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: Moaz Mohamed Helmy

Sec: 4   BN: 20 ID: 9203533

Student Name: Ahmed Mohamed Ahmed Shiba

Sec: 1 BN: 24 ID: 9202162

Contents

Table of figures

1. ***The transmitter***

This part contains the following tasks

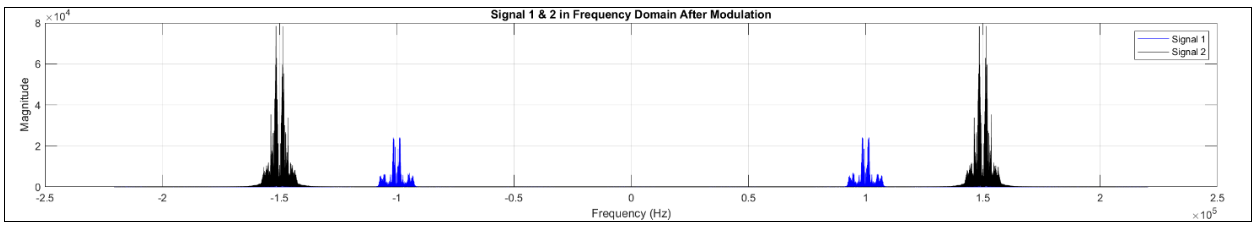
1. Reading monophonic audio signals into MATLAB.
2. Up sampling the audio signals.
3. Modulating the audio signals (each on a separate carrier).
4. Addition of the modulated signals.

Discussion

1. Turning the stereo signals into monophonic was done via adding both columns of each signal into only one column,
2. Both of the used signals were up sampled using the function interp() to adjust Fs to be larger than the carrier frequency, this step is needed to fulfill Nyquist Rate.
3. After that signal 1 & signal 2 were up-converted to frequencies 100KHz & 150KHz respectively through multiplication with carriers (cos(2*pi*100KHz)& cos(2*pi*150kHz)) respectively as well.
4. After completing the up-conversion of both signals, they were added together into one variable as shown in figure one.

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

This stage is mandatory to filter out any signal other than the desired signal. This helps with:

1. Cancelling the effect of the leaked power from other signals in the near frequencies before entering any amplifiers.
2. Cancel out the image signals that might exist at double the intermediate frequency, which can significantly distort the desired signal after down converting the signal from the intermediate frequency to the baseband 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)

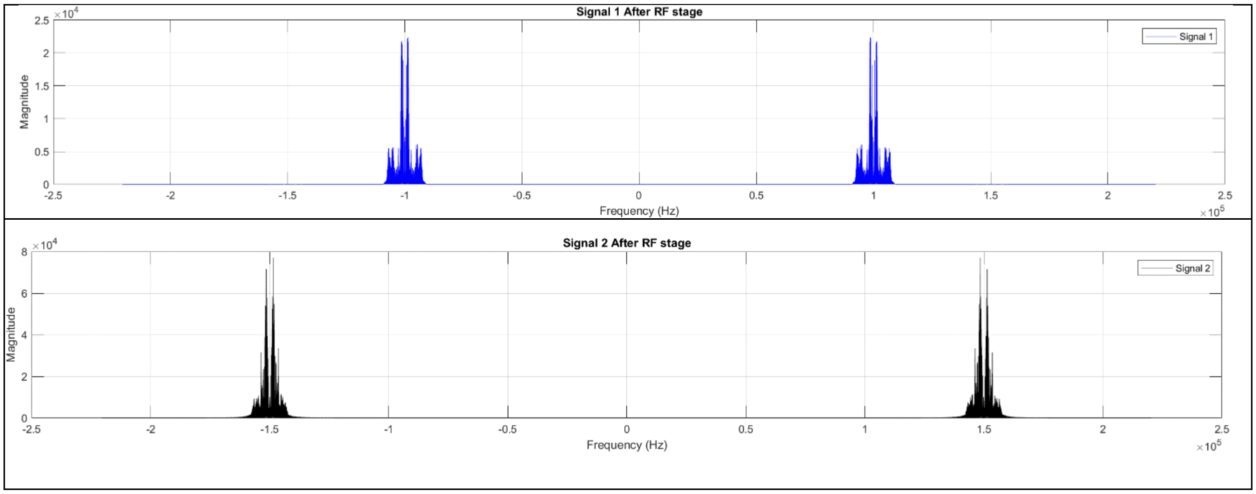
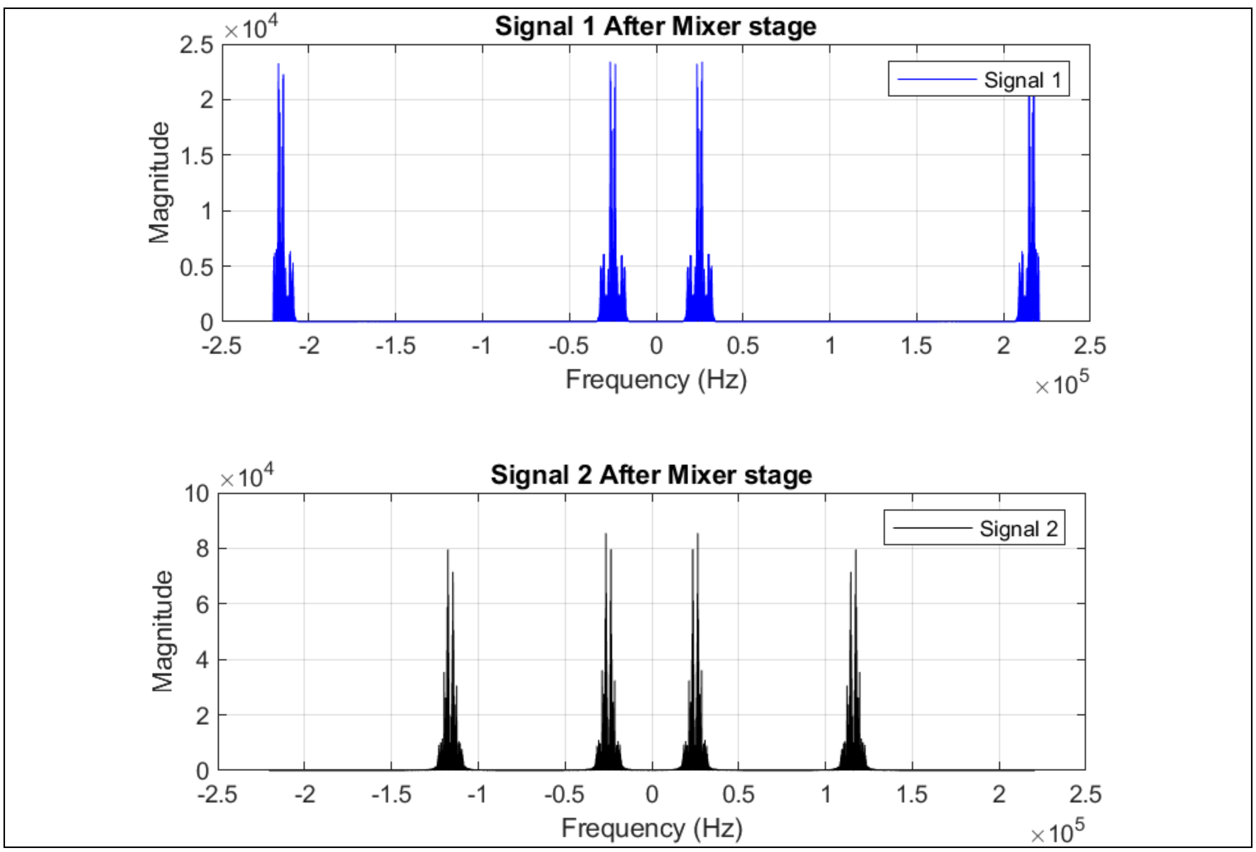


