MATLAB ASSIGNMENT #1

ECEN-649, FALL-2019

Name: Mrinmoy Sarkar

Banner ID: 950363260

Date: 11/08/2019

Contents

[Introduction 3](#_Toc24045733)

[Description of The System 3](#_Toc24045734)

[AM 3](#_Toc24045735)

[FM 3](#_Toc24045736)

[Design consideration for the experiment 4](#_Toc24045737)

[Result and Discussion 5](#_Toc24045738)

[Appendix 5](#_Toc24045739)

[MATLAB Code 5](#_Toc24045740)

[Plot of all signals and their spectrum 8](#_Toc24045741)

# Introduction

In this assignment, A voice signal is recorded using a laptop and MATLAB. The recorded voice signal is modulated using amplitude modulation technique and frequency modulation technique. Then the modulated signal is sent through a channel and demodulated to get the message signal. The demodulated signal is passed through a low pass filter to get the original voice signal. All these analyses are done using MATLAB programming language. The overall system is described in the following sections.

# Description of The System

First a message signal is recorded with 8KHz sampling frequency, 8-bit resolution and 5 second duration using a MAC BOOK AIR laptop and MATLAB software. The audio signal is saved in a .wav file format. The message in the audio signal is **“This is ECEN649. My name is Mrinmoy Sarkar. I am doing an assigned project.”** After this amplitude modulation (AM) and frequency modulation (FM) is applied to the recorded signal and transmitted virtually in the MATLAB environment and the transmitted signal is demodulated accordingly and passed through a low pass filter to recover the original audio signal and saved in the computer in .wav format. The AM and FM process are described in the following paragraph.

## AM

To generate the AM signal a modulation index of 0.7 is used and a carrier frequency is used as the half of the sampling frequency. A cosine signal is used to modulate the message signal. From the receiver side the modulated signal is passed through a diode means a low cost envelop detector and the filtered with a low pass filter with a cutoff frequency of 4000Hz. The AM modulation and demodulation are shown in the following block diagram.



Figure 1: AM modulator and demodulator block diagram.

## FM

To generate the FM signal, the message signal is integrated first, then multiplied with the 90-degree phase shifted carrier signal and the multiplied by the modulation index and then deducted from carrier signal. The modulated signal is transmitted through the channel and then the received signal is demodulated by differentiating the received signal and using an envelope detector. The detected signal is passed through a low pass filter. In this project a narrow band FM is generated with modulation index of 0.7. The FM modulation and demodulation are shown in the following block diagram.



Figure 2: FM modulator and demodulator block diagram.

# Design consideration for the experiment

To implement the AM and FM modulation and demodulation, several parameters needs to be tuned. The parameters are listed in the following table with justification.

|  |  |  |  |
| --- | --- | --- | --- |
| AM parameters | | | |
| Name | Symbol | Value | Justification |
| Modulation index | µ | 0.7 | µ should be between 0 and 1 for un distorted received signal. |
| Carrier frequency | fc | 4000 Hz | Since the voice signal is recorded using 8000Hz, fc cannot be more than half of 8000Hz or the signal will face aliasing by violating Nyquist theorem. |
| Cutoff frequency of LPF | fcutoff | 4000Hz | Cutoff frequency of the low pass filter is set to be 4000Hz because human voice has 0-3400hz frequency. |
| FM parameters | | | |
| Carrier frequency | fc | 4000 Hz | Since the voice signal is recorded using 8000Hz, fc cannot be more than half of 8000Hz or the signal will face aliasing by violating Nyquist theorem. |
| Modulation index | β | 0.07 | For narrow band FM β should be very less than 1. |
| Cutoff frequency of LPF | fcutoff | 4000Hz | Cutoff frequency of the low pass filter is set to be 4000Hz because human voice has 0-3400hz frequency. |

# Result and Discussion

The recorded signal is transmitted successfully using AM and FM modulation which is verified by MATLAB plot and by listening the received signal. Though the quality of the received signal was not as good as the original message signal, the message is clearly recognizable. The possible reason is the low sampling rate and low order of the low pass filter. There is always some quantization error in the digital signal processing. Moreover, AM and FM are analog signal but in the project it is done using digital domain.

