COMMUNICATION ENGINEERING

(EAEPC11)

PRACTICAL RECORD FILE

2023-24

Submitted by

Name: NIKHIL NEGI

Roll No.: 2022UEA6634

Branch: ECAM

Section: 2

Semester: 4

Submitted to:

Dr. Patteti Krishna

Netaji Subhas University of Technology

East Campus, Geeta Colony, New Delhi-110031

INDEX

S.No.	TOPIC	DATE	SIGN.

INTRODUCTION TO MATLAB

MATLAB stands for "**MATrix LABoratory**". It is a high-level programming language and interactive environment developed by MathWorks. MATLAB is widely used in engineering, science, mathematics, and other technical fields for numerical computation, data analysis, visualization, and algorithm development

Operating MATLAB typically involves several steps, including installation, launching the software, writing and executing scripts or commands, and analyzing the results. Here's a step-by-step procedure to operate MATLAB:

1. **Installation**:
    - Download the MATLAB installer from the MathWorks website or install it from a provided DVD.
    - Run the installer and follow the on-screen instructions to install MATLAB on your computer.
    - Activate MATLAB using the license key provided by MathWorks.
2. **Launching MATLAB**:
    - Once installed, double-click the MATLAB icon on your desktop or search for MATLAB in your system's applications menu.

- Alternatively, you can launch MATLAB from the command line by typing matlab and pressing Enter.

3. **MATLAB Desktop Environment**:
   - After launching, MATLAB opens the MATLAB Desktop, which consists of the Command Window, Workspace, Editor, Current Folder Browser, and other tool windows.
   - Familiarize yourself with the MATLAB Desktop environment and its various components.

4. **Writing Scripts or Commands**:
   - MATLAB allows you to execute commands interactively in the Command Window or write scripts and functions in the Editor.
   - To execute commands interactively, simply type the command in the Command Window and press Enter.
   - To write a script or function, open the Editor, write your code, and save the file with a .m extension.

5. **Running Scripts or Commands**:
   - To run a script, simply type its name (without the .m extension) in the Command Window and press Enter.
   - To run a specific section of code in the Editor, place the cursor in that section and press Ctrl + Enter.
   - You can also execute individual commands or lines of code by selecting them and pressing Ctrl + Enter.

6. **Analyzing Results**:
   - MATLAB provides various functions and visualization tools for analyzing and plotting data.

- Use built-in functions for numerical computations, data analysis, signal processing, image processing, etc.
- Visualize data using plotting functions like plot, imshow, surf, histogram, etc.

7. **Saving and Sharing**:
   - Save your scripts and functions for future use by clicking File > Save or pressing Ctrl + S in the Editor.
   - Export figures and results using the Export options available in the figure windows.
   - Share your code and results with others by providing them with the MATLAB script files or exporting results to common file formats like PDF or PNG.
8. **Exiting MATLAB**:
   - To exit MATLAB, you can simply close the MATLAB window by clicking the close button (X) or by selecting File > Exit MATLAB.
   - Ensure that you save any unsaved work before exiting to avoid losing data.

Built in Functions:

1**. Scalar Functions**:

Certain MATLAB functions are essentially used on scalars, but operate element-wise when applied to a matrix (or vector). They are summarized below.

1. sin - trigonometric sine
2. cos - trigonometric cosine
3. tan - trigonometric tangent
4. asin - trigonometric inverse sine (arcsine)
5. acos - trigonometric inverse cosine (arccosine)

6. atan - trigonometric inverse tangent (arctangent)

7. exp - exponential

8. log - natural logarithm Department of Electronics and Communication Engineering Page 6

2. **Vector Functions**: Other MATLAB functions operate essentially on vectors returning a scalar value. Some of these functions are given below.

1. max largest component: get the row in which the maximum element lies

2. min smallest component

3. length of a vector

4. sort in ascending order

5. sum of elements

6. prod product of elements

7. median value

8. mean value std standard deviation

3. **Matrix Functions**: Much of MATLAB s power comes from its matrix functions. These can be further separated into two sub-categories. The first one consists of convenient matrix building functions, some of which are given below.

1. eye - identity matrix

2. zeros - matrix of zeros

3. ones - matrix of ones

4. diag - extract diagonal of a matrix or create diagonal matrices

5. triu - upper triangular part of a matrix

6. tril - lower triangular part of a matrix

MathWorks | Training Services

Course Completion Certificate

Nikhil Negi

has successfully completed **100%** of the self-paced training course

MATLAB Onramp

DIRECTOR, TRAINING SERVICES

9 January 2024

EXPERIMENT-1

AIM: *to study and evaluate the concept of amplitude modulation.*

SOFTWARE USED: *MATLAB (Version R23B)*

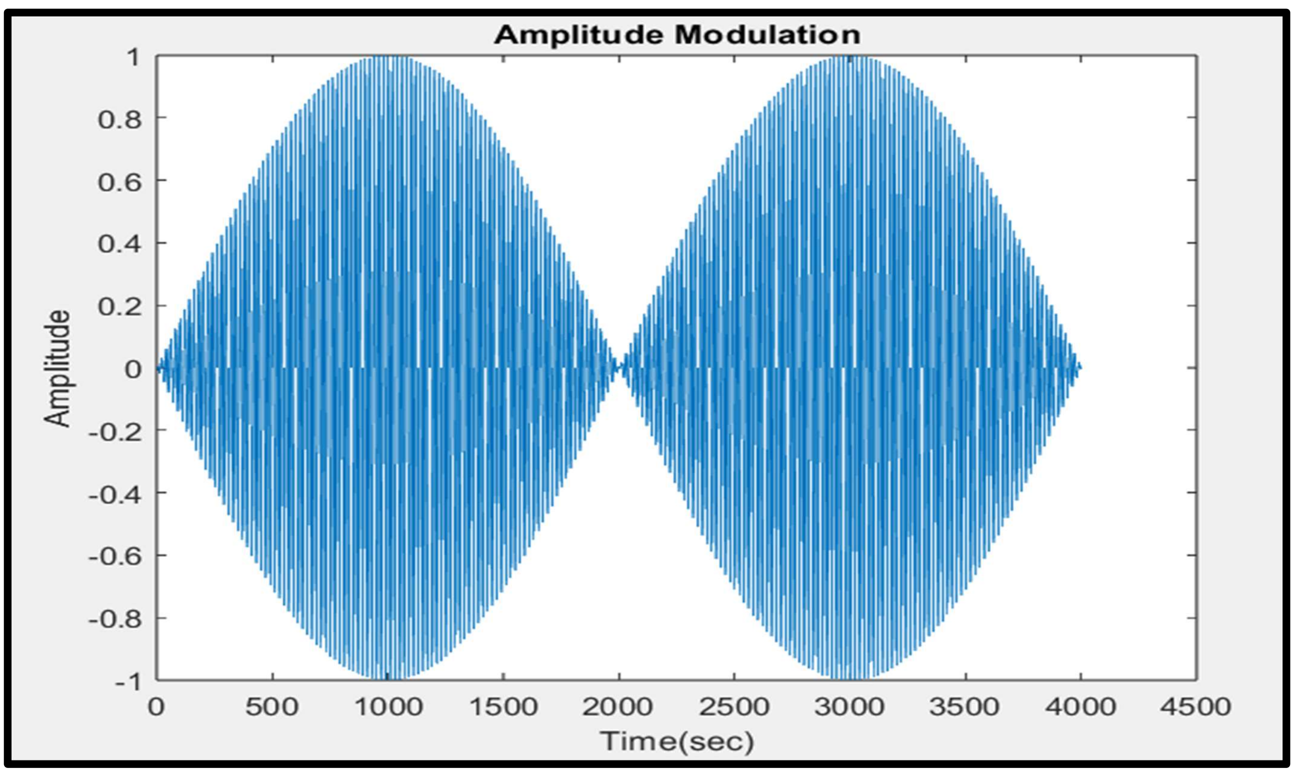
THEORY:

