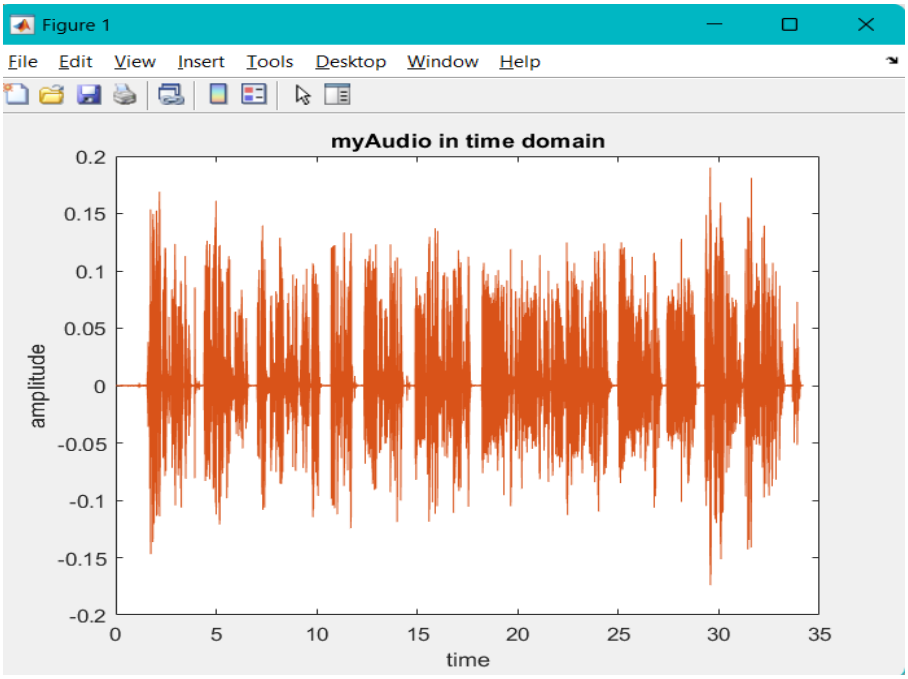
Lab 1 Communication Engineering

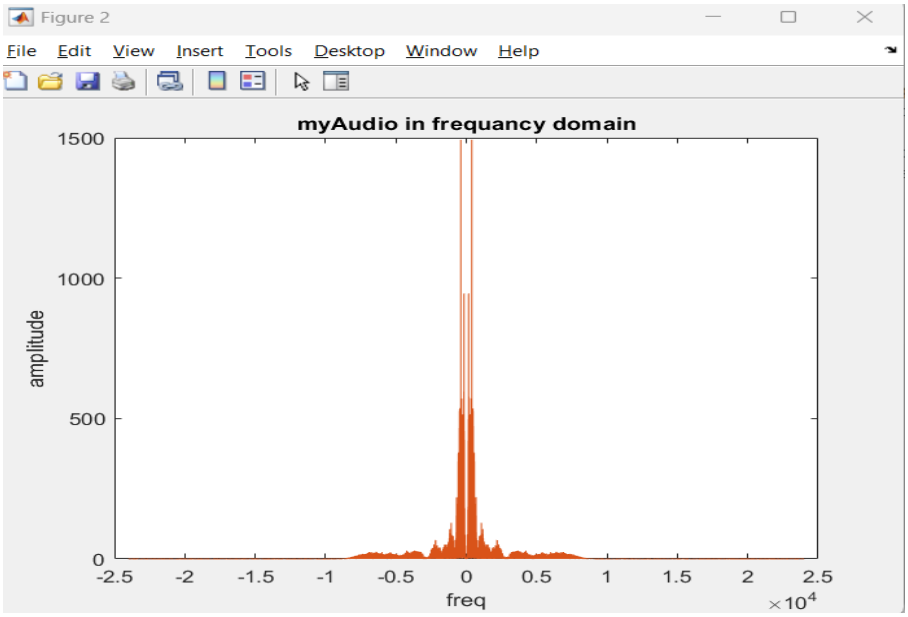
Name: Abdelaziz Salah Mohammed Abdou
BN: 1
Sec: 2

The Original Signal:

- Time Domain:



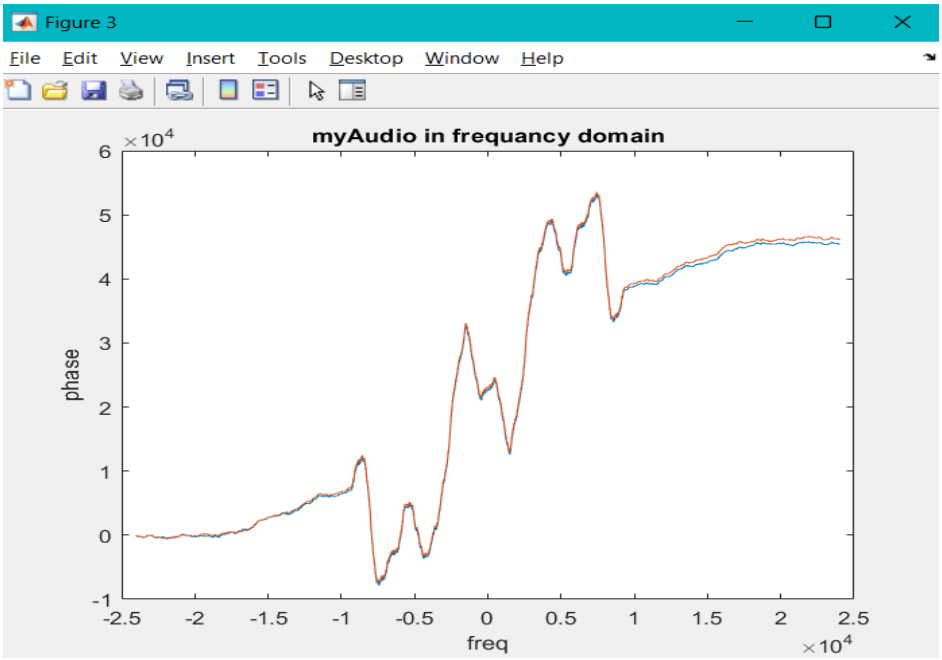
- Amplitude in Frequency Domain:



Lab 1 Communication Engineering

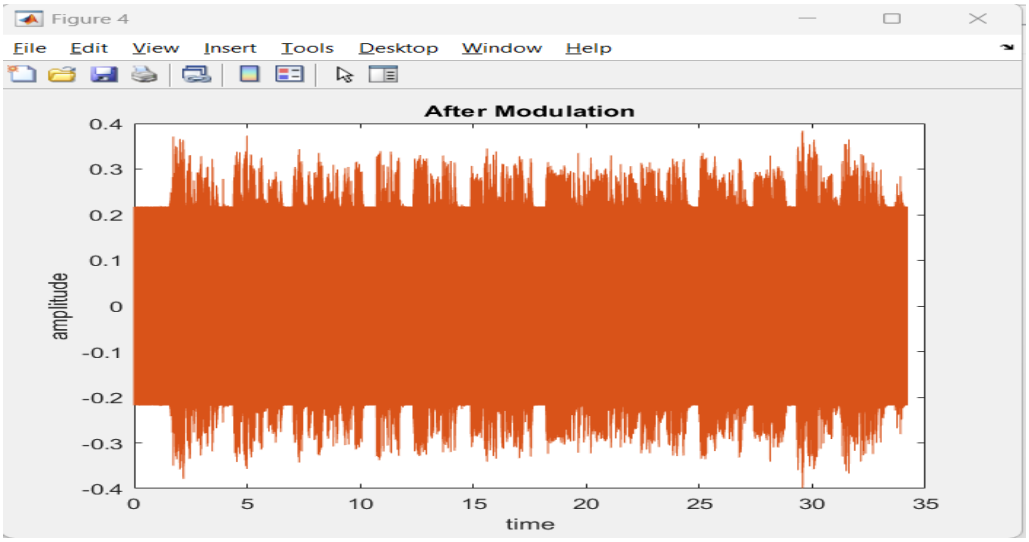
Name: Abdelaziz Salah Mohammed Abdou
BN: 1
Sec: 2

- Phase in the Frequency Domain:



Modulation:

