Cairo University- Faculty of Engineering
Computer Engineering Department
Communications Engineering – Fall 2022

# Modulation project

| Names | BN | SEC |
|---|---|---|
| Abdelaziz Salah Mohammed | 1 | 2 |
| Adham Ali Abdelaal | 12 | 1 |

Communication project

Delivered to:

DR/Michael Malek and DR/Magdy Elsoudani

And TA/Alaa Kheirallah

**1/5/2023**

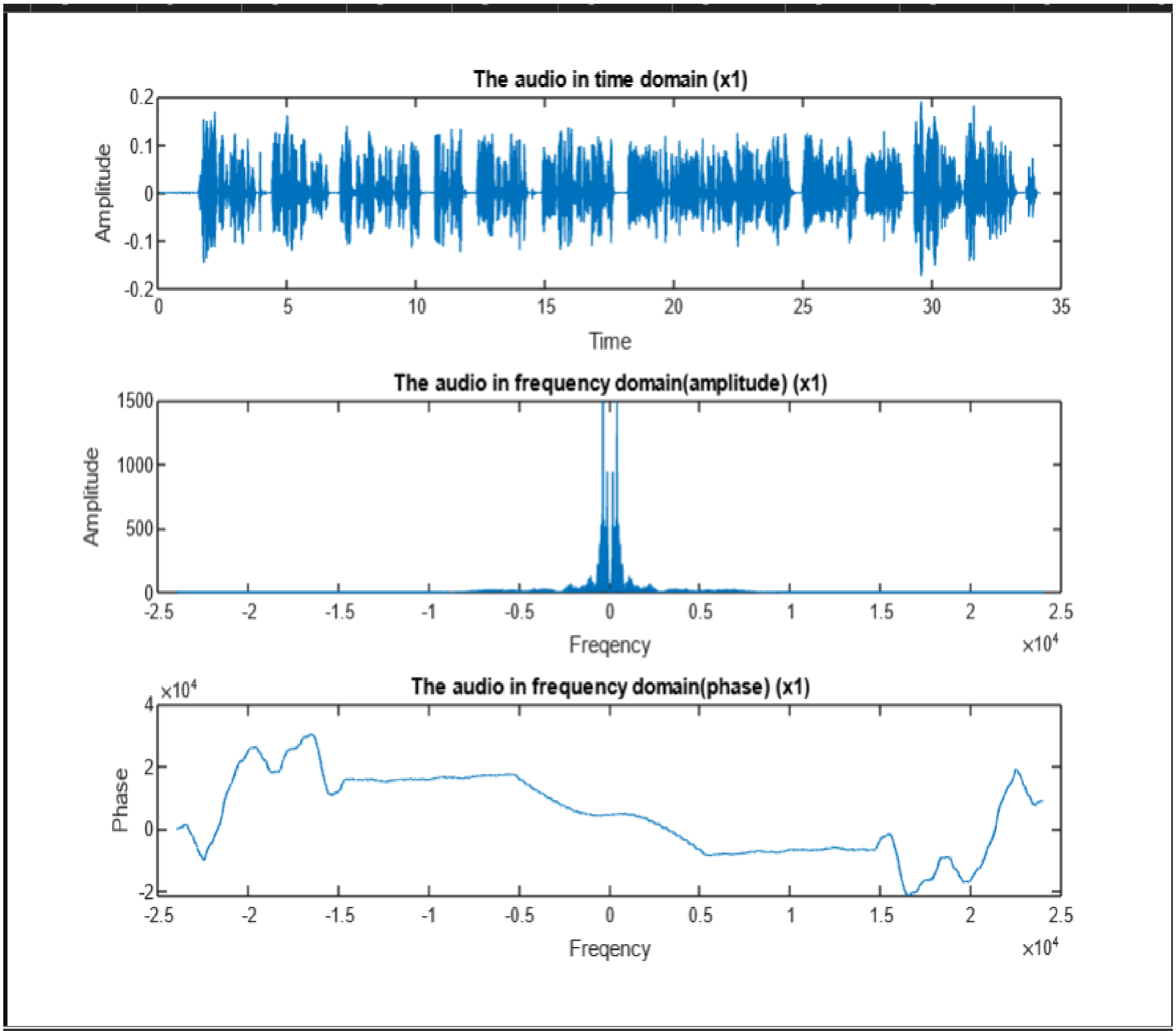
Cairo University- Faculty of Engineering
Computer Engineering Department
Communications Engineering – Fall 2022

# Modulation project

## Q1) Obtain the modulated signal. Plot it in time domain. Plot its magnitude spectrum.

### Signals before modulation

**First Signal:**

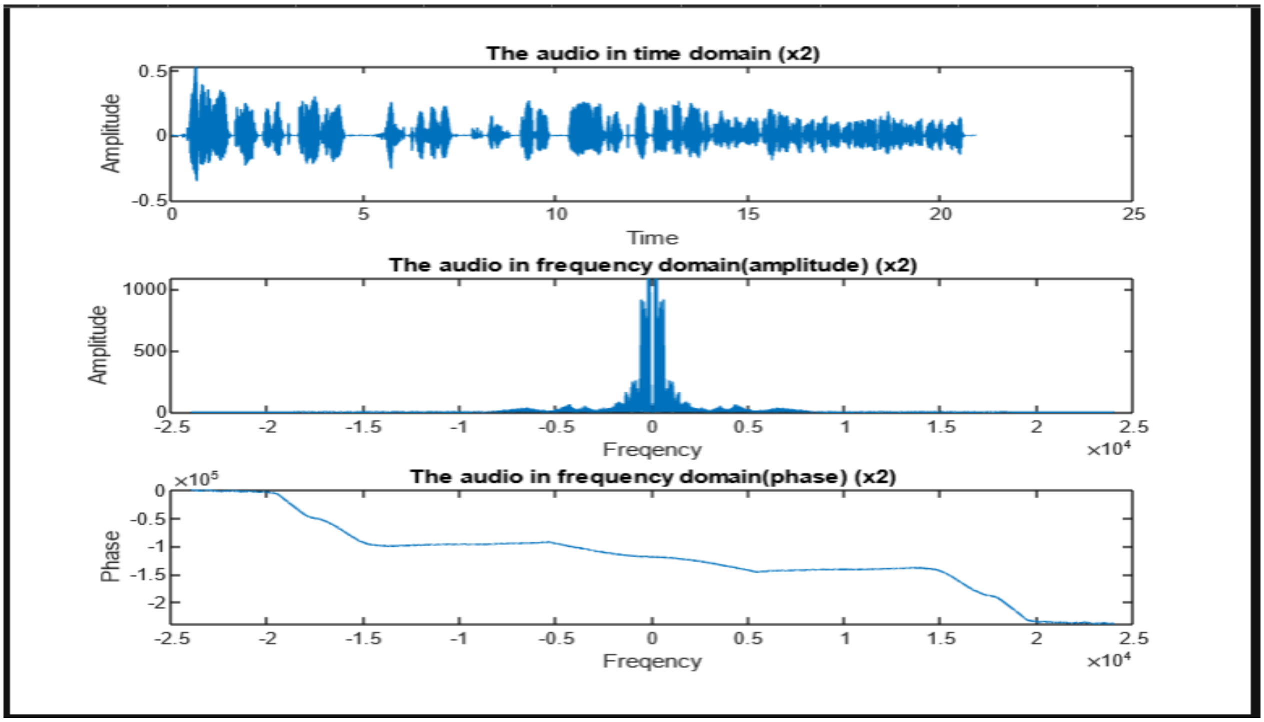



**1/5/2023**

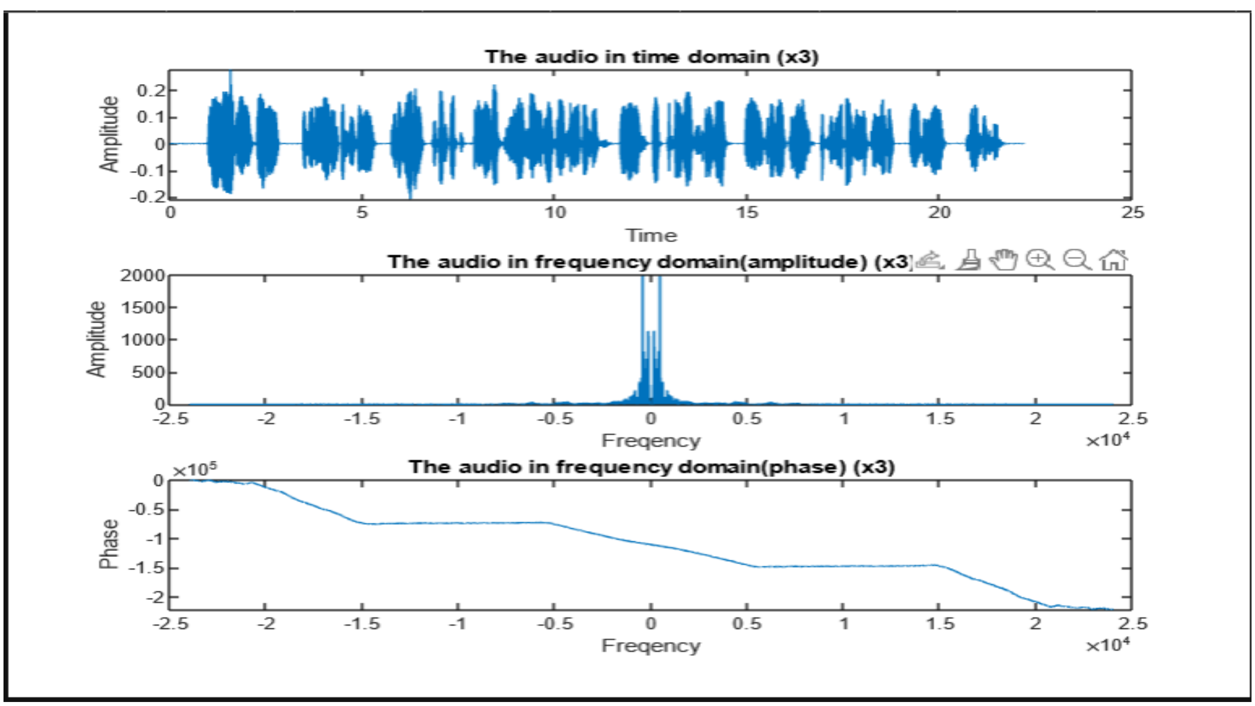
Cairo University- Faculty of Engineering
Computer Engineering Department
Communications Engineering – Fall 2022

