Communication Project

Ziad Sherif Muhammed BN:27 Sec:1

Eslam Ashraf Ibrahim BN:13 Sec:1

a) Our Work Explanation:

Firstly, we read three signals with different sounds and make them have the same frequency sampling =250000 by make resemble to them and find the length of each one along with max length in-order to be able to sum all modulated signal of each input. We make all audios have the same length by adjusting all of them by adding zero to make have same length. Secondly got time and frequencies intervals. Then we calculated omega **(ω1, ω2)** by **ω=2*pi*const** in frequency domain. Then we got carrier in cos and sin domain

Carrier Signal One=cos (2*pi* ω1)

Carrier Signal Two=sin (2*pi* ω2)

Modulation:

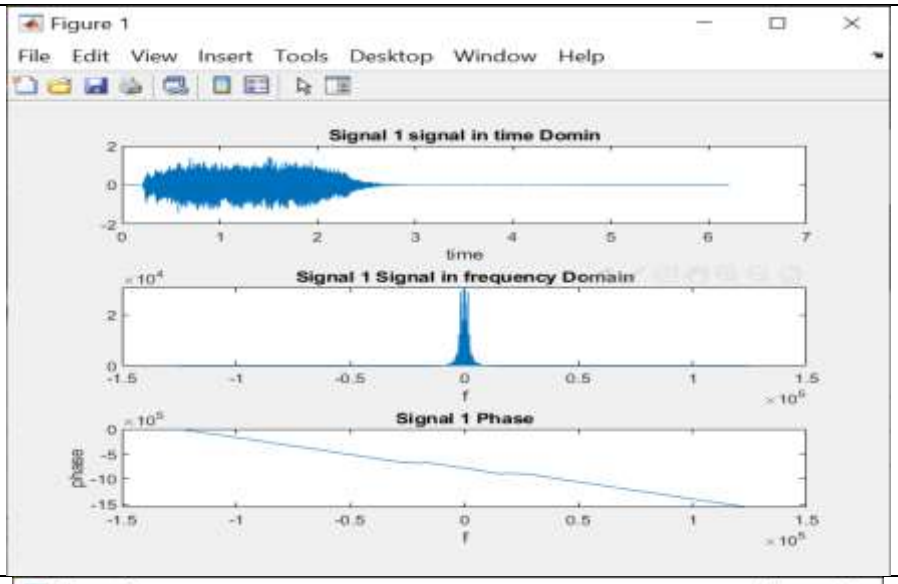
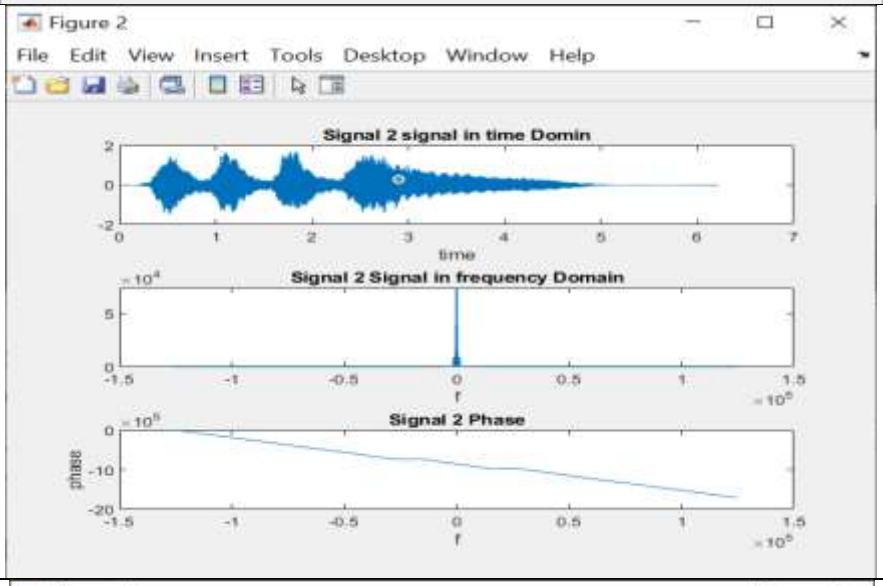
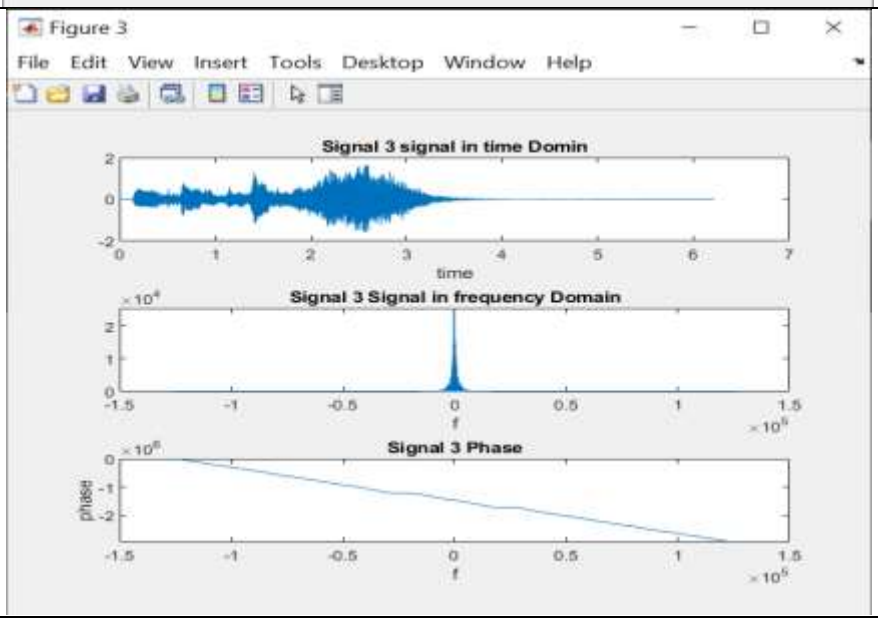
1. Calculate modulated signal by multiplying signal and carrier
2. Calculate Fourier transform of modulated signal returned from step 1
3. Calculate phase of modulated signal returned from step 1
4. Sum all modulated signals of all audios
5. FFT to summation of modulated signals
6. Calculate phase of modulated signals
7. Calculate frequency band pass which is used in de-modulation

De-modulation:

1. Calculate demodulated signal by multiplying carrier and modulation signal
2. Calculate low pass filter used frequency sampling.
3. Perform demodulation three times with phase shifts of 10, 30, 90 degrees for both carriers **note (before each phase shift calculate carrier phase)**.
4. perform demodulation two times with a local carrier frequency that is different by 2 Hz and 10 Hz from its carrier frequency.