Figure 3: The output of the mixer



Note on figure 3 (Signal 2):

- After multiplying signal 2 by a carrier with frequency equal to the intermediate frequency which is 25KHz, it should have been shifted to 25KHz, -25KHz, 325KHz, and -325KHz. However, it was observed on the plot of signal 2 in figure 3 that the signal has been shifted to 115KHz and -115KHz instead of 325KHz and -325KHz. This was done because of the fact that MATLAB always attempts to draw all of the resulting signal on the same plot, and it faced the issue that the scale of frequencies of the plot is limited from -205KHz to 205KHz as was specified in the plot function. Hence, MATLAB used the margins of this scale as a mirror to reflect any components outside of the range inside the range and with the same difference between this component and the limit of the scale which is equal to (325KHz – 205KHz). That's why there are components setting at 115KHz and -115KHz, as (115KHz = 205KHz – the difference), .i.e., (205KHz – (325KHz – 205KHz)). Similarly for the component at -115KHz.

3. ***The IF stage***

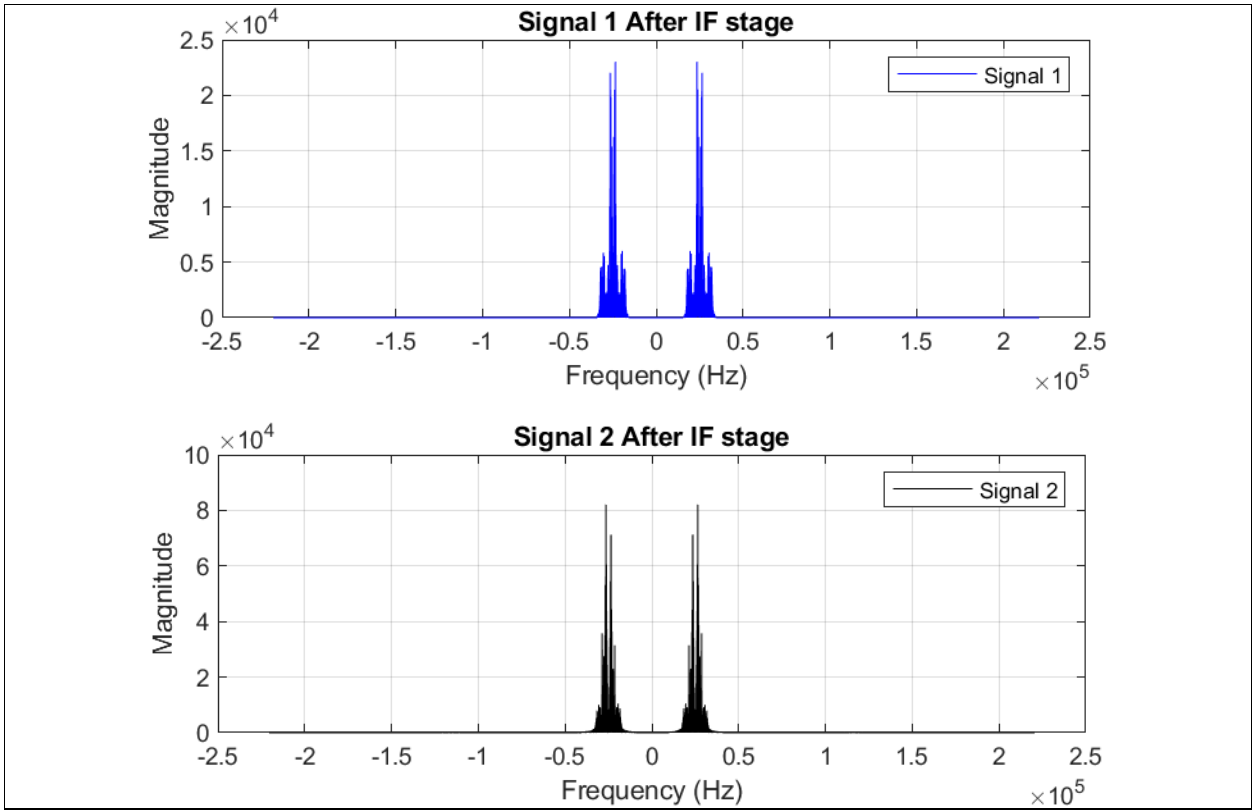
This part addresses the IF filter.

Discussion

```
This stage is responsible for transferring the desired signal to an
intermediate frequency before down converting it to the baseband frequency
in the digital domain. It has two main advantages over the ordinary homodyne
receiver. Those advantages are:
   1. It significantly helps with cancelling any flicker noise that exists
      at the low frequencies.
   2. It also helps with rejecting any dc components in the signal which
      also exists at the zero frequency.
After shifting the desired signal to the intermediate frequency, it's
filtered out using a band-pass filter centered at the IF, then retrieved to
the baseband frequency in the digital domain through multiplying it by a
carrier with frequency equal to the intermediate frequency, then passing the
output through a low- pass filter.
```

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

```
This stage is simply responsible for returning the desired signal to the
baseband frequency via multiplying it by a carrier with the same frequency
as the intermediate frequency. The output of this multiplication is then
passed through a low-pass filter to retrieve the received signal.
```

