Cairo University

Faculty of Engineering

Electronics and Electrical Communications Engineering Department

Third Year

Analog Communications

Term Project

MATLAB implementation of a superheterodyne receiver

Student Name: يوسف أحمد محمد أحمد يس

Sec: 4   BN: 50

Contents

Table of figures

1. ***The transmitter***

This part contains the following tasks

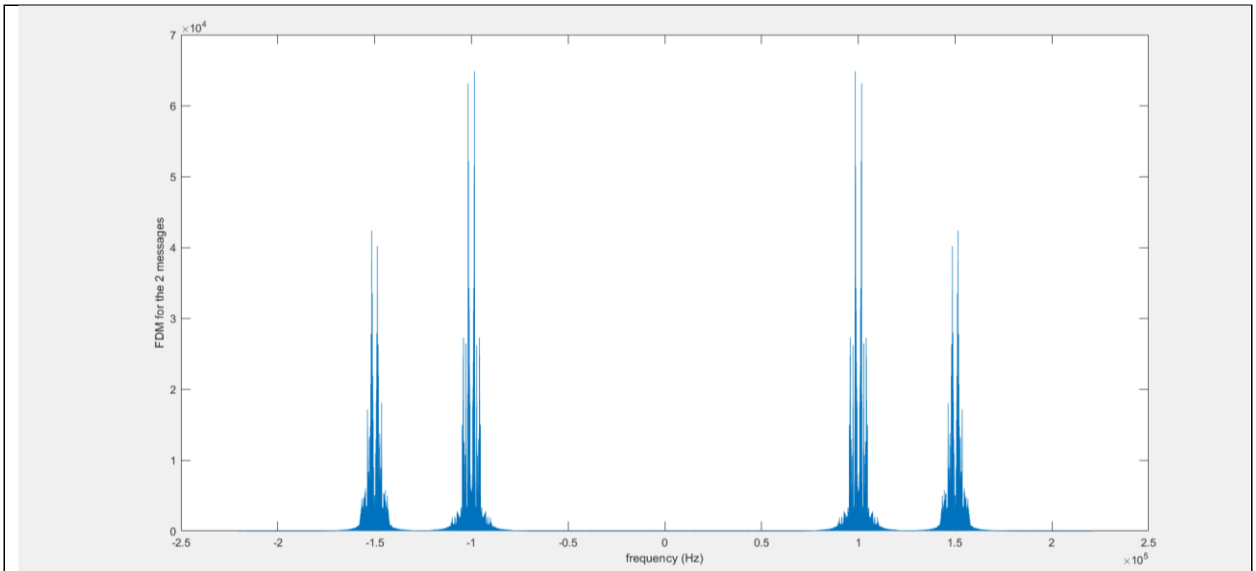
1. Reading monophonic audio signals into MATLAB.
2. Upsampling the audio signals.
3. Modulating the audio signals (each on a separate carrier).
4. Addition of the modulated signals.

Discussion

First we read the 2 stereo signals and convert them to be monophonic. Then we want to send the 2 messages on the same channel so we modulate them by multiplying by 2 different carriers ($\cos(200\pi t)$, $\cos(300\pi t)$). The modulated signals are used to construct an FDM signal by adding them ($m_1 \cos(200\pi t) + m_2 \cos(300\pi t)$). Matlab default sampling frequency=44.1kHz which is smaller than 2*100kHz,2*150kHz so we do upsampling and make the sampling frequency equal to 10*44100 (>2*100kHz,2*150kHz) (satisfying Nyquist criteria)

The figures

Figure 1: The spectrum of the output of the transmitter



2. ***The RF stage***

This part addresses the RF filter and the mixer following it.

Discussion

The RF stage is implemented as a band-pass Filter (BPF) only, centered at the carrier frequency. This stage is important as it rejects image frequencies (at $F = F_{Carrier} + 2F_{IF}$ that is at 150kHz for m_1 and 200kHz for m_2), noise and unwanted signals. The BPF of this stage is tunable to select the desired station. Then comes the mixer that shifts the required signal (in frequency domain) at the intermediate frequency (F_{IF} which =25kHz in our case). We do that to avoid direct conversion problems.

The figures

Assume we want to demodulate the first signal (at ω_o).

Figure 2: the output of the RF filter (before the mixer)