# Modulation project

## Second Signal:




## Third signal:

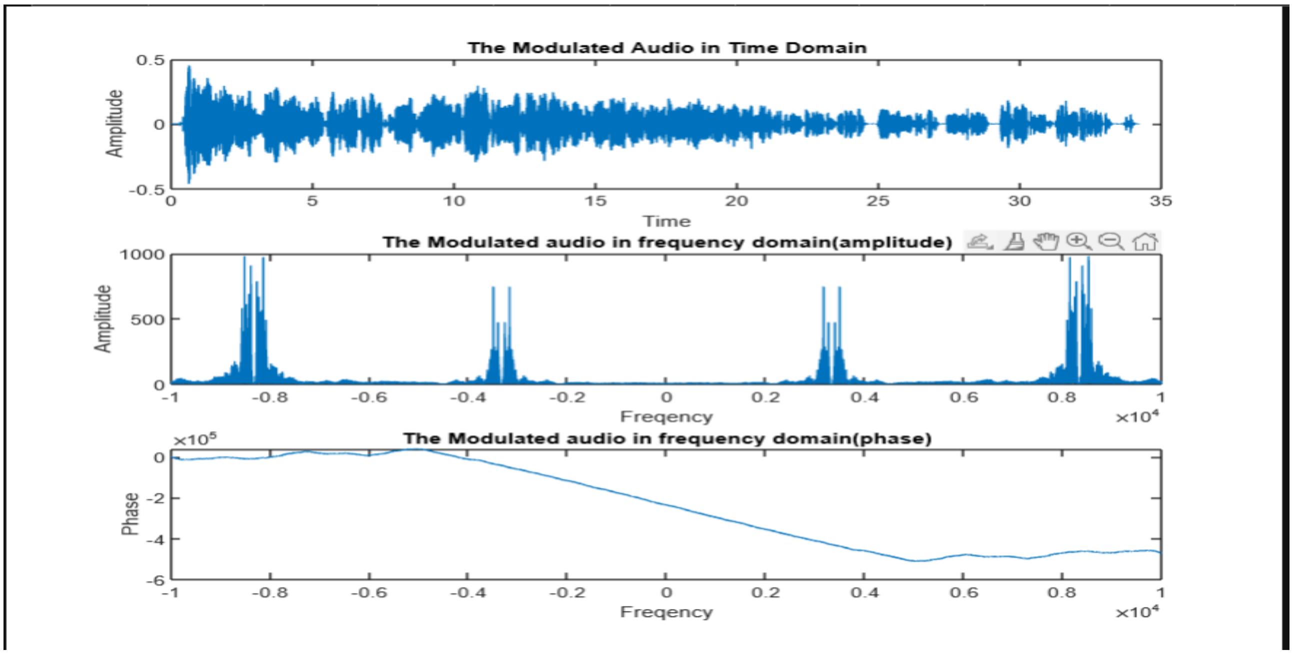



**1/5/2023**

Cairo University- Faculty of Engineering
Computer Engineering Department
Communications Engineering – Fall 2022

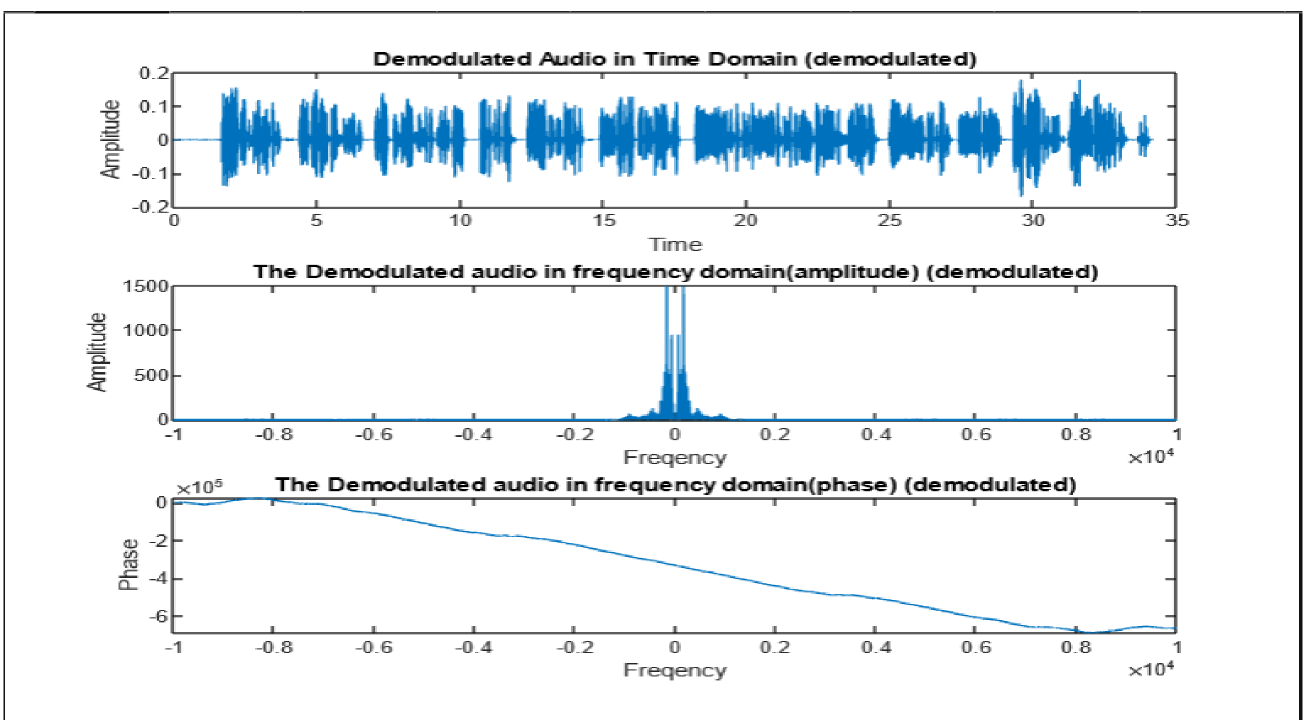
# Modulation project

## Signals after modulation




## Q2) Perform synchronous demodulation to restore the three signals.

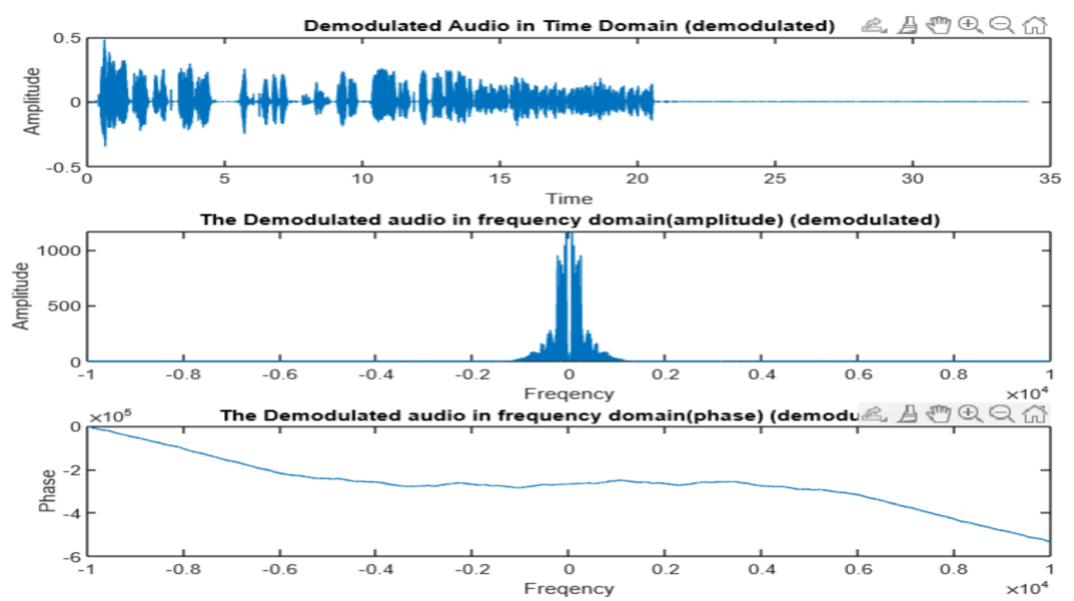
## First Signal:



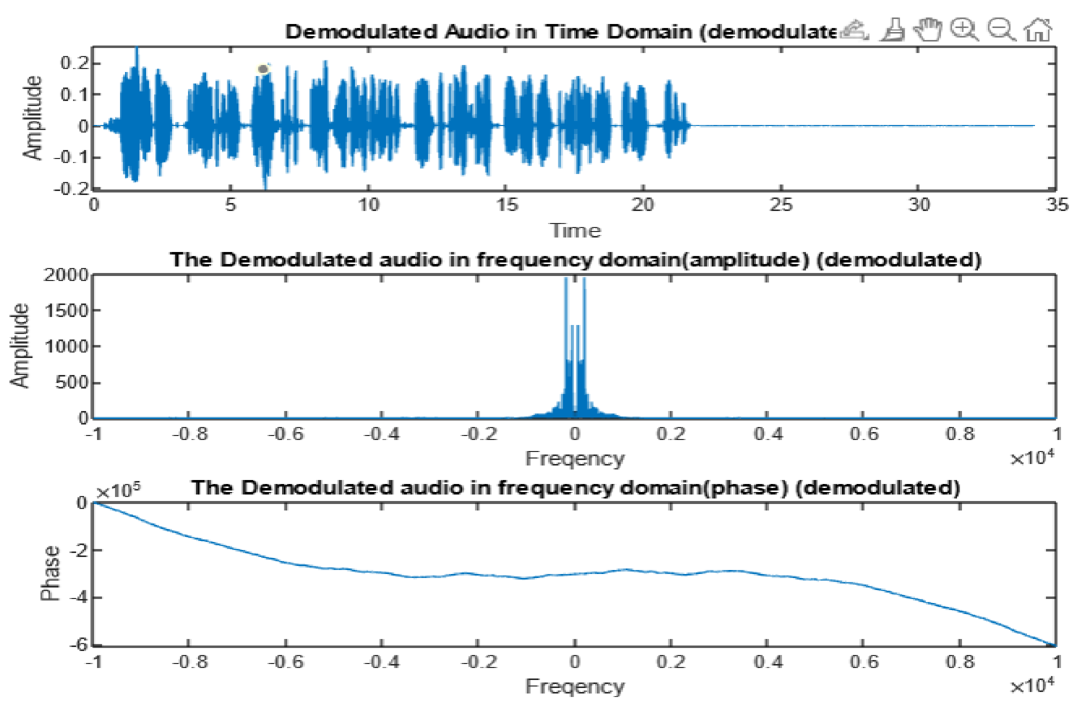
Cairo University- Faculty of Engineering
Computer Engineering Department
Communications Engineering – Fall 2022

# Modulation project

## Second signal:




## Third signal:




**1/5/2023**

# Modulation project

## Justification:

This happened because we have chosen a carrier frequency for the first signal which equals to 8000 Hz and we assigned the second frequency to 2.5 times of the first one, so after applying the synchronized demodulation we multiplied by the locally generated carrier with the same frequency by which we have applied the modulation and after that we applied a low pass filter to discard the high frequencies, which we don't want to take.

But notice that we got the same signals for each on although we have transmitted all signals together on the same spectrum, but that happens because during the demodulation we can select which signal we want to hear by multiplying by a local-generated carrier with the same frequency as that which we have modulated the signal on, so in case of the first signal we multiplied by cos(2*pi*t * carrier Freq) and after applying the LPF (low-pass filter) we can only extract the signal on the frequency only, while for the other signals, notice that we used the QAM idea that we modulated two signals on the same frequency, but we made a trick which is sending them on different 90 phase, so by applying some trigonometric equations we know that this phase shift will cause no interference between the two signals, so by this idea we transmitted two signals, on the same carrier frequency, and by applying the same logic we discussed before we achieved the same signals by the demodulation.

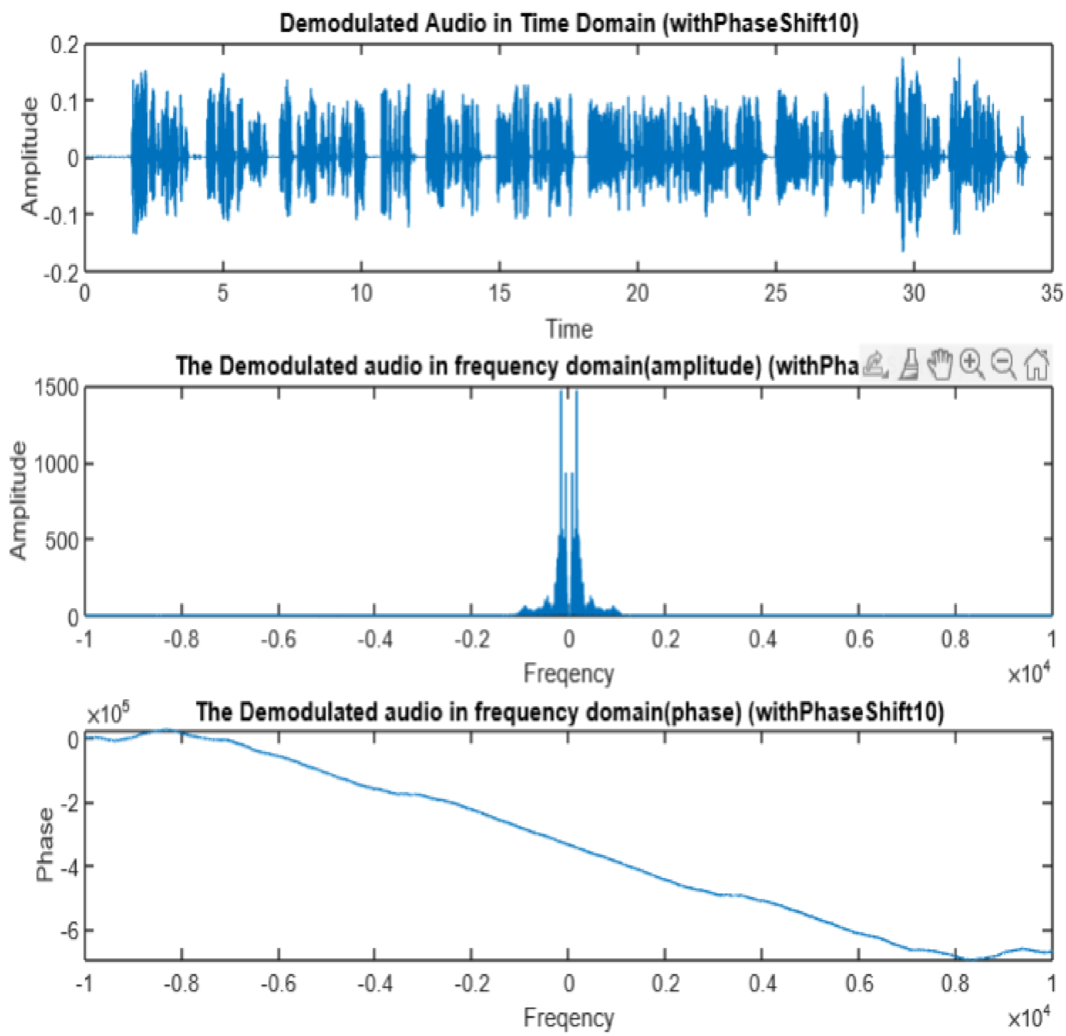
Moreover, during the modulation no interference happened that is because the frequency on which we modulate the signals are greater than the bandwidth of both of them, so none of them has affected the other.