Amplitude modulation, or just AM, is one of the earliest modulation methods that is used in transmitting information over the radio. This technique was devised in the 20th century at a time when Landell de Moura and Reginald Fessenden were conducting experiments using a radiotelephone in the 1900s. After successful attempts, the modulation technique was established and used in electronic communication. In general, amplitude modulation definition is given as a type of modulation where the amplitude of the carrier wave varies in some proportion with respect to the modulating data or the signal. As for the mechanism, when amplitude modulation is used, there is a variation in the amplitude of the carrier. Here, the voltage or the power level of the information signal changes the amplitude of the carrier. In AM, the carrier does not vary in amplitude. However, the modulating data is in the form of signal components consisting of frequencies either higher or lower than that of the carrier. The signal components are known as sidebands, and the sideband power is responsible for the variations in the overall amplitude of the signal. The AM technique is totally different from frequency modulation and phase modulation, where the frequency of the carrier signal is varied in the first case and in the second one, the phase is varied.

CODE:

```
% carrier Frequency
Fc = 200;

% sampling frequency
Fs= 4000;

% time Duration
t = (0 : 1 / Fs : 1);

% sine Wave with time duration of 't'
x = sin(2*pi*t);

% Amplitude Modulation
y = ammod(x, Fc, Fs);

plot(y);
title('Amplitude Modulation');
xlabel('Time(sec)');
ylabel('Amplitude');
```

OUTPUT:



Conclusion:

Hence, I have studied and observe the amplitude modulation.

EXPERIMENT-2

AIM: *to study and evaluate the concept of amplitude modulation*.

SOFTWARE USED: *MATLAB (Version R23B)*

THEORY:

Amplitude Modulation (AM): *Amplitude Modulation (AM) is a modulation technique used in communication systems to transmit information by varying the amplitude of a high-frequency carrier wave according to the characteristics of a modulating signal. This modulating signal typically represents the information to be transmitted, such as voice, data, or video. AM is a fundamental method employed in radio broadcasting, where the modulated signal is transmitted through the airwaves and received by receivers tuned to the carrier frequency. AM signals can be classified into different types based on their spectral characteristics. Double-Sideband (D S B) AM involving the transmission of both the upper and lower sidebands along with the carrier wave, resulting in a signal that occupies a wide bandwidth. Single-Sideband (S S B) AM, on the other hand, transmits only one sideband along with the carrier, effectively conserving bandwidth. Vestigial Sideband (V S B) AM is a compromise between D S B and S B, where a portion of one sideband is transmitted to balance bandwidth efficiency and signal quality.*

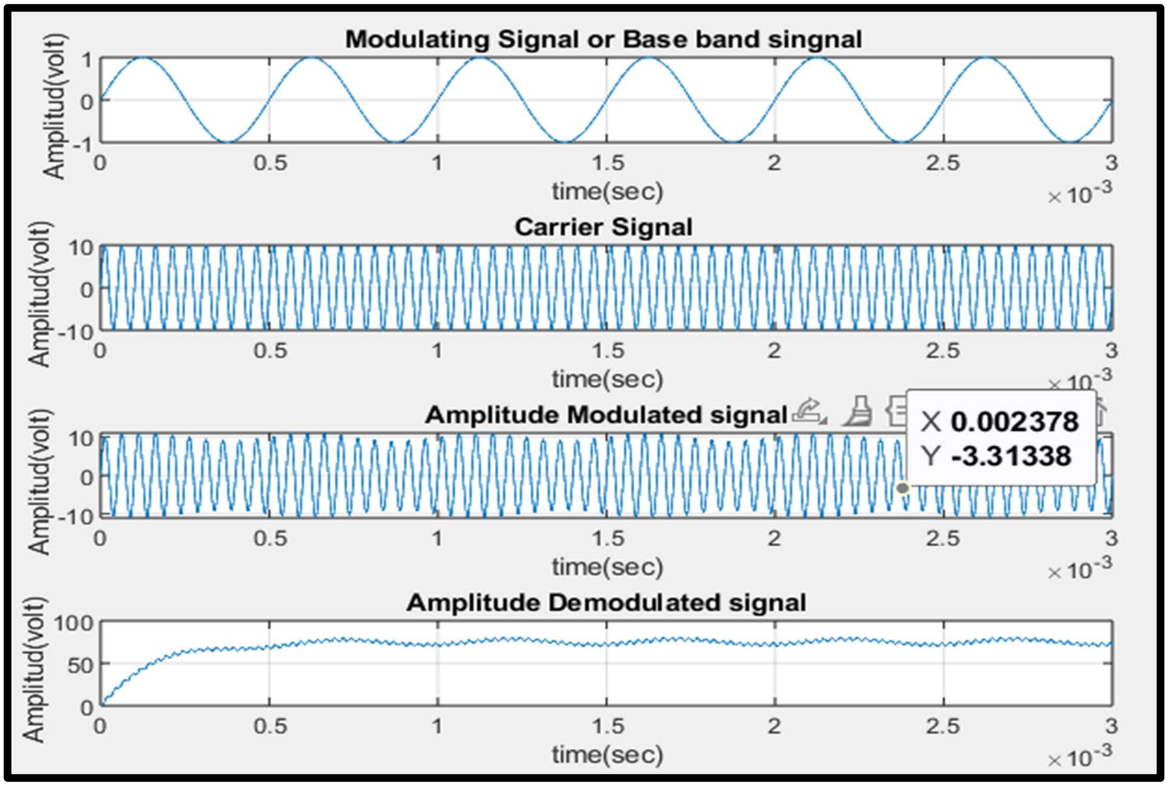
Demodulation of AM: *Demodulation is the process of extracting the original modulating signal from the modulated carrier wave at the receiver end. The primary method used for demodulating AM signals is envelope detection. In envelope detection, the modulated signal is rectified using a diode, which allows only the positive part of the signal* *to pass through. This rectified signal, which represents the envelope of the modulated signal, is then filtered using a low-pass filter to remove the carrier frequency and extract the original modulating signal. Another demodulation technique is coherent detection, which involves multiplying the received signal by a local oscillator signal at the carrier frequency. This multiplication process effectively shifts the modulated signal down to baseband, where it can be filtered to extract the modulating signal.*

