Cairo University

Faculty of Engineering

Electronics and Electrical Communications Engineering Department

Third Year

Analog Communications

Term Project

MATLAB implementation of a superheterodyne receiver

Code	BN	Sec	Student Name
9203221	30	3	محمد خالد الرفاعي عبد اللطيف
9202710	25	2	شريف احمد فوزي عبد الوكيل

Contents

Table of figures

1. ***The transmitter***

This part contains the following tasks

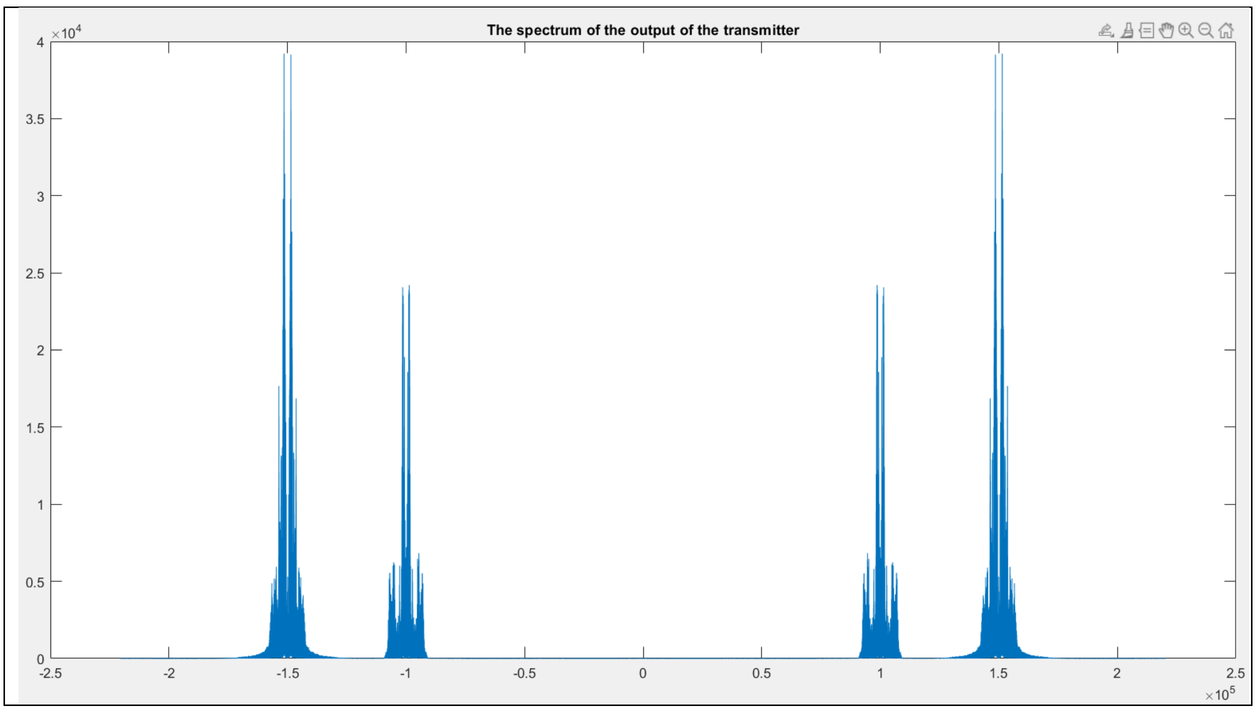
1. Reading monophonic audio signals into MATLAB.
2. Upsampling the audio signals.
3. Modulating the audio signals (each on a separate carrier).
4. Addition of the modulated signals.

Discussion

After reading two signals, changing them from stereo to mono channel and making them have the same length so that I can add them together, we increase the number of samples to avoid aliasing, we modulate each signal with its carrier and add them after increasing signal1 amplitude as it was small compared to signal2 amplitude, then we apply fourier transform and use fftshift command to plot the spectrum and see every signal places at its carrier frequency.

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 RF stage: we do band pass filter (centered at ωc and with BW = 2 Fif kHz) to filter the desired signal and reject the other one, we will filter the first signal which its carrier frequency is f1 in the code then we will reject its image, in this case the image of the first signal is the second signal because it lies at

2*ωIF from it (**the user will be asked in the code which channel he wants to play, accordingly the desired signal will be filtered**). In Mixer stage: we generate oscillator with ω = ωC + ωIF and multiply it by the output of RF stage to get the desired signal at ωIF and at high frequency.