The figures

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

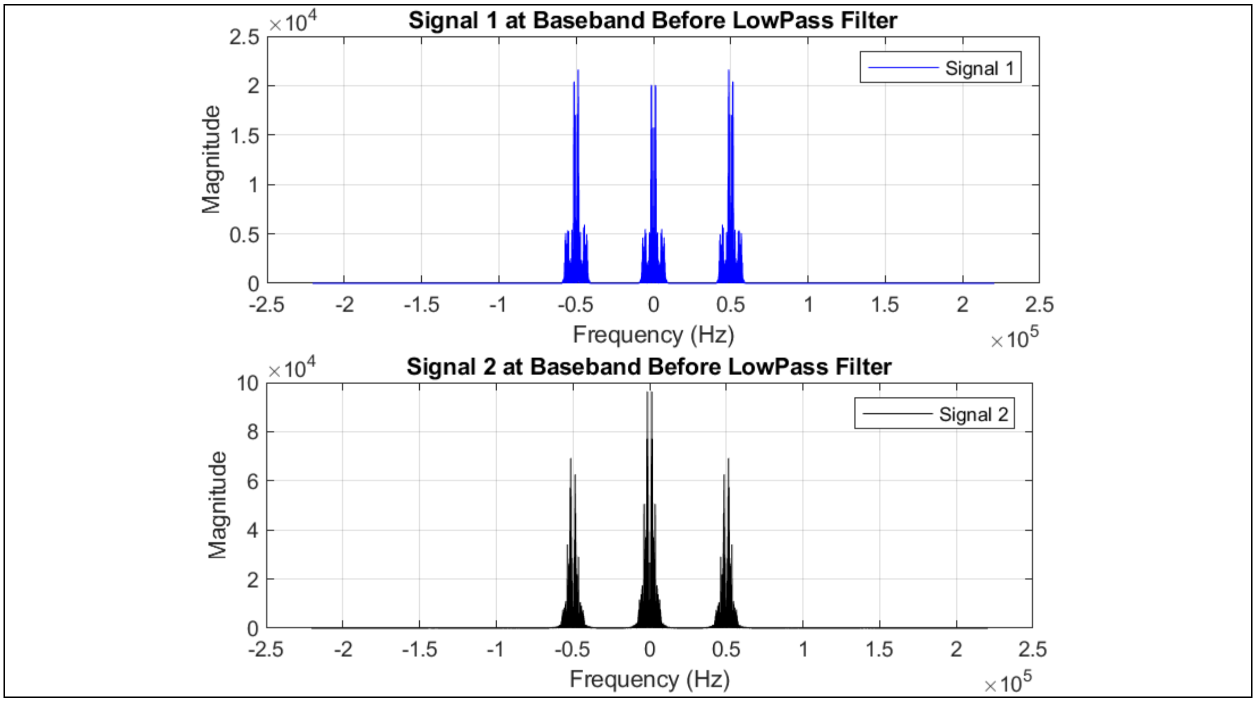
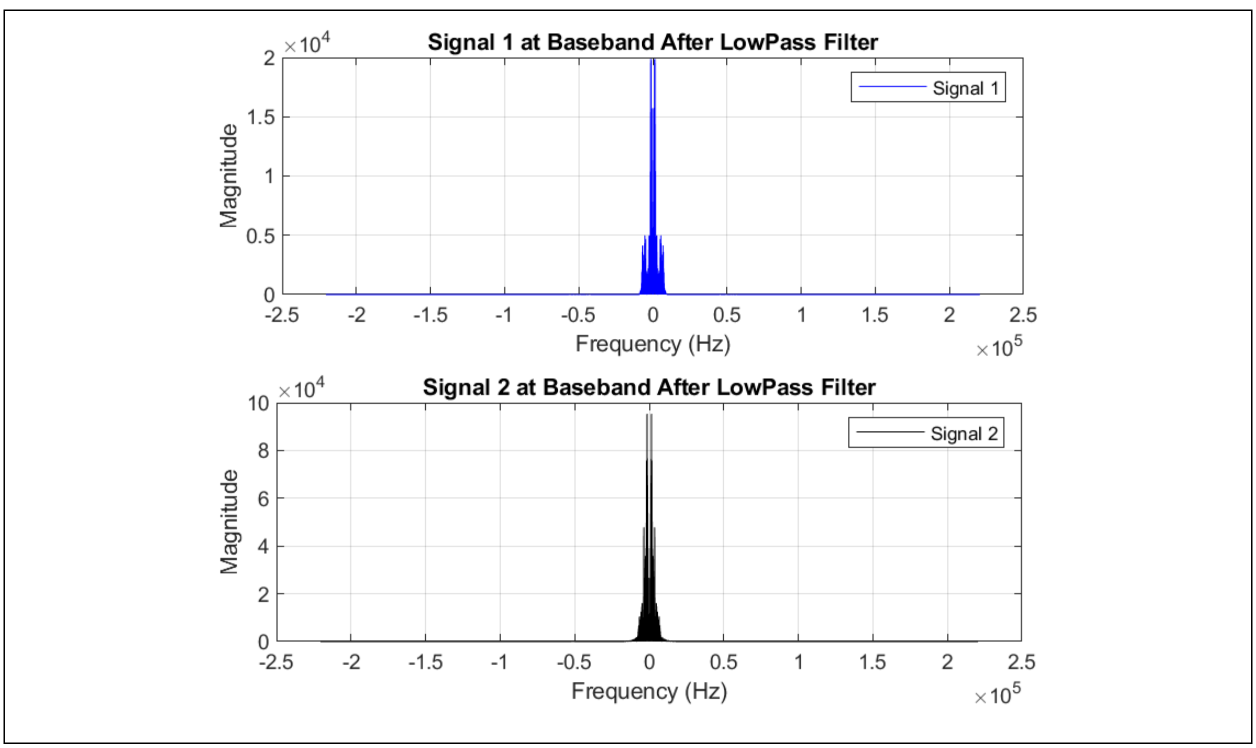


Figure 6: Output of the LPF



5. *Performance evaluation without the RF stage*

The figures

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

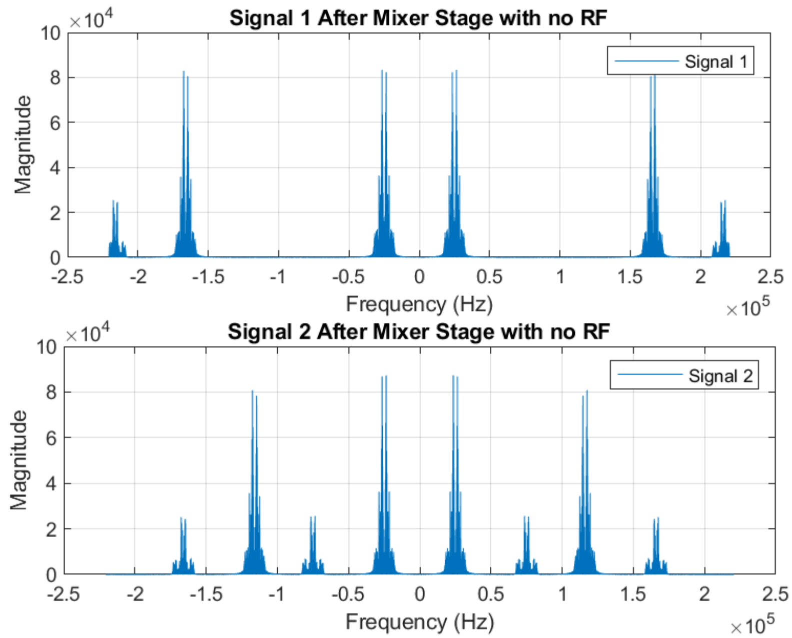


Figure 8: Output of the IF filter (no RF filter)