# Modulation project

## Q3) Perform demodulation three times with phase shifts of 10, 30, 90 degrees for both carriers.

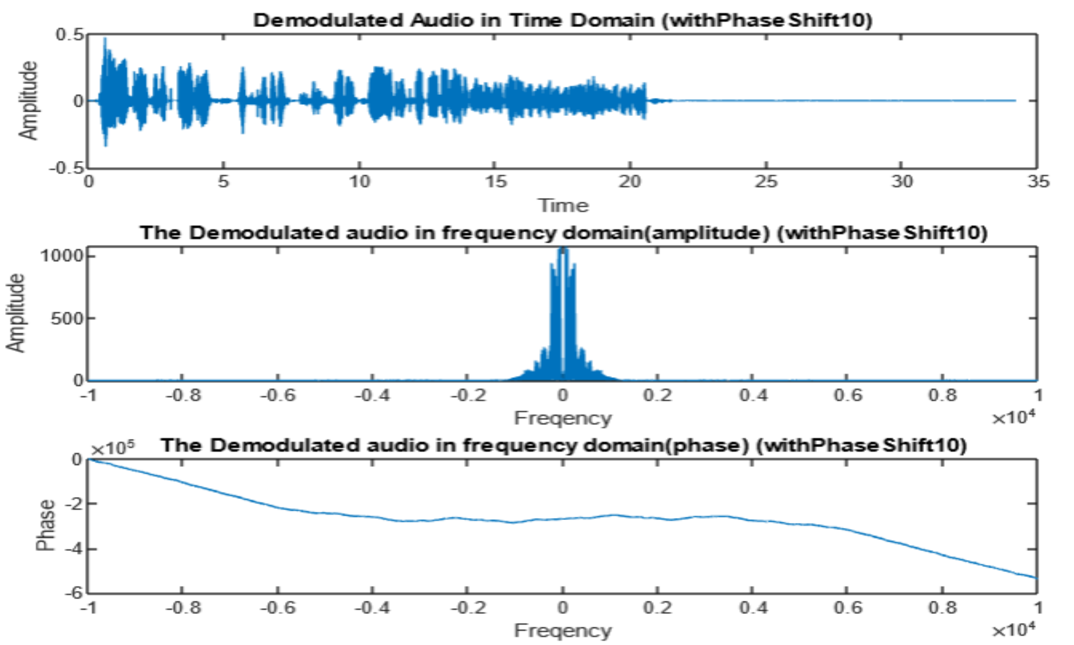
## Q3) a- with phase shift 10:

**First Signal:**

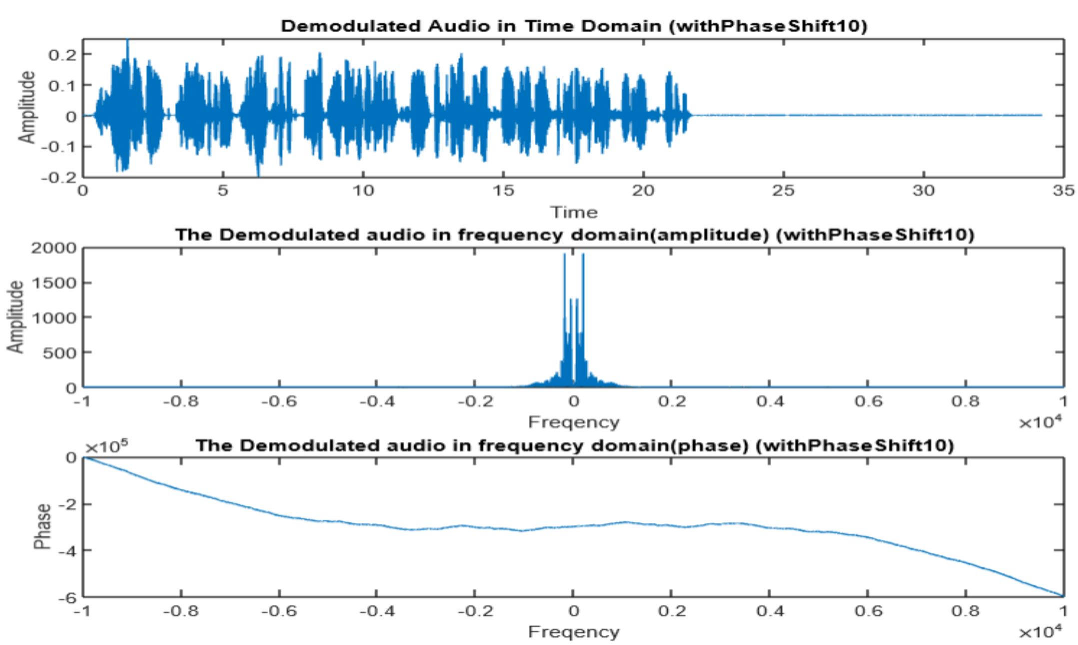


# Modulation project

## Second Signal:



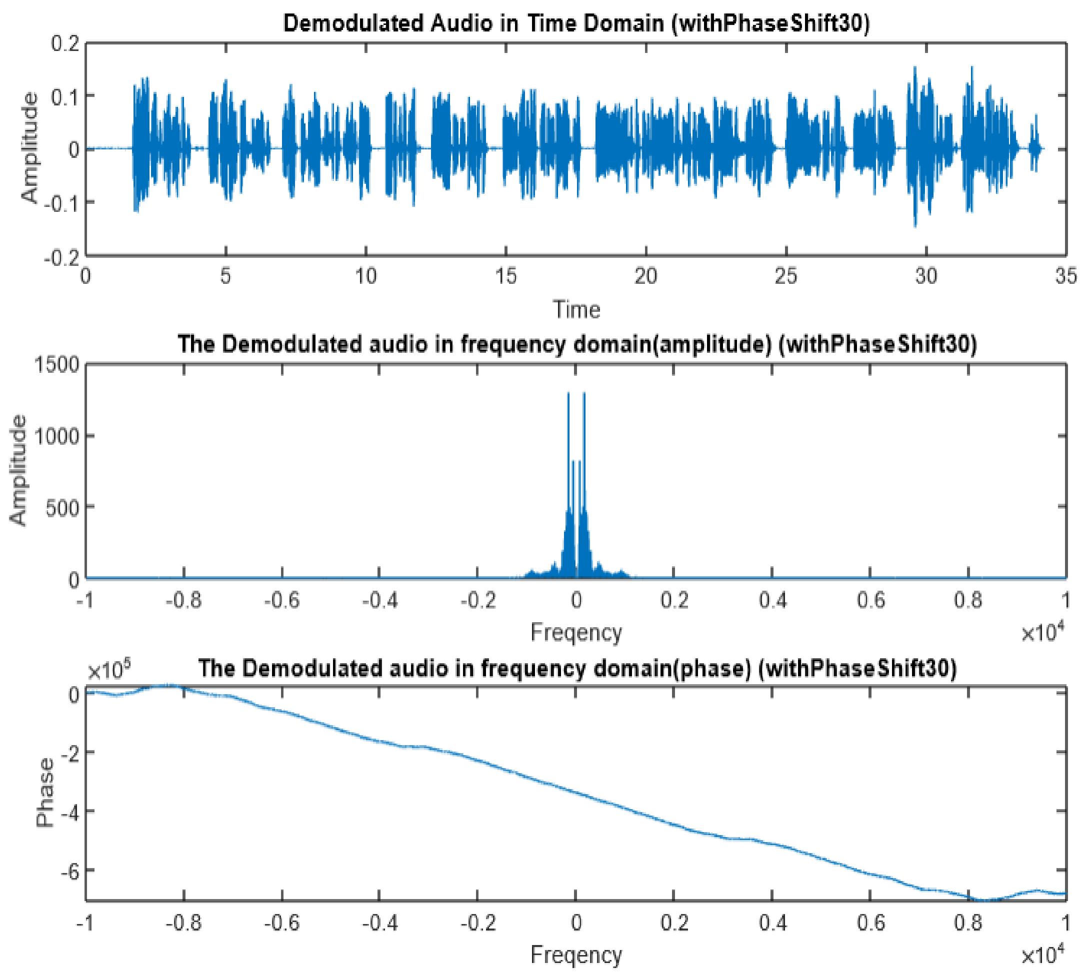

## Third Signal



# Modulation project

## Q3) b- with phase shift 30:

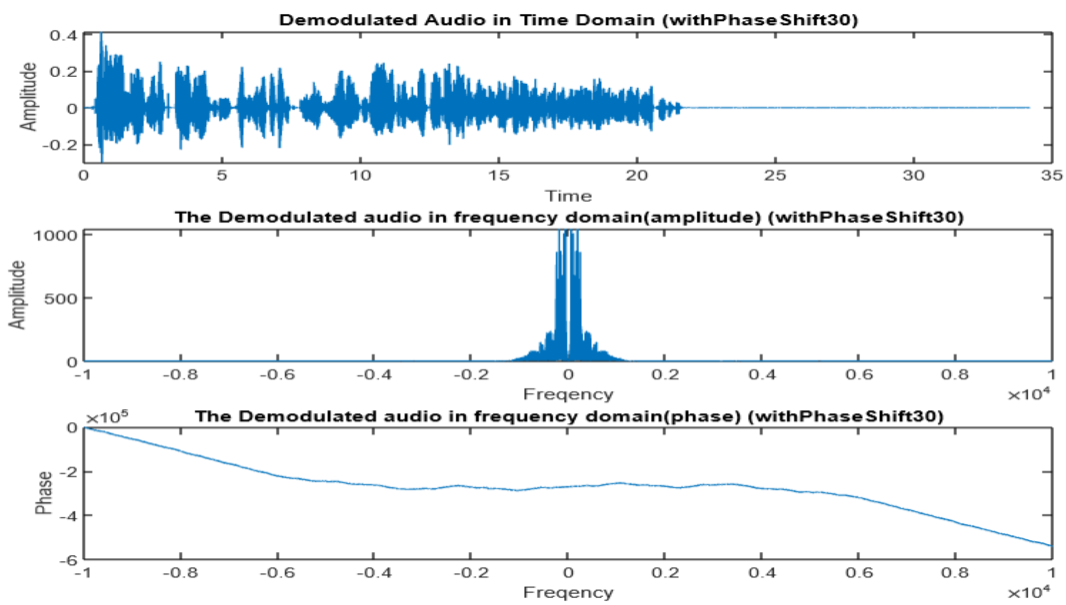
**First Signal:**



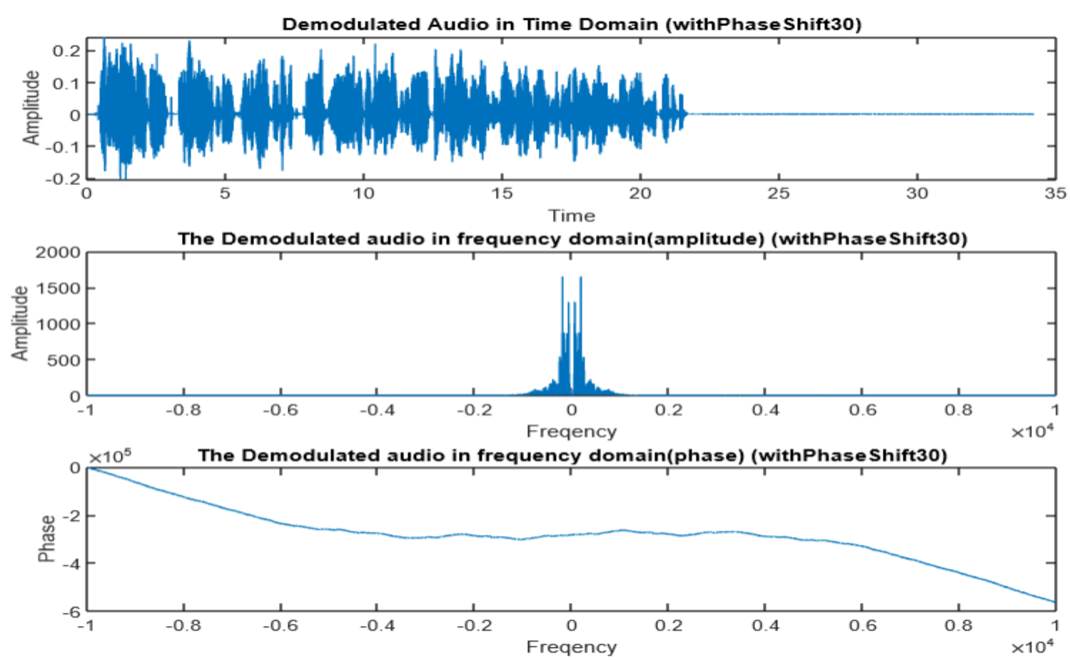
Cairo University- Faculty of Engineering
Computer Engineering Department
Communications Engineering – Fall 2022

# Modulation project

## Second Signal:



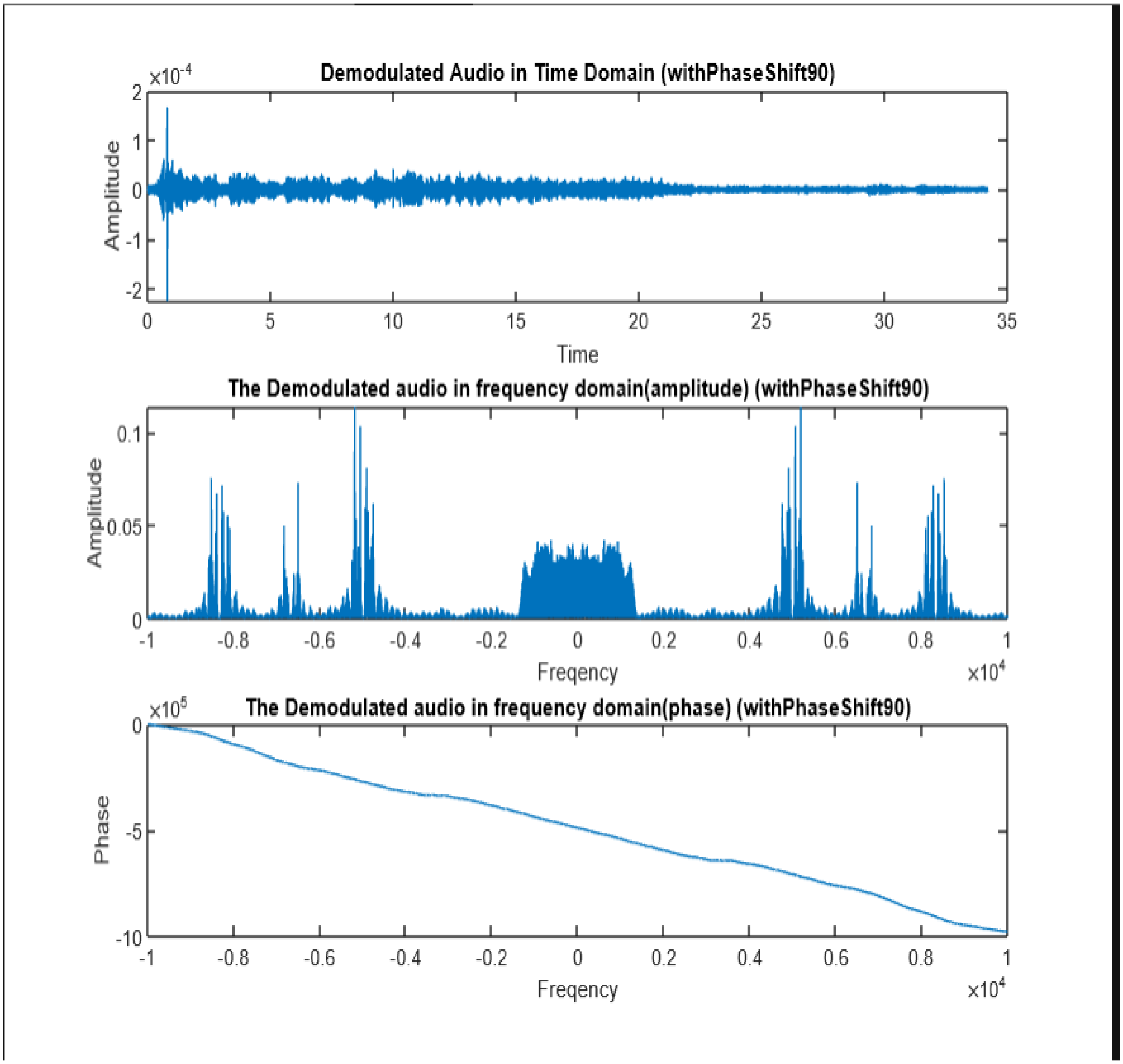

## Third Signal:



# Modulation project

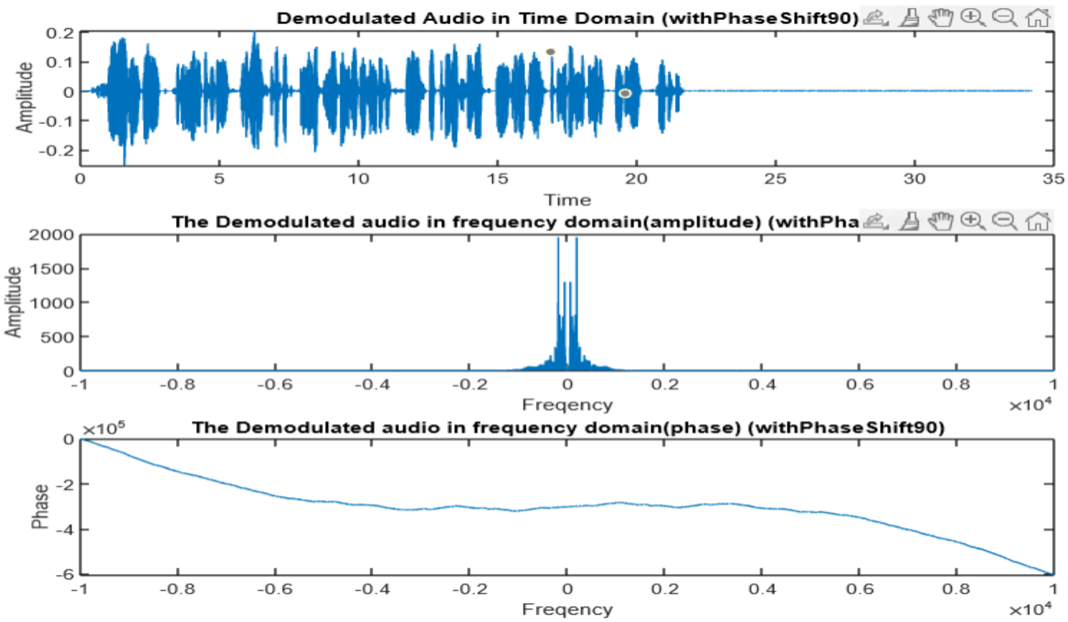
## Q3) c- with phase shift 90:

## First Signal:

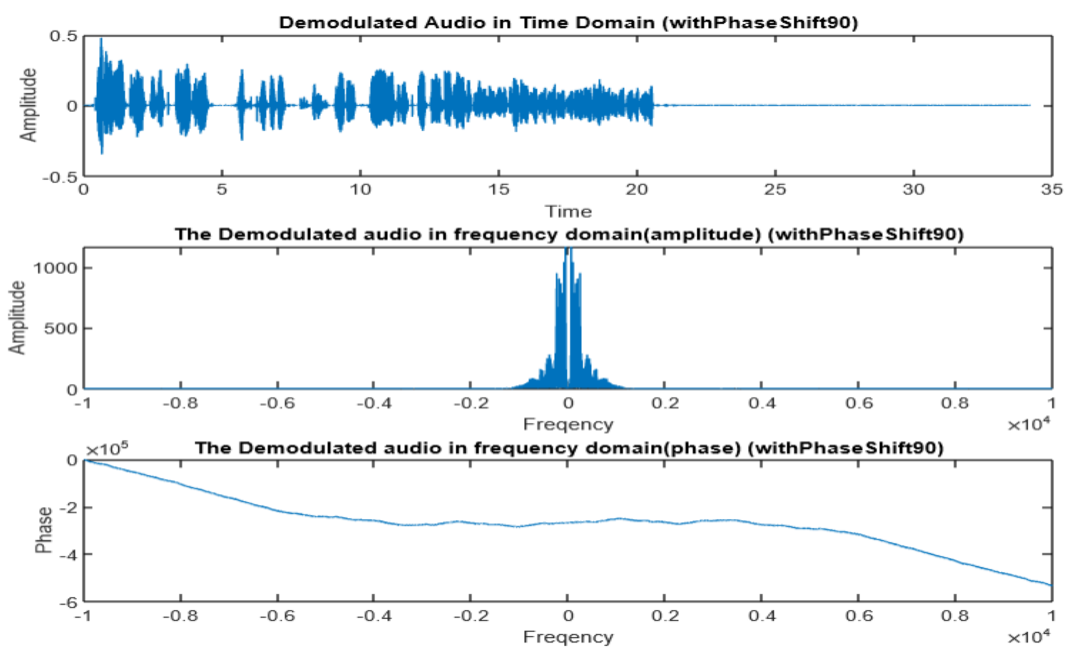


# Modulation project

## Second Signal:




## Third Signal:



# Modulation project

**Justifications:**

Any shift will cause attenuation on the signals, but in case of the first signal It will cause only attenuation, while in the second and third signals, since they are sent on the same bandwidth and we use the idea of QAM, so we depend on the idea that the sin and cos must have phase difference 90 exactly to can avoid any component from any of the two signals on each other, but since we have changed the phase, so we will find some components of each of them affecting the other, so it will cause a distortion because the two signals will affect on each other.

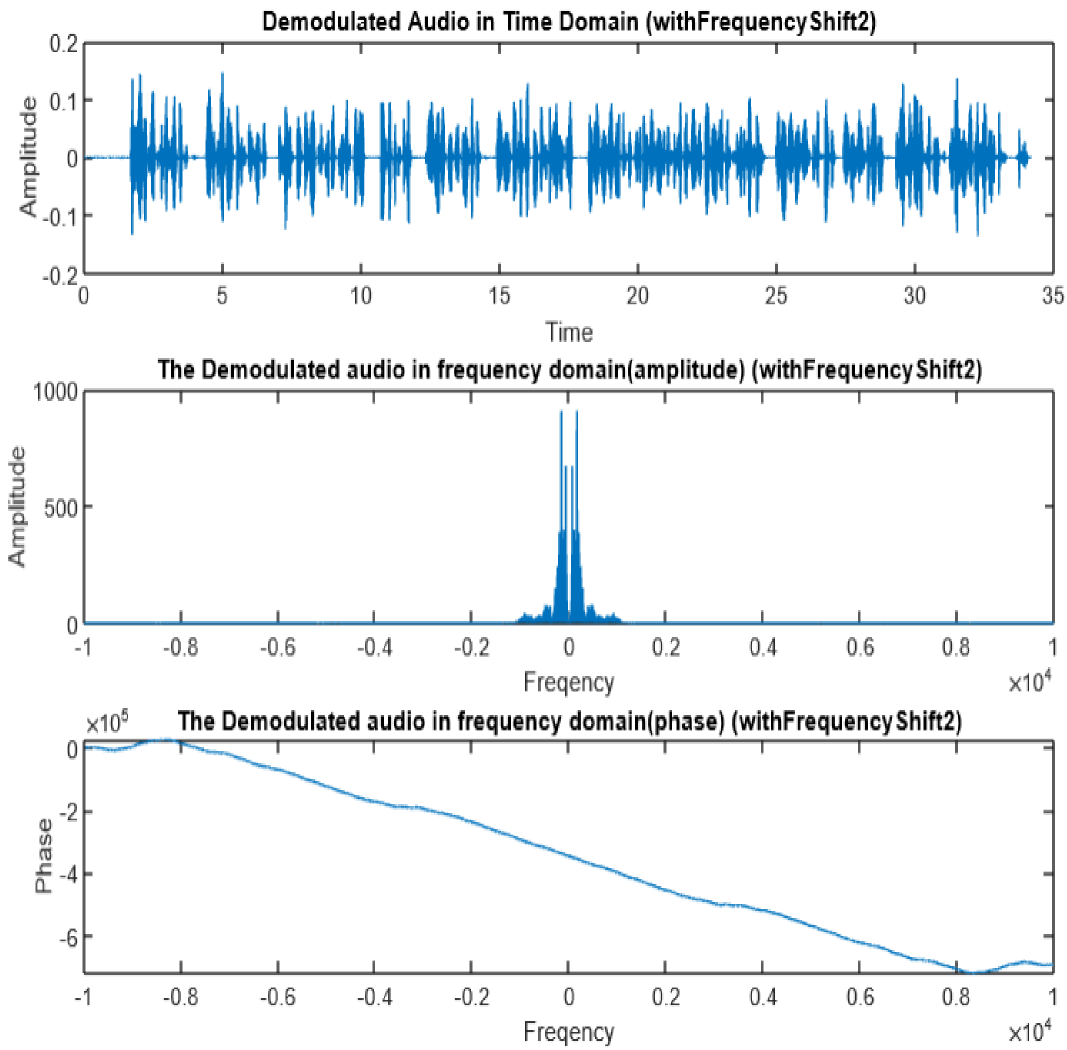
But in case of 90 phase shift no attenuation will take place, that is because we will just replace the signals, as the cos will turn into sin and vice versa so you will hear the second signal on the first port and vice versa.

# Modulation project

## Q4) Perform demodulation two times with a local carrier frequency that is different by 2 Hz and 10 Hz from its carrier frequency.