CODE:

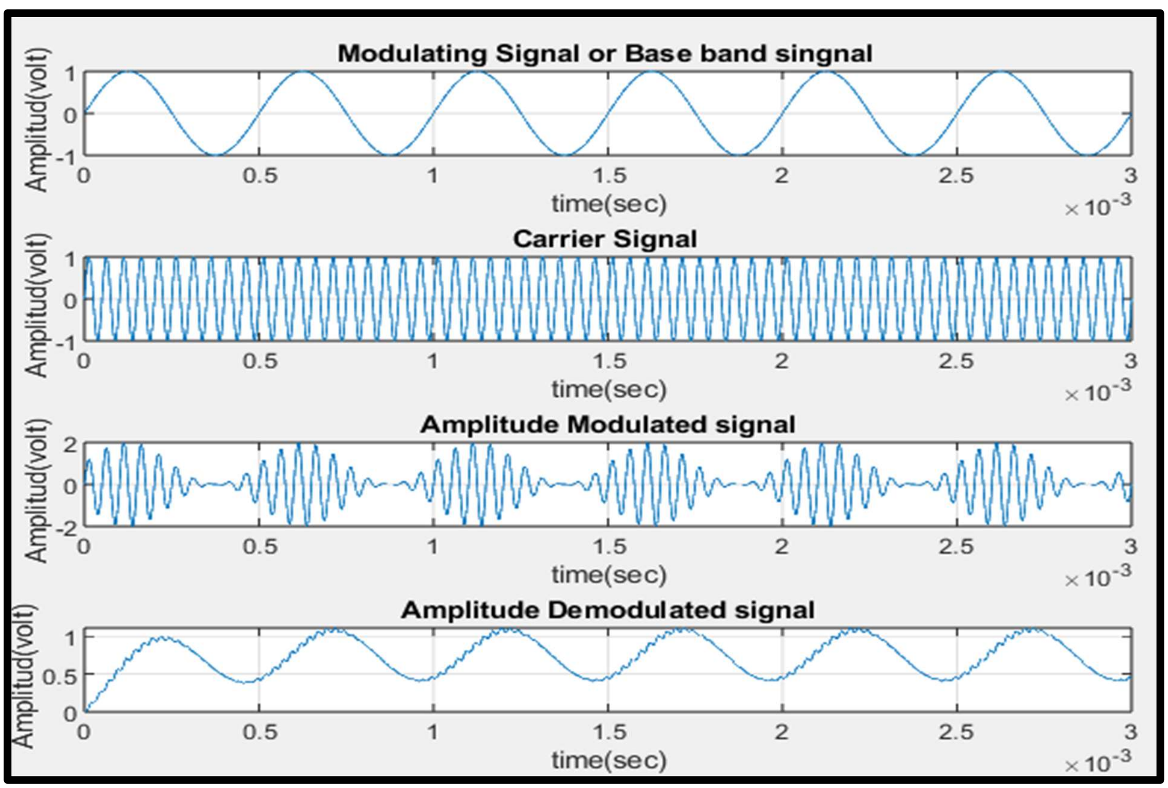
```
clc;
clear all;
close all;
disp('Enter the modulation Index.M<1(under)M>1(over) M=1(100%)modulation');
m=input(' Enter the value of modulation index M = ');
Am=1; % Amplitude of modulating signal
fa=2000; % Frequency of modulating signal
Ta=1/fa; % Time period of modulating signal
t=0:Ta/1000:6*Ta; % Total time for simulation
ym=Am*sin(2*pi*fa*t); % Eqation of modulating signal
figure(1)
subplot(4,1,1);
plot(t,ym), grid on;% plot of Modulating signal
title ( '  Modulating Signal or Base band singnal   ');
xlabel ( ' time(sec) ');
ylabel (' Amplitud(volt)   ');
Ac=Am/m;% Amplitude of carrier signal , modulation Index M=Am/Ac ]
fc=fa*10;% Frequency of carrier signal
Tc=1/fc;% Time period of carrier signal
yc=Ac*sin(2*pi*fc*t);% Eqation of carrier signal
subplot(4,1,2);
plot(t,yc), grid on;% plot of carrier signal
title ( '  Carrier Signal   ');
xlabel ( ' time(sec) ');
ylabel (' Amplitud(volt)   ');
y=Ac*(1+m*sin(2*pi*fa*t)).*sin(2*pi*fc*t); % Equation of Amplitude
subplot(4,1,3);
plot(t,y);% Graphical representation of AM signal
title ( '  Amplitude Modulated signal  ');
grid on;
xlabel ( ' time(sec) ');
ylabel (' Amplitud(volt)   ');
z=y.*yc;
[b a]=butter(1,0.001); %BUTTERWORTH FILTER
lpf=filter(b,a,z);
Rx=1.5.*lpf; %Amplification
subplot(4,1,4);
plot(t,Rx);% Graphical representation of AM signal
title ( '  Amplitude Demodulated signal  ');
grid on;
xlabel ( ' time(sec) ');
ylabel (' Amplitud(volt)   ');
```

