. After demodulation, we have obtained the original message signal. Also, the spectrum was plotted where carrier and sidebands are visible.

NAME: Siddharth Bose

REGISTRATION NO.: 19BEC1278
EXP. NO: 2
DATE : 25/01/2022

DSB-SC MODULATION AND DEMODULATION

AIM: Write a MATLAB program to execute and display the output for DSB-SC modulation and demodulation.

SOFTWARE REQUIRED: MATLAB

THEORY:
Double Sideband Suppressed Carrier Modulation or DSB-SC is an amplitude modulated wave transmission scheme in which only sidebands are transmitted and the carrier is not transmitted as it gets suppressed. The carrier does not contain any information and its transmission results in loss of power. Thus, only sidebands are transmitted that contains information. This results in saving of power used in transmission.

If modulation signal $\mathbf{m(t)=A_m cos(2\pi f_m t)}$ and carrier signal $\mathbf{c(t)=A_c cos(2\pi f_c t)}$, then modulated signal $\mathbf{s(t)= m(t).c(t) = A_m A_c\ cos(2\pi f_m t)\ cos(2\pi f_c t))}$ where,

A_m: Amplitude of modulation signal
f_m: Frequency of modulation signal
A_c: Amplitude of carrier signal
f_c: Frequency of message signal

ALGORITHM:
Step 1: Define the values for A (message and carrier signal amplitude), fm (message signal frequency), fc (carrier signal frequency).
Step 2: Use the equation

- msg=A*sin(2*pi*fm*t) to define the message signal
- car=A*sin(2*pi*fc*t) to define the carrier signal
- mod=msg.*car to define the modulated signal

Step 3: Use the stem() command to obtain the spectrum of modulated signal and the amdemod() command to obtain the demodulated signal.

Step 4: Plot the message signal, carrier signal, modulated signal, spectrum of modulated signal and the demodulated signal in a single window using the subplot() command.

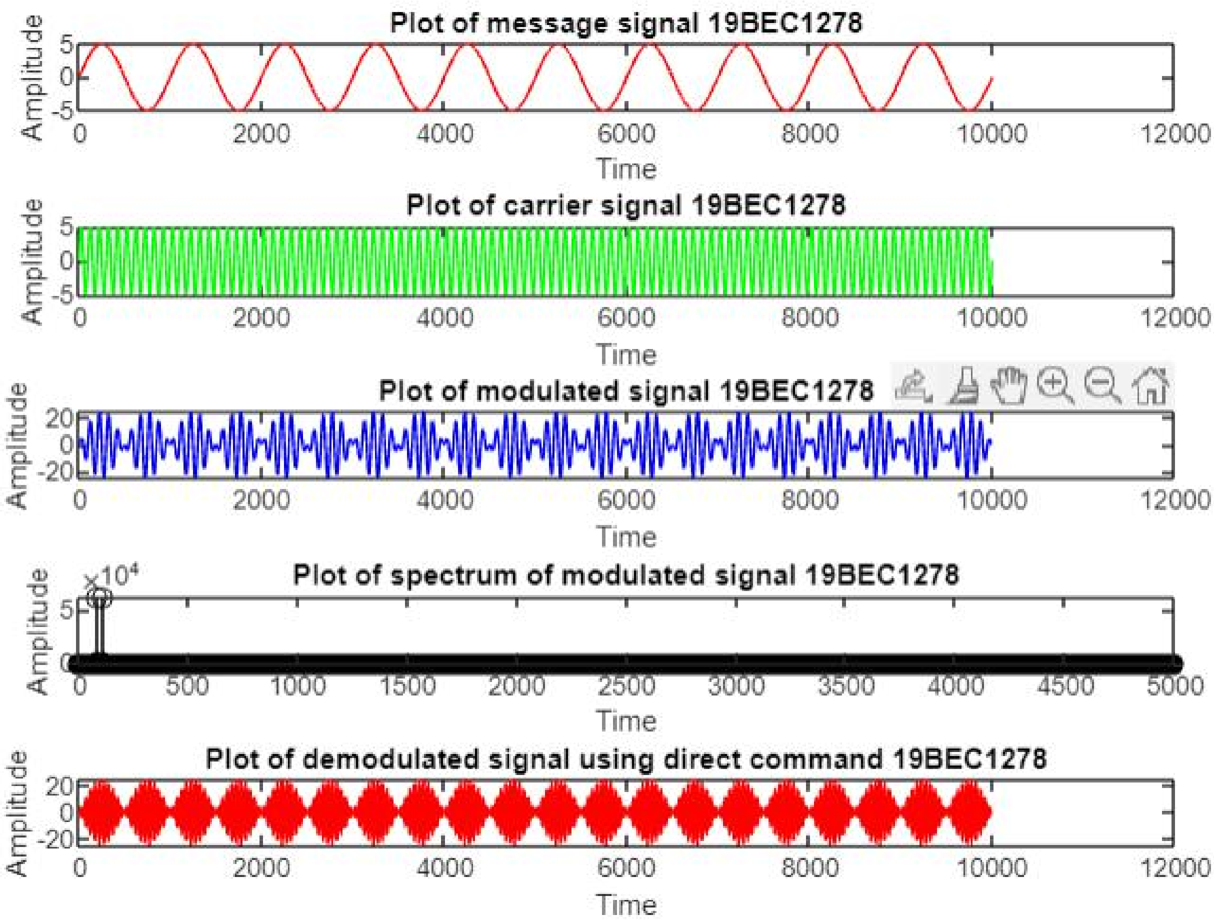
Step 5: Finally, save and click on Run to obtain the output graphs.

MATLAB CODE:

```
clc;

clear all;

close all;


%%Message signal

t=0:0.0001:1;

fm=10;

A=5;

msg=A*sin(2*pi*fm*t);

subplot(5,1,1)

plot(msg,'r')

xlabel('Time')

ylabel('Amplitude')

title('Plot of message signal 19BEC1278')


%%Carrier signal

fc=100;

car=A*sin(2*pi*fc*t)

subplot(5,1,2)

plot(car,'g')

xlabel('Time')

ylabel('Amplitude')

title('Plot of carrier signal 19BEC1278')
```

```
%%Modulated signal

mod=msg.*car

subplot(5,1,3)

plot(mod,'b')

xlabel('Time')

ylabel('Amplitude')

title('Plot of modulated signal 19BEC1278')


%%Spectrum of modulated signal

N=length(t);

w=fft(mod,N);

q=w(1:floor(N/2)+1);

k=0:floor(N/2);

subplot(5,1,4);

stem(k,abs(q),'k')

title('Plot of spectrum of modulated signal 19BEC1278')

xlabel('Time')

ylabel('Amplitude')


%%Demodulation using inbuilt command

Fs=1000;

Z= amdemod(mod,fc,Fs);

subplot(5,1,5);

plot(Z,'r');

xlabel('Time')

ylabel('Amplitude')

title('Plot of demodulated signal using direct command 19BEC1278')
```

OUTPUT:



INFERENCE:

The plot of AM modulated signal was generated and it is observed that the modulated wave has an envelope which is the same as the of message signal. After demodulation, we have obtained the original message signal. Also, the spectrum was plotted where the carrier was suppressed and only the sidebands are visible.

NAME: Siddharth Bose

REGISTRATION NO.: 19BEC1278
DATE : 25/01/2022
EXP. NO: 3

DATE 1 : 01/02/2022

FREQUENCY MODULATION AND DEMODULATION

AIM : Write a MATLAB program to execute and display the output for Frequency modulation and demodulation.

SOFTWARE REQUIRED: MATLAB

THEORY:

Frequency Modulation is a modulation in which the frequency of the carrier wave is altered in accordance with the instantaneous amplitude of the modulating signal, keeping phase and amplitude constant. Modification of carrier wave frequency is performed for the purpose of sending data or information over small distances. In the case of FM, the amplitude of the modulated signal is kept or it remains constant.

If modulation signal $\mathbf{m(t)=A_m cos(2\pi f_m t)}$ and carrier signal $\mathbf{c(t)=A_c cos(2\pi f_c t)}$, then modulated signal $\mathbf{f_m(t) = cos(2*pi*f_c*t+ m(t))}$ where,

A_m: Amplitude of modulation signal
f_m: Frequency of modulation signal
A_c: Amplitude of carrier signal
f_c: Frequency of message signal

ALGORITHM:

Step 1: Define the values for Am (message signal amplitude), fm (message signal frequency), Ac (carrier signal amplitude), fc (carrier signal frequency).

Step 2: Use the equation

- msg=Am*cos(2*pi*fm*t) to define the message signal
- carrier=Ac*cos(2*pi*fc*t); to define the carrier signal
- FM=cos(2*pi*fc*t+ msg) to define the modulated signal

Step 3: Plot the message signal, carrier signal and the modulated signal in a single window using the subplot() command.

Step 4: Use fmmod() and the fmdemod() command to obtain the frequency modulated and demodulated signal respectively.

Step 5: Plot the message signal, modulated signal and the demodulated signal in a single window using the subplot() command.

Step 6: Finally, save and click on Run to obtain the output graphs.

MATLAB CODE:

USING THEORETICAL FORMULA

```
clc

clear all

%Message Signal

Am=5;

t=0:0.001:1;

fm=10;

msg=Am*cos(2*pi*fm*t);

subplot(3,1,1); plot(t,msg)

title('Message signal 19BEC1278');

xlabel('time')

ylabel('amplitude')


%Carrier SignalAc=5;

fc=80;

carrier=Ac*cos(2*pi*fc*t);

subplot(3, 1,2);

plot(t,carrier)

xlabel('Time')
```

```
ylabel('Amplitude')

title('Carrier signal 19BEC1278')


%Modulated Signal

FM=cos(2*pi*fc*t+ msg);

subplot(3,1,3);

plot(t,FM,'r');

title('Modulated Signal 19BEC1278')

xlabel('time')

ylabel('Amplitude')
```

