3rd Year Computer Engineering

Communication Engineering

Project

Name	Sec	B.N
Asmaa Adel Abdelhamed Kawashty	1	14
Zeinab Moawad Fayez Hassan	1	29

A. Explanation of your work.

1- Modulating the input signals on different carriers by multiplying :

- **The first signal by cos(ω1 * t) where ω1 is an angular frequency in rad/sec**
- **The second signal by cos(ω2 * t) where ω2 is an angular frequency in rad/sec**
- **The third signal by sin(ω2 * t) where ω2 is an angular frequency in rad/sec**

ω1 = 2 * M_PI * f1 where f1 = 100,000 Hz

ω2 = 2 * M_PI * f2 where f2 = 150,000 Hz

2- Inorder to demodulate the received signal at the receiver we use a synchronous carrier as follows:

- **The Modulated signal is multiplied by cos(ω1 * t) where ω1 is an angular frequency in rad/sec then a low pass filter with freq = f1 is applied to restore signal 1**
- **The Modulated signal is multiplied by cos(ω2 * t) where ω2 is an angular frequency in rad/sec then a low pass filter with freq = f2 is applied to restore the second signal**
- **The Modulated signal is multiplied by sin(ω2 * t) where ω2 is an angular frequency in rad/sec then a low pass filter with freq = f2 is applied to restore the third signal**

ω1 = 2 * M_PI * f1 where f1 = 100,000 Hz

ω2 = 2 * M_PI * f2 where f2 = 150,000 Hz

3-Inorder to demodulate the received signal at the receiver by demodulation three times with phase shifts of 10, 30, 90 degrees for both carriers.

The Modulated signal is multiplied by `cos(2*pi*f1*t + phase_shift_rad)`; where ω1 is an angular frequency in rad/sec then a low pass filter with freq = f1 is applied to restore signal 1

- **The Modulated signal is multiplied by `cos(2*pi*f2*t + phase_shift_rad)`; where ω2 is an angular frequency in rad/sec then a low pass filter with freq = f2 is applied to restore the second signal**

- **The Modulated signal is multiplied by `sin(2*pi*f2*t + phase_shift_rad)` where ω2 is an angular frequency in rad/sec then a low pass filter with freq = f2 is applied to restore the third signal**

ω1 = 2 * M_PI * f1 where f1 = 100,000 Hz

ω2 = 2 * M_PI * f2 where f2 = 150,000 Hz

4-Inorder to demodulate the received signal at the receiver by 1) demodulation two times with a local carrier frequency that is different by 2 Hz and 10 Hz from its carrier frequency.

- **The Modulated signal is multiplied by `cos(2*pi*(f1 + 2) * t );` where ω1 is an angular frequency in rad/sec then a low pass filter with freq = f1 is applied to restore signal 1**

ω1 = 2 * M_PI * f1 where f1 = 100,000 HZ

6- creating two function

1- to generate carrier: “`gen_carriers`”

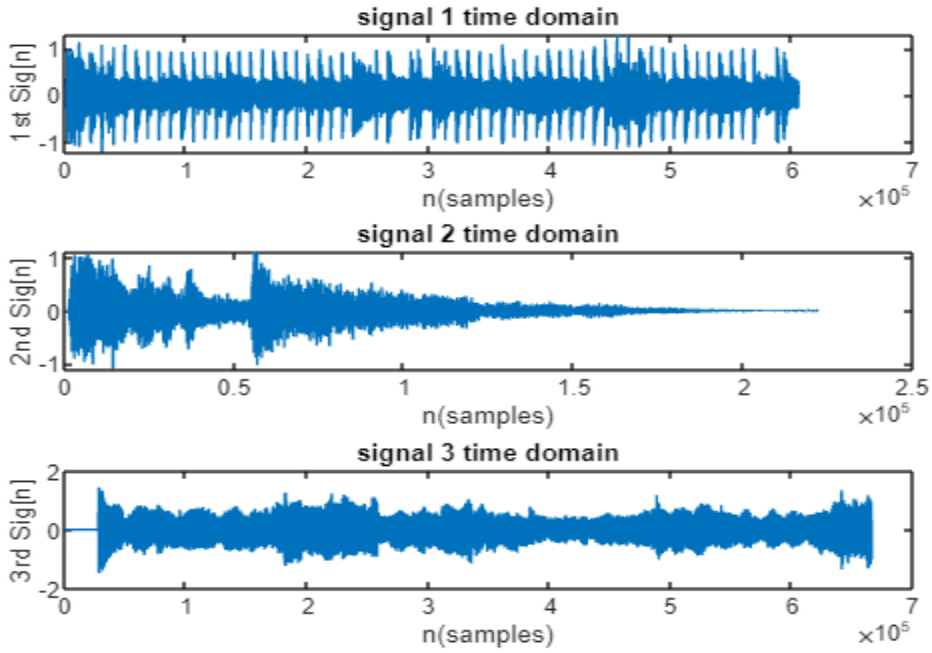
2- to demodulate signal: “ `demodulate_signal`“

B. All the required results and answers to questions.

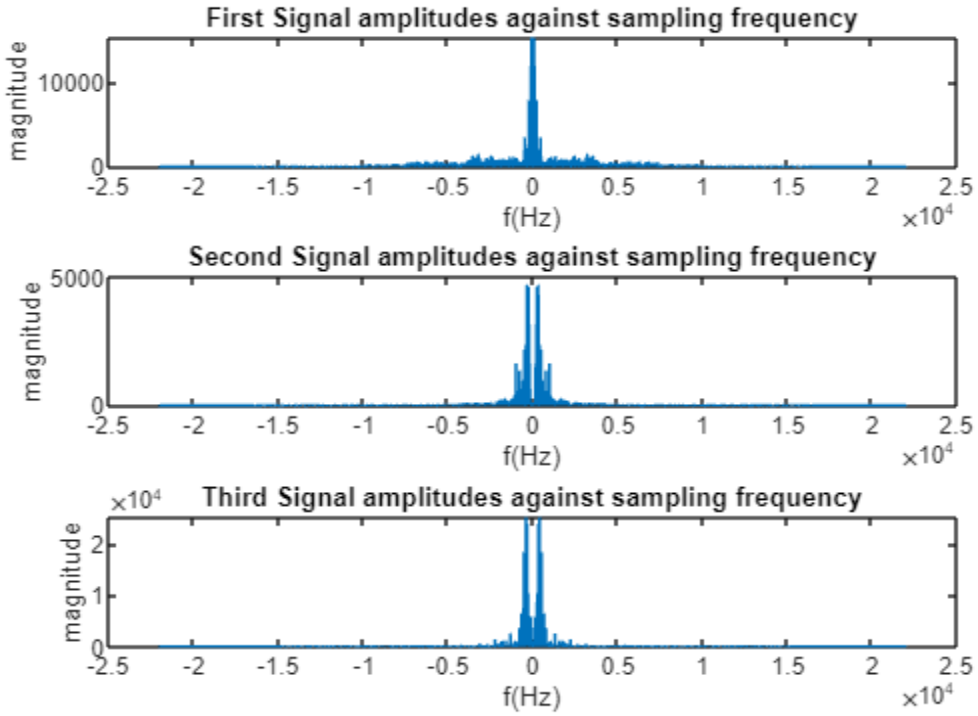
Compare between	synchronous demodulation	demodulation three times with phase	demodulation two times with a local carrier frequency that is different by 2 Hz and 10 Hz from its carrier frequency.
Conclusion	-Signals approximately same as original signal no phase error and frequency	-At 10 degrees: signal partially attenuated -At 30 degrees: signal attenuation increased -At 90 degrees: signal fully attenuated and become no signal exist	-At 2HZ: signal suffer from small distortion and attenuation -At 10 HZ:distortion and attenuation increased

