Islamic University of Technology

Name	: Shazzad Ahmed Chowdhury
Student ID	: 220021108
Section	: A-(02)
Department	: Electrical & Electronic Engineering
Course No	: EEE-4404
Course Title	: Communication Engineering I Lab
Experiment No	: 01
Experiment Name	: Generation and Detection of AM Waves
Date of Performance	: 19/05/2025
Date of Submission	: 22/06/2025

Objectives:

1. Understanding the principle of amplitude modulation (AM).
2. Understanding the waveform and frequency spectrum of AM signals and calculating the percentage of modulation.

3. Understand the generation & detection of AM waveforms.

Part A: Hardware

Apparatus Required:

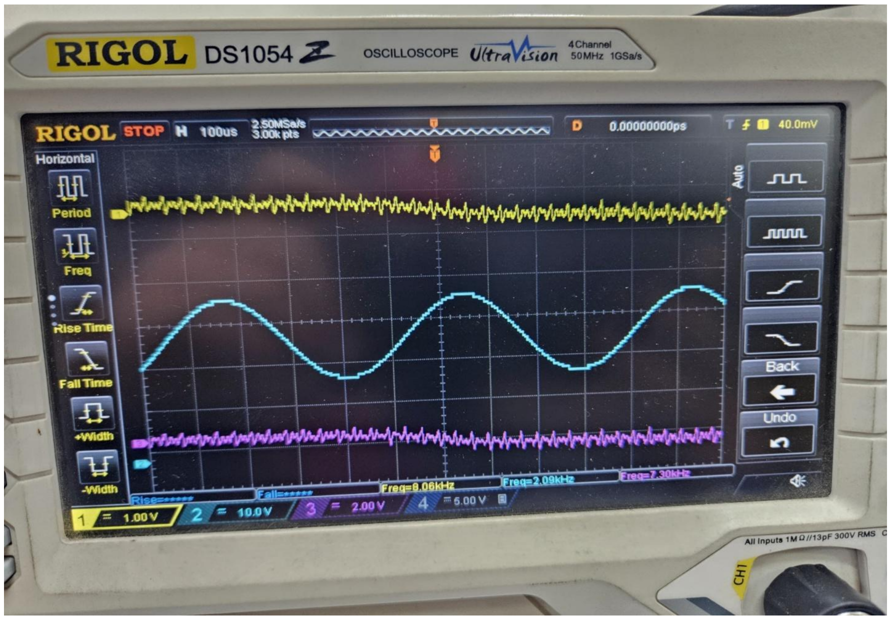
• Emona ETT-101/C Trainer
• Dual-channel 20 MHz Oscilloscope
• Patch leads and BNC connectors
• Headphones (optional for audio check)

Theory:

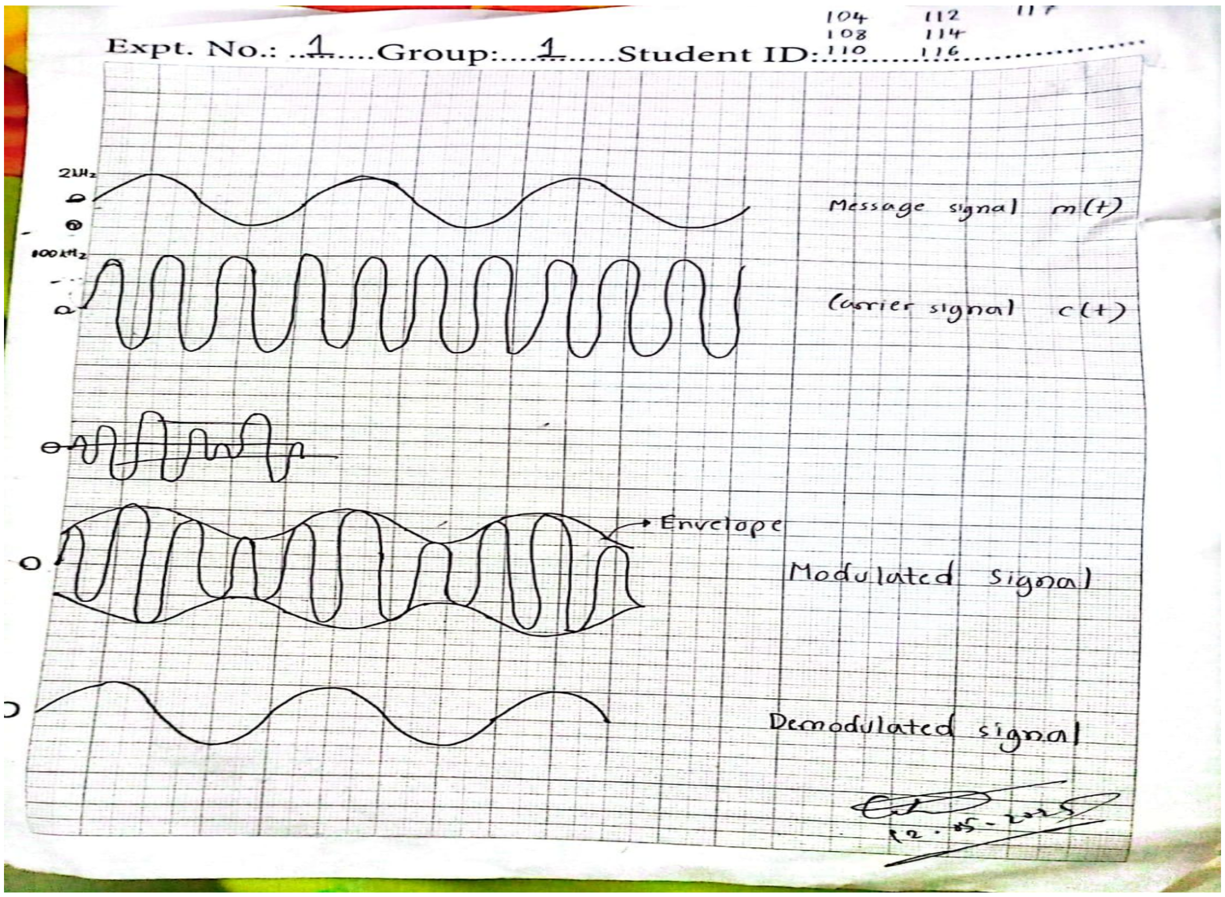
To effectively transmit any type of signal, modulation is essential. In this lab, we will focus on a technique known as amplitude modulation (AM). As the name implies, this method involves varying the amplitude of a high-frequency carrier wave based on the information signal. In an AM communication system, audio inputs like speech or music are first converted into electrical signals using devices such as microphones—these are known as message or baseband signals. These signals then modulate the amplitude of a high-frequency sine wave, called the carrier. To retrieve the original message, we will later use a receiver equipped with an envelope detector to demodulate the amplitude-modulated signal.

