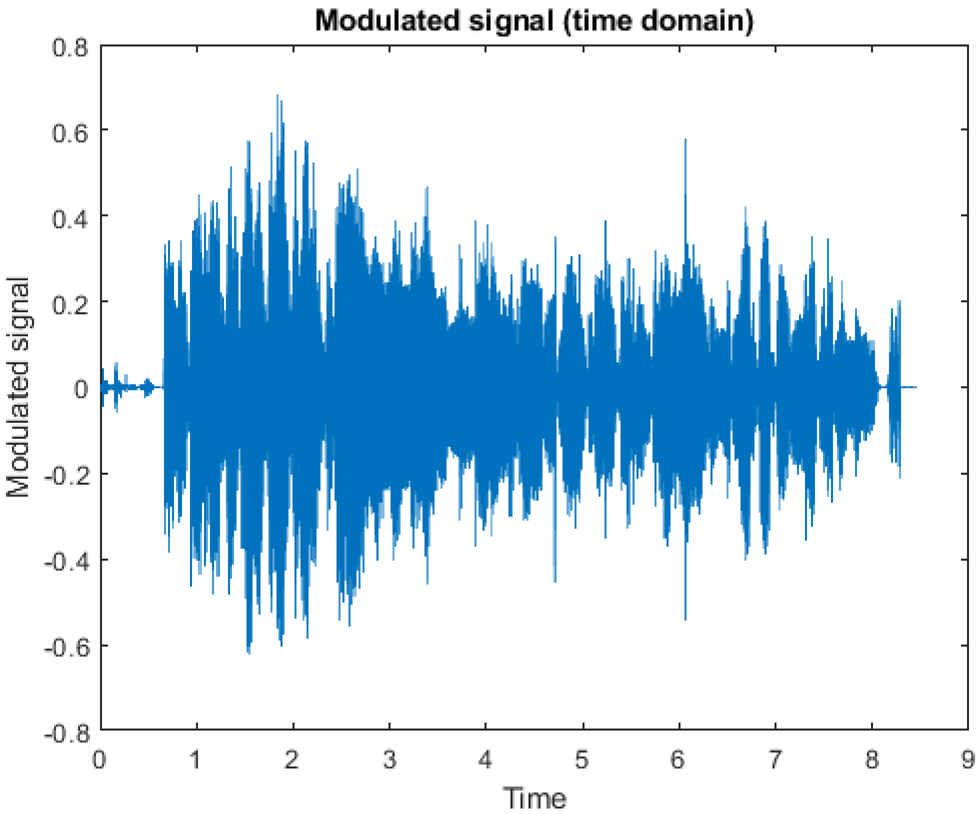
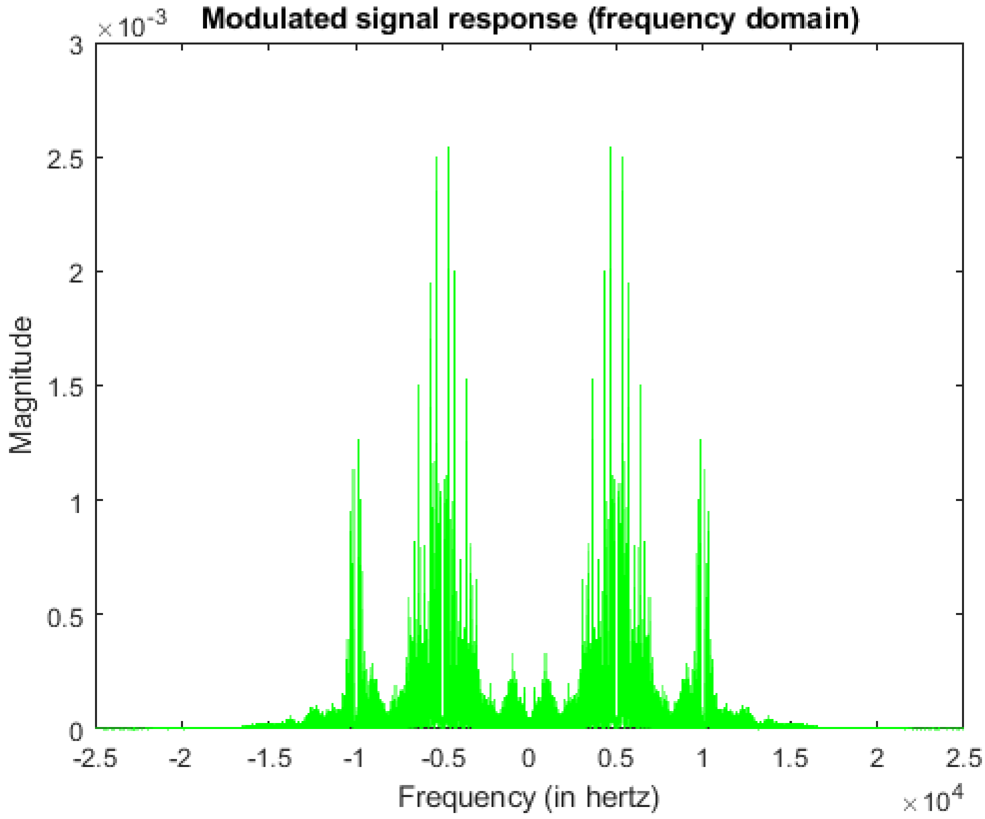
Team 29

