*Alexandria University,*

*Faculty of Engineering.*

*Electronics and Communication Department*

# Analog Communication

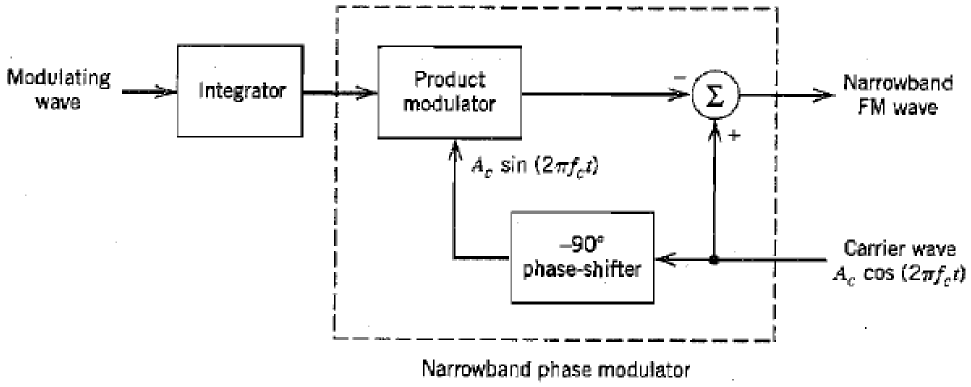
## Frequency Modulation Assignment

| *Name* | *ID* | *Section* |
|---|---|---|
| **Omar Mohamed Ahmed Arslan** | **143** | **4** |
| **Mohamed Said Mohamed Abdel Moneim** | **179** | **5** |

## Table of Contents

# INTRODUCTION

Frequency modulation (FM) is a modulation type in which the instantaneous frequency of the carrier is changed according to the message amplitude. The motive behind the frequency modulation was to develop a scheme with inherent ability to combat noise. The noise, being usually modeled as additive, has a negative effect on the amplitude by introducing unavoidable random variations which are superimposed on the desired signal. Unlike the amplitude, frequency has a latent immunity against noise. Since it resides "away" from the amplitude, any changes in the amplitude would be completely irrelevant to the frequency. In other words, there is no direct correlation between the variation in amplitude and frequency, thus making FM a better candidate over AM with respect to noise immunity. However, what FM gains in noise immunity lacks in bandwidth efficiency. Since FM usually occupies larger bandwidth, AM is considered more bandwidth wise.




# AIM

In this experiment, we investigate the narrowband frequency modulation. Students are expected to:

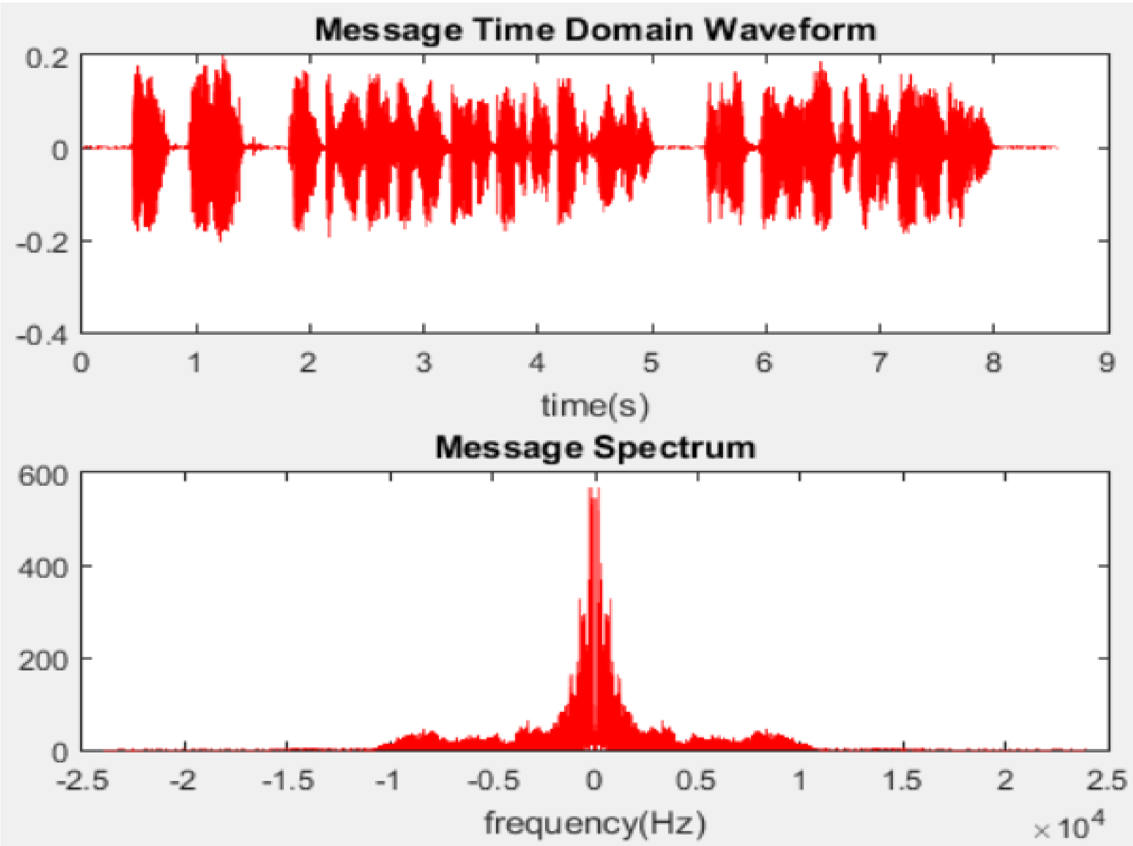
1. Develop an appreciation of FM ability to counteract noise.
2. Be able to simulate the generation and the demodulation of NBFM using MATLAB.
3. To be able to tell the similarities and differences between AM and NBFM.