Graph Analysis:

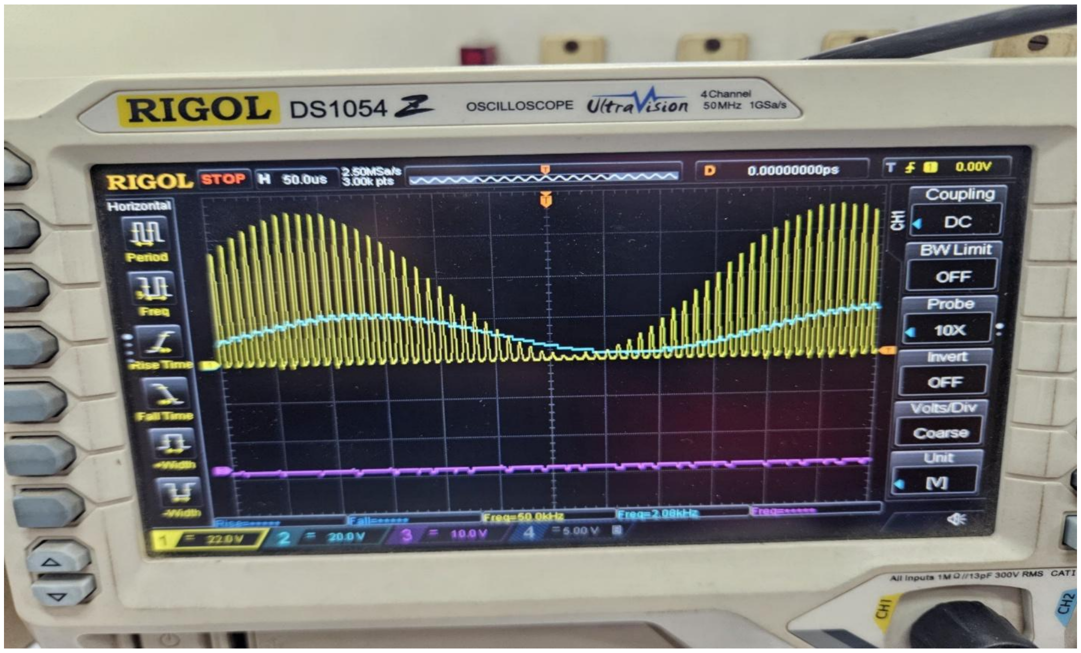
Message Signal m(t) –



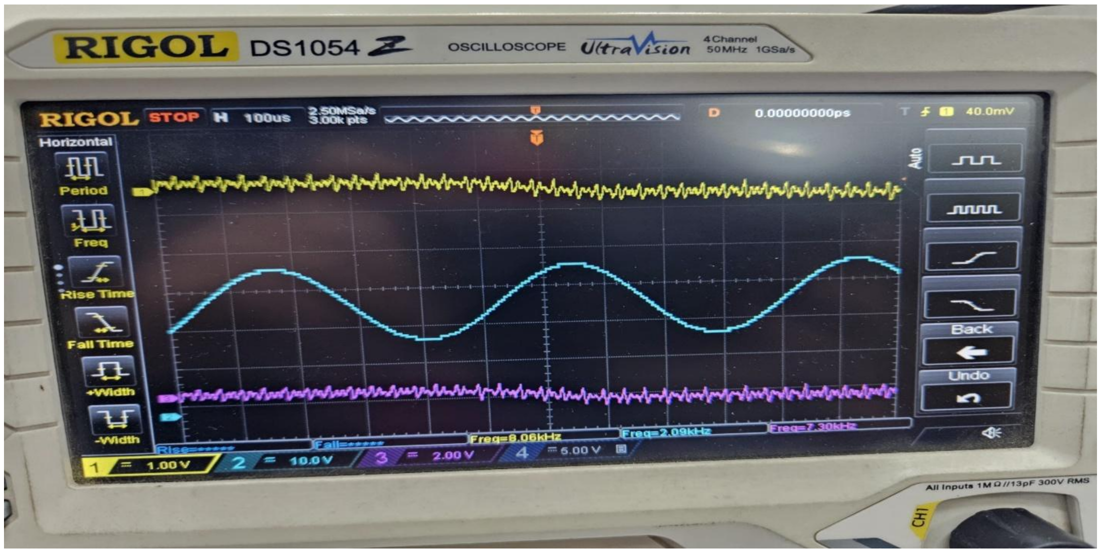
Graph:



Recovered message signal after demodulation –



Coupling DC
BW Limit OFF
Probe 10X
Invert OFF
Volts/Div Coarse
Unit [V]



By applying an envelope detector and then passing the output through a low-pass filter, we can recover the original signal from the modulated waveform.

<u>Report Questions:</u>

(1) Modulation is basically the process of shaping a high-frequency signal (called the carrier) using our actual message signal so that it can be transmitted more efficiently. In Amplitude Modulation (AM), we change the carrier's amplitude based on the message. To create an AM signal, we first add a DC offset to the message signal to avoid any unwanted phase flips, then we multiply it with the carrier wave. Later, we demodulate the signal to retrieve the original message.

(2)What happens if we over-modulate –
When we push the modulation depth too far (above 1), the signal starts to get distorted. In the time domain, this means the peaks of the wave get cut off, especially the negative ones, which leads to loss of information. The distorted shape also makes it harder for the receiver to pick up the message properly.
In the frequency domain, over-modulation causes unwanted extra frequencies to appear because of the clipping, and it also ends up using more bandwidth than necessary.

(3) Over-modulation should be avoided because it leads to several issues:

❖ The signal becomes less clear, making it harder to understand.
❖ The receiver struggles to properly decode the message, which results in distortion.

- ❖ It can cause interference with nearby frequencies, disrupting other transmissions.
- ❖ It also uses more power than necessary, making the system less efficient.