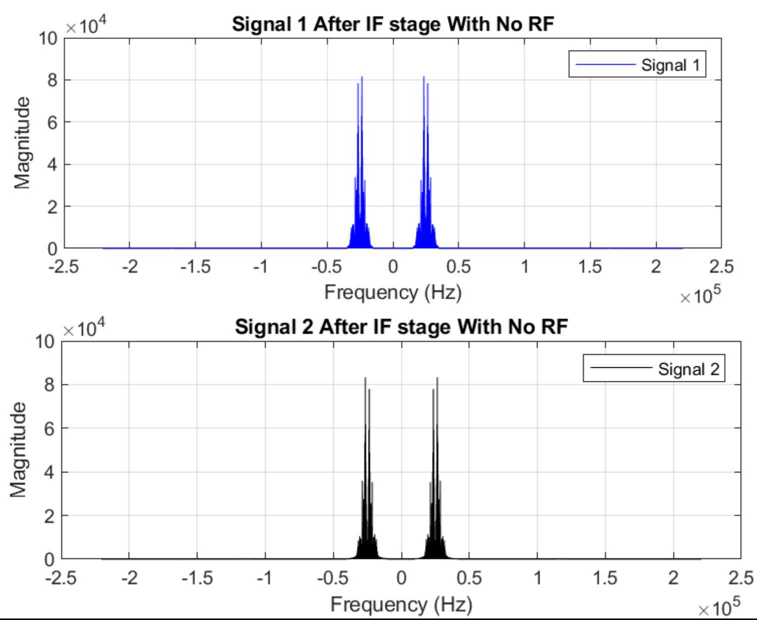


Figure 9: Output of the IF mixer before the LPF (no RF filter)

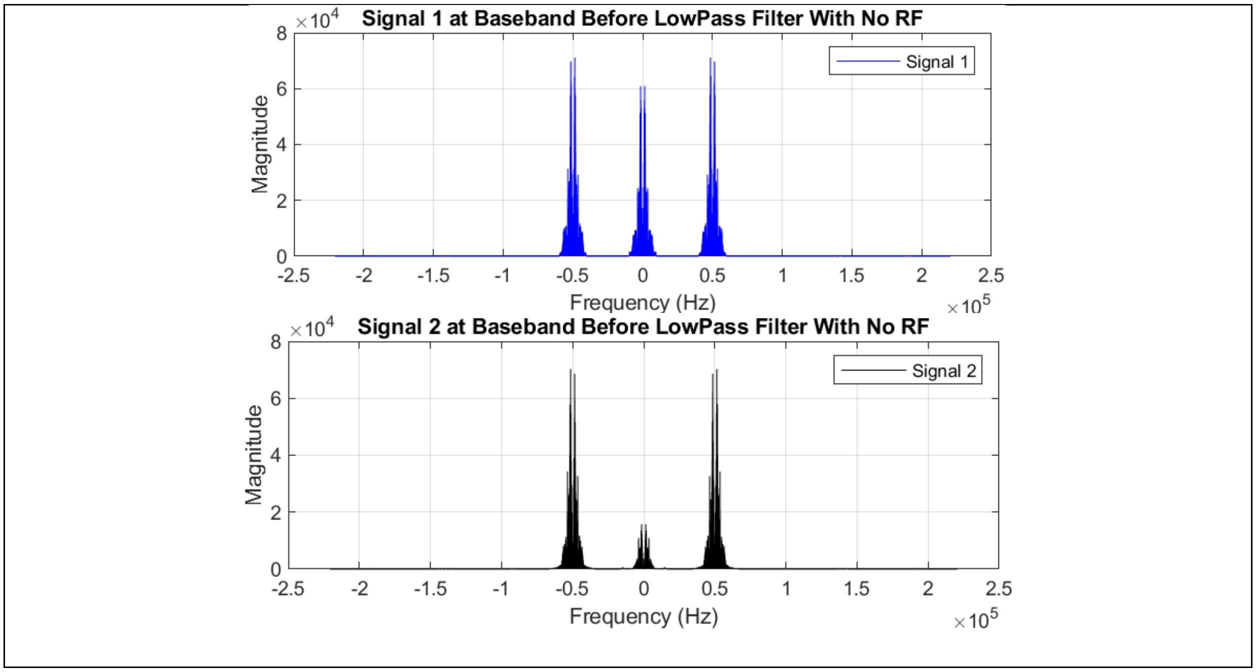
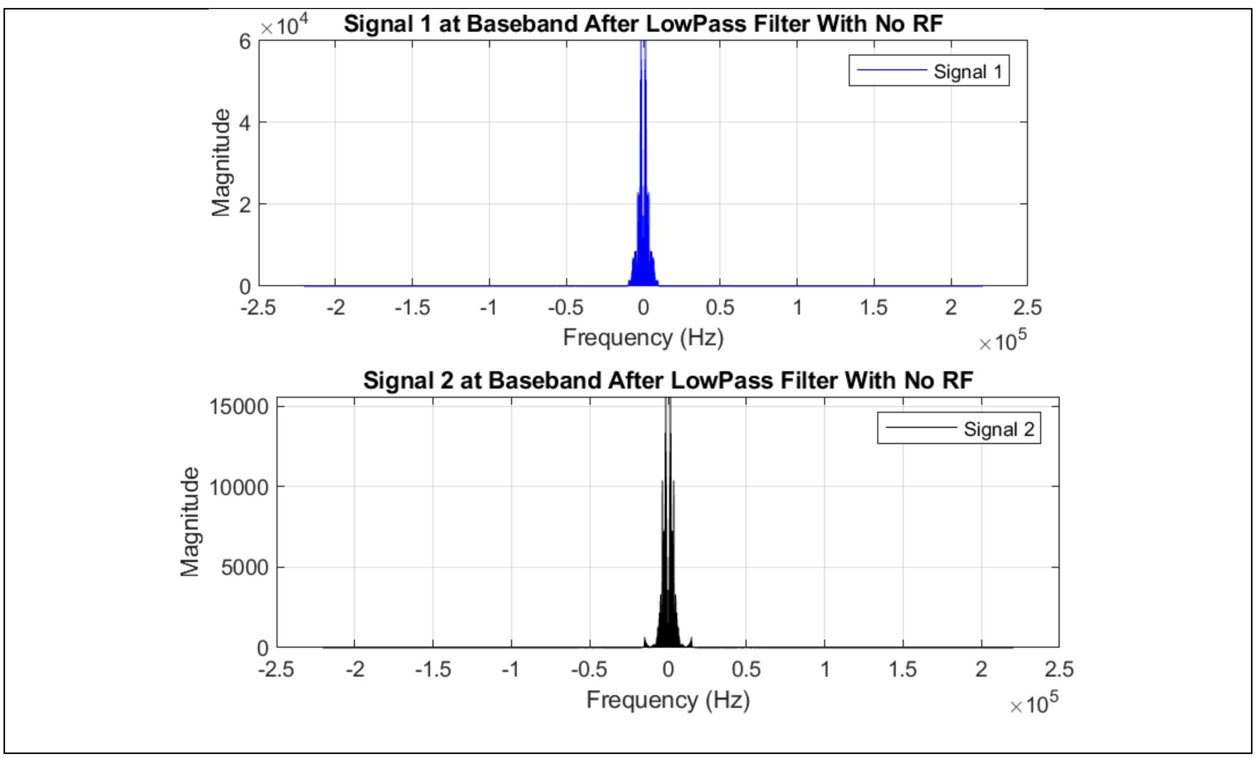