Simulation RESULTS:

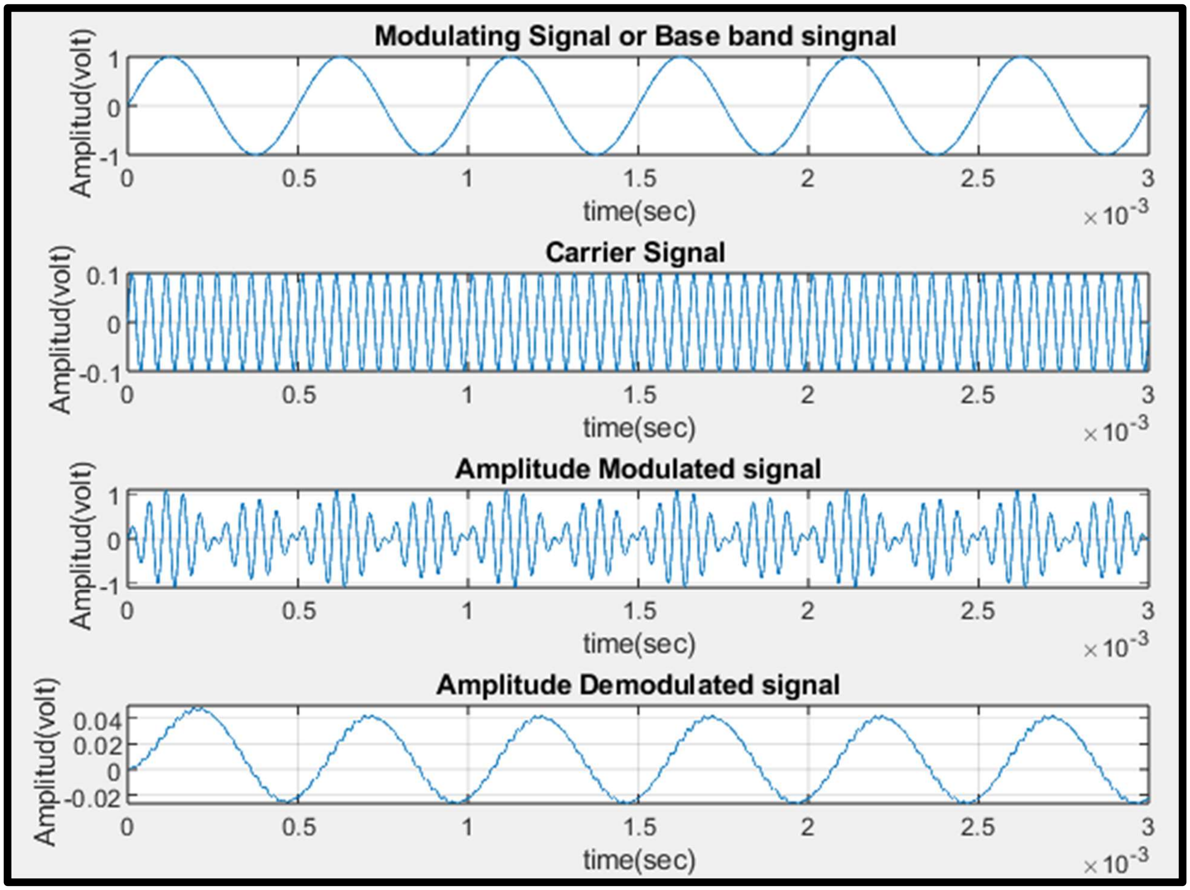
(a) Modulation index < 1



(b) Modulation index = 1



(c) Modulation index > 1



Conclusion:

Hence, I have studied and observe the amplitude modulation by varying Modulation index.

EXPERIMENT-3

AIM: *to study and evaluate the concept of DSB-SC modulation.*

SOFTWARE USED: *MATLAB (Version R23B)*

THEORY: *Double-Sideband Suppressed Carrier Modulation (D S B -S C) is a modulation technique used in communication systems. Unlike traditional AM, D S B - S C only transmits the upper and lower sidebands of the modulating signal, effectively eliminating the carrier signal. This method conserves b2andwidth, making it advantageous in scenarios where bandwidth efficiency is critical. The modulation process involves multiplying the message signal by the carrier wave, resulting in a D S B - S C modulated signal. Mathematically, this can be expressed as $s(t)=m(t)\cdot\cos(2\pi fct)$, where m(t) represents the message signal and fc is the carrier frequency. B y suppressing the carrier, D S B -S C modulation reduces spectral redundancy and optimizes bandwidth utilization. Demodulation of D S B -S C signals typically employs coherent detection, where the received signal is multiplied by a local oscillator at the carrier frequency. This process shifts the signal down to baseband, allowing for the extraction of the original message signal through filtering. However, coherent detection may be sensitive to phase errors, requiring precise synchronization between transmitter and receiver. Despite the need for synchronization, D S B -S C modulation offers advantages in spectral efficiency and bandwidth conservation. It finds applications in various communication systems, including radio broadcasting, telecommunication, and audio transmission. D S B -S C modulation provides an efficient means of transmitting information while minimizing* *spectral redundancy, making it suitable for bandwidth constrained environments.*