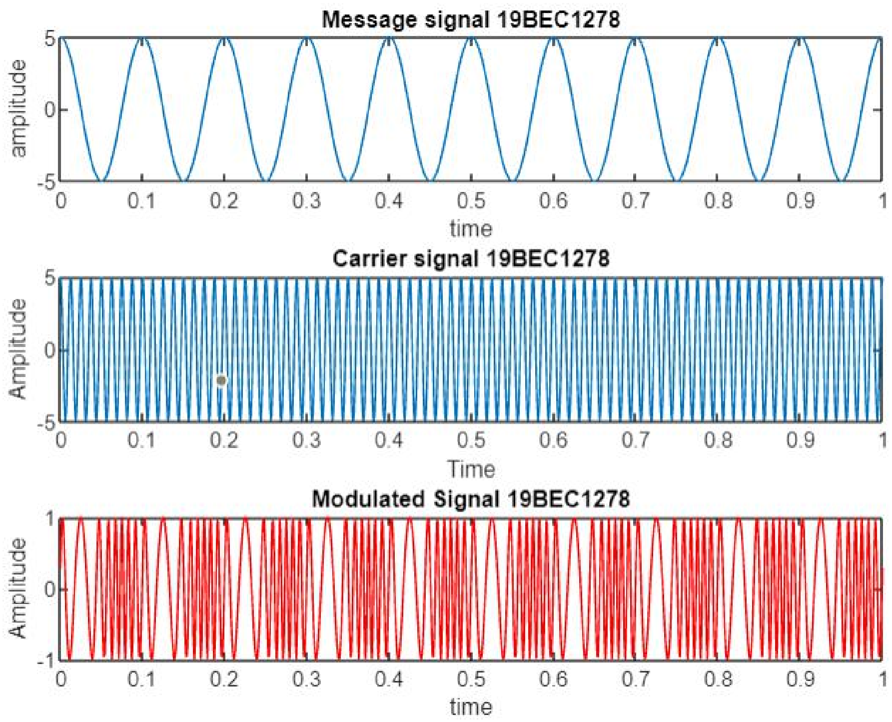
USING IN BUILT COMMAND

```
clc

clear all

t=0:0.0001:0.1;

fm=20;

fc=100;


%Message Signal

x=12;

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

subplot(3,1,1), plot(t,m)

xlabel('Time');

ylabel('Amplitude');

title('Message Signal 19BEC1278');


%%Frequency modulation using in built command

y1=fmmod(m,fc,800,50);

subplot(3,1,2), plot(t,y1)
```

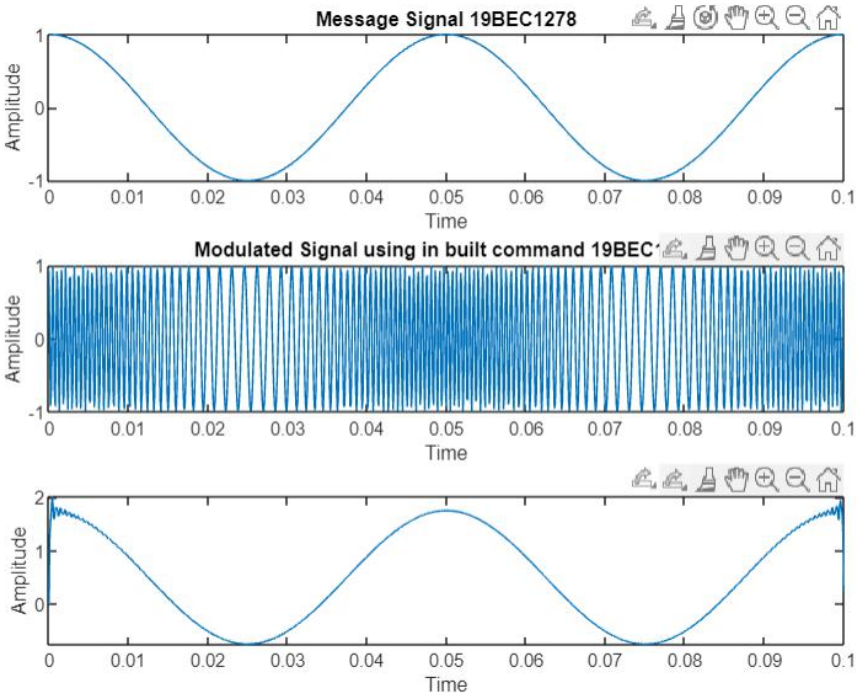
```
xlabel('Time');

ylabel('Amplitude');

title('Modulated Signal using in built command 19BEC1278');


%%Frequency demodulation using in built command

y2=fmdemod(y1,fc,1000,50);

subplot(3,1,3), plot(t,y2)

xlabel('Time');

ylabel('Amplitude');

title('Demodulated Signal using in built command 19BEC1278');
```

OUTPUT:

USING THEORETICAL FORMULA



USING IN BUILT COMMAND



INFERENCE:

The FM modulated and demodulated signal was generated using theoretical formulae and in built commands.

NAME: Siddharth Bose

REGISTRATION NO.: 19BEC1278
CHALLENGING EXERCISE
DATE : 01/02/2022

AIM: Simulate Amplitude modulation and DSBSC modulation using Simulink in MATLAB.

SOFTWARE REQUIRED: MATLAB

THEORY:
Amplitude modulation is a process by which the wave signal is transmitted by modulating the amplitude of the signal. The instantaneous value of the carrier amplitude changes in accordance with the amplitude and frequency variations of the modulating signal. The carrier signal frequency is greater than the modulation signal frequency.

If modulation signal **m(t)=Amcos(2πfmt)** and carrier signal **c(t)=Accos(2πfct)**, then modulated signal **s(t)=[Ac+Amcos(2πfmt)] cos(2πfct)** where,

A_m: Amplitude of modulation signal f_m: Frequency of modulation signal A_c: Amplitude of carrier signal
f_c: Frequency of message signal

Double Sideband Suppressed Carrier Modulation or DSB-SC is an amplitude modulated wave transmission scheme in which only sidebands are transmitted and the carrier is not transmitted as it gets suppressed. The carrier does not contain any information and its transmission results in loss of power. Thus, only sidebands are transmitted that contains information. This results in saving of power used in transmission.

If modulation signal **m(t)=Amcos(2πfmt)** and carrier signal **c(t)=Accos(2πfct)**, then modulated signal **s(t)= m(t).c(t) = AmAc cos(2πfmt) cos(2πfct))** where,

A_m: Amplitude of modulation signal f_m: Frequency of modulation signal A_c: Amplitude of carrier signal
f_c: Frequency of message signal

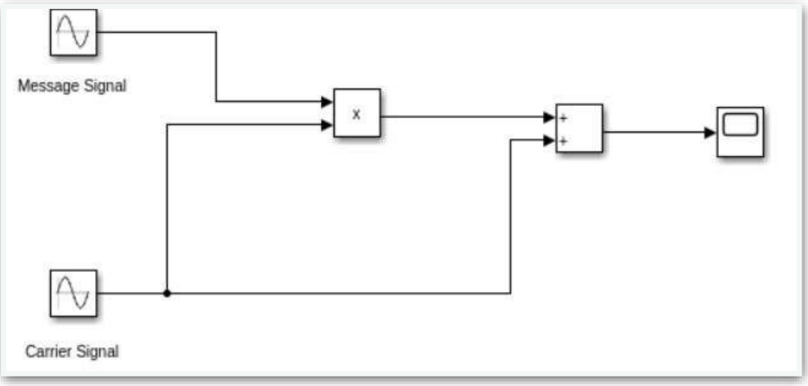
ALGORITHM:
Step 1: Identify the components required and select them from the Simulink library browser.

Step 2: Construct the model for AM modulation.
Step 3: Define the amplitude and frequency of the Message and Carrier signal. Save and click on Run to obtain the output.
Step 4: Construct the model for DSBSC modulation.

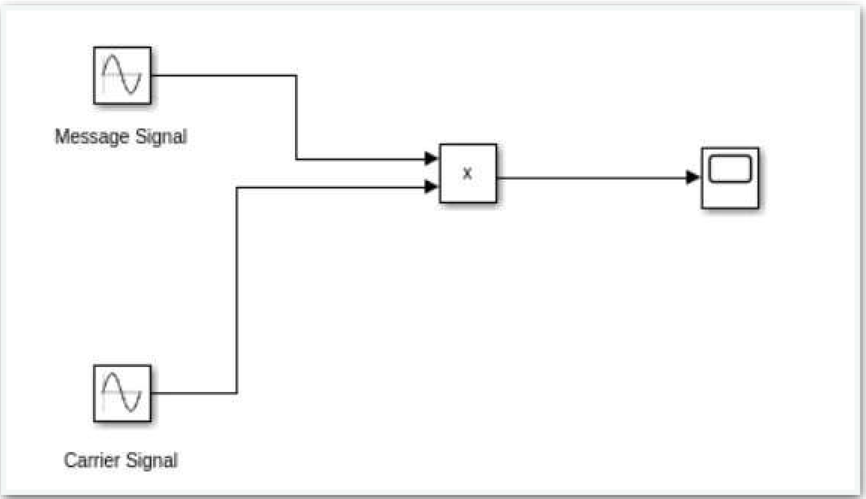
Step 5: Define the amplitude and frequency of the Message and Carrier signal. Save and click on Run to obtain the output.

MATLAB MODEL:

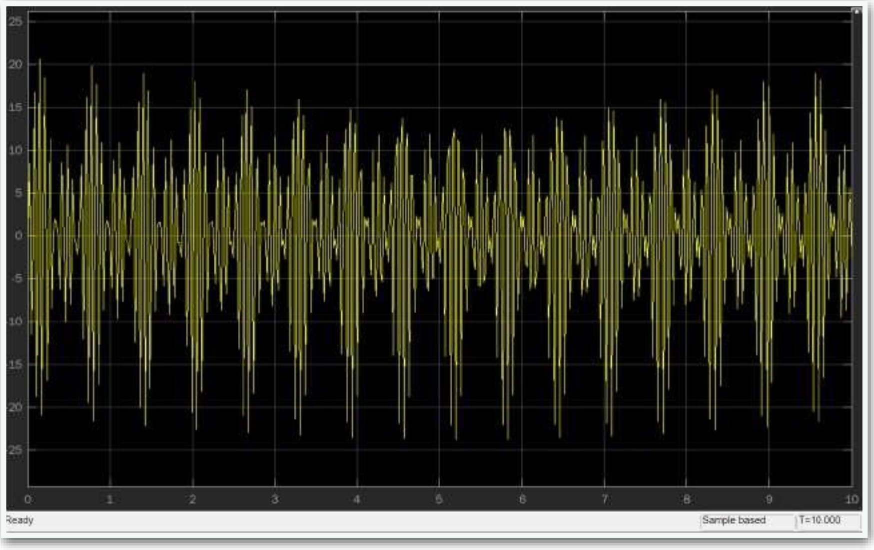
AMPLITUDE MODULATION



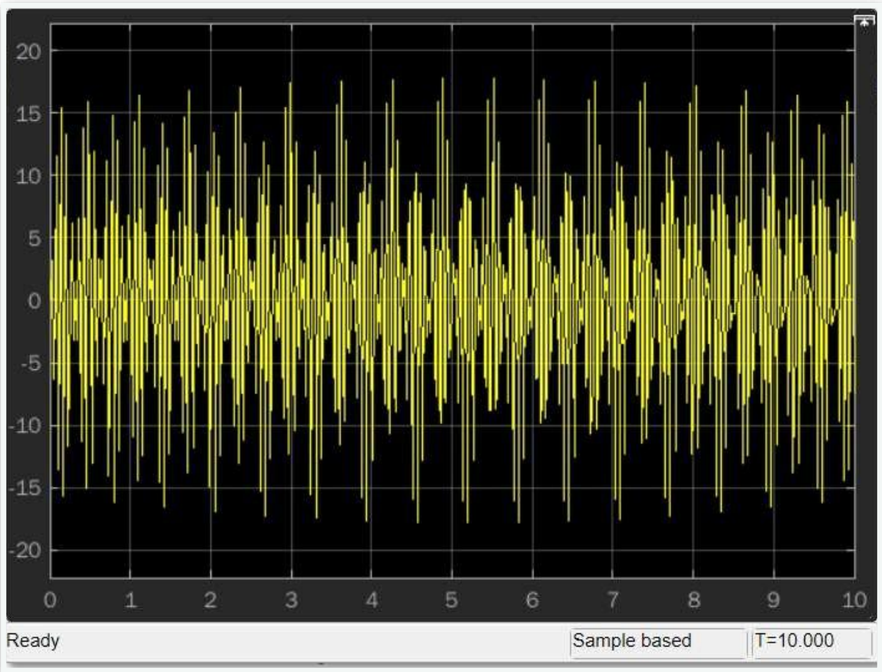
DSB-SC



OUTPUT:
AMPLITUDE MODULATION



DSB-SC



INFERENCE:

Amplitude modulation and DSBSC modulation have been simulated using Simulink in MATLAB.