(4) The Adder module combines the message signal with a DC component to create a signal suitable for amplitude modulation (AM). Since the DC component is steady, there should ideally be no phase difference between the two signals. However, if a phase shift is noticed, it could be caused by imperfections in the hardware—like circuit delays. By fine-tuning the adder's gain settings, we can ensure the signals blend correctly without introducing unwanted phase shifts.

(5) The amplitude of the carrier wave changes based on how strong the message signal is.

- ❖ When the message amplitude is too low, the modulation is shallow, resulting in a weak signal.
- ❖ When it's just right, you get clear modulation with a well-defined envelope and no distortion.
- ❖ But if the message amplitude is too high, it causes over-modulation, leading to a distorted signal.
  By adjusting the gain on the Adder, we could see how increasing or decreasing the message amplitude directly affected the modulation depth.

(6) When an AM signal is over-modulated, several problems arise:

- ❖ The signal's negative cycles can flip, making it impossible to demodulate properly.
- ❖ It introduces harmonic distortion, which adds extra noise and interference.
- ❖ The receiver struggles to recover the original message, hurting the overall quality of communication.

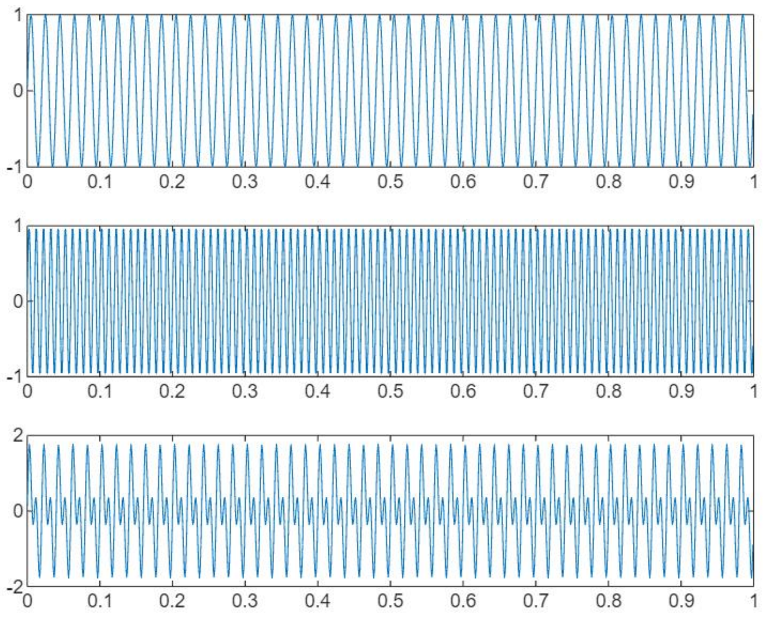
(7) After demodulating the AM signal, the recovered message still had some ripple noise, caused by imperfections in the envelope detector. To clean up the signal:

- ❖ An RC low-pass filter was used to remove high-frequency noise.
- ❖ The cutoff frequency was fine-tuned to around 2.6 kHz, which removed the ripples without affecting the clarity of the message.
- ❖ We compared the signal before and after filtering to confirm the improvement.