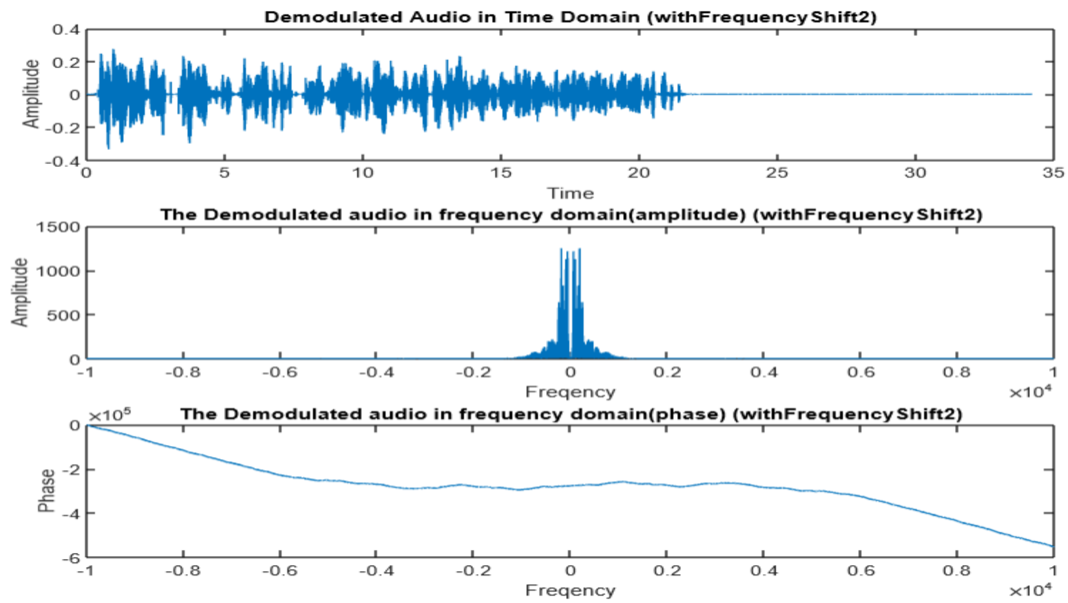
### Q4) a- by 2 Hz

**First Signal:**

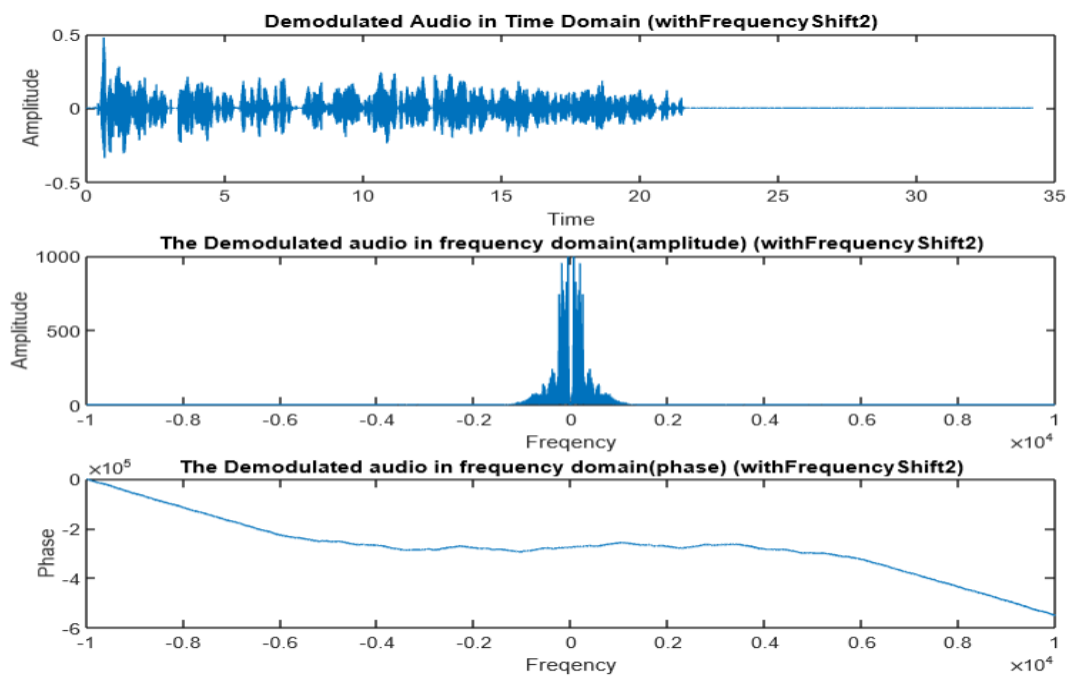


# Modulation project

## Second Signal:



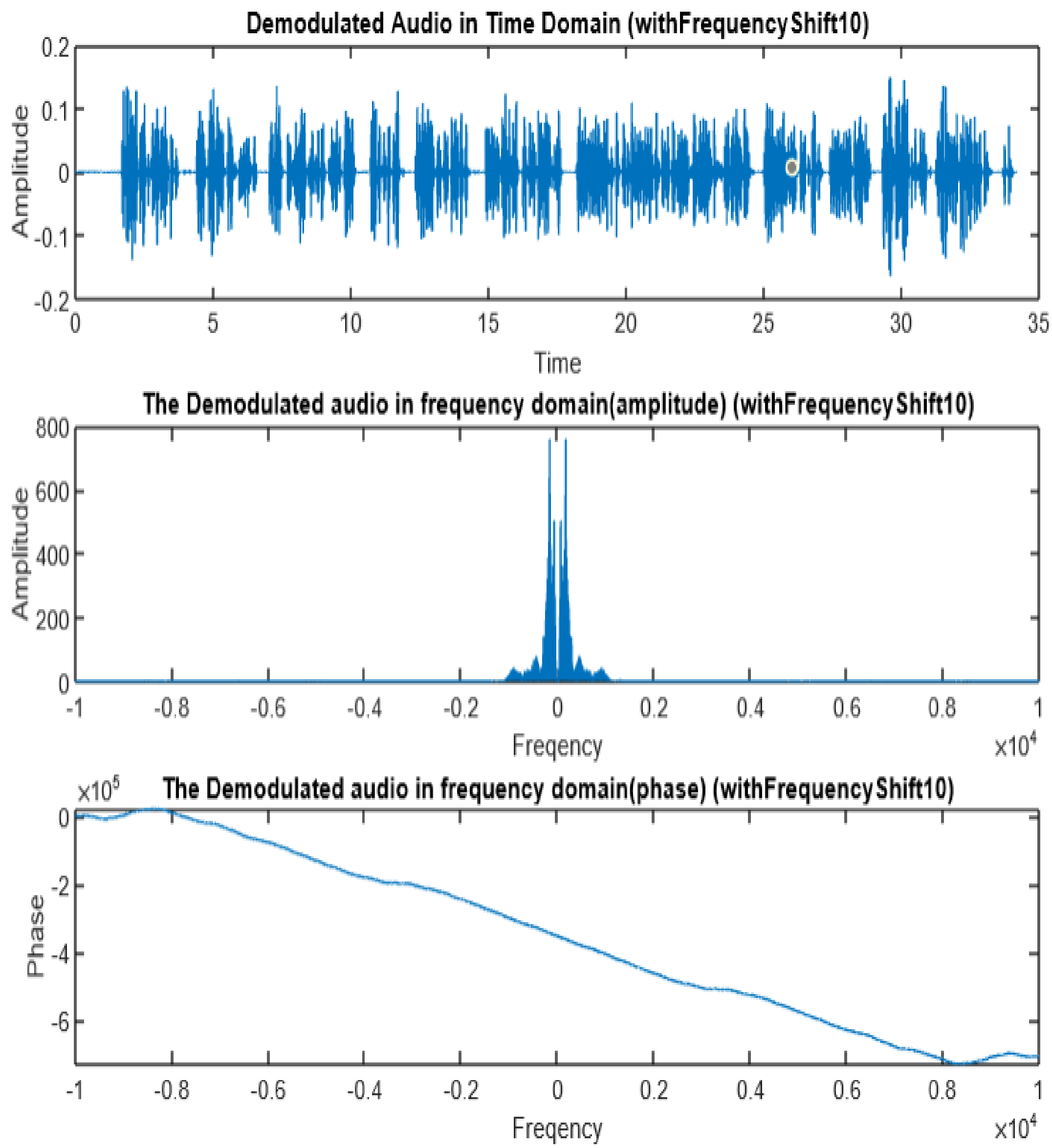

## Third Signal:



# Modulation project

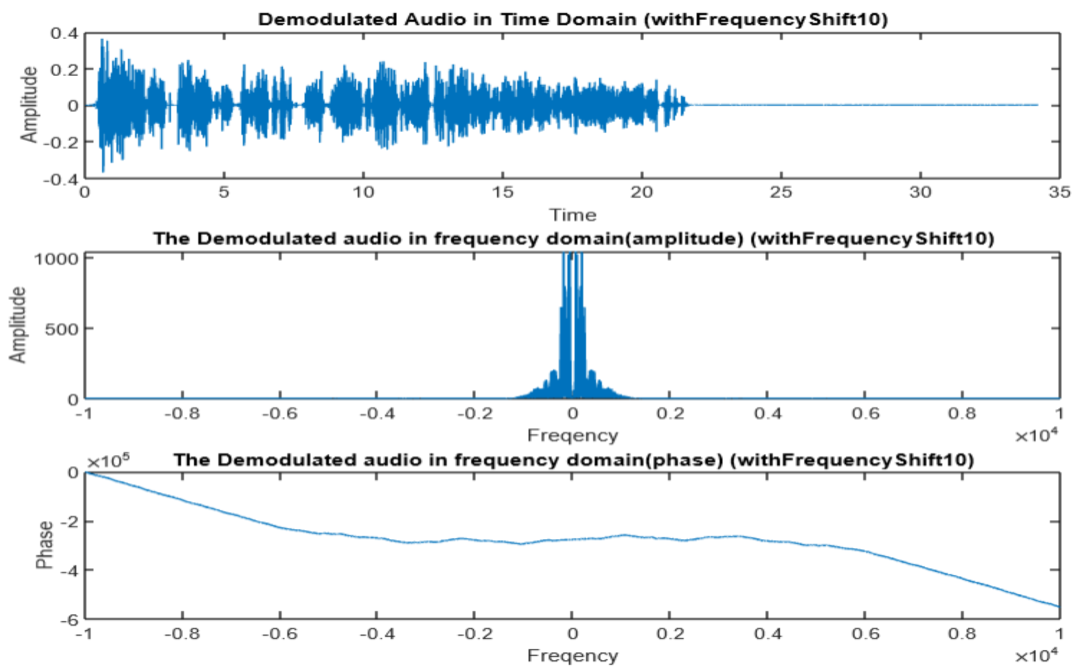
## Q4) b- by 10 Hz

**First Signal:**

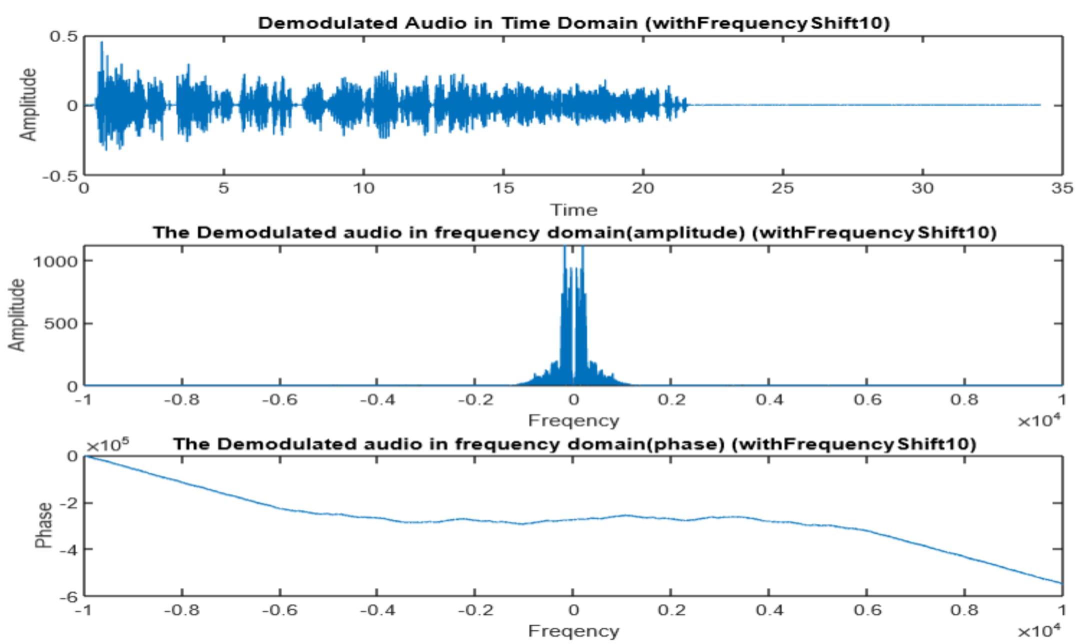


# Modulation project

## Second Signal:




## Third Signal:



**Modulation project**

For frequency shift, it will not cause an interference between signals, so it will cause distortion on signals, so for 2Hz shift it will make the sound a little bit noisy and low, while for 10Hz shift it will distort the signal completely and you won't be able to listen well.

# Modulation project

## Code Snippets:

```
[x1,fs1] = audioread('myRecord.m4a');
[x2,fs2] = audioread('secondRecording.m4a');
[x3,fs3] = audioread('thirdRecording.m4a');
x1=x1(:,1);
x2=x2(:,1);
x3=x3(:,1);

%==============================Plot The Three
Signals======================================

plot_all(x1,fs1,'x1');
plot_all(x2,fs2,'x2');
plot_all(x3,fs3,'x3');
fs=min([fs1,fs2,fs3]);
max_len=max([length(x1),length(x2),length(x3)]);
t = linspace(0,max_len/fs,max_len);
x1 = [x1; transpose(zeros(1, max_len - length(x1)))];
x2 = [x2; transpose(zeros(1, max_len - length(x2)))];
x3 = [x3; transpose(zeros(1, max_len - length(x3)))];

%==============================Modulation=====================================
carrier_one_freq=8000;
carrier_two_freq=2.5*carrier_one_freq;

carrier1=cos(2*pi*carrier_one_freq*t);
carrier2=cos(2*pi*carrier_two_freq*t);
carrier3=sin(2*pi*carrier_two_freq*t);

s=x1.*carrier1.'+x2.*carrier2.'+x3.*carrier3.';

%Draw The Modulated Audio
figure;
subplot(3,1,1);
plotting(t,s,'Time','Amplitude','The Modulated Audio in Time Domain');

%Calculate Fourier Transform
[m,phase,f]=calc_fft(s,carrier_two_freq);

%Draw The Modulated Audio Amplitude in Freqency Domain
subplot(3,1,2);
plotting(f,m,'Freqency','Amplitude','The Modulated audio in frequency
domain(amplitude)');

%Draw The Modulated Audio Phase in Freqency Domain
subplot(3,1,3);
plotting(f,phase,'Freqency','Phase','The Modulated audio in frequency
domain(phase)');
```

# Modulation project

```
%==============================Demodulation 1====================================
phase_shift=0;
frequency_shift=0;
demodulation_function(carrier_one_freq,carrier_two_freq,s,fs,t,phase_shift,frequency_
shift,'demodulated');

%==============================Demodulation 2(with phase
shift)=====================================
phase_shift=10;
frequency_shift=0;
title='withPhaseShift10';
demodulation_function(carrier_one_freq,carrier_two_freq,s,fs,t,phase_shift,frequency_
shift,title)

phase_shift=30;
frequency_shift=0;
title='withPhaseShift30';
demodulation_function(carrier_one_freq,carrier_two_freq,s,fs,t,phase_shift,frequency_
shift,title)

phase_shift=90;
frequency_shift=0;
title='withPhaseShift90';
demodulation_function(carrier_one_freq,carrier_two_freq,s,fs,t,phase_shift,frequency_
shift,title)

%==============================Demodulation 3(with frequency