NAME: Siddharth Bose

REGISTRATION NO.: 19BEC1278

EXP. NO: 5

PHASE MODULATION AND DEMODULATION

AIM: Write a MATLAB program to execute and display the output for Phase modulation and demodulation.

SOFTWARE REQUIRED: MATLAB

THEORY:

PM is a type of angle modulation and it is defined as the change in phase of the carrier signal in correspondence with the amplitude of the message signal. Here, both the frequency and amplitude of the carrier signal stays as constant whereas phase varies in accordance. When there is a positive amplitude, the phase varies in one direction, while there is a negative amplitude, the phase varies in other directions.

If modulation signal $m(t)=Amcos(2\pi fmt)$ and carrier signal $c(t)=Accos(2\pi fct)$, then modulated signal $s(t)=Accos(2\pi fct+kpm(t))$ where,

Am: Amplitude of modulation signal
fm: Frequency of modulation signal
Ac: Amplitude of carrier signal
fc: Frequency of message signal
kp: phase sensitivity

ALGORITHM:

Step 1: Define the values for Am (message signal amplitude), Ac (carrier signal amplitude) ,fm (message signal frequency), fc (carrier signal frequency).

Step 2: Use the equation

- m=A*sin(2*pi*fm*t) to define the message signal
- C=A*sin(2*pi*fc*t) to define the carrier signal

Step 3: Use the pmmod() command to obtain phase modulated signal and the pmdemod() command to obtain the demodulated signal.

Step 4: Finally, save and click on Run to obtain the output graphs.

ALGORITHM:

Step 1: Define the values for Am (message signal amplitude), fm (message signal frequency), Ac (carrier signal amplitude), fc (carrier signal frequency) and m (modulation index).

Step 2: Use the equation

- m=Am*sin(2*pi*fm*t) to define the message signal
- C=Ac*sin(2*pi*fc*t) to define the carrier signal
- s=Ac*(1+ma*m).*sin(2*pi*fc*t) to define the modulated signal

Step 3: Use the pmmod() command to obtain phase modulated signal and the pmdemod() command to obtain the demodulated signal.

Step 4: Plot the message signal, carrier signal, modulated signal and demodulated signal in a single window using the subplot() command.

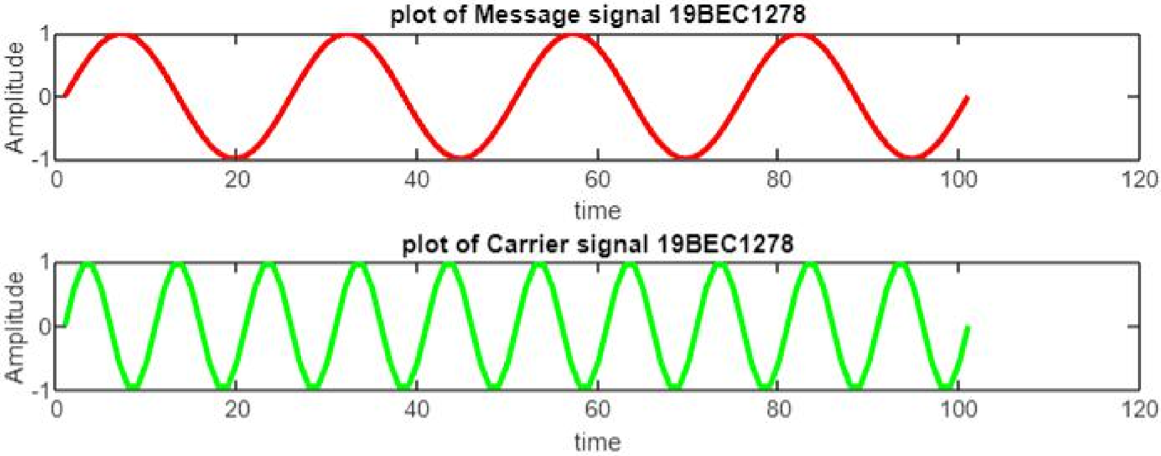
Step 5: Finally, save and click on Run to obtain the output graphs.

MATLAB CODE:

```
clc;

clear;

close all;

%% Message Signal

t=0:0.01:1;

fm=4;  %% message Signal Freq

Am=1;

msg = Am*sin(2*pi*fm*t);

subplot(4,1,1)

plot(msg,'r',LineWidth=2);

xlabel('time');

ylabel('Amplitude');
```

```
title('plot of Message signal 19BEC1278');

%% Carrier Signal

Ac=1;

t=0:0.01:1;

fc=10; %% carrier Signal Freq . must be higher than message signal freq

car = Ac*sin(2*pi*fc*t);

subplot(4,1,2)

plot(car,'g',LineWidth=2);

xlabel('time');

ylabel('Amplitude');

title('plot of Carrier signal 19BEC1278');


%% Phase Modulation using inbuilt Command

Fs=178;

tx = pmmod(msg,fc,Fs,phasedev);

subplot(4,1,3)

plot(tx,'k',LineWidth=2);

xlabel('time');

ylabel('Amplitude');

title('plot of Modulated signal (Using inbuilt Cmd) 19BEC1278 ');

%% Phase De-Modulation using inbuilt Command

Z = pmdemod(tx, fc, Fs,phasedev); %% Y is the Modulated signal from Previous

subplot(4,1,4)

plot(Z,'m',LineWidth=2);

xlabel('time');

ylabel('Amplitude');

title('plot of DeModulated signal (Using inbuilt Cmd)19BEC1278 ');
```

<u>OUTPUT</u>


plot of Message signal 19BEC1278
Amplitude
time
1
0
-1
0
20
40
60
80
100
120
plot of Carrier signal 19BEC1278
Amplitude
time
1
0
-1
0
20
40
60
80
100
120

INFERENCE:

The PM modulated and demodulated signal was generated using in built commands.

NAME: Siddharth Bose

REGISTRATION NO.: 19BEC1278

EXP. NO: 6

Noise in AM Receiver

AIM: Write a MATLAB program to execute and display the output for Amplitude modulation and demodulation.

SOFTWARE REQUIRED: MATLAB

THEORY:

Noise is an unwanted signal, which interferes with the original message signal and corrupts the parameters of the message signal. This alteration in the communication process, leads to the message getting altered. It most likely enters at the channel or the receiver. Noise is some signal which has no pattern and no constant frequency or amplitude. It is quite random and unpredictable.

If modulation signal m(t)=Amcos(2πfm*t) and carrier signal c(t)=Accos(2πfc*t), then the noise signal n(t) = (Ac+NF*(cos(2πfm*t))) *cos(2πfc*t) where,

Ac = Amplitude of carrier signal
NF = Noise factor
fm = Frequency of modulating signal
fc = Frequency of carrier signal

ALGORITHM:

Step 1: Define the values for Am (message signal amplitude), Ac (carrier signal amplitude) ,fm (message signal frequency), fc (carrier signal frequency).
Step 2: Use the equation

- m=A*sin(2*pi*fm*t) to define the message signal
- Carr=A*sin(2*pi*fc*t) to define the carrier signal

Step 3: Use the equation

- s=(Ac+NF*cos(wm*t)).*cos(wc*t) to plot the noise signal and vary the value of NF from 1 to 10 with an increment of 2.

Step 4: Use the awgn() command to plot the noise signal using in-built command.

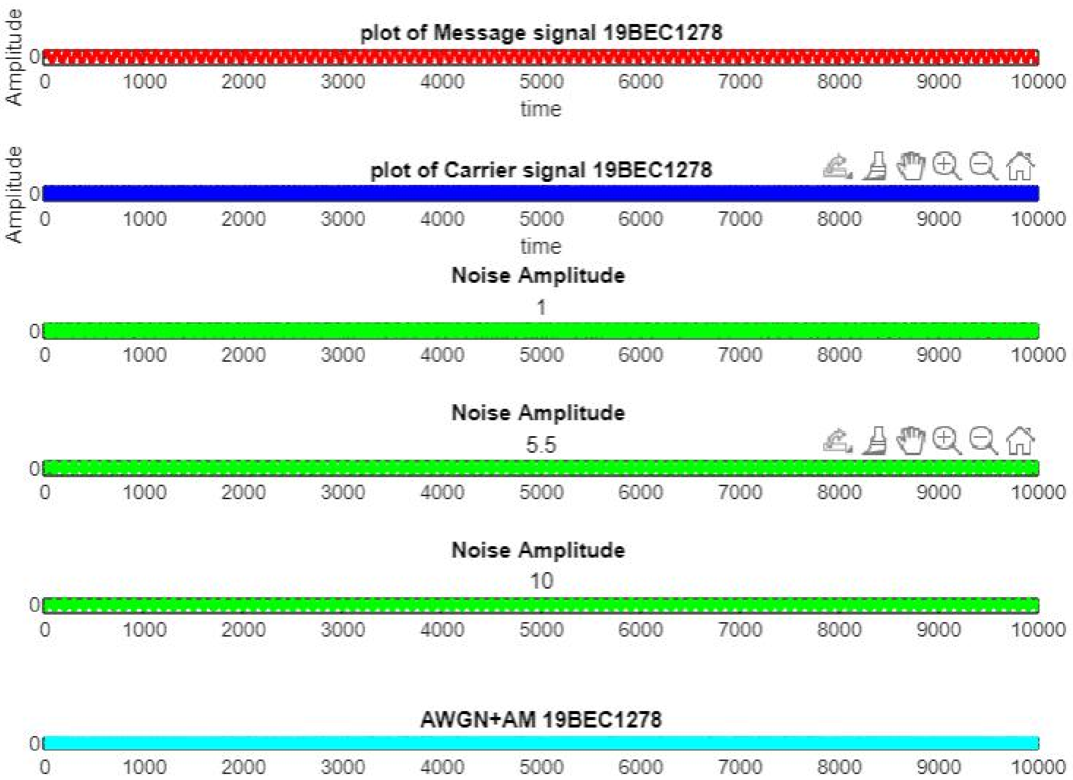
Step 5: Finally, save and click on Run to obtain the output graphs.