The figures

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

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

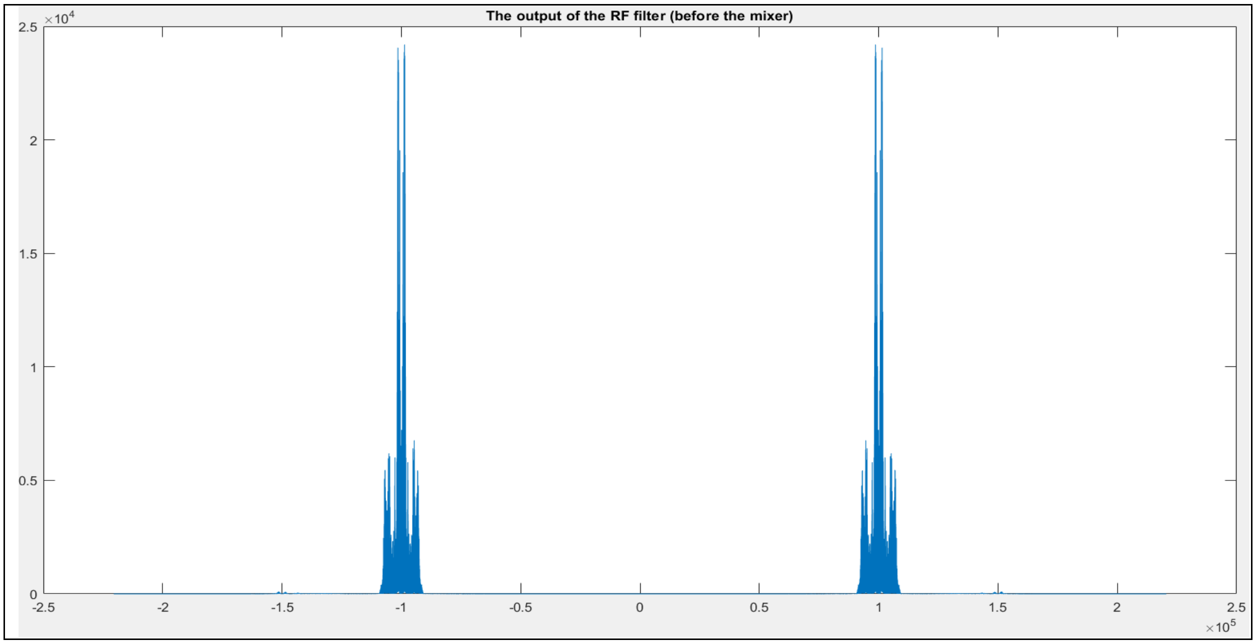
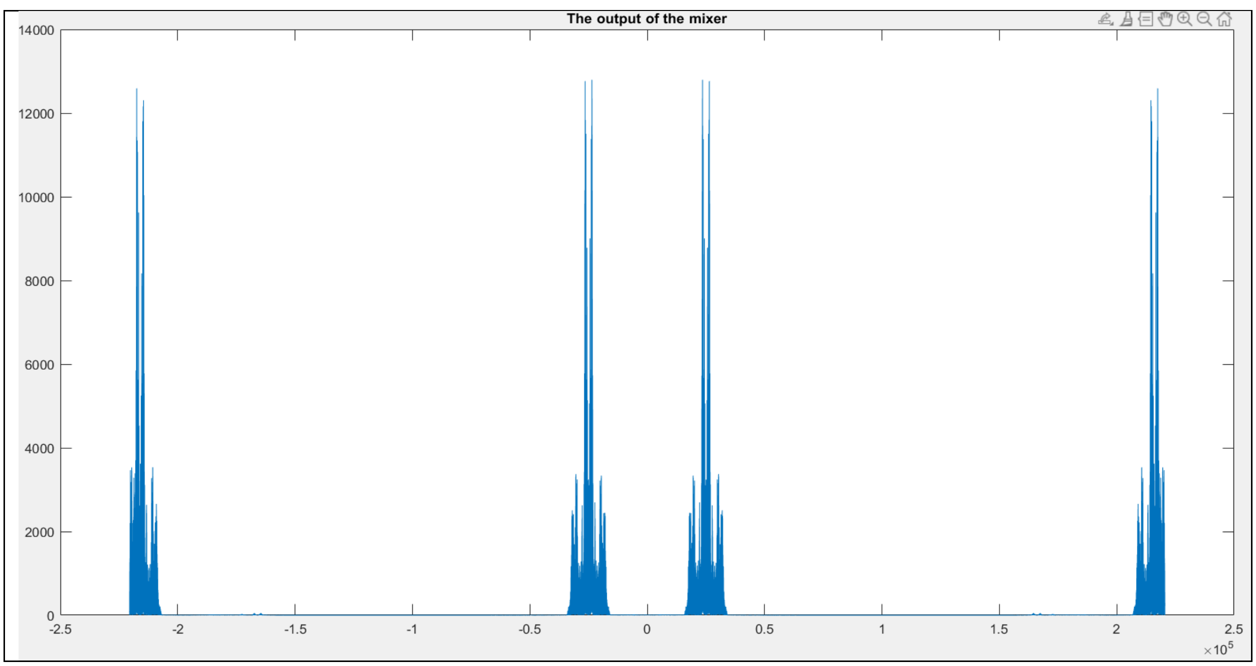