# Part (1)

Use Matlab to read the attached audio file, which has a sampling frequency Fs=48 KHz. Find the spectrum of this signal (the signal in frequency domain). [*audioread, fft , fftshift , plot*]

## *M.code*

```
%% Reading Audio
% read the audio file
[message , FS] = audioread ('eric.wav') ;

%sound time domain plot
timeAxis = linspace(0, length(message)/FS , length(message)) ;
figure
subplot(2,1,1)
plot (timeAxis , message,'r') ; xlabel('time(s)') ; title('Message Time Domain Waveform');
%message spectrum plot
Fdom1 = linspace(-FS/2 , FS/2 , length(message));
message_freq_domain = fftshift(fft(message)) ;
message_mag = abs(message_freq_domain) ;
subplot(2,1,2);
plot (Fdom1, message_mag,'r') ; xlabel('frequency(Hz)') ; title('Message Spectrum');
```

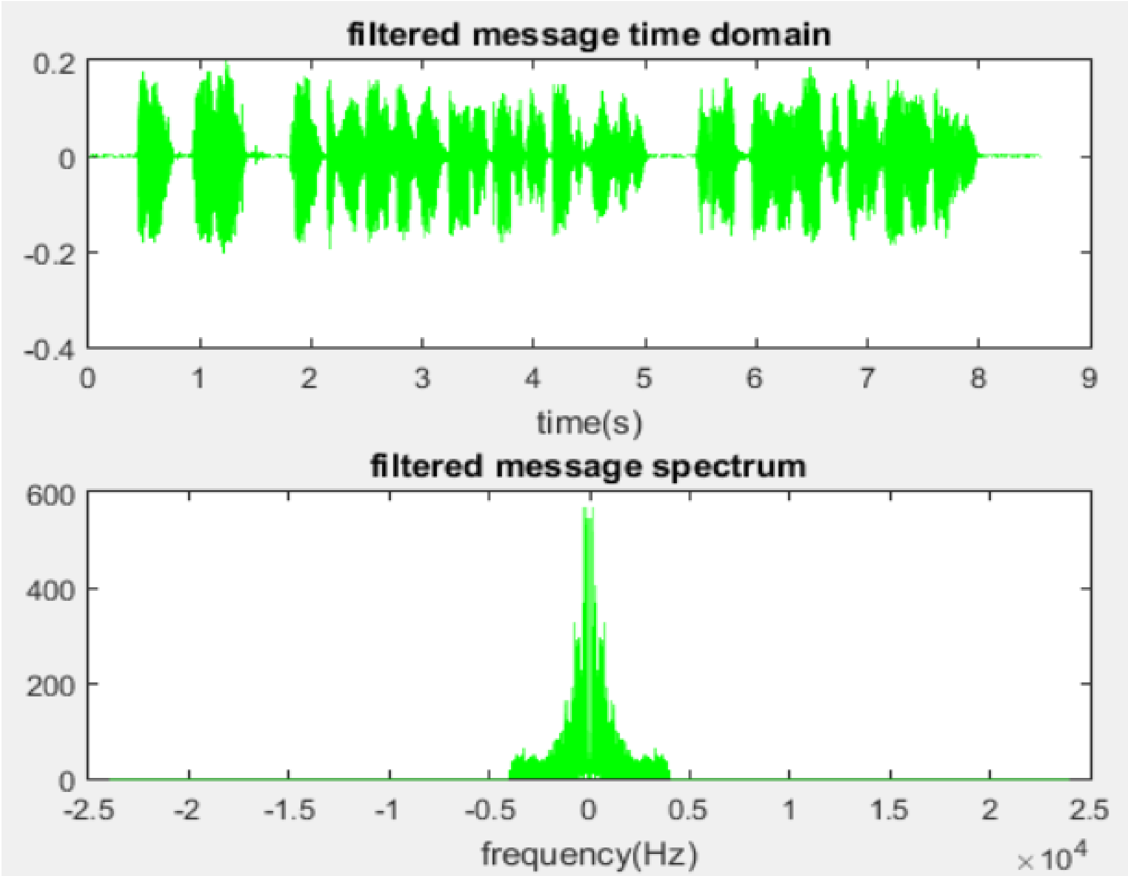


## Part (2)

- Using an ideal Filter, remove all frequencies greater than 4 KHz.
- Obtain the filtered signal in time domain, this is a band limited signal of BW=4KHz. *[ifftshift ,ifft]*
- sound the filtered audio signal (make sure that there is only a small error in the filtered signal)

## *M.code*

```
%% Filtering
n = FS/(length(message)-1) ;
% first interval