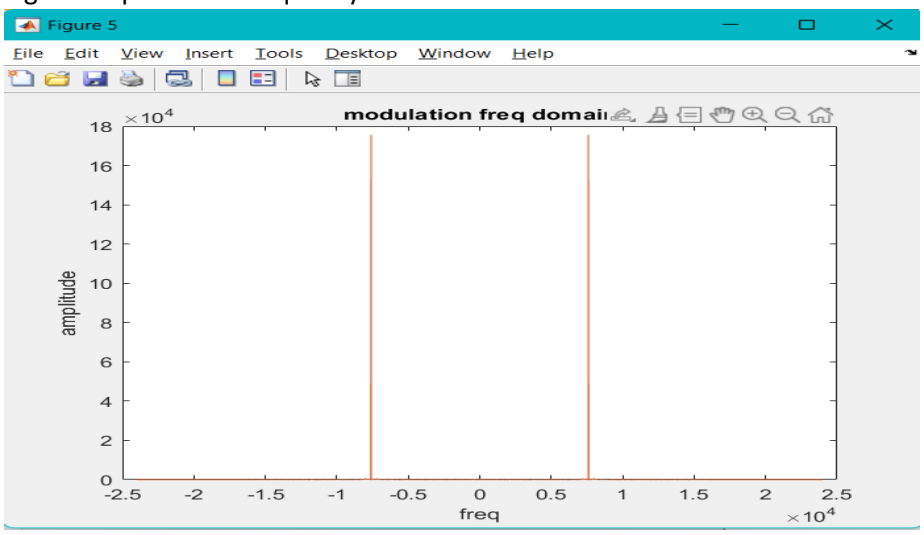
- Signal After Modulation:
  - Signal in Time Domain:



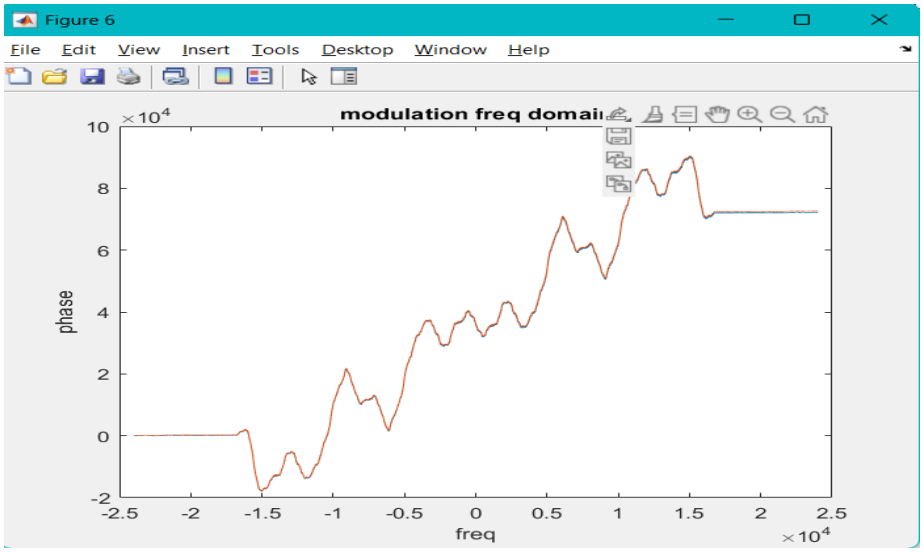
Lab 1 Communication Engineering

Name: Abdelaziz Salah Mohammed Abdou
BN: 1
Sec: 2

- Signal Amplitude in frequency Domain:



- Signal Phase in frequency Domain:



I choose A as the absolute of the minimum value of the signal, that is because we need the modulation index to be in range [0,1] to avoid over modulation.

Overmodulation can distort the signal and lead to information loss, moreover the envelope detector will not be able to detect the original signal again.

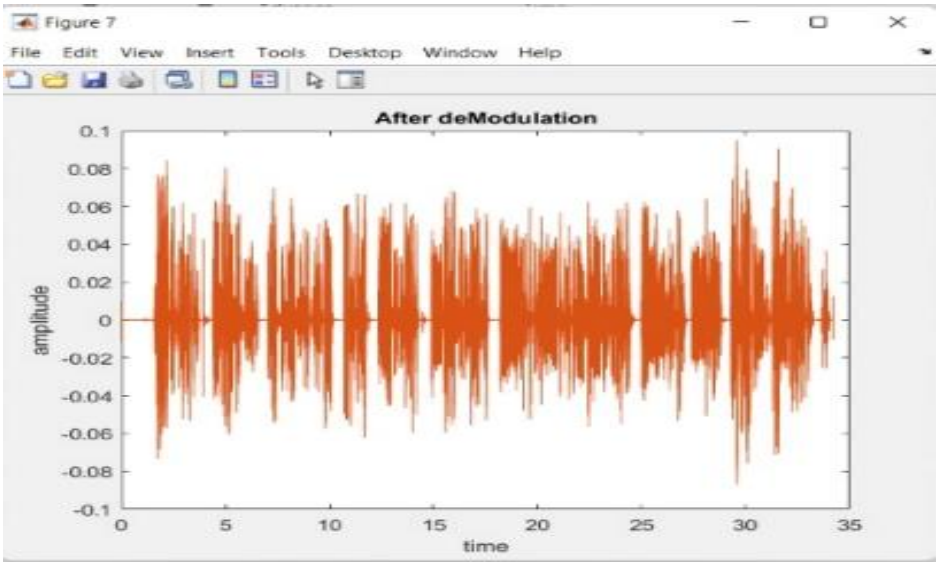
I chose this w to be able to meet Nyquist criteria.

