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

[1. The transmitter 3](#_Toc406272378)

[Discussion 3](#_Toc406272379)

[The figures 3](#_Toc406272380)

[2. The RF stage 3](#_Toc406272381)

[Discussion 3](#_Toc406272382)

[The figures 4](#_Toc406272383)

[Comments **Error! Bookmark not defined.**](#_Toc406272384)

[3. The IF stage 5](#_Toc406272385)

[Discussion 5](#_Toc406272386)

[The figures 6](#_Toc406272387)

[4. The baseband demodulator 6](#_Toc406272388)

[Discussion 6](#_Toc406272389)

[The figures 7](#_Toc406272390)

[Comments **Error! Bookmark not defined.**](#_Toc406272391)

[5. Performance evaluation without the RF stage 8](#_Toc406272392)

[The figures 8](#_Toc406272393)

[6. Comment on the output sound 10](#_Toc406272394)

[7. The code 10](#_Toc406272395)

**Table of figures**

[Figure 1: The spectrum of the output of the transmitter 3](#_Toc406272396)

[Figure 2: the output of the RF filter (before the mixer) 4](#_Toc406272397)

[Figure 3: The output of the mixer 4](#_Toc406272398)

[Figure 4: Output of the IF filter 6](#_Toc406272399)

[Figure 5: Output of the mixer (before the LPF) 7](#_Toc406272400)

[Figure 6: Output of the LPF 7](#_Toc406272401)

[Figure 7: output of the RF mixer (no RF filter) 8](#_Toc406272402)

[Figure 8: Output of the IF filter (no RF filter) 8](#_Toc406272403)

[Figure 9: Output of the IF mixer before the LPF (no RF filter) 9](#_Toc406272404)

[Figure 10: Ouptut of the LPF (no RF filter) 9](#_Toc406272405)

# 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 : The spectrum of the output of the transmitter

|  |
| --- |
|  |

# 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 ).

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

|  |
| --- |
|  |
|  |

Figure : 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.

# 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 : Output of the IF filter

|  |
| --- |
|  |

# 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 : Output of the mixer (before the LPF)

|  |
| --- |
|  |

Figure : Output of the LPF

|  |
| --- |
|  |

# Performance evaluation without the RF stage

## The figures

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

|  |
| --- |
|  |

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

|  |
| --- |
|  |

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

|  |
| --- |
|  |

Figure : Output of the LPF (no RF filter)

|  |
| --- |
|  |

# 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>

# 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(Signal2\_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\_filtered));  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\_filtered));  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\_Signal1\_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\_Signal2\_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\_demod));  OFF\_IF\_Signal1\_filtered=filter(BandPass25,OFF\_IF\_Signal1\_demod);  OFF\_IF\_Signal1\_filtered\_freq=fft(OFF\_IF\_Signal1\_filtered,length(OFF\_IF\_Signal1\_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); |