MATLAB CODE:

```
clc;

clear;

close all;

t=linspace(0,1,10000);

%% Message Signal

fm=100;  %% message Signal Freq

Am=2;

msg = Am*sin(2*pi*fm*t);

subplot(6,1,1)

plot(msg,'r',LineWidth=2);

xlabel('time');

ylabel('Amplitude');

title('plot of Message signal 19BEC1278');

%% Carrier Signal

Ac=5;

fc=500; %% carrier Signal Freq . must be higher than message signal freq

car = Ac*sin(2*pi*fc*t);

subplot(6,1,2)

plot(car,'b',LineWidth=2);

xlabel('time');

ylabel('Amplitude');

title('plot of Carrier signal 19BEC1278');

n=2;


for NF=linspace(1,10,3)
```

```
    s=(Ac+ NF*cos(2*pi*fm*t)).*cos(2*pi*fc*t);n=n+1;

    subplot(6,1,n)

    plot(s,'g',LineWidth=2);

    title('Noise Amplitude ' ,NF);


end

mod=msg.*car;

noised=awgn(mod,20,0);

subplot(6,1,6)

plot(noised,'c',LineWidth=2);

title('AWGN+AM 19BEC1278');
```

<u>OUTPUT:</u>



<u>INFERENCE:</u>

The noise signal in AM receiver was generated using theoretical formulae and in built command.

NAME: Siddharth Bose

REGISTRATION NO.: 19BEC1278

EXP. NO: 7

DATE: 08/03/2022

PPM, PAM AND PWM SIGNALS

AIM : Write a MATLAB program to execute and display the output for PPM (Pulse position modulation), PAM (Pulse amplitude modulation) and PWM (Pulse width modulation) signals.

SOFTWARE REQUIRED: MATLAB

THEORY:

a) Pulse position modulation (PPM)

Pulse Width Modulation (PWM) or Pulse Duration Modulation (PDM) or Pulse Time Modulation (PTM) 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.

b) Pulse amplitude modulation (PAM)

Pulse Amplitude Modulation (PAM) is an analog modulating scheme in which the amplitude of the pulse carrier varies proportional to the instantaneous amplitude of the message signal.

The pulse amplitude modulated signal, will follow the amplitude of the original signal, as the signal traces out the path of the whole wave. In natural PAM, a signal sampled at the Nyquist rate is reconstructed, by passing it through an efficient Low Pass Frequency (LPF) with exact cutoff frequency.

c) Pulse width modulation (PWM)

Pulse Position Modulation (PPM) 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.

Pulse position modulation is done in accordance with the pulse width modulated signal. Each trailing of the pulse width modulated signal becomes the starting point for pulses in PPM signal. Hence, the position of these pulses is proportional to the width of the PWM pulses.

ALGORITHM:

a) Pulse position modulation (PPM)

Step 1: Define the value for carrier frequency (fc), sampling frequency (fs) and modulating frequency (fm).

Step 2: Generate a message signal defined by X= 0.5*cos(2*pi*fm*t)+0.5.

Step 3: Modulate the message signal by using the modulate(x,fc,fs,method) command (modulation technique specified by method) in Matlab.

Step 4: Plot the message and modulated signal using the plot () command.

Step 5: Finally, save and click on Run to obtain the output graphs.

b) Pulse amplitude modulation (PAM)

Step 1: Define the value for amplitude and frequency.

Step 2: Generate an impulse signal using the stem(t) command and a sine signal using the sin (2 * pi * f * t) command in Matlab.

Step 3: Pulse amplitude modulation is achieved by multiplying pulse signal and message signal.

Step 4: Plot the impulse signal and sine signal using the plot () command. Plot the Pulse amplitude modulation signal by using the stem () command.

Step 5: Finally, save and click on Run to obtain the output graphs.

c) Pulse width modulation (PWM)

Step 1: Define the value for amplitude and frequency.

Step 2: Generate a sawtooth signal using the sawtooth(t) command and a sine signal using the sin (2 * pi * f * t) command in Matlab.

Step 3: Now compare the message signal and the sawtooth signal. If the message signal is greater, modulated signal is made as 1 or else it is set as 0.

Step 4: Plot the impulse signal, sine signal and modulated signal using the plot ()command.

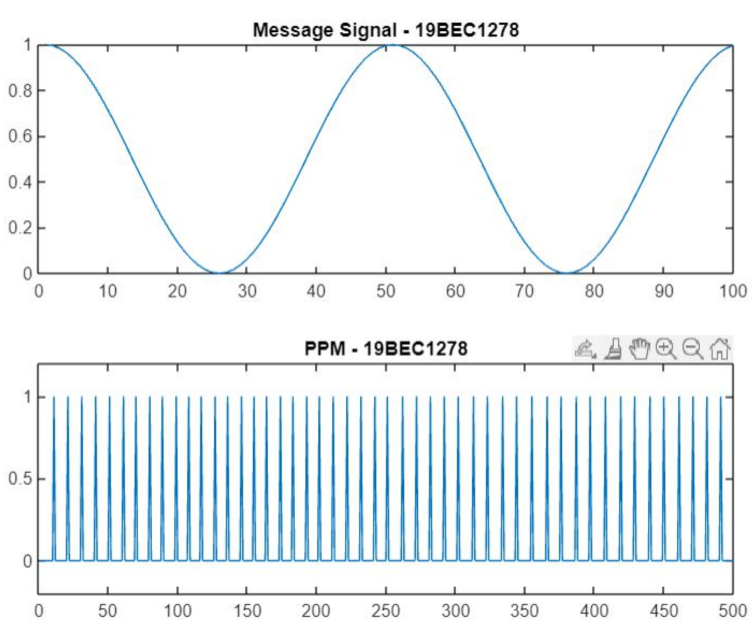
Step 5: Finally, save and click on Run to obtain the output graphs.

PPM:

Matlab Code:

```
clc;

clear all;

close all;

fc=100;

fs=1000;

fm=20;

t=0:1/fs:((2/fm)-(1/fs));

X=  0.5*cos(2*pi*fm*t)+0.5;

Y= modulate(X,fc,fs,'PPM');

subplot(2,1,1);

plot(X);

title('Message Signal - 19BEC1278');

subplot(2,1,2);

plot(Y);

axis([0 500 -0.2 1.2]);

title('PPM - 19BEC1278');
```

Output:

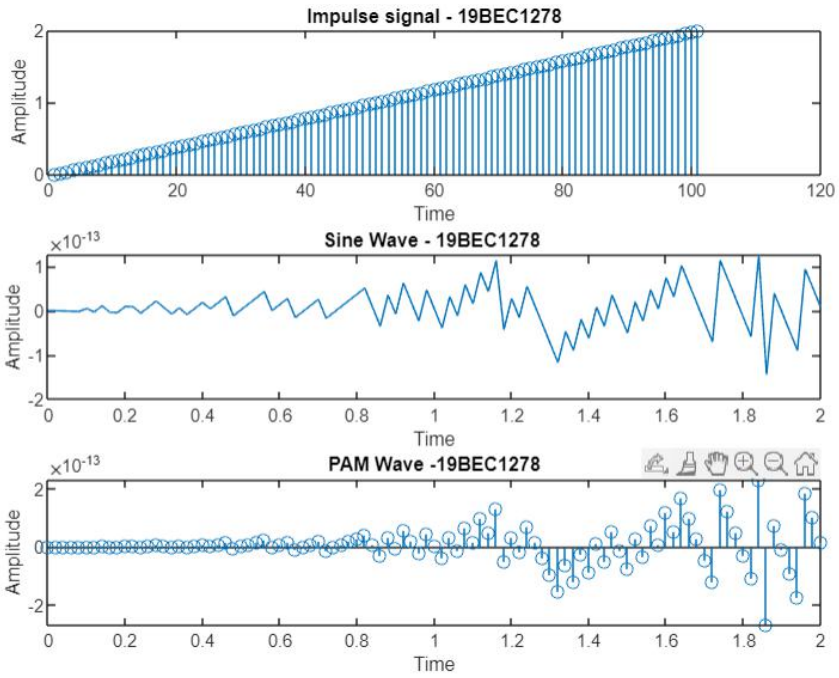


PAM:

Code:

```
clc
close all
clear all
a=input("Enter the amplitude: ");
f=input("Enter the frequency: ")'
t=0:0.02:2;
x1=stem(t);
x2=sin(2*pi*f*t);
y=t.*x2;
subplot(3,1,1);
stem(t);
title("Impulse signal - 19BEC1278");
xlabel("Time");   ylabel("Amplitude");
subplot(3,1,2);
plot(t,x2);
title("Sine Wave - 19BEC1278");
xlabel("Time");
ylabel("Amplitude");
subplot(3,1,3);
stem(t,y);
title("PAM Wave -19BEC1278");
xlabel("Time");
ylabel("Amplitude");
```

Output:



PWM:

Code:

clc;

clear;

close all;

t=0:0.001:10;

y=sin(2*pi*t);

z=sawtooth(4*pi*t);

subplot(3,1,1)

plot(y);