Figure 3: The output of the mixer



3. ***The IF stage***

This part addresses the IF filter.

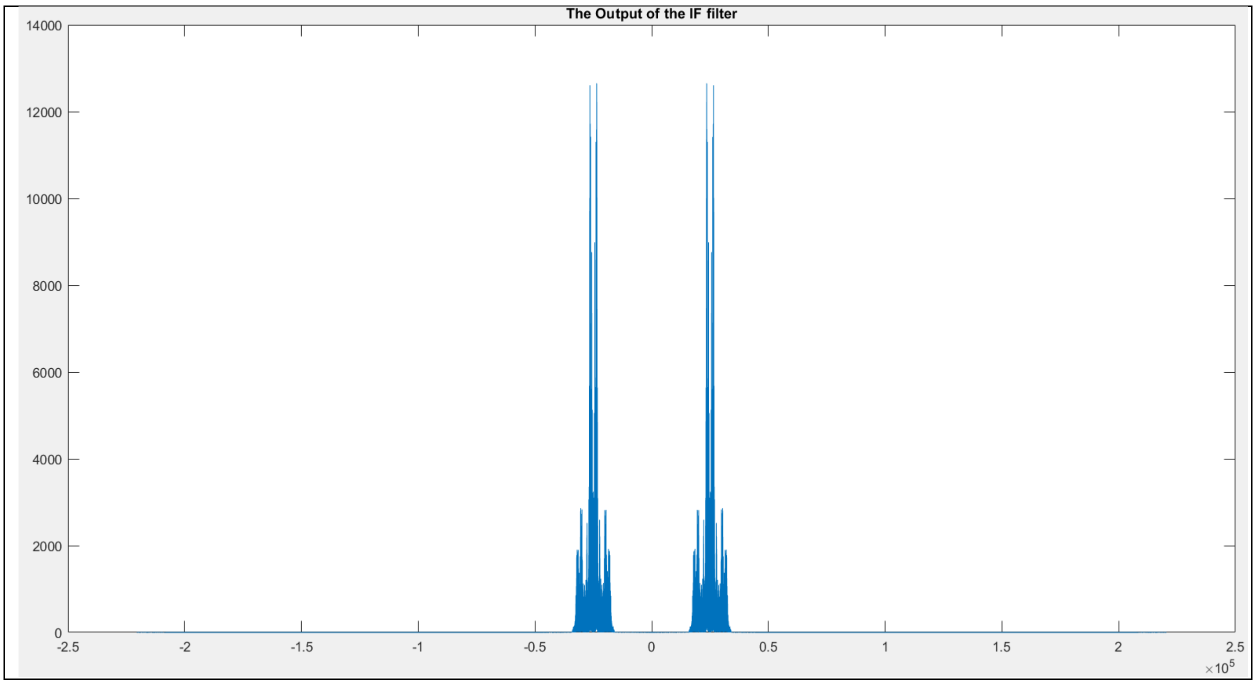
Discussion

In this stage:

We made a band pass filter centered at ωIF and with Band width (BW)= 20 kHz, and I filter the output signal of Mixer, so I get the desired signal at ωIF without any noise interference.

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 baseband stage: we demodulate the output signal of IF stage by a carrier signal with carrier frequency at ωIF, so we can have the desired signal at the baseband and at 2 ωIF. In LPF stage we apply low pass filter with cut off frequency= 9 kHz to obtain the desired signal at baseband only, after that we do downsampling to return the original sampling rate (fs = 44100) , finally I can listen to the chosen channel well without any noise interference.