# Appendix

## MATLAB Code

|  |
| --- |
| %% Name: Mrinmoy Sarkar  % email: msarkar@aggies.ncat.edu  clear all;  close all;      %% record audio and save it to a file  % Fs = 8000;  % nBits = 8;  % duration = 5;  % info = audiodevinfo;  % recorder = audiorecorder(Fs,nBits,1);  % disp('Start speaking.')  % recordblocking(recorder, duration);  % disp('End of Recording.');  % y = getaudiodata(recorder);  % audiowrite('assign1.wav',y,Fs);    %% read a audio file  [m\_sig,Fs] = audioread('assign1.wav');  m\_sig = m\_sig';  sound(m\_sig,Fs)  pause(10)    fc = Fs/2;  T = length(m\_sig)/Fs;  Ts = 1/Fs;  t = 0:Ts:T-Ts;    Lfft = length(t);  Lfft = 2^ceil(log2(Lfft));  M\_fre = fftshift(fft(m\_sig,Lfft));  freqm = (-Lfft/2:Lfft/2-1)/(Lfft\*Ts);    %% Low pass filter design  B\_m = 4000;  h = fir1(256, B\_m\*Ts);    %% AM modulation  mu = 0.7;  mp = max(abs(m\_sig));  A = mp/mu;  s\_am = (A+m\_sig).\*cos(2\*pi\*fc\*t);  Lfft = length(t);  Lfft = 2^ceil(log2(Lfft)+1);  S\_am = fftshift(fft(s\_am,Lfft));  freq\_am = (-Lfft/2:Lfft/2-1)/(Lfft\*Ts);    %% AM demodulation  s\_dem\_am = s\_am.\*(s\_am>0);  s\_rec\_am = filter(h,1,s\_dem\_am);  S\_rec\_am = fftshift(fft(s\_rec\_am,Lfft));    sound(s\_rec\_am,Fs)  pause(10)  audiowrite('assign1\_rec\_am.wav',s\_rec\_am,Fs);    %% FM modulation  beta = 0.07;  fDev = B\_m\*beta;  s\_fm = fmmod(m\_sig,fc,Fs,fDev);  Lfft = length(t);  Lfft = 2^ceil(log2(Lfft)+1);  S\_fm = fftshift(fft(s\_fm,Lfft));  freq\_fm = (-Lfft/2:Lfft/2-1)/(Lfft\*Ts);    %% FM demodulation  s\_fmdem = fmdemod(s\_fm,fc,Fs,fDev);  s\_rec\_fm = filter(h,1,s\_fmdem);  S\_rec\_fm = fftshift(fft(s\_rec\_fm,Lfft));    sound(s\_rec\_fm,Fs)  pause(10)  audiowrite('assign1\_rec\_fm.wav',s\_rec\_fm,Fs);    %% plot of message signal, transmitted and received am, fm signal and their spectrum  figure(1)  Frange = [-4000 4000 0 150];      subplot(5,2,1);  plot(t,m\_sig);  ylabel('m(t)','FontSize', 16)  xlabel('t(sec)','FontSize', 16)  title('Message signal','FontSize', 16)    subplot(5,2,2);  plot(freqm,abs(M\_fre));  axis(Frange)  ylabel('M(f)','FontSize', 16)  xlabel('f(Hz)','FontSize', 16)  title('Message spectrum','FontSize', 16)      subplot(5,2,3);  plot(t,s\_am);  ylabel('s\_{am}(t)','FontSize', 16)  xlabel('t(sec)','FontSize', 16)  title('AM transmitted signal','FontSize', 16)    subplot(5,2,4);  plot(freq\_am,abs(S\_am));  axis(Frange)  ylabel('S\_{am}(f)','FontSize', 16)  xlabel('f(Hz)','FontSize', 16)  title('AM transmitted spectrum','FontSize', 16)      subplot(5,2,5);  plot(t,s\_rec\_am);  ylabel('s\_{rec\\_am}(t)','FontSize', 16)  xlabel('t(sec)','FontSize', 16)  title('AM received signal','FontSize', 16)    subplot(5,2,6);  plot(freq\_am,abs(S\_rec\_am));  axis(Frange)  ylabel('S\_{rec\\_am}(f)','FontSize', 16)  xlabel('f(Hz)','FontSize', 16)  title('AM received spectrum','FontSize', 16)      subplot(5,2,7);  plot(t,s\_fm);  ylabel('s\_{fm}(t)','FontSize', 16)  xlabel('t(sec)','FontSize', 16)  title('FM transmitted signal','FontSize', 16)    subplot(5,2,8);  plot(freq\_fm,abs(S\_fm));  axis(Frange)  ylabel('S\_{fm}(f)','FontSize', 16)  xlabel('f(Hz)','FontSize', 16)  title('FM transmitted spectrum','FontSize', 16)      subplot(5,2,9);  plot(t,s\_rec\_fm);  ylabel('s\_{rec\\_fm}(t)','FontSize', 16)  xlabel('t(sec)','FontSize', 16)  title('FM received signal','FontSize', 16)    subplot(5,2,10);  plot(freq\_fm,abs(S\_rec\_fm));  axis(Frange)  ylabel('S\_{rec\\_fm}(f)','FontSize', 16)  xlabel('f(Hz)','FontSize', 16)  title('FM received spectrum','FontSize', 16) |

## Plot of all signals and their spectrum