Required results and answers to questions:

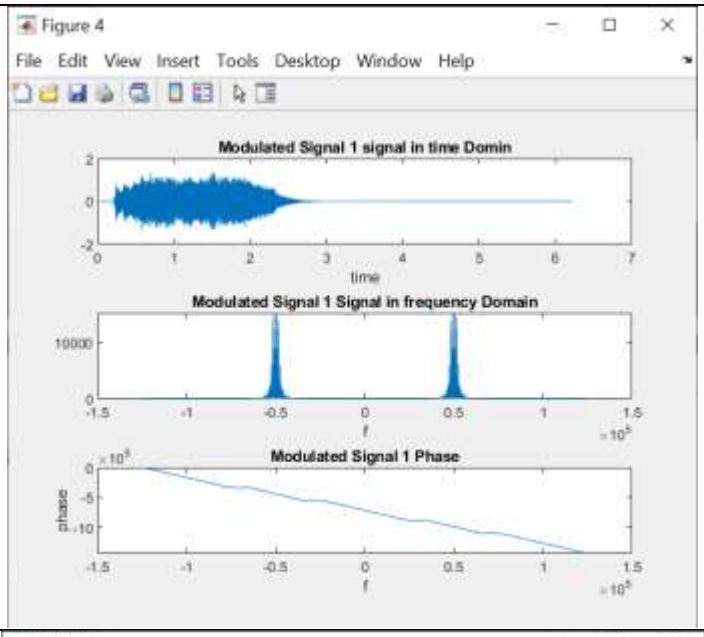
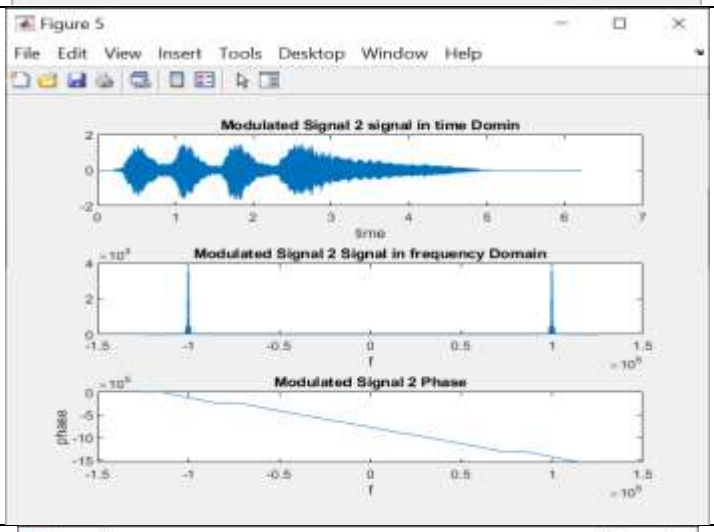
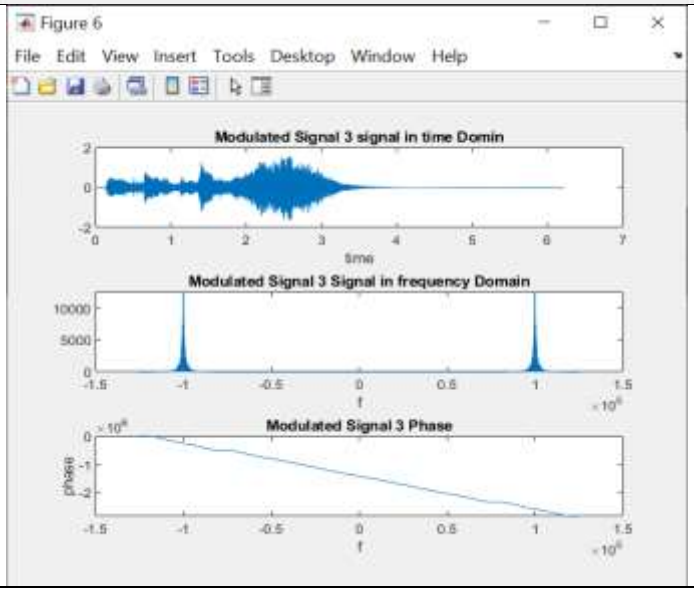
Input signals

