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: Ahmed Mokhtar Mahfouz Emam

Sec: 1   BN: 31

Contents

Table of figures

1. The transmitter

This part contains the following tasks

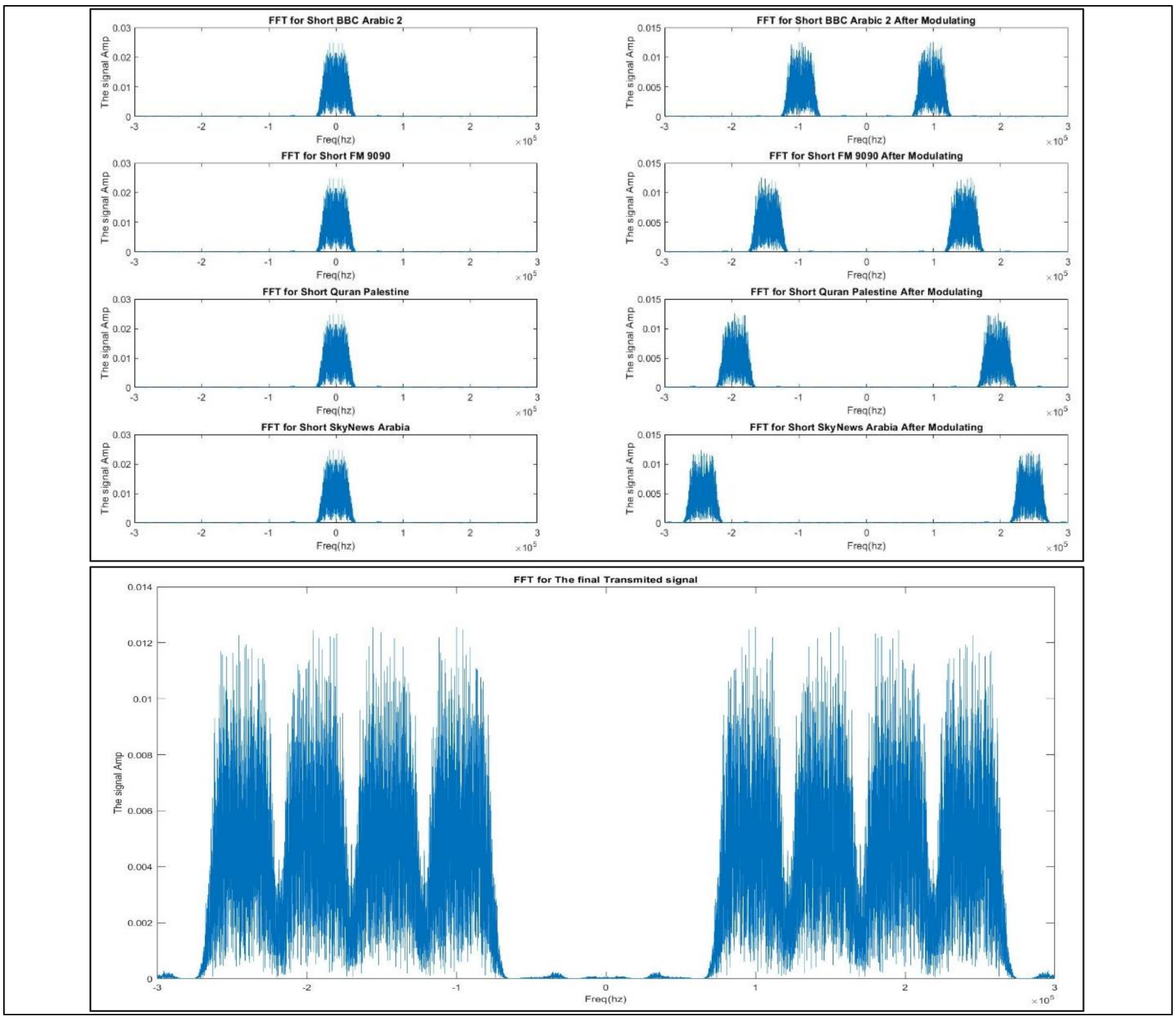
1. Reading monophonic audio signals into MATLAB.
2. Upsampling the audio signals.
3. Modulating the audio signals (each on a separate carrier).
4. Addition of the modulated signals.