freq1 = -FS/2 : n : -4000 ;
mag1 = zeros(1, length(freq1)) ;
% second interval
freq2 = -4000: n : 4000 ;
mag2 = ones(1, length(freq2)) ;
% third interval
freq3 = 4000 : n : FS/2 ;
mag3 = zeros(1, length(freq3));
% filtering
filter = [mag1 mag2 mag3];
filtered_message_F = filter' .* message_mag ;

% converting from frequency domain to time domain
size(filtered_message_F); size(message) ;
filtered_message_T = real(ifft(ifftshift(message_freq_domain))) ;

% PLOTTING
figure
subplot(2,1,2);
plot(Fdom1 , filtered_message_F,'g') ; xlabel('frequency(Hz)') ; title('filtered message spectrum');
subplot(2,1,1);
plot(timeAxis , filtered_message_T,'g') ; xlabel('time(s)')  ; title('filtered message time domain') ;

%play sound
sound(filtered_message_T , FS) ;
```

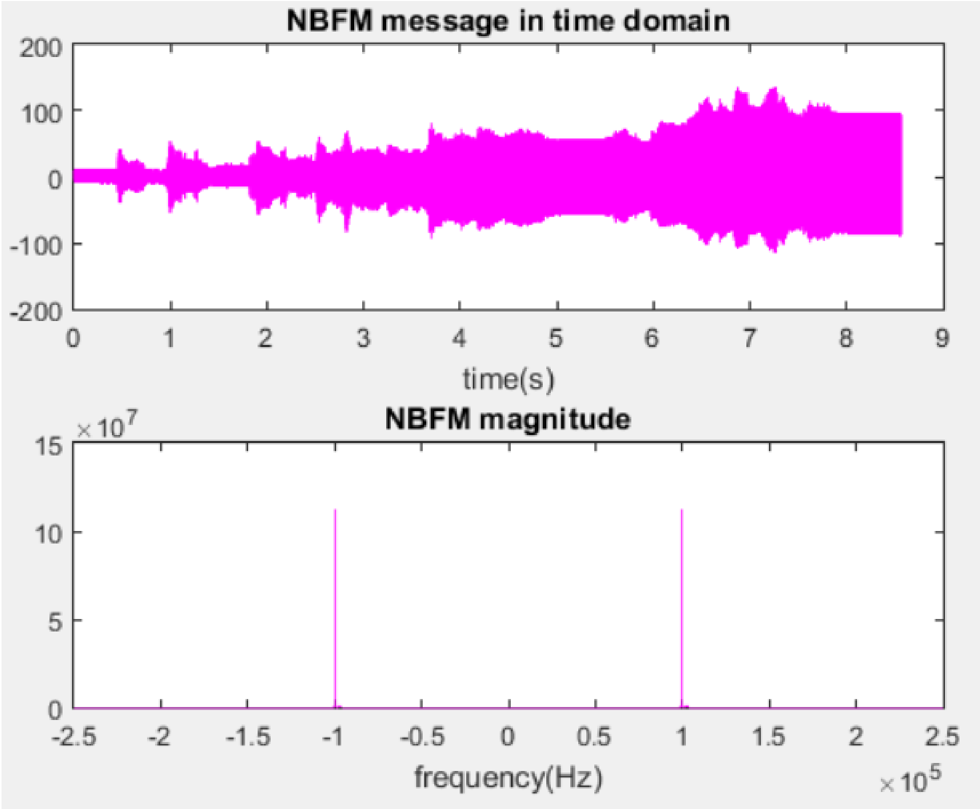

filtered message time domain
0.2
0
-0.2
-0.4
0
1
2
3
4
5
6
7
8
9
time(s)
filtered message spectrum
600
400
200
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
frequency(Hz)
$\times 10^4$

## Part (3)

Generate the NBFM signal. Use a carrier frequency of 100kHz and a sampling frequency of $Fs = 5Fc$. Plot the resulting spectrum. What can you make out of the resulting plot?