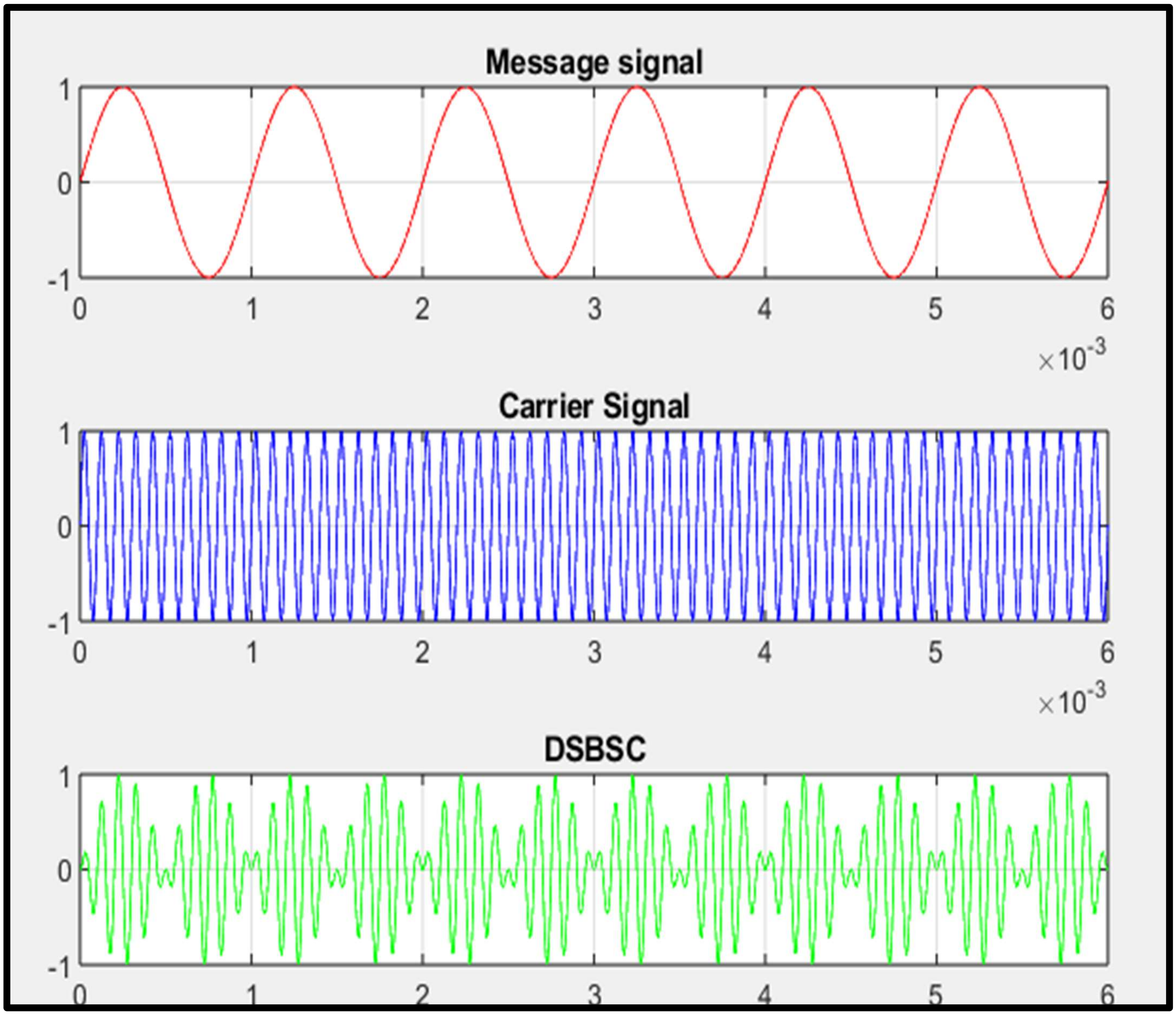
CODE:

```
function amplitude = CE_exp_3
fm = input('Enter the value of message signal frequency: ');
fc = input('Enter the value of carrier signal frequency: ');
Am = input('Enter the value of message signal amplitude: ');
Ac = input('Enter the value of carrier signal amplitude: ');
Tm = 1/fm;
t1 = 0:Tm/999:6*Tm;
message_signal = Am*sin(2*pi*fm*t1);
subplot(3,1,1)
plot(t1, message_signal, 'r');
grid();
title('Message signal');
carrier_signal = Ac*sin(2*pi*fc*t1);
subplot(3,1,2)
plot(t1, carrier_signal, 'b');
grid();
title('Carrier Signal');
amplitude = message_signal.*carrier_signal;
subplot(3,1,3)
plot(t1,amplitude, 'g');
grid();
title('DSBSC');
end
```

INPUT:

```
Enter the value of message signal frequency: 1000
Enter the value of carrier signal frequency: 10000
Enter the value of message signal amplitude: 1
fx Enter the value of carrier signal amplitude: 1
```

OUTPUT:



Conclusion:

Hence, I have studied and observe the Double-Sideband Suppressed Carrier Modulation (DSB-SC).

EXPERIMENT-4

AIM: *to study and evaluate the concept of SSB-SC modulation and demodulation.*

SOFTWARE USED: *MATLAB (Version R23B)*

THEORY: *Single-Sideband Suppressed Carrier Modulation (SSB-SC) is a modulation technique used in communication systems to transmit only one of the sidebands of the modulating signal while suppressing the carrier and the other sideband. This approach effectively reduces bandwidth consumption compared to traditional AM and DSB-SC modulation.*

The modulation process of SSB-SC involves filtering out one of the sidebands and the carrier signal from the DSB-SC modulated signal, leaving only the desired sideband. This can be achieved using a variety of methods such as frequency mixing, phasing methods, or filter techniques.

Mathematically, SSB-SC modulation can be expressed as:

$s(t) = m(t) \cdot \cos(2\pi f_c t)$ for upper sideband

$s(t) = m(t) \cdot \sin(2\pi f_c t)$ for lower sideband.

Where s(t) is the modulated signal, m(t) is the message signal, fc is the carrier frequency, and the modulated signal is either multiplied by cosine or sine depending on the desired sideband.

SSB-SC modulation offers improved spectral efficiency compared to DSB-SC modulation since it eliminates the redundant sideband, allowing for better utilization of the available bandwidth. However, demodulation of

