**MODELINF FREQUENCY DIVISION MULTIPLEXING/DEMULTIPLEXING**

**LAB # 09**

****

**Fall 2023**

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

**Submitted by: Muhammad Shahab**

**Registration No. : 21PWCSE1980**

**Class Section: C**

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

**Submitted to:**

**Dr. Yasir Saleem Afridi**

**DATE: 8 / JAN / 2023**

**Department of Computer Systems Engineering**

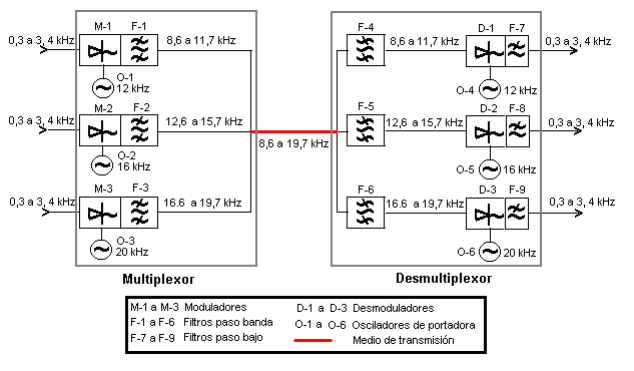
**University of Engineering and Technology, Peshawar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Demonstration of Concepts** | **Poor (Does not meet expectation (1))**  The student failed to demonstrate a clear understanding of the assignment concepts | **Fair (Meet Expectation (2-3))**  The student demonstrated a clear understanding of some of the assignment concepts | **Good (Exceeds Expectation (4-5)**  The student demonstrated a clear understanding of the assignment concepts | **Score**  **30%** |
| **Accuracy** | The student completed ( <50%) tasks and provided MATLAB code and/or Simulink models with errors. Outputs shown are not correct in form of graphs (no labels) and/or tables along with incorrect analysis or remarks. | The student completed partial tasks (50% - <90%) with accurate MATLAB code and/or Simulink models. Correct outputs are shown in form of graphs (without labels) and/or tables along with correct analysis or remarks. | The student completed all required tasks (90%-100%) with accurate MATLAB code and/or Simulink models. Correct outputs are shown in form of labeled graphs and/or tables along with correct analysis or remarks. | **30%** |
| **Following Directions** | The student clearly failed to follow the verbal and written instructions to successfully complete the lab | The student failed to follow the some of the verbal and written instructions to successfully complete all requirements of the lab | The student followed the verbal and written instructions to successfully complete requirements of the lab | **20%** |
| **Time Utilization** | The student failed to complete even part of the lab in the allotted amount of time | The student failed to complete the entire lab in the allotted amount of time | The student completed the lab in its entirety in the allotted amount of time | **20%** |

**OBJECTIVE:**

* Implement a logic in MATLAB to Multiplex three input voice signals at the transmitter end and Demultiplex and play them back at the Receiver end. Add random noise to the signal while propagating via the channel.

**LOGIC DIAGRAM:**

****

**Code:**

% Step 1

% Load Audio Files

[V1, Fs1] = audioread('V1.mp3');

[V2, Fs2] = audioread('V2.mp3');

[V3, Fs3] = audioread('V3.mp3');

[beep, Fs4] = audioread('beep.mp3');

passBand = 500;

V1 = V1(:, 1);

V2 = V2(:, 1);

V3 = V3(:, 1);

V1(135049:188056) = 0;

V3(113177:188056) = 0;

fprintf('\n\nsize of Voice 1 is %s\n', mat2str(size(V1)));

fprintf('size of Voice 2 is %s\n', mat2str(size(V2)));

fprintf('size of Voice 3 is %s\n', mat2str(size(V3)));

% Play them

sound(V1, Fs1);

pause(3.5);

sound(beep, Fs4);

pause(1);

sound(V3, Fs3);

pause(3);

sound(beep, Fs4);

pause(1);

sound(V2, Fs2);

pause(4);

sound(beep, Fs4);

pause(1);