Part B: Software

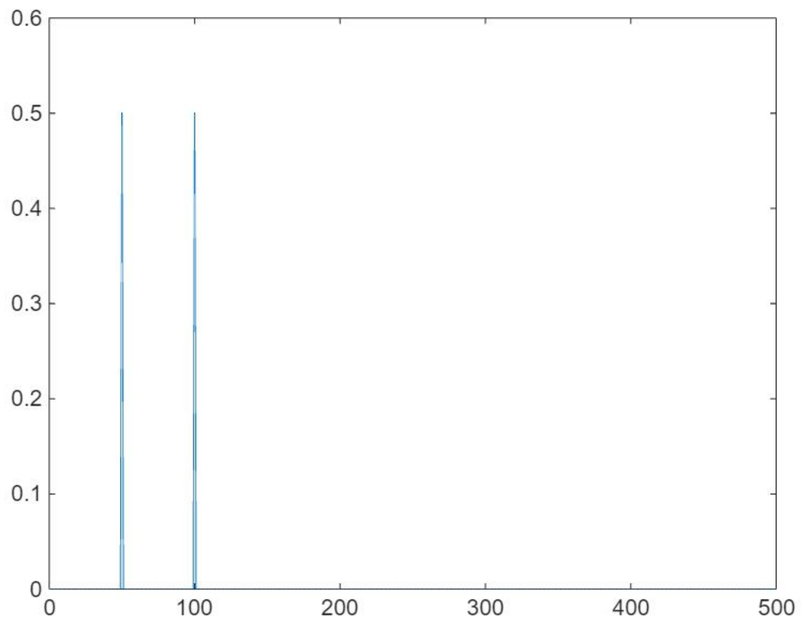
Code-01:

```
clear all
clc
% Sampling parameters
Fs = 1000;          % Sampling frequency (Hz)
T = 1;              % Time Period
t = 0:1/Fs:T-1/Fs;
f1 = 50;
f2 = 100;
x1 = sin(2*pi*f1*t);
x2 = sin(2*pi*f2*t);
% New Signal
x = x1 + x2;
% Plot the individual and combined signals
figure;
subplot(3,1,1);
plot(t, x1);
subplot(3,1,2);
plot(t, x2);
subplot(3,1,3);
plot(t, x);
```



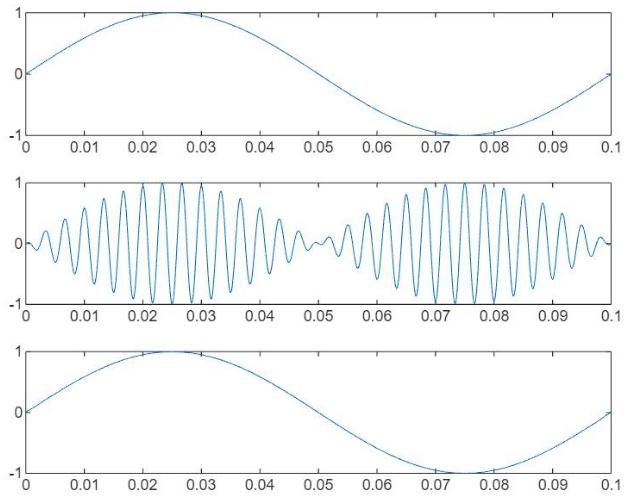
```
% Compute and plot the frequency spectrum
N = length(x);
X = fft(x);                       % FFT of combined signal
```

```
X_mag = abs(X)/N;              % Normalize magnitude
f = (0:N-1)*(Fs/N);            % Frequency vector
% Plot only positive frequencies
half = floor(N/2);
figure;
plot(f(1:half), X_mag(1:half));
```



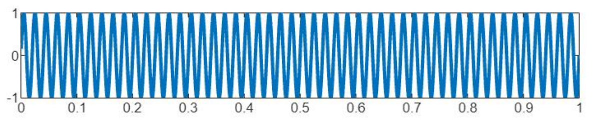
Code-02:

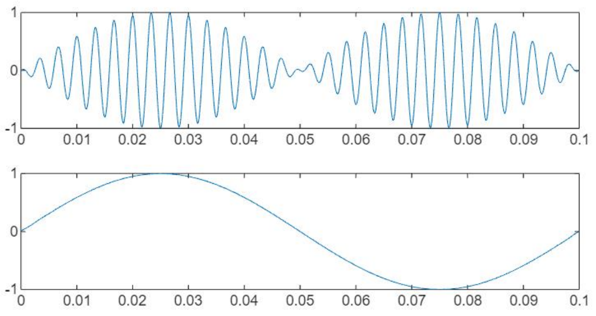
```
% Initialization of Parameters
T = 0.1;
fs = 8000; % Sampling Frequency
t = 0:1/fs:T; % Taking 1000 sample points between 0 to 0.1 sec
Vm = 1; % Amplitude
fm = 10; % Modulating signal frequency
m = Vm * sin(2 * pi * fm * t); % Modulating Signal
% ----- AM Modulation -----
fc = 300; % Carrier Signal
v = ammod(m, fc, fs); % Modulated Signal
% ----- AM Demodulation -----
mr = amdemod(v, fc, fs); % Demodulated Signal
% ----- Plotting Signals -----
figure(1);
subplot(3,1,1);plot(t, m);
subplot(3,1,2);plot(t,v);
subplot(3,1,3);plot(t,mr);
```



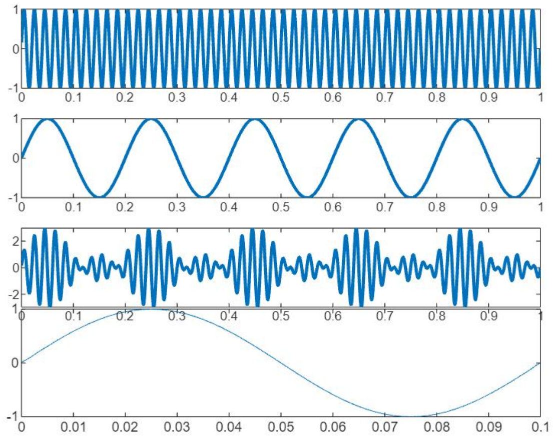
Code-03:

```
% Initialization of parameters
fs = 2000;          % Sampling frequency
f = 5;              % Signal frequency
fc = 50;            % Carrier frequency
N = 2000;           % Use 1 sec of data
t = (1:N)/fs;       % Time axis for plotting
% ----- AM Modulation -----
w = 2*pi*fc*t;      % Carrier frequency = 250 Hz
w1 = 2*pi*f*t;      % Signal frequency = 5 Hz
vc = sin(w);
figure(1);
subplot(4,1,1)
plot(t, vc, 'LineWidth', 2);
```

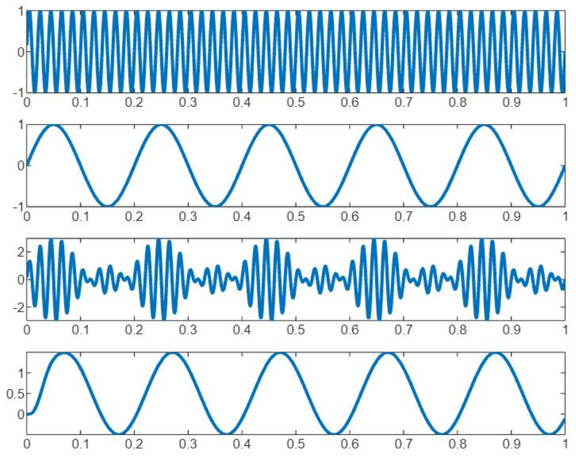




```
vsig = sin(w1);
figure(1);
subplot(4,1,2)
plot(t, vsig, 'LineWidth', 2);
vm = (1 + 2 * vsig) .* vc; % Modulation constant = 0.5
figure(1);
subplot(4,1,3)
plot(t, vm, 'LineWidth', 2); % Plot AM Signal
```

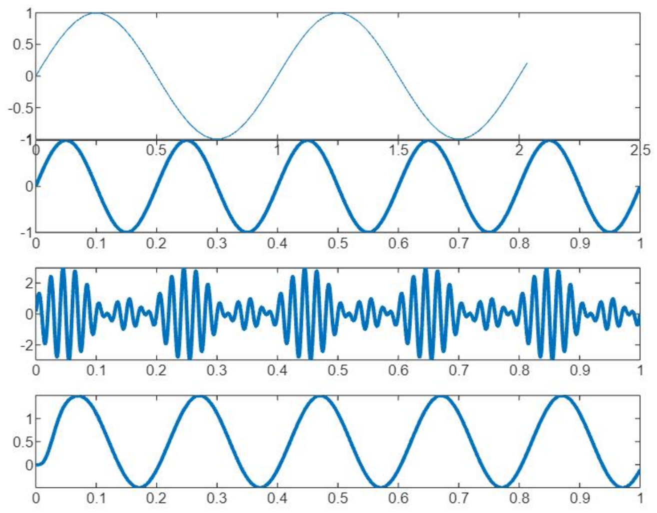


```
% ----- AM Demodulation -----
v1 = vc .* vm; % Multiplier
wn = 0.02; % Lowpass filter cut-off frequency
[b, a] = butter(4, wn); % Design lowpass filter
vout = filter(b, a, v1); % Apply lowpass filter
figure(1);
subplot(4,1,4);
plot(t, vout, 'LineWidth', 2);
```