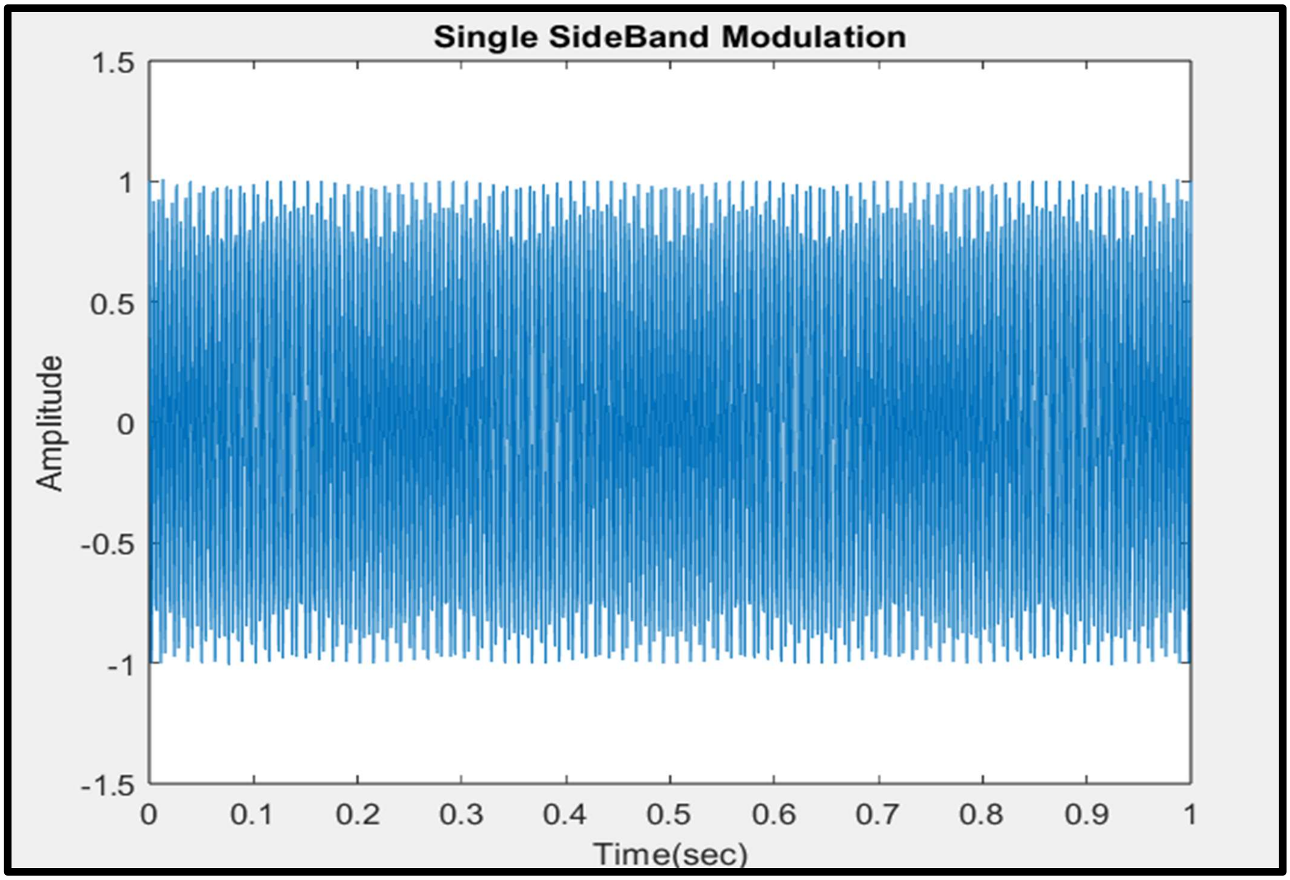
SSB-SC signals requires more complex techniques compared to DSB-SC, often involving coherent detection and precise frequency synchronization to recover the original message signal.

Despite its complexity, SSB-SC modulation is widely used in communication systems where bandwidth efficiency is critical, such as in radio broadcasting, telecommunication, and audio transmission, due to its ability to conserve bandwidth while transmitting information effectively.

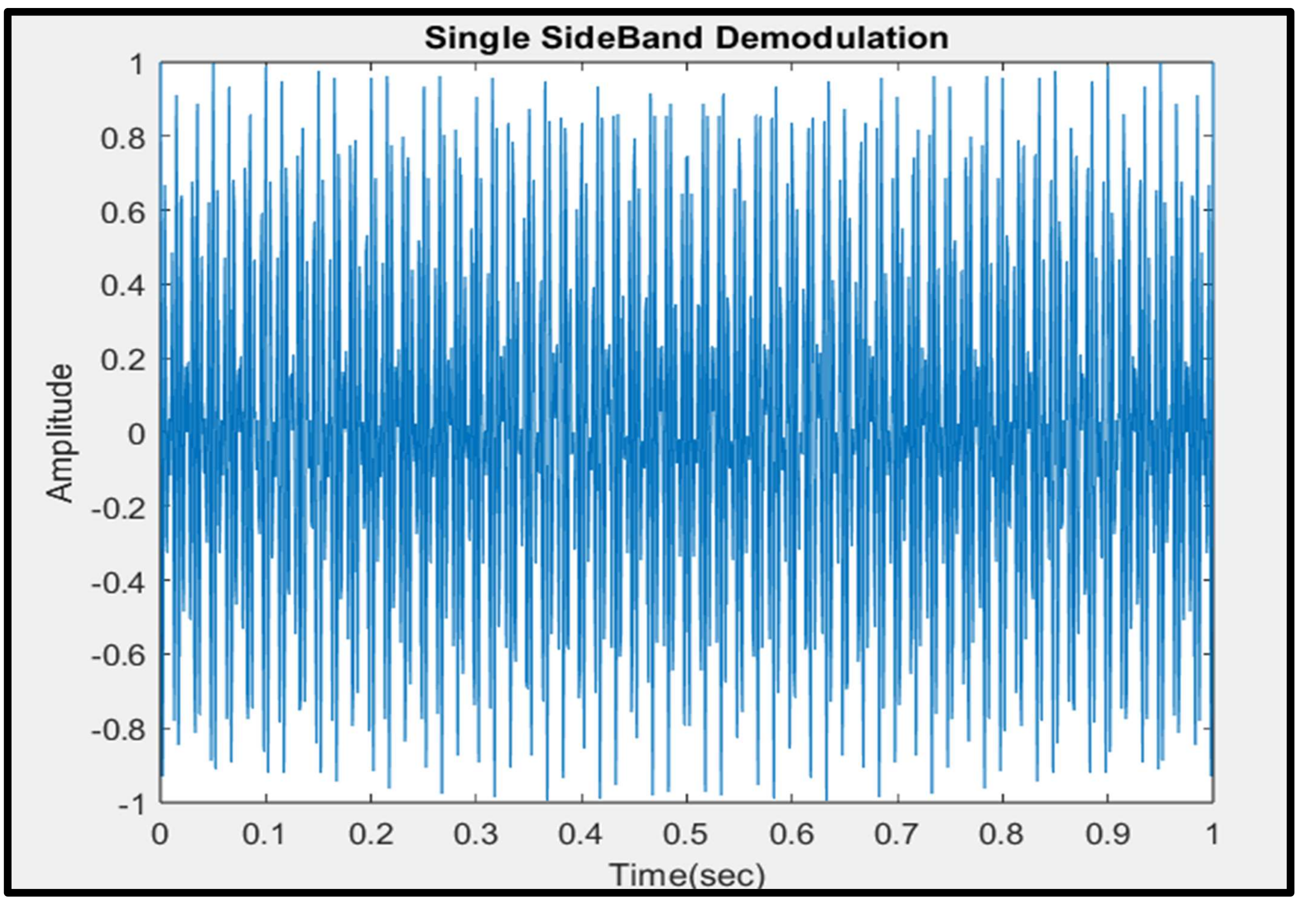
CODE:

```
fc = 200;
fm = 30;
fs= 4000;
t = linspace(0, 1, 1000);
m = cos(2 * pi * fm * t);
mh = imag(hilbert(m));
sb = m .* cos(2 * pi * fc * t) - mh .* sin(2 * pi * fc * t);
em = sb.*m;
[ n,w ] = buttord(2/1000,4/1000, .5, 5);
[ a,b ] = butter(n,w,'low');
dem = filter(a,b,em);
figure;
plot(t, sb);
title('Single SideBand Modulation');
xlabel('Time(sec)');
ylabel('Amplitude');
figure;
plot(t, em);
title('Single SideBand Demodulation');
xlabel('Time(sec)');
ylabel('Amplitude');
figure;
plot(t, dem);
title('Filtered Signal');
xlabel('Time(sec)');
ylabel('Amplitude');
```