Signal 1	
Signal 2	
Signal 3	

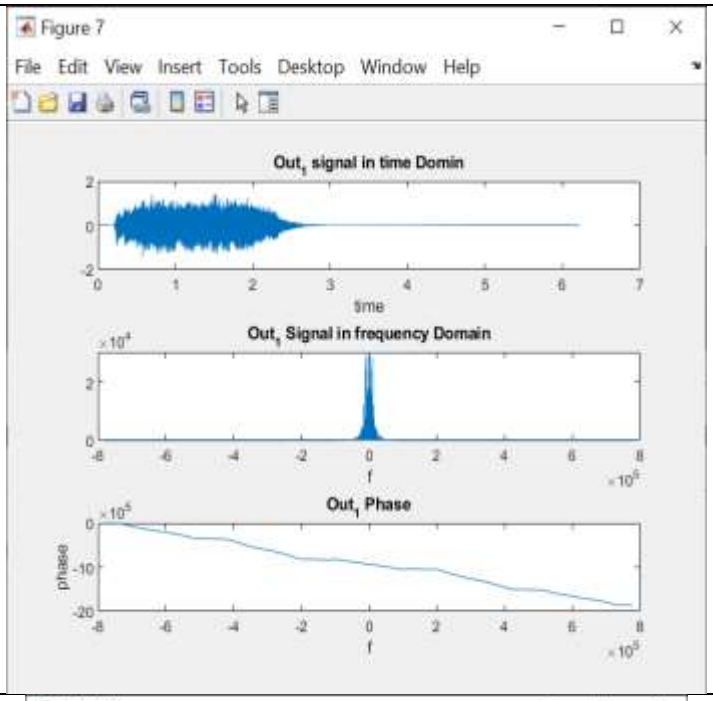
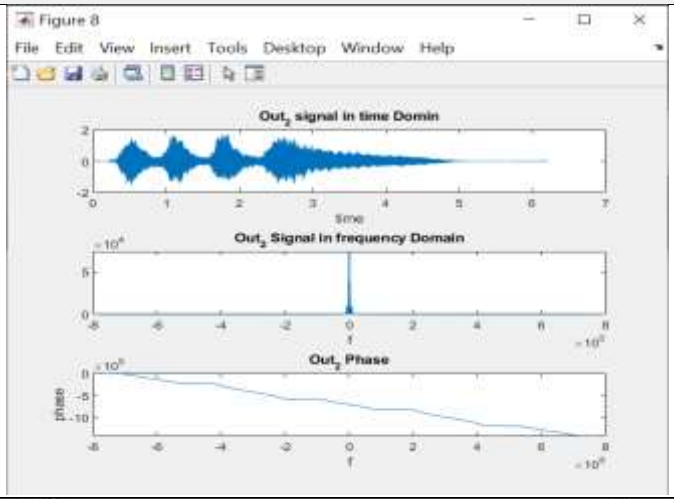
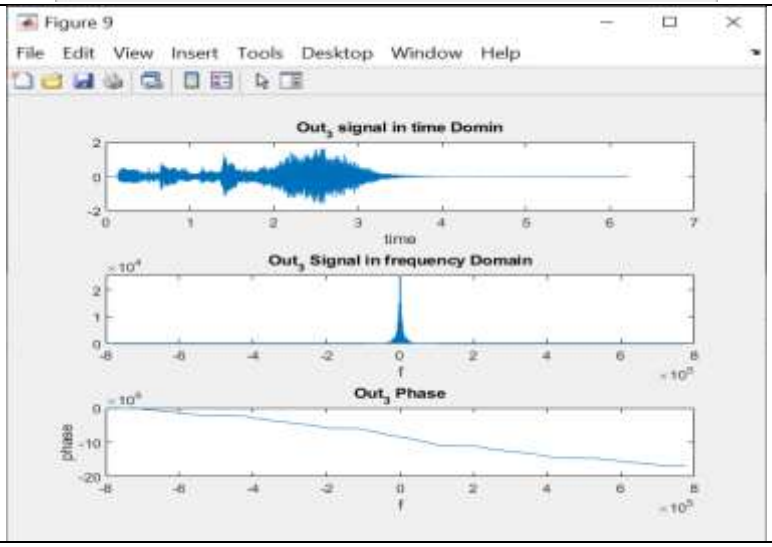
Modulation:

w1 = 2*pi*500000

w1 = 2*pi*1000000

Signal 1		
Signal 2		
Signal 3		

Synchronous Demodulation:

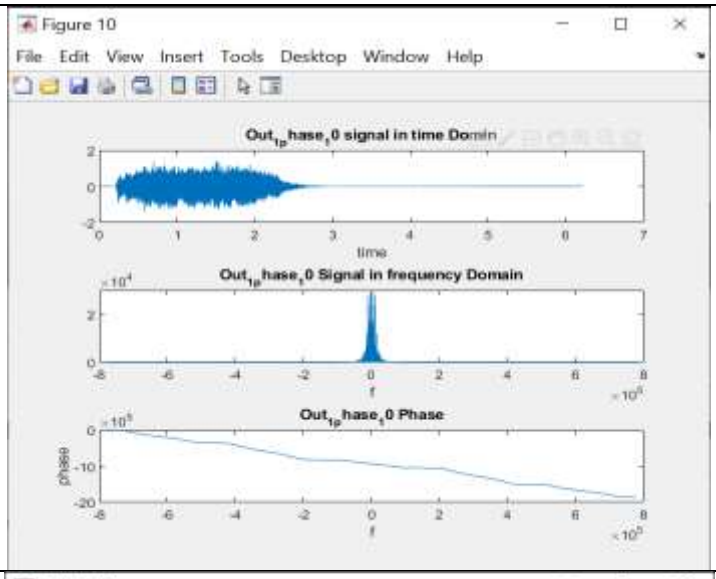
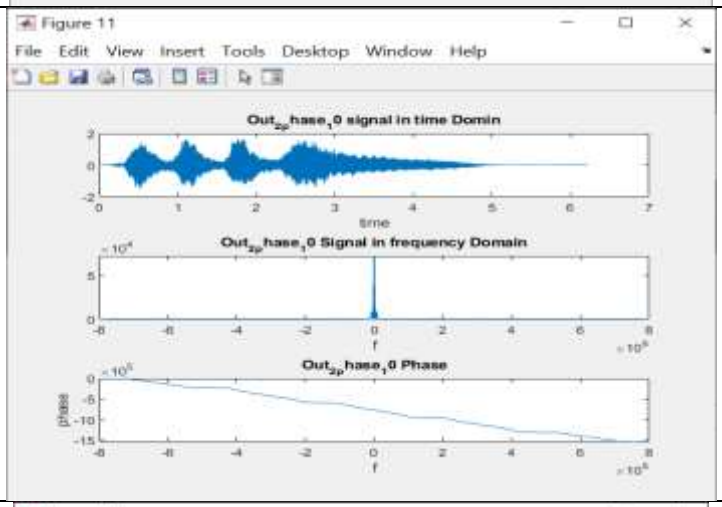
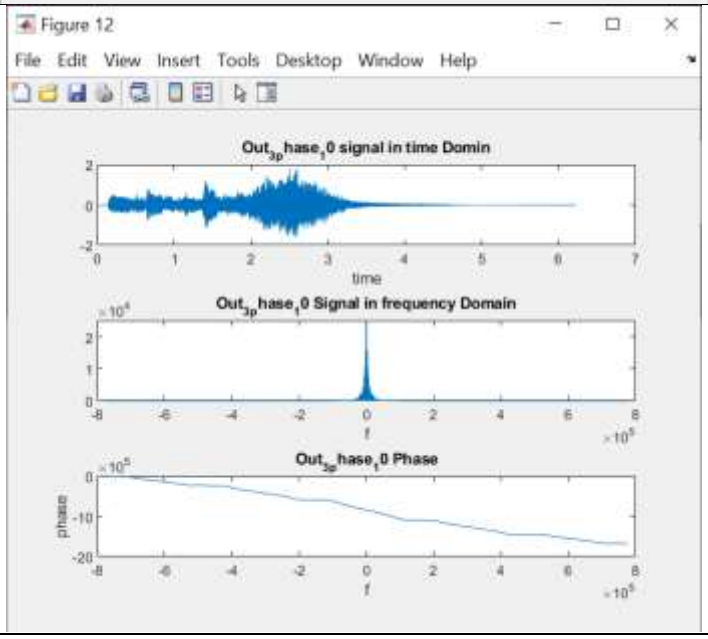
Signal 1	
Signal 2	
Signal 3	