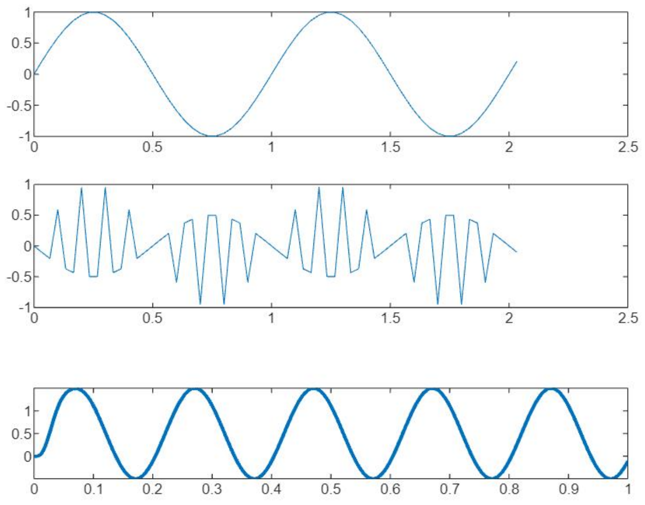


Code-04:

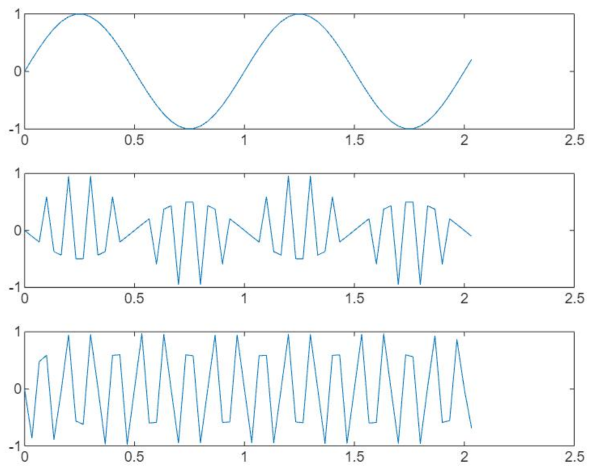
```
% Sample the signal 100 times per second, for 2 seconds.
Fs = 30;
t = [0:2*Fs+1]'/Fs;
Fc = 10;           % Carrier frequency
x = sin(2*pi*t);    % Sinusoidal signal
% Modulate x using single- and double-sideband AM.
ydouble = ammod(x,Fc,Fs);
ysingle = ssbmod(x,Fc,Fs);
figure(1)
subplot(3,1,1); plot(t, x);
```



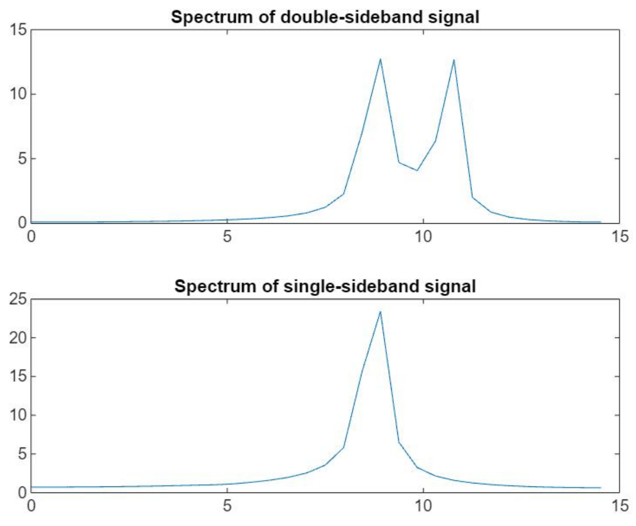
```
subplot (3,1,2); plot (t, ydouble);
```



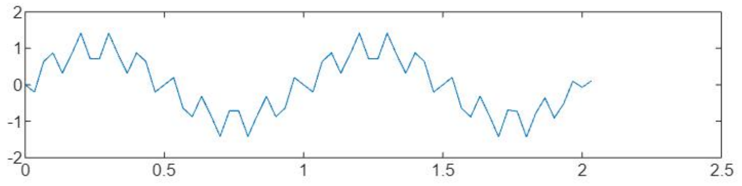
```
subplot (3,1,3); plot (t, ysingle);
```

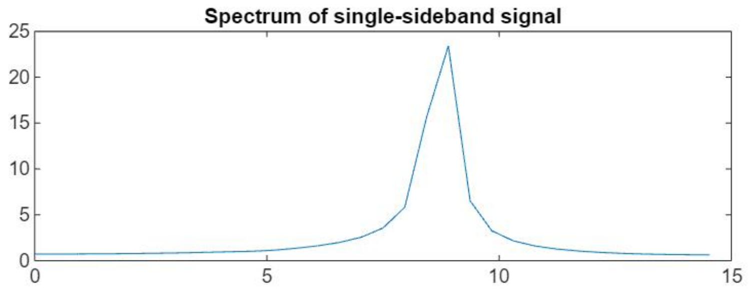


```
% Compute spectra of both modulated signals.
zdouble = fft(ydouble);
zdouble = abs(zdouble(1:length(zdouble)/2+1));
frqdouble = [0:length(zdouble)-1]*Fs/length(zdouble)/2;
zsingle = fft(ysingle);
zsingle = abs(zsingle(1:length(zsingle)/2+1));
frqsingle = [0:length(zsingle)-1]*Fs/length(zsingle)/2;
% Plot spectra of both modulated signals.
figure(2);
subplot(2,1,1); plot(frqdouble, zdouble);
title('Spectrum of double-sideband signal');
subplot(2,1,2); plot(frqsingle, zsingle);
title('Spectrum of single-sideband signal');
```

