LAB #09

Modeling Frequency Division Multiplexing/DE-multiplexing



Fall 2023

**CSE-402L Digital Signal Processing Lab**

Submitted by: MUHAMMAD SADEEQ

Registration No.: 21PWCSE2028

Section: C

“On my honor, as a student of the University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work”

Submitted to:

Dr. Yasir Saleem Afridi

(7 Jan 2024)

Department of Computer systems engineering

University of Engineering and Technology, Peshawar

**CODE:**

close all;

clear all;

bandwidth = 4000;

guard\_band = 300;

signal\_to\_noise\_ratio = 20;

ssb\_modulation = 1;

carrier\_freq1 = bandwidth \* 3;

carrier\_freq2 = bandwidth \* 4;

carrier\_freq3 = bandwidth \* 5;

sampling\_freq = carrier\_freq3 \* 2 + 5000;

cutoff\_freq = 2500;

show\_graphics = 1;

play\_sound = 1;

[B,A] = butter(3,cutoff\_freq/(sampling\_freq/2));

low\_pass\_filter = @(signal) filter(B,A,signal);

[C1,D1] = butter(2,[bandwidth\*2+guard\_band,bandwidth\*3-guard\_band]/(sampling\_freq/2));

band\_filter3=@(signal) filter(C1,D1,signal);

[C2,D2] = butter(2,[bandwidth\*3+guard\_band,bandwidth\*4-guard\_band]/(sampling\_freq/2));

band\_filter4=@(signal) filter(C2,D2,signal);

[C3,D3] = butter(2,[bandwidth\*4+guard\_band,bandwidth\*5-guard\_band]/(sampling\_freq/2));

band\_filter5=@(signal) filter(C3,D3,signal);

signal1 = audioread("Sound1.m4a");

length\_signal1 = length(signal1);

signal2 = audioread("Sound2.m4a");

length\_signal2 = length(signal2);

signal3 = audioread("Sound3.m4a");

length\_signal3 = length(signal3);

beep\_sound = audioread("beep.mp3");

beep\_player = audioplayer(beep\_sound,44100);

min\_length = min([length\_signal1,length\_signal2]);

time = linspace(0,5,min\_length);

signal1 = signal1(1:min\_length);

signal2 = signal2(1:min\_length);

signal3 = signal3(1:min\_length);

flag = input("Step2, the signals are reproduced as they arrive");

if(play\_sound>0)

player1 = audioplayer(signal1,44100);

playblocking(player1);

playblocking(beep\_player);

player2 = audioplayer(signal2,44100);

playblocking(player2);

playblocking(beep\_player);

player3 = audioplayer(signal3,44100);

playblocking(player3);

end

flag = input("Step 3 plot the spectra of the signals as they arrive");

if(show\_graphics>0)

figure

spectrum1 = abs(fft(signal1));

subplot(3,1,1),plot(spectrum1),grid on;zoom,title('Spectrum of filtered signal\_1');

spectrum2 = abs(fft(signal2));

subplot(3,1,2),plot(spectrum2),grid on,zoom,title('Spectrum of filtered signal\_2');

spectrum3 = abs(fft(signal1));

subplot(3,1,3),plot(spectrum3),grid on,zoom,title('Spectrum of filtered signal\_3');

end

flag = input('Step 4 reproduce the signals after passing them through the filter');

if(play\_sound>0)

beep\_player = audioplayer(beep\_sound,44100);

player1 = audioplayer(signal1,44100);

playblocking(player1);

playblocking(beep\_player);

player2 = audioplayer(signal2,44100);

playblocking(player2);

playblocking(beep\_player);

player3 = audioplayer(signal1,44100);

playblocking(player3);

playblocking(beep\_player);

end

flag = input('Step 5 the signals are modulated to different carriers');

if(ssb\_modulation>0)

modulated\_signal1 = ssbmod(signal1,carrier\_freq1,sampling\_freq);

modulated\_signal2 = ssbmod(signal2,carrier\_freq2,sampling\_freq);

modulated\_signal3 = ssbmod(signal3,carrier\_freq3,sampling\_freq);

else

modulated\_signal1 = ammod(signal1,carrier\_freq1,sampling\_freq);