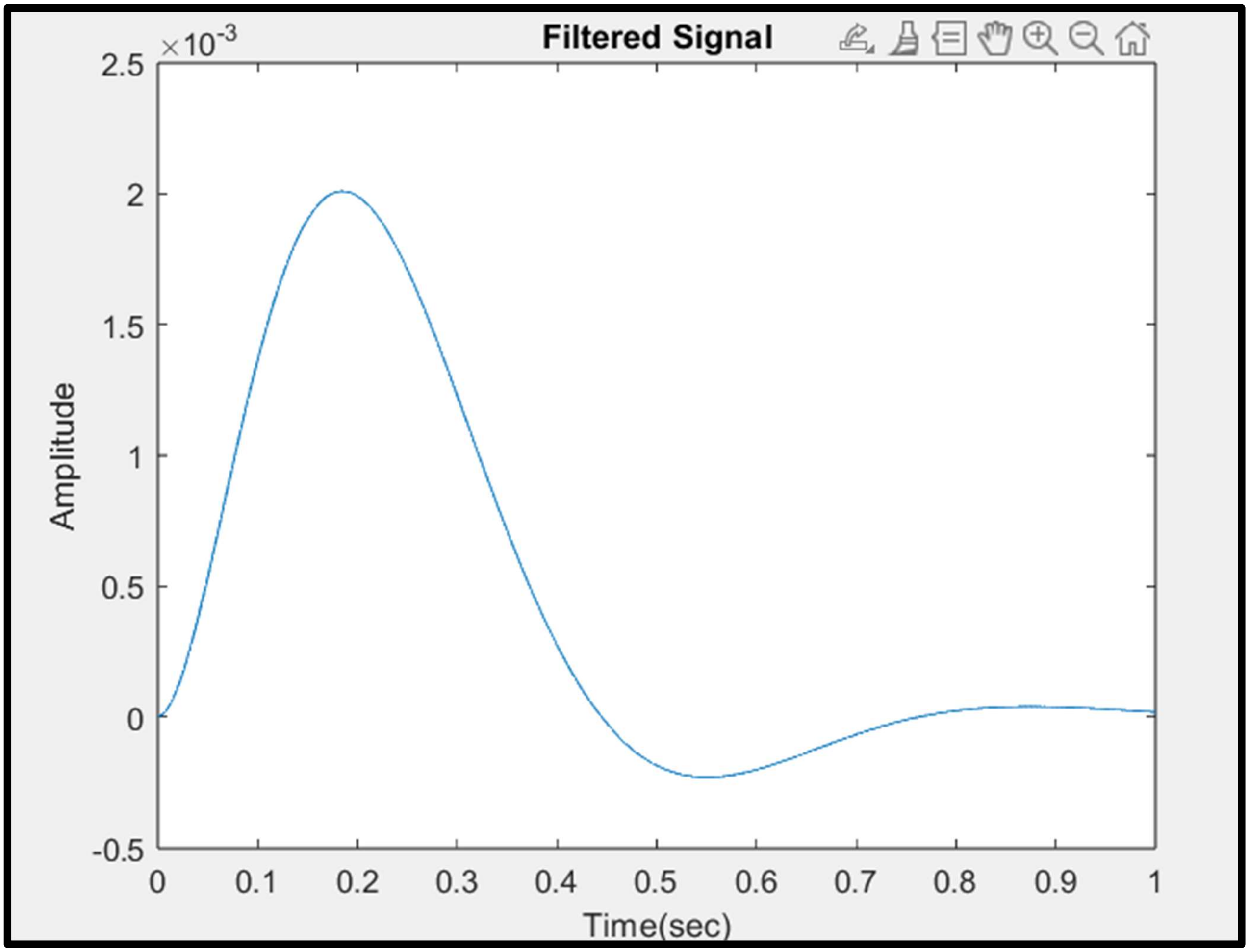
OUTPUT:



Single SideBand Modulation
Amplitude
Time(sec)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Single SideBand Demodulation
Amplitude
Time(sec)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Conclusion:

Hence, I have studied and observe the Single-Sideband Suppressed Carrier Modulation (DSB-SC).

EXPERIMENT-5

AIM: *to study and evaluate the concept of Frequency Modulation*

SOFTWARE USED: *MATLAB (Version R23B)*

THEORY: *Frequency Modulation (FM) is a modulation technique used in communication systems to encode information in the frequency of a carrier wave. Unlike amplitude modulation (AM), where the amplitude of the carrier signal is varied, in FM, the frequency of the carrier wave is varied in proportion to the amplitude of the modulating signal.*

The process of frequency modulation can be described as follows:

$s(t) = Ac * cos(2\pi fct + \beta * m(t))$

Where s(t) is the modulated signal, Ac is the carrier amplitude, fc is the carrier frequency, m(t) is the message signal, and β is the modulation index, which determines the degree of frequency deviation from the carrier frequency based on the amplitude of the modulating signal.

FM offers several advantages over AM, including improved signal-to-noise ratio (SNR), which results in better signal quality and less susceptibility to noise and interference. FM signals also have a constant amplitude, which reduces distortion and improves fidelity compared to AM signals.