Comment:

Multiply output *2 because **M(t)[output] =0.5*M(t)[original]**

Output is less weaker than original sound

Demodulation with phase shift 10:

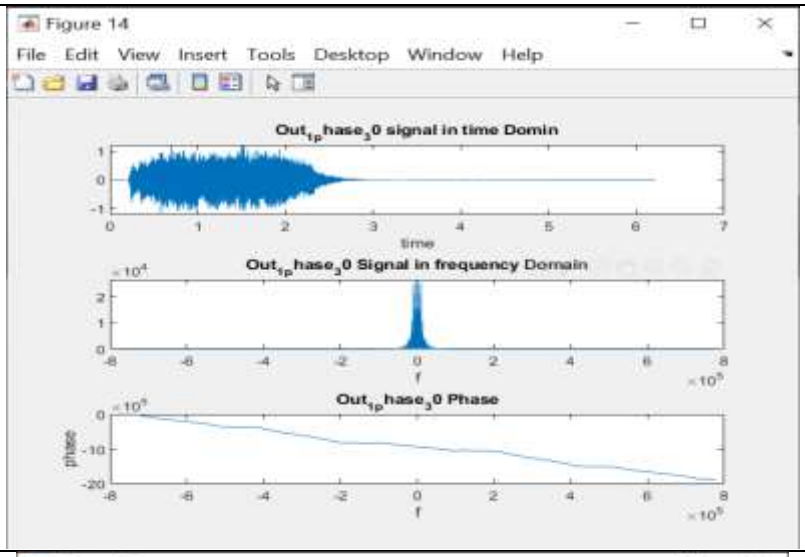
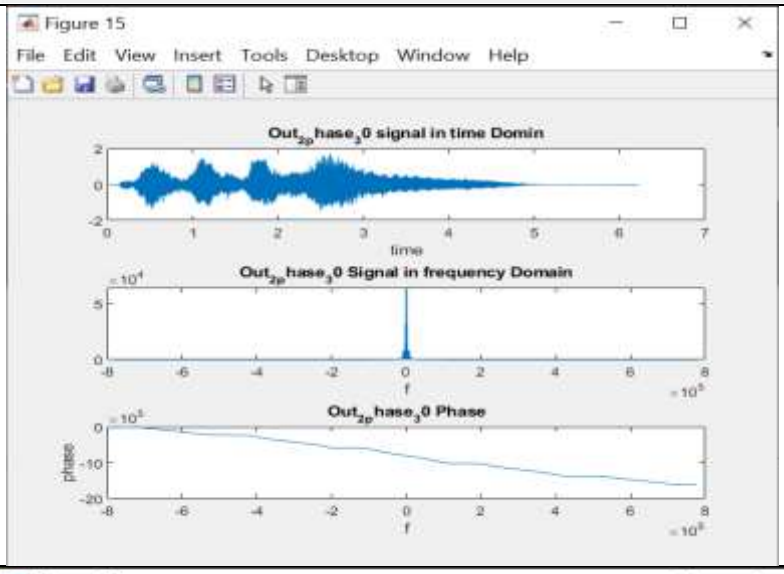
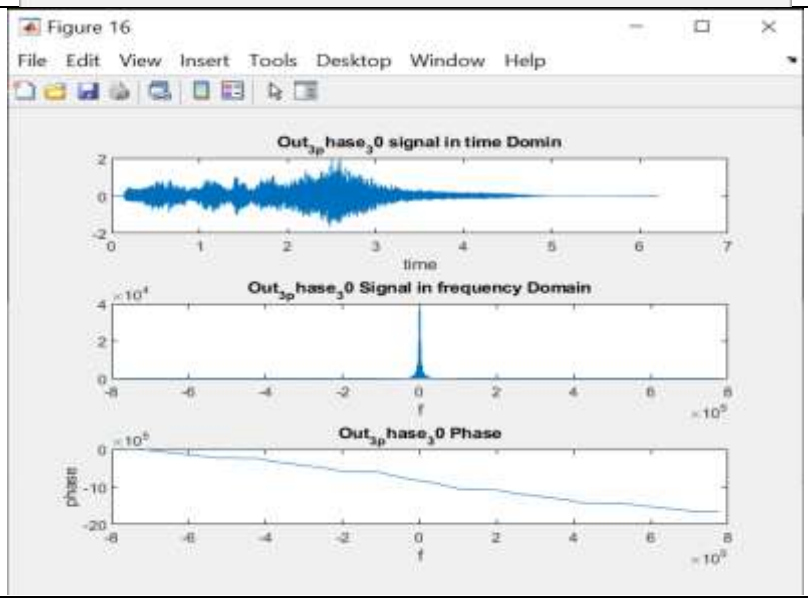
Signal 1		
Signal 2		
Signal 3		

Comment:

Interference in sounds between signal 2 and signal 3 when make demodulation to them.