***M.code***

```
%% NBFM Modulation
% Modulation equation
%   s(t) = A cos(2 pi Fc t) - A Kf sin(2 pi Fc t) integration(message)

Fc = 100000;            % Carrier Frequency
FS2 = 5 * Fc;           % Sampling Frequency
Kf = 2 * pi * 0.01;     % Frequency Sensitivity
A = 10;                 % Carrier Amplitude

% resampling the message
message_resampled = resample(filtered_message_T , FS2 , FS);

% generating linspaces(defining time and freq intervals)
t = linspace (0 ,  (length(message_resampled)) / FS2 , length(message_resampled)) ;
Fdom2 = linspace( (-FS2)/2 , FS2/2 , length(message_resampled) ) ;

% generating the signal
NBFM_signal = A * cos(2*pi*Fc* t)' - A * Kf * sin(2*pi*Fc* t)' .*
cumsum(message_resampled);
%converting from time to freq. domain
NBFM_signal_spectrum = fftshift ( fft (NBFM_signal)) ;

% PLOTTING
figure
subplot(2,1,1);
plot ( t,NBFM_signal,'m') ; xlabel('time(s)') ;  title('NBFM message in time domain');
subplot(2,1,2);
plot(Fdom2, abs(NBFM_signal_spectrum),'m') ; xlabel('frequency(Hz)') ; title('NBFM
magnitude');
```




Q: What can you make out of the resulting plot?

A: We notice that in the time-domain, the amplitude is increasing because of the integration part in the modulation equation. In the frequency domain, we notice that the spectrum is mostly all about the carrier frequency due to the small frequency variations.

Q: What is the condition we needed to achieve NBFM.

A: the condition is

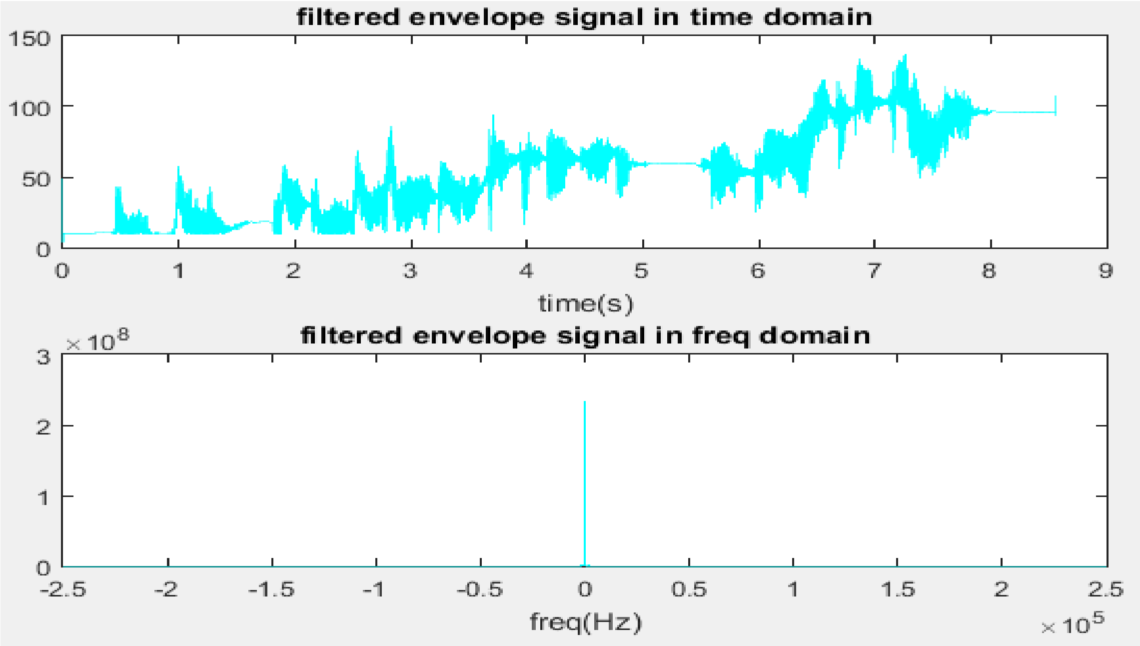
- from relation of frequency deviation ratio $\boldsymbol{\beta = \frac{\Delta f}{f_m}}$ so, if $\boldsymbol{\beta \ll 1}$ ,the modulation is called **narrowband FM** and its bandwidth is approximately $\boldsymbol{2f_m}$ as a result we choose $\boldsymbol{k_f}$ (freq. sensitivity) as low as possible to keep pace with the message signal to reduce the frequency deviation.
- The input signal to noise ratio exceeds a threshold value. Below this threshold, the signal to noise output deteriorates rapidly, and for low inputs the signal to noise is worse than SSB.
- Generally, Frequency modulation can be classified as narrowband if the change in the carrier frequency is about the same as the signal frequency in which the carrier is allowed to deviate only 2.5 kHz above and below the center frequency like in 'family radio service'