The figures:

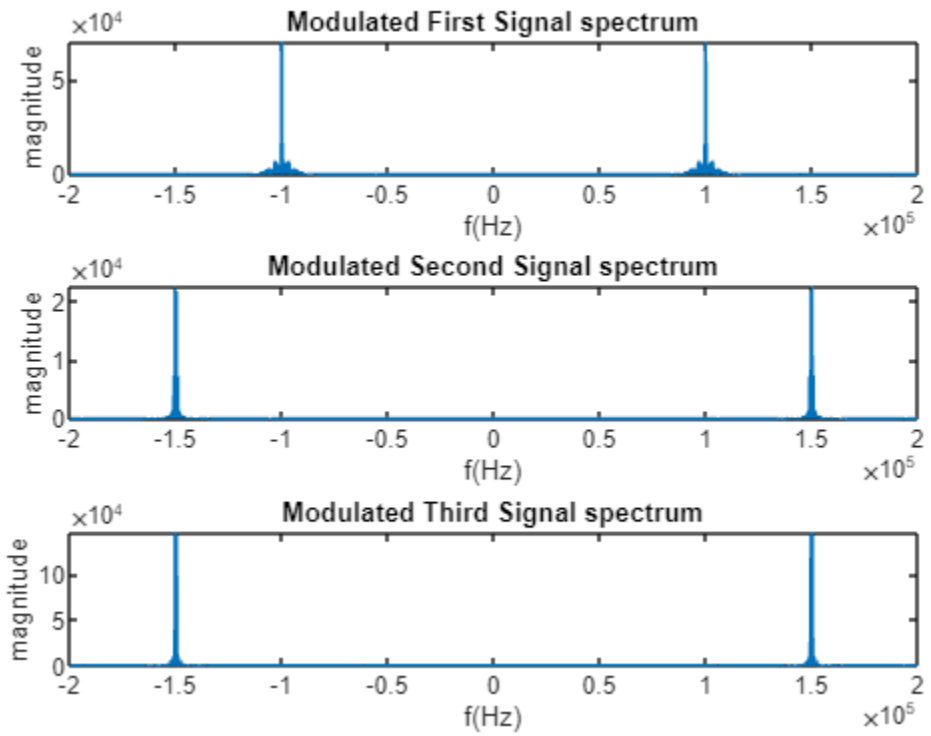
Three Signals in Time domain:



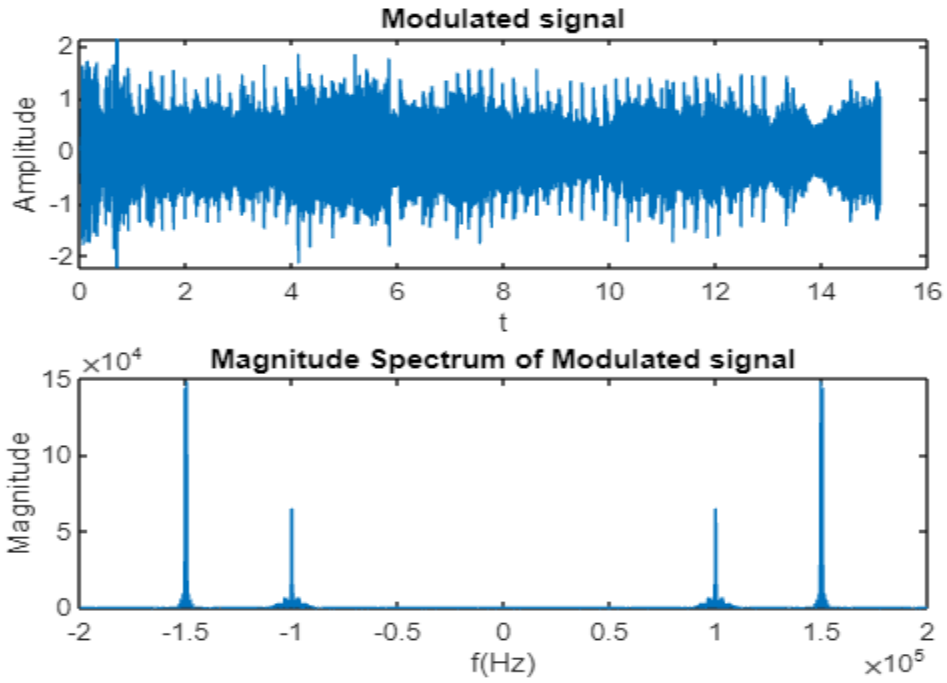
Three Signals in Frequency domain:



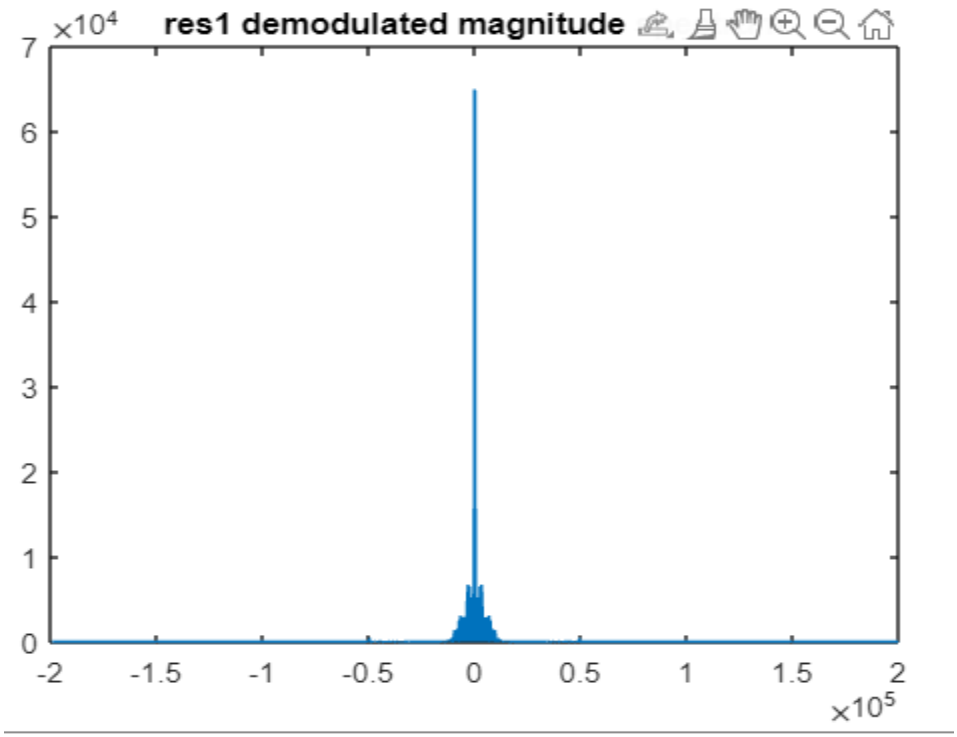
Three Modulated Signals in frequency domain:



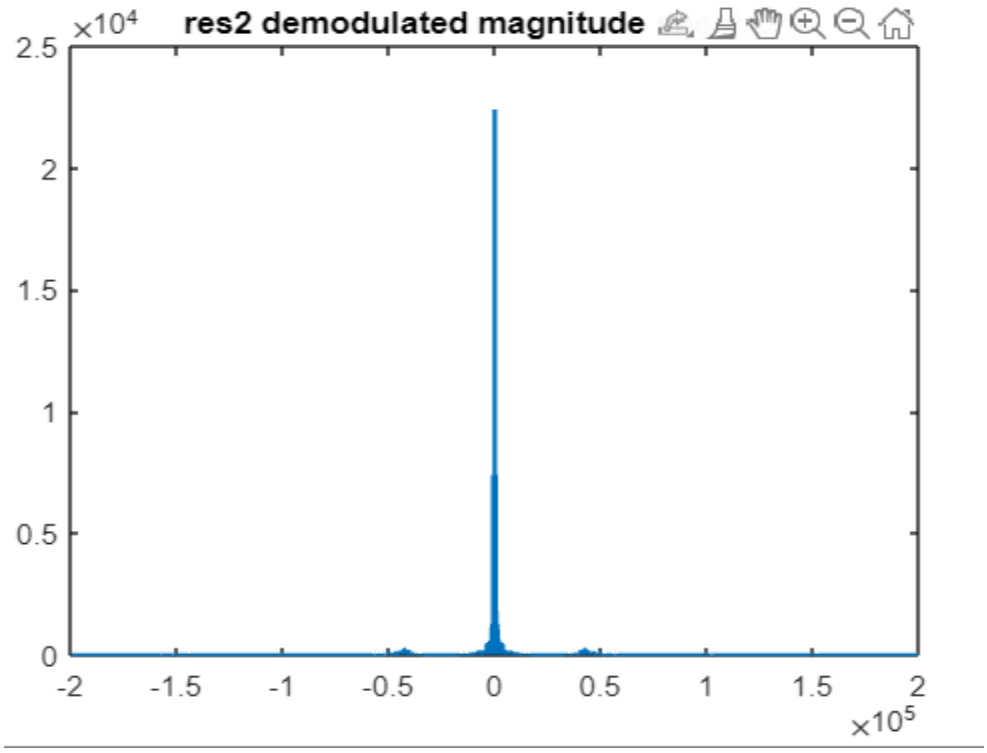
Modulated Signal: **s(t)=x_1 (t)*cos[ω1*t]+x_2 (t)*cos[ω2*t]+x_3 (t)*sin[ω2*t]**



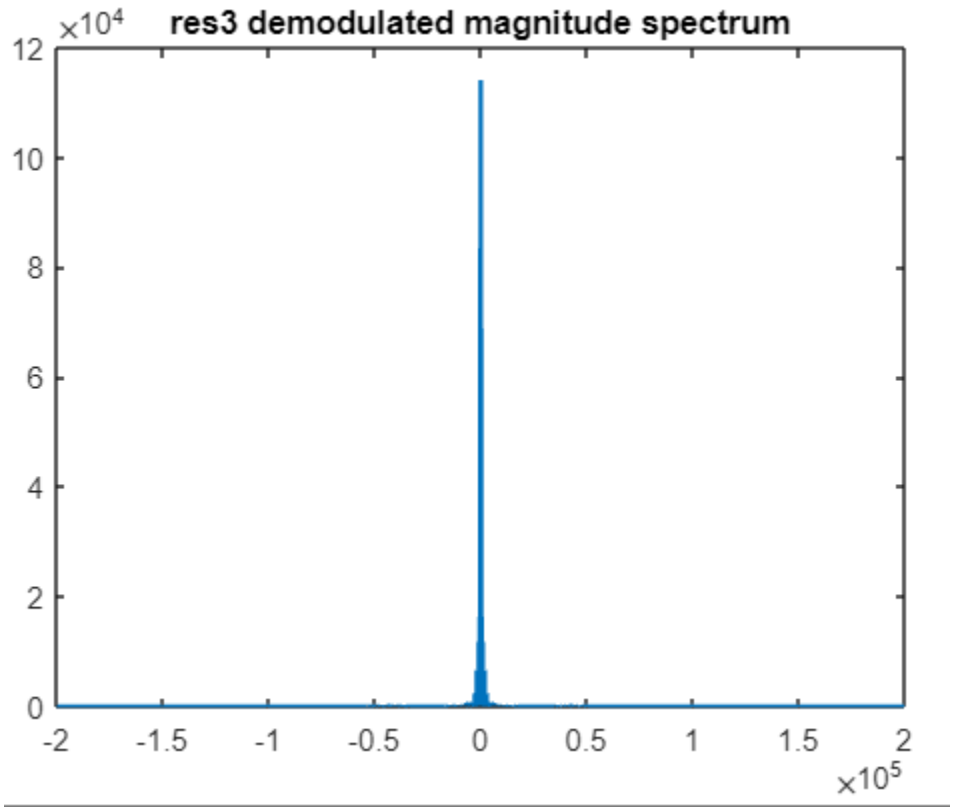
Result of performing synchronous demodulation in first signal:



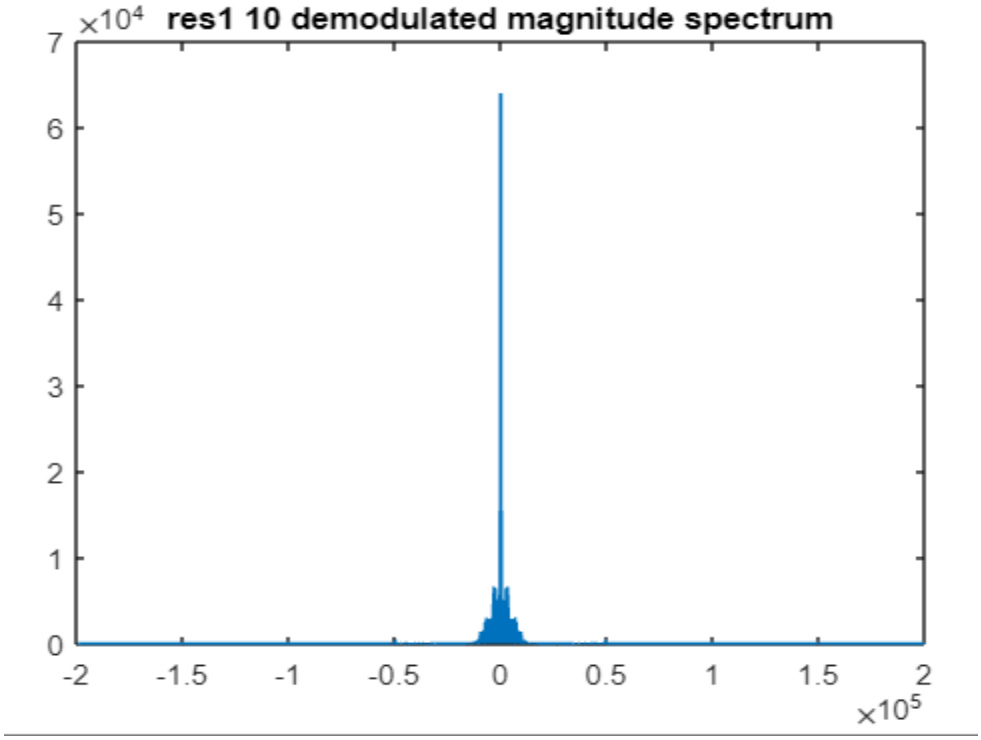
Result of performing synchronous demodulation in second signal:



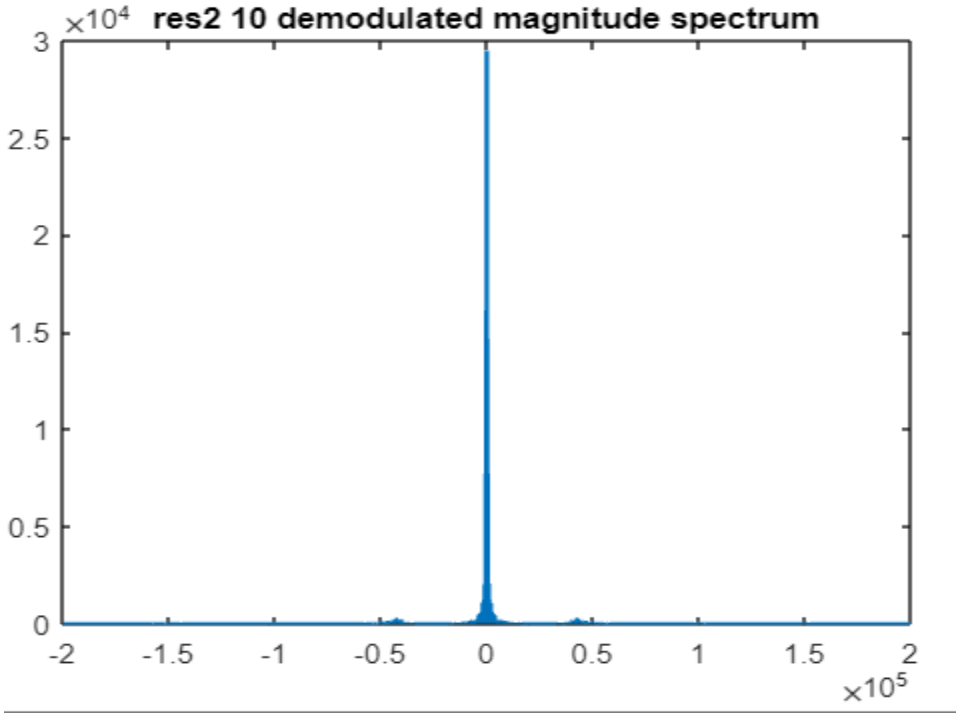
Result of performing synchronous demodulation in third signal:



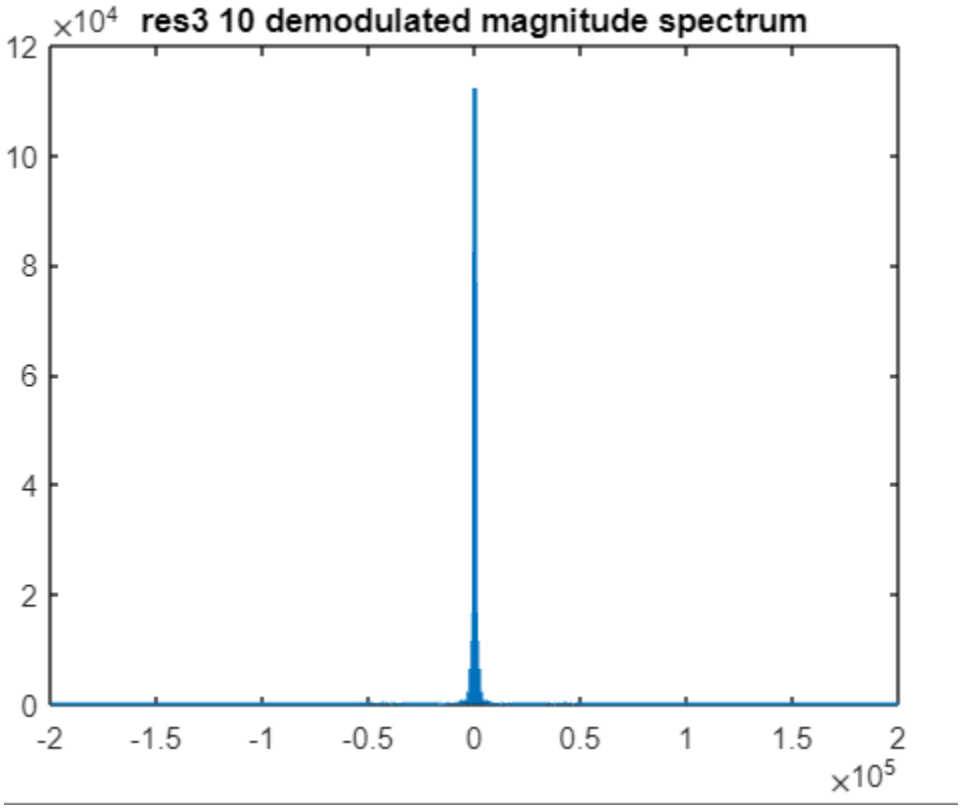
Result of performing demodulation with phase shift 10 in first signal:



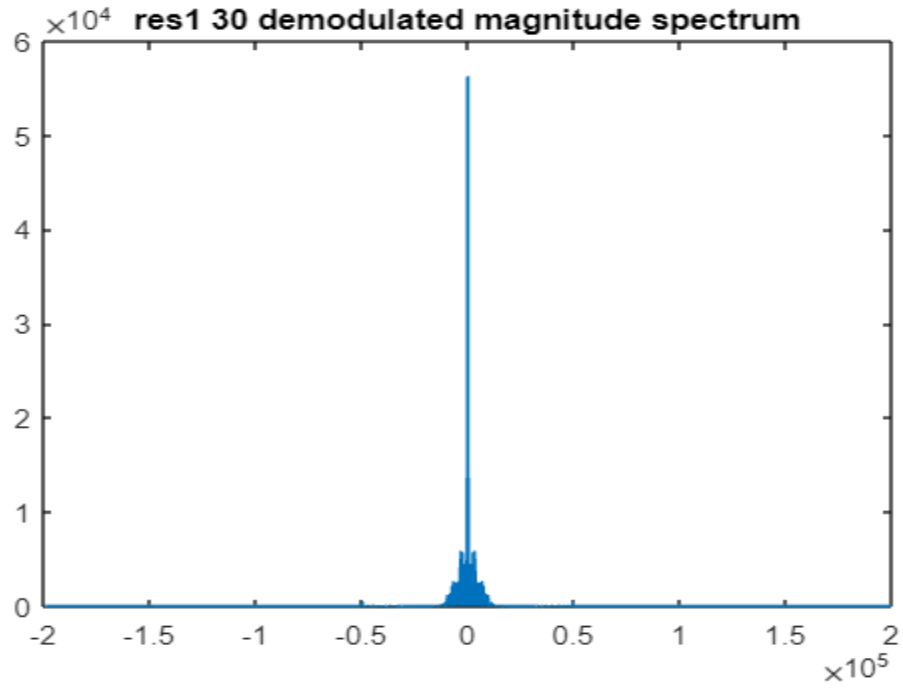
Result of performing demodulation with phase shift 10 in second signal:



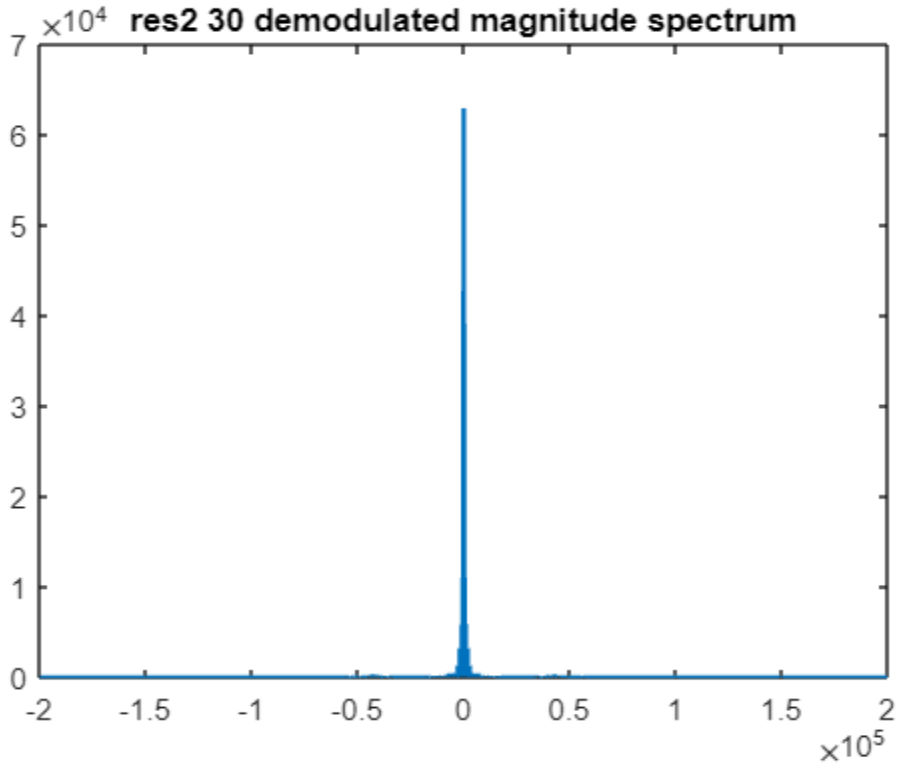
Result of performing demodulation with phase shift 10 in third signal:



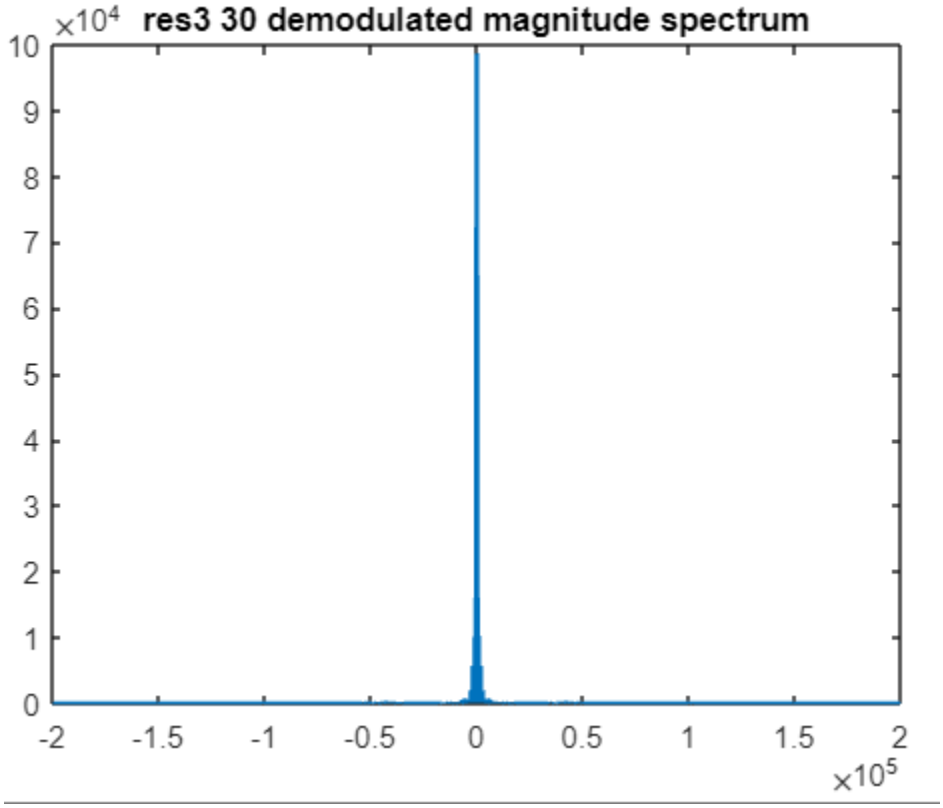
Result of performing demodulation with phase shift 30 in first signal:



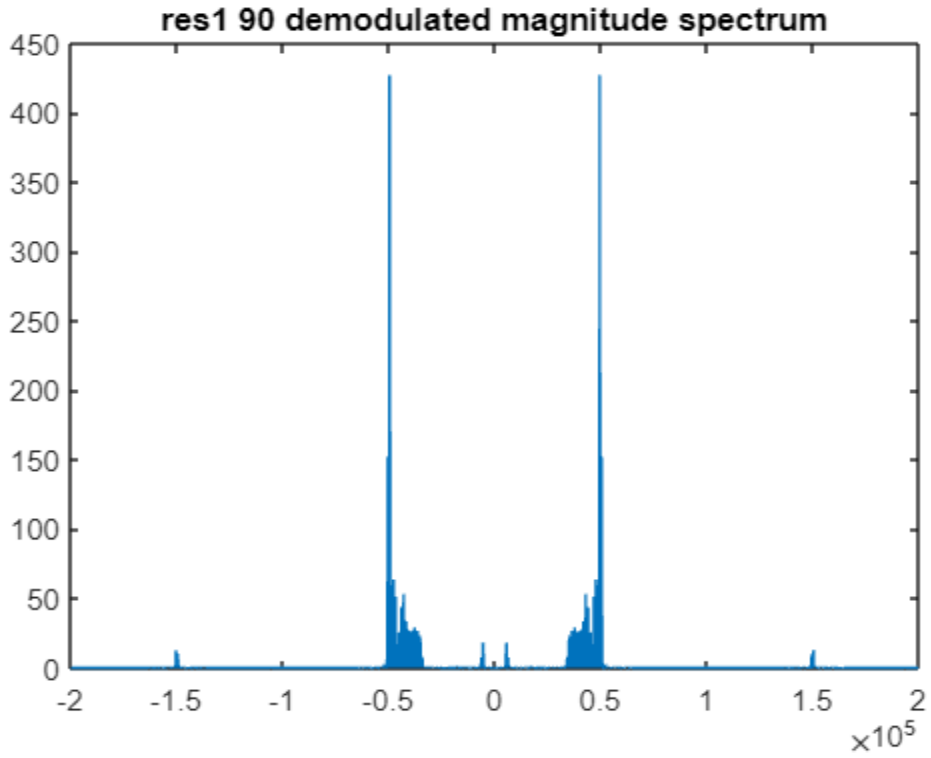
Result of performing demodulation with phase shift 30 in second signal:



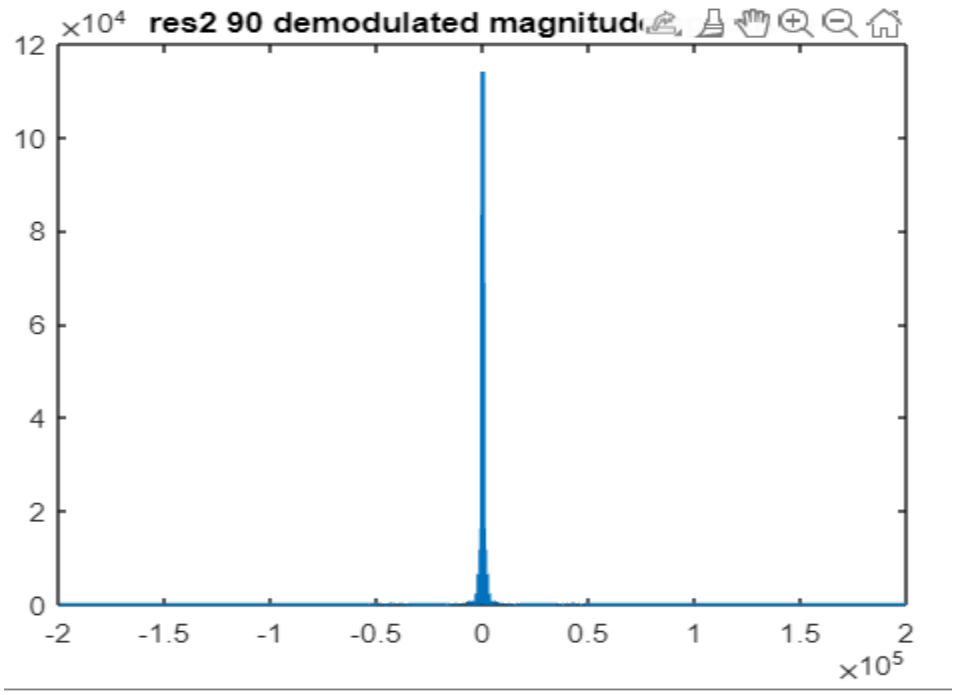
Result of performing demodulation with phase shift 30 in third signal:



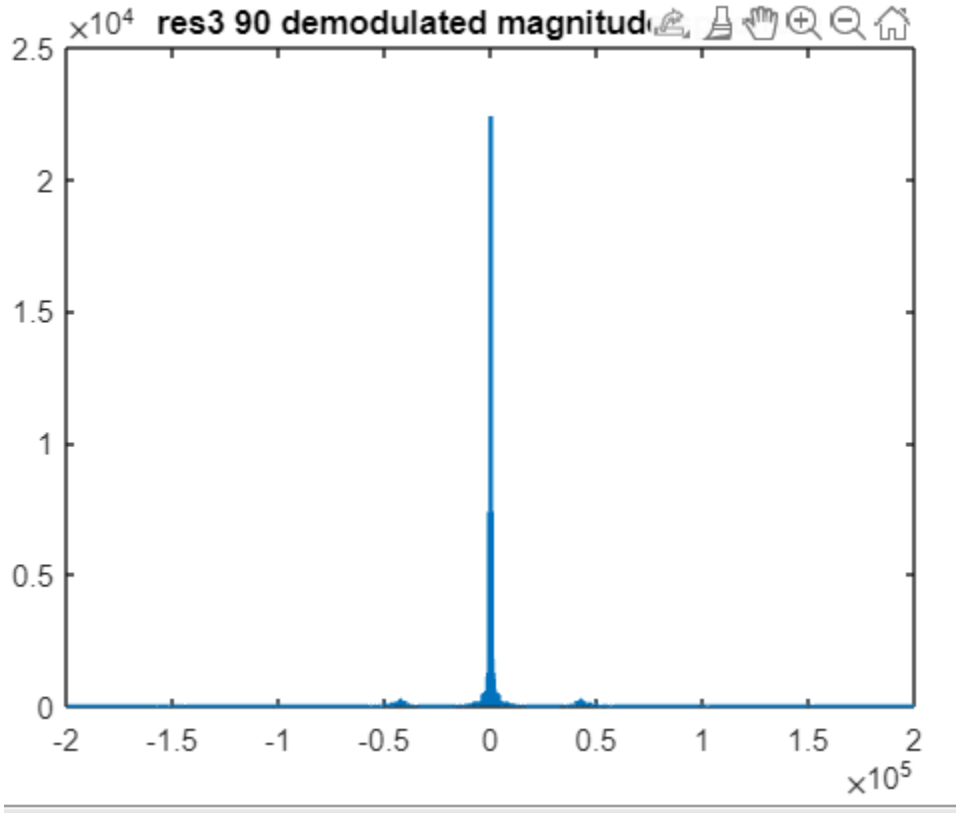
Result of performing demodulation with phase shift 90 in first signal:



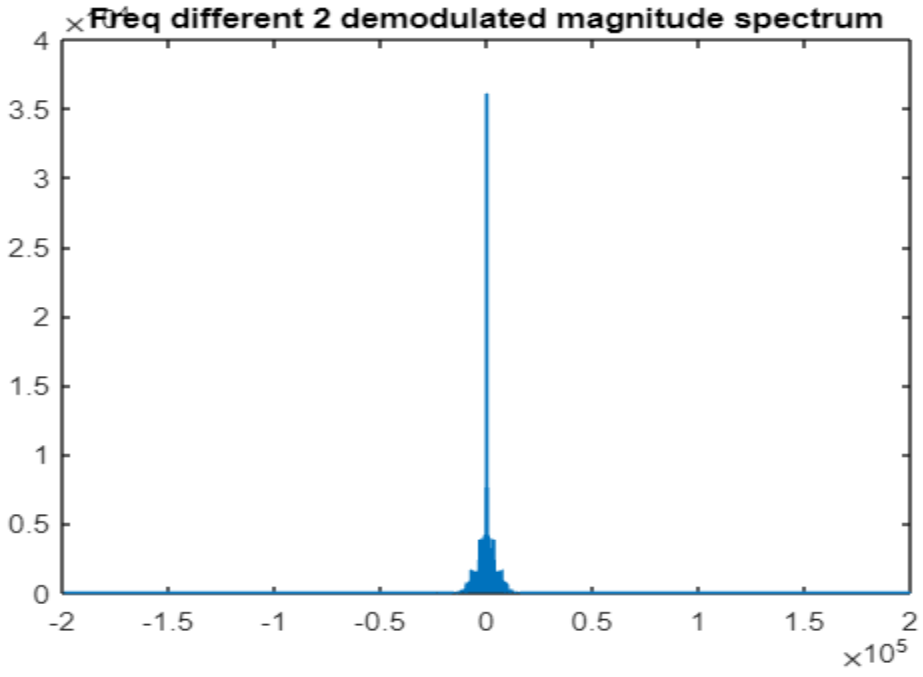
Result of performing demodulation with phase shift 90 in second signal:



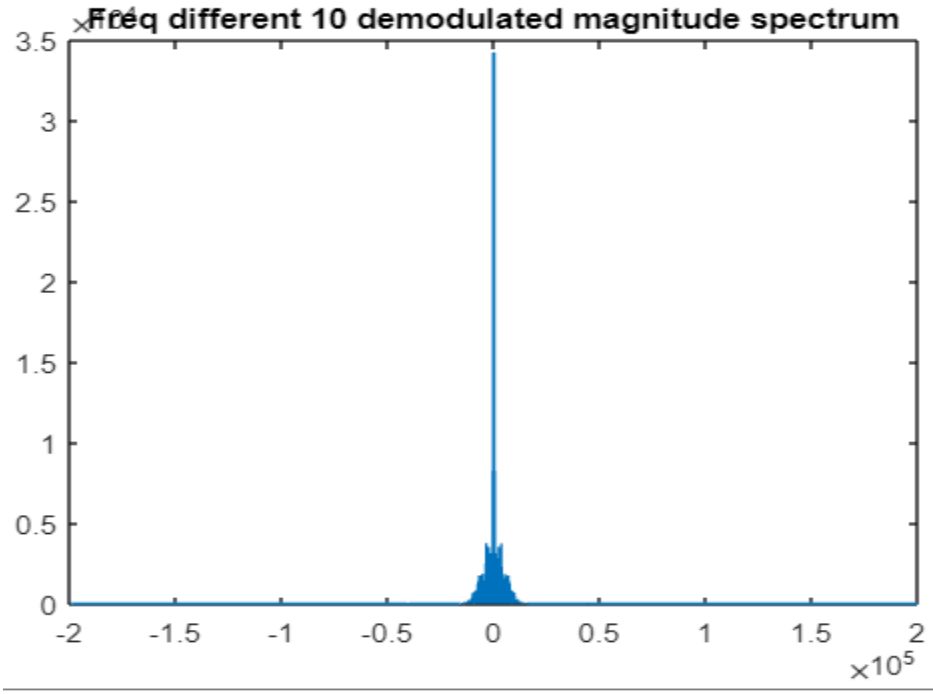
Result of performing demodulation with phase shift 90 in third signal:



Result of performing demodulation with a local carrier frequency that is different by 2 Hz from its carrier frequency:



Result of performing demodulation with a local carrier frequency that is different by 10 Hz from its carrier frequency:



The codes:

```
%////////////////////////////////////////////////////////%
%       Read audio files, Plotting in time domain       %
%////////////////////////////////////////////////////////%
clear;
% File 1
[First_Signal, Fs_First_Signal] = audioread("wav1.wav");
First_Signal = First_Signal(:,1) + First_Signal(:,2);
Length_First_Signal = length(First_Signal);
figure(1)
%plotting in time domain
subplot(3,1,1);
plot(First_Signal)
title("signal 1 time domain")
xlabel("n(samples)")
ylabel("1st Sig[n]")
% File 2
[Second_Signal, Fs_Second_Signal] = audioread("wav2.wav");
Second_Signal = Second_Signal(:,1) + Second_Signal(:,2);
Length_Second_Signal = length(Second_Signal);
subplot(3,1,2);
%plotting in time domain
plot(Second_Signal)
title("signal 2 time domain")
xlabel("n(samples)")
ylabel("2nd Sig[n]")
```

```
% File 3

[Third_Signal, Fs_Third_Signal] = audioread("wav3.wav");

Third_Signal = Third_Signal(:,1) + Third_Signal(:,2);

Length_Third_Signal = length(Third_Signal);

subplot(3,1,3);

%plotting in time domain

plot(Third_Signal)

title("signal 3 time domain")

xlabel("n(samples)")

ylabel("3rd Sig[n]")

%////////////////////////////////////////////////////////////////////////%

%        Convert to Frequency domain, Get the frequency increments        %

%////////////////////////////////////////////////////////////////////////%

% first signal

F_First_Signal=(-Fs_First_Signal/2:Fs_First_Signal/Length_First_Signal:Fs_Firs
t_Signal/2-Fs_First_Signal/Length_First_Signal);%frequency axis

FFT_First_Signal = abs(fft(First_Signal));

figure(2)

subplot(3,1,1);

plot(F_First_Signal, fftshift(FFT_First_Signal))

xlabel("f(Hz)")

ylabel("magnitude")

title("First Signal amplitudes against sampling frequency")

% second signal

F_Second_Signal=(-Fs_Second_Signal/2:Fs_Second_Signal/Length_Second_Signal:Fs_
Second_Signal/2-Fs_Second_Signal/Length_Second_Signal);%frequency axis

FFT_Second_Signal = abs(fft(Second_Signal));
```

```
subplot(3,1,2);

plot(F_Second_Signal, fftshift(FFT_Second_Signal))

xlabel("f(Hz)")

ylabel("magnitude")

title("Second Signal amplitudes against sampling frequency")

% third signal

F_Third_Signal=(-Fs_Third_Signal/2:Fs_Third_Signal/Length_Third_Signal:Fs_Thir
d_Signal/2-Fs_Third_Signal/Length_Third_Signal);%frequency axis

FFT_Third_Signal = abs(fft(Third_Signal));

subplot(3,1,3);

plot(F_Third_Signal, fftshift(FFT_Third_Signal))

xlabel("f(Hz)")

ylabel("magnitude")

title("Third Signal amplitudes against sampling frequency")

%////////////////////////////////////////////////////////%

%                  Resampling the signals                %

%////////////////////////////////////////////////////////%

%first signal

% fs larger than double band width of the largest band width

Fs_new = 400000;

[P, Q] = rat(Fs_new/Fs_First_Signal);

Resample_First_Signal = resample(First_Signal, P, Q);

%second signal

[P, Q] = rat(Fs_new/Fs_Second_Signal);

Resample_Second_Signal = resample(Second_Signal, P, Q);

%third signal
```

```
[P, Q] = rat(Fs_new/Fs_Third_Signal);
Resample_Third_Signal = resample(Third_Signal, P, Q);
%////////////////////////////////////////////////////////%
%            Modulation, Generate the carrier            %
%////////////////////////////////////////////////////////%
%larger than largest bandwidth
freq1 = 100000;
freq2 = 150000;
% first carrier
Test_Time_Scale_First = (0:length(Resample_First_Signal) - 1) * (1/Fs_new);
carrier1 = cos(2*pi*(freq1) * Test_Time_Scale_First);
% second carrier
Test_Time_Scale_Second = (0:length(Resample_Second_Signal) - 1) * (1/Fs_new);
carrier2 = cos(2*pi*(freq2) * Test_Time_Scale_Second);
% third carrier
Test_Time_Scale_Third = (0:length(Resample_Third_Signal) - 1) * (1/Fs_new);
carrier3 = sin(2*pi*(freq2) * Test_Time_Scale_Third);
%////////////////////////////////////////////////////////%
%                Modulating the  carriers                %
%////////////////////////////////////////////////////////%
%Modulaing first signal
Mod_First_Signal = Resample_First_Signal .* carrier1';
%Modulaing second signal
Mod_Second_Signal = Resample_Second_Signal .* carrier2';
%Modulaing third signal
Mod_Third_Signal = Resample_Third_Signal .* carrier3';
```

```
%//////////////////////////////////////////////////////%

%  Get the frequency increments, Plot Modulated signals  %

%//////////////////////////////////////////////////////%

%Modulated first signal

Length_Mod_First_Signal = length(Mod_First_Signal);

F_M_First_Signal = (Fs_new/Length_Mod_First_Signal)  *
(-Length_Mod_First_Signal/2: Length_Mod_First_Signal/2 - 1);

FFT_M_First_Signal = abs(fft(Mod_First_Signal));

figure(3)

subplot(3,1,1);

plot(F_M_First_Signal, fftshift(FFT_M_First_Signal))

xlabel("f(Hz)")

ylabel("magnitude")

title("Modulated First Signal spectrum")

%Modulated second signal

Length_Mod_Second_Signal = length(Mod_Second_Signal);

F_M_Second_Signal = (Fs_new/Length_Mod_Second_Signal)  *
(-Length_Mod_Second_Signal/2: Length_Mod_Second_Signal/2 - 1);

FFT_M_Second_Signal = abs(fft(Mod_Second_Signal));

subplot(3,1,2);

plot(F_M_Second_Signal, fftshift(FFT_M_Second_Signal))

xlabel("f(Hz)")

ylabel("magnitude")

title("Modulated Second Signal spectrum")

%Modulated third signal

Length_Mod_Third_Signal = length(Mod_Third_Signal);
```

```
F_M_Third_Signal = (Fs_new/Length_Mod_Third_Signal)  *
(-Length_Mod_Third_Signal/2: Length_Mod_Third_Signal/2 - 1);

FFT_M_Third_Signal = abs(fft(Mod_Third_Signal));

subplot(3,1,3);

plot(F_M_Third_Signal, fftshift(FFT_M_Third_Signal))

xlabel("f(Hz)")

ylabel("magnitude")

title("Modulated Third Signal spectrum")

%////////////////////////////////////////////////////////%

%                 Adding the modulated Signal              %

%////////////////////////////////////////////////////////%

max_len = max(Length_Mod_First_Signal, max(Length_Mod_Second_Signal,
Length_Mod_Third_Signal));

sig1 = [Mod_First_Signal;zeros(max_len-Length_Mod_First_Signal, 1)];

sig2 = [Mod_Second_Signal;zeros(max_len-Length_Mod_Second_Signal, 1)];

sig3 = [Mod_Third_Signal;zeros(max_len-Length_Mod_Third_Signal, 1)];

mod_s = sig1 + sig2 + sig3;

t_ms = (0: max_len - 1) * (1 / Fs_new);

f_ms = (-max_len/2 : max_len/2 - 1) * (Fs_new / max_len);

figure(4)

subplot(2,1,1);

plot(t_ms, mod_s);xlabel("t");ylabel("Amplitude")

title("Modulated signal")

fft_mod_s = abs(fft(mod_s));

subplot(2,1,2);

plot(f_ms, fftshift(fft_mod_s))
```

```
xlabel("f(Hz)");ylabel("Magnitude");title("Magnitude Spectrum of Modulated
signal")

%////////////////////////////////////////////////////////%

%             Demodulation synchronous carriers          %

%////////////////////////////////////////////////////////%

%time

tdem = (0:max_len - 1) * (1/Fs_new);

%first sync carrier

sync_carr1 = cos(2*pi*(freq1) * tdem);

%second sync carrier

sync_carr2 = cos(2*pi*(freq2) * tdem);

%third sync carrier

sync_carr3 = sin(2*pi*(freq2) * tdem);

%////////////////////////////////////////////////////////%

%                  Demodulation & Filtering              %

%////////////////////////////////////////////////////////%

f_res = (-max_len/2 : max_len/2 - 1) * (Fs_new / max_len);

%demodulation first signal

demodulate_signal(mod_s, sync_carr1, Length_First_Signal, Fs_First_Signal,
Fs_new, 24000, f_res, 'res1')

%demodulation second signal

demodulate_signal(mod_s, sync_carr2, Length_Second_Signal, Fs_Second_Signal,
Fs_new, 24000, f_res, 'res2')

%demodulation third signal

demodulate_signal(mod_s, sync_carr3, Length_Third_Signal, Fs_Third_Signal,
Fs_new, 24000, f_res, 'res3')

%////////////////////////////////////////////////////////%

%                       Phase shift 10                   %
```

```
%/////////////////////////////////////////////////////////%

[carr1_10, carr2_10, carr3_10] = gen_carriers(freq1, freq2, tdem, 10);

%demodulation first signal phase 10

demodulate_signal(mod_s, carr1_10, Length_First_Signal, Fs_First_Signal,
Fs_new, 24000, f_res, 'res1 10')

%demodulation second signal phase 10

demodulate_signal(mod_s, carr2_10, Length_Second_Signal, Fs_Second_Signal,
Fs_new, 24000, f_res, 'res2 10')

%demodulation third signal phase 10

demodulate_signal(mod_s, carr3_10, Length_Third_Signal, Fs_Third_Signal,
Fs_new, 24000, f_res, 'res3 10')

%/////////////////////////////////////////////////////////%

%                       Phase shift 30                    %

%/////////////////////////////////////////////////////////%

[carr1_30, carr2_30, carr3_30] = gen_carriers(freq1, freq2, tdem, 30);

%demodulation first signal phase 30

demodulate_signal(mod_s, carr1_30, Length_First_Signal, Fs_First_Signal,
Fs_new, 24000, f_res, 'res1 30')

%demodulation second signal phase 30

demodulate_signal(mod_s, carr2_30, Length_Second_Signal, Fs_Second_Signal,
Fs_new, 24000, f_res, 'res2 30')

%demodulation third signal phase 30

demodulate_signal(mod_s, carr3_30, Length_Third_Signal, Fs_Third_Signal,
Fs_new, 24000, f_res, 'res3 30')

%/////////////////////////////////////////////////////////%

%                       Phase shift 90                    %

%/////////////////////////////////////////////////////////%

[carr1_90, carr2_90, carr3_90] = gen_carriers(freq1, freq2, tdem, 90);

%demodulation first signal phase 90
```

```
demodulate_signal(mod_s, carr1_90, Length_First_Signal, Fs_First_Signal,
Fs_new, 24000, f_res, 'res1 90')

%demodulation second signal phase 90

demodulate_signal(mod_s, carr2_90, Length_Second_Signal, Fs_Second_Signal,
Fs_new, 24000, f_res, 'res2 90')

%demodulation third signal phase 90

demodulate_signal(mod_s, carr3_90, Length_Third_Signal, Fs_Third_Signal,
Fs_new, 24000, f_res, 'res3 90')

%////////////////////////////////////////////////////////%

%                Demodulated with different freq          %

%////////////////////////////////////////////////////////%

%time

tdem = (0:max_len - 1) * (1/Fs_new);

%first carrier

carrier_freq_2 = cos(2*pi*(freq1 + 2) * tdem);

demodulate_signal(sig1, carrier_freq_2, Length_First_Signal, Fs_First_Signal,
Fs_new, 24000, f_res, 'Freq different 2')

%second carrier

carrier_freq_10 = cos(2*pi*(freq1 + 10) * tdem);

demodulate_signal(sig1, carrier_freq_10, Length_First_Signal, Fs_First_Signal,
Fs_new, 24000, f_res, 'Freq different 10')

%////////////////////////////////////////////////////////%

%                demodulate_signal Function               %

%////////////////////////////////////////////////////////%

function  demodulate_signal(s, carr, old_len, fs_old, fs_new, f_pass, f_res,
filename)

   res_s = s .* carr';

   res = lowpass(res_s, f_pass , fs_new);
```

```
    fft_res_s = abs(fft(res));
    figure()
    plot(f_res, fftshift(fft_res_s))
    title(strcat(filename, " demodulated magnitude spectrum"))

    [P, Q] = rat(fs_old/fs_new);
    res = resample(res, P, Q);

    % clip to original size
    res = res(1: old_len);

    audiowrite(strcat(filename, '.wav'), res, fs_old)
end
%////////////////////////////////////////////////////////%
%                   gen_carriers Function                %
%////////////////////////////////////////////////////////%
function [c1, c2, c3] = gen_carriers(f1, f2, tdim, phase_shift_deg)
    phase_shift_rad = (phase_shift_deg * pi) / 180;
    c1 = cos(2*pi*f1*tdim + phase_shift_rad);
    c2 = cos(2*pi*f2*tdim + phase_shift_rad);
    c3 = sin(2*pi*f2*tdim + phase_shift_rad);
end
```