Signal 1 become little weaker.

Demodulation with phase shift 30:

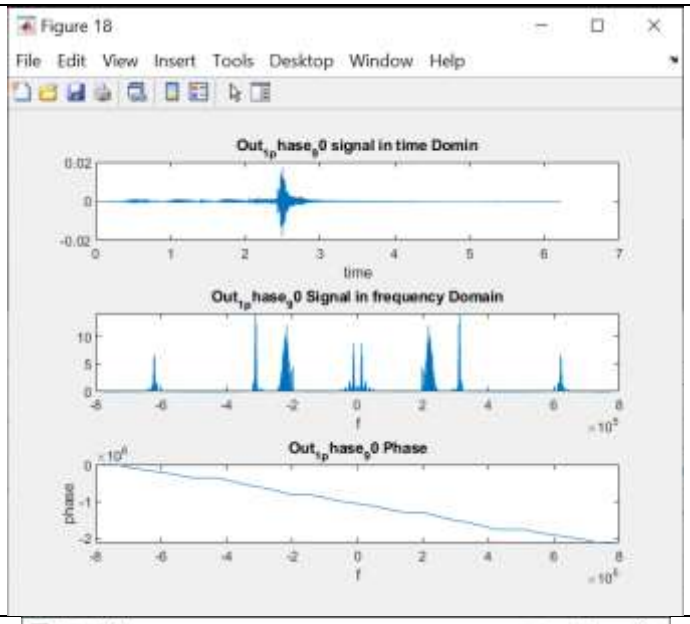
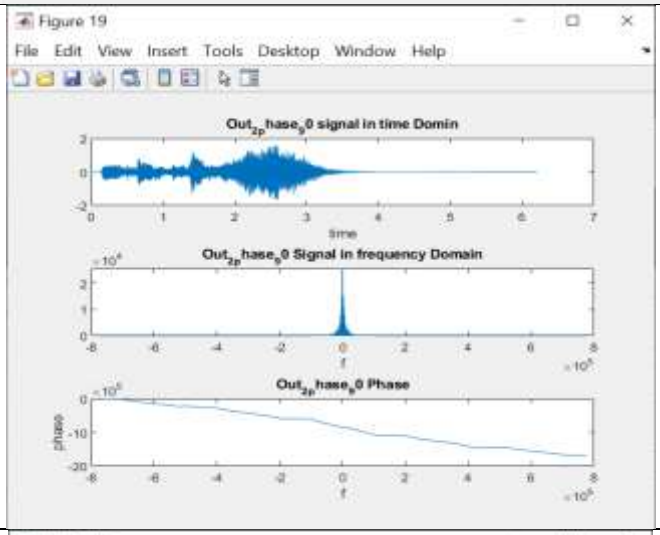
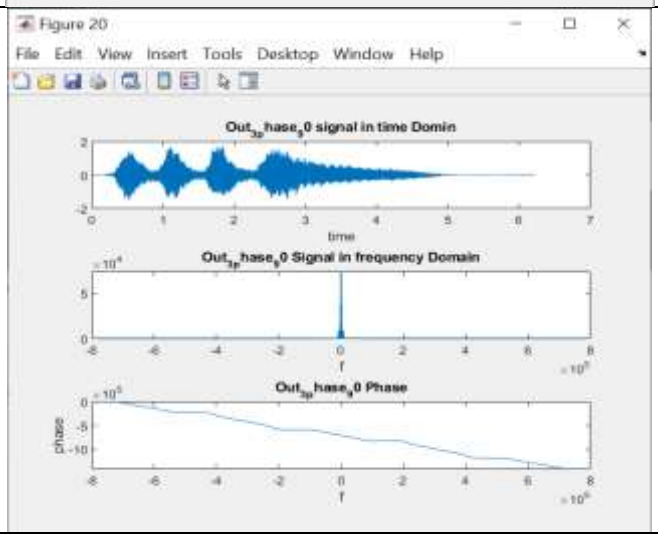
Signal 1	
Signal 2	
Signal 3	

Comment:

More Interference in sounds between signal 2 and signal 3 when make demodulation to them.

Signal 1 become Weaker.

Demodulation with phase shift 90:

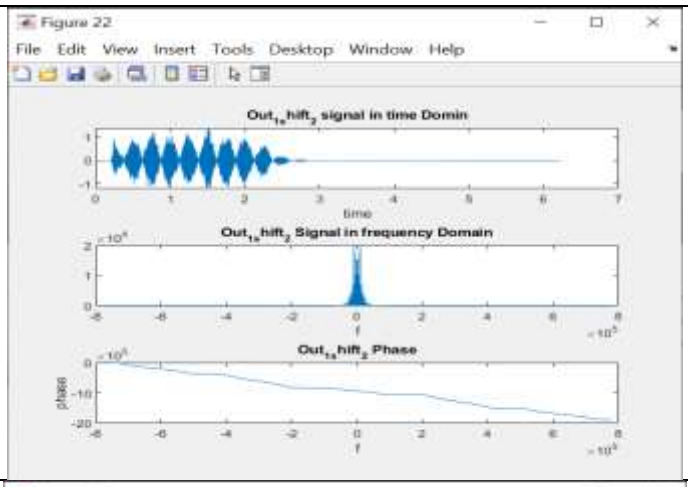
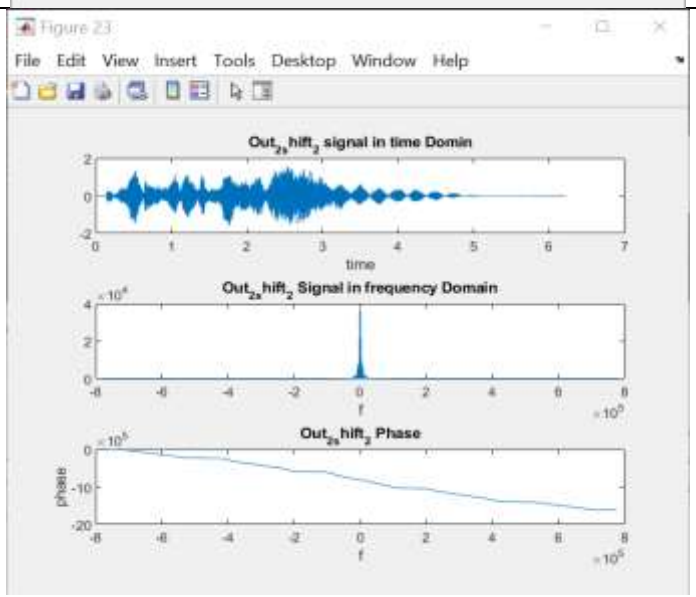
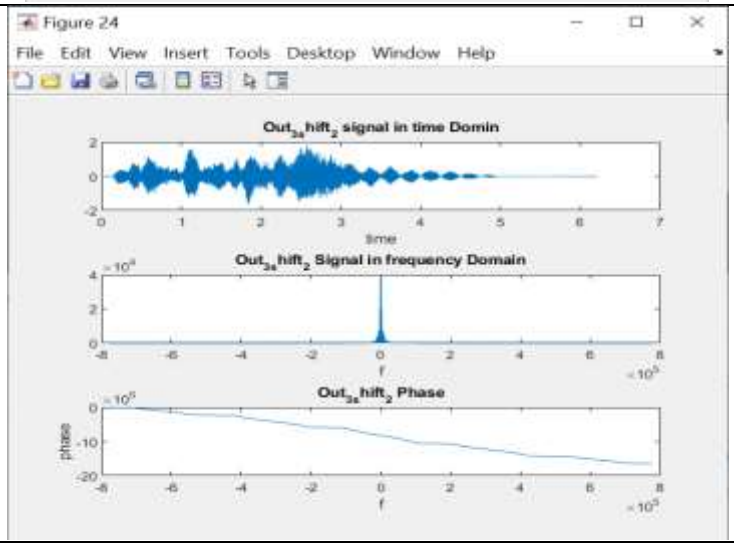
Signal 1		
Signal 2		
Signal 3		

Comment:

When make demodulation to signal 3 output is signal 2 and make demodulation to signal 2 output is signal3

Signal 1 be equal zero (no sound)

Demodulation with a local carrier frequency that is different by 2 Hz:

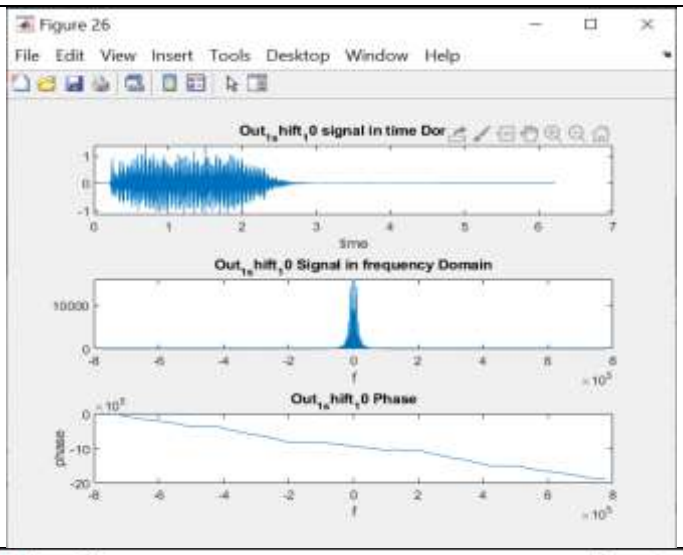
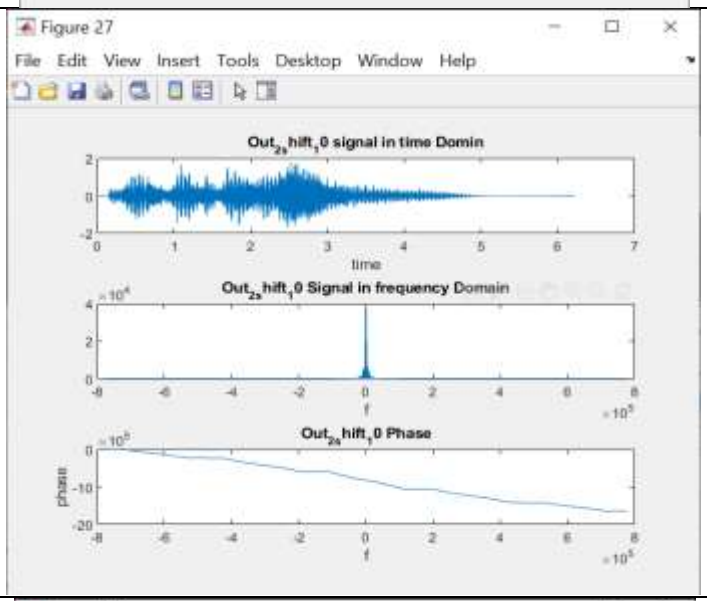
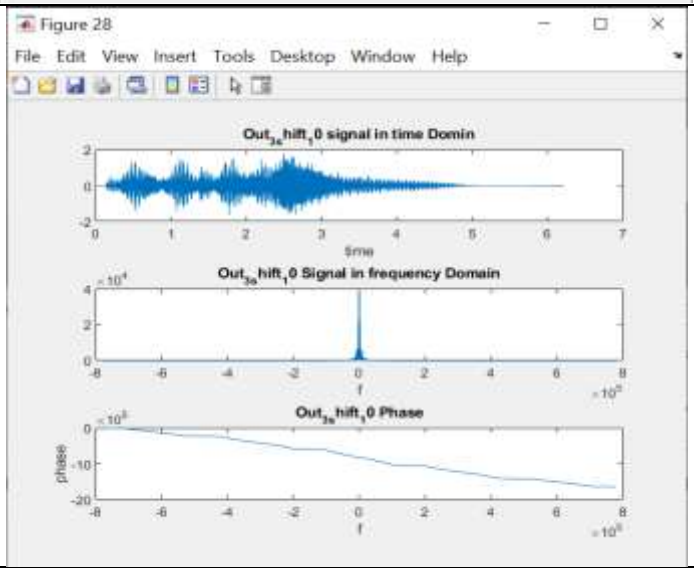
Signal 1	
Signal 2	
Signal 3	