modulated\_signal2 = ammod(signal2,carrier\_freq2,sampling\_freq);

modulated\_signal3 = ammod(signal3,carrier\_freq3,sampling\_freq);

end

if(show\_graphics>0)

figure

spectrum1 = abs(fft(modulated\_signal1));

subplot(3,1,1),plot(spectrum1),grid on,zoom,title('Spectrum of modulated signal\_1');

spectrum2 = abs(fft(modulated\_signal2));

subplot(3,1,2),plot(spectrum2),grid on,zoom,title('Spectrum of modulated signal\_2');

spectrum3 = abs(fft(modulated\_signal3));

subplot(3,1,3),plot(spectrum1),grid on,zoom,title('Spectrum of modulated signal\_3');

end

flag = input('Step 6 the modulated signals are filtered in the defined bands and added');

filtered\_signal1 = band\_filter3(modulated\_signal1);

filtered\_signal2 = band\_filter4(modulated\_signal2);

filtered\_signal3 = band\_filter5(modulated\_signal3);

complete\_signal = filtered\_signal1 + filtered\_signal2 + filtered\_signal3;

if(show\_graphics > 0)

figure

spectrum1 = abs(fft(filtered\_signal1));

subplot(4,1,1),plot(spectrum1),grid on,zoom,title('Spectrum signal\_1 modulated and filtered');

spectrum2 = abs(fft(filtered\_signal2));

subplot(4,1,2),plot(spectrum2),grid on,zoom,title('Spectrum signal\_2 modulated and filtered');

spectrum3 = abs(fft(filtered\_signal3));

subplot(4,1,3),plot(spectrum3),grid on,zoom,title('Spectrum signal\_3 modulated and filtered');

total\_spectrum = abs(fft(complete\_signal));

subplot(4,1,4),plot(total\_spectrum),grid on,zoom,title('Summed Spectrum');

end

if(show\_graphics > 0)

figure

total\_spectrum = abs(fft(complete\_signal));

subplot(2,1,1),plot(total\_spectrum),grid on,zoom,title('Full signal spectrum without noise');

end

flag = input('Step 8 upon arrival each band is filtered');

demod\_signal1 = band\_filter3(complete\_signal);

demod\_signal2 = band\_filter4(complete\_signal);

demod\_signal3 = band\_filter5(complete\_signal);

if(show\_graphics > 0)

figure

spectrum1 = abs(fft(demod\_signal1));

subplot(3,1,1),plot(spectrum1),grid on,zoom,title('Spectrum signal\_1 filtered');

spectrum2 = abs(fft(demod\_signal2));

subplot(3,1,2),plot(spectrum2),grid on,zoom,title('Spectrum signal\_2 filtered');

spectrum3 = abs(fft(demod\_signal3));

subplot(3,1,3),plot(spectrum3),grid on,zoom,title('Spectrum signal\_3 filtered');

end

flag = input('Step 9 each recovered band is demodulated to return the signal to the baseband frequency');

if(ssb\_modulation >0)

demod\_signal1 = ssbdemod(demod\_signal1,carrier\_freq1,sampling\_freq);

demod\_signal2 = ssbdemod(demod\_signal2,carrier\_freq2,sampling\_freq);

demod\_signal3 = ssbdemod(demod\_signal3,carrier\_freq3,sampling\_freq);

else

demod\_signal1 = amdemod(demod\_signal1,carrier\_freq1,sampling\_freq);

demod\_signal2 = amdemod(demod\_signal2,carrier\_freq2,sampling\_freq);

demod\_signal3 = amdemod(demod\_signal3,carrier\_freq3,sampling\_freq);

end

if(show\_graphics > 0)

figure

spectrum1 = abs(fft(demod\_signal1));

subplot(3,1,1),plot(spectrum1),grid on,zoom,title('Spectrum of demodulated signal\_1');

spectrum2 = abs(fft(demod\_signal2));

subplot(3,1,2),plot(spectrum2),grid on,zoom,title('Spectrum of demodulated signal\_2');

spectrum3 = abs(fft(demod\_signal3));

subplot(3,1,3),plot(spectrum3),grid on,zoom,title('Spectrum of demodulated signal\_3');

end

flag = input('Step 10 the recovered signal is passed through a low pass filter');