Discussion

In this stage we read the messages then make them a single channel stream by adding the two channel together, then we use 'interp' function to up sampling the audio to the sampling frequency greater than double carrier frequency to validate the Nyquist criteria, then modulate each message by multiply by a cosine has frequency = 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. In the demodulating process I demodulate the 4 messages in parallel and show you the output of this and make it tunable in the end by choosing the channel to get the output, I have another way to think is to choose the channel before processing but I prefer this way to show you the output of the 4 message together but the second way is more practically, we can do by moving the switch statement in the end before the RF stage then go through each case it will reduce the processing time and reduce the consumption power.

Discussion

In this stage we pass the signal through a BPF to immune ourselves from image problem while demodulating the signal, here I use a BPF with Bandwidth = 50 KHz centered in the carrier frequency.

3. ***The IF stage***

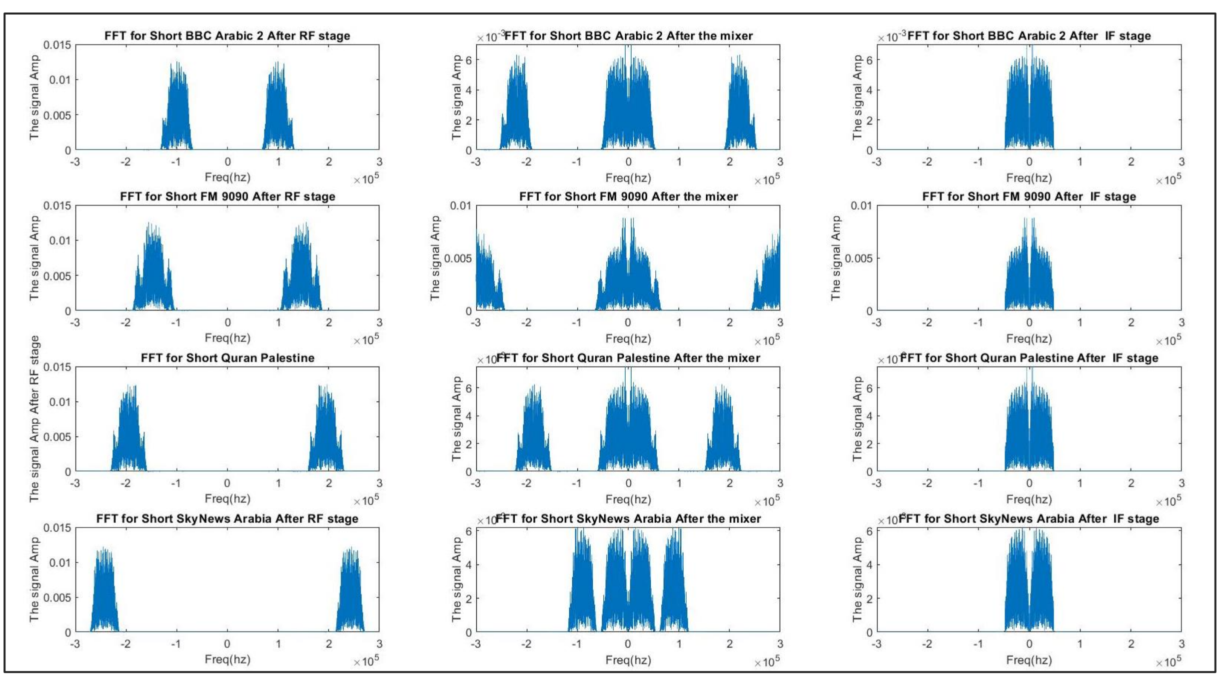
This part addresses the IF filter.

Discussion

In this stage we apply BPF to the signal to reject the upper part of the signal then we have the signal in the IF which called 'Intermediate frequency' not at zero to reject the dc and filter the noise which happened in the channel.