Comment:

Interference in sounds between signal 2 and signal 3 when make demodulation to them.

All signals have been occurred distortion on them

Demodulation with a local carrier frequency that is different by 10 Hz:

Signal 1		
Signal 2		
Signal 3		

Comment:

More Interference in sounds between signal 2 and signal 3 when make demodulation to them.

All signals have been occurred more distortion on them

Codes:

```
%% Read Voice
[signal_1, frequency_sampling1] = audioread('signals/ziad.wav');
[signal_2, frequency_sampling2] = audioread('signals/esoo.wav');
[signal_3, frequency_sampling3] = audioread('signals/mohey.wav');

%%  Resampling the signals
FsNew = 250000;
frequency_sampling=FsNew;
ratio=FsNew/frequency_sampling1;
[temp_P, temp_Q] = rat(ratio);
signal_1 = resample(signal_1, temp_P, temp_Q);

ratio=FsNew/frequency_sampling2;
[temp_P, temp_Q] = rat(ratio);
signal_2 = resample(signal_2, temp_P, temp_Q);

ratio=FsNew/frequency_sampling3;
[temp_P, temp_Q] = rat(ratio);
signal_3 = resample(signal_3, temp_P, temp_Q);
%% Sum first and second channel
signal_1 = signal_1(:, 1)+signal_1(:, 2);
signal_2 = signal_2(:, 1)+ signal_2(:, 2);
signal_3 = signal_3(:, 1)+signal_3(:, 2);

%% set maxuim length between 3 signals
length_1 = length(signal_1);
length_2 = length(signal_2);
length_3 = length(signal_3);

N = max([length_1, length_2, length_3]);
```

```
%% make 3 signals be same
signal_1 = [signal_1;zeros(N-length_1, 1)];
signal_2 = [signal_2;zeros(N-length_2, 1)];
signal_3 = [signal_3;zeros(N-length_3, 1)];

%% get time and frequencyuency interval
t = linspace(0, N/frequency_sampling, N);
frequency_d = frequency_sampling/2;
frequency = -frequency_d : frequency_sampling/N: frequency_d - frequency_sampling/N;

%% frequencyuency Domin
signal_1_fft = fftshift(fft(signal_1));
signal_2_fft = fftshift(fft(signal_2));
signal_3_fft = fftshift(fft(signal_3));

phase_signal_1 = unwrap(angle(signal_1_fft));
phase_signal_2 = unwrap(angle(signal_2_fft));
phase_signal_3 = unwrap(angle(signal_3_fft));

%% Carrie in frequencyuency Domin
Carrier_frequencyuency_1 = 50000;
Carrier_frequencyuency_2 = 2*50000;

WC_1 = 2*pi * Carrier_frequencyuency_1;
WC_2 = 2*pi * Carrier_frequencyuency_2;

%% Carrier in time Domain
carrier_signal_1 = cos(WC_1 * t);
carrier_signal_2 = cos(WC_2 * t);
carrier_signal_3 = sin(WC_2 * t);

phase_carrier_signal_1 = unwrap(angle(fftshift(fft(carrier_signal_1))));
phase_carrier_signal_2 = unwrap(angle(fftshift(fft(carrier_signal_2))));
phase_carrier_signal_3 = unwrap(angle(fftshift(fft(carrier_signal_3))));
```

```
%% Modulation
modulatedSignal_t1 = signal_1' .* carrier_signal_1;
modulatedSignal_t2 = signal_2' .* carrier_signal_2;
modulatedSignal_t3 = signal_3' .* carrier_signal_3;

modulated_signal_f1 = fftshift(fft(modulatedSignal_t1));
modulated_signal_f2 = fftshift(fft(modulatedSignal_t2));
modulated_signal_f3 = fftshift(fft(modulatedSignal_t3));

phase_modulation_1 = unwrap(angle(modulated_signal_f1));
phase_modulation_2 = unwrap(angle(modulated_signal_f2));
phase_modulation_3 = unwrap(angle(modulated_signal_f3));

%% Add modulated signals
Modulated_Signal = modulatedSignal_t1 + modulatedSignal_t2 + modulatedSignal_t3;

Modulated_Signal_f = fftshift(fft(Modulated_Signal));
phase_mod = unwrap(angle(Modulated_Signal_f));

frequncy_BandPass =24000;
frequncy_BandPass2 =24000;
%% Synchronous Modulation for 3 signals
FileNames=["Out_1" "Out_2" "Out_3"];
Carries=[carrier_signal_1 ;carrier_signal_2; carrier_signal_3];
for i = 1:3
   if i == 1
      LowPass=demodulation(Modulated_Signal,frequncy_BandPass,FileNames(i), Carries(i,:,:),
frequency_sampling,i+6,t);
   else
      LowPass=demodulation(Modulated_Signal,frequncy_BandPass2,FileNames(i), Carries(i,:,:),
frequency_sampling,i+6,t);
   end
end
```

```
%%
% Phase Shift: 10
Carries_phase_10 = CarriersPhase(Carrier_frequencyuency_1, Carrier_frequencyuency_2, 10,t);
FileNames_phases_10=["Out_1_phase_10" "Out_2_phase_10" "Out_3_phase_10"];
for i = 1:3
   demodulation(Modulated_Signal,frequncy_BandPass,FileNames_phases_10(i), Carries_phase_10(i,:,:), frequency_sampling,i+9,t);
end

% Phase Shift: 30