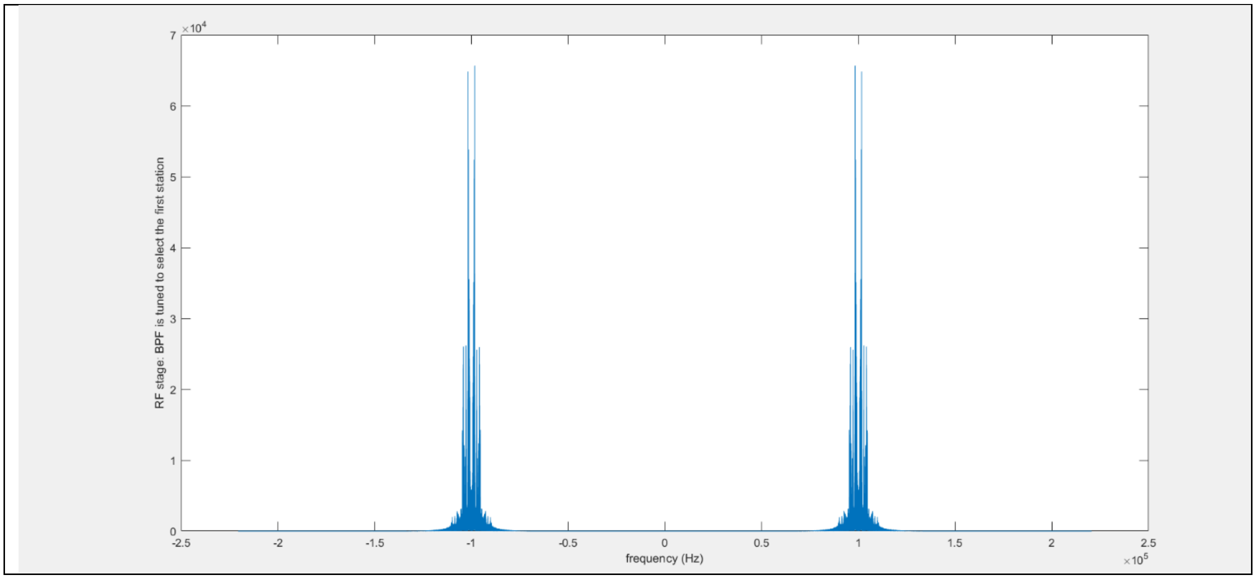
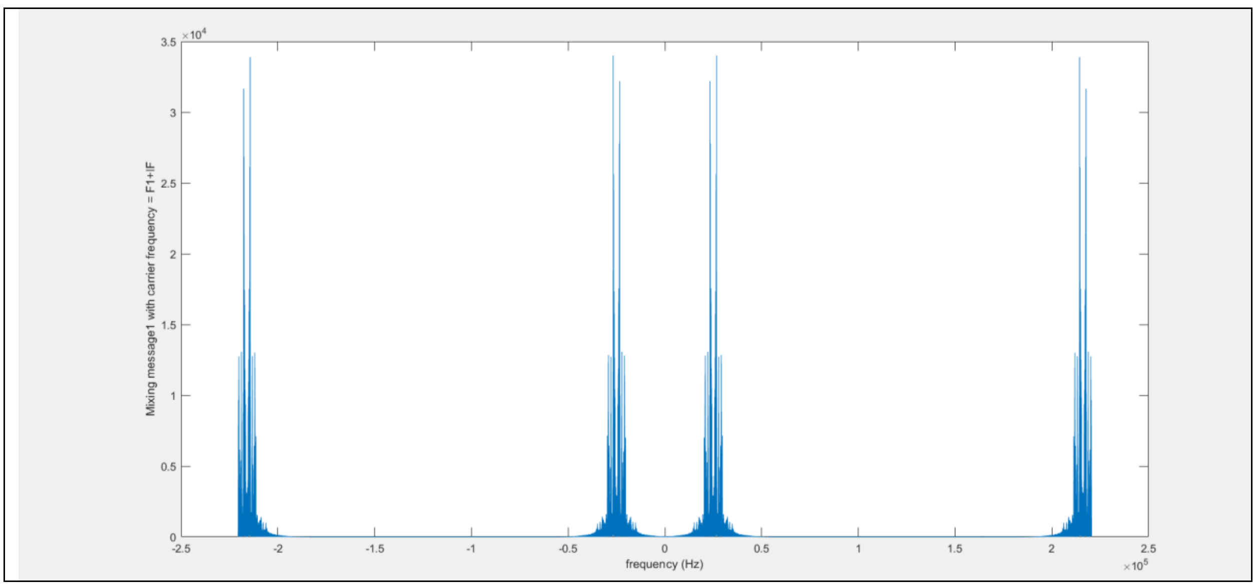


Figure 3: The output of the mixer



Comments

The image frequency at 150kHz ($F = F_{Carrier} + 2F_{IF}$) is rejected. Then we do intermediate conversion for the message by the mixer and its output is at $F_{IF} = 25kHz$ and the high frequency component will be suppressed by BPF of the IF stage.