The figures

Figure 2: Output of the RF & IF filter and after the mixer



4. ***The baseband demodulator***

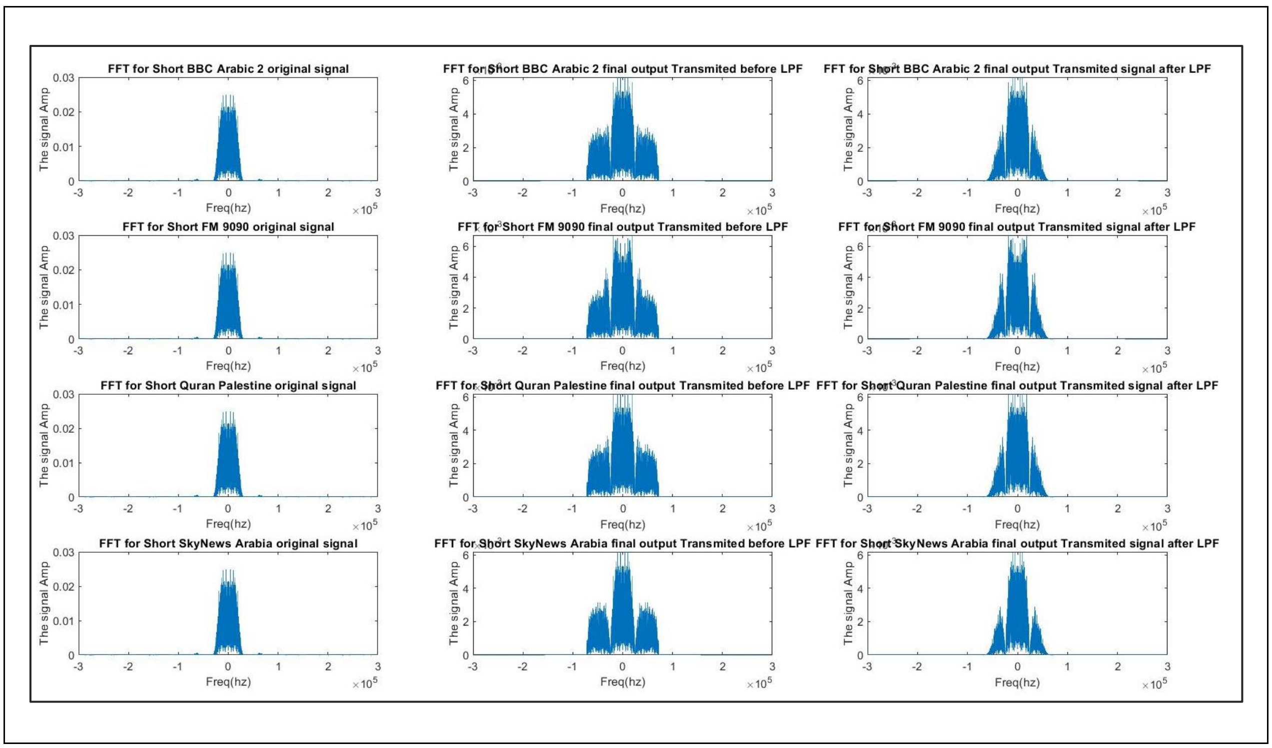
This part addresses the coherent detector used to demodulate the signal from the IF stage. Here I want to comment in the result it has some difference between the final output and the original one because we use ‘interp’ function in the preparing stage while modulating the messages.

Discussion

```
This is the final stage it apply LPF on the message after multiply by cosine
with frequency = Fif, and get it in the baseband to hear it, then, we finish
the demodulating process, we now can play the message but with the new
sampling frequency that we use in the ‘interp’.
```

The figures

Figure 3: Output of the mixer (before the LPF) and the final output (After LPF)



Please note that I can’t use sound function because I get an error ‘invalid sampling rate’, this is due to the frequency of the output message is 617400 Hz and the maximum frequency that can sound function deal with it is 384000, So, I use ‘audiowrite’ to play the output sound.