Figure 10: Output of the LPF (no RF filter)



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

With RF stage:
Both signals were retrieved successfully without any distortion as the potential distortion would have been caused by the image signal at double the intermediate frequency, but this image signal was rejected after passing the modulated signal through the bandpass filter in the RF stage.
Without RF stage:
Without the RF stage, the image signal was not rejected before the mixer stage which will shift the signal and its image to the intermediate frequency. This image signal will be superimposed on the desired signal, which will highly distort the retrieved signal as the heard audio will be both signals playing at the same time.

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

Generally, the addition of positive offset in the mixer stage will cause the signal to be shifted a little to the right of the IF with difference equal to the value of the offset. Consequently, if this offset is large as in the case when it's equal to 1KHz, most of the signal high frequency components will be rejected in the IF stage, causing significant amount of distortion to the signal. On the other hand, if the offset is small (0.1KHz), the signal will also be distorted but with less severity than case 1, hence, it'll still be recoverable and audible.

Important Note:

The filters used during simulation were designed using MATLAB built-in "Filter Designer", and were exported as .mat files, then loaded into the script file. Thus, to run the script successfully and without any inconvenience, the script + the filter .mat files are required. So, the full project MATLAB code files have been uploaded to this drive folder where you can download and test it:

https://drive.google.com/drive/folders/1CRX3pBNp13zSvh31Q8j7UBWSUt9D_rKq?usp=sharing