% Step 2

% Plot the spectra of the signals as they arrive

V1Spec = fft(V1);

n1 = length(V1);

ts1 = 1 / Fs1;

f1 = (-n1/2:n1/2-1)\*(ts1/n1);

fshift1 = fftshift(V1Spec);

subplot(6,1,1)

plot(f1, abs(fshift1));

xlabel('Frequency (Hz)')

ylabel('Magnitude')

title('Spectrum of V1');

V3Spec = fft(V3);

n2 = length(V3);

ts2 = 1 / Fs3;

f2 = (-n2/2:n2/2-1)\*(ts2/n2);

fshift2 = fftshift(V3Spec);

subplot(6,1,2)

plot(f2, abs(fshift2));

xlabel('Frequency (Hz)')

ylabel('Magnitude')

title('Spectrum of V3');

V2Spec = fft(V2);

n3 = length(V2);

ts3 = 1 / Fs2;

f3 = (-n3/2:n3/2-1)\*(ts3/n3);

fshift3 = fftshift(V2Spec);

subplot(6,1,3)

plot(f3, abs(fshift3));

xlabel('Frequency (Hz)')

ylabel('Magnitude')

title('Spectrum of V2');

% Spectrum Analyzer V1

h = dsp.SpectrumAnalyzer();

h.SampleRate = Fs1;

h.PlotAsTwoSidedSpectrum = true;

h.Name = 'V1s Spectrum Analyzer';

h(V1Spec)

% Spectrum Analyzer tooba

t = dsp.SpectrumAnalyzer();

t.SampleRate = Fs3;

t.PlotAsTwoSidedSpectrum = true;

t.Name = 'V3s Spectrum Analyzer';

t(V3Spec)

% Spectrum Analyzer V2

s = dsp.SpectrumAnalyzer();

s.SampleRate = Fs2;

s.PlotAsTwoSidedSpectrum = true;

s.Name = 'V2s Spectrum Analyzer';

s(V2Spec)

% Step 3

% The signals are passed through a low pass filter and plotted

LPF\_H = lowpass(V1, passBand, Fs1);

LPF\_S = lowpass(V2, passBand, Fs2);

LPF\_s = lowpass(V3, passBand, Fs3);

% Plot after LPF

% Plot the spectra of the signals as they arrive

V1Spec = fft(LPF\_H);

n1 = length(LPF\_H);

ts1 = 1 / Fs1;

f1 = (-n1/2:n1/2-1)\*(ts1/n1);

fshift1 = fftshift(V1Spec);

subplot(6,1,4)

plot(f1, abs(fshift1));

xlabel('Frequency (Hz)')

ylabel('Magnitude')

title('Spectrum of V1 after LPF');

V3Spec = fft(LPF\_s);

n2 = length(LPF\_s);

ts2 = 1 / Fs3;

f2 = (-n2/2:n2/2-1)\*(ts2/n2);

fshift2 = fftshift(V3Spec);

subplot(6,1,5)

plot(f2, abs(fshift2));

xlabel('Frequency (Hz)')

ylabel('Magnitude')

title('Spectrum of V3 after LPF');

V2Spec = fft(LPF\_S);

n3 = length(LPF\_S);

ts3 = 1 / Fs2;

f3 = (-n3/2:n3/2-1)\*(ts3/n3);

fshift3 = fftshift(V2Spec);

subplot(6,1,6)

plot(f3, abs(fshift3));

xlabel('Frequency (Hz)')

ylabel('Magnitude')

title('Spectrum of V2 after LPF');

% Step 4

% Reproduce the signals after LPF

sound(LPF\_H, Fs1);

pause(3.5);

sound(beep, Fs4);

pause(1);

sound(LPF\_s, Fs3);

pause(3);

sound(beep, Fs4);

pause(1);

sound(LPF\_S, Fs2);

pause(4);

sound(beep, Fs4);

pause(1);

% Step 5

% The signals are modulated to different carriers

CF1 = 12000; % 12KHz

CF2 = 16000; % 16KHz

CF3 = 20000; % 20KHz

FMOD1 = fmmod(LPF\_H, CF1, Fs1, passBand);