The figures

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

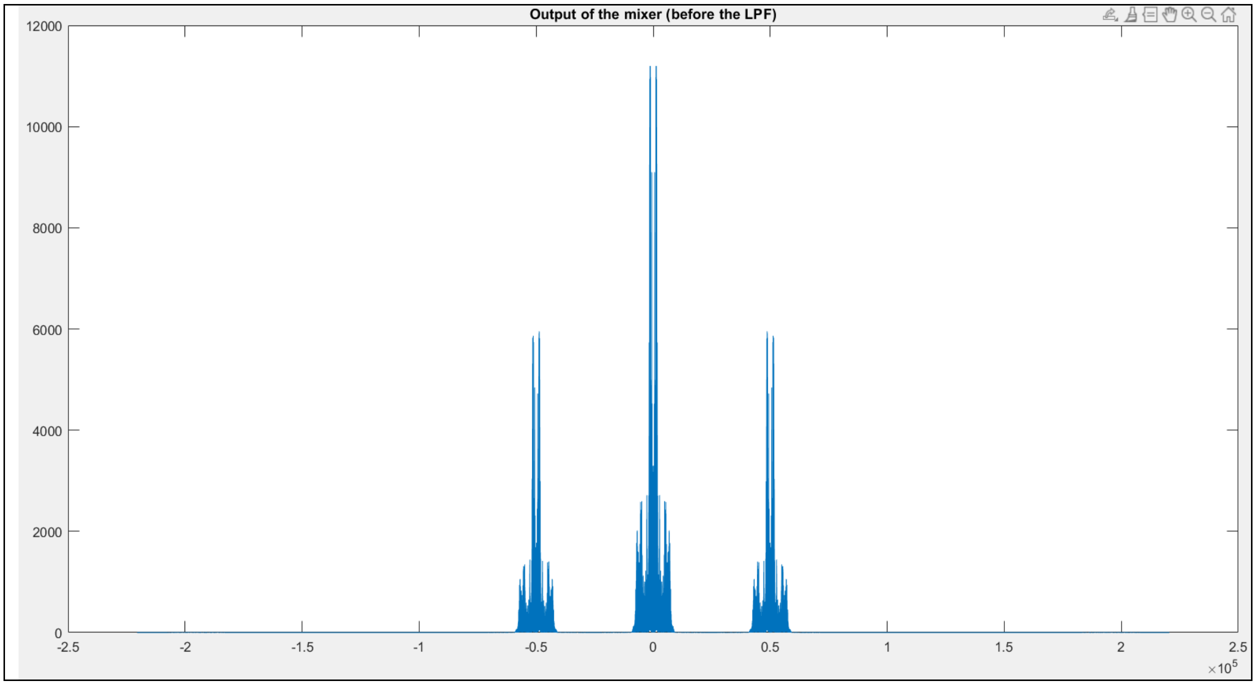
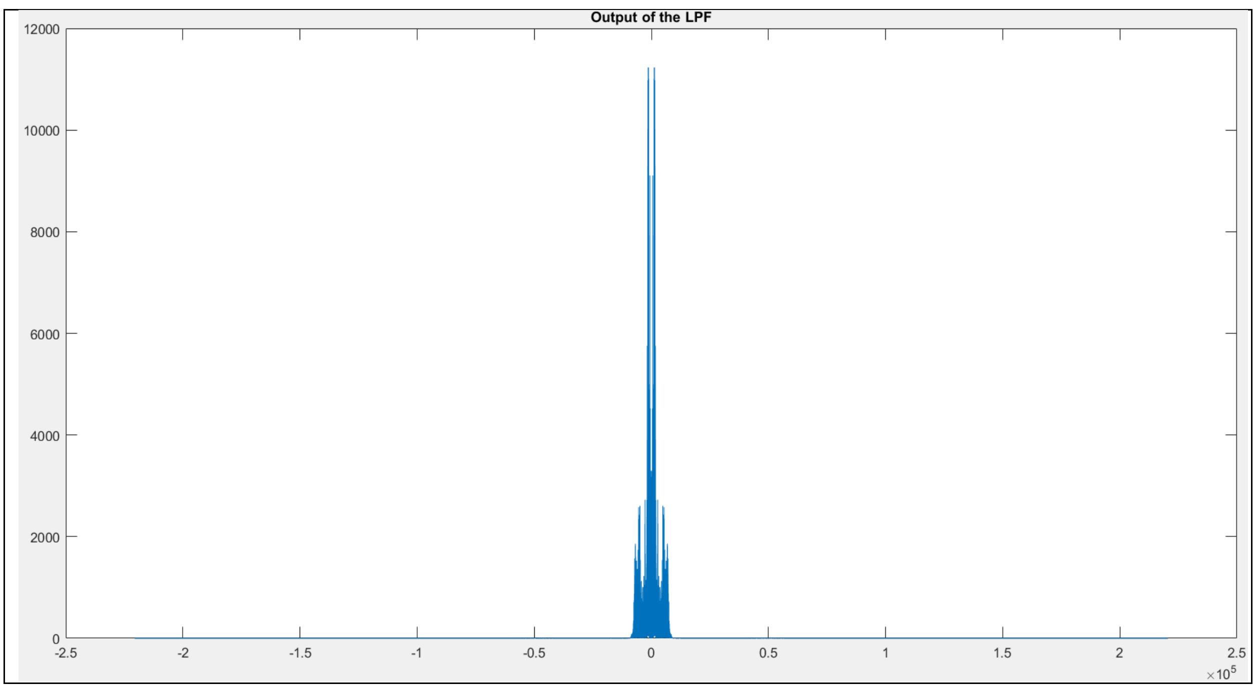


Figure 6: Output of the LPF



5. *Performance evaluation without the RF stage*

The figures

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

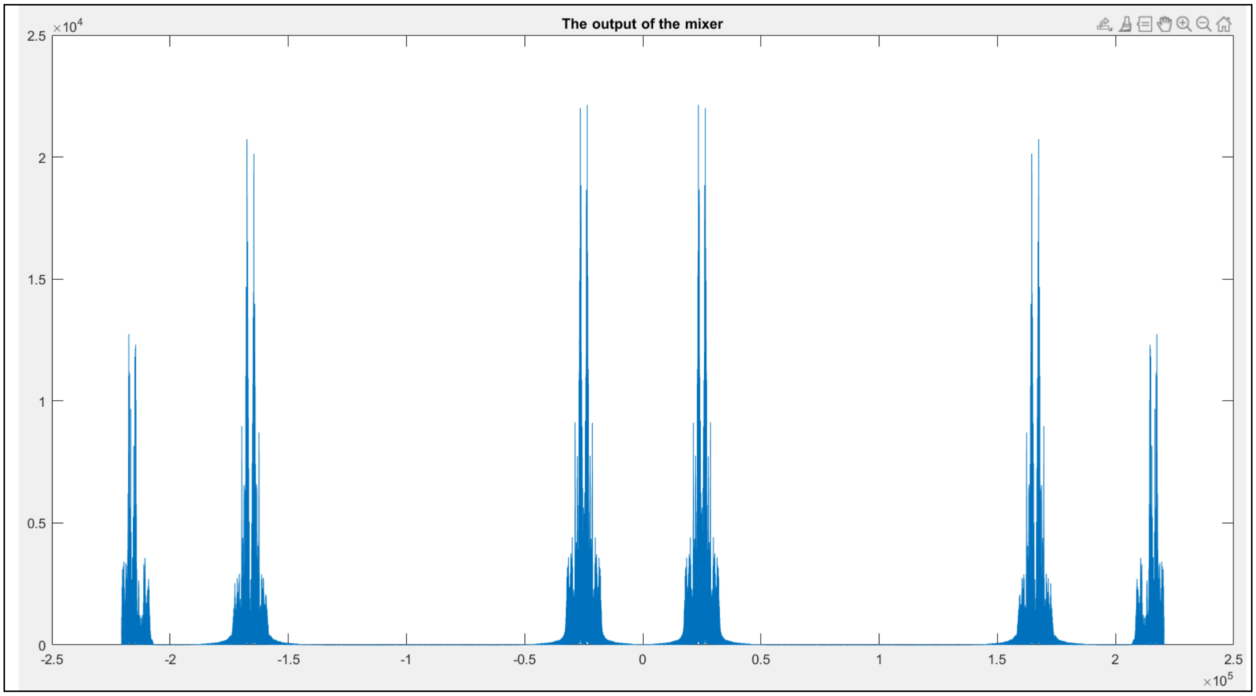


Figure 8: Output of the IF filter (no RF filter)