7. The code

```
Please insert the code of your project here. A good code needs to be
readable and understandable. Use clear names for variables. Use comments as
much as you want.
close all;
clear all;
% load all filter files first
load('BandPass100.mat');
load('BandPass150.mat');
load('BandPass25.mat');
load('LowPass25.mat');
load('BandPass252.mat');
%% Reading The Signals
[Signal1,~]=audioread('Short_BBCArabic2.wav');
[Signal2,~]=audioread('Short_FM9090.wav');
[Signal3,~]=audioread('Short_QuranPalestine.wav');
[Signal4,~]=audioread('Short_RussianVoice.wav');
[Signal5,FS]=audioread('Short_SkyNewsArabia.wav');

%% Padding Short Signals With Zeros
Signal2=padarray(Signal2,(length(Signal1)-length(Signal2)),0,'post');
Signal3=padarray(Signal3,(length(Signal1)-length(Signal3)),0,'post');
Signal4=padarray(Signal4,(length(Signal1)-length(Signal4)),0,'post');
Signal5=padarray(Signal5,(length(Signal1)-length(Signal5)),0,'post');

%% Monophonic Signals
Signal1=Signal1(:,1)+Signal1(:,2);
Signal2=Signal2(:,1)+Signal2(:,2);
Signal3=Signal3(:,1)+Signal3(:,2);
Signal4=Signal4(:,1)+Signal4(:,2);
Signal5=Signal5(:,1)+Signal5(:,2);

%% Monophonic Signals Befor Modulation
%% Signal 1 Before Modulation
%Signal 1 Time Domain
T=0:1/(FS):((length(Signal1)-1)/(FS));
figure;
subplot(2,1,1),plot(T,Signal1,'b');
grid on;
xlabel("Time (sec)");
ylabel("Amplitude");
legend("Signal 1");
title("Time Domain Plot of Signal 1 Before Modulation");

%Signal 1 Frequency Domain
%Plot Signal 1 in Frequency domain
Signal1_len=length(Signal1);
Signal1_freq=fft(Signal1,Signal1_len);
F=(-Signal1_len/2:Signal1_len/2-1).*(FS/Signal1_len);
subplot(2,1,2),plot(F,abs(fftshift(Signal1_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
```

```
legend("Signal 1");
title("Frequency Domain Plot of Signal 1 Before Modulation");

%% Signal 2
%Signal 2 Time Domain
T=0:1/FS:((length(Signal2)-1)/FS);
figure;
subplot(2,1,1),plot(T,Signal2,'k');
grid on;
xlabel("Time (sec)");
ylabel("Amplitude");
legend("Signal 2");
title("Time Domain Plot of Signal 2 Before Modulation");

%Signal 2 Frequency Domain
%Plot Signal 2 in Frequency domain
Signal2_len=length(Signal2);
Signal2_freq=fft(Signal2,Signal2_len);
F=(-Signal2_len/2:Signal2_len/2-1).*(FS/Signal2_len);
subplot(2,1,2),plot(F,abs(fftshift(Signal2_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Frequency Domain Plot of Signal 2 Before Modulation");

%% AM Modulator
%% Signal 1 After Modulation
Fc=100e3;

%Increase the samples 10 times using interpolation
Signal1_interp=interp(Signal1,10);
T=0:1/(10*FS):((length(Signal1_interp)-1)/(10*FS));
Carrier1=2*cos(2*pi*Fc*T);
Signal1_mod=Signal1_interp.*Carrier1';
figure;
subplot(2,1,1),plot(T,Signal1_mod,'b');
grid on;
xlabel("Time (sec)");
ylabel("Amplitude");
legend("Signal 1");
title("Time Domain Plot of Signal 1 After Modulation");

Signal1_len=length(Signal1_mod);
Signal1_mod_freq=fft(Signal1_mod,Signal1_len);
F=(-Signal1_len/2:Signal1_len/2-1).*(10*FS/Signal1_len);
subplot(2,1,2),plot(F,abs(fftshift(Signal1_mod_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1");
title("Frequency Domain Plot of Signal 1 After Modulation");

%% Signal 2 After Modulation
Fc=150e3;
```

```
%Increase the samples 10 times using interpolation
Signal2_interp=interp(Signal2,10);

T=0:1/(10*FS):((length(Signal2_interp)-1)/(10*FS));
Carrier2=2*cos(2*pi*Fc*T);
Signal2_mod=Signal2_interp.*Carrier2';
figure;
subplot(2,1,1),plot(T,Signal2_mod,'k');
grid on;
xlabel("Time (sec)");
ylabel("Amplitude");
legend("Signal 2");
title("Time Domain Plot of Signal 2 After Modulation");

Signal2_len=length(Signal2_mod);
Signal2_mod_freq=fft(Signal2_mod,Signal2_len);
F=(-Signal2_len/2:Signal2_len/2-1).*(10*FS/Signal2_len);
subplot(2,1,2),plot(F,abs(fftshift(Signal2_mod_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Frequency Domain Plot of Signal 2 After Modulation");

%% RF Stage
%Combining Signal 1 & Signal 2
FDM=Signal1_mod+Signal2_mod;
FDM_len=length(FDM);
F=(-FDM_len/2:FDM_len/2-1).*(10*FS/FDM_len);
figure;
subplot(3,1,1),plot(F,abs(fftshift(Signal1_mod_freq)),'b',F,abs(fftshift(Sig
nal2_mod_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend('Signal 1','Signal 2');
title("Signal 1 & 2 in Frequency Domain After Modulation");

%% No RF case
% Defining New Variables
NRF_F=F;



%% Bandpass filter in RF stage
% Filtering Signal 1 at 100KHz
RF_Signal1_filtered=filter(BandPass100,FDM);
RF_Signal1_filtered_freq=fft(RF_Signal1_filtered,length(RF_Signal1_filtered)
);
subplot(3,1,2),plot(F,abs(fftshift(RF_Signal1_filtered_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1");
title("Signal 1 After RF stage");
```

```
%Filtering Signal 2 at 150KHz
RF_Signal2_filtered=filter(BandPass150,FDM);
RF_Signal2_filtered_freq=fft(RF_Signal2_filtered,length(RF_Signal2_filtered)
);
subplot(3,1,3),plot(F,abs(fftshift(RF_Signal2_filtered_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Signal 2 After RF stage");

%% Mixer Stage (Signal 1)
%down-conversion of Signal 1 from 100K to IF=25KHz
IF=25e3;
Fc=100e3;
T=0:1/(10*FS):((length(RF_Signal1_filtered)-1)/(10*FS));
IF_Carrier1=2*cos(2*pi*(Fc+IF)*T);
IF_Signal1_demod=RF_Signal1_filtered.*IF_Carrier1';
IF_Signal1_demod_freq=fft(IF_Signal1_demod,length(IF_Signal1_demod));
F=(-length(IF_Signal1_demod)/2:length(IF_Signal1_demod)/2-
1).*(10*FS/length(IF_Signal1_demod));
figure;
subplot(2,1,1),plot(F,abs(fftshift(IF_Signal1_demod_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1");
title("Signal 1 After Mixer stage");

%% IF Stage (Signal 1)
%Filtering signal 1 at IF=25KHz
IF_Signal1_filtered=filter(BandPass25,IF_Signal1_demod);
IF_Signal1_filtered_freq=fft(IF_Signal1_filtered);
F=(-length(IF_Signal1_filtered)/2:length(IF_Signal1_filtered)/2-
1).*(10*FS/length(IF_Signal1_filtered));
