ECE3101L: Signals and Systems Laboratory

Spring 2025

Lab 2: Sampling, Quantization, Error

Sam Duong

Sunny Nguyen

March 19, 2025

Table of Contents

1. Introduction…………………………………………………………………………………..3

1.1 Executive Summary……………………………………………………………….….3

1.2 Objective………………………………………………………………………………3

2. Sampling……………………………………………………………………………………..4

2.1 Theory…………………………………………………………………………………4

2.1.1 Sampling Frequency…………………………………………………...4

2.1.2 Dirac-Delta, Aliases……………………………………………………4

2.2 Simulated Data…………………………………………………………………….….6

2.2.1 Nyquist Sampling……………………………………………………...6

2.2.2 Filter Design Requirements……………………………………………9

2.2.3 Aliasing……………………………………………………………….10

3. Quantization………………………………………………………………………………...12

3.1 Theory………………………………………………………………………………..12

3.1.1 Quantization Algorithm Design………………………………………12

3.2 Simulated Data……………………………………………………………………….12

3.2.1 Image Quantization…………………………………………………...13

3.2.2 Audio Quantization…………………………………………………...15

4. Error…………………………………………………………………………………………...18

4.1 Theory………………………………………………………………………………..18

4.1.1 Quantization Error……………………………………………………18

4.1.2 Correlation……………………………………………………………18

4.1.3 Signal-to-Noise Ratio………………………………………………...18

4.2 Simulated Data……………………………………………………………………….19

4.2.1 Quantization Error……………………………………………………19

4.2.2 Correlation……………………………………………………………20

4.2.3 Distortion Curves……………………………………………………..23

5. Conclusion…………………………………………………………………………………..23

1. Introduction

1.1 Executive Summary

The digitization of analog signals is crucial in applications of storage, transmission, and processing in electronics. The Nyquist-Shannon sampling theorem where sampling frequency $f_s \geq 2B$ allows for evenly spaced measurements based on the length of the signal's bandwidth. This sampling rate must always be at least twice the bandwidth for proper performance, avoiding aliasing errors; the lower the sampling rate, the higher the chance of signal distortion. Sampling a signal creates an infinite set of aliases centered around the signal's harmonic frequencies, which are filtered out using a low pass filter, with the corner frequency f_s being half of f_s.

Quantization is the mapping of continuous infinite values to a smaller set of discrete finite values. The step size determines the approximation of the digital representation of the signal; the larger the step size, the less accurate the approximation, resulting in a lower signal-to-noise ratio (SNR).

Signal-to-noise ratio is the comparison of a signal's strength to its background noise. In signal processing, this ratio must be considered in sampling and quantization in order to not completely distort a signal.

The best way to observe these techniques is through audio and image processing, where the sampling and quantization of a certain image or audio file will manipulate the perceived quality and resolution.

1.2 Objective

In this lab, we utilize the Nyquist sampling rate to sample an analog signal to generate its digital conversion, and then pass it through a low pass filter to reconstruct the original signal as closely as possible. We then quantize a set of image and audio files and observe how it affects the quality and resolution, comparing different algorithms. Finally, we perform error analysis on the quantization techniques, comparing the difference of the true value of the analog signal with the quantized value. This is all done through Simulink and MATLAB code.

2. Sampling

2.1 Theory

2.1.1 Sampling Frequency

Sampling takes an analog signal and transforms it into a digital signal - or, turns a continuous input into a discrete output. This is achieved by measuring the value of the signal at a certain point in time T_s, and measured again repeatedly at every integer multiple of T_s. This is known as the time step

$$T_s = \frac{1}{f_s},$$

where sampling frequency f_s is at least double the bandwidth of the input signal to ensure the integrity of the discrete information sampled from the continuous signal, as dictated by the Nyquist-Shannon sampling theorem

$$f_s \geq 2B.$$

For example, for a signal of form

$$2.5cos(2\pi 3000t),$$

our Nyquist rate will be at least double that of the signal's fundamental frequency, and thus $f_s \geq$ 6kHz. For functions made up of multiple sinusoids, such as

