MANDATORY PART

Question 1

In this project, we learn how a simple analog communication system can be simulated in MATLAB. Consider the general block diagram of Fig. 1.

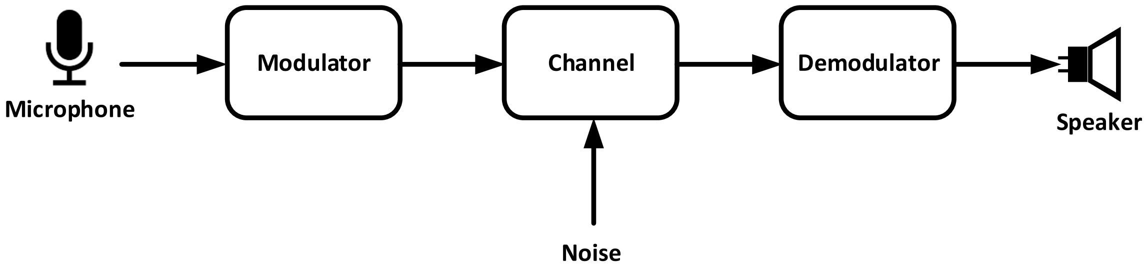


Figure 1: Block diagram of an analog communication system.

(a) Assume that the modulator and demodulator are FM. Write a MATLAB code to simulate the FM communication system. Create separate MATLAB functions for the modulator, demodulator, and channel. Then, connect them in a main mfile. Name the functions fm_modulator, channel, and fm_demodulator.

1.FM Modulator
matlab code:

```
function Y = fm_modulator(x,fc,Fs)

t = 0:1/Fs:(length(x)/Fs);
opt = (fc/Fs)*2*pi/(max(max(x)));
Y = cos(2*pi*fc*t(1:end-1) + opt*cumsum(x));

end
```

Question 1 continued on next page. . .

2.FM Demodulator
matlab code:

```
function y = fm_demodulator(y,beta)
[r, c] = size(y);
if r*c == 0
    y = [];
    return;
end;
if (r == 1)
    y = y(:);
    len = c;
else
    len = r;
end
Fc = 40000;
Fs = 8000*20;
pi2 = 2*pi;
[num, den] = butter(5, Fc * 2 / Fs);
sen = 2*pi*beta*100;
    %pre-process the filter.
    if abs(den(1)) < eps
        error('First denominator filter coefficient must be non-zero.');
    else
        num = num/den(1);
        if (length(den) > 1)
            den = - den(2:length(den)) / den(1);
        else
            den = 0;
        end;
        num = num(:)';
        den = den(:)';
    end;
```

Question 1 continued on next page. . .

```
len_den = length(den);
len_num = length(num);
x = y;
y = 2 * y;
ini_phase = pi/2;
    for ii = 1 : size(y, 2)
        zl = zeros(length(den), 1);
        sl = zeros(len_num, 1);
        intgl = 0;
        memo = 0;
        for i = 1:size(y, 1)
            %start with the zero-initial condition integer.
            vco_out = cos(pi2 * intgl+ini_phase);
            if len_num > 1
                sl = [y(i, ii) * vco_out; sl(1:len_num-1)];
            else
                sl = y(i, ii);
            end
            tmp = num * sl + den * zl;
            if len_den > 1
                zl = [tmp; zl(1:len_den-1)];
            else
                zl = tmp;
            end;
            intgl = rem(((tmp*sen + Fc)/ Fs + intgl), 1);
            x(i, ii) = tmp;
        end;
    end;
x = x;
x = decimate(x,20);
```

Question 1 continued on next page. . .

3.FM Channel
matlab code:

```
%[input1,Fs] = audioread('m2.wav');
%x = input1;
 x = load('input8000.mat');
 x = x.input1';
fc= 50 ;
plot(x),title('orginal sound')
```

fm

```
Fs = 8000;
fc= (Fs/2)-100 ;
% modulator
Y1 = fm_modulator(x,fc,Fs);
%Y1 = modulate(x,fc,Fs,'fm');
figure;subplot 121; plot(Y1),title('signal after modulator')
% awgn channel
SNR=100 ; %EsNo
y1=awgn(Y1,SNR);
%demodulator
x2 = demod(y1,fc,Fs,'fm');
%x2 = fm_demodulator(y1,fc,Fs);
subplot 122; plot(x2),title('signal after demodulator')
```

```
audiowrite('am_d2_snr100.wav',x2,Fs);
```

(b) Repeat the previous part for a conventional AM communication system. Note that the channel function does not change. You only need to code two new functions for the AM modulator and demodulator. Name these new functions am_modulator and am_demodulator.

1.AM Modulator
matlab code:

```
function Y = am_modulator(x,Fc,Fs)
t = 0:1/Fs:length(x)*1/Fs;
y = x.*cos(2*pi*Fc*t(1:end-1));
Y=y;
end
```

2.AM Demodulator
matlab code:

```
function Y = am_demodulator(y,fc,fs)
t = 0:1/fs:length(y)*1/fs;
x = y.*cos(2*pi*fc*t(1:end-1));
[b,a] = butter(5,fc*2/fs);
x = filtfilt(b,a,x);
Y=x;
end
```

3.AM Channel
matlab code:

```
[input1,Fs] = audioread('m2.wav');
x = input1;
% x = load('input8000.mat');
% x = x.input1';
fc= 50 ;
plot(x)
```