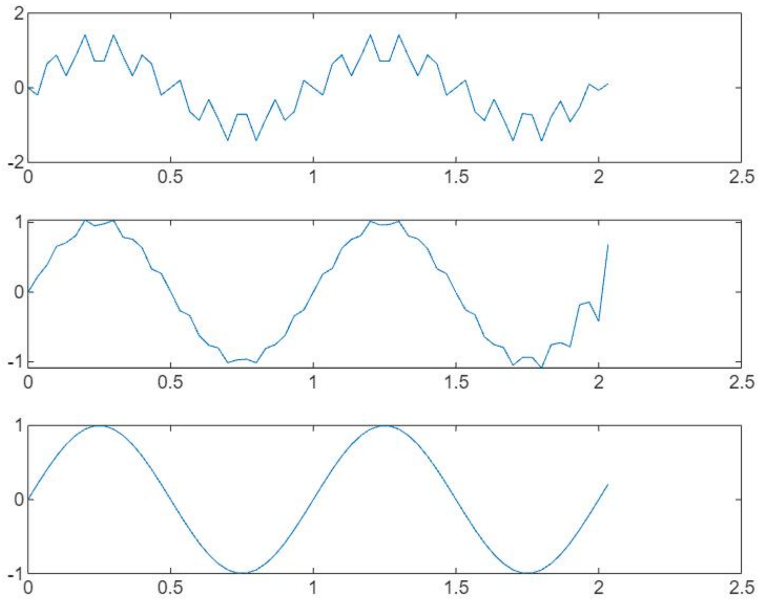


```
% Demodulation and plotting
ydouble1 = amdemod(ydouble, Fc, Fs);
ysingle1 = ssbdemod(ysingle, Fc, Fs);
figure(3)
subplot(3,1,1); plot(t, ydouble1);
```

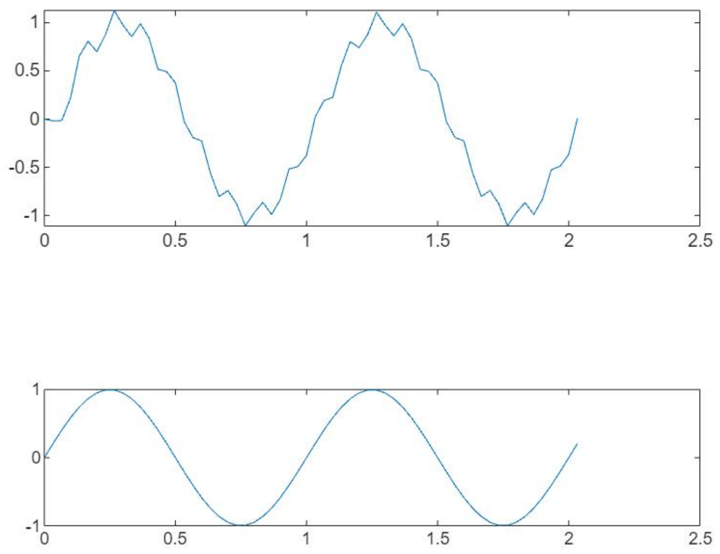




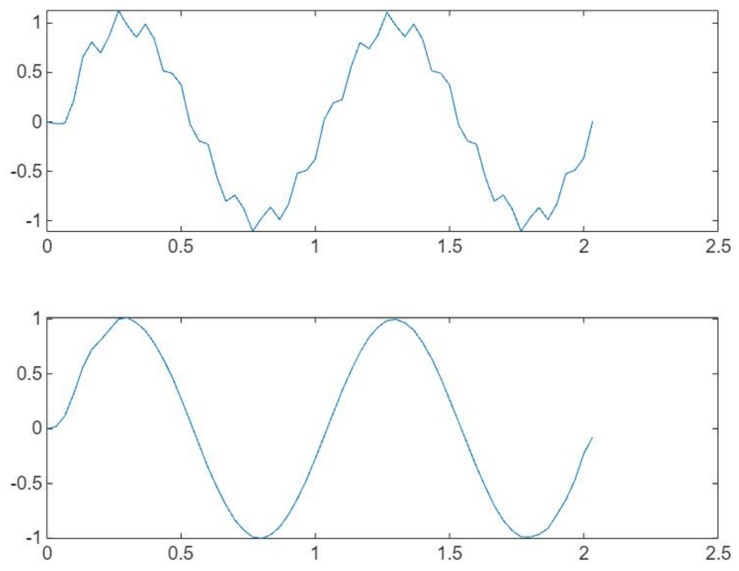
```
subplot(3,1,2); plot(t, ysingle1);
subplot(3,1,3); plot(t, x);
```



```
% Filtering out the noise
wn = 0.5;
[b, a] = butter(4, wn);
vout_double = filter(b, a, ydouble1);
vout_single = filter(b, a, ysingle1);
figure(4)
subplot(2,1,1); plot(t, vout_double);
```



```
subplot(2,1,2); plot(t, vout_single);
```