Lab 1 Communication Engineering

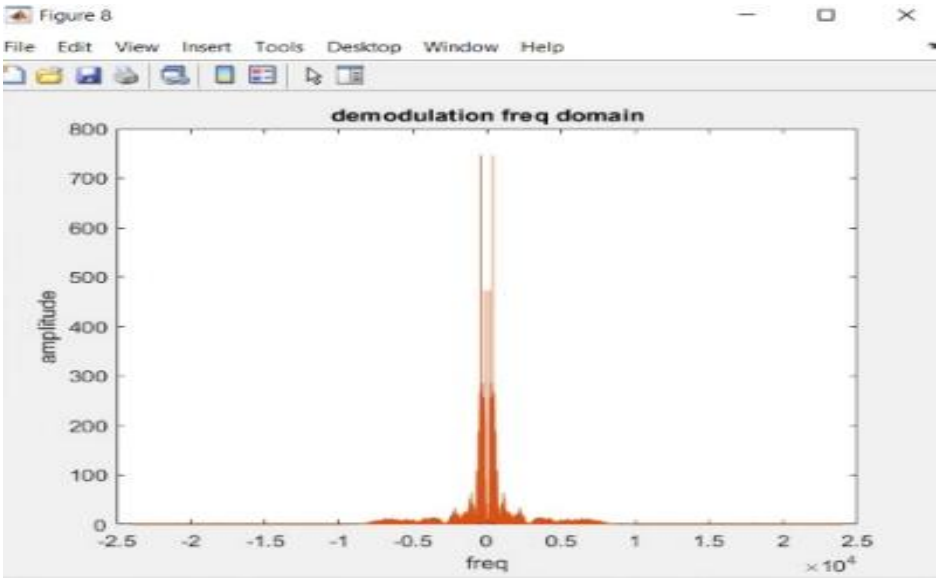
Name: Abdelaziz Salah Mohammed Abdou
BN: 1
Sec: 2

Demodulation:

- Signal after demodulation:
  - In Time Domain:



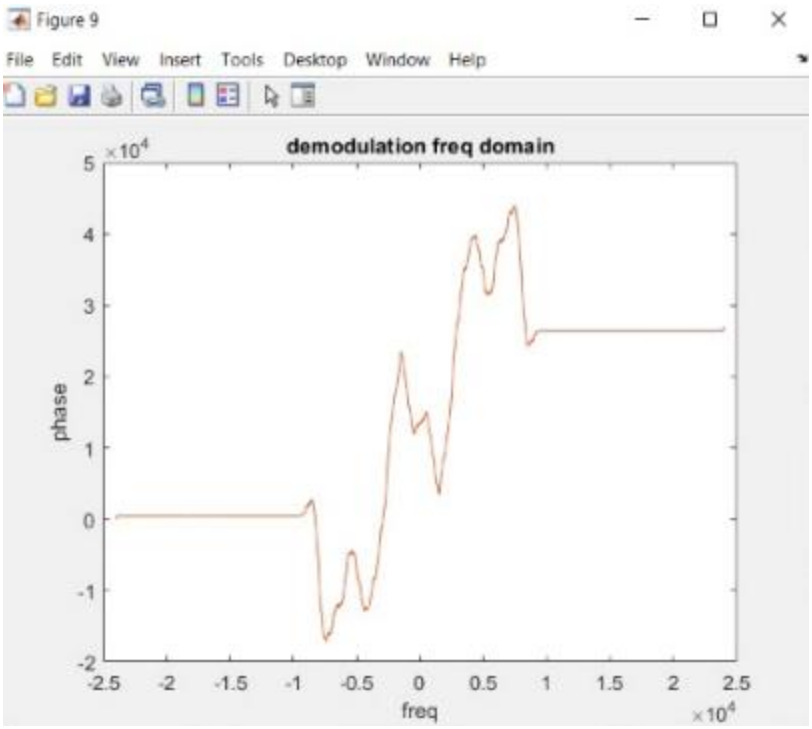
  - Amplitude in Frequency Domain:



Lab 1 Communication Engineering

Name: Abdelaziz Salah Mohammed Abdou
BN: 1
Sec: 2

- Phase in frequency domain:



Notice here that the amplitude of the signal has been halved, this is because of the demodulation process.

This leads to having a little bit lower sound.

Lab 1 Communication Engineering

Name: Abdelaziz Salah Mohammed Abdou
BN: 1
Sec: 2

Code Snippets:

```
% clearing the previous
clc;
close all;
clear all;

% reading the audio
[mySignal, mySignalFreq] = audioread('myRecord.m4a');

%ploting the time domain

% Signal Time step should be with length(mySignal)/mySignalFreq per step
t = linspace(0, length(mySignal)/mySignalFreq , length(mySignal));
f = -mySignalFreq/2 : mySignalFreq/length(t) : mySignalFreq/2 - mySignalFreq/length(t);
figure(1);
plot(t, mySignal);
title ('myAudio in time domain');
xlabel('time');
ylabel('amplitude');

%plotting frequency domain
messageFreq = fftshift(fft(mySignal));
figure(2);
plot(f, abs(messageFreq));
title ('myAudio in frequancy domain');
ylabel('amplitude');
xlabel('freq');
```

```
%plotting frequency domain
messageFreq = fftshift(fft(mySignal));
figure(2);
plot(f, abs(messageFreq));
title ('myAudio in frequancy domain');
ylabel('amplitude');
xlabel('freq');

% ploting the phase
figure(3);
plot(f, unwrap(angle(messageFreq)));
title ('myAudio in frequancy domain');
ylabel('phase');
xlabel('freq');


% Modulation phase
CarrierTime = cos(2*pi*9000*t);
CarrierFreq = abs(fftshift(fft(CarrierTime)))/mySignalFreq;

% Modulation idx.
ModulationIdx = 0.8;
Ac = abs(min(mySignal))/ModulationIdx;
SignalTime = (mySignal + Ac).*CarrierTime';
```

Lab 1 Communication Engineering

Name: Abdelaziz Salah Mohammed Abdou
BN: 1
Sec: 2

```
% plotting the signal after the modualtion in time
figure(4);
plot(t,SignalTime);
title ('After Modulation');
xlabel('time');
ylabel('amplitude');


% plotting the signal after the modualtion in frequancy
SignalFreq = fftshift(fft(SignalTime));
figure(5);
plot(f, abs(SignalFreq));
title ('modulation freq domain');
xlabel('freq');
ylabel('amplitude');

% plotting the signal after the modualtion in frequancy in phase
figure(6);
plot(f, unwrap(angle(SignalFreq)));
title ('modulation freq domain');
xlabel('freq');
ylabel('phase');
```

```
% Demodulation phase.
RecivedSignal = SignalTime.*CarrierTime';
RecivedSignal_after_lowPassFilter = lowpass(RecivedSignal,9000, mySignalFreq, 'Steepness', 0.95);

RecivedSignal_after_lowPassFilter = RecivedSignal_after_lowPassFilter - 0.5*Ac;

% playing the message to ensure that it is the same and working.
sound (RecivedSignal_after_lowPassFilter, mySignalFreq);

%plotting the signal in the time domain
figure(7);
plot(t,RecivedSignal_after_lowPassFilter);
title ('After deModulation');
xlabel('time');
ylabel('amplitude');

%plotting the signal in the frequancy domain in amplitude.
RecivedSignal_after_lowPassFilter_freq = fftshift(fft(RecivedSignal_after_lowPassFilter));
figure(8);
plot(f, abs(RecivedSignal_after_lowPassFilter_freq));
title ('demodulation freq domain');
xlabel('freq');
ylabel('amplitude');
```

```
%plotting the signal in the frequancy domain in phase.
figure(9);
plot(f, unwrap(angle(RecivedSignal_after_lowPassFilter_freq)));
title ('demodulation freq domain');
xlabel('freq');
ylabel('phase');
```