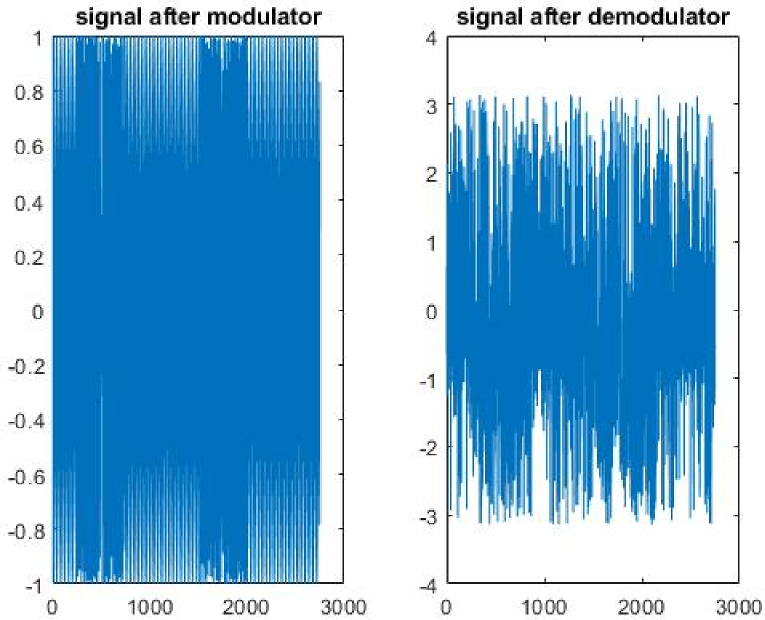
am

```
Fs = 8000;
fc= (Fs/2)-10 ;
% modulator
Y1 = am_modulator(x,fc,Fs);
figure;subplot 121; plot(Y1)
% awgn channel
SNR=100 ; %EsNo
y1=awgn(Y1,SNR);
%demodulator
x2 = am_demodulator(y1,fc,Fs);
subplot 122; plot(x2)
```

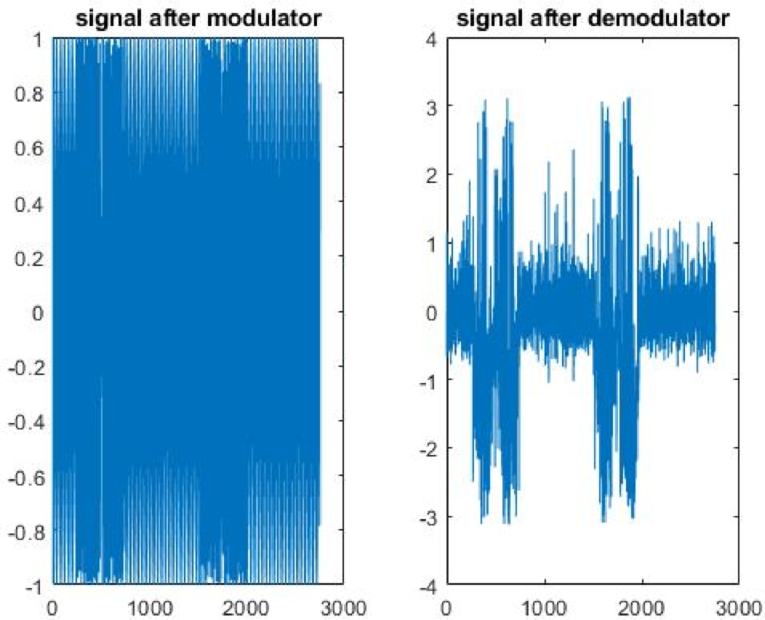
```
audiowrite('am_d2.wav',x2,Fs);
```

Question 1 continued on next page...

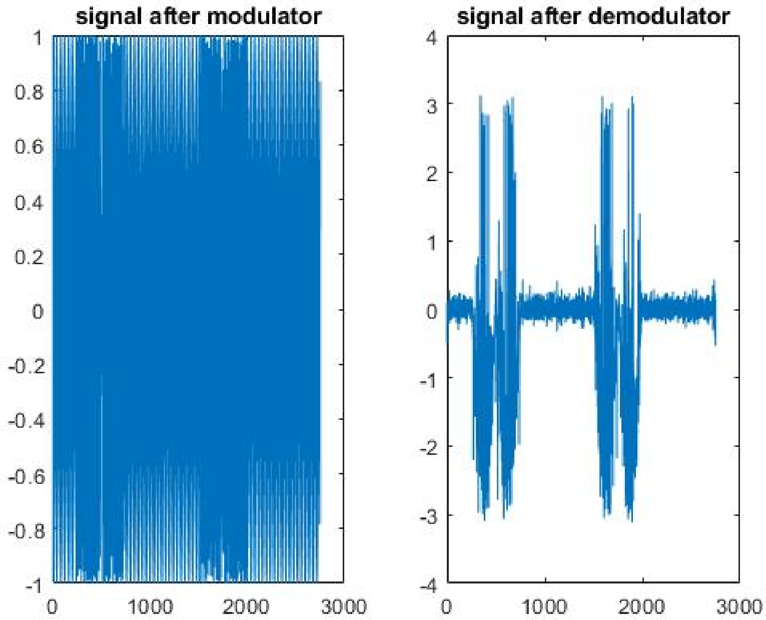
(c) Feed your simulation codes with a recorded audio file and play the demodulated signal and hear it for different noise levels in the channel. How do you feel when you hear the demodulated signal? Note that you can record your voice from your laptop microphone and feed it to the modulator. You can also play the demodulated signal and hear it from your laptop speaker. MATLAB has useful internal commands for working with microphones and speakers!