Task:

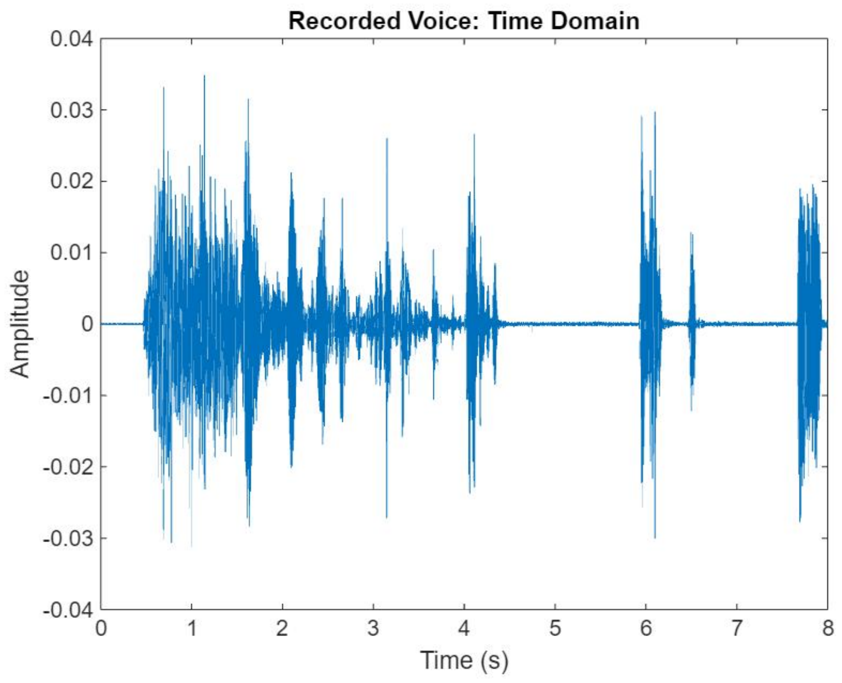
```
Fs = 8000;
bits = 16;
channels = 1;
recTime = 8;
recObj = audiorecorder(Fs, bits, channels);
disp('Start speaking...');
```

```
Start speaking...
```

```
recordblocking(recObj, recTime);
disp('Recording finished.');
```

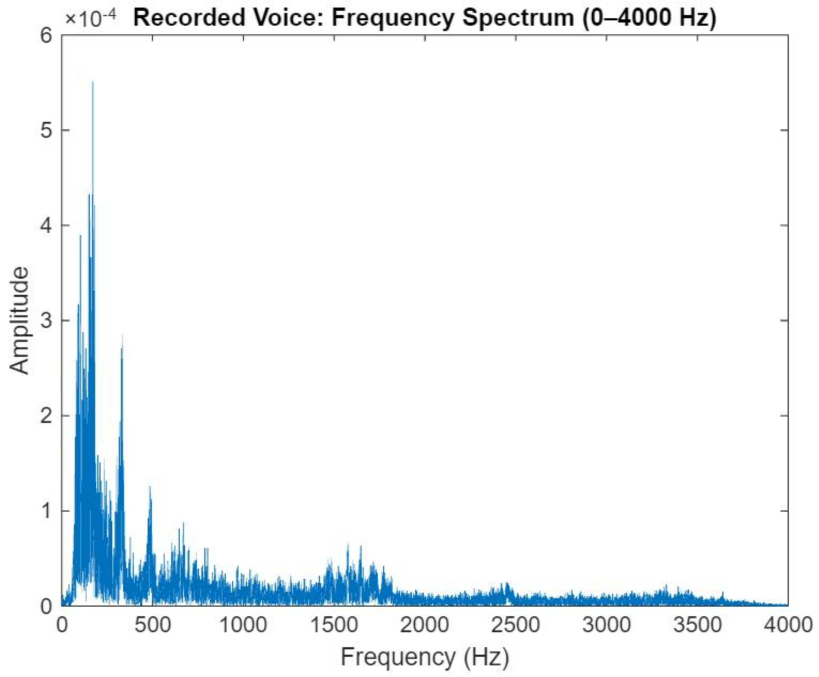
```
Recording finished.
```

```
y = getaudiodata(recObj);
t = (0:length(y)-1) / Fs;
figure;
plot(t, y);
xlabel('Time (s)');
ylabel('Amplitude');
title('Recorded Voice: Time Domain');
```



```
N = length(y);
Y = fft(y);
P2 = abs(Y)/N;
P1 = P2(1:floor(N/2)+1);
```

```
P1(2:end-1) = 2*P1(2:end-1);
f = (0:floor(N/2)) * (Fs/N);
idx = (f <= 4000);
figure;
plot(f(idx), P1(idx));
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Recorded Voice: Frequency Spectrum (0–4000 Hz)');
xlim([0 4000]);
```


Recorded Voice: Frequency Spectrum (0–4000 Hz)
×10^-4
Amplitude
Frequency (Hz)