$$2.5cos(2\pi 300t) + 2cos(2\pi 700t) + 2cos(2\pi 1200t),$$

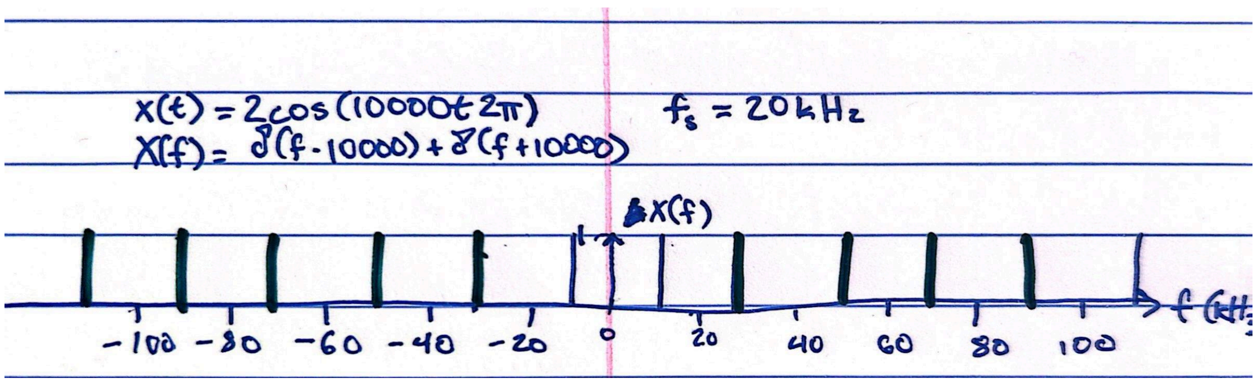
Our Nyquist rate will be at least double that of the function's highest frequency within the signal, 1200 Hz. Thus our Nyquist rate f_s is $\geq$ 2.4kHz.

2.2.2 Dirac-Delta Function, Aliases

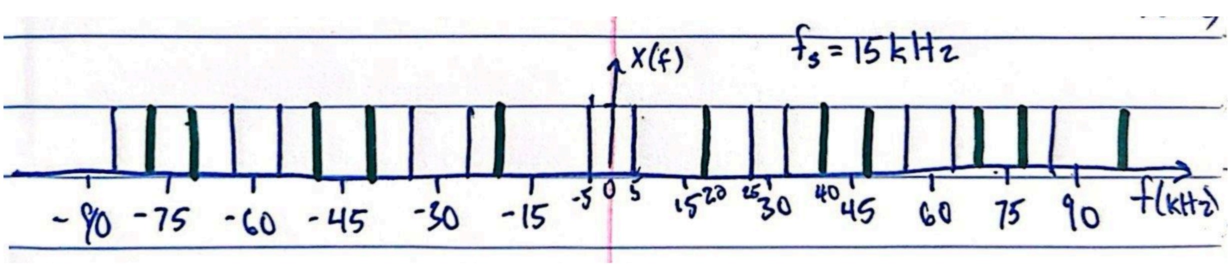
In an ideal environment, sampling is done with the dirac-delta function, a function in which the output is an impulse at 0. In applications of sampling in the ideal environment, this dirac delta is multiplied with an input signal, and these impulses will be shifted from 0 based on the signal's frequency. Recall that for a sinusoidal signal, the Fourier Transform of a cosine signal is

$$X(f) = \frac{A}{2}\left[\delta\left(f - f_c\right) + \delta\left(f + f_c\right)\right].$$

The sinusoidal impulse of the frequency domain will appear in a set of two, centered around multiple integers of the determined sampling frequency **at an increment of the sinusoidal input's fundamental frequency**. For example, suppose that a message signal of form x(t) = $2cos(2\pi 10000t)$ is sampled at a frequency of f_s = 20kHz. Following the sampling procedure, we will see that a set of "double spikes," which are the impulses of the Fourier transform of the message signal, $X(f) = [\delta(f - 10000) + \delta(f + 10000)]$, are centered around multiple integers of 20kHz spaced out by 10kHz, the fundamental frequency of the signal. Ergo, we find the impulses occurring at $\pm\ 10kHz, \pm\ 30kHz,\ \ \pm\ 50kHz$...