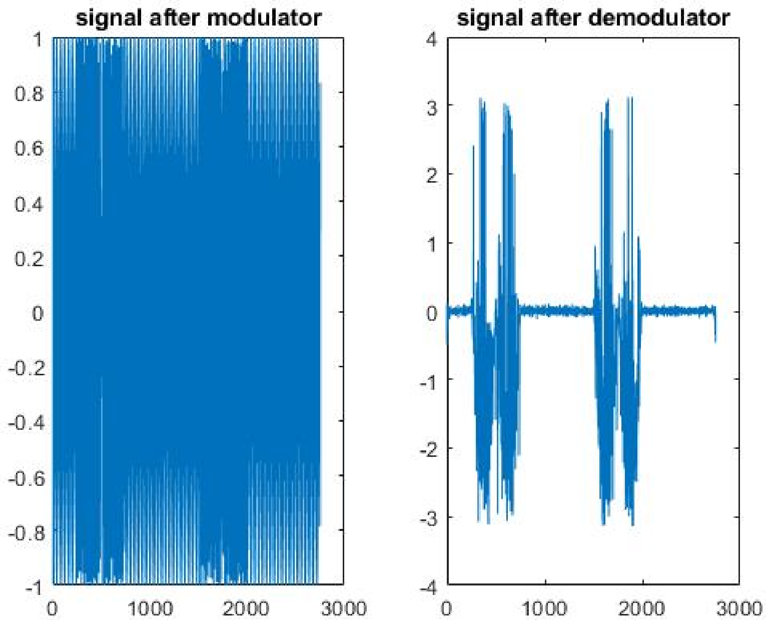
Fm SNR 2dB



Fm SNR 12dB

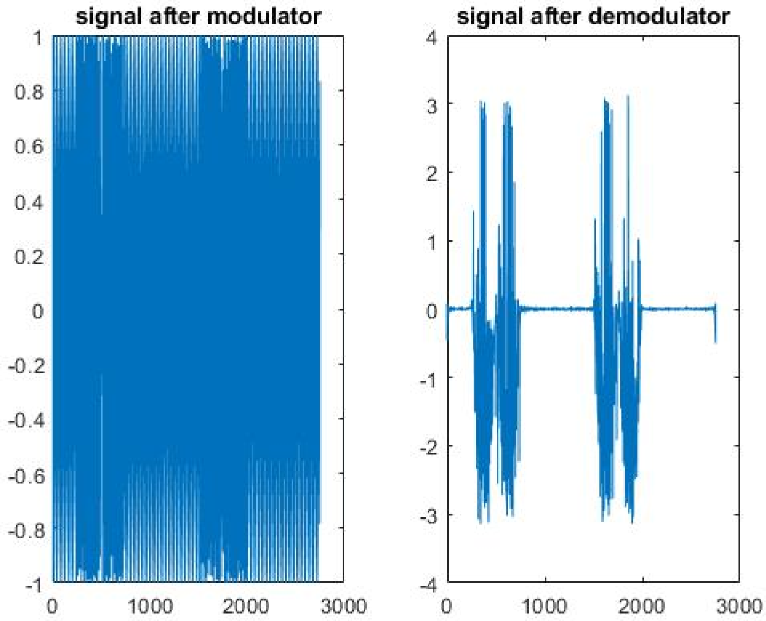


Fm SNR 22dB

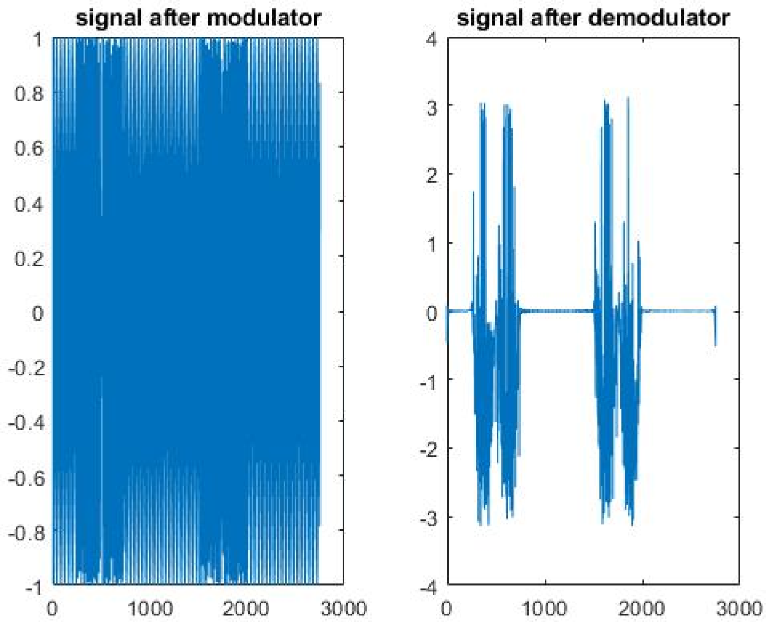


Fm SNR 32dB

Question 1 continued on next page. . .