Carries_phase_30 = CarriersPhase(Carrier_frequencyuency_1, Carrier_frequencyuency_2, 30,t);
FileNames_phases_30=["Out_1_phase_30" "Out_2_phase_30" "Out_3_phase_30"];
for i = 1:3
   demodulation(Modulated_Signal,frequncy_BandPass,FileNames_phases_30(i), Carries_phase_30(i,:,:), frequency_sampling,i+13,t);
end

% Phase Shift: 90
Carries_phase_90 = CarriersPhase(Carrier_frequencyuency_1, Carrier_frequencyuency_2, 90,t);
FileNames_phases_90=["Out_1_phase_90" "Out_2_phase_90" "Out_3_phase_90"];
for i = 1:3
    if i == 1
      demodulation(Modulated_Signal,frequncy_BandPass,FileNames_phases_90(i), Carries_phase_90(i,:,:), frequency_sampling,i+17,t);
    else
      demodulation(Modulated_Signal,frequncy_BandPass2,FileNames_phases_90(i), Carries_phase_90(i,:,:), frequency_sampling,i+17,t);
    end
end

%%
% local carrier frequencyuency different than Fc by 2 Hz in Demodulation
Carries_shift_2 = CarriersDifferentFc(Carrier_frequencyuency_1, Carrier_frequencyuency_2, 2,t);
FileNames_shift_2=["Out_1_shift_2" "Out_2_shift_2" "Out_3_shift_2"];
```

```
for i = 1:3
   demodulation(Modulated_Signal,frequncy_BandPass,FileNames_shift_2(i), Carries_shift_2(i,:,:), frequency_sampling,i+21,t);
end

%  local carrier frequencyuency different than Fc by 10 Hz in Demodulation

Carries_shift_10 = CarriersDifferentFc(Carrier_frequencyuency_1, Carrier_frequencyuency_2, 10,t);
FileNames_shift_10=["Out_1_shift_10" "Out_2_shift_10" "Out_3_shift_10"];
for i = 1:3
   demodulation(Modulated_Signal,frequncy_BandPass,FileNames_shift_10(i), Carries_shift_10(i,:,:), frequency_sampling,i+25,t);
end

% Ploting  Signals
plot_signal(t, frequency.', phase_signal_1,signal_1, abs(signal_1_fft),1, 'Signal 1');
plot_signal(t, frequency.', phase_signal_2,signal_2, abs(signal_2_fft),2, 'Signal 2');
plot_signal(t, frequency.', phase_signal_3,signal_3, abs(signal_3_fft),3, 'Signal 3');

% Ploting Modulated Signals
plot_signal(t, frequency.',phase_modulation_1,modulatedSignal_t1, abs(modulated_signal_f1),4, 'Modulated Signal 1');
plot_signal(t, frequency.', phase_modulation_2,modulatedSignal_t2, abs(modulated_signal_f2),5, 'Modulated Signal 2');
plot_signal(t, frequency.', phase_modulation_3,modulatedSignal_t3, abs(modulated_signal_f3),6, 'Modulated Signal 3');

function plot_signal(length_t, f,angle, time, freq,counter, Time)
   figure(counter)
   subplot(3, 1, 1)
   plot(length_t, time)
   xlabel('time')
   %title of time Domain
   title(strcat(Time, ' signal in time Domin'))
   subplot(3, 1, 2)
```

```
    plot(f, freq)
    xlabel('f')
    %title of frequency Domain
    title(strcat(Time, ' Signal in frequency Domain'))
    subplot(3, 1, 3)
    plot(f, angle)
    ylabel('phase')
    xlabel('f')
    %title of phase
    title(strcat(Time, ' Phase'))
end

function lpf=demodulation(signal_modulation, frequncy_BandPass, out_name, carrier_signal,
frequency_signal_sampling,figure,t)
    demodulationdSignal =  2* (signal_modulation .* carrier_signal);
    lpf = lowpass(demodulationdSignal, frequncy_BandPass, frequency_signal_sampling);
    fftSignal = fftshift(fft(lpf));
    N=(length(fftSignal)/2);
    frequency = (-1*N:N-1);
    phase=unwrap(angle(fftSignal));
    plot_signal(t, frequency.',phase,lpf, abs(fftSignal),figure, out_name);
    audiowrite(strcat("Output_signals/", out_name, '.wav'), lpf, frequency_signal_sampling);
end

function Array_of_carries = CarriersPhase(Carrier_frequencyuency_1, Carrier_frequencyuency_2, deg,time)

     carry_1 = cos((2*pi * Carrier_frequencyuency_1 * time) + ((deg * pi) / 180));
     carry_2 = cos((2*pi * Carrier_frequencyuency_2 * time) + ((deg * pi) / 180));
     carry_3 = sin((2*pi * Carrier_frequencyuency_2 * time) + ((deg * pi) / 180));
     Array_of_carries=[carry_1; carry_2; carry_3];
end
function Array_of_carries  = CarriersDifferentFc(Carrier_frequencyuency_1, Carrier_frequencyuency_2,
frequency_dc,t)
```

```
    carry_1 = cos(2*pi * (Carrier_frequencyuency_1 + frequency_dc) * t);
    carry_2 = cos(2*pi * (Carrier_frequencyuency_2 + frequency_dc) * t);
    carry_3 = sin(2*pi * (Carrier_frequencyuency_2 + frequency_dc) * t);
    Array_of_carries=[carry_1; carry_2; carry_3];
end
```