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: فتحى مصطفى فتحى عبدالحميد**

**Sec: 3 BN:19 ID: 9210808**

**Student Name: محمود حمدى جابر عبدالتواب**

**Sec: 3 BN: ID:9211083**

Contents

[1. The transmitter 3](#_Toc406272378)

[Discussion 3](#_Toc406272379)

[The figures 3](#_Toc406272380)

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

[Discussion 3](#_Toc406272382)

[The figures 3](#_Toc406272383)

[Comments 3](#_Toc406272384)

[3. The IF stage 4](#_Toc406272385)

[Discussion 4](#_Toc406272386)

[The figures 4](#_Toc406272387)

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

[Discussion 4](#_Toc406272389)

[The figures 4](#_Toc406272390)

[Comments 4](#_Toc406272391)

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

[The figures 5](#_Toc406272393)

[6. Comment on the output sound 5](#_Toc406272394)

[7. The code 5](#_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) 3](#_Toc406272397)

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

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

[Figure 5: Output of the mixer (before the LPF) 4](#_Toc406272400)

[Figure 6: Output of the LPF 4](#_Toc406272401)

[Figure 7: output of the RF mixer (no RF filter) 5](#_Toc406272402)

[Figure 8: Output of the IF filter (no RF filter) 5](#_Toc406272403)

[Figure 9: Output of the IF mixer before the LPF (no RF filter) 5](#_Toc406272404)

[Figure 10: Ouptut of the LPF (no RF filter) 5](#_Toc406272405)

# 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 reading signals, then making signal monophonic channel and making them have the same length so that I can add them together, we make up sampling to avoid aliasing, then we modulate each signal with different carrier and after multiply with carriers add them together , then make FFT and display the spectrum and transmit the FDM Signal. |

## 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

|  |
| --- |
| The RF stage is implemented as a BPF, centered at the carrier frequency. This stage is very. important as it rejects image frequency (at 𝐹 = +2 )noise and unwanted signals. Then comes the mixer. that shifts the required signal at the intermediate frequency (=27.5kHz in us case). We do that to avoid direct conversion problems, for the first signal image will be at F=155KHZ and for second signal image will be at 210KHZ so the BPF is very important . |

## 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

|  |
| --- |
|  |

# The IF stage

This part addresses the IF filter.

## Discussion

|  |
| --- |
| This stage is implemented as a BPF only, center at the intermediate frequency, to select the IF signal & reject its higher frequency version, and the importance of carrying the received signal on IF before baseband to get rid of leakage & flicker noise & to improve the filters’ selectivity. |

## 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

|  |
| --- |
| In baseband stage: we demodulate the output signal of IF stage by a carrier signal with carrier frequency at , so we can have the desired signal at the baseband and at 2 . In LPF stage we apply low pass filter to obtain the desired signal at baseband only, after that we do downsampling to return the original sampling rate, then I can listen the desired sound |

## 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 : Ouptut of the LPF (no RF filter)

|  |
| --- |
|  |

# Comment on the output sound

|  |
| --- |
| The output in case of RF stage existence is the same as the original audio signal without any interference with other signals. The output without RF stage sound1 is the original signal interfered with sound2 and the output audio contains the two audio signals. This happens because the second message is modulated with carrier frequency equal 155 KHZ and this frequency is image frequency of the first message so there is will interface with the two messages |

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

First case Offset = 0.1 KHZ: the sound was a little distorted; it is different than the original sound but still recognizable

Second case Offset = 1 KHZ: the sound was more distorted, the sound will be very bad it is different than the original and neither recognizable nor understood.

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.