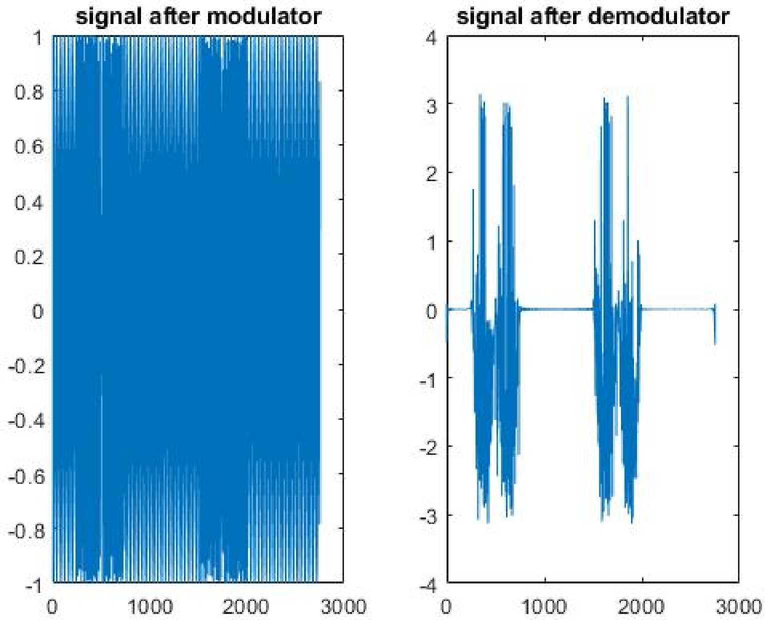


Fm SNR 42dB



Fm SNR 52dB

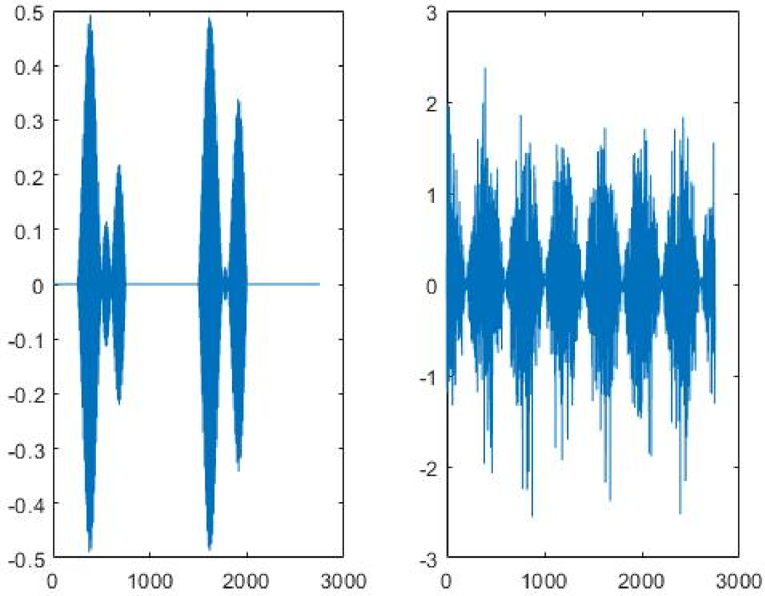
Question 1 continued on next page. . .



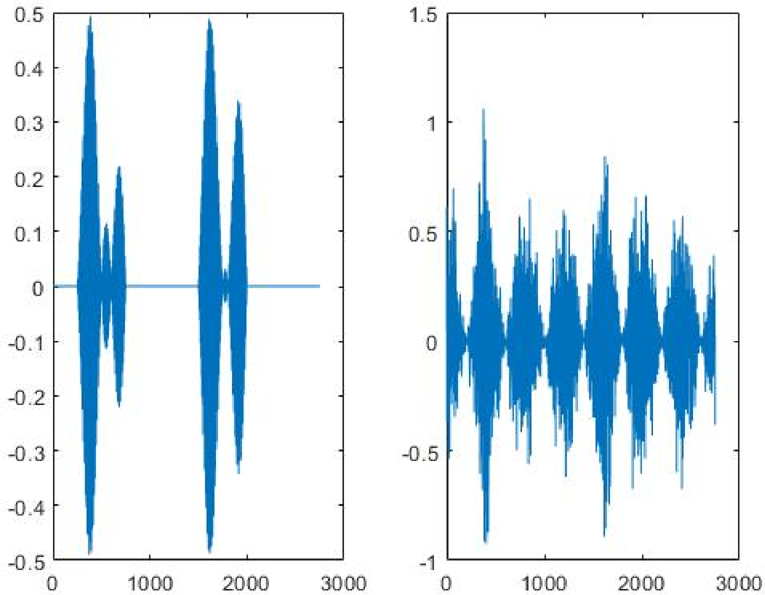
Fm SNR 100dB

The last mode is where the passing channel has a weak noise effect so we can understand the exact function of the modulation. The sound produced in this mode is very close to the original sound.

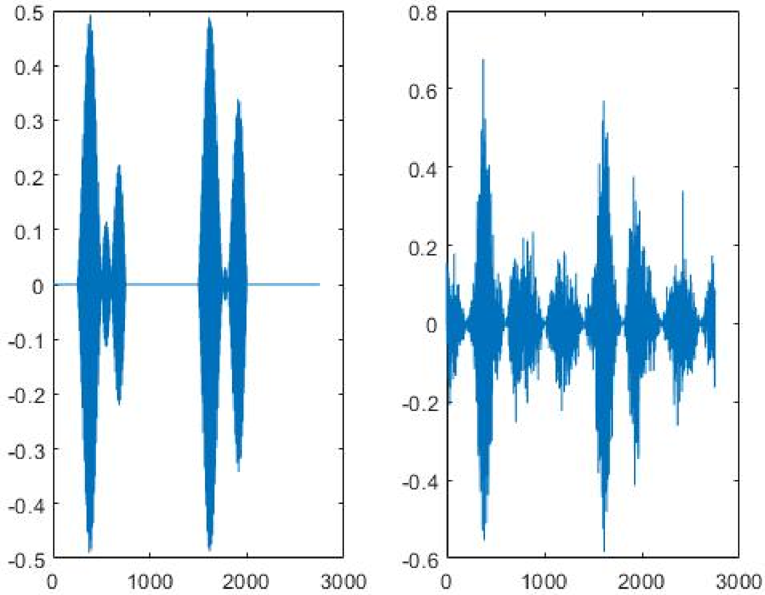
(All step's Audios is available)



