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: Mohamed Hosam Osman

Sec: 3 BN: 37

Contents

Table of figures

1. The transmitter

This part contains the following tasks

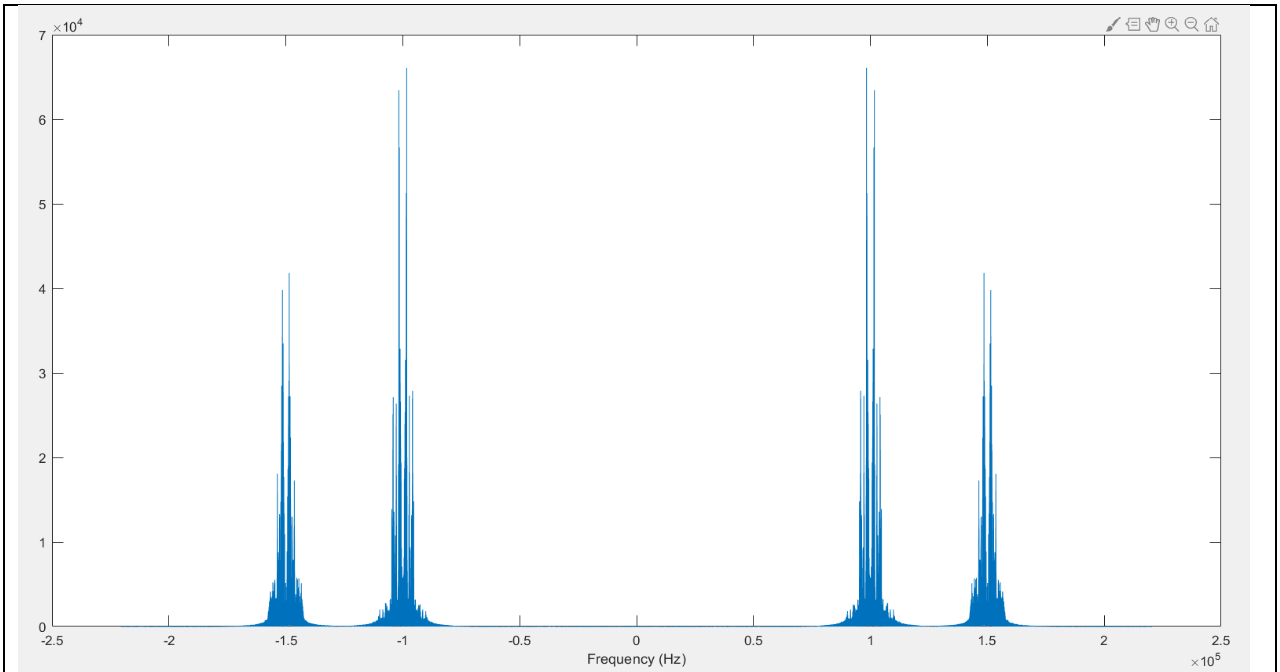
1. Reading monophonic audio signals into MATLAB.
2. Upsampling the audio signals.
3. Modulating the audio signals (each on a separate carrier).
4. Addition of the modulated signals.

Discussion

Firstly, the MATLAB has read the 2 audio signals using audioread() function and then increase their sampling frequency to 10 times its value to be more than the double of the carrier frequencies on which each signal of them will be modulated separately and then they will be added together on the same channel in order to apply frequency devision multiplexing.

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

In this stage, we made a bandpass filter to filter the received signal at the carrier frequency ω_n which is tuned to be the center frequency of any of the modulated signals in order to allow it to pass to the second stage. The filter bandwidth must be less than $2*\omega_{IF}$ so as not to allow signals at $\omega_n+2*\omega_{IF}$ to pass and thus when the output signals out of the filter are multiplied by a carrier of frequency $\omega_n+\omega_{IF}$,the signal at ω_n only is shifted at ω_{IF}.

The figures

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

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

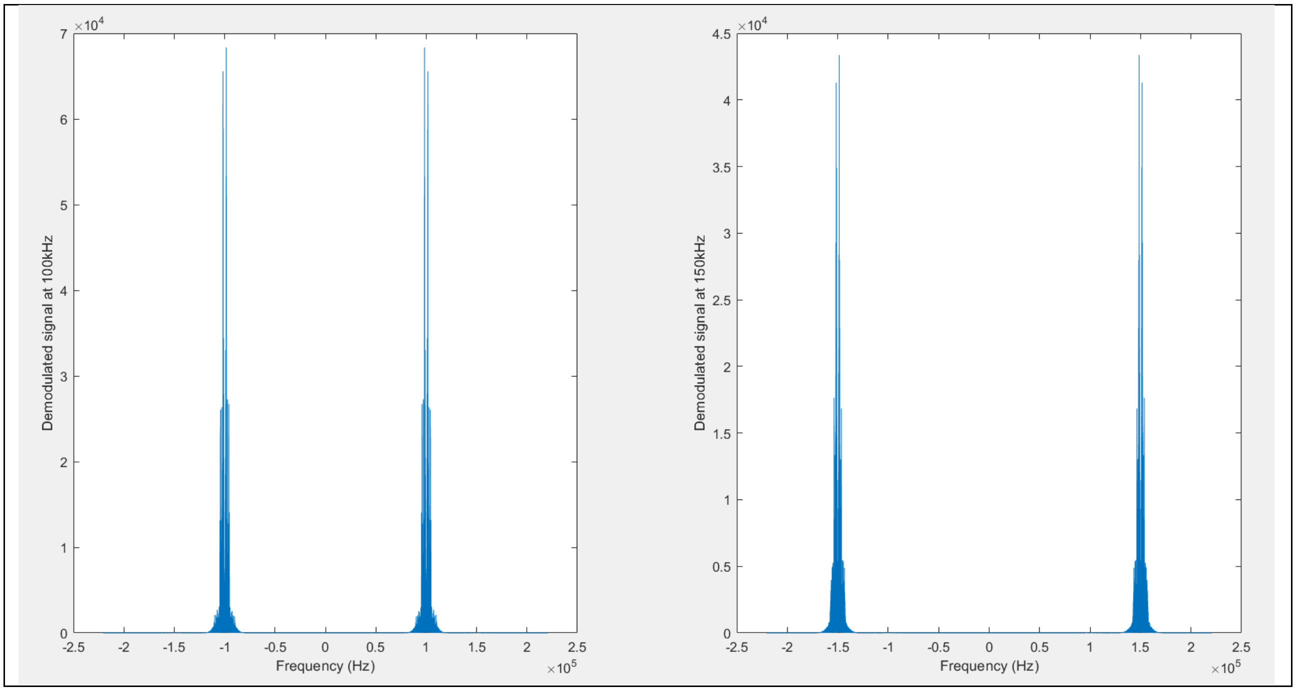
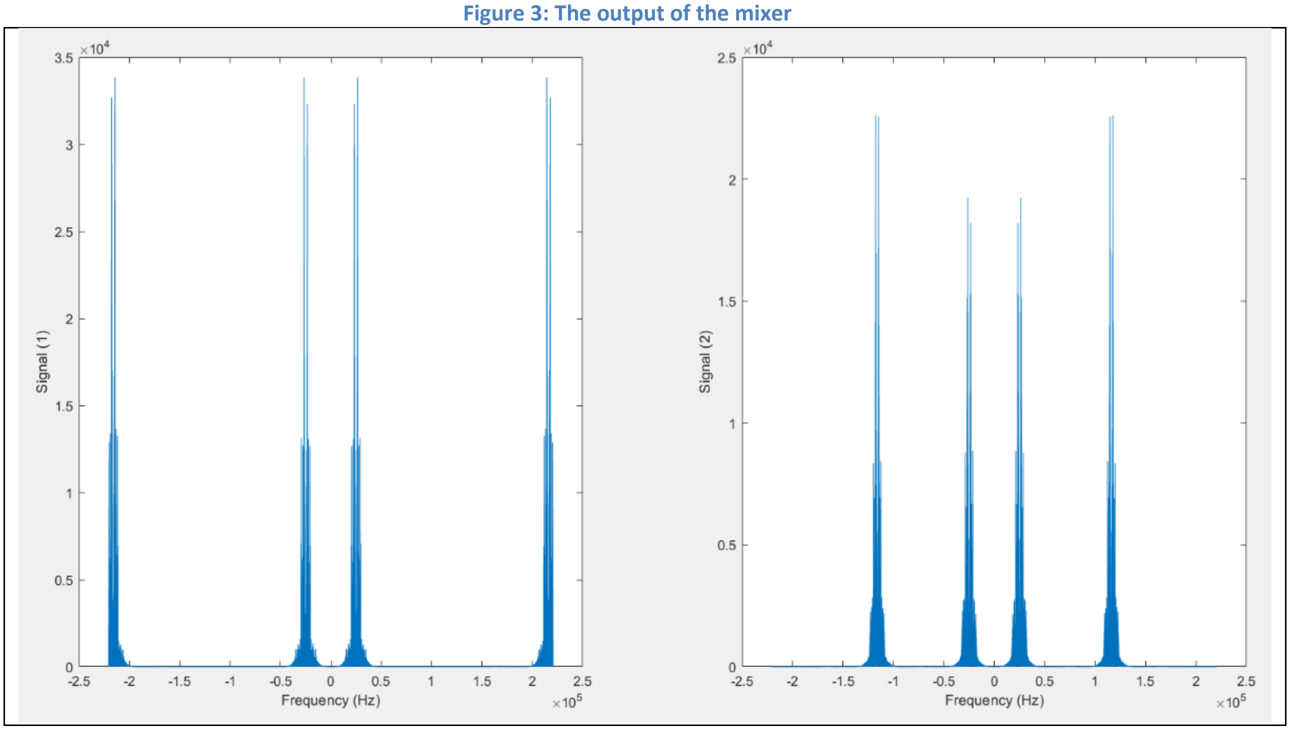


Figure 3: The output of the mixer



3. ***The IF stage***

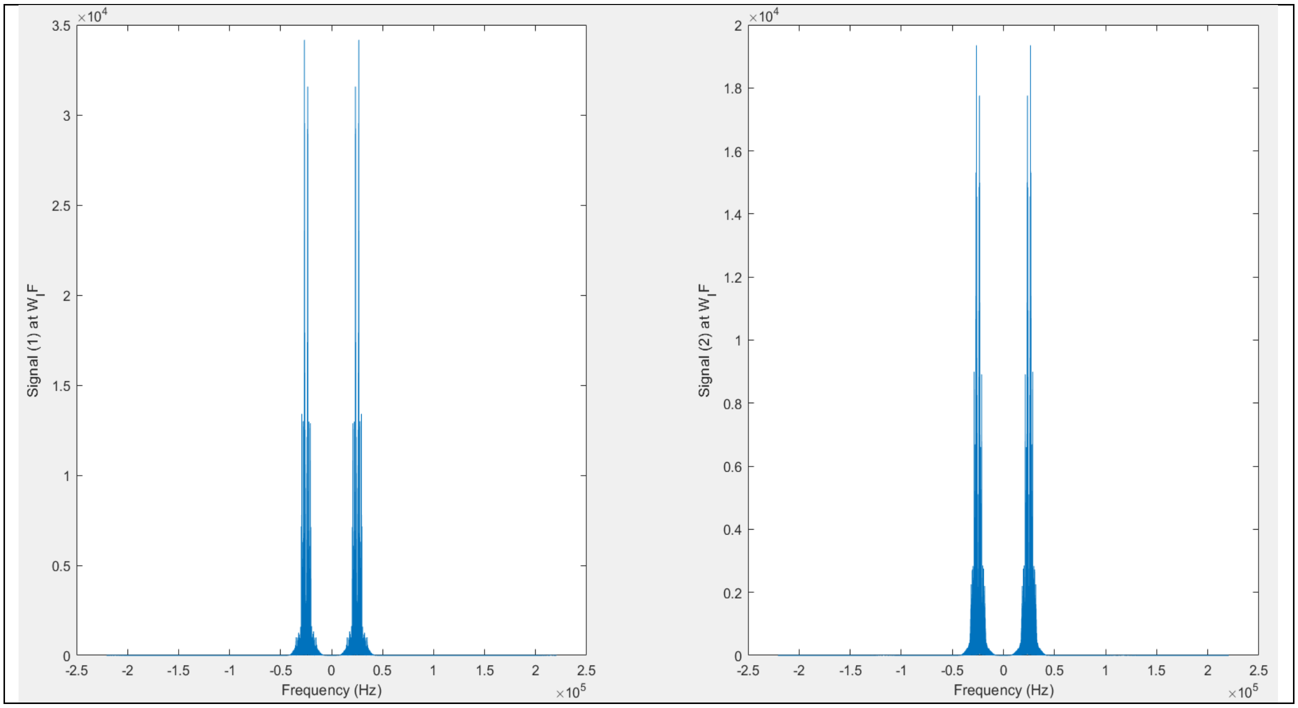
This part addresses the IF filter.

Discussion

In this stage, we filter the output of the mixer with a bandpass filter centered at ω_{IF}.

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

In this stage, we multiply each signal by a carrier of frequency ω_{IF} to return each signal to the baseband and then we filter each of them by a lowpass filter to have each signal back as it was.

The figures

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

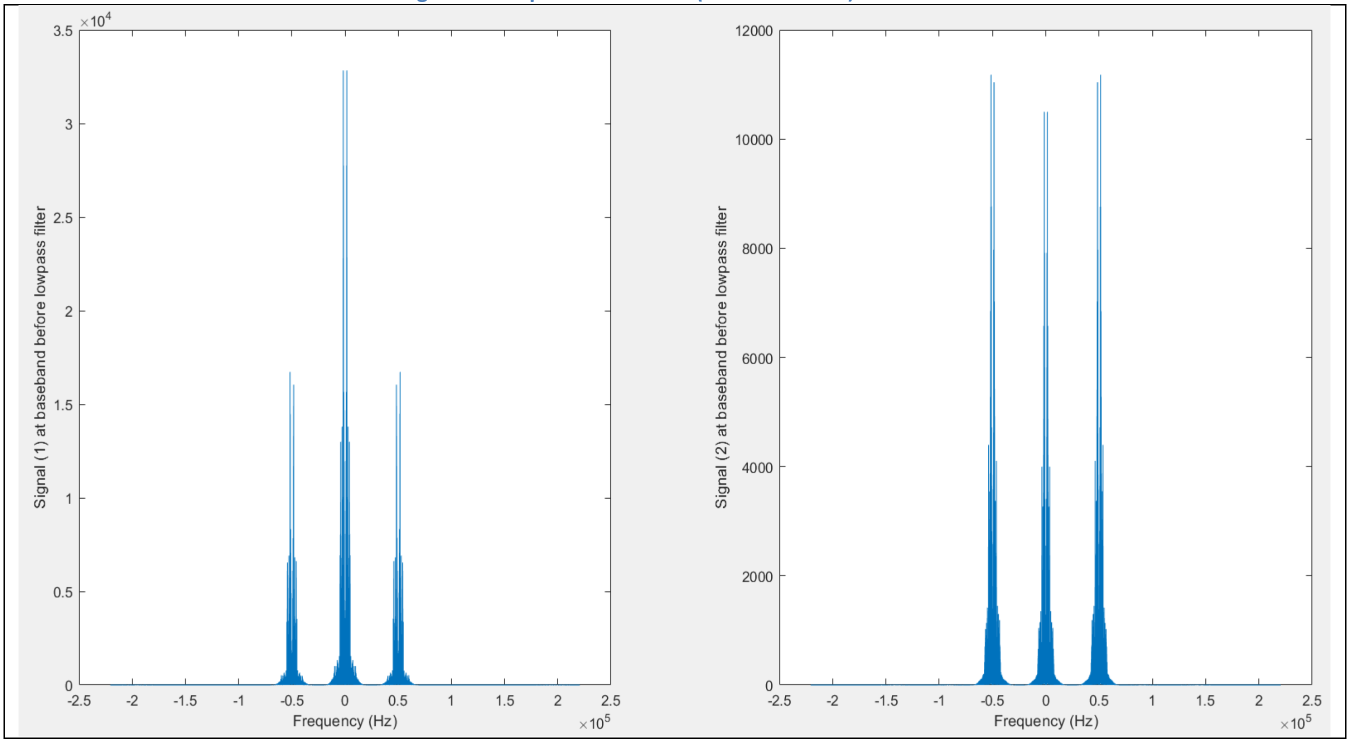
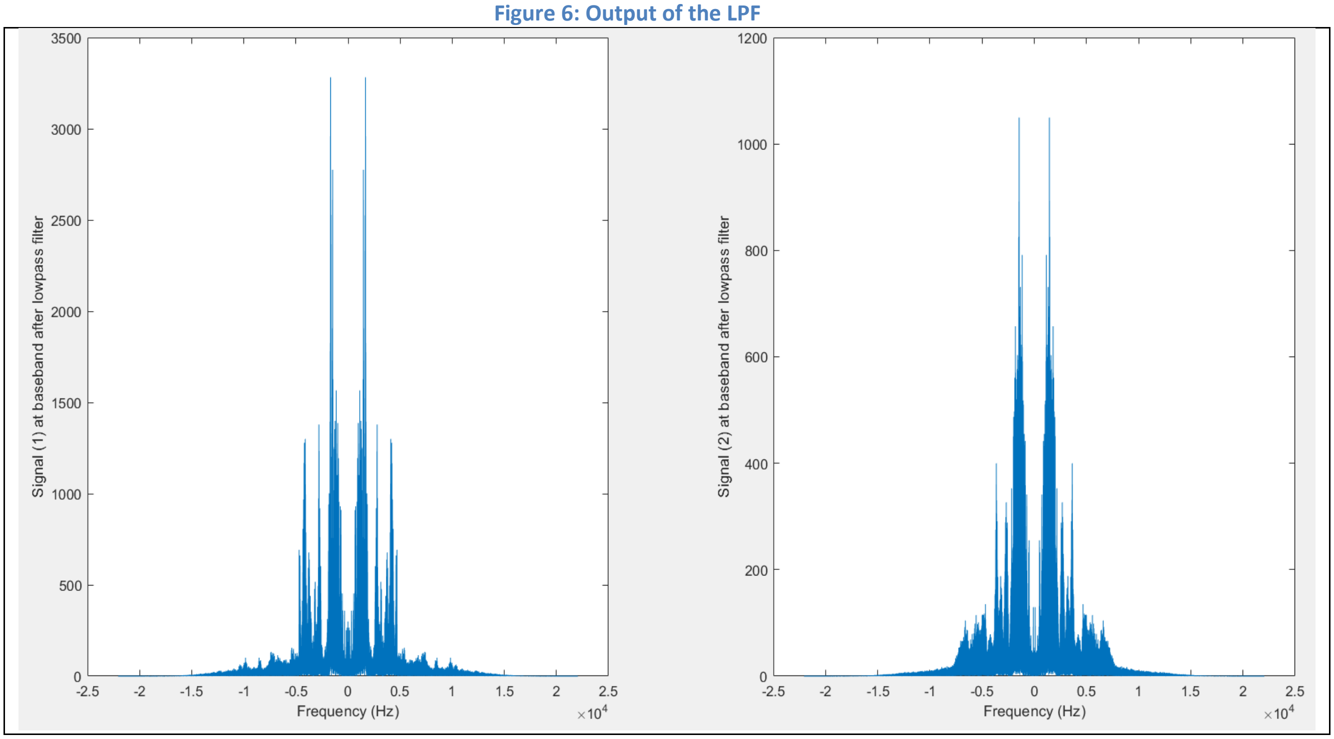


Figure 6: Output of the LPF



5. ***Performance evaluation without the RF stage***

The figures

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

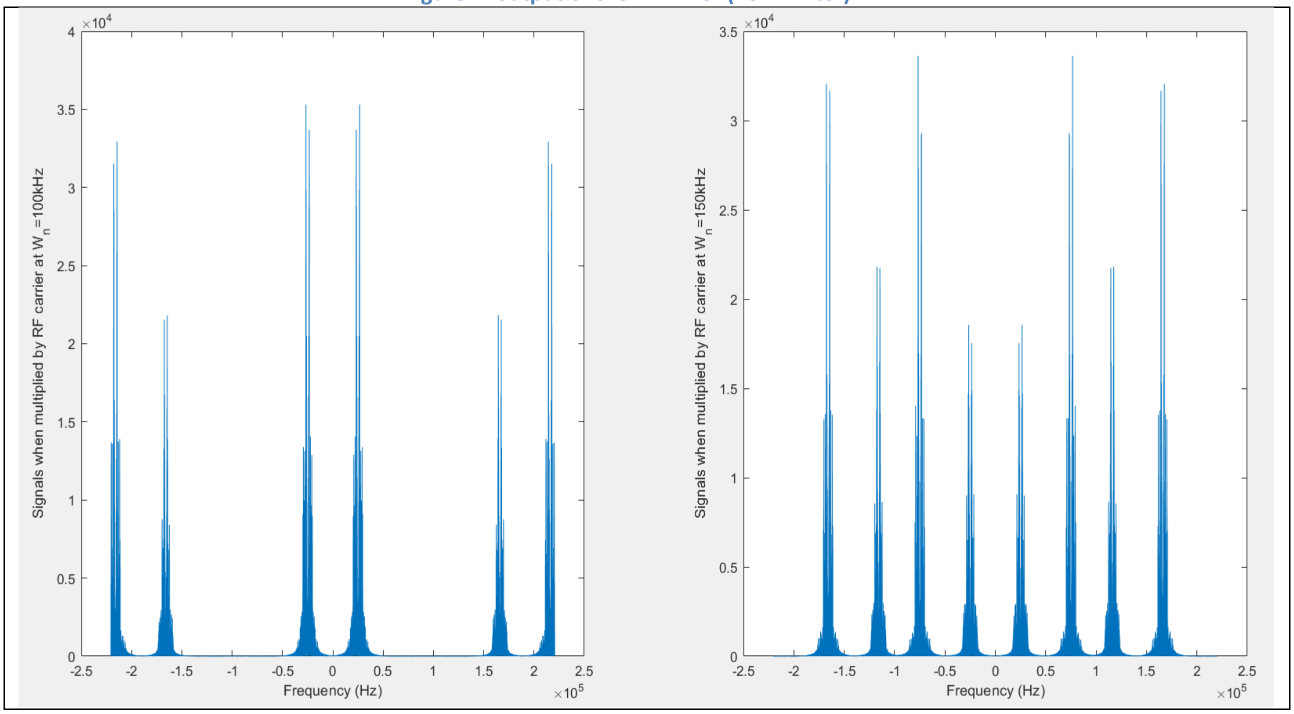


Figure 8: Output of the IF filter (no RF filter)

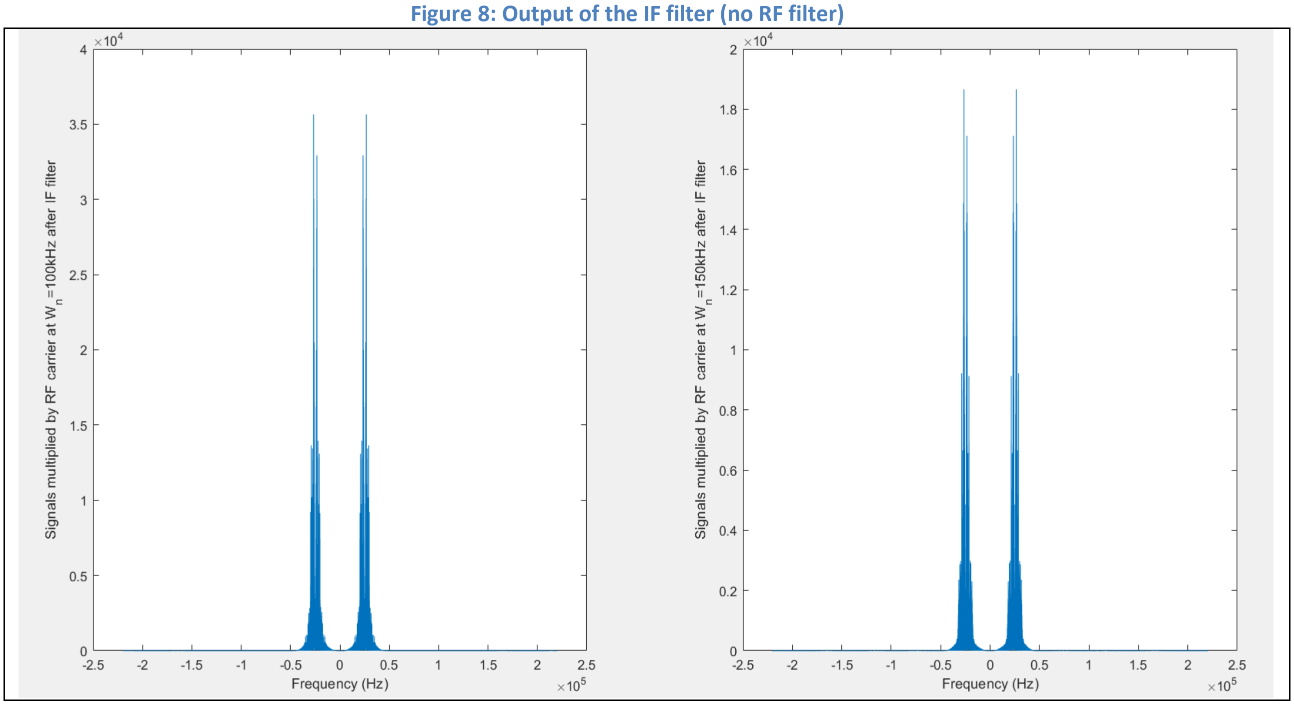


Figure 9: Output of the IF mixer before the LPF (no RF filter)

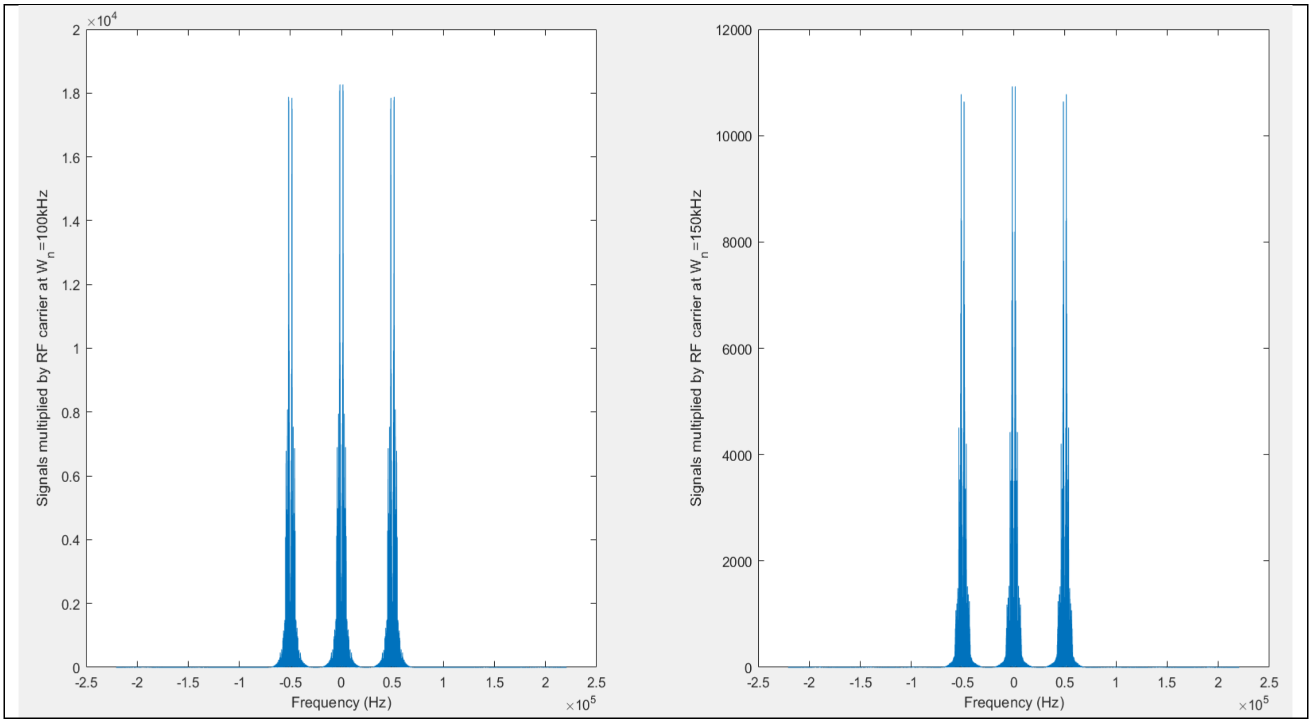
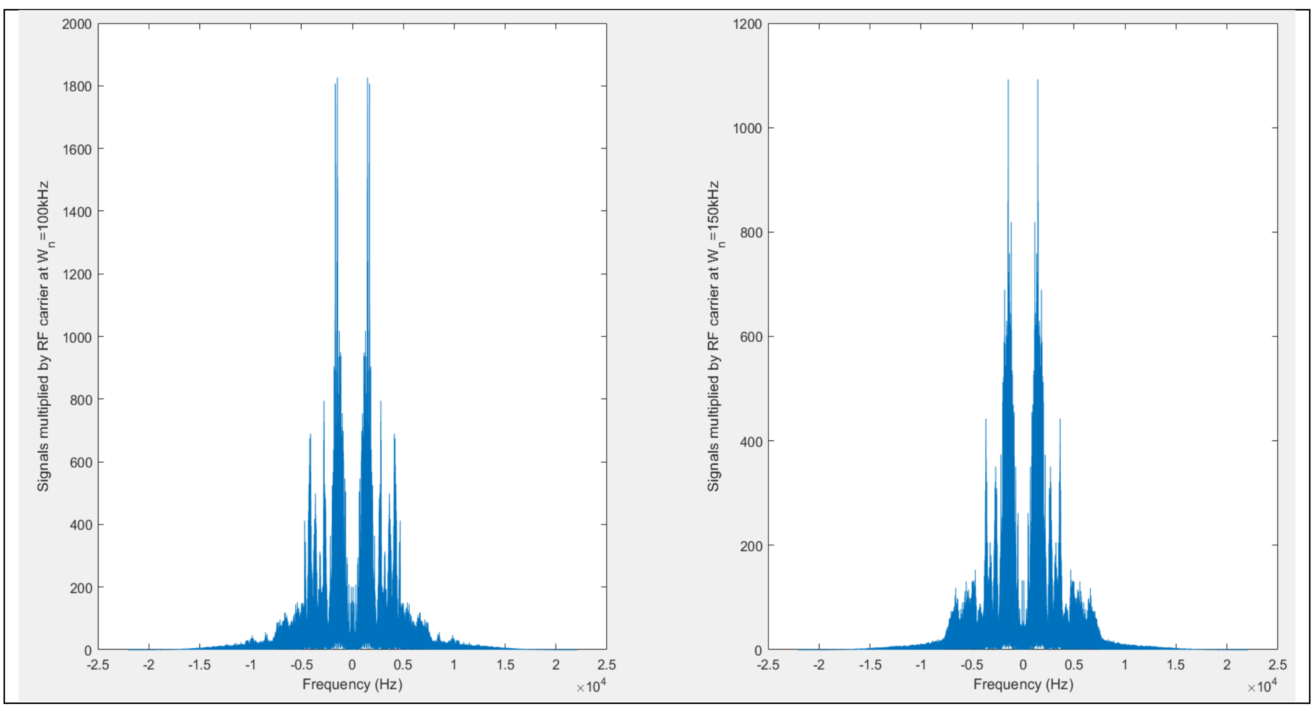


Figure 10: Output of the LPF (no RF filter)

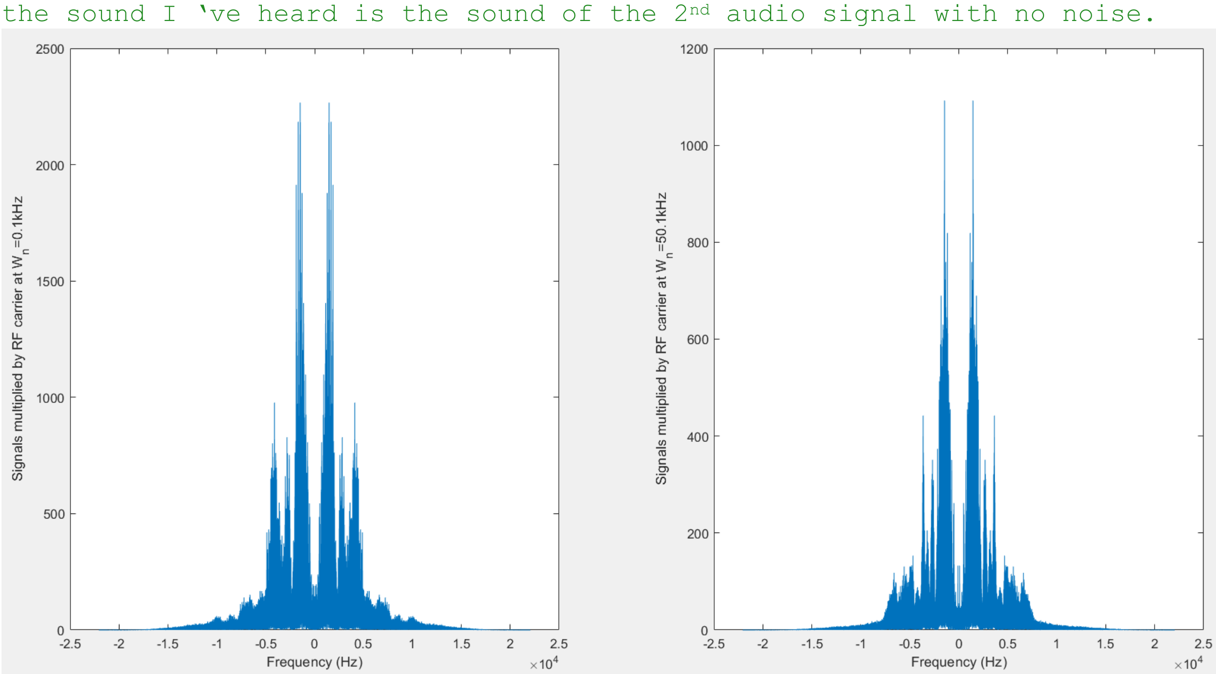


6. Comment on the output sound

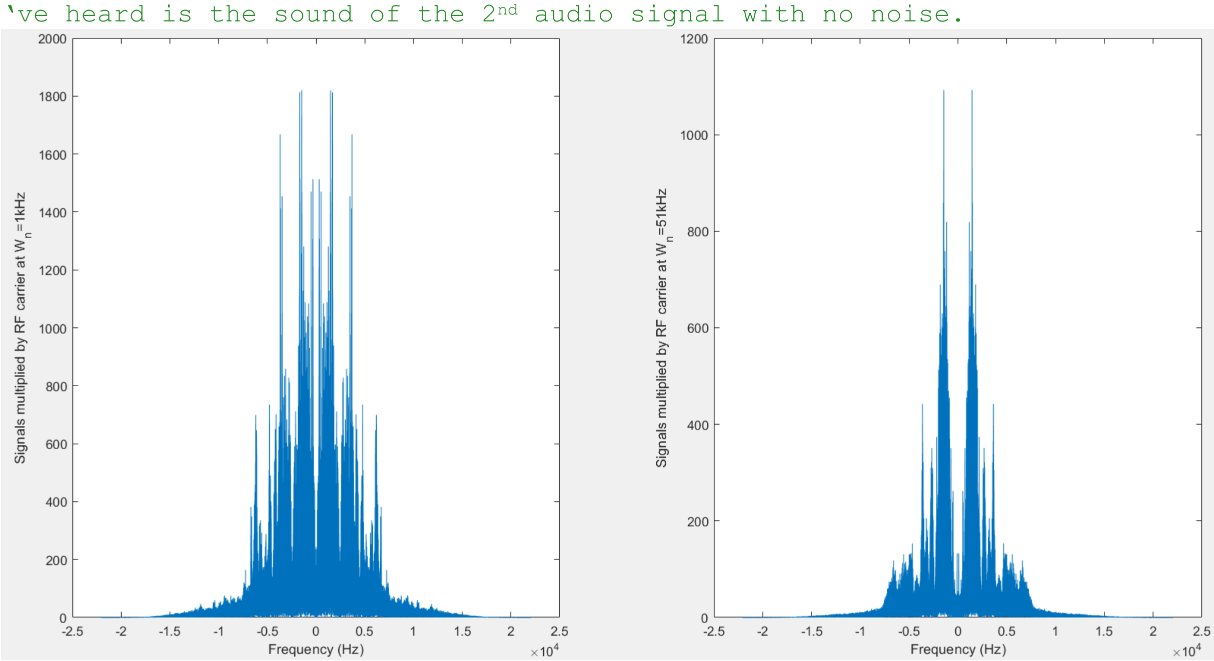
At the absence of the RF stage, when ω_n=100kHz the sound I 've heard is the sound of the 2 audio signals overlapped over each other, while at ω_n=150kHz the sound I 've heard is the sound of the 2^{nd} audio signal only. At the presence of the RF stage, when ω_n=100kHz the sound I 've heard is the sound of the 1^{st} audio signal only, while at ω_n=150kHz the sound I 've heard is the sound of the 2^{nd} audio signal only.

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

-In case of having a receiver oscillator frequency offset =0.1kHz, at ω_n=0.1kHz the sound I 've heard has some noise in it, while at ω_n=50.1kHz the sound I 've heard is the sound of the 2nd audio signal with no noise.



-In case of having a receiver oscillator frequency offset =1kHz, at ω_n=1kHz the sound I 've heard has so much noise in it, while at ω_n=51kHz the sound I 've heard is the sound of the 2nd audio signal with no noise.



7. The code

```
                              %%The signals%%
[signal1 fs1]=audioread('Short_QuranPalestine.wav') ;
[signal2 fs2]=audioread('Short_FM9090.wav') ;
signal1=signal1(:,1)+signal1(:,2) ;
signal2=signal2(:,1)+signal2(:,2) ;
if size(signal1,1)<741440
    z=zeros(741440-size(signal1,1),1) ;
    signal1=[signal1 ; z] ;
end
if size(signal2,1)<741440
    z=zeros(741440-size(signal2,1),1) ;
    signal2=[signal2 ; z] ;
end
N=length(signal1) ;
                                %%Plotting%%
signal1_fft=fft(signal1,N) ;
signal2_fft=fft(signal2,N) ;
k=-N/2:N/2-1 ;
%subplot(1,2,1) ;
%plot(k*fs1/N,fftshift(abs(signal1_fft))) ;
%subplot(1,2,2) ;
%plot(k*fs2/N,fftshift(abs(signal2_fft))) ;

                             %%The AM modulator%%
signal1=interp(signal1,10) ; signal2=interp(signal2,10) ;N=length(signal1);
fs1=fs1*10 ; fs2=fs1 ;       %increase the fs of both siganls%
Ts1=1/fs1 ; Ts2=1/fs2 ; Ts=[Ts1 ; Ts2] ;
fn=zeros(2,1) ;
carrier=zeros(2,N) ;
for i=1:2
fn(i,1)=(100*10^3)+(i-1)*(50*10^3) ;
n=-N/2:N/2-1 ;
carrier(i,:)=cos(2*pi*fn(i,1)*n*Ts(i,1)) ;
end
carrier=carrier.' ;
modulated_signal1=signal1.*carrier(:,1) ;
modulated_signal2=signal2.*carrier(:,2) ;
modulated_signals=modulated_signal1+modulated_signal2 ;
                                %%Plotting%%
%modulated_signals_fft=fft(modulated_signals,N) ;
%k=-N/2:N/2-1 ;
%plot(k*fs1/N,fftshift(abs(modulated_signals_fft))) ;
%xlabel('Frequency (Hz)') ;

                              %%The RF stage%%
BW=13*10^3 ;
for i=1:2
Fstop1=fn(i,1)-BW-(10*10^3) ;
Fpass1=fn(i,1)-BW ;
Fpass2=fn(i,1)+BW ;
Fstop2=fn(i,1)+BW+(10*10^3) ;
Fs=fs1 ;
bandpassspecs=fdesign.bandpass('N,Fst1,Fp1,Fp2,Fst2,C',100,Fstop1,Fpass1,Fpass2,Fsto
p2,Fs);
bandpassspecs.Stopband1Constrained = true;
bandpassspecs.Astop1 = 60;
bandpassspecs.Stopband2Constrained = true;
bandpassspecs.Astop2 = 60;
bandpassFilter = design(bandpassspecs,'Systemobject',true) ;
%fvtool(bandpassFilter) ;
if i==1
```

```matlab
   modulated_signal1=bandpassFilter(modulated_signals) ;
else
   modulated_signal2=bandpassFilter(modulated_signals) ;
end
end
                                 %%Plotting%%
%modulated_signal1_fft=fft(modulated_signal1,N) ;
%modulated_signal2_fft=fft(modulated_signal2,N) ;
%k=-N/2:N/2-1 ;
%subplot(1,2,1) ;
%plot(k*fs1/N,fftshift(abs(modulated_signal1_fft))) ;
%xlabel('Frequency (Hz)') ;
%ylabel('Demodulated signal at 100kHz') ;
%subplot(1,2,2) ;
%plot(k*fs1/N,fftshift(abs(modulated_signal2_fft))) ;
%xlabel('Frequency (Hz)') ;
%ylabel('Demodulated signal at 150kHz') ;

                            %%The Oscillator%%
F_IF=25*10^3 ;
carrier_nIF=zeros(2,N) ;
fc=zeros(2,1) ;
for i=1:2
fc(i,1)=fn(i,1)+F_IF ;
n=-N/2:N/2-1 ;
carrier_nIF(i,:)=cos(2*pi*fc(i,1)*n*Ts(i,1)) ;
end
carrier_nIF=carrier_nIF.' ;
signal1_nIF = modulated_signal1.*carrier_nIF(:,1) ;
signal2_nIF = modulated_signal2.*carrier_nIF(:,2) ;
                                 %%Plotting%%
%signal1_nIF_fft=fft(signal1_nIF,N) ;
%signal2_nIF_fft=fft(signal2_nIF,N) ;
%k=-N/2:N/2-1 ;
%subplot(1,2,1) ;
%plot(k*fs1/N,fftshift(abs(signal1_nIF_fft))) ;
%xlabel('Frequency (Hz)') ;
%ylabel('Signal (1)') ;
%subplot(1,2,2) ;
%plot(k*fs1/N,fftshift(abs(signal2_nIF_fft))) ;
%xlabel('Frequency (Hz)') ;
%ylabel('Signal (2)') ;

                             %%The IF stage%%
Fstop1=F_IF-BW-(10*10^3) ;
Fpass1=F_IF-BW ;
Fpass2=F_IF+BW ;
Fstop2=F_IF+BW+(10*10^3) ;
Fs=fs1 ;
bandpassspecs=fdesign.bandpass('N,Fst1,Fp1,Fp2,Fst2,C',100,Fstop1,Fpass1,Fpass2,Fsto
p2,Fs);
bandpassspecs.Stopband1Constrained = true;
bandpassspecs.Astop1 = 60;
bandpassspecs.Stopband2Constrained = true;
bandpassspecs.Astop2 = 60;
bandpassFilter = design(bandpassspecs,'Systemobject',true) ;
%fvtool(bandpassFilter) ;
signal1_IF=bandpassFilter(signal1_nIF) ;
signal2_IF=bandpassFilter(signal2_nIF) ;
                                 %%Plotting%%
%signal1_IF_fft=fft(signal1_IF,N) ;
%signal2_IF_fft=fft(signal2_IF,N) ;
%k=-N/2:N/2-1 ;
```

```
%subplot(1,2,1) ;
%plot(k*fs1/N,fftshift(abs(signal1_IF_fft))) ;
%xlabel('Frequency (Hz)') ;
%ylabel('Signal (1) at W_IF') ;
%subplot(1,2,2) ;
%plot(k*fs1/N,fftshift(abs(signal2_IF_fft))) ;
%xlabel('Frequency (Hz)') ;
%ylabel('Signal (2) at W_IF') ;

                        %%The Baseband detection%%
carrier_IF=zeros(1,N) ;
n=-N/2:N/2-1 ;
carrier_IF(1,:)=cos(2*pi*F_IF*n*Ts(1,1)) ;
carrier_IF=carrier_IF.' ;
demodulated_signal1=signal1_IF.*carrier_IF ;
demodulated_signal2=signal2_IF.*carrier_IF ;
                                  %%Plotting%%
%demodulated_signal1_fft=fft(demodulated_signal1,N) ;
%demodulated_signal2_fft=fft(demodulated_signal2,N) ;
%k=-N/2:N/2-1 ;
%subplot(1,2,1) ;
%plot(k*fs1/N,fftshift(abs(demodulated_signal1_fft))) ;
%xlabel('Frequency (Hz)') ;
%ylabel('Signal (1) at baseband before lowpass filter') ;
%subplot(1,2,2) ;
%plot(k*fs1/N,fftshift(abs(demodulated_signal2_fft))) ;
%xlabel('Frequency (Hz)') ;
%ylabel('Signal (2) at baseband before lowpass filter') ;

Fs = fs1;
Fpass = BW;
Fstop = BW+(10*10^3);
Apass = 0.01;
Astop = 80;
lowpass_filtSpecs = fdesign.lowpass(Fpass,Fstop,Apass,Astop,Fs);
lowpass_Filter = design(lowpass_filtSpecs,'equiripple','SystemObject',true) ;
%fvtool(lowpass_Filter,'Fs',Fs);
signal1_baseband=lowpass_Filter(demodulated_signal1) ;
signal2_baseband=lowpass_Filter(demodulated_signal2) ;
%signal1_baseband=4*signal1_baseband ;
%signal2_baseband=4*signal2_baseband ;
signal1_baseband=downsample(signal1_baseband,10) ;
signal2_baseband=downsample(signal2_baseband,10) ;
fs=fs1/10 ;
                                  %%Plotting%%
%N=length(signal1_baseband) ;
%signal1_baseband_fft=fft(signal1_baseband,N) ;
%signal2_baseband_fft=fft(signal2_baseband,N) ;
%k=-N/2:N/2-1 ;
%subplot(1,2,1) ;
%plot(k*fs/N,fftshift(abs(signal1_baseband_fft))) ;
%xlabel('Frequency (Hz)') ;
%ylabel('Signal (1) at baseband after lowpass filter') ;
%subplot(1,2,2) ;
%plot(k*fs/N,fftshift(abs(signal2_baseband_fft))) ;
%xlabel('Frequency (Hz)') ;
%ylabel('Signal (2) at baseband after lowpass filter') ;

sound(signal1_baseband,fs) ;
pause(17) ;
sound(signal2_baseband,fs) ;
```