shift)=====================================

phase_shift=0;
frequency_shift=2;
title='withFrequencyShift2';
demodulation_function(carrier_one_freq,carrier_two_freq,s,fs,t,phase_shift,frequency_
shift,title)

phase_shift=0;
frequency_shift=10;
title='withFrequencyShift10';
demodulation_function(carrier_one_freq,carrier_two_freq,s,fs,t,phase_shift,frequency_
shift,title)


function [m,phase,f] = calc_fft(x,fs)
    N=length(x);
    ftx=fft(x);
    m=abs(fftshift(ftx));
    phase=unwrap(angle(ftx));
    f=(0:N-1)*fs/N;
    f=f-fs*(N-1)/(2*N);
end
```

# Modulation project

```
function plotting(x,y,labelx,labely,ptitle)
    plot(x,y);
    title(ptitle);
    ylabel(labely);
    xlabel(labelx);
end

function plot_all(y,fs,title)
% Set The Time Vector
t = linspace(0,length(y)/fs,length(y));

figure();
% Draw The Audio in Time Domain
subplot(3,1,1);
plotting(t,y,'Time','Amplitude',strcat('The audio in time domain (',title,')'));

% Calculate Fourier Transform
[m,phase,f]=calc_fft(y,fs);

% Draw The Audio Amplitude in Freqency Domain
subplot(3,1,2);
plotting(f,m,'Freqency','Amplitude',strcat('The audio in frequency domain(amplitude)
(',title,')'));


% Draw The Audio Phase in Freqency Domain
subplot(3,1,3);
plotting(f,phase,'Freqency','Phase',strcat('The audio in frequency domain(phase)
(',title,')'));
end

function plot_demodulated(demodulated,fc,t,title)
    % Draw The Demodulated Audio
figure;
subplot(3,1,1);
plotting(t,demodulated,'Time','Amplitude',strcat('Demodulated Audio in Time Domain
(',title,')'));

% Calculate Fourier Transform
[m,phase,f]=calc_fft(demodulated,fc);

% Draw The Demodulated Audio Amplitude in Freqency Domain
subplot(3,1,2);
plotting(f,m,'Freqency','Amplitude',strcat('The Demodulated audio in frequency
domain(amplitude) (',title,')'));

% Draw The Demodulated Audio Phase in Freqency Domain
subplot(3,1,3);
plotting(f,phase,'Freqency','Phase',strcat('The Demodulated audio in frequency
domain(phase) (',title,')'));
end
```

# Modulation project

```
function
demodulation_function(carrier_one_freq,carrier_two_freq,s,fs,t,phase_shift,frequency_
shift,title)
d_carrier1=cos(2*pi*(carrier_one_freq+frequency_shift)*t+phase_shift/180*pi);
d_carrier2=cos(2*pi*(carrier_two_freq+frequency_shift)*t+phase_shift/180*pi);
d_carrier3=sin(2*pi*(carrier_two_freq+frequency_shift)*t+phase_shift/180*pi);

demodulated_x1=s.*d_carrier1.';
demodulated_x1=2*demodulated_x1;
demodulated_x1=lowpass(demodulated_x1,2000,fs,'Steepness',0.95);
audiowrite(strcat(title,'x1.wav'),demodulated_x1,fs);

demodulated_x2=s.*d_carrier2.';
demodulated_x2=2*demodulated_x2;
demodulated_x2=lowpass(demodulated_x2,2000,fs,'Steepness',0.95);
audiowrite(strcat(title,'x2.wav'),demodulated_x2,fs);

demodulated_x3=s.*d_carrier3.';
demodulated_x3=2*demodulated_x3;
demodulated_x3=lowpass(demodulated_x3,2000,fs,'Steepness',0.95);
audiowrite(strcat(title,'x3.wav'),demodulated_x3,fs);



plot_demodulated(demodulated_x1,carrier_two_freq,t,title);
plot_demodulated(demodulated_x2,carrier_two_freq,t,title);
plot_demodulated(demodulated_x3,carrier_two_freq,t,title);
end
```