3. *The IF stage*

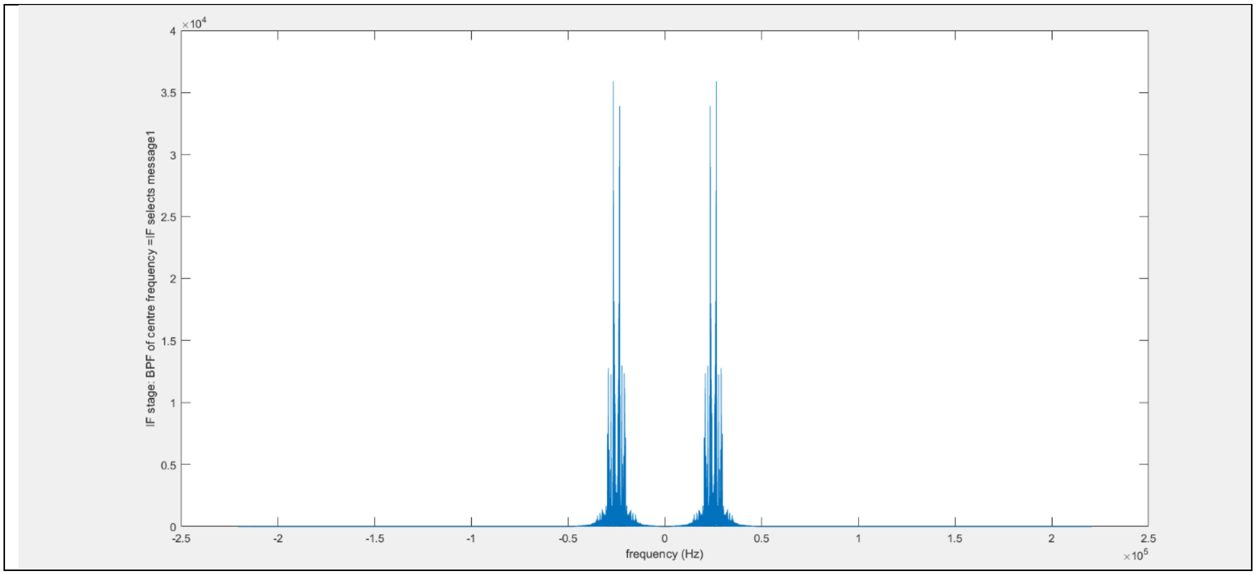
This part addresses the IF filter.

Discussion

This stage is implemented as a band-pass Filter (BPF) only, centered at the intermediate frequency F_{IF}. The BPF input is $A_{m1}m(t)\cos(\omega_{Carrier}t)\cos\big((\omega_{Carrier}+\omega_{IF})t\big) = A_{m1}m(t)(\cos(\omega_{IF}t)+\cos((2\omega_{Carrier}+\omega_{IF})t)$. The BPF removes the high frequency signal and the output is $m(t)\cos(\omega_{IF}t)$.

The figures

Figure 4: Output of the IF filter



4. *The baseband demodulator*

This part addresses the coherent detector used to demodulate the signal from the IF stage.

Discussion

The baseband demodulator consists of a mixer and a LPF. The mixer multiplies the signal $A_{m1}m(t)\cos(\omega_{IF}t)$ by $\cos(\omega_{IF}t)$ to shift it at baseband. The output of the mixer is $A_{m2}m(t) + A_{m2}\cos(2\omega_{IF}t)$. Then the LPF ,whose cutoff frequency= BW of the signal, passes the baseband signal $A_{m2}m(t)$ and reject the high frequency component $A_{m2}\cos(2\omega_{IF}t)$.

The figures

Figure 5: Output of the mixer (before the LPF)

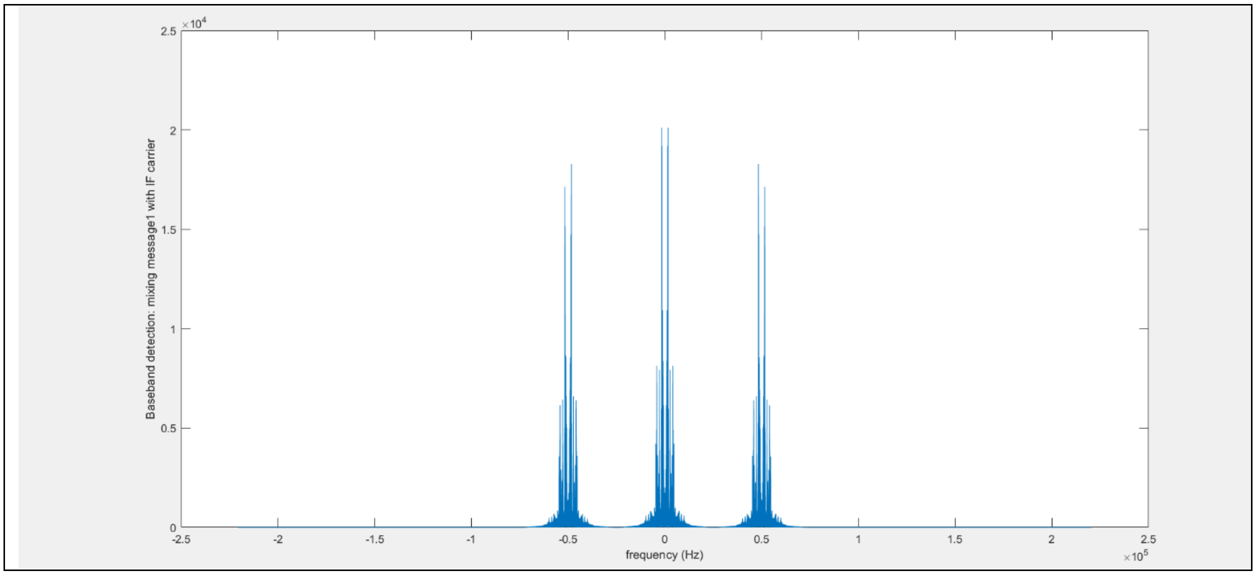
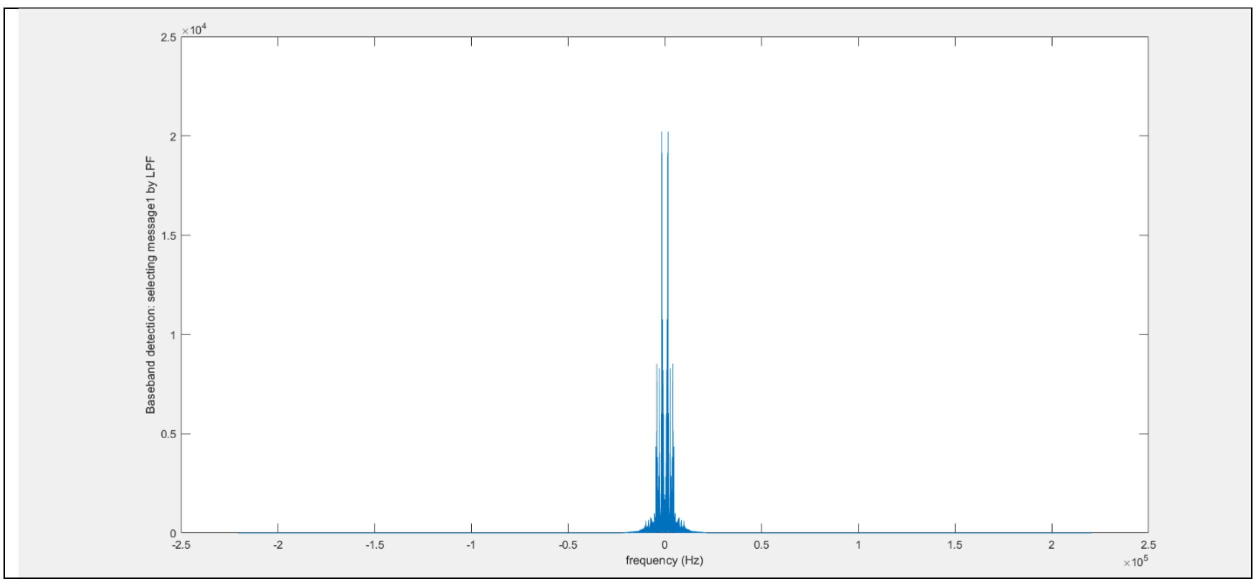