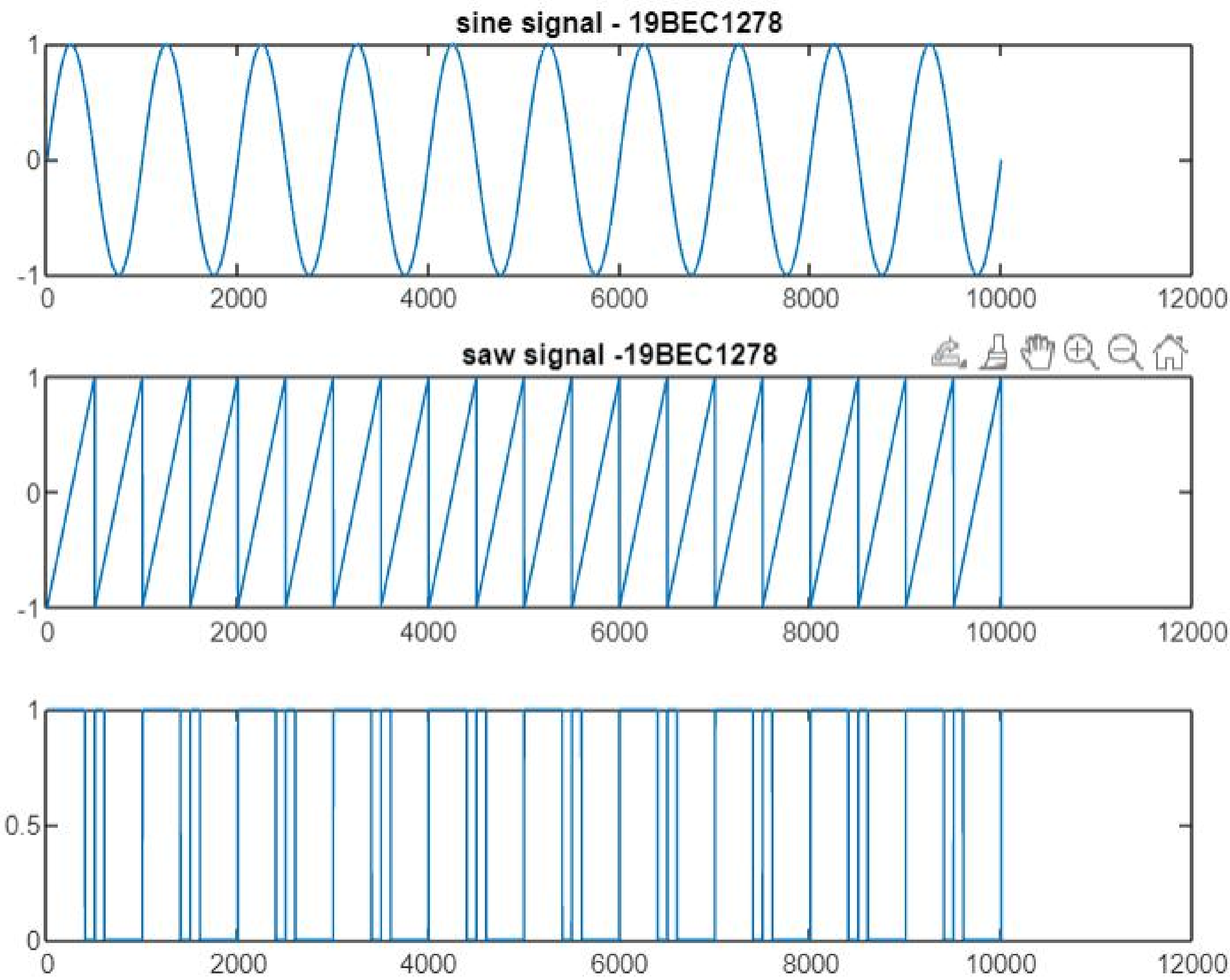
title("sine signal - 19BEC1278");

subplot(3,1,2)

plot(z);

```
title("saw signal -19BEC1278");
for i=1:length(t)
   if(y(i)>=z(i))
      u(i)=1;
   else
      u(i)=0;
   end
end
subplot(3,1,3)
plot(u);
```

Output:



Expt. No. 08

Date 15/3/22

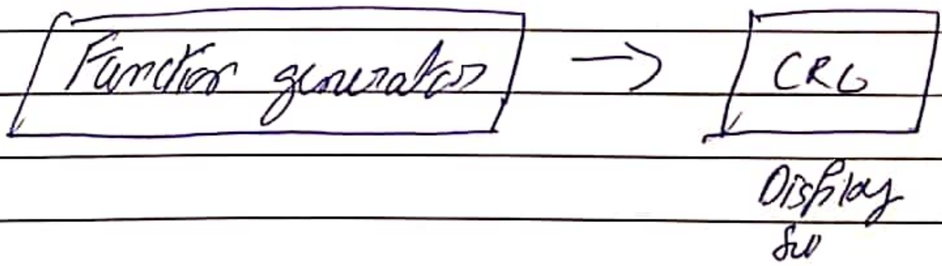
Page No. 1

Aim :- The study of resistor function generator and cathod ray osillo...
To find the given resistor value.
To plot the sine, square, Triangle wave at different freq (low, middle, high)

Components :-

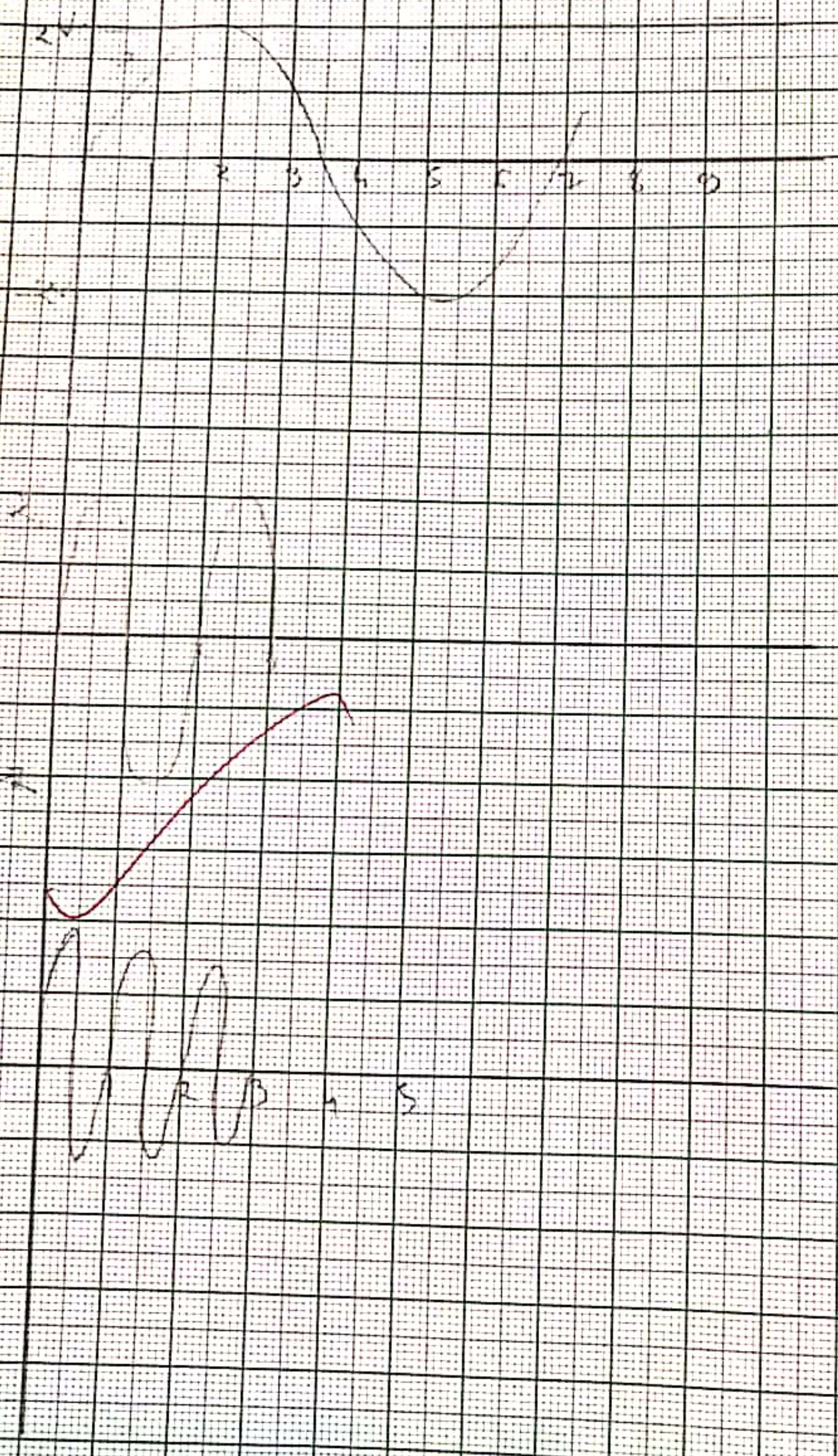
(i) Resistor
(ii) Function Generator
(iii) CRO
(iv) wires

Block diagram :-

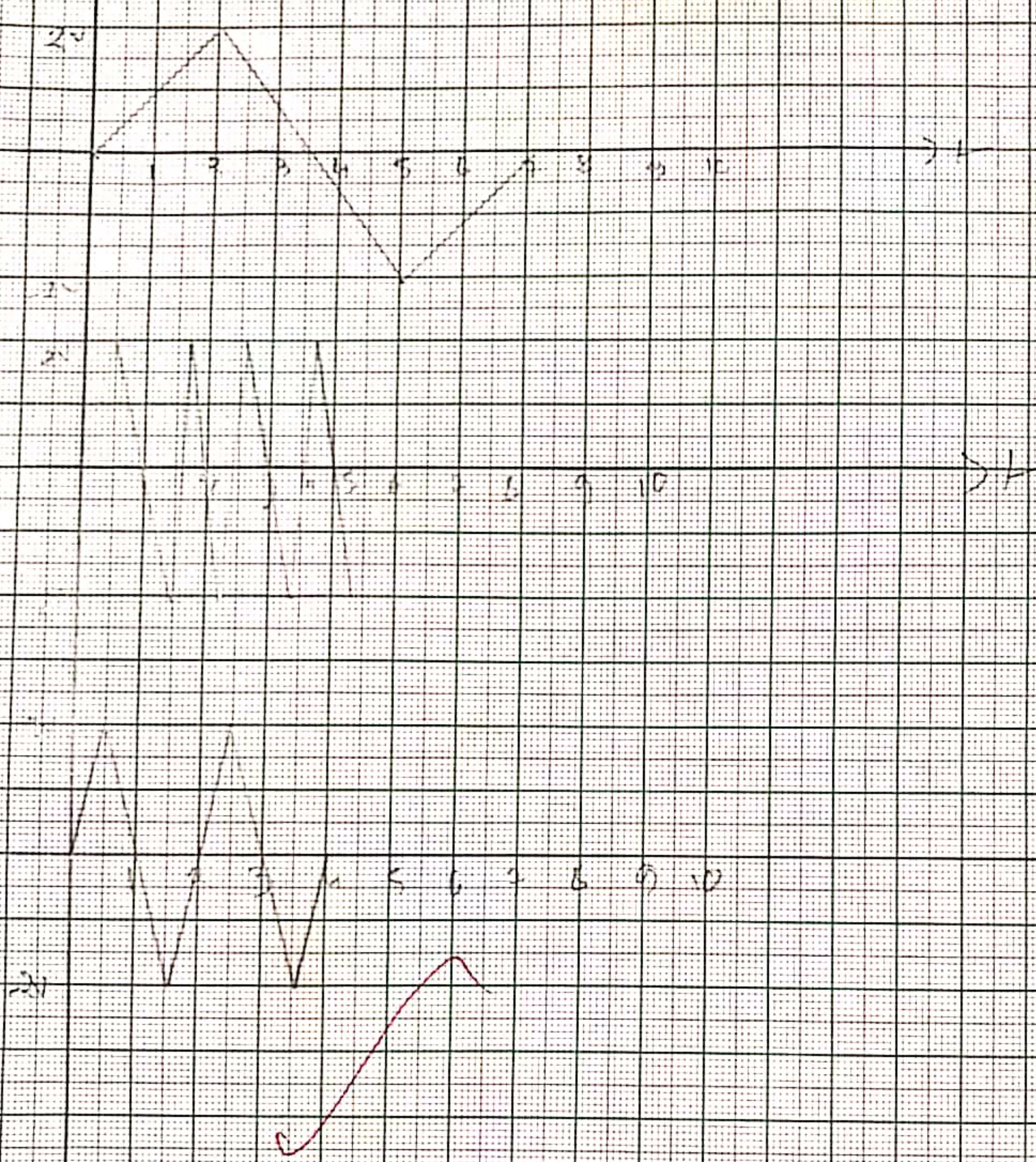


Teacher's Signature

Sine Wave



Triangle.