To demodulate an FM signal, a frequency discriminator or phase-locked loop (PLL) can be used. These circuits detect the frequency variations in the FM signal and convert them back to the original modulating signal.

FM is widely used in various communication systems, including radio broadcasting, two-way radio communication, and analog television transmission, due to its superior noise performance and signal quality.

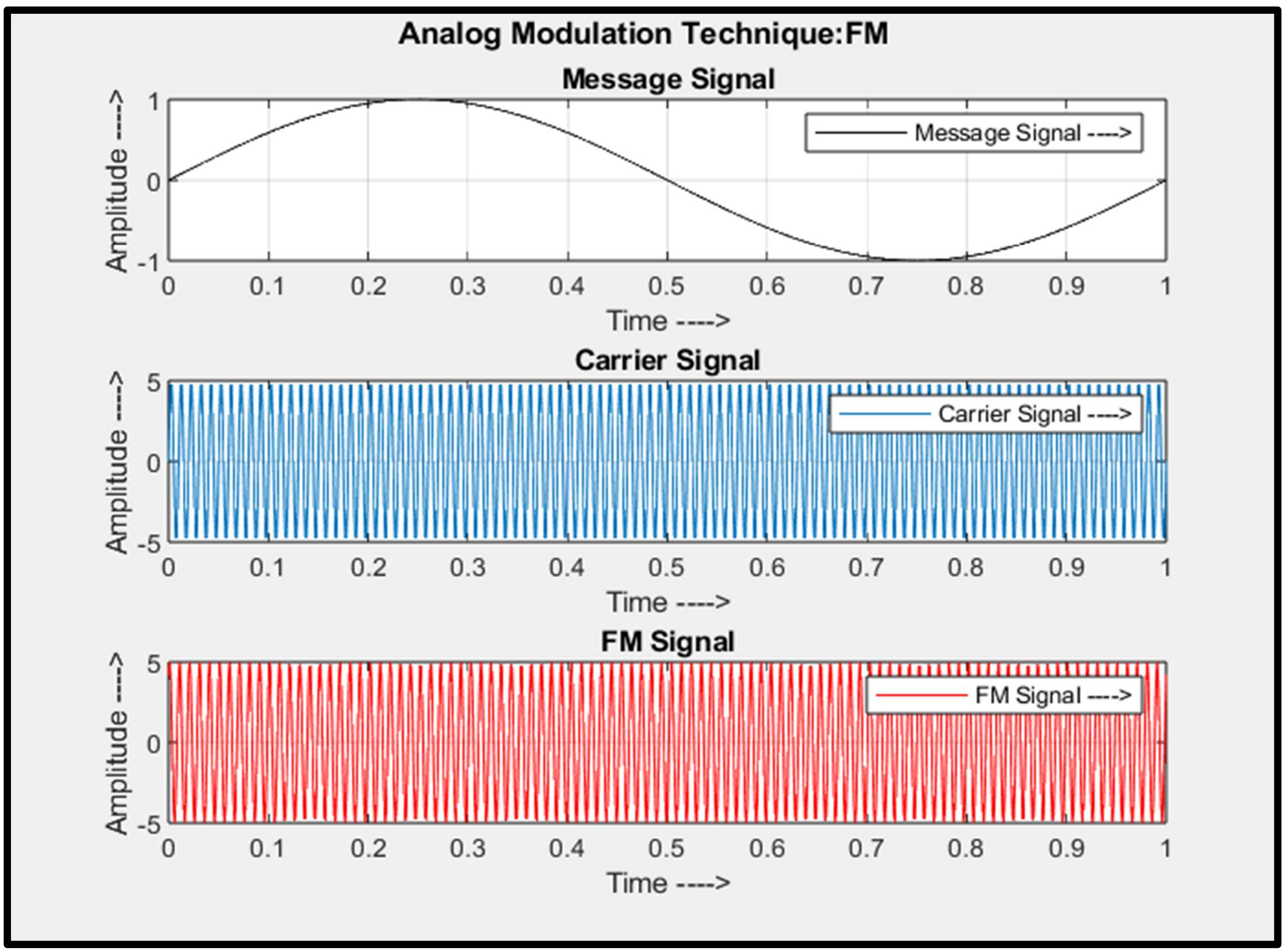
CODE:

```
clc;
clear;
t = 0:0.001:1;
Message_Signal_Amplitude = input('Enter the Amplitude of Message Signal = '); %1
Carrier_Signal_Amplitude = input('Enter the Amplitude of Carrier Signal = '); %5
Message_Signal_Frequency = input('Enter the Message frequency = '); %1
Carrier_Signal_Frequency = input('Enter the Carrier frequency = '); %100
m = input('Enter the Modulation Index = '); %1
sm = Message_Signal_Amplitude*sin(2*pi*Message_Signal_Frequency*t);
subplot(3,1,1);
plot(t,sm,'black');
xlabel('Time ---->');
ylabel('Amplitude ---->');
title('Message Signal');
legend('Message Signal ---->');
grid on;
sc = Carrier_Signal_Amplitude*sin(2*pi*Carrier_Signal_Frequency*t);
subplot(3,1,2);
plot(t,sc);
xlabel('Time ---->');
ylabel('Amplitude ---->');
title('Carrier Signal');
legend('Carrier Signal ---->');
grid on;
y = Carrier_Signal_Amplitude*sin(2*pi*Carrier_Signal_Frequency*t+m.*cos(2*pi*Message_Signal_Frequency*t));
subplot(3,1,3);
plot(t,y,'red');
xlabel('Time ---->');
ylabel('Amplitude ---->');
title('FM Signal');
legend('FM Signal ---->');
grid on;
ha = axes ('Position',[0 0 1 1],'Xlim',[0 1],'Ylim',[0 1],'Box','off','Visible','off','Units','normalized', 'clipping' , 'off');
text (0.5, 1,'\bf Analog Modulation Technique:FM ','HorizontalAlignment','center','VerticalAlignment', 'top')
```

INPUT:

```
Command Window
Enter the Amplitude of Message Signal = 1
Enter the Amplitude of Carrier Signal = 5
Enter the Message frequency = 1
Enter the Carrier frequency = 100
Enter the Modulation Index = 1
fx >>
```

OUTPUT:



Conclusion:

Hence, I have studied and observe the concept of Frequency Modulation.