Team members info

Name	Sec	BN
Abdallah Ahmed Hemdan	2	1
Adel Mohamed Abd-Elhamid	1	31

Modulated signal





Interference between modulated signals

- The Interference was between the low amplitude frequencies which has a very low impact into the final demodulated signals.
- The interference between the high amplitude frequencies was very low and the high amplitude frequencies have the most sound properties.
- So our final results didn't affected by the interference.

Demodulated signals in (2) - (3)

- $$e(t) = m(t)\cos(\omega_c t)\cos[(\omega_c + \Delta\omega)t + \varphi]$$
  $$= \frac{1}{2} m(t)\{\cos(\Delta\omega\, t + \varphi) + \cancel{\cos[(2\omega_c + \Delta\omega)t + \varphi]}\}$$

- The demodulated signals in (2) was as the original signal because of no shifting in phase (phi = 0)

- In (3), shifting by 10, 30 degrees attenuates the demodulated signals (speech signals) making them lower than the original ones.

- Shifting by 90 degree strongly attenuates the speech signals and it seems like no output speech.

Code

```
clear;clc;

[x1, Fs1] = audioread('Team 29_ speech signal_1.mp3');
[x2, Fs2] = audioread('Team 29_ speech signal_2.mp3');
[x3, Fs3] = audioread('Team 29_ speech signal_3.mp3');

Fc1 = 1 * 1e4;
Fc2 = 1 * 5e3;

Fs = max(Fc1, Fc2) * 5;

% Step (1): Modulation %

[y1, t1] = modAndTime(x1, Fs, Fc1, 1);
[y2, t2] = modAndTime(x2, Fs, Fc2, 1);
[y3, t3] = modAndTime(x3, Fs, Fc2, 0);

y2(numel(y1)) = 0;
y3(numel(y1)) = 0;

% Get the modulated signal and plot it %

s = y1 + y2 + y3;

dt = 1/Fs;
t = (0:dt:(length(s)*dt)-dt)';

plot(t, s)
xlabel('Time')
ylabel('Modulated signal')
title('Modulated signal (time domain)')

plotSignal(s, t, Fs, 'g')

ylabel('Magnitude')
xlabel('Frequency (in hertz)');
title('Modulated signal response (frequency domain)');

% Step (2): Demodulation %

[num1,den1] = butter(10, Fc1*2/Fs);
[num2,den2] = butter(10, Fc2*2/Fs);

[z1, z2, z3] = demodulateSignals(y1, y2, y3, Fc1, Fc2, Fs, num1, den1, num2,
den2, 0);

% Step (3): Demodulation with phase shifting %

[z1_10, z2_10, z3_10] = demodulateSignals(y1, y2, y2, Fc1, Fc2, Fs, num1,
den1, num2, den2, 10*pi/180);
[z1_30, z2_30, z3_30] = demodulateSignals(y1, y2, y2, Fc1, Fc2, Fs, num1,
den1, num2, den2, 30*pi/180);
```

```
    [z1_90, z2_90, z3_90] = demodulateSignals(y1, y2, y2, Fc1, Fc2, Fs, num1,
den1, num2, den2, 90*pi/180);

    % Writing demodulated audios %

    audiowrite('Team 29_1_out.mp4', z1, Fs1)
    audiowrite('Team 29_2_out.mp4', z2, Fs2)
    audiowrite('Team 29_3_out.mp4', z3, Fs3)

    % Functions %

    function [y, t] = modAndTime(x, Fs, Fc, cos_)
        dt = 1/Fs;
        t = (0:dt:(length(x)*dt)-dt)';

        w = 2 * pi * Fc;
        if cos_
            y = x(:, 1) .* cos(w * t);
        else
            y = x(:, 1) .* sin(w * t);
        end
    end

    function [] = plotSignal(x, t, Fs, color)
        % Frequency domain %
        N = size(t,1);

        %%Fourier Transform:
        X = fftshift(fft(x));

        %%Frequency specifications:
        dF = Fs / N;                        % hertz
        f = -Fs/2:dF:Fs/2-dF;               % hertz

        %%Plot the spectrum:
        figure
        plot(f,abs(X)/N, color);
    end

    function [z1, z2, z3] = demodulateSignals(y1, y2, y3, Fc1, Fc2, Fs, num1,
den1, num2, den2, phase)
        z1 = amdemod(y1, Fc1, Fs, phase, 0, num1, den1);
        z2 = amdemod(y2, Fc2, Fs, phase, 0, num2, den2);
        z3 = amdemod(y3, Fc2, Fs, phase+pi/2, 0, num2, den2);
    end
```