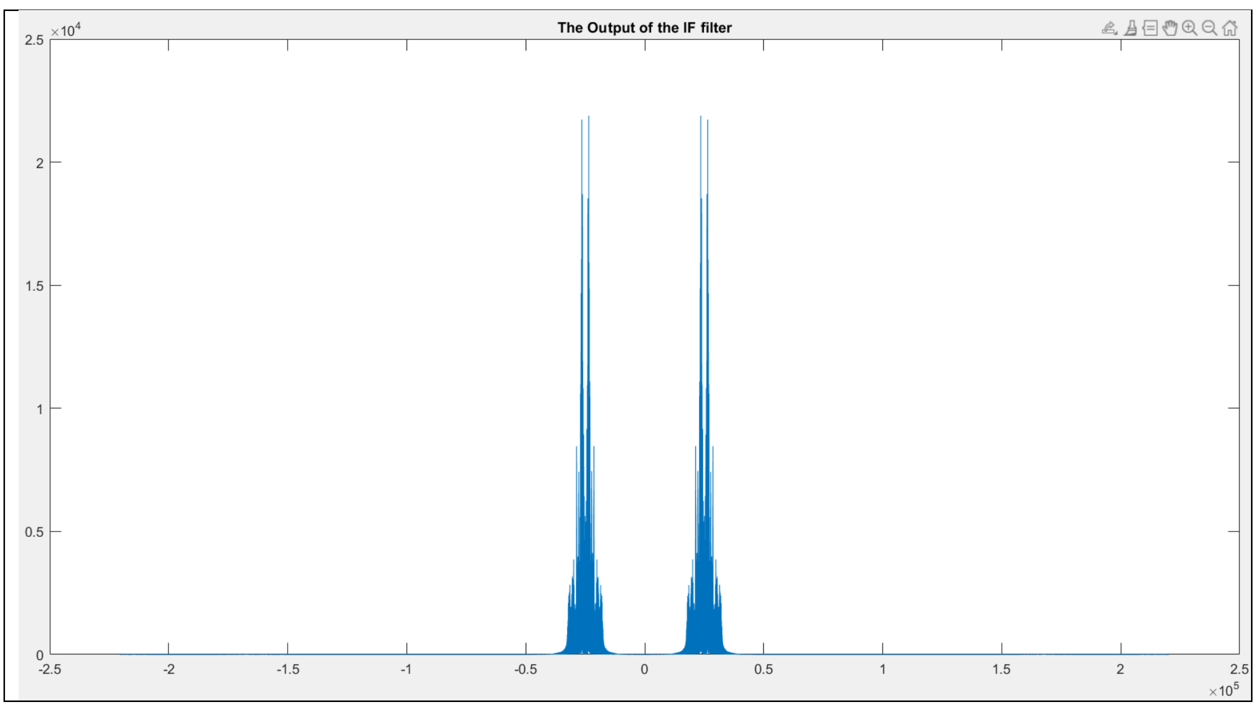


Figure 9: Output of the IF mixer before the LPF (no RF filter)