subplot(2,1,2),plot(F,abs(fftshift(IF_Signal1_filtered_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1");
title("Signal 1 After IF stage");

%% Baseband Detection Stage (Signal 1)
Fc=25e3;
T=0:1/(10*FS):((length(IF_Signal1_filtered)-1)/(10*FS));
BBD_Carrier1=2*cos(2*pi*Fc*T);
BBD_Signal1_demod=IF_Signal1_filtered.*BBD_Carrier1';
BBD_Signal1_demod_freq=fft(BBD_Signal1_demod,length(BBD_Signal1_demod));
F=(-length(BBD_Signal1_demod)/2:length(BBD_Signal1_demod)/2-
1).*(10*FS/length(BBD_Signal1_demod));
figure;
subplot(2,1,1),plot(F,abs(fftshift(BBD_Signal1_demod_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
```

```
legend("Signal 1");
title("Signal 1 at Baseband Before LowPass Filter");

%Signal 1 After Lowpass filter
BBD_Signal1_filtered=filter(LowPass25,BBD_Signal1_demod);
BBD_Signal1_filtered_freq=fft(BBD_Signal1_filtered,length(BBD_Signal1_filter
ed));
subplot(2,1,2),plot(F,abs(fftshift(BBD_Signal1_filtered_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1");
title("Signal 1 at Baseband After LowPass Filter");

%% Retrieval of Signal 1 at the Receiver
Signal1_Retrieved=BBD_Signal1_filtered;

%downsampling of signal 1
Signal1_Retrieved= downsample(Signal1_Retrieved,10);

%Plot of Signal 1 after downsampling in Time
T=0:1/(FS):((length(Signal1_Retrieved)-1)/(FS));
figure;
subplot(2,1,1),plot(T,Signal1_Retrieved,'b');
grid on;
xlabel("Time (sec)");
ylabel("Amplitude");
legend("Signal 1");
title("Time Domain of Signal 1 After LowPass Filter & Downsampling");

%Plot of Signal 1 after downsampling in Frequency
F=(-(length(Signal1_Retrieved))/2:(length(Signal1_Retrieved)/2-
1)).*(FS/(length(Signal1_Retrieved)));
Signal1_Retrieved_freq=fft(Signal1_Retrieved);
subplot(2,1,2),plot(F,abs(fftshift(Signal1_Retrieved_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1");
title("Frequency Domain of Signal 1 After LowPass Filter & Downsampling");
%sound(Signal1_Retrieved,FS);

%% Mixer Stage (Signal 2)
%down-conversion of Signal 2 from 150K to IF=25KHz
IF=25e3;
Fc=150e3;
T=0:1/(10*FS):((length(RF_Signal2_filtered)-1)/(10*FS));
IF_Carrier2=2*cos(2*pi*(Fc+IF)*T);
IF_Signal2_demod=RF_Signal2_filtered.*IF_Carrier2';
IF_Signal2_demod_freq=fft(IF_Signal2_demod,length(IF_Signal2_demod));
F=(-length(IF_Signal2_demod)/2:length(IF_Signal2_demod)/2-
1).*(10*FS/length(IF_Signal2_demod));
figure;
subplot(2,1,1),plot(F,abs(fftshift(IF_Signal2_demod_freq)),'k');
grid on;
```

```
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Signal 2 After Mixer stage");

%% IF Stage (Signal 2)
%Filtering signal 2 at IF=25KHz
IF_Signal2_filtered=filter(BandPass252,IF_Signal2_demod);
IF_Signal2_filtered_freq=fft(IF_Signal2_filtered);
F=(-length(IF_Signal2_filtered)/2:length(IF_Signal2_filtered)/2-
1).*(10*FS/length(IF_Signal2_filtered));
subplot(2,1,2),plot(F,abs(fftshift(IF_Signal2_filtered_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Signal 2 After IF stage");

%% Baseband Detection Stage (Signal 2)
Fc=25e3;
T=0:1/(10*FS):((length(IF_Signal2_filtered)-1)/(10*FS));
BBD_Carrier2=2*cos(2*pi*Fc*T);
BBD_Signal2_demod=IF_Signal2_filtered.*BBD_Carrier2';
BBD_Signal2_demod_freq=fft(BBD_Signal2_demod,length(BBD_Signal2_demod));
F=(-length(BBD_Signal2_demod)/2:length(BBD_Signal2_demod)/2-
1).*(10*FS/length(BBD_Signal2_demod));
figure;
subplot(2,1,1),plot(F,abs(fftshift(BBD_Signal2_demod_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Signal 2 at Baseband Before LowPass Filter");

%Signal 2 After Lowpass filter
BBD_Signal2_filtered=filter(LowPass25,BBD_Signal2_demod);
BBD_Signal2_filtered_freq=fft(BBD_Signal2_filtered,length(BBD_Signal2_filter
ed));
subplot(2,1,2),plot(F,abs(fftshift(BBD_Signal2_filtered_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Signal 2 at Baseband After LowPass Filter");

%% Retrieval of Signal 2 at the Receiver
Signal2_Retrieved=BBD_Signal2_filtered;

%downsampling of signal 2
Signal2_Retrieved= downsample(Signal2_Retrieved,10);

%Plot of Signal 2 after downsampling in Time
T=0:1/(FS):((length(Signal2_Retrieved)-1)/(FS));
figure;
subplot(2,1,1),plot(T,Signal2_Retrieved,'k');
```

```
grid on;
xlabel("Time (sec)");
ylabel("Amplitude");
legend("Signal 2");
title("Time Domain of Signal 2 After LowPass Filter & Downsampling");

%Plot of Signal 2 after downsampling in Frequency
F=(-(length(Signal2_Retrieved))/2:(length(Signal2_Retrieved)/2-
1)).*(FS/(length(Signal2_Retrieved)));
Signal2_Retrieved_freq=fft(Signal2_Retrieved);
subplot(2,1,2),plot(F,abs(fftshift(Signal2_Retrieved_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Frequency Domain of Signal 2 After LowPass Filter & Downsampling");
%sound(Signal2_Retrieved,FS);

%% NO RF
%% Mixer Stage with no RF (Signal 1)

Fc=100e3;
IF=25e3;
T=0:1/(10*FS):((length(FDM)-1)/(10*FS));
NRF_Carrier1=2*cos(2*pi*(Fc+IF)*T);
NRF_Signal1_demod=FDM.*NRF_Carrier1';
NRF_Signal1_demod_freq=fft(NRF_Signal1_demod,length(NRF_Signal1_demod));
F=(-FDM_len/2:FDM_len/2-1).*(10*FS/FDM_len);

figure;
subplot(2,1,1),plot(F,abs(fftshift(NRF_Signal1_demod_freq)));
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1");
title("Signal 1 After Mixer Stage with no RF");

%% Mixer Stage with no RF (Signal 2)
Fc=150e3;
IF=25e3;
T=0:1/(10*FS):((length(FDM)-1)/(10*FS));
NRF_Carrier2=2*cos(2*pi*(Fc+IF)*T);
NRF_Signal2_demod=FDM.*NRF_Carrier2';
NRF_Signal2_demod_freq=fft(NRF_Signal2_demod,length(NRF_Signal2_demod));
F=(-FDM_len/2:FDM_len/2-1).*(10*FS/FDM_len);

subplot(2,1,2),plot(F,abs(fftshift(NRF_Signal2_demod_freq)));
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Signal 2 After Mixer Stage with no RF");
%% IF Stage With No RF (Signal 1)
%Filtering signal 1 at IF=25KHz
NRF_IF_Signal1_filtered=filter(BandPass25,NRF_Signal1_demod);
```

```
NRF_IF_Signal1_filtered_freq=fft(NRF_IF_Signal1_filtered);
F=(-length(NRF_IF_Signal1_filtered)/2:length(NRF_IF_Signal1_filtered)/2-