The impulses found at the integer multiples of the sampling frequency are known as aliases. The process of reconstructing the original message from a sampled signal involves filtering out these aliases through a low pass filter to leave only the original impulse representing the input sinusoidal signal. In the example above, we see that aliases overlap each starting at 30kHz and 50kHz, but they do not overlap at 10kHz. This shows that signal reconstruction is possible by applying a low pass filter with a corner frequency that cuts off all aliases after 20kHz. If we choose a sampling frequency that does not follow the Nyquist-Shannon theorem, say, 15kHz, we can see that the aliases will be centered around multiple integers of 15kHz, while still being spaced 10kHz apart.



As shown above, the aliases are spaced 10kHz from each multiple integer of sampling frequency 15kHz, which means there is a set of impulses at 5kHz and -5kHz – half of the fundamental frequency of the original signal input. This is known as aliasing, where due to the nature of the undersampling sampling frequency of 15kHz, the reconstructed message signal will be misrepresentative of the original message signal.

In a practical setting, no such impulse generator exists, to which we must make do with a square wave generator of a small duty cycle. Recall that the Fourier Transform of a square wave is $X_n = (hd)sinc(nd)$. Using this square wave as our sampler will approximate discrete values of our continuous sinusoid (Note: as the duty cycle gets smaller and smaller, the harmonics of the square wave will also decrease. As it approaches 0, we will find that this sinc function will behave more and more impulse-like, producing our ideal Dirac-delta sampler).

2.2 Simulated Data

2.2.1 Nyquist Sampling

Using Simulink, we multiply a sine wave against a sampling pulse train. Because there is no such thing as an impulse generator, we must generate a square pulse train with a duty cycle as small as possible.