Date 15/3/22

Expt. No. 8 Page No. 2

S.L	Wave	Amplitude (V)	Time (u sec)
1	Sine (low) mid high	4 4 4	7 2 1
2	Square low, mid high	4 4 4	7 2 1
3	Triangle low mid high	4 4 4	72 1

8/10

Tabulation

Component	Parameter	Value
Resistor 1	Band = 4	
	colour1 = Brown	10
	colour2 = Black	0
	colour3 = Red	10^2
	colour 4 = Gold	5%
		10 kΩ ± 5%
Resistor 2	# Band = 4	
	colour1 = yellow	40
	colour2 = Blue	6
	colour3 = Brown.	10^1
	colour 4 = silver	10
		460 kΩ ±10%
Resistor 3	# Band = 4	
	colour = 1 = Red	20
	colour 2 = Violet	7
	colour 3 = Red.	10^2
	colour 4 = Silver.	10
		2.7 kΩ ± 10%

Teacher's Signature

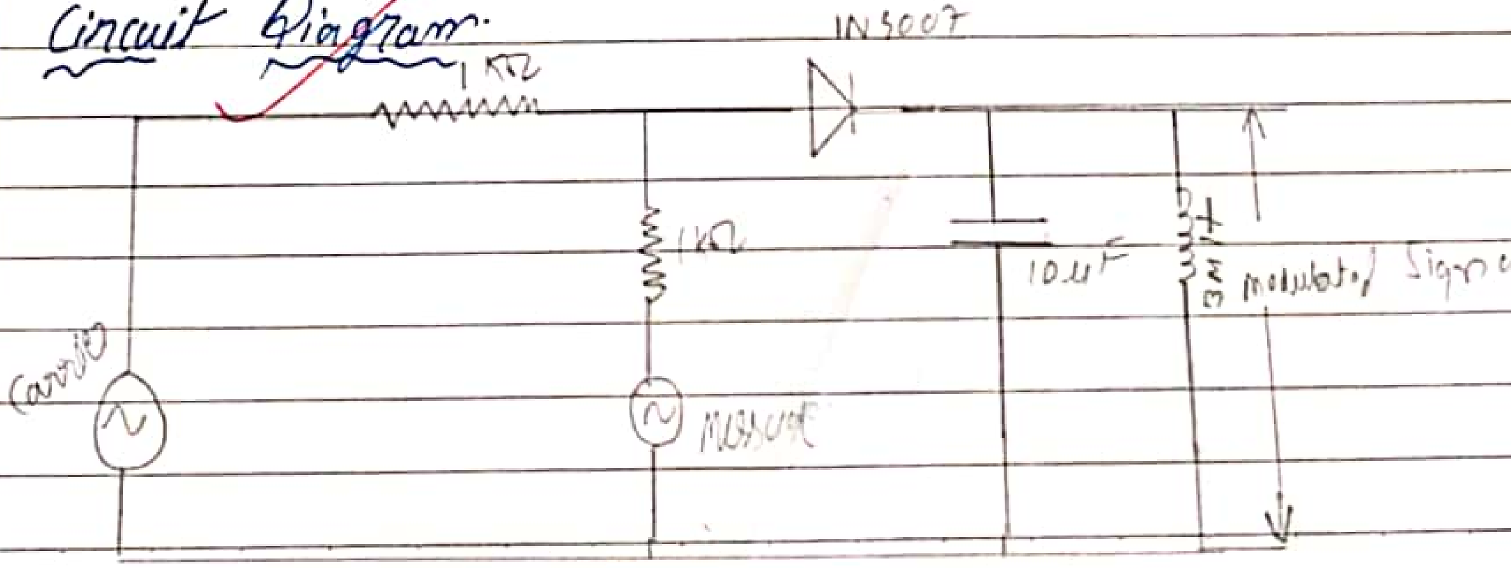
Amplitude Modulation

(i) Aim:- To generate Amplitude modulated wave using electronics components.

(ii) Apparatus Required :-

SNO	Components	NO.
1.	diode	1
2.	Resistor.	2
3.	capacitor	1
4.	Inductor.	1

Circuit Diagram:

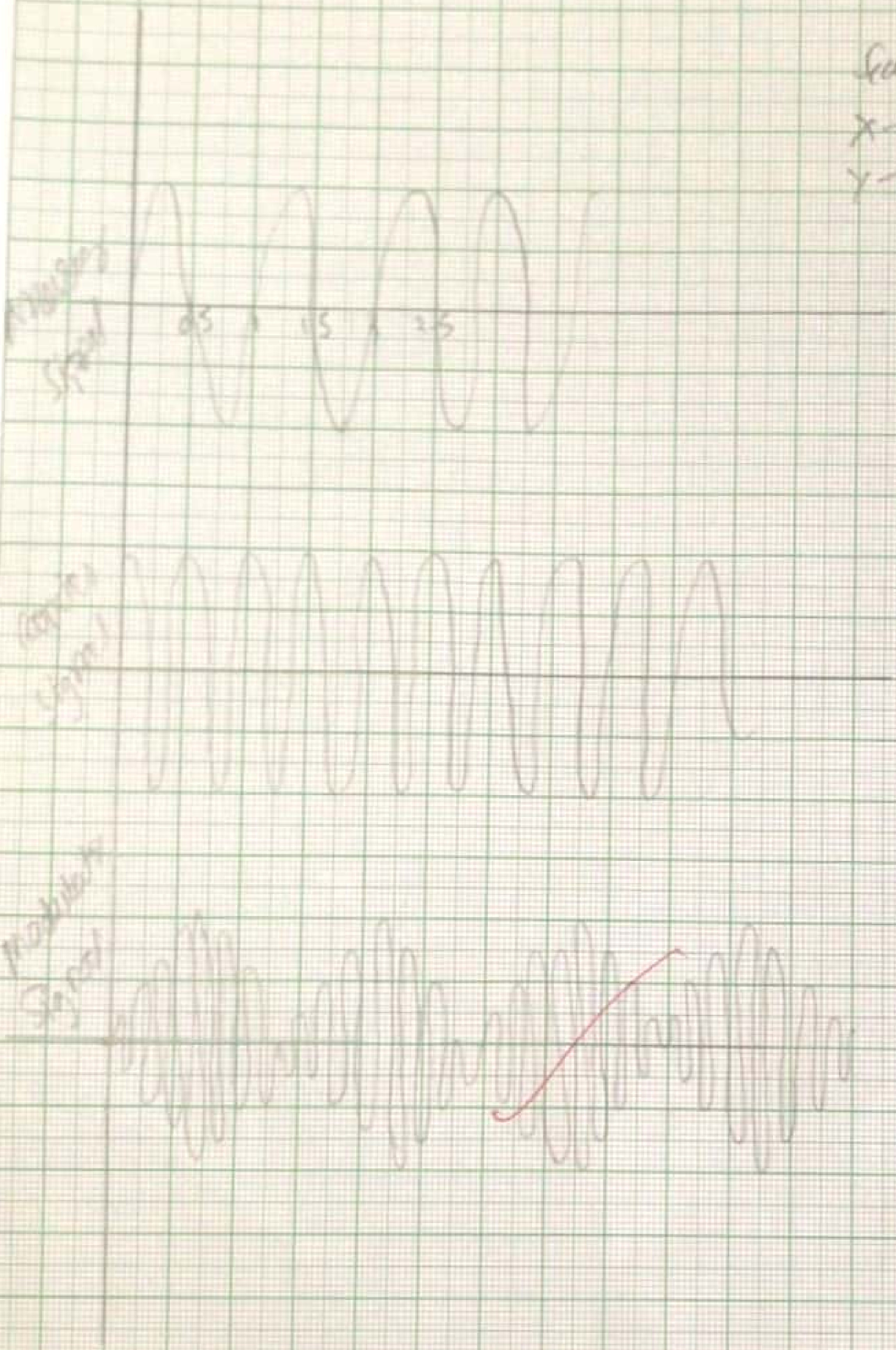


Teacher's Signature

Scale -

x-axis 1cm = 0.2 mil s

y-axis 1cm = 0.5V



Date 19/03/2022

Expt. No. 9 Page No. 4

Signal.	Frequency	Amplitude	Time Period.
carrier	10KHZ	4 V	0.1ms
message	1KHZ	1 V	1ms
AM	10.05KHZ	837 mV	0.1ms.

o/p verified
19/3/22

Inference:-

We have constructed circuit and gave input message. and carrier sin waves & got a modulated output signal.