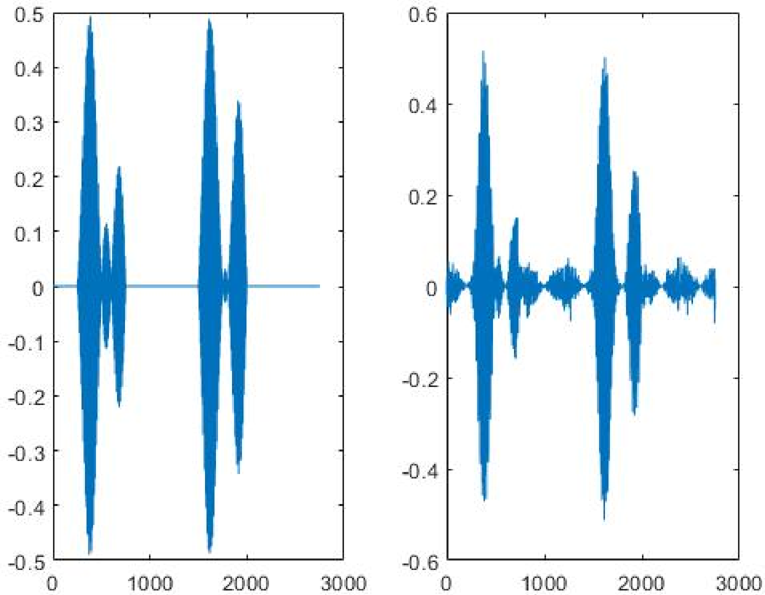
Am SNR 2dB



Am SNR 12dB

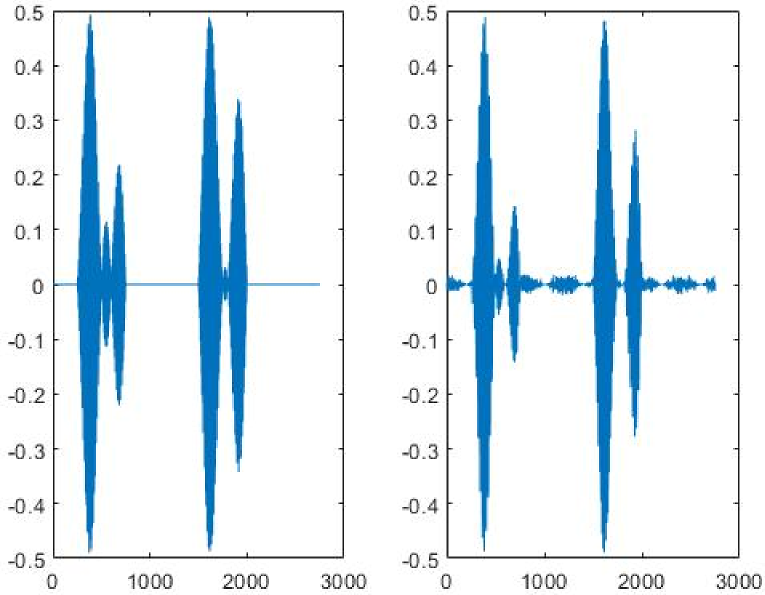


Am SNR 22dB

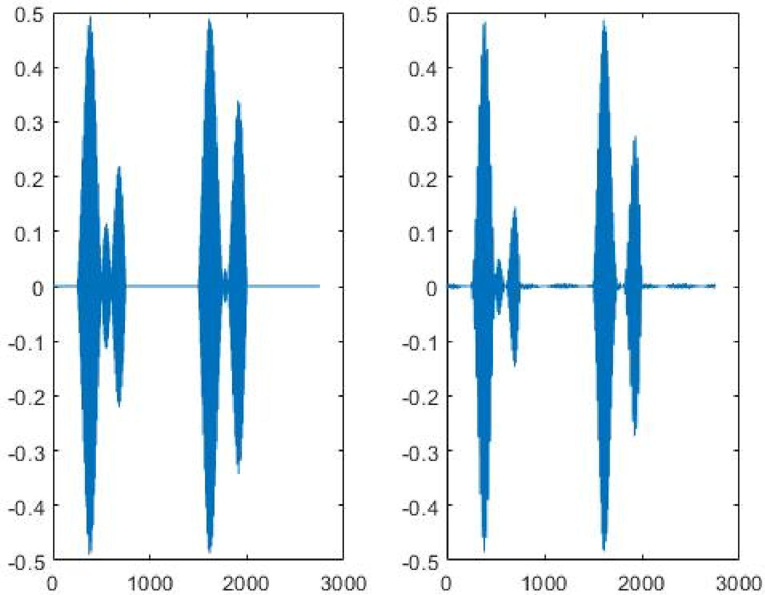


Am SNR 32dB

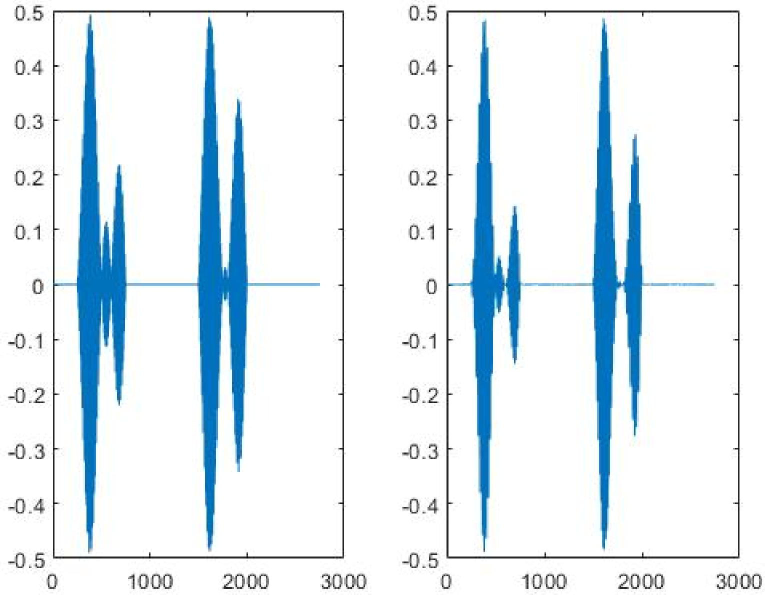
Question 1 continued on next page. . .



