

EE-232 Signals & Systems

Task 2 Report

**Group Members**

1. **Amur Saqib Pal 288334**
2. **Junaid Ali 290927**
3. **Faiez Kazi 295848**
4. **Sannan Abbasi 293786**

**Steps Performed in Task 2**

1. Reading of Audio Signal finding its length and plotting it in MATLAB:

**Chart, line chart

Description automatically generated**

1. Finding the Frequency Spectrum of the signal and plotting it:

**A picture containing histogram

Description automatically generated**

1. Finding and Plotting the Compressed Signal:

Chart

Description automatically generated with low confidence

1. Plotted the Reconstructed Signal using Compressed Signal in Time Domain:

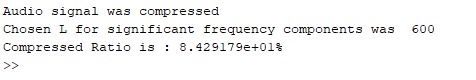
Chart, line chart

Description automatically generated

1. Calculated the Compressed Ratio and Distortions in the signals:

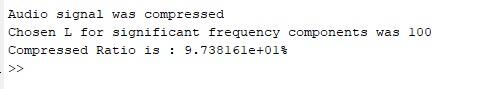
**Compression rate for L=600:**

Compression ratio = 100−(𝑙𝑒𝑛𝑔𝑡ℎ 𝑜𝑓 𝑠𝑖𝑔𝑛𝑖𝑓𝑖𝑐𝑎𝑛𝑡 𝑓𝑟𝑒𝑞𝑢𝑒𝑛𝑐𝑦 𝑐𝑜𝑚𝑝𝑜𝑛𝑒𝑛𝑡𝑠/𝑙𝑒𝑛𝑔𝑡ℎ 𝑜𝑓 𝑜𝑟𝑖𝑔𝑖𝑛𝑎𝑙 𝑠𝑖𝑔𝑛𝑎𝑙)(100)

****

**Compression rate for L=100:**

Compression ratio = 100−(𝑙𝑒𝑛𝑔𝑡ℎ 𝑜𝑓 𝑠𝑖𝑔𝑛𝑖𝑓𝑖𝑐𝑎𝑛𝑡 𝑓𝑟𝑒𝑞𝑢𝑒𝑛𝑐𝑦 𝑐𝑜𝑚𝑝𝑜𝑛𝑒𝑛𝑡𝑠/𝑙𝑒𝑛𝑔𝑡ℎ 𝑜𝑓 𝑜𝑟𝑖𝑔𝑖𝑛𝑎𝑙 𝑠𝑖𝑔𝑛𝑎𝑙)(100)

****

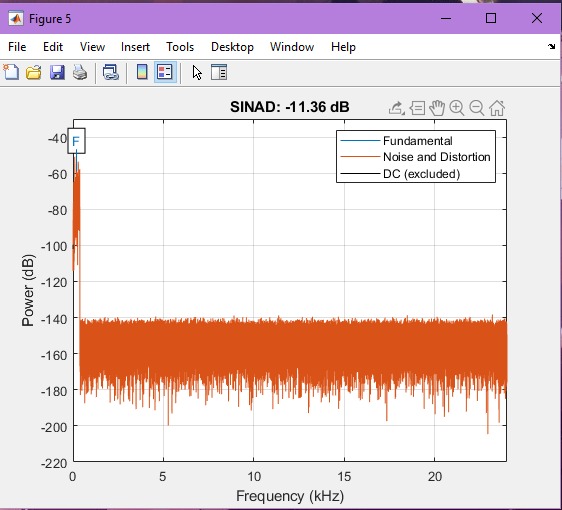
To quantify signal distortion in our reconstructed signal, we used **SINAD,** a function that returns signal to noise distortion ratio.