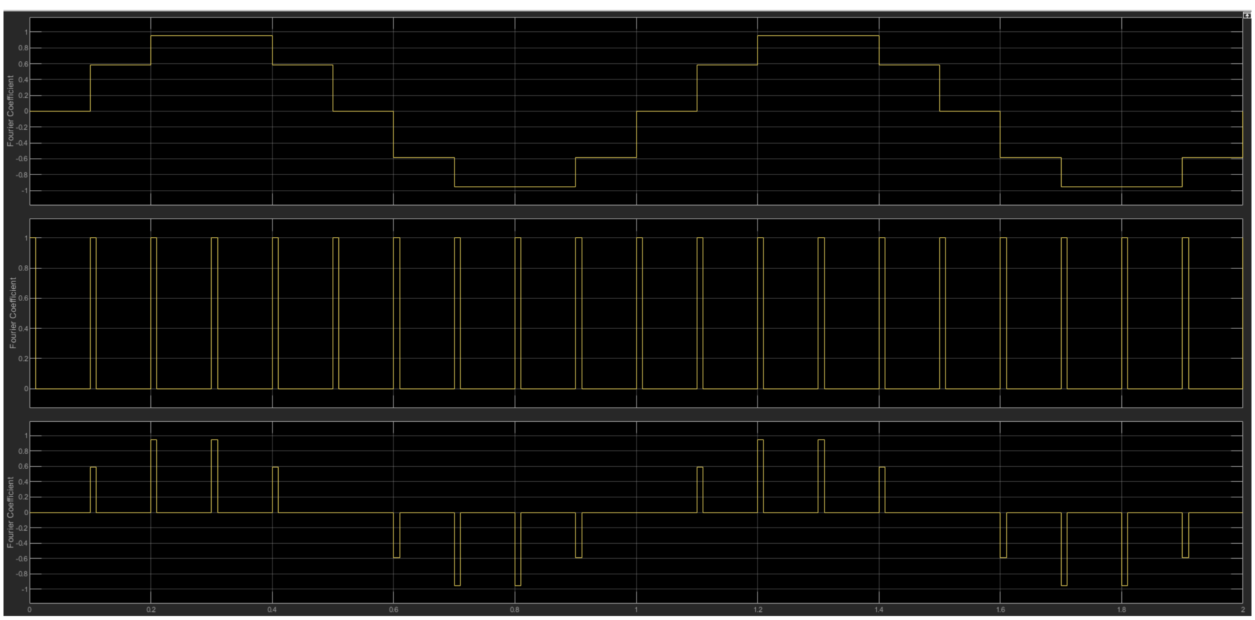
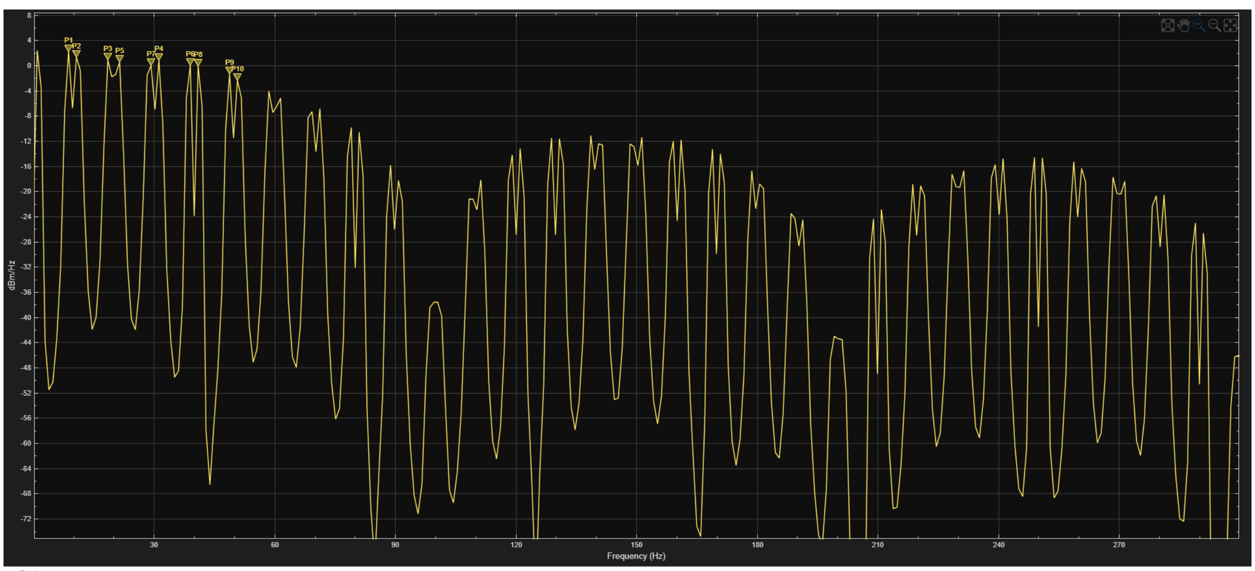


Figure 1, 1 Hz Sine Wave (top), 10 Hz 10% Duty Cycle Pulse Train (middle), Resultant Sampled Sine Wave (bottom)

Here, we can see a 1 Hz sine wave as well as a 10 Hz pulse train with a duty cycle of 10%. As this pulse train represents our sampling signal, its fundamental frequency is the sampling frequency, 10 Hz. As there is an apparent pulse width in our sampling signal, we use a sine wave that makes its shape in steps. This way, we can sample our signal properly while staying as accurate as possible. The sampling frequency 10 Hz follows the Nyquist-Shannon theorem where f_s = 10B, since our baseband is 1 Hz from the original sine wave. 10 Hz is 5 times the minimum sampling frequency required, so we should expect the sampled signal to output an accurate waveform. The bottom waveform is the product of these two signals. This is the resultant sampled sine wave, where each sample point in the signal represents the value of the original sine wave at the exact moment that pulse occurred. We can see that the pulse train takes on a shape relatively similar to the sine wave. This brings us to the core of sampling: when we digitize an analog signal, we take discrete measurements of the signal that while not precisely like a continuous signal, we take on the fundamental shape and representation of the analog signal in the discrete domain. A higher sampling frequency will give us a more "complete" representation of the signal, but requires more processing in a practical environment.