Figure 6: Output of the LPF



Comments

The mixer shifts the signal to the baseband and at $2F_{IF}$ (in frequency domain). The LPF suppresses the signal at $2F_{IF}$ and selects the baseband signal which is the output of the receiver (same as the transmitted message).

5. *Performance evaluation without the RF stage*

The figures

Figure 7: output of the RF mixer (no RF filter)

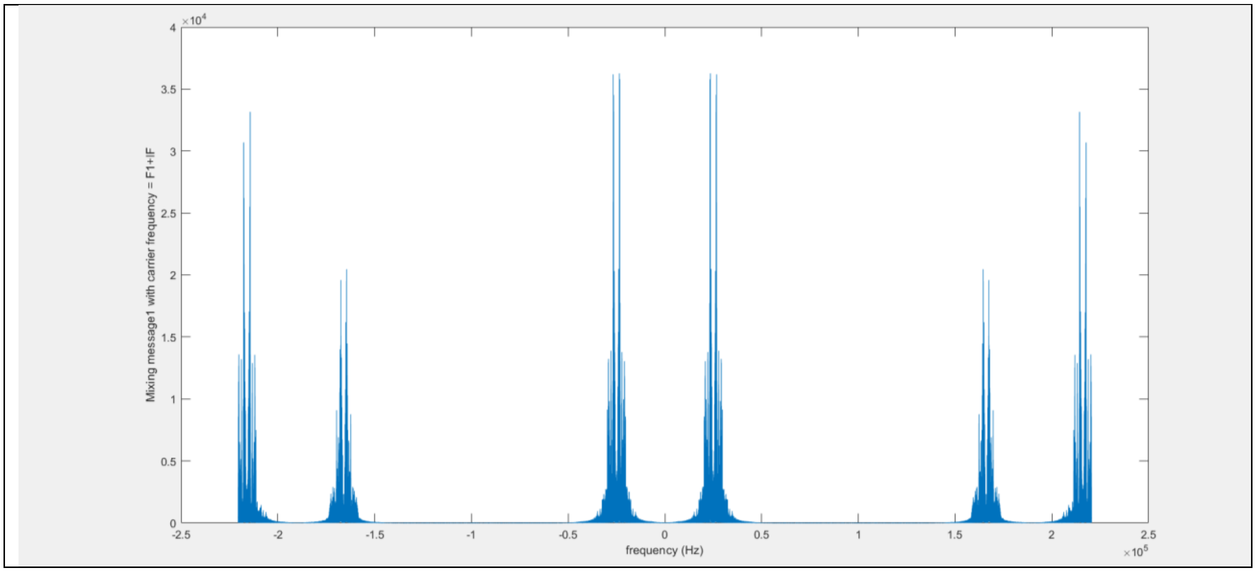


Figure 8: Output of the IF filter (no RF filter)

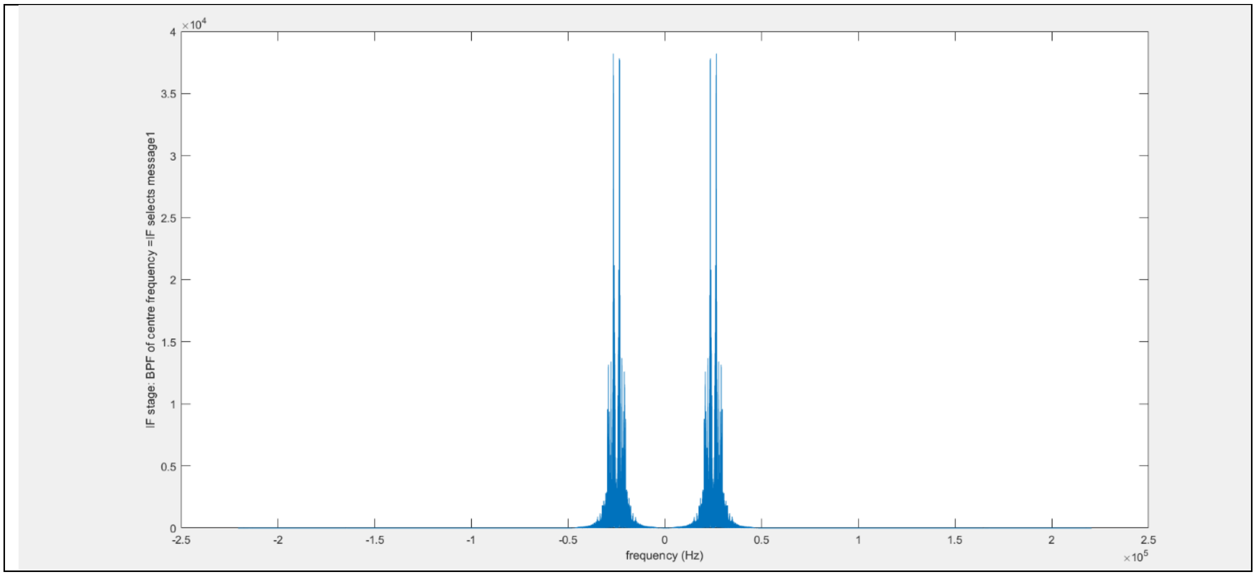


Figure 9: Output of the IF mixer before the LPF (no RF filter)

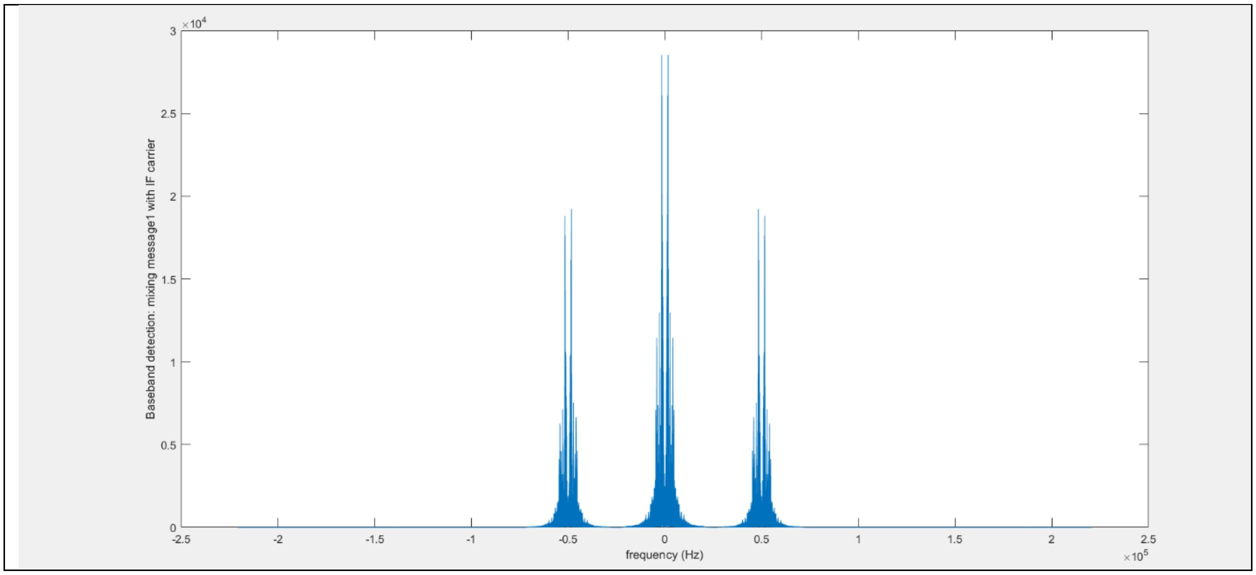
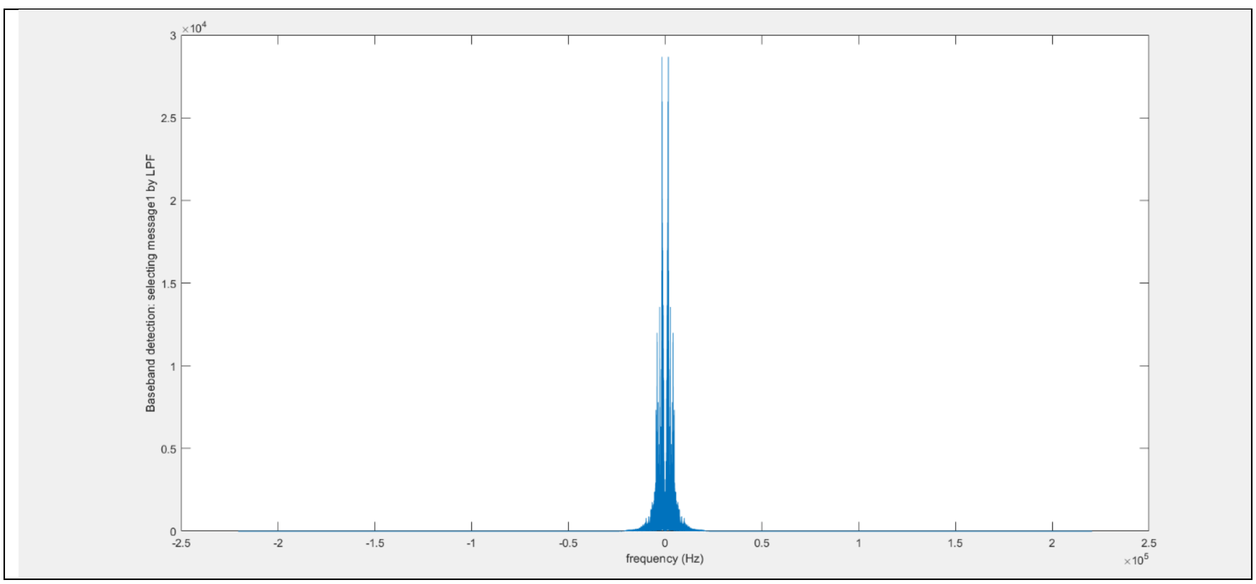


Figure 10: Output of the LPF (no RF filter)



6. ***Comment on the output sound***

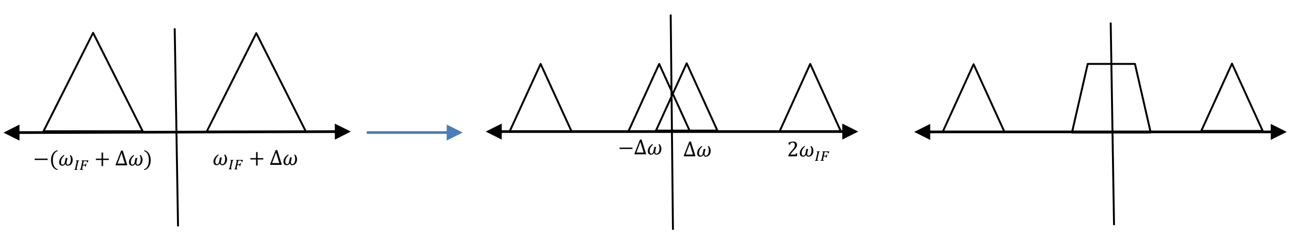
The output in case of RF stage existence is the same as the original audio signal without any interference with other signals. The output without RF stage (in case of receiving message1) is the original signal interfered with message2 and the output audio contains the two audio signals. This happens because the second message is modulated with carrier frequency: $F_{carrier2} = F_{carrier1} + 2F_{IF} = 100kHz + 2 * 25kHz = 150kHz$ which is an image frequency of $F_{carrier1}$. So with the absence of RF stage the image frequency is not rejected and interference occurred between the two messages.
From figure 7: output of the RF mixer (no RF filter) (figure of message1) message1 is corrupted as there are only 6 signals (not 8 signals) resulting from multiplying 4 signals by cos as a result of interference.

What happens (in terms of spectrum and the sound quality) if the receiver oscillator has frequency offset by 0.1 KHz and 1 KHz

The quality of the sound is bad (the sound is noisy) in case of 0.1kHz offset and worsens (becomes more noisy) in case of 1kHz offset. The input signal to the IF stage is $A_{m1}m(t)(\cos((\omega_{IF} + \Delta\omega)t) + \cos((2\omega_{Carrier} + \omega_{IF} + \Delta\omega)t)$. The high frequency component (second term) will be suppressed by the BPF of the IF stage. The output of IF stage is $A_{m1}m(t)\cos((\omega_{IF} + \Delta\omega)t)$ and this signal is now the input to the mixer that is multiplied by $\cos(\omega_{IF}t)$ and the output of the mixer is: $A_{m1}m(t)\cos((\omega_{IF} + \Delta\omega)t)\cos(\omega_{IF}t) = 0.5A_{m1}m(t)(\cos(\Delta\omega t) + \cos((2\omega_{IF} + \Delta\omega)t))$

The second term is suppressed by the LPF and the output is $0.5A_{m1}m(t)\cos(\Delta\omega t)$

The output is multiplied by a low frequency sinusoid, this causes attenuation and distortion of the output signal. The larger offset, the larger distortion in the output.



Before the mixer

After mixer: the output is distorted than the original signal

7. The code

```
clear all;clc;
% read the input sound file
[message1,fs]=audioread('Short_QuranPalestine.wav');
[message2,fs]=audioread('Short_FM9090.wav');

if(length(message1)>length(message2))
    message2=wextend('ar','zpd',message2,(length(message1)-
length(message2)),'d');
elseif (length(message2)>length(message1))
    message1=wextend('ar','zpd',message1,(length(message2)-
length(message1)),'d');
end

%length(message1)=length(message2)=N
N=length(message1);
%converting 2 channel signals to 1 channel signals
message1(:,1)=message1(:,1)+message1(:,2);
message1(:,2) = [];
message2(:,1)=message2(:,1)+message2(:,2);
message2(:,2) = [];
%Upsampling(interpolation)
message1 = interp(message1,10);
message2 = interp(message2,10);
fs=10*fs;

N=length(message1);
dt = 1/fs; % seconds per sample
StopTime = N/fs; % seconds
t = (0:dt:StopTime-dt)';
F1 = 100000; % cosine wave frequency (hertz)
carrier1 = cos(2*pi*F1*t);
carrier2 = cos(2*pi*(F1+50000)*t);

MESSAGE1=fft(message1);
MESSAGE2=fft(message2);
k=-N/2:N/2-1;
%figure
%plot(k*fs/N,fftshift(abs(MESSAGE1)));
%xlabel('frequency (Hz)');
%ylabel('1st message');

figure('units','normalized','outerposition',[0 0 1 1]);
myPlot(1,{'1st message (m_1)'},MESSAGE1,k,fs,N);

%figure
%plot(k*fs/N,fftshift(abs(MESSAGE2)));
%xlabel('frequency (Hz)');
%ylabel('2nd message');
myPlot(2,'2nd message (m_2)',MESSAGE2,k,fs,N);

transmitter_output1= message1.*carrier1;
transmitter_output2= message2.*carrier2;
clear message1 MESSAGE1 message2 MESSAGE2;
TRANSMITTER_OUTPUT1=fft(transmitter_output1);
TRANSMITTER_OUTPUT2=fft(transmitter_output2);
```

```
%FDM
TRANSMITTER_OUTPUT=TRANSMITTER_OUTPUT1+TRANSMITTER_OUTPUT2;
transmitter_output=ifft(TRANSMITTER_OUTPUT);

%figure
%plot(k*fs/N,fftshift(abs(TRANSMITTER_OUTPUT)));
%xlabel('frequency (Hz)');
%ylabel('FDM for the 2 messages');
myPlot(3,'FDM for the 2 messages',TRANSMITTER_OUTPUT,k,fs,N);

clear transmitter_output1 TRANSMITTER_OUTPUT1...
    transmitter_output2 TRANSMITTER_OUTPUT2;

%RF_BPF1 specs
Fstop1=0.65e+05; Fpass1=0.75e+05; Astop1=100;
Fpass2=1.25e+05;Fstop2=1.35e+05; Astop2=100;
Apass=1; Fs=fs;
BPF_specs=fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2', ...
    Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass,Astop2,Fs);

BPF = design(BPF_specs);
RF_output1= filter(BPF,transmitter_output);
RF_OUTPUT1=fft(RF_output1);

%figure
%plot(k*fs/N,fftshift(abs(RF_OUTPUT2)));
%xlabel('frequency (Hz)');
%ylabel('RF stage: BPF is tuned to select the first station');
myPlot(4,{'RF stage: BPF_1'},RF_OUTPUT1,k,fs,N);

%RF_BPF2 specs
BPF_specs.Fstop1=1.15e+05;
BPF_specs.Fpass1=1.25e+05;
BPF_specs.Fpass2=1.75e+05;
BPF_specs.Fstop2=1.85e+05;
BPF = design(BPF_specs);
RF_output2= filter(BPF,transmitter_output);
RF_OUTPUT2=fft(RF_output2);

%figure
%plot(k*fs/N,fftshift(abs(RF_OUTPUT2)));
%xlabel('frequency (Hz)');
%ylabel('RF stage: BPF is tuned to select the second station');
myPlot(5,'RF stage: BPF_2',RF_OUTPUT2,k,fs,N);

clear transmitter_output TRANSMITTER_OUTPUT;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%discussion (absence of RF stage)
%%%%%%%%%%%%%%
%RF_output1=a;
%RF_output2=a;
%RF_OUTPUT2=fft(RF_output1);
%RF_OUTPUT2=fft(RF_output2);
%message1 is corrupted as at C(figure4) there are only 6 signals resulting
%from multiplying 4 signals by cos
%%%%%%%%%%%%%%
```

```
F1_IF = 125000; % cosine wave frequency (hertz)
carrier1_IF = cos(2*pi*F1_IF*t);
mixer_output1=RF_output1.*carrier1_IF;
MIXER_OUTPUT1=fft(mixer_output1);
%
%figure
%plot(k*fs/N,fftshift(abs(MIXER_OUTPUT1)));
%xlabel('frequency (Hz)');
%ylabel('Mixing message1 with carrier frequency = F1+IF');
myPlot(6,'Mixer: mixing m_1, cos(F_C_1+F_I_F)',MIXER_OUTPUT1,k,fs,N);

F2_IF = 175000; % cosine wave frequency (hertz)
carrier2_IF = cos(2*pi*F2_IF*t);
mixer_output2=RF_output2.*carrier2_IF;
MIXER_OUTPUT2=fft(mixer_output2);
%
%figure
%plot(k*fs/N,fftshift(abs(MIXER_OUTPUT2)));
%xlabel('frequency (Hz)');
%ylabel('Mixing message2 with carrier frequency = F2+IF');
myPlot(7,'Mixer: mixing m_2, cos(F_C_2+F_I_F)',MIXER_OUTPUT2,k,fs,N);

clear RF_output1 RF_OUTPUT2 RF_output2 RF_OUTPUT2;

%IF_BPF design
BPF_specs.Fstop1=0.001e+05;
BPF_specs.Fpass1=0.05e+05;
BPF_specs.Fpass2=0.45e+05;
BPF_specs.Fstop2=0.499e+05;
BPF = design(BPF_specs);
IF_output1= filter(BPF,mixer_output1);
IF_OUTPUT1=fft(IF_output1);

%figure
%plot(k*fs/N,fftshift(abs(IF_OUTPUT1)));
%xlabel('frequency (Hz)');
%ylabel('IF stage: BPF of centre frequency =IF selects message1');
myPlot(8,'IF stage: BPF_1 of W_c_e_n_t_e_r=F_I_F',IF_OUTPUT1,k,fs,N);

IF_output2= filter(BPF,mixer_output2);
IF_OUTPUT2=fft(IF_output2);

%figure
%plot(k*fs/N,fftshift(abs(IF_OUTPUT2)));
%xlabel('frequency (Hz)');
%ylabel('IF stage: BPF of centre frequency =IF selects message2');
myPlot(9,'IF stage: BPF_2 of W_c_e_n_t_e_r=F_I_F',IF_OUTPUT2,k,fs,N);
clear mixer_output1 MIXER_OUTPUT1 mixer_output2 MIXER_OUTPUT2;

F_BaseBand = 25000; % cosine wave frequency (hertz)
carrier_BaseBand = cos(2*pi*F_BaseBand*t);

BB_mixer_output1 = IF_output1.*carrier_BaseBand;
BB_MIXER_OUTPUT1 = fft(BB_mixer_output1);

%figure
%plot(k*fs/N,fftshift(abs(BB_MIXER_OUTPUT1)));
```

```
%xlabel('frequency (Hz)');
%ylabel('Baseband detection: mixing message1 with IF carrier');
myPlot(10,'BB detector: mixing m_1, cos(F_I_F)',BB_MIXER_OUTPUT1,k,fs,N);

BB_mixer_output2 = IF_output2.*carrier_BaseBand;
BB_MIXER_OUTPUT2 = fft(BB_mixer_output2);

%figure
%plot(k*fs/N,fftshift(abs(BB_MIXER_OUTPUT2)));
%xlabel('frequency (Hz)');
%ylabel('Baseband detection: mixing message2 with IF carrier');
myPlot(11,'BB detector: mixing m_2, cos(F_I_F)',BB_MIXER_OUTPUT2,k,fs,N);
clear IF_output1 IF_OUTPUT1 IF_output2 IF_OUTPUT2;

%LPF design
Fpass=0.2e+05;
Fstop=0.3e+05;
Apass=1;
Astop=100;
Fs=fs;
LPF_specs=fdesign.lowpass('Fp,Fst,Ap,Ast', ...
    Fpass, Fstop, Apass, Astop, Fs);
LPF = design(LPF_specs);
LPF_output1= filter(LPF,BB_mixer_output1);
LPF_OUTPUT1= fft(LPF_output1);

%figure
%plot(k*fs/N,fftshift(abs(LPF_OUTPUT1)));
%xlabel('frequency (Hz)');
%ylabel('Baseband detection: selecting message1 by LPF');
myPlot(12,'BB detector: select m_1 by LPF',LPF_OUTPUT1,k,fs,N);

LPF_output2= filter(LPF,BB_mixer_output2);
clear BB_mixer_output1 BB_MIXER_OUTPUT1 BB_mixer_output2 BB_MIXER_OUTPUT2;
LPF_OUTPUT2= fft(LPF_output2);

%figure
%plot(k*fs/N,fftshift(abs(LPF_OUTPUT2)));
%xlabel('frequency (Hz)');
%ylabel('Baseband detection: selecting message2 by LPF');
myPlot(13,'BB detector: select m_2 by LPF',LPF_OUTPUT2,k,fs,N);
xlabel('frequency (kHz)');

audiowrite('message1.wav',LPF_output1,fs);
audiowrite('message2.wav',LPF_output2,fs);

%function for plotting
function [] = myPlot(plot_number,title,signal,k,fs,N)
subplot(13,1,plot_number)
plot(k*fs/(N*10e3),fftshift(abs(signal)));
set(gca,'FontSize',6)
%xlabel('frequency (Hz)');
y=ylabel(title, 'FontSize', 8);
set(get(gca,'YLabel'),'Rotation',0)
set(y, 'position', get(y,'position')-[3,0,0]);
set(gca,'YTick',[])
end
```