Am SNR 42dB



Am SNR 52dB

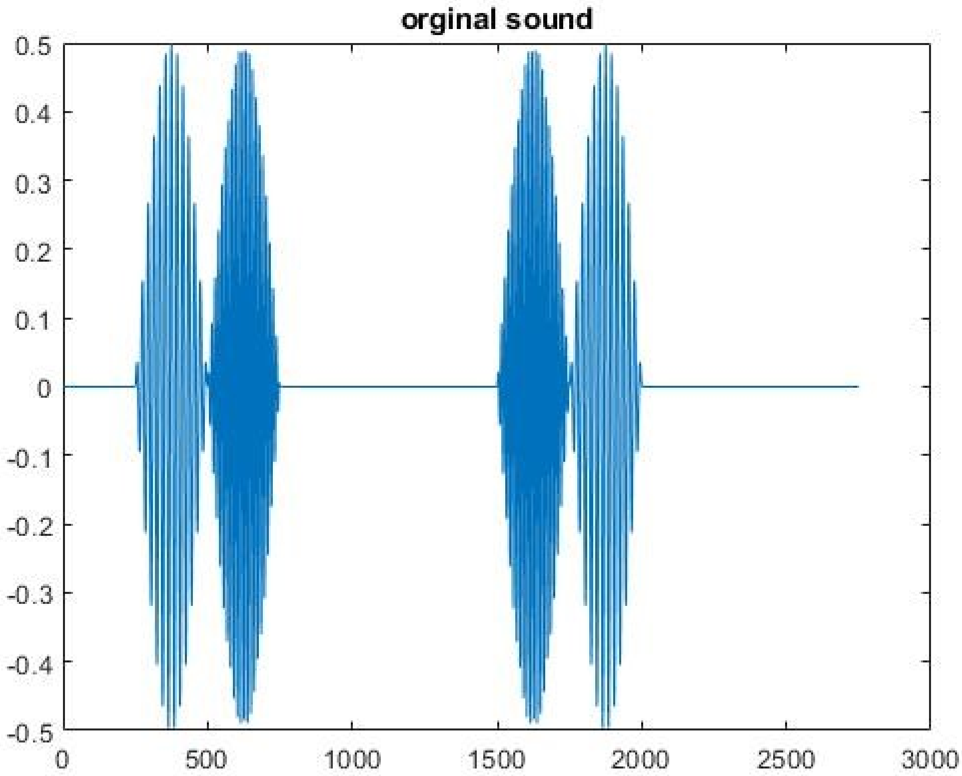


Am SNR 100dB

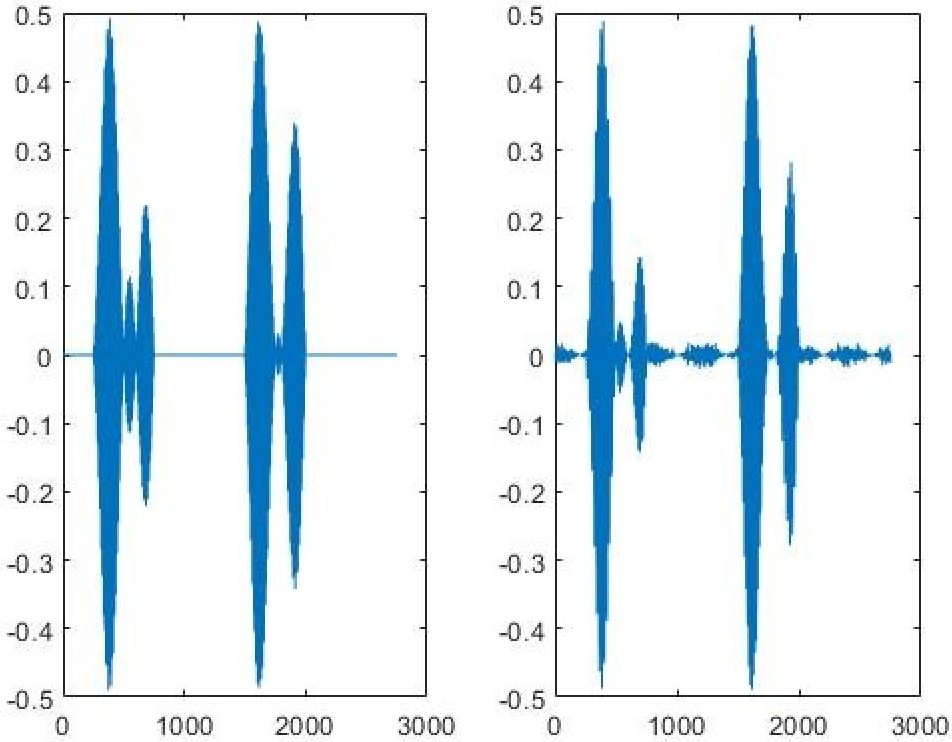
In this case, the sound we hear will be really close to the original sound.
(All step's Audios is available)

(d) Compare the SNR performance of the simulated FM and AM communication systems. To do this, you can plot the output SNR of both systems in terms of the SNR of the benchmark baseband system, as discussed in the course lecture.

