Lab 1: Sampling and Quantization (Audio)

Rami Saad
Section 4

***Abstract* - In this lab the effects of manipulating the sampling rate and quantization levels were observed. In the case of down sampling the effects were in line with the theory, when less samples were taken the audio quality suffered. For both Uniform and μ Law Quantization it was observed that when less bits were allocated, the audio quality also suffered.**

Sampling; Quantization; Resolution; Levels; Function

I. Introduction

The purpose of this lab was to understand the principles of sampling and quantization with an audio signal. More specifically, the effects of rate conversion and quantization upon an audio signal. An input signal was down sampled and the effects of increasingly removing samples was noted. Another experiment on decreasing the number of bits alloted to each sample was also conducting using both Uniform and μ Law quantization.

The lab was primarily conducted using Matlab, rather than OpenCV as there is not much support for audio signal processing with OpenCV. Matlab on the other hand has a suite of built in function to handle audio processing such as audioread() and audioinfo().

II. Theory

Given an input signal x[n], which can be down sampled if decreased by the integer factor M > 1, keeping every M$^{\text{th}}$ signal. This gives the equation

$$y[n] = x[n \cdot M] \tag{1}$$

Where $y[n]$ is the newly outputted down sampled signal and $[n \cdot M]$ relates to every (M - 1) samples being discarded.

One type of quantization used in this lab was uniform quantization. With uniform quantization the quantization levels are equally spaced among the amplitude range of the signal. The same number of bits are used to represent each level and there are 2^{N} quantization levels where N is the number of bits available.

A midrise quantizer was used in this lab to handle all quantizations rather than a midtread quantizer because the midrise allows for 2^{N} number of levels compared to the 2^{N} - 1 levels of a midtread quantizer. The midtread quantizer employs the following equation for Quantization Level (2) and Reconstructed Signal (3)

$$Q(x) = floor(\frac{x}{q}) + \frac{1}{2} \tag{2}$$

$$S(Q) = q \cdot Q(x) \tag{3}$$

Another type of Quantization used in this lab was μ Law quantization. μ Law quantization was used to map the input values to a non-linear space where lower signals are assigned more bits. The signal is compressed (4) then expanded (5) using these equations

$$y = X_{max}\frac{log[1+\mu\frac{|x|}{X_{max}}]}{log[1+\mu]}sign[x] \tag{4}$$

$$x = \frac{X_{max}}{\mu}(10^{\frac{log(1+\mu)}{X_{max}}|y|} - 1)sign(y) \tag{5}$$

First the signal is compressed using equation (4) and then the transformed values are then quantized using uniform quantization seen in equation (2) and (3). Finally, the signal is expanded using equation (5).

III. Methodology

Audio File Properties

The ELE725_lab1.wav file was read and assigned to two variables y and Fs using the builtin audioread() function in Matlab. Then using another builtin Matlab function audioinfo() to retrieve audio information which was assigned to a struct variable named info.

By navigating to the Matlab workspace the information such as the filename, compression method, number of channel, sample rate, total samples, duration, and bits per sample. Using this information the bit rate and the file size of the wav file are calculated in the results section.

Sampling

All down sampling performed in this lab utilizes the DownSample.m function. This function takes in 3 parameters, inFile, which is the name of the wav file, How many samples to remove which is an unsigned integer named N, and a flag, pf, that when high uses the decimate() function to reduce samples and when low simply keeps every N_{th} sample and discards the rest. The output of this function is stored in a variable named outFile which contains the filename of the newly created down sampled wav file for the purpose of being replayed in the main Matlab script.

An attempt at restoring the down sampled audio is made with the interp1() a built Matlab function. This function is passed 3 parameters, the total number of down samples, the array of the down sampled audio data, and the number of interpolated samples needed.

Quantization

The first Matlab quantization function created in this lab is the UniformQuant.m function. This function takes in 2 parameters, inFile which is the name of the wav file, and N the number of quantization levels. First the maximum value of the audio signal data is found using matlabs max() function and then using the equation (6) to retrieve step size

$$q = \frac{2 \cdot vMax}{N^2} \tag{6}$$

equations (2) and (3) can then be used to perform a uniform midrise quatization of the audio data. A new wav file is created for the uniform quantized audio signal its name is stored in the output variable outFile.

In the case of the second quantization function MuLawQuant.m which is aptly named to employ a μ Law quantization algorithm. This function takes in 3 parameters, inFile which is the name of the wav file, N, the number of quantization levels and Mu which is the compression parameter. This function uses the same algorithm as described in the Theory section only repeated twice for each of the two audio channels. A new wav file is created for the μ Law quantized audio signal its name is stored in the output variable outFile.

The inbuilt Matlab function immse() was used to calculate the MSE error over all the samples for both Uniform and μLaw Quantizations. The newly created arrays were passed in to the function along with the original audio data.

IV. Results

Audio File Properties

Bit Rate Calculation:

$$BitRate = Fs \cdot N \cdot NChannels$$
$$BitRate = 44100 \cdot 16 \cdot 2$$
$$BitRate = 1.4Mbs$$

File Size Calculation:

$$FileSize = NChannels \cdot TotalSamples \cdot N$$
$$FileSize = 2 \cdot 116736 \cdot 16$$
$$FileSize = 456KB$$

Sampling

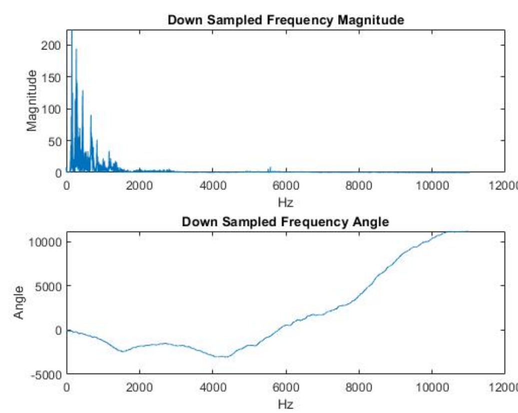
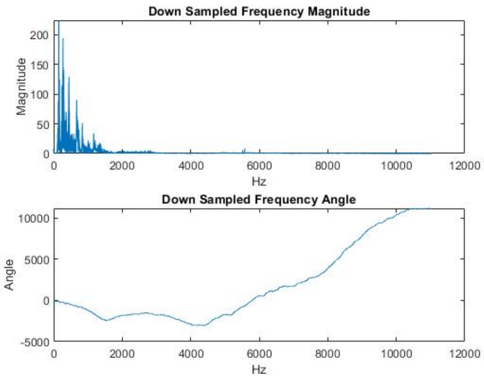
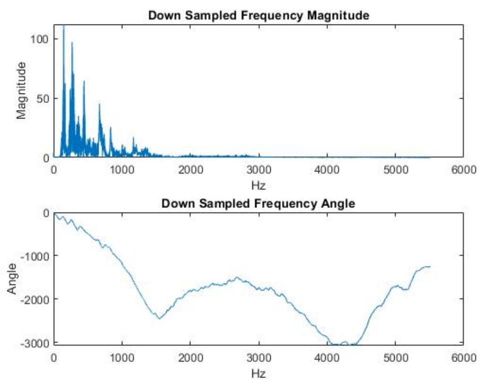


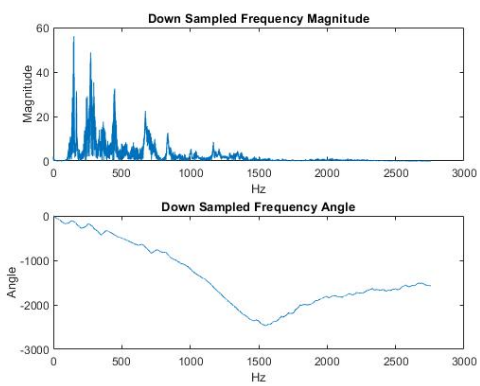
Figure 1: Original Sample Frequency Data



(a) N = 2

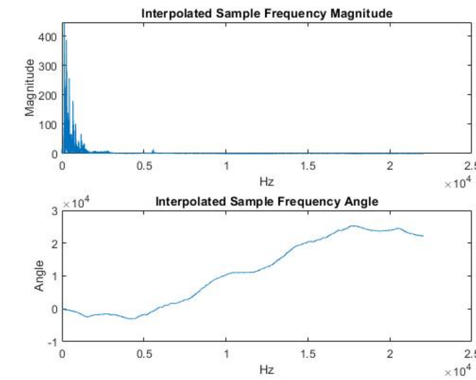


(b) N = 4

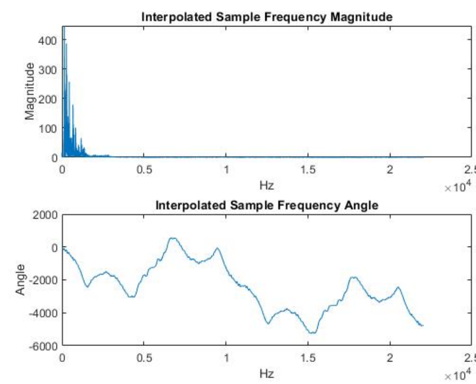


(c) N = 8

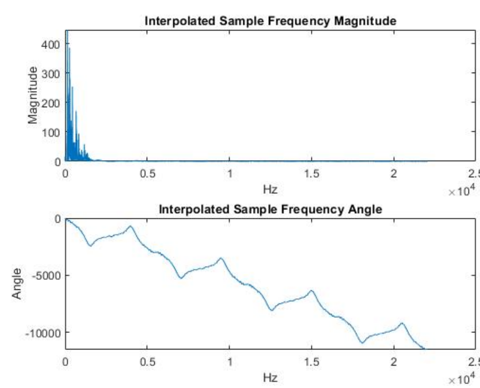
Figure 2: Down Sampled Frequency Data



(a) N = 2



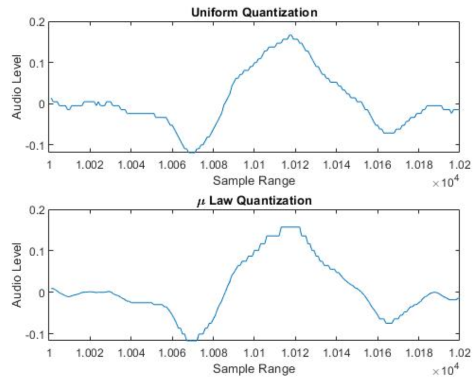
(b) N = 4



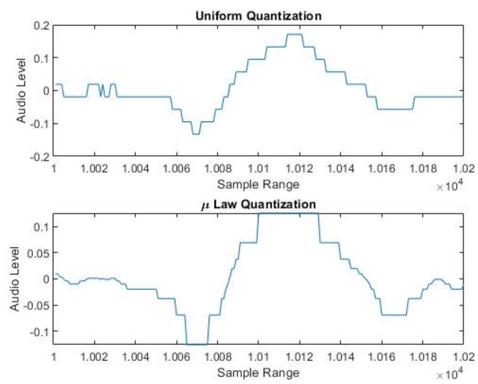
(c) N = 8

Figure 3: Interpolated Sample Frequency Data

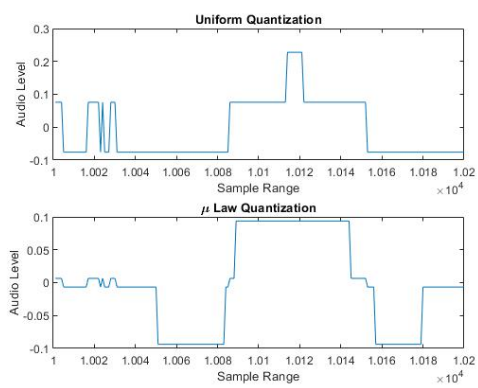
Quantization



(a) N = 8



(b) N = 4



(c) N = 2

Figure 4: Uniform and μ law Small Sample Comparison

Table 1: MSE error over Total Samples

Quantization	**N = 8**	**N = 4**	**N = 2**
Uniform	$8.79 \cdot 10^{-6}$	$1.70 \cdot 10^{-4}$	$3.45 \cdot 10^{-3}$
μ Law	$3.63 \cdot 10^{-6}$	$6.10 \cdot 10^{-5}$	$7.49 \cdot 10^{-4}$

V. Discussion

Audio File Properties

The calculated file size and the file size on the operating system match as they are both 456 KB. The size on disk have a slight variation however due to how the file system table allocates memory.

Sampling

As the integer factor of the down samples increase the audio quality gets worse when compared to the original audio. The case is similar for the reconstructed version as the interpolation only estimates the values between the down sampled signal so much of the higher frequency data can not be recovered as shown in figures 3a, 3b, and 3c.

Quantization

The μ Law sound quality is better when compared to that of the Uniform Quantization. The way that humans perceive how loud something sounds is logarithmic meaning for a sound to be twice as loud it would need to be an order of magnitude higher. This fact can be exploited by giving more resolution to the lower value levels and less to higher value levels. This audio quality difference is also reflected in Table 1 as the MSE error for the μ Law is lower for every case.

VI. Conclusion

In this lab, the principles of sampling and quantization with an audio signal were observed. As an audio file was down sampled by a greater integer factor, the audio quality was reduced and even trying to reconstruct the data through interpolation was ineffective as more high frequency audio data was lost.

In the case of quantization it was observed that as more bits were allocated per sample the audio quality increase until there was the same amount of bits per sample as the original audio. Furthermore, it was confirmed experimentally that μ Law quantization was more effective at quantizing audio than uniform quantization as humans are more likely to notice a difference between lower audio levels than higher ones.

References

[1] "Lab 1: Sampling and Quantization (Audio)," D2L, 2020.

[2] P. Havaldar and Medioni Gerard, Multimedia systems: algorithms, standards, and industry practices. Boston, MA: Course Technology Cengage Learning, 2010.