# The code

|  |
| --- |
| 1. %\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Super-heterodyne Receiver\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*% 2. clear all; 3. clc ; 4. close all; 5. % read sounds 6. [Sound1,FS]=audioread("D:\3rd year\Communications\CODES\New folder\Project\Short\_QuranPalestine.wav"); 7. [Sound2,FS]=audioread("D:\3rd year\Communications\CODES\New folder\Project\Short\_SkyNewsArabia.wav");  10. % get sizes 11. Length\_Sound1 = length(Sound1); 12. Length\_Sound2 = length(Sound2); 13. if(Length\_Sound1>Length\_Sound2) 14. Sound2=wextend('ar','zpd',Sound2,(Length\_Sound1-Length\_Sound2),'d'); 15. elseif (Length\_Sound2>Length\_Sound1) 16. Sound1=wextend('ar','zpd',Sound1,(Length\_Sound2-Length\_Sound1),'d'); 17. end 19. %make signal Monophonic 20. Sound1(:,1)=Sound1(:,1)+Sound1(:,2); 21. Sound1(:,2) = []; 22. Sound2(:,1)=Sound2(:,1)+Sound2(:,2); 23. Sound2(:,2) = []; 25. %achieving Nyquist rule=10\*FS 26. Message1=interp(Sound1 , 10 ) ; 27. Message2=interp(Sound2 , 10 ) ; 29. % F(new)=10\*FS 30. FS = FS \* 10 ; 32. %get n for Carrier 33. N = length(Message1) ; 35. TS = 1/FS ; %get time 36. Stop\_Time=N/FS; %get stop time 37. t = (0:TS:Stop\_Time-TS)'; 38. Carrier1 = cos(2\*pi\*100\*1000\*t) ; 39. %carrier2 with 100+50n KHZ 40. Carrier2 = cos(2\*pi\*(100+55)\*1000\*t) ; 41. %get frequency responce of Two Messages 42. Message1\_Spectrum=fft(Message1); 43. Message2\_Spectrum=fft(Message2); 45. k=-N/2:N/2-1; 46. %figure 47. %plot(k\*FS/N,fftshift(abs(Message1\_Spectrum))); 48. %xlabel('Frequency'); 49. %title('Message1 Spectrum'); 50. %figure 51. %plot(k\*FS/N,fftshift(abs(Message2\_Spectrum))); 52. %xlabel('Frequency'); 53. %title('Message 2 Spectrum'); 55. %Modulating signals 56. Transmitter1\_Sound=Message1.\*Carrier1 ; 57. Transmitter2\_Sound=Message2.\*Carrier2 ; 58. % Create The Frequency Division Multiplexed Signal By Addition Of The Modulated Signals 59. Transmitter\_Output=Transmitter1\_Sound+Transmitter2\_Sound; 60. %plotting spectrum of the channel 61. % perform FFT on signal 62. FDM = fft(Transmitter\_Output ); 64. figure 65. plot(k\*FS/N,fftshift(abs(FDM))); 66. xlabel('Frequency'); 67. title('Frequency Division Multiplexing Spectrum'); 69. %\*\*\*\*\*\*\*\*\*RF Stage\*\*\*\*\*\*\*\*\*\*% 70. %\*\*\*\*\*choose The Channel\*\*\*\*\*\*\*% 71. % Choose The Required Audio Signals 72. disp (" "); 73. disp ("\*\*\*\*\*\*\*\*\*Channels\*\*\*\*\*\*\*\* "); 74. disp ("1. Short Quran Palestine: 100 KHz"); 75. disp ("2. Short Sky News Arabia 155 KHz"); 76. %disp ("3. Russian Voice On Carrier: 200 KHz"); 77. Freq\_Channel = input ("Please Select The Desired Channel Frequency in KHz: "); 78. Freq\_Channel = 1000 \* Freq\_Channel; % convert it to KHZ 80. %design BPF for First sound 81. Fstop1=Freq\_Channel-24000; % Edge of the stopband 82. Fpass1=Freq\_Channel-22000; % Edge of the passband 83. Astop1=80; % Attenuation in the first stopband 84. Fpass2=Freq\_Channel+22000; % Closing edge of the passband 85. Fstop2=Freq\_Channel+24000; % Edge of the second stopband 86. Astop2=80; % Attenuation in the second stopband 87. Apass=0.001; % Amount of ripple allowed in the passband 88. %specs of Bpf 89. BPF\_specs=fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2', ... 90. Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass,Astop2,FS); 91. BPF = design(BPF\_specs); 92. RF\_Message= filter(BPF,Transmitter\_Output); 93. RF=fft(RF\_Message); 94. figure 95. plot(k\*FS/N,fftshift(abs(RF))); 96. xlabel('Frequency'); 97. title(' BPF Outout');  100. %\*\*\*\*\*\*\*\*\*\* Mixer\*\*\*\*\*\*\*\*% 101. Freq\_IF=27500; 102. Mixer\_Carrier=cos(2\*pi\*(Freq\_IF+Freq\_Channel)\*t); 103. Mixer\_Output=RF\_Message.\*Mixer\_Carrier; 104. Mixer\_Output\_FFT=fft(Mixer\_Output); 105. figure 106. plot(k\*FS/N,fftshift(abs(Mixer\_Output\_FFT))); 107. xlabel('Frequency'); 108. title('Mixer Output'); 109. ylabel('Mixer Output');  112. %\*\*\*\*\*\*\*\*\*IF Stage\*\*\*\*\*\*\*\*\*\*\*% 113. %Design Baseband BPF 114. Fstop1=Freq\_IF-24000; % Edge of the stopband 115. Fpass1=Freq\_IF-22000; % Edge of the passband 116. Astop1=80; % Attenuation in the first stopband 117. Fpass2=Freq\_IF+22000; % Closing edge of the passband 118. Fstop2=Freq\_IF+24000; % Edge of the second stopband 119. Astop2=80; % Attenuation in the second stopband 120. Apass=0.001; % Amount of ripple allowed in the passband 121. BPF\_specs=fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2', ... 122. Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass,Astop2,FS); 123. BPF = design(BPF\_specs); 124. %fvtool(BPF) %response of filter 125. IF\_Output= filter(BPF,Mixer\_Output); 126. IF\_Output\_FFT=fft(IF\_Output); 128. figure 129. plot(k\*FS/N,fftshift(abs(IF\_Output\_FFT))); 130. xlabel('Frequency'); 131. title('IF Stage Output'); 133. %\*\*\*\*\*\*\*\*\*\*\*Baseband Detection\*\*\*\*\*\*\*\*\*\*\*\*\*% 134. Carrier\_Detection=cos(2\*pi\*27500\*t); 135. Detection\_Output=IF\_Output.\*Carrier\_Detection; 136. Detection\_Output\_FFT=fft(Detection\_Output); 137. figure 138. plot(k\*FS/N,fftshift(abs(Detection\_Output\_FFT))); 139. xlabel('Frequency'); 140. title('Baseband Detection Output'); 142. %\*\*\*\*\*\*Filter\*\*\*\*\*\*% 143. F\_pass = 22000; % Edge of the lowband 144. F\_stop = 24000; % Edge of the stopband 145. A\_pass = 0.001; % Amount of ripple allowed in the band 146. A\_stop = 80; % Attenuation in the band 147. LPF\_specs=fdesign.lowpass('Fp,Fst,Ap,Ast', ... 148. F\_pass, F\_stop, A\_pass, A\_stop, FS); 149. LPF = design(LPF\_specs); 150. LPF\_Output= filter(LPF,Detection\_Output); 151. LPF\_Output\_FFT= fft(LPF\_Output); 152. figure 153. plot(k\*FS/N,fftshift(abs(LPF\_Output\_FFT))); 154. xlabel('Frequency'); 155. title('Output after Low Pass Filter'); 156. LPF\_Output=4.\*LPF\_Output; 157. Reciever=downsample(LPF\_Output,10); %down sampling /10 158. sound(Reciever,FS/10); |