1).*(10*FS/length(NRF_IF_Signal1_filtered));

figure;
subplot(2,1,1),plot(F,abs(fftshift(NRF_IF_Signal1_filtered_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1");
title("Signal 1 After IF stage With No RF");
%% IF Stage With No RF (Signal 2)
%Filtering signal 2 at IF=25KHz
NRF_IF_Signal2_filtered=filter(BandPass252,NRF_Signal2_demod);
NRF_IF_Signal2_filtered_freq=fft(NRF_IF_Signal2_filtered);
F=(-length(NRF_IF_Signal2_filtered)/2:length(NRF_IF_Signal2_filtered)/2-
1).*(10*FS/length(NRF_IF_Signal2_filtered));

subplot(2,1,2),plot(F,abs(fftshift(NRF_IF_Signal2_filtered_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Signal 2 After IF stage With No RF");

%% Baseband Detection Stage with No RF (Signal 1)
Fc=25e3;
T=0:1/(10*FS):((length(NRF_IF_Signal1_filtered)-1)/(10*FS));
NRF_BBD_Carrier1=2*cos(2*pi*Fc*T);
NRF_BBD_Signal1_demod=NRF_IF_Signal1_filtered.*NRF_BBD_Carrier1';
NRF_BBD_Signal1_demod_freq=fft(NRF_BBD_Signal1_demod,length(NRF_BBD_Signal1_
demod));
F=(-length(NRF_BBD_Signal1_demod)/2:length(NRF_BBD_Signal1_demod)/2-
1).*(10*FS/length(NRF_BBD_Signal1_demod));
figure;
subplot(2,1,1),plot(F,abs(fftshift(NRF_BBD_Signal1_demod_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1");
title("Signal 1 at Baseband Before LowPass Filter With No RF");

%Signal 1 After Lowpass filter with no Rf
NRF_BBD_Signal1_filtered=filter(LowPass25,NRF_BBD_Signal1_demod);
NRF_BBD_Signal1_filtered_freq=fft(NRF_BBD_Signal1_filtered,length(NRF_BBD_Si
gnal1_filtered));
subplot(2,1,2),plot(F,abs(fftshift(NRF_BBD_Signal1_filtered_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1");
title("Signal 1 at Baseband After LowPass Filter With No RF");

%% Baseband Detection Stage with No RF (Signal 2)
Fc=25e3;
T=0:1/(10*FS):((length(NRF_IF_Signal2_filtered)-1)/(10*FS));
```

```
NRF_BBD_Carrier2=2*cos(2*pi*Fc*T);
NRF_BBD_Signal2_demod=NRF_IF_Signal2_filtered.*NRF_BBD_Carrier2';
NRF_BBD_Signal2_demod_freq=fft(NRF_BBD_Signal2_demod,length(NRF_BBD_Signal2_
demod));
F=(-length(NRF_BBD_Signal2_demod)/2:length(NRF_BBD_Signal2_demod)/2-
1).*(10*FS/length(NRF_BBD_Signal2_demod));
figure;
subplot(2,1,1),plot(F,abs(fftshift(NRF_BBD_Signal2_demod_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Signal 2 at Baseband Before LowPass Filter With No RF");

%Signal 2 After Lowpass filter with no Rf
NRF_BBD_Signal2_filtered=filter(LowPass25,NRF_BBD_Signal2_demod);
NRF_BBD_Signal2_filtered_freq=fft(NRF_BBD_Signal2_filtered,length(NRF_BBD_Si
gnal2_filtered));
subplot(2,1,2),plot(F,abs(fftshift(NRF_BBD_Signal2_filtered_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Signal 2 at Baseband After LowPass Filter With No RF");

%% Retrieval and Downsampling of both signals with no RF
NRF_Signal1_Retrieved=downsample(NRF_BBD_Signal1_filtered,10); %Signal 1
NRF_Signal2_Retrieved=downsample(NRF_BBD_Signal2_filtered,10); %Signal 2

%Plot Retrieved Signal 1 in Time and Frequency domains
%In Time
T=0:1/(FS):((length(NRF_Signal1_Retrieved)-1)/(FS));
figure;
subplot(2,1,1),plot(T,NRF_Signal1_Retrieved,'b');
grid on;
xlabel("Time (sec)");
ylabel("Amplitude");
legend("Signal 1");
title("Time Domain of Signal 1 After LowPass Filter & Downsampling With No
RF");

%In Frequency
F=(-(length(NRF_Signal1_Retrieved))/2:(length(NRF_Signal1_Retrieved)/2-
1)).*(FS/(length(NRF_Signal1_Retrieved)));
NRF_Signal1_Retrieved_freq=fft(NRF_Signal1_Retrieved);
subplot(2,1,2),plot(F,abs(fftshift(NRF_Signal1_Retrieved_freq)),'b');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1");
title("Frequency Domain of Signal 1 After LowPass Filter & Downsampling With
No RF");

%Plot Retrieved Signal 2 in Time and Frequency domains
%In Time
T=0:1/(FS):((length(NRF_Signal2_Retrieved)-1)/(FS));
```

```
figure;
subplot(2,1,1),plot(T,NRF_Signal2_Retrieved,'k');
grid on;
xlabel("Time (sec)");
ylabel("Amplitude");
legend("Signal 2");
title("Time Domain of Signal 2 After LowPass Filter & Downsampling With No
RF");

%In Frequency
F=(-(length(NRF_Signal2_Retrieved))/2:(length(NRF_Signal2_Retrieved)/2-
1)).*(FS/(length(NRF_Signal2_Retrieved)));
NRF_Signal2_Retrieved_freq=fft(NRF_Signal2_Retrieved);
subplot(2,1,2),plot(F,abs(fftshift(NRF_Signal2_Retrieved_freq)),'k');
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 2");
title("Frequency Domain of Signal 2 After LowPass Filter & Downsampling With
No RF");

%% Adding Offset to the carrier in the Mixer stage with RF (Trying with
Signal 1 only)
IF=25e3;
Fc=100e3;
Offset=1e3;
T=0:1/(10*FS):((length(RF_Signal1_filtered)-1)/(10*FS));
% Then try with Offset=1KHz
%Offset=1e3;

%Mixer Stage
OFF_RF_Carrier=2*cos(2*pi*(Fc+IF+Offset)*T);
OFF_IF_Signal1_demod=RF_Signal1_filtered.*OFF_RF_Carrier';
OFF_IF_Signal1_demod_freq=fft(OFF_IF_Signal1_demod,length(OFF_IF_Signal1_dem
od));
OFF_IF_Signal1_filtered=filter(BandPass25,OFF_IF_Signal1_demod);
OFF_IF_Signal1_filtered_freq=fft(OFF_IF_Signal1_filtered,length(OFF_IF_Signa
l1_filtered));
F=(-(length(OFF_IF_Signal1_demod))/2:(length(OFF_IF_Signal1_demod)/2-
1)).*(10*FS/(length(OFF_IF_Signal1_demod)));
figure;
plot(F,abs(fftshift(OFF_IF_Signal1_filtered_freq)));
grid on;
xlabel("Frequency (Hz)");
ylabel("Magnitude");
legend("Signal 1 Distorted");
title("Frequency Domain of Signal 1 After IF Stage with Offset 0.1KHz");

%BaseBand & detection & downsampling
OFF_IF_Carrier=2*cos(2*pi*IF*T);
OFF_BBD_Signal1_demod=OFF_IF_Signal1_filtered.*OFF_IF_Carrier';
OFF_Signal1_Retrieved=downsample(filter(LowPass25,OFF_BBD_Signal1_demod),10)
;

%% Retrieved signals in different cases for Testing
```

```
%1. Original signal
sound(Signal1,FS);

%2. Retrieved signal with RF bandpass filter present
sound(Signal1_Retrieved,FS);

%3. Retrieved signal with no RF bandpass filter present
sound(NRF_Signal1_Retrieved,FS);

%4. Retrieved signal with mixer offset = 1KHz or 0.1 KHz
sound(OFF_Signal1_Retrieved,FS);
```