FMOD2 = fmmod(LPF\_S, CF2, Fs2, passBand);

FMOD3 = fmmod(LPF\_s, CF3, Fs3, passBand);

% Step 6

% The modulated signals are filtered in the given band and added together

BP1 = bandpass(FMOD1,[8600 11700], Fs1); % 8.6-11.7 KHz

BP2 = bandpass(FMOD2,[12600 15700], Fs2); % 12.6-15.7 KHz

BP3 = bandpass(FMOD3,[16600 19700], Fs3); % 16.6-19.7 KHz

multiplexer = BP1 + BP2 + BP3;

% Step 7

% Some noise is added to the transmitted signal.

Noise = awgn(multiplexer, 10);

% Step 8

% Upon arrival each band is filtered.

SIG1 = FMOD3 + FMOD2;

demux\_signal1 = multiplexer - SIG1;

SIG2 = FMOD3 + FMOD1;

demux\_signal2 = multiplexer - SIG2;

SIG3 = FMOD1 + FMOD2;

demux\_signal3 = multiplexer - SIG3;

% Step 9

% Each recovered band is demodulated to return the signal to the baseband frequency.

DMOD1 = ssbdemod(demux\_signal1, CF1, Fs1);

DMOD2 = ssbdemod(demux\_signal2, CF2, Fs2);

DMOD3 = ssbdemod(demux\_signal3, CF3, Fs3);

% Step 10

% The recovered signal is passed through a low pass filter.

receivedV1 = lowpass(DMOD1, passBand, Fs1);

receivedV2 = lowpass(DMOD2, passBand, Fs2);

receivedV3 = lowpass(DMOD3, passBand, Fs3);

% Step 11

% Play the reproduced signal after transmission.

sound(receivedV1, Fs1);

pause(3.5);

sound(beep, Fs4);

pause(1);

sound(receivedV3, Fs3);

pause(3);

sound(beep, Fs4);

pause(1);

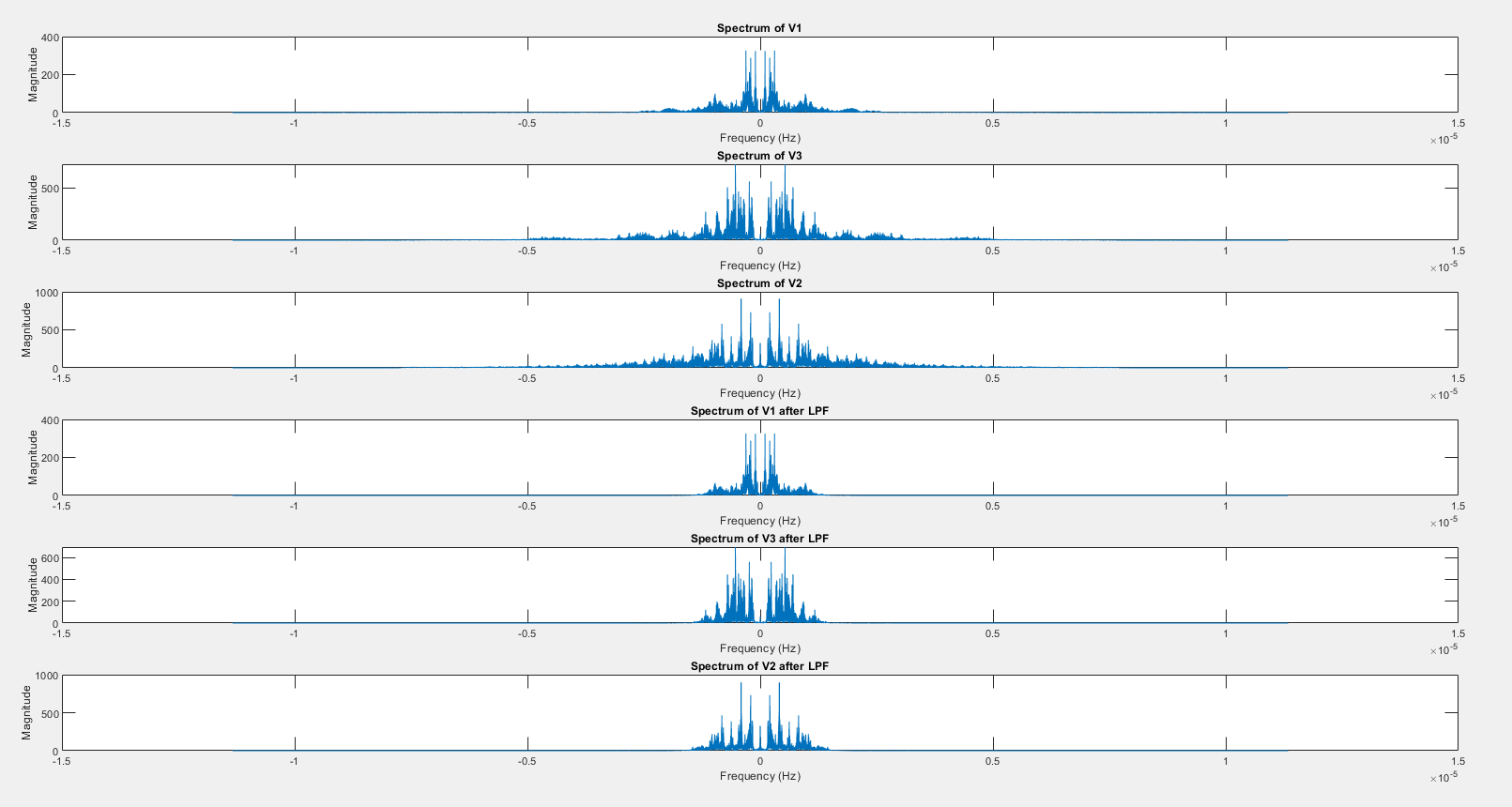
sound(receivedV2, Fs2);