10/10

Result

Modulated Index

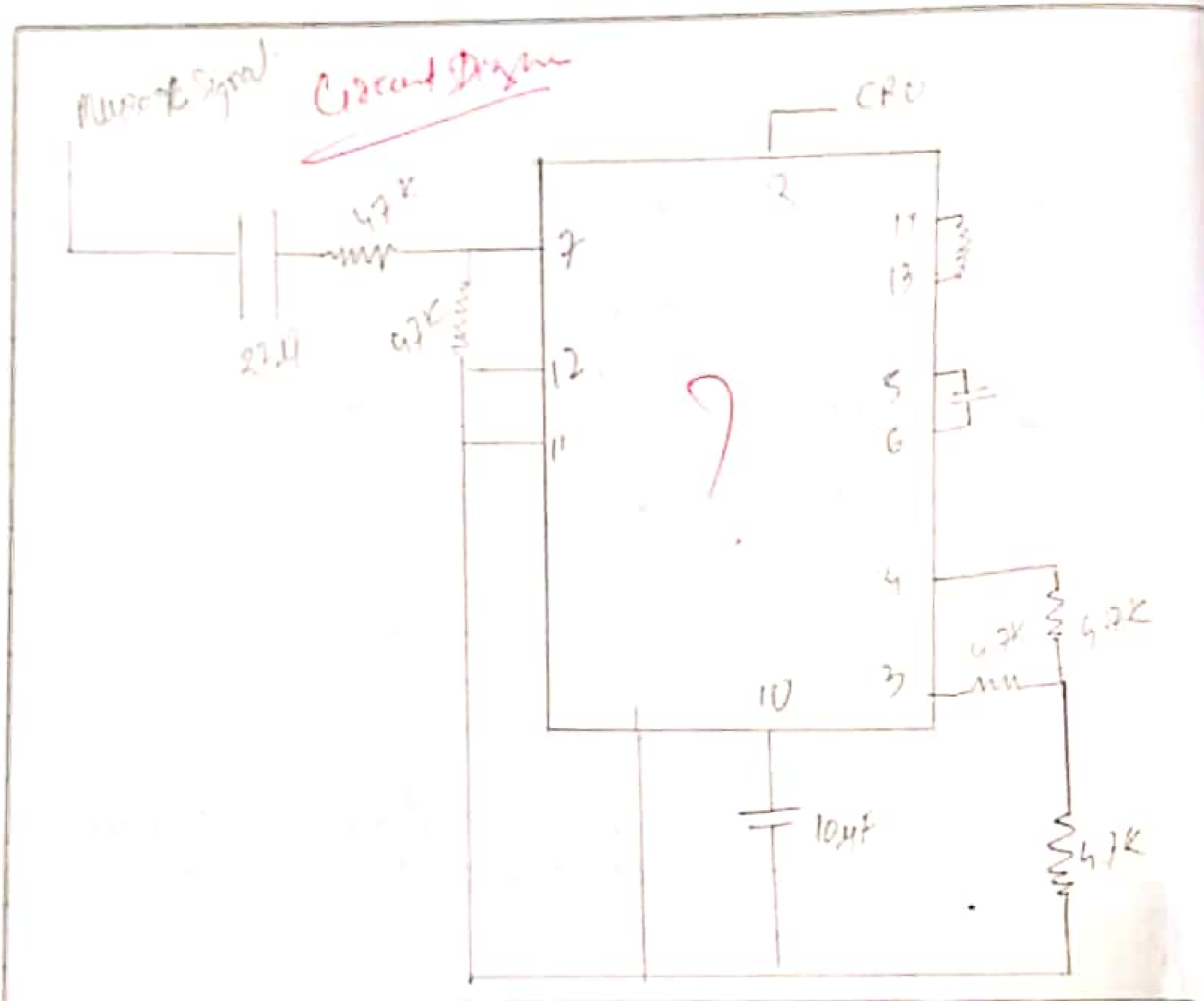
$$= \frac{A_m}{A_c}$$

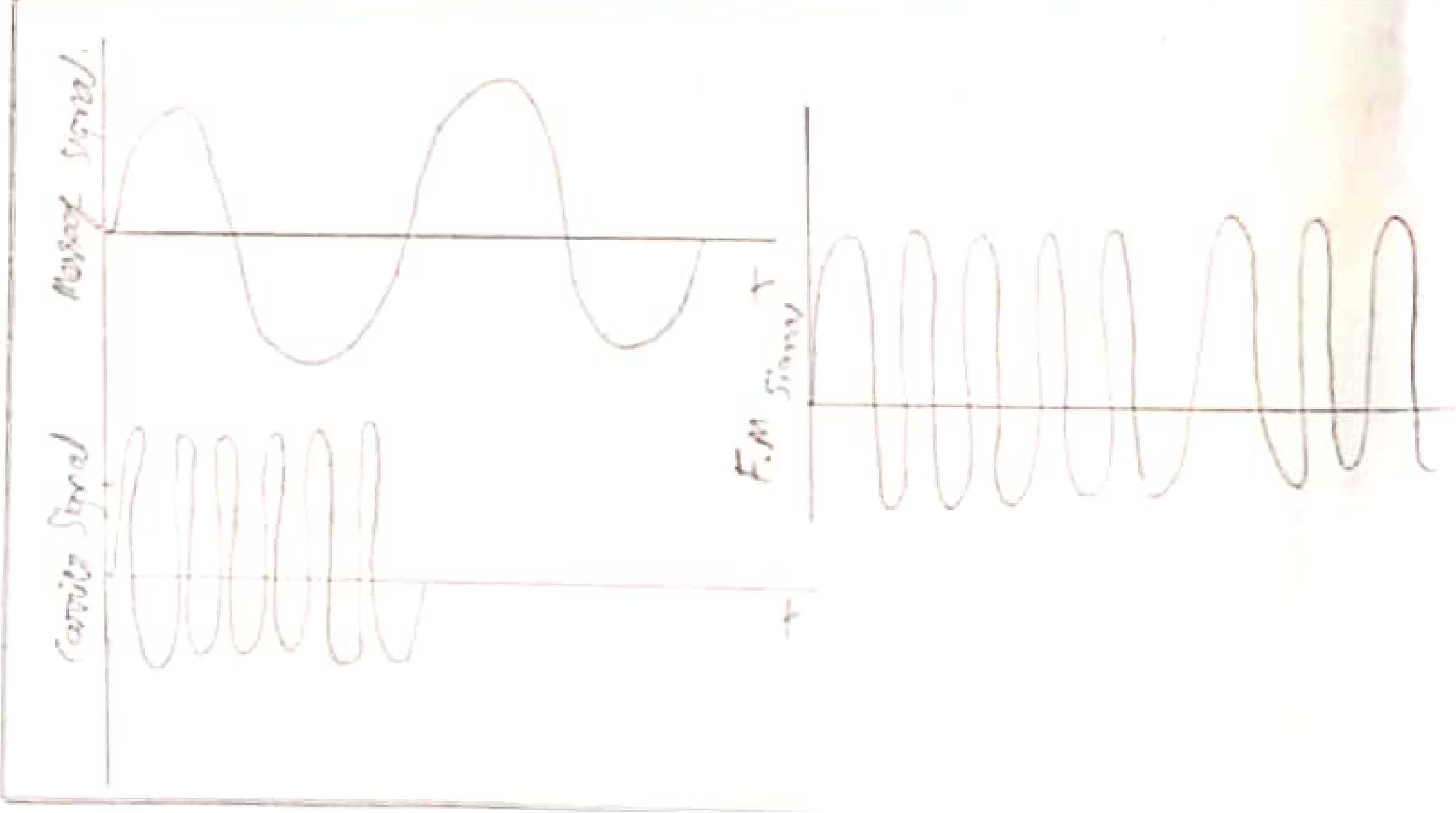
$$= 1/4 = 0.25$$

Thus, we generate amplitude modulation wave

Teacher's Signature

Circuit Diagram





Date 22/03/2022

Expt. No. 10 Page No. 5

Frequency Modulation.

Aim :- To calculate frequency modulated wave using electronic components

Apparatus Required :-

SNo	Components	Numbers
1	Function Generator	~~Numbers~~ 1
2	IC	1
3	Resistor	6
4	Capacitors	~~2~~ 3
5	Bread board	~~1~~ 1
6	CRO	1

8/10

Calculation

$f_m = 1 KHz$

$f_{max} = 1.04 \times 10^4 Hz$

$f_{min} = 2.54 \times 10^3 Hz$

$V_{pp} = 5.675$ $\Delta f = 7.56 \times 10^3 Hz$

$m_f = \frac{\Delta f}{f_m} = 7.56.$ $B.W = 2(\Delta f + f_m)$

$= 2(7.6+1) KHz$

$= 17.2 KHz.$

Result? **Inference :-**

Bandwidth $= 2(\Delta f + f_m) = 17.35 KHz$ and modulation index, $\beta = \frac{\Delta f}{f_m} = 8.4$. During the half cycle of message signal FM has high frequency and other half has low frequency.

Teacher's Signature

NAME: Siddharth Bose

REGISTRATION NO.: 19BEC1278

EXP. NO : 11 DATE : 12-04-2022

TDM&FDM

AIM: Write a MATLAB program to execute and display the output TDM and FDM.

SOFTWARE REQUIRED: MATLAB

THEORY:

FDM

frequency-division multiplexing (FDM) is a technique by which the total bandwidth available in a communication medium is divided into a series of non-overlapping frrequency , each of which is used to carry a separate signal. This allows a single transmission medium such as a cable or optical fiber to be shared by multiple independent signals. Another use is to carry separate serial bits or segments of a higher rate signal in parallel.

The most common example of frequency-division multiplexing is radio and television broadcasting, in which multiple radio signals at different frequencies pass through the air at the same time.

TDM

Time-division multiplexing (TDM) is a method of transmitting and receiving independent signals over a common signal path by means of synchronized switches at each end of the transmission line so that each signal appears on the line only a fraction of time in an alternating pattern. This method transmits two or more digital signals or analog signals over a common channel.

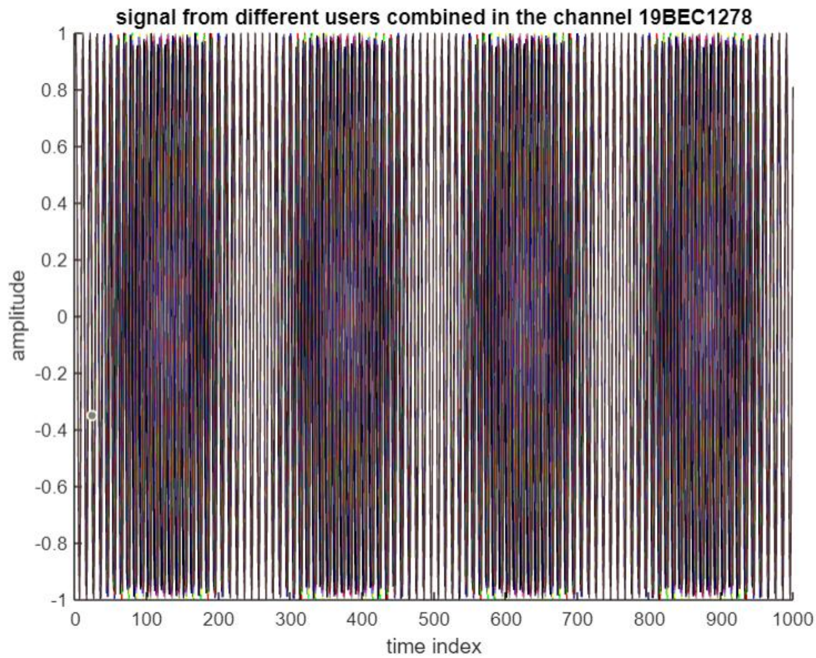
FDM:

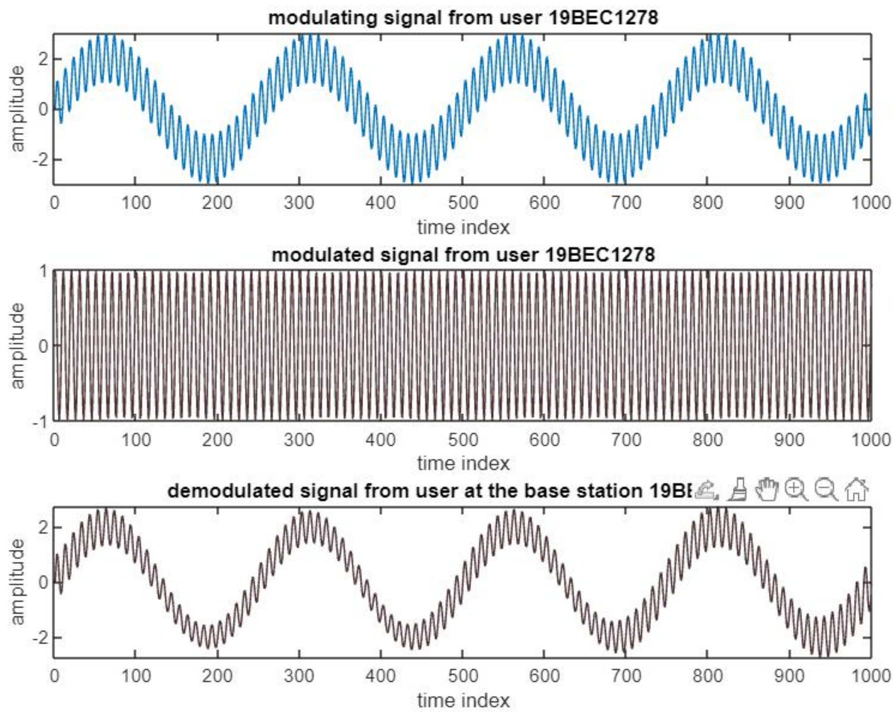
MATLAB CODE:

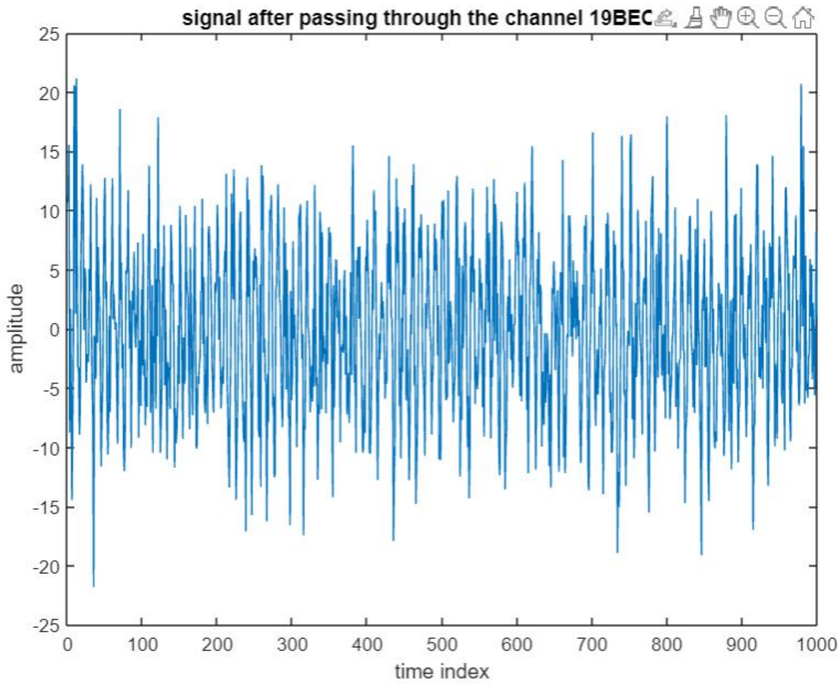
```
clc;
clear all
close all
samples=1000;
nos=8;
mfreq=[30 40 50 60 70 80 90 100];
cfreq=[300 600 900 1200 1500 1800 2100 2400];
freqdev=10;
t=linspace(0,1000,samples);
for i=1:nos
    m(i,:)=sin(2*pi*mfreq(1,i)*t)+2*sin(pi*8*t);
end
for i=1:nos
    y(i,:)=fmmod(m(i,:),cfreq(1,i),10*cfreq(1,i),freqdev);

end
ch_op=awgn(sum(y),0,'measured');
for i=1:nos
    z(i,:)=fmdemod(y(i,:),cfreq(1,i),10*cfreq(1,i),freqdev);

end

C={'k','b','r' ,'g','y',[.5 .6 .7],[.8 .2 .6],[.3 .2 .2]};
for i=1:nos
    figure(1)
    hold on
    plot(y(i,:),'color',C{i});
```

```
    xlabel('time index');
    ylabel('amplitude');
    title('signal from different users combined in the channel 19BEC1278')
    figure
    subplot(3,1,1)
    plot(m(i,:))
    xlabel('time index');
    ylabel('amplitude');
    title('modulating signal from user 19BEC1278')
    subplot(3,1,2)
    plot(y(i,:),'color',C{i});
    xlabel('time index');
    ylabel('amplitude');
    title('modulated signal from user 19BEC1278')
    subplot(3,1,3)
    plot(z(i,:),'color',C{i})
    xlabel('time index');
    ylabel('amplitude');
    title('demodulated signal from user at the base station 19BEC1278')
end
figure
plot(ch_op)
xlabel('time index');
ylabel('amplitude');
title('signal after passing through the channel 19BEC1278')
```

OUTPUT:


signal from different users combined in the channel 19BEC1278
amplitude
time index


modulating signal from user 19BEC1278
amplitude
time index
modulated signal from user 19BEC1278
amplitude
time index
demodulated signal from user at the base station 19BI
amplitude
time index



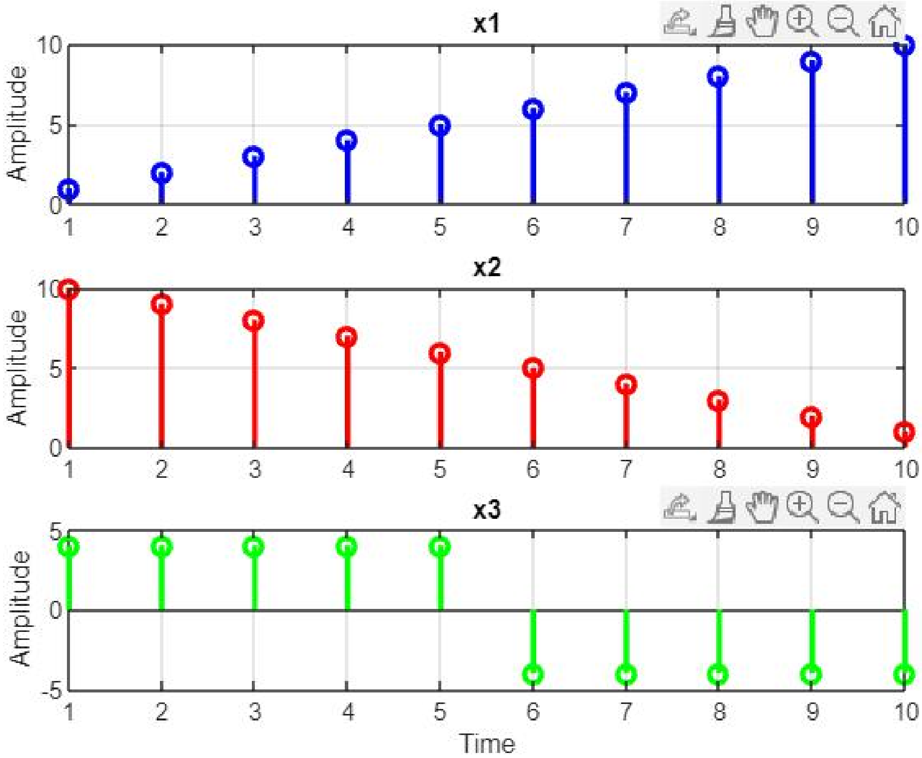
TDM:

MATLAB CODE:

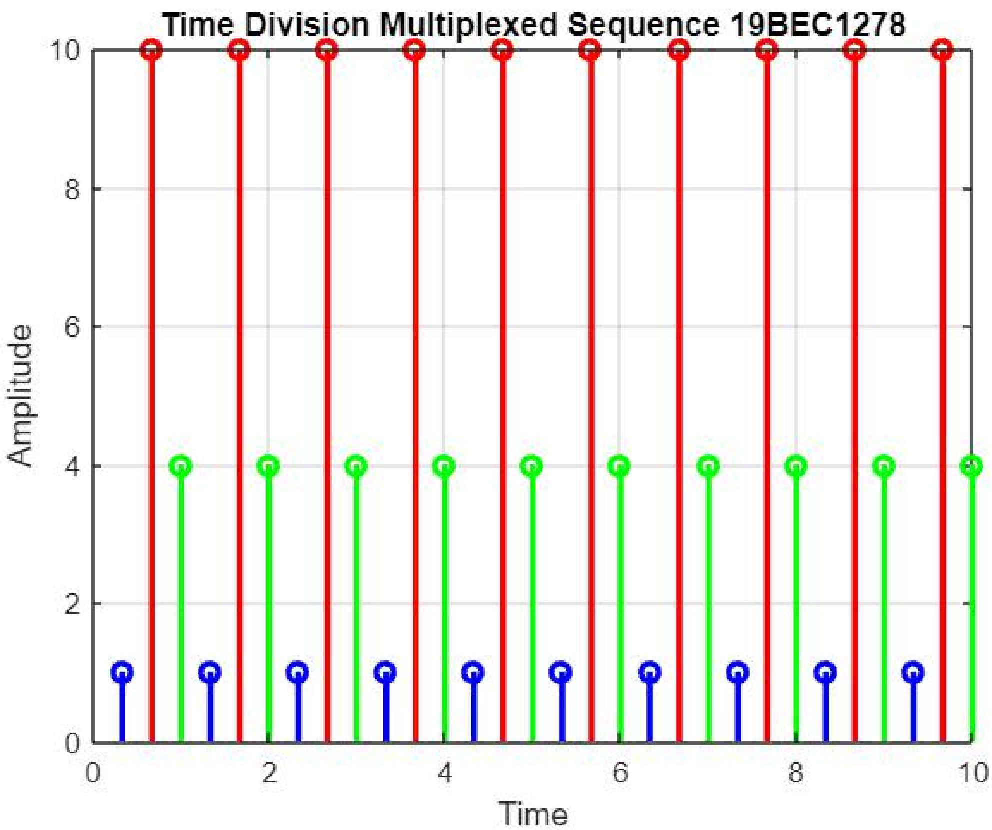
```
%% TDM
clc; clear;
close all;
x1=1:10,
x2=10:-1:1,
x3(1:5)=4,
x3(6:10)=-4,
x(1,:)=x1,
x(2,:)=x2,
x(3,:)=x3
[r c]=size(x);
k=0;
% Multiplexing
for i=1:c
    for j=1:r
        k=k+1;
```

```
        y(k)=x(j,1);
    end
end
% Plotting
color='ybrgmkc';
figure(1)
sig='x1';
for i=1:r
    sig(2)=i+48;
    j=mod(i,7)+1;
    subplot(r,1,i)
    stem(x(i,:),color(j),'linewidth',2)
    title(sig)
    ylabel('Amplitude')
    grid
end
xlabel('Time')
t=1/r:1/r:c;
figure(2)
for i=1:r
    j=mod(i,7)+1;
    stem(t(i:r:r*c),y(i:r:r*c),color(j),'linewidth',2)
    hold on
    grid
end
hold off
title(['Time Division Multiplexed Sequence 19BEC1278'])
xlabel('Time')
ylabel('Amplitude')
```

OUTPUT:



x1
Amplitude
10
5
0
1
2
3
4
5
6
7
8
9
10
x2
Amplitude
10
5
0
1
2
3
4
5
6
7
8
9
10
x3
Amplitude
5
0
-5
1
2
3
4
5
6
7
8
9
10
Time



Time Division Multiplexed Sequence 19BEC1278
Amplitude
10
8
6
4
2
0
0
2
4
6
8
10
Time

INFERENCE:

Hence, a MATLAB program is implemented to execute and display the output of TDM and FDM.