demod\_signal1 = low\_pass\_filter(demod\_signal1);

demod\_signal2 = low\_pass\_filter(demod\_signal2);

demod\_signal3 = low\_pass\_filter(demod\_signal3);

if(show\_graphics > 0)

figure

spectrum1 = abs(fft(demod\_signal1));

subplot(3,1,1),plot(spectrum1),grid on,zoom,title('Spectrum signal\_1 demodulated');

spectrum2 = abs(fft(demod\_signal2));

subplot(3,1,2),plot(spectrum2),grid on,zoom,title('Spectrum signal\_2 demodulated');

spectrum3 = abs(fft(demod\_signal3));

subplot(3,1,3),plot(spectrum3),grid on,zoom,title('Spectrum signal\_3 demodulated');

end

flag = input('Step 11 play the reproduced signal after transmission');

player4 = audioplayer(demod\_signal1,44100);

playblocking(player4);

playblocking(beep\_player);

player5 = audioplayer(demod\_signal2,44100);

playblocking(player5);

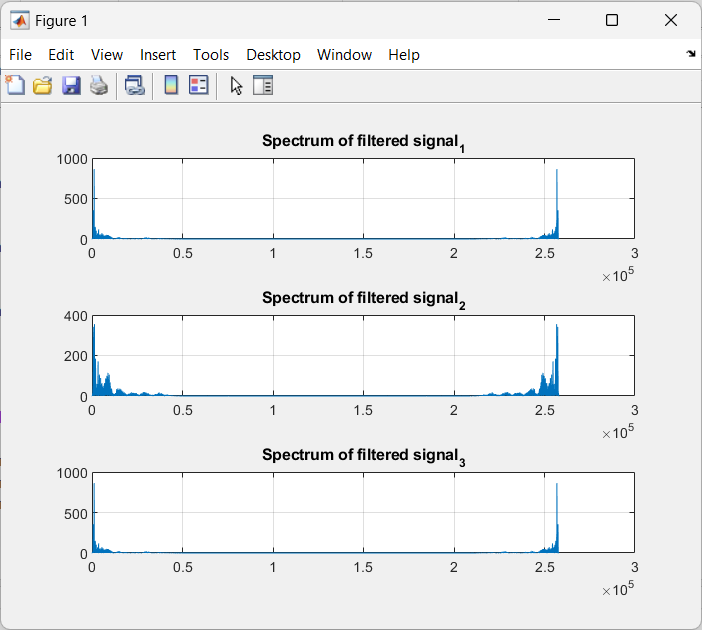
playblocking(beep\_player);

player6 = audioplayer(demod\_signal3,44100);

playblocking(player6);

playblocking(beep\_player);

**Output:**

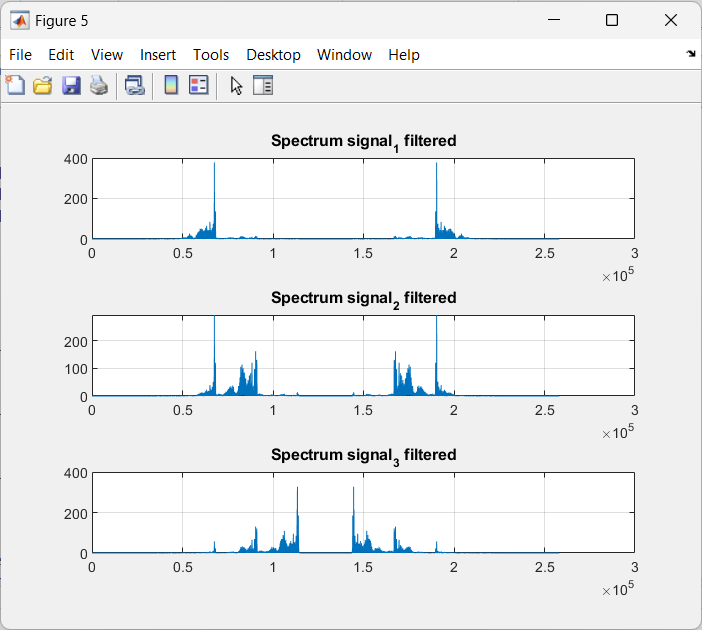
****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**