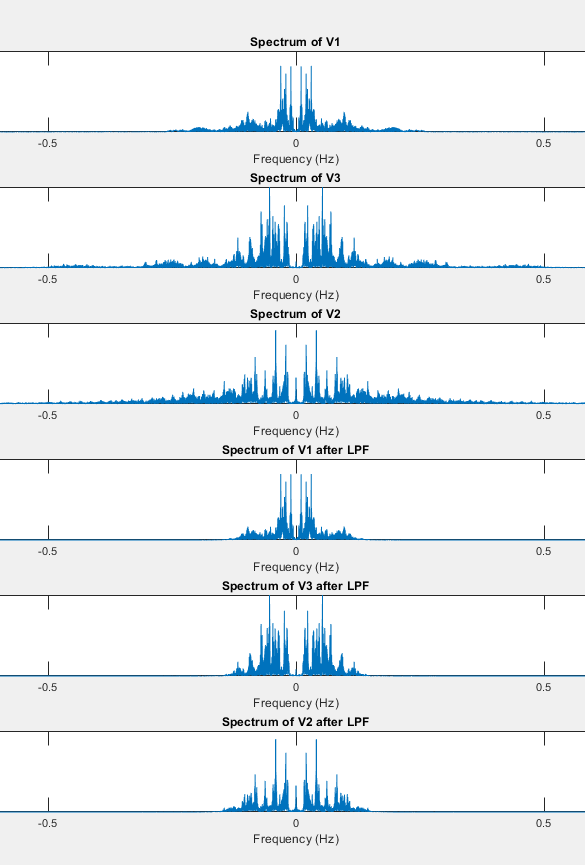
pause(4);