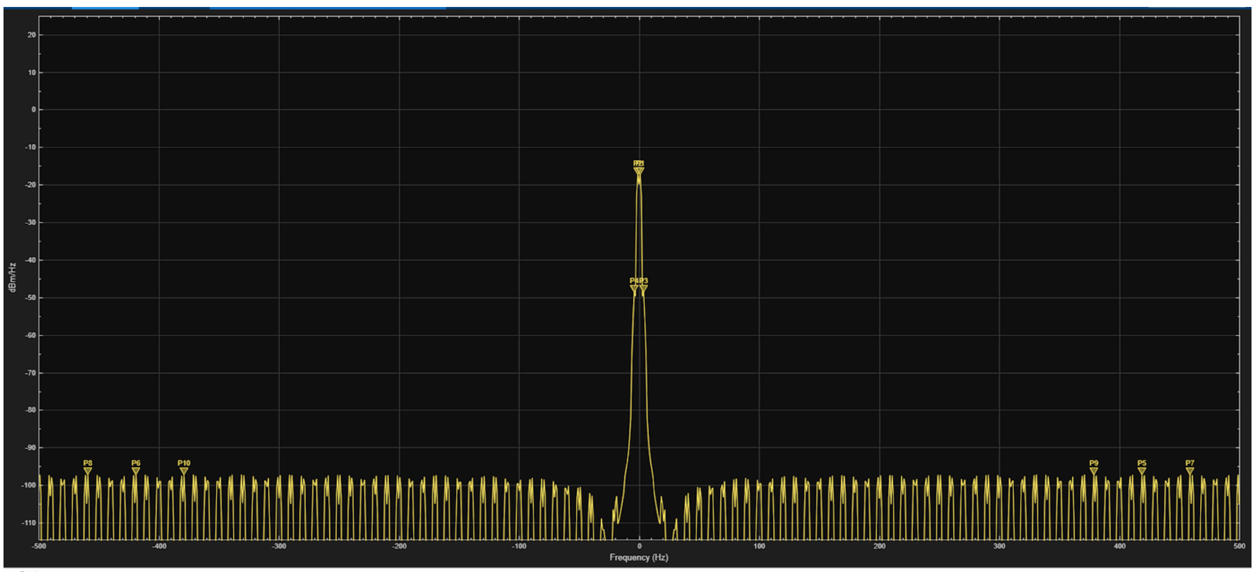
Peaks

Product	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Frequency (Hz)	8.7891	10.7422	18.5547	31.2500	21.4844	39.0625	29.2969	41.0156	48.8281	50.7813
Power Density (dBm/Hz)	2.1627	1.2800	0.8527	0.7832	0.5139	0.0049	-0.0200	-0.1226	-1.3735	-2.3937

Figure 2, Frequency Domain of the Sampled Signal

From this windowing of 240 Hz, we can see that the sampled signal's amplitude spectrum follows the sinc function of the sampling pulse train, where every 10th harmonic is a null amplitude. There are two peaks centered around every integer multiple of 10 Hz, spaced at 1 Hz each, as discussed in theory.

In signal reconstruction, the sampled signal is passed through a low pass filter with cutoff frequency f_c set at half of the sampling rate. In the case of the sampled signal shown in **Figure 1**, the cutoff frequency is 5Hz. This will filter out the aliases existing at the integer multiples of the 10 Hz sampling frequency and pass the aliases existing at the original signal's fundamental frequency, 1 Hz.



Lowpass Filter	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Frequency (Hz)	0.9766	-0.9766	3.9063	-3.9063	418.9453	-418.9453	458.9844	-458.9844	378.9063	-378.9063
Power Density (dBm/Hz)	-17.2918	-17.2918	-48.5977	-48.5977	-97.1995	-97.1995	-97.2068	-97.2068	-97.2173	-97.2173

Figure 3, Filtered Sample Circuit Amplitude Spectrum

The filter, with a cutoff frequency of 5 Hz, filters out the aliases at the higher frequencies of the sampled signal's amplitude spectrum. Figure 3 displays the filtered aliases, leaving only the set of double spikes centered around 0 Hz at -1 Hz and 1 Hz. The resultant time-domain displays an approximation of the original signal.