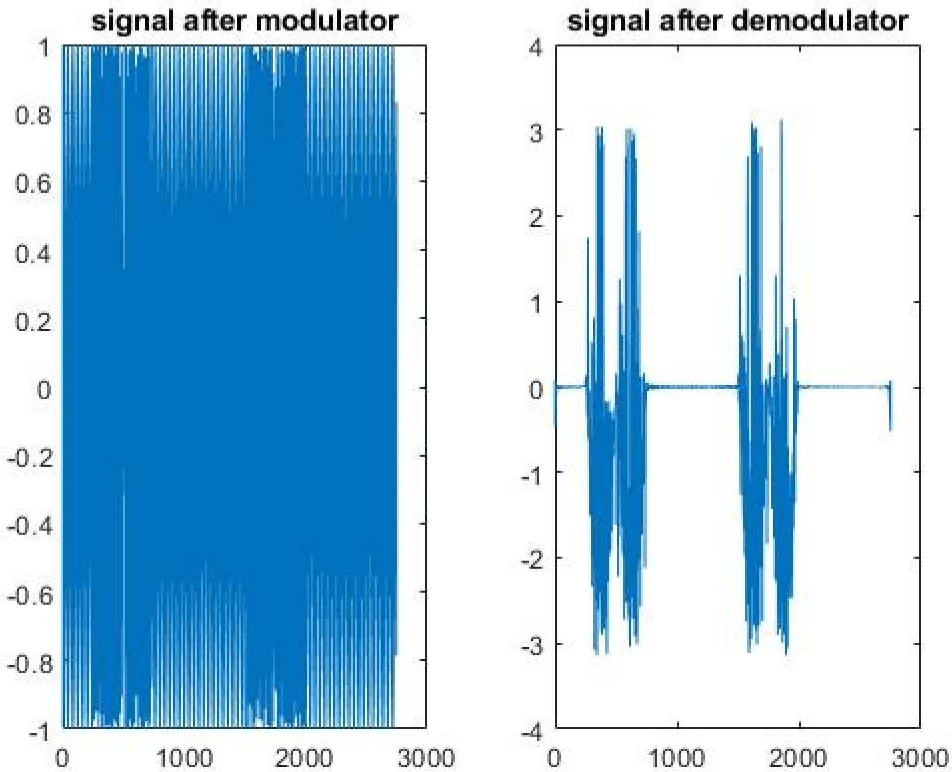
By comparing the plots in both AM and FM modes with the actual value of the waveforms, we find that signal which is modulated by FM modulator perform better in detecting noise-impregnated data because the main waveform has 4 parts with the same amplitude. They are detected in frequency modulation, but in amplitude modulation, the amplitude of the signals is strongly influenced by noise, so in AM demodulation, it could not display the equality of all 4 parts correctly.



Original Sound



AM SNR 42dB



FM SNR 42dB

Question 2 continued on next page. . .

Question 2

Prepare a short report and describe your work concisely. Use suitable figures to better describe the developed codes and make your report more readable and understandable. Attach a sample of the recorded audios as well as the developed codes to your sent report.

BONUS PART

Question 3

Can you make your simulation realtime? In this way, you talk to the microphone and hear the demodulated signal from the speaker simultaneously!

Record voice

```
Fs= 8000;
fc= (Fs/2)-100 ;
%recorder = audiorecorder(Fs,nBits,nChannels)
recObj  = audiorecorder(Fs,16,1);
get(recObj);
% Record your voice for 5 seconds.
recObj = audiorecorder;
disp('Start speaking.')
recordblocking(recObj, 5);
disp('End of Recording.');
% Play back the recording.
play(recObj);
% Store data in double-precision array.
myRecording = getaudiodata(recObj);
% Plot the waveform.
plot(myRecording);
x= myRecording';
%audiowrite('org.wav',x,Fs);
```

Question 3 continued on next page. . .

1.AM Modulator
matlab code:

```
function Y = am_modulator(x,Fc,Fs)
t = 0:1/Fs:length(x)*1/Fs;
y = x.*cos(2*pi*Fc*t(1:end-1));
Y=y;
end
```

2.AM Demodulator
matlab code:

```
function Y = am_demodulator(y,fc,fs)
t = 0:1/fs:length(y)*1/fs;
x = y.*cos(2*pi*fc*t(1:end-1));
[b,a] = butter(5,fc*2/fs);
x = filtfilt(b,a,x);
Y=x;
end
```

Question 3 continued on next page...

3.FM Modulator
matlab code:

```
function Y = fm_modulator(x,fc,Fs)

t = 0:1/Fs:(length(x)/Fs);
opt = (fc/Fs)*2*pi/(max(max(x)));
Y = cos(2*pi*fc*t(1:end-1) + opt*cumsum(x));

end
```

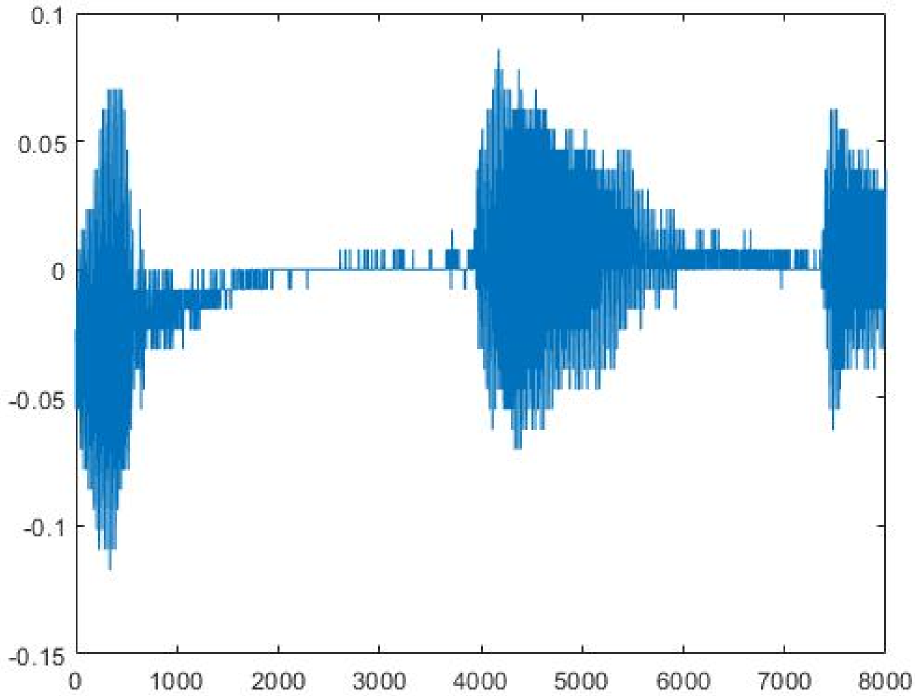
4.FM Demodulator
matlab code:

```
function y = fm_demodulator(y,beta)

[r, c] = size(y);

if r*c == 0
    y = [];
    return;
end;
if (r == 1)
    y = y(:);
    len = c;
else
    len = r;
end

Fc = 40000;
Fs = 8000*20;
pi2 = 2*pi;
[num, den] = butter(5, Fc * 2 / Fs);
sen = 2*pi*beta*100;

    %pre-process the filter.
    if abs(den(1)) < eps
        error('First denominator filter coefficient must be non-zero.');
    else
        num = num/den(1);
        if (length(den) > 1)
            den = - den(2:length(den)) / den(1);
        else
            den = 0;
        end;
        num = num(:)';
        den = den(:)';
    end;
```