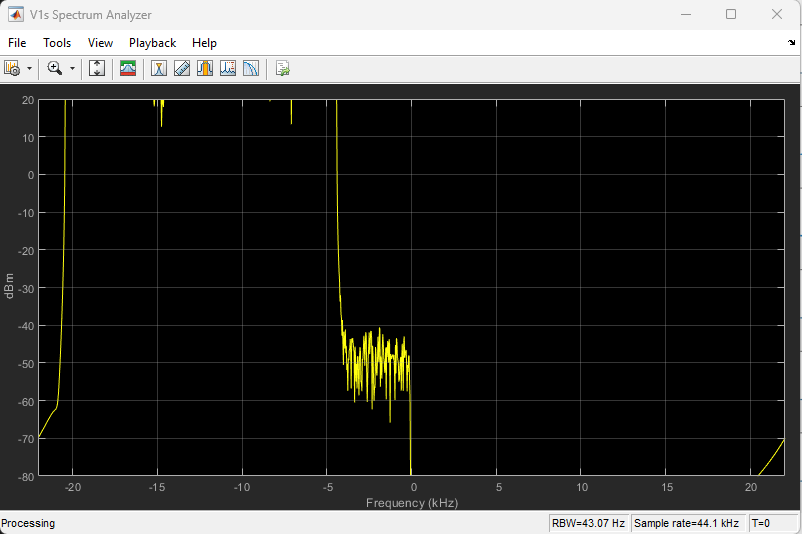
sound(beep, Fs4);

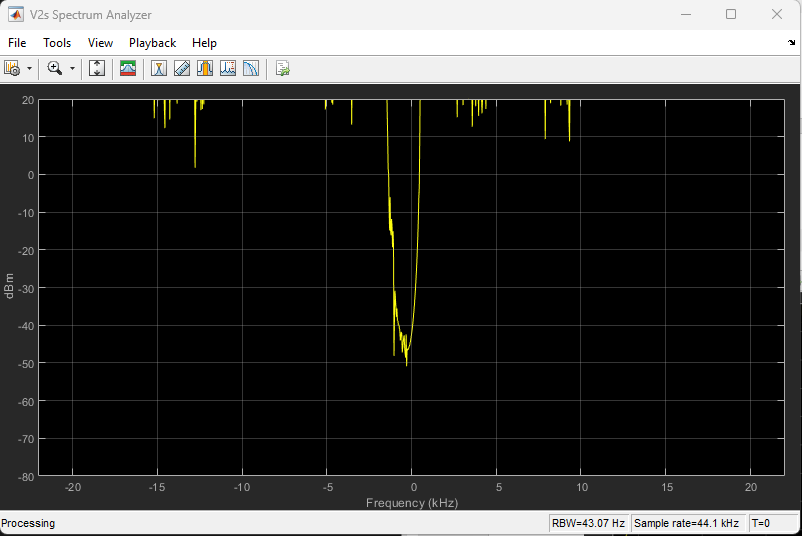
pause(1);

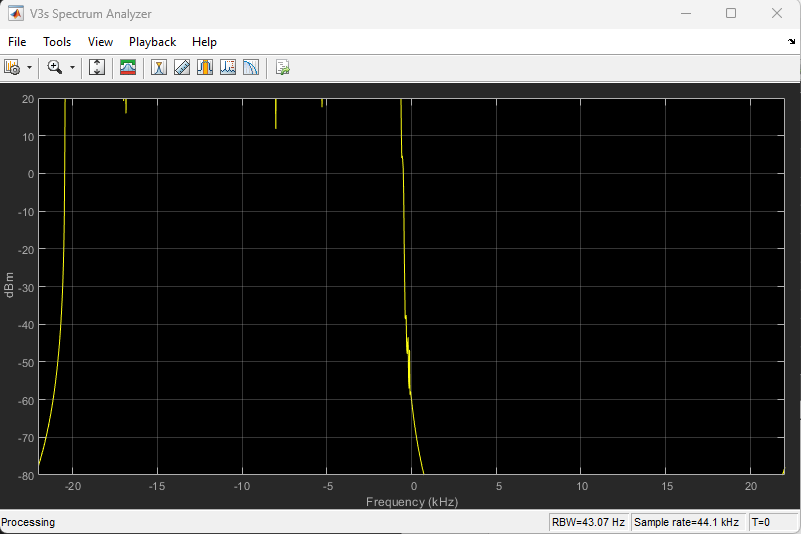
**Output:**

****









The demultiplexed voices are same as the ones that are multiplexed.