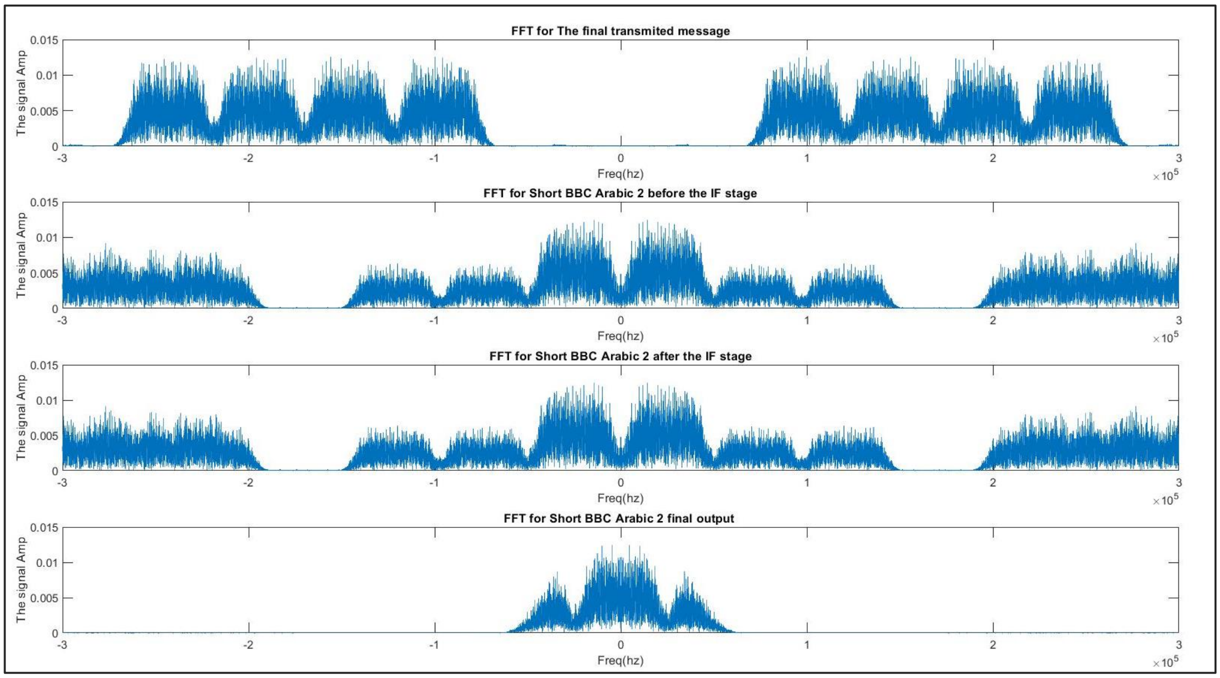
One of the solution I found is to use ‘decimate’ function that down sampling the message again and make it with the original frequency again and I found that in the practically life we use ‘multirate filter’, a digital filter using for down and up sampling.

5. *Performance evaluation without the RF stage*

The figures

Assuming we demodulate BBC Arabia 2 message which is at w0

Figure 4: output after each stage (No RF filter)



6. *Comment on the output sound*

```
I can hear the message very well when there is RF stage but when we remove
this stage I get the first message (BBC Arabia 2) interfaced with the second
message(FM 9090), I can hear the two message together when I try to demodulate
the first message 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

```
For 0.1 KHz I can hear a noisy sound, the sound wasn't clear, but in 1 KHz I
can't hear the sound, it was a very bad sound that you can't recognize.
```

7.The Code

```
%-------------------------------------------------------------------------
% Preparing and transmitting the data
%-------------------------------------
[msg0,fs0] = audioread('Short_BBCArabic2.wav');     %read the audio files
[msg1,fs1] = audioread('Short_FM9090.wav');
[msg2,fs2] = audioread('Short_QuranPalestine.wav');
[msg3,fs3] = audioread('Short_SkyNewsArabia.wav');

msg0 = msg0(:,1) + msg0(:,2);                       % Make each signal as a single channel stream
msg1 = msg1(:,1) + msg1(:,2);
msg2 = msg2(:,1) + msg2(:,2);
msg3 = msg3(:,1) + msg3(:,2);

y=14;                                               % The factor used to increase the sampling rate

msg0 = interp(msg0,y);                              % Make The sampling freq grater than fn/2 => FSnew=y fs
msg1 = interp(msg1,y);
msg2 = interp(msg2,y);
msg3 = interp(msg3,y);

if ( length(msg1)> length(msg0))                    % Get the maximum length of our messages
    x = length(msg1);
else
    x= length(msg0);
end
if(length(msg2) > x)
    x= length(msg2);
elseif (length(msg3)> x )
    x= length(msg3);
end

msg0=[msg0;zeros(x-length(msg0),1)];                % Make all message with the same length
msg1=[msg1;zeros(x-length(msg1),1)];
msg2=[msg2;zeros(x-length(msg2),1)];
msg3=[msg3;zeros(x-length(msg3),1)];

Wn0=2*pi*100000;                                    % Get the carrier frequncies for 4 signals
Wn1=2*pi*150000;
Wn2=2*pi*200000;
Wn3=2*pi*250000;

Ts0= 0:(1/(y*fs0)):(length(msg0)-1)/(y*fs0);        % Make a sampling intervals with length of msg0 and with step 1/(y*fs)
Ts1= 0:(1/(y*fs1)):(length(msg1)-1)/(y*fs1);
Ts2= 0:(1/(y*fs2)):(length(msg2)-1)/(y*fs2);
Ts3= 0:(1/(y*fs3)):(length(msg3)-1)/(y*fs3);


msg0_mod = msg0 .* cos(Wn0*Ts0') ;                  % Get the modulated signal
msg1_mod = msg1 .* cos(Wn1*Ts1') ;
msg2_mod = msg2 .* cos(Wn2*Ts2') ;
msg3_mod = msg3 .* cos(Wn3*Ts3') ;

N=20000;
MSG0= fft(msg0,N);                                  % Apply fft to visualize the data
MSG1= fft(msg1,N);
MSG2= fft(msg2,N);
MSG3= fft(msg3,N);

MSG0_MOD= fft(msg0_mod,N);
MSG1_MOD= fft(msg1_mod,N);
MSG2_MOD= fft(msg2_mod,N);
MSG3_MOD= fft(msg3_mod,N);

MSG0_MOD = fftshift(MSG0_MOD);                      % Apply fftshift to put it in the form we are used to with +ve and -ve freq.
MSG1_MOD = fftshift(MSG1_MOD);
MSG2_MOD = fftshift(MSG2_MOD);
MSG3_MOD = fftshift(MSG3_MOD);

MSG0 = fftshift(MSG0);
MSG1 = fftshift(MSG1);
MSG2 = fftshift(MSG2);
MSG3 = fftshift(MSG3);

Transmited_signal_TD= msg0_mod + msg1_mod + msg2_mod + msg3_mod;    % The final transmited signal
Transmited_signal_FD= MSG0_MOD + MSG1_MOD + MSG2_MOD + MSG3_MOD;
```

```
f= linspace(-300000,300000,length(MSG0));           % The x-axis we will use to visualize the data

subplot(4,2,1)                                      % Plot the data
plot(f,abs(MSG0))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2')
subplot(4,2,2)
plot(f,abs(MSG0_MOD))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 After Modulating')

subplot(4,2,3)
plot(f,abs(MSG1))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short FM 9090')
subplot(4,2,4)
plot(f,abs(MSG1_MOD))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short FM 9090 After Modulating')

subplot(4,2,5)
plot(f,abs(MSG2))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short Quran Palestine')
subplot(4,2,6)
plot(f,abs(MSG2_MOD))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short Quran Palestine After Modulating')

subplot(4,2,7)
plot(f,abs(MSG3))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short SkyNews Arabia')
subplot(4,2,8)
plot(f,abs(MSG3_MOD))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short SkyNews Arabia After Modulating')

figure
plot(f,abs(Transmited_signal_FD))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for The final Transmited signal')
%-------------------------------------------------------------------------------------------------------------
% Receiving the signal process
%----------------------------
msg_recived_RF_0 = bandpass(Transmited_signal_TD,[75000 125000],y*fs0);     %apply bandpass filter to separate each signal
msg_recived_RF_1 = bandpass(Transmited_signal_TD,[125000 175000],y*fs1);
msg_recived_RF_2 = bandpass(Transmited_signal_TD,[175000 225000],y*fs2);
msg_recived_RF_3 = bandpass(Transmited_signal_TD,[225000 275000],y*fs3);


MSG_recived_RF_0= fft(msg_recived_RF_0,N);                              % Apply fft on each signal to visualuize it
MSG_recived_RF_1= fft(msg_recived_RF_1,N);
MSG_recived_RF_2= fft(msg_recived_RF_2,N);
MSG_recived_RF_3= fft(msg_recived_RF_3,N);

MSG_recived_RF_0 = fftshift(MSG_recived_RF_0);                          % Apply fftshift to put it in the form we are used to
MSG_recived_RF_1 = fftshift(MSG_recived_RF_1);
MSG_recived_RF_2 = fftshift(MSG_recived_RF_2);
MSG_recived_RF_3 = fftshift(MSG_recived_RF_3);

Wif = 2*pi*25000;                                                       % The IF freq
msg_recived_IF_0 = msg_recived_RF_0 .* cos((Wn0+Wif)*Ts0') ;            % To make the signal centered at IF freq
msg_recived_IF_1 = msg_recived_RF_1 .* cos((Wn1+Wif)*Ts1') ;
msg_recived_IF_2 = msg_recived_RF_2 .* cos((Wn2+Wif)*Ts2') ;
msg_recived_IF_3 = msg_recived_RF_3 .* cos((Wn3+Wif)*Ts3') ;

MSG_recived_IF_0= fft(msg_recived_IF_0,N);                              % Apply fft on each signal to visualuize it
MSG_recived_IF_1= fft(msg_recived_IF_1,N);
MSG_recived_IF_2= fft(msg_recived_IF_2,N);
MSG_recived_IF_3= fft(msg_recived_IF_3,N);
```

```
msg_recived_IF_filtered_0 = bandpass(msg_recived_IF_0,[1000 49000],y*fs0); % Apply BPF to reject the higer freq image
msg_recived_IF_filtered_1 = bandpass(msg_recived_IF_1,[1000 49000],y*fs0);
msg_recived_IF_filtered_2 = bandpass(msg_recived_IF_2,[1000 49000],y*fs0);
msg_recived_IF_filtered_3 = bandpass(msg_recived_IF_3,[1000 49000],y*fs0);


MSG_recived_IF_0 = fftshift(MSG_recived_IF_0);                          % Apply fftshift to put it in the form we are used to
MSG_recived_IF_1 = fftshift(MSG_recived_IF_1);
MSG_recived_IF_2 = fftshift(MSG_recived_IF_2);
MSG_recived_IF_3 = fftshift(MSG_recived_IF_3);

MSG_recived_IF_filtered_0= fft(msg_recived_IF_filtered_0,N);            % Apply fft on each signal to visualuize it
MSG_recived_IF_filtered_1= fft(msg_recived_IF_filtered_1,N);
MSG_recived_IF_filtered_2= fft(msg_recived_IF_filtered_2,N);
MSG_recived_IF_filtered_3= fft(msg_recived_IF_filtered_3,N);

MSG_recived_IF_filtered_0 =fftshift(MSG_recived_IF_filtered_0);         % Apply fftshift to put it in the form we are used to
MSG_recived_IF_filtered_1 =fftshift(MSG_recived_IF_filtered_1);
MSG_recived_IF_filtered_2 =fftshift(MSG_recived_IF_filtered_2);
MSG_recived_IF_filtered_3 =fftshift(MSG_recived_IF_filtered_3);

figure                                                                  % Used to separate the images in two windows
subplot(4,3,1)                                                          % Plot the data
plot(f,abs(MSG_recived_RF_0))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 After RF stage')
subplot(4,3,2)
plot(f,abs(MSG_recived_IF_0))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 After the mixer')
subplot(4,3,3)
plot(f,abs(MSG_recived_IF_filtered_0))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 After  IF stage')

subplot(4,3,4)
plot(f,abs(MSG_recived_RF_1))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short FM 9090 After RF stage')
subplot(4,3,5)
plot(f,abs(MSG_recived_IF_1))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short FM 9090 After the mixer')
subplot(4,3,6)
plot(f,abs(MSG_recived_IF_filtered_1))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short FM 9090 After  IF stage')

subplot(4,3,7)
plot(f,abs(MSG_recived_RF_2))
xlabel('Freq(hz)')
ylabel('The signal Amp After RF stage')
title('FFT for Short Quran Palestine')
subplot(4,3,8)
plot(f,abs(MSG_recived_IF_2))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short Quran Palestine After the mixer')
subplot(4,3,9)
plot(f,abs(MSG_recived_IF_filtered_2))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short Quran Palestine After  IF stage')

subplot(4,3,10)
plot(f,abs(MSG_recived_RF_3))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short SkyNews Arabia After RF stage')
subplot(4,3,11)
plot(f,abs(MSG_recived_IF_3))
xlabel('Freq(hz)')
ylabel('The signal Amp')
```

```
title('FFT for Short SkyNews Arabia After the mixer')
subplot(4,3,12)
plot(f,abs(MSG_recived_IF_filtered_3))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short SkyNews Arabia After  IF stage')


msg_recived_BP_0 = msg_recived_IF_filtered_0 .* cos(Wif*Ts0') ;             % Move the signals to the Baseband
msg_recived_BP_1 = msg_recived_IF_filtered_1 .* cos(Wif*Ts1') ;
msg_recived_BP_2 = msg_recived_IF_filtered_2 .* cos(Wif*Ts2') ;
msg_recived_BP_3 = msg_recived_IF_filtered_3 .* cos(Wif*Ts3') ;

msg_recived_Final_0=lowpass(msg_recived_BP_0,25000,y*fs0);                  % Apply LPF to filterize the signal
msg_recived_Final_1=lowpass(msg_recived_BP_1,25000,y*fs1);
msg_recived_Final_2=lowpass(msg_recived_BP_2,25000,y*fs2);
msg_recived_Final_3=lowpass(msg_recived_BP_3,25000,y*fs3);

MSG_recived_BP_0= fft(msg_recived_BP_0,N);                                  % Apply fft on each signal to visualuize it
MSG_recived_BP_1= fft(msg_recived_BP_1,N);
MSG_recived_BP_2= fft(msg_recived_BP_2,N);
MSG_recived_BP_3= fft(msg_recived_BP_3,N);

MSG_recived_BP_0 = fftshift(MSG_recived_BP_0);
MSG_recived_BP_1 = fftshift(MSG_recived_BP_1);
MSG_recived_BP_2 = fftshift(MSG_recived_BP_2);
MSG_recived_BP_3 = fftshift(MSG_recived_BP_3);


MSG_recived_Final_0= fft(msg_recived_Final_0,N);                            % Apply fft on each signal to visualuize it
MSG_recived_Final_1= fft(msg_recived_Final_1,N);
MSG_recived_Final_2= fft(msg_recived_Final_2,N);
MSG_recived_Final_3= fft(msg_recived_Final_3,N);

MSG_recived_Final_0 = fftshift(MSG_recived_Final_0);                        % Apply fftshift to put it in the form we are used to
MSG_recived_Final_1 = fftshift(MSG_recived_Final_1);
MSG_recived_Final_2 = fftshift(MSG_recived_Final_2);
MSG_recived_Final_3 = fftshift(MSG_recived_Final_3);


figure
subplot(4,3,1)                                                              % Plot the data
plot(f,abs(MSG0))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 original signal')
subplot(4,3,2)
plot(f,abs(MSG_recived_BP_0))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 final output Transmited before LPF')
subplot(4,3,3)
plot(f,abs(MSG_recived_Final_0))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 final output Transmited signal after LPF')


subplot(4,3,4)
plot(f,abs(MSG1))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short FM 9090 original signal')
subplot(4,3,5)
plot(f,abs(MSG_recived_BP_1))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short FM 9090 final output Transmited before LPF')
subplot(4,3,6)
plot(f,abs(MSG_recived_Final_1))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short FM 9090 final output Transmited signal after LPF')

subplot(4,3,7)
plot(f,abs(MSG2))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short Quran Palestine original signal')
subplot(4,3,8)
```

```
plot(f,abs(MSG_recived_BP_2))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short Quran Palestine final output Transmited before LPF')
subplot(4,3,9)
plot(f,abs(MSG_recived_Final_2))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short Quran Palestine final output Transmited signal after LPF')

subplot(4,3,10)
plot(f,abs(MSG3))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short SkyNews Arabia original signal')
subplot(4,3,11)
plot(f,abs(MSG_recived_BP_3))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short SkyNews Arabia final output Transmited before LPF')
subplot(4,3,12)
plot(f,abs(MSG_recived_Final_3))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short SkyNews Arabia final output Transmited signal after LPF')


Ch_number = input ('Please, Input the channel you want from 1 to 4 :)\n');

switch Ch_number                                                            % Write the final recived messages
    case 1
        audiowrite('msg0.wav',msg_recived_Final_0,y*fs0)
        msg_recived_Final_0 = decimate(msg_recived_Final_0,y);
        sound(msg_recived_Final_0,fs0)
    case 2
        audiowrite('msg1.wav',msg_recived_Final_1,y*fs1)
        msg_recived_Final_1 = decimate(msg_recived_Final_1,y);
        sound(msg_recived_Final_1,fs1)
    case 3
        audiowrite('msg2.wav',msg_recived_Final_2,y*fs2)
        msg_recived_Final_2 = decimate(msg_recived_Final_2,y);
        sound(msg_recived_Final_2,fs2)
    case 4
        audiowrite('msg3.wav',msg_recived_Final_3,y*fs3)
        msg_recived_Final_3 = decimate(msg_recived_Final_3,y);
        sound(msg_recived_Final_3,fs3)
    otherwise
        disp('Sorry, Wrong channel :(')
end


%-----------------------------------------------------------------------------------------------------------------------------
```

% requirement 4 code:

```
msg_recived_no_filter = Transmited_signal_TD .* cos((Wn0+Wif)*Ts0') ;
MSG_recived_no_filter = fft(msg_recived_no_filter,N);
MSG_recived_no_filter = fftshift(MSG_recived_no_filter);

msg_recived_no_filter_IF = msg_recived_no_filter .* cos(Wif*Ts0') ;
MSG_recived_no_filter_IF = fft(msg_recived_no_filter,N);
MSG_recived_no_filter_IF = fftshift(MSG_recived_no_filter_IF);

msg_recived_no_filter_final=lowpass(msg_recived_no_filter_IF,25000,y*fs0);
MSG_recived_no_filter_final = fft(msg_recived_no_filter_final,N);
MSG_recived_no_filter_final = fftshift(MSG_recived_no_filter_final);

audiowrite('msg_no_RF.wav',msg_recived_no_filter_final,y*fs1)


figure
subplot(4,1,1)
plot(f,abs(Transmited_signal_FD))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for The final transmited message')

subplot(4,1,2)
```

```
plot(f,abs(MSG_recived_no_filter))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 before the IF stage')

subplot(4,1,3)
plot(f,abs(MSG_recived_no_filter_IF))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 after the IF stage')

subplot(4,1,4)
plot(f,abs(MSG_recived_no_filter_final))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 final output ')

%-------------------------------------------------------------------------------------------------------------
```

% requirement 5 code:

```
Wif_1 = 2*pi*(25000+100);                                                            % The IF freq with 0.1 KHz offset

msg_recived_BP_With_offset_1 = msg_recived_IF_filtered_0 .* cos(Wif_1*Ts0') ;         % Move the signals to the Baseband
MSG_recived_BP_With_offset_1 = fft(msg_recived_BP_With_offset_1,N);
MSG_recived_BP_With_offset_1 = fftshift(MSG_recived_BP_With_offset_1);

msg_recived_Final_With_offset_1=lowpass(msg_recived_BP_With_offset_1,25000,y*fs0);          % Apply LPF to filterize the signal
MSG_recived_Final_With_offset_1 = fft(msg_recived_Final_With_offset_1,N);
MSG_recived_Final_With_offset_1 = fftshift(MSG_recived_Final_With_offset_1);


Wif_2 = 2*pi*(25000+1000);                                                           % The IF freq with 1 KHz offset

msg_recived_BP_With_offset_2 = msg_recived_IF_filtered_0 .* cos(Wif_2*Ts0') ;         % Move the signals to the Baseband
MSG_recived_BP_With_offset_2 = fft(msg_recived_BP_With_offset_2,N);
MSG_recived_BP_With_offset_2 = fftshift(MSG_recived_BP_With_offset_2);

msg_recived_Final_With_offset_2=lowpass(msg_recived_BP_With_offset_2,25000,y*fs0);          % Apply LPF to filterize the signal
MSG_recived_Final_With_offset_2 = fft(msg_recived_Final_With_offset_2,N);
MSG_recived_Final_With_offset_2 = fftshift(MSG_recived_Final_With_offset_2);

audiowrite('msg_Offset_0.wav',msg_recived_Final_With_offset_1,y*fs1)
audiowrite('msg_Offset_1.wav',msg_recived_Final_With_offset_2,y*fs1)


figure
subplot(4,1,1)
plot(f,abs(MSG_recived_BP_With_offset_1))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 with 0.1 KHz offset')

subplot(4,1,2)
plot(f,abs(MSG_recived_BP_With_offset_2))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for Short BBC Arabic 2 with 1 KHz offset')

subplot(4,1,3)
plot(f,abs(MSG_recived_Final_With_offset_1))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for final output of Short BBC Arabic 2 due 0.1 KHz offset')

subplot(4,1,4)
plot(f,abs(MSG_recived_Final_With_offset_2))
xlabel('Freq(hz)')
ylabel('The signal Amp')
title('FFT for final output of Short BBC Arabic 2 due 1 KHz offset')
```

Published with MATLAB® R2020a