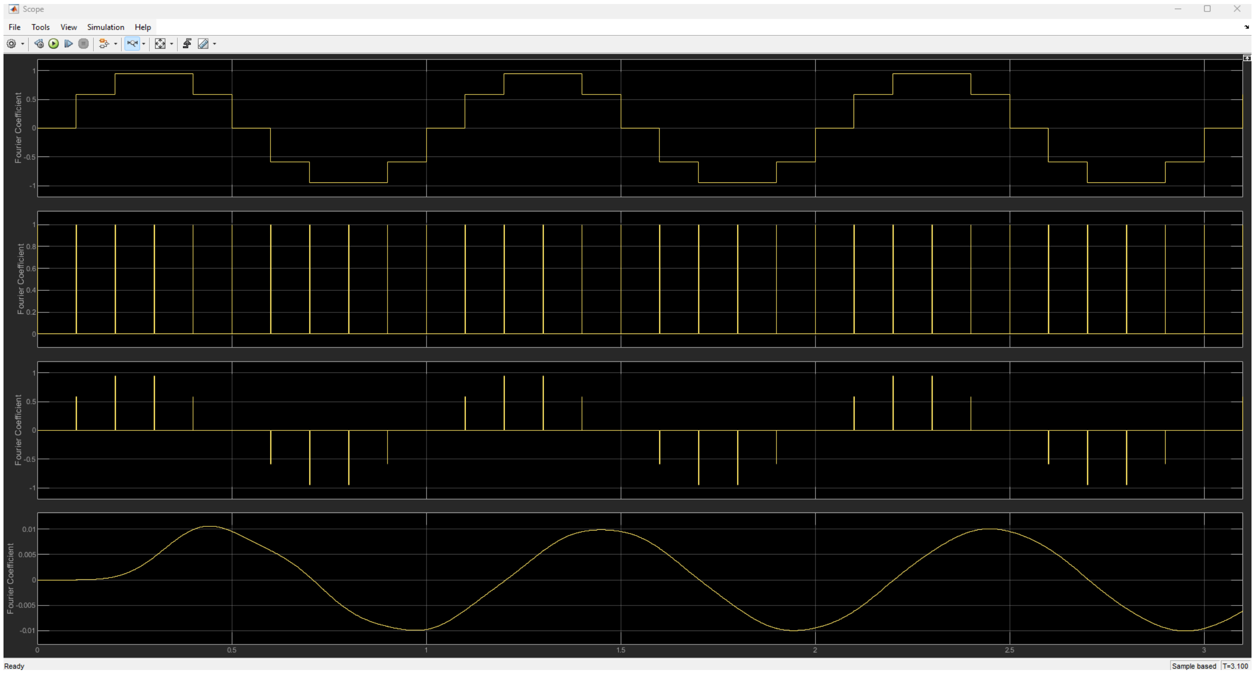


Figure 4, Filtered Sample Signal: Reconstructed Signal

2.2.2 Filter Design Requirements

Setting an appropriate corner frequency is crucial in signal reconstruction. If the corner frequency is set to a value less than the fundamental frequency of the original signal, we will find that our reconstructed signal in the time domain will not be as close to the original signal as a filter with the proper corner frequency.