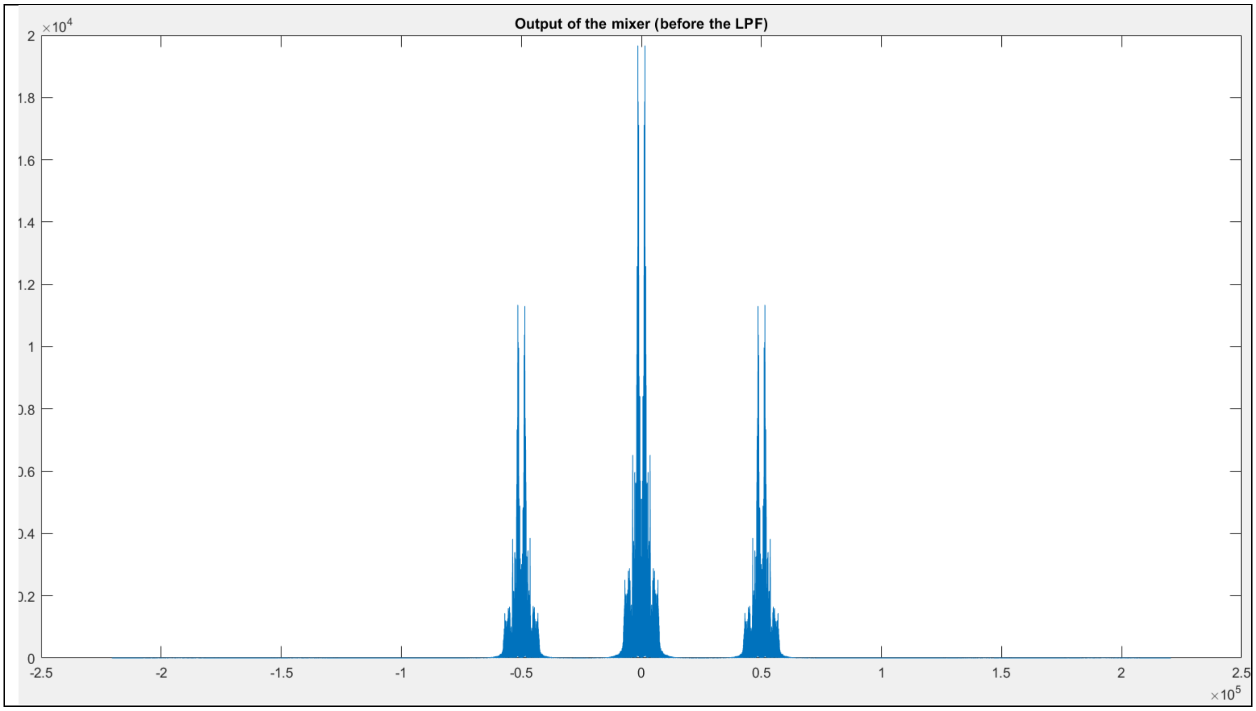
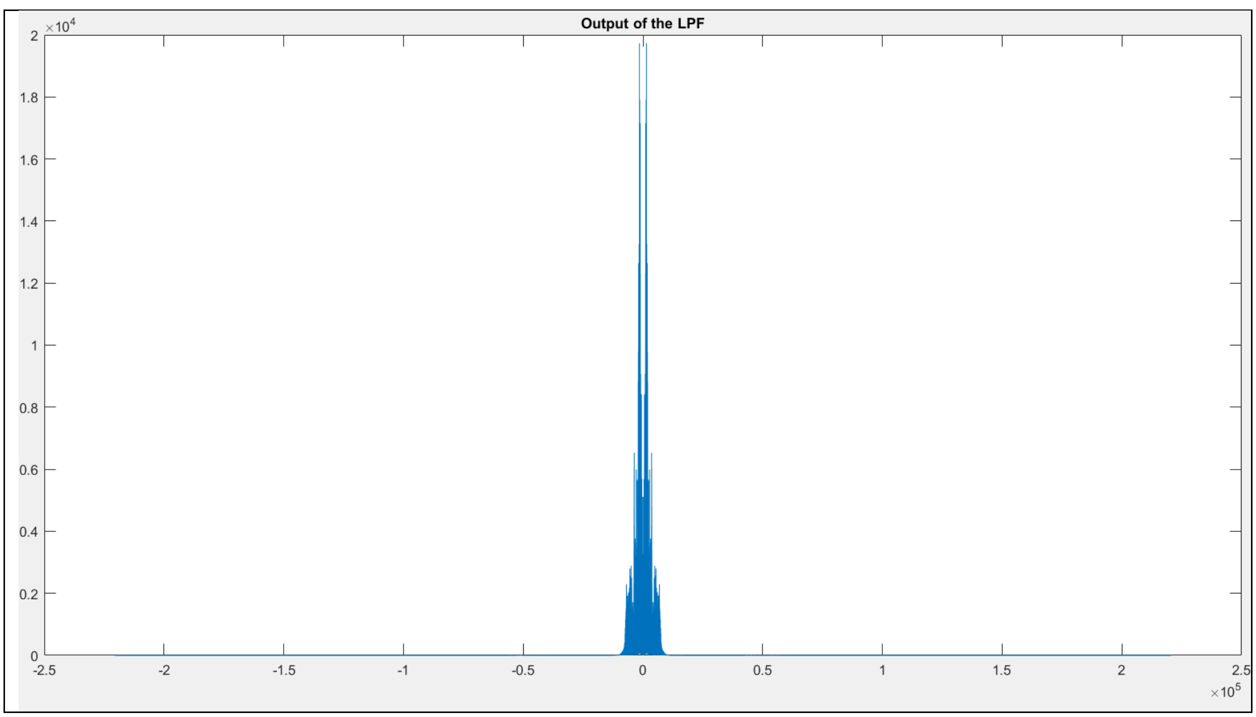


Figure 10: Output of the LPF (no RF filter)



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

In existence of the RF stage: when we apply sound command on the output signal after downsampling it to the original sampling rate, we can listen to the channels well without any noise interferences with it.
In absence of the RF stage: we can listen to the second channel interfering with the first channel because of that the second signal far of the first signal by 2ωIF, and in this case second signal is the image of the first signal. But, that is not happening to the second channel as the first channel is not the image of the second signal.

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 this case: transmitter and receiver have a phase shift at their carrier frequencies, so I can't demodulate the signal well, and in case of 0.1kHz the desired signal is distortion and the sound differences from the last time we listen it, but in case of 1KHz the sound becomes very bad and the signal is more distorted.
But, in the spectrum I note that amplitude of the signal decreases but with small factor, so in the spectrum I can't notice the difference but and the difference is very clear in the sound quality of the channels.
To avoid this problem: we must guarantee that there is no phase shift in the carrier frequency at the Transmitter and receiver.

7. The code

```
clear;
clc;
[sig1,fs1] = audioread("D:\Documents\Third Year First
Semister\Communications\Project\Short_BBCArabic2.wav");
%single channel stream of sig1
sig2 = sig1(:,1) + sig1 (:,2);
[sig3,fs2] = audioread("D:\Documents\Third Year First
Semister\Communications\Project\Short_FM9090.wav");
%single channel stream of sig2
sig4 = sig3(:,1) + sig3 (:,2);
%padding the short signal with zeros, so the two signals have the same
length
sig5 = [sig4;zeros(43008,1)];
L=length(sig5);
k=-L/2:L/2-1;
Y1=fft(sig2,L);
Y2=fft(sig5,L);
subplot(4,2,1);
plot(k*fs1/L,fftshift(abs(Y1))); %bw = 10^4 HZ
title('fft of msg1')
subplot(4,2,2);
plot(k*fs2/L,fftshift(abs(Y2))); %bw = 10^4 HZ
title('fft of msg2')

%....The transmitter....%
message1= interp(sig2,10); %increasing the number of samples of the
modulating signals
message2= interp(sig5,10); %increasing the number of samples of the
modulating signals
fs_new=10*fs1; %fs after using interp function
L_new=10*L; %no of samples after using interp function
f1=100*10^3; %the carrier freq to modualte the first signal
f2=150*10^3; %the carrier freq to modualte the second signal
BW = 10^4; %bandwidth of each siganl
Ts=1/(fs_new);
n1=0:L_new-1;
c1=cos(2*pi*f1*n1'.*Ts);
c2=cos(2*pi*f2*n1'.*Ts);
modulated_message1= 2*message1.*c1;
modulated_message2= message2.*c2;
final_modulated= modulated_message1 + modulated_message2;
Y3=fft(final_modulated,L_new);
k1=-L_new/2:L_new/2-1;
subplot(4,2,3);
plot(k1*fs_new/L_new,fftshift(abs(Y3)));
title('The spectrum of the output of the transmitter');

%.....The RF stage.....%
Fif=25*10^3;
%...f_desired determines which message we want at the Receiver...%
%when choosing 1 then f_desired=f1 and you will Receive the first Message at
receiver..%
%when choosing 2 then f_desired=f2 and you will Receive the second message
at receiver..%
while 1
```

```
prompt= "enter the channel number (1 or 2): ";
ch = input(prompt);
if (ch==1)
f_desired=f1;
break;
elseif (ch==2)
f_desired=f2;
break;
else
fprintf ("wrong channel number please try again \n");
end
end
bpFilt = designfilt('bandpassfir','FilterOrder',35, ...
'CutoffFrequency1',f_desired-Fif,'CutoffFrequency2',f_desired+Fif,...
'SampleRate', fs_new);
message_filtered = filter(bpFilt,final_modulated);
Y4=fft(message_filtered,L_new);
subplot(4,2,4);
plot(k1*fs_new/L_new,abs(fftshift(Y4)));
title('The output of the RF filter (before the mixer)');

%....The mixer stage...%
desired_message= message_filtered.*cos(2*pi*(f_desired+Fif)*n1'*Ts);
Y5=fft(desired_message,L_new);
subplot(4,2,5);
plot(k1*fs_new/L_new,fftshift(abs(Y5)));
title('The output of the mixer');

%....The IF stage....%
bpFilt1 = designfilt('bandpassfir','FilterOrder',50, ...
'CutoffFrequency1',Fif-BW,'CutoffFrequency2',Fif+BW,...
'SampleRate', fs_new);
message_filtered1 = filter(bpFilt1,desired_message);
Y6=fft(message_filtered1,L_new);
subplot(4,2,6);
plot(k1*fs_new/L_new,abs(fftshift(Y6)));
title('The Output of the IF filter');

%....Baseband Detection....%
detected_message = message_filtered1.*cos(2*pi*(Fif)*n1'*Ts);
Y7=fft(detected_message,L_new);
subplot(4,2,7);
plot(k1*fs_new/L_new,abs(fftshift(Y7)));
title('Output of the mixer (before the LPF)');
%...low pass filter to get the Output Message....%
lpFilt = designfilt('lowpassfir', 'filterorder',
200,'CutoffFrequency',9000,'SampleRate', fs_new);
Output_Message = filter(lpFilt,detected_message);
Y8=fft(Output_Message,L_new);
subplot(4,2,8);
plot(k1*fs_new/L_new,abs(fftshift(Y8)));
title('Output of the LPF');
%...listening to the final message....%
Output= downsample(Output_Message,10);
sound(Output,fs1);
```