General syntax is:

Sinad(x, Fs)

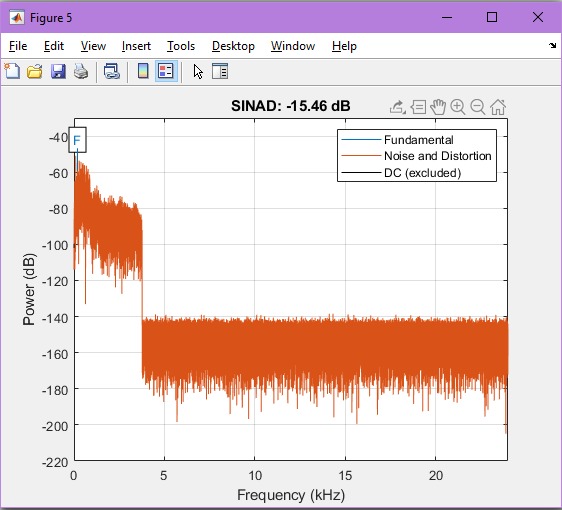
Where x is the input signal(reconstructed audio) and Fs is the sampling rate of the reconstructed signal. The graphs have been plotted and their trend observed in the next part where the DC level and fundamental are excluded as no noise computation is required.

As the value of L is increased, we observe the distortion becomes lesser and vice versa



Here as we can see, for L=60 it is visible from the graph that there is a large amount of distortion. To reduce this distortion, we increase the value of L (our significant frequency components).

Let’s say we increase it up to 600. The resulting distortion plot is shown below:



As we can see, the signal has less distortion and is more understandable.

Value of SINAD is lowered, it is visible from the graph that distortion has surely decreased.

We can increase L, up to a limit where most of its significant components are included in the signal, at this point we get a clear audio, and above that point won’t make a very noticeable difference.

**Division of Work:**

Task 2 of the project was completed by Junaid and Faiez. Both worked together in compiling and formatting the code. They both were responsible for completing the Task 2 report as well.

**Code**

clc;

close all;

clear all;

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Reading of Audio Signal\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[y,Fs]=audioread('samplewave.wav');

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Finding the length of Signal and Plotting it\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sl = length(y); % length of the wave

Slength = sl/Fs; % time for input signal

t = linspace(0, Slength, sl);

figure;

plot(t,y); %graphing the signal

xlabel('Time[s]')

ylabel('Amplitude')

title('Original Audio Signal')

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Finding and Plotting the Frequency Spectrum of the Audio Signal\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

N = length(y); % Length of vector y, number of samples

Y = fft(y,N); % Fourier transform of y

Yn= fftshift(Y);

w = ((-N/2:N/2-1)\*(Fs/N)); % Frequency vector

F = w/(2\*pi); % Angular frequency vector

Fn = Fs/2; % Nyquist Frequency

magnitudeY = abs(Yn); % Magnitude of the FFT

phaseY = unwrap(angle(Yn)); % Phase of the FFT

figure;

plot(F, magnitudeY);

ylabel('Magnitude, dB');

xlabel('Frequency, Hz');

title('Audio Wave Frequency Graph');

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Finding and Plotting the Compressed Signal\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

L=600; %Significant Frequency Component Value taken by US!

start = find(ceil(F) == -L);%this would return a matrix

stop = find(floor(F) == L);

lowindex = start(length(start)); %last value is closest to required frequency

upindex = stop(1); %first value is closest to required values

compressed\_y=zeros(1,N);

compressed\_y(lowindex:upindex)=Yn(lowindex:upindex);

figure;

plot(F,abs(compressed\_y));

ylabel('Magnitude, dB');

xlabel('Frequency, Hz');

title('Compressed Frequency Signal')

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Plotting the Reconstructred Signal using Comressed Signal in Time Domain\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

figure;

convert = real(ifft(fftshift(compressed\_y)));

plot(t,convert);

xlabel('Time[s]')

ylabel('Signal')

title('Reconstructed Signal using the Compressed Signal in Time Domain')

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Calculating the Compressed Ratio\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

compressionratio = 100 - ((upindex-lowindex)/N)\*100;

fprintf('Audio signal was compressed');

fprintf('\nChosen L for significant frequency components was %d',L);

fprintf('\nCompressed Ratio is : %d%%\n',compressionratio);

audiowrite('compressedsample.wav',convert,Fs);

[newy,newFs]=audioread('compressedsample.wav');

sound(newy,newFs);

figure;

sinad(newy,newFs)