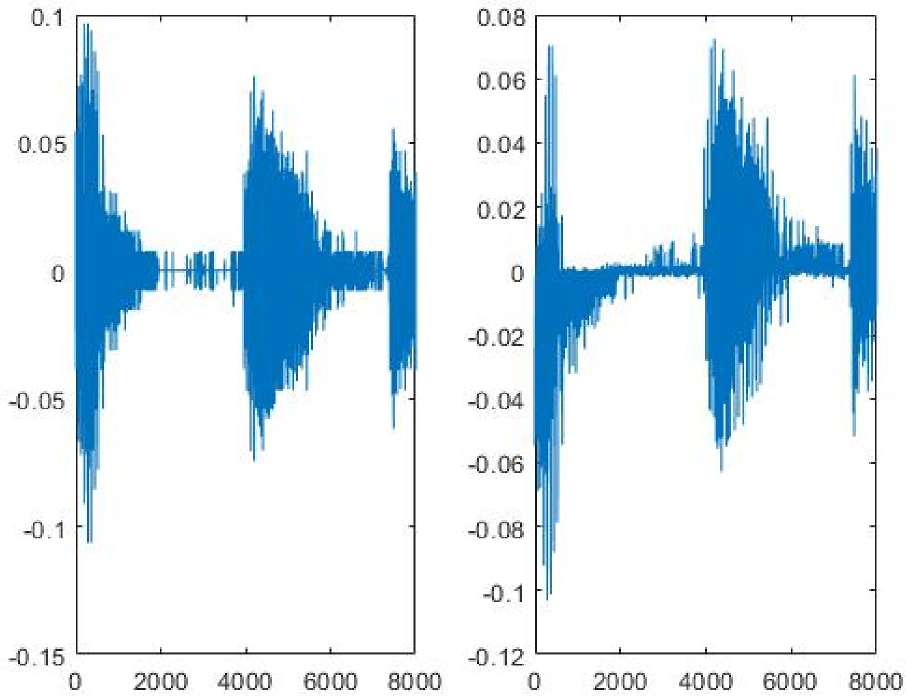
Question 3 continued on next page. . .

```
len_den = length(den);
len_num = length(num);
x = y;
y = 2 * y;
ini_phase = pi/2;
    for ii = 1 : size(y, 2)
        zl = zeros(length(den), 1);
        sl = zeros(len_num, 1);
        intgl = 0;
        memo = 0;
        for i = 1:size(y, 1)
            %start with the zero-initial condition integer.
            vco_out = cos(pi2 * intgl+ini_phase);
            if len_num > 1
                sl = [y(i, ii) * vco_out; sl(1:len_num-1)];
            else
                sl = y(i, ii);
            end
            tmp = num * sl + den * zl;
            if len_den > 1
                zl = [tmp; zl(1:len_den-1)];
            else
                zl = tmp;
            end;
            intgl = rem(((tmp*sen + Fc)/ Fs + intgl), 1);
            x(i, ii) = tmp;
        end;
    end;
x = x;
x = decimate(x,20);
```

Question 3 continued on next page. . .

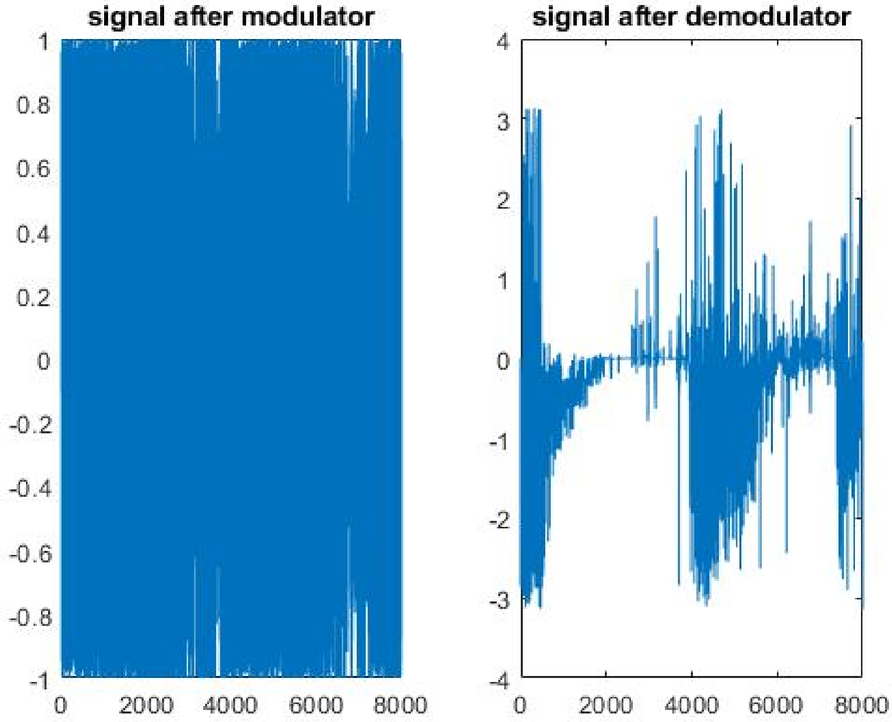


Original Real Time sound



AM Real Time

Question 3 continued on next page...



FM Real Time

// In real time modulation, we set that our audio signal would be received once every 5 seconds and that its modulation would be calculated and displayed. These 5 seconds could be more or less.

Question 4

Return your report by filling the LaTeXtemplate of the project.