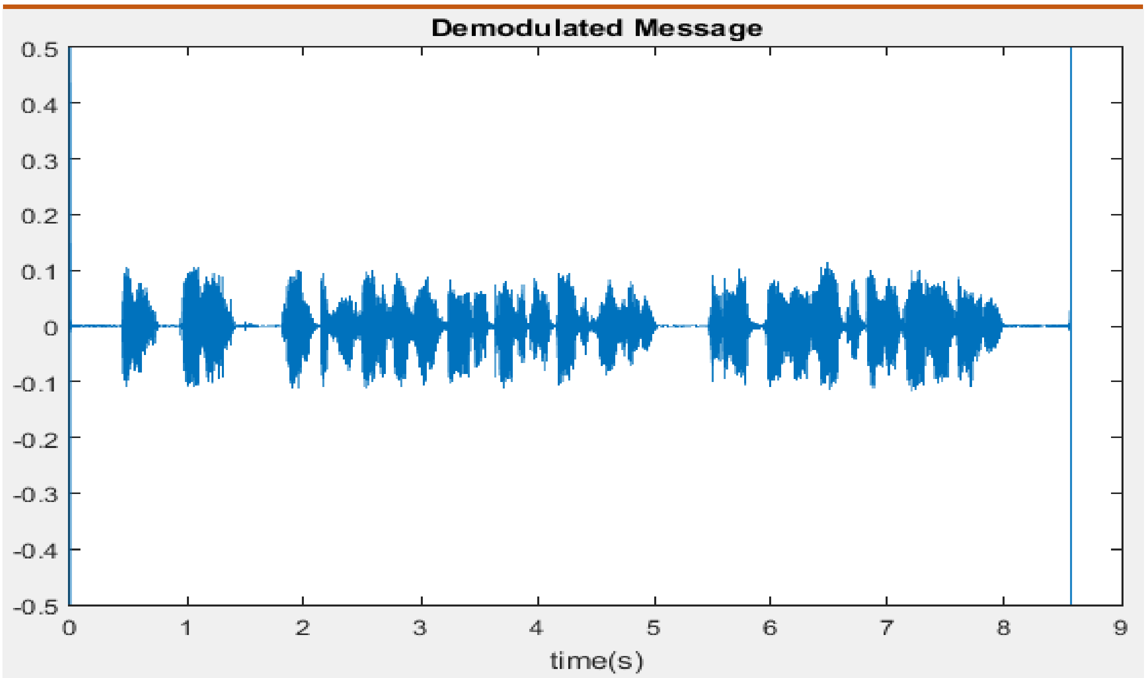
## Part (4)

Demodulate the NBFM signal using a differentiator and an ED. For the differentiator, you can use the following command: diff. Assume no noise is introduced.

### *M.code*

```
%% Demodulation
% demodulation is achieved by using a differentiator followed by
% an envelope detector

% envelope detector
out2 = abs(hilbert(NBFM_signal));
out2_f = fftshift(fft(out2)) ;
%plot
figure
subplot(2,1,1)
plot(t, out2, 'c') ; xlabel('time(s)') ; title(' filtered envelope signal in time domain ') ;
subplot(2,1,2)
plot(Fdom2, abs(out2_f), 'c') ; xlabel('freq(Hz)') ; title('filtered envelope signal in freq
domain') ;

% differentiator
out1 = [0; diff(out2)];
%plot
figure
plot(t, out1); xlabel('time(s)') ;title('Demodulated Message') ;
ylim([-0.5 0.5]);
%playing final sound
sound(3 * resample(out1 , FS , FS2), FS)
```




## ***Comment***

- The result is highly favorable as there is nearly zero noise. However, with lower quality sound.
- Also, the final message we receive is nearly the same to that of the original one.