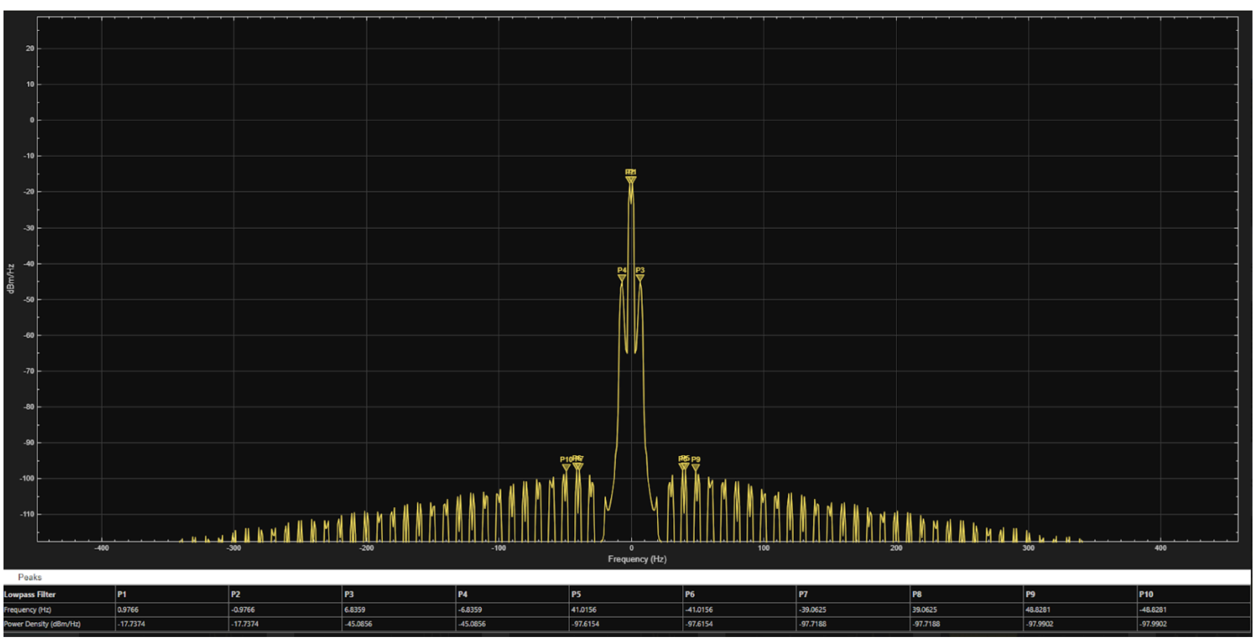


Figure 5, Filtered Sampled Signal, ω_c = 0.8 Hz

The amplitude spectrum in the figure above shows that due to an inappropriate corner frequency, not all of the aliases in the higher harmonics have been completely filtered out. This in turn will affect the way the reconstructed signal will appear in the time domain. Figure 6 (next page) shows that the reconstruction appears to be bumpy. As more and more aliases are passed by the filter, the time-domain reconstruction will appear more and more like the sampled signal in Figure 1.

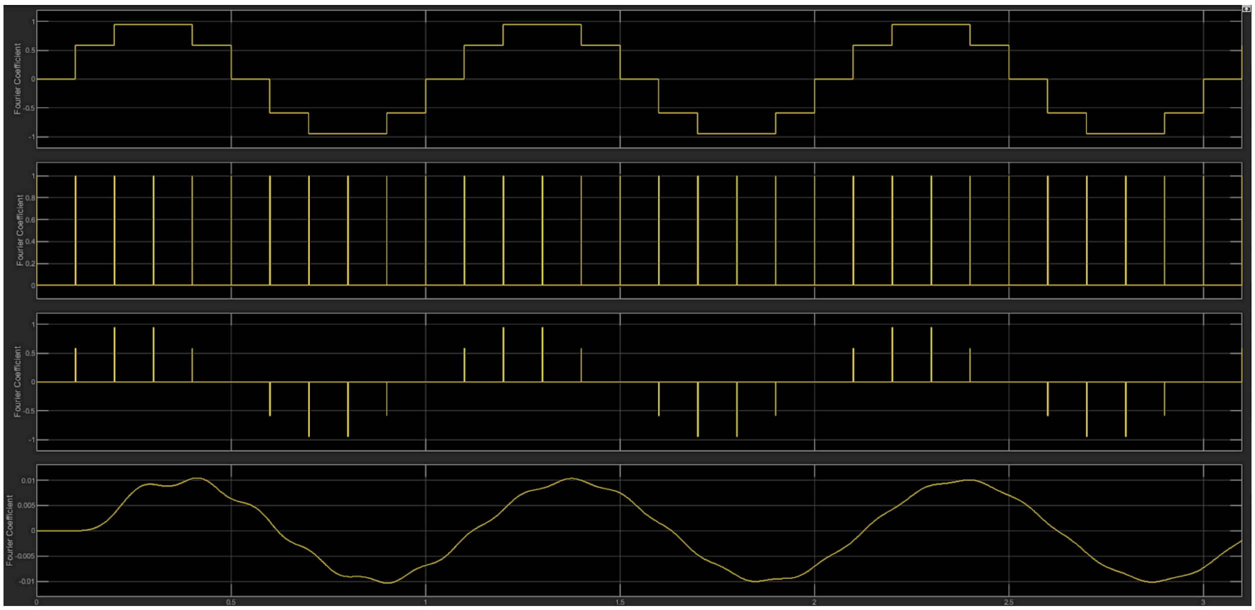


Figure 6, Reconstructed Signal, ω_c = 0.8 Hz

2.2.3 Aliasing

As mentioned earlier, aliasing is a result of undersampling, where a sampling pulse train does not follow the Nyquist-Shannon sampling theorem. The undersampling causes aliases in the higher frequencies to be misrepresented as lower frequencies, and thus in reconstruction generates an inaccurate signal.

Suppose that we sample an 8 Hz sine wave with the same 10 Hz 10% duty cycle pulse train we used earlier to sample our 1 Hz signal. Aliasing occurs and the resultant sampled signal is misrepresentative of our original sine wave.

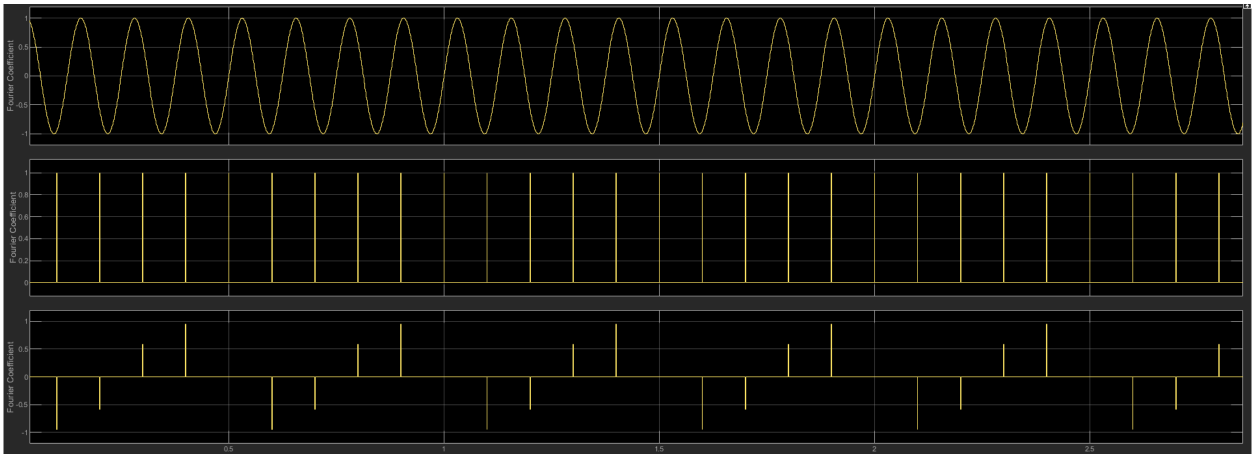


Figure 7, 8 Hz Sine Wave Sampled with 10 Hz Pulse Train

In the frequency domain, the sampled aliases appear at a much lower frequency than what the original signal is. Recall that multiplication in the time domain translates to convolution in the frequency domain. From our sampling rate f_s, we know that each alias will be centered around integer multiples of 10 Hz. Each alias will be spaced ±8 Hz from these multiples. As a result, Figure 8 shows aliases at ±2 Hz, ±18 Hz, ±22Hz, and so on and so forth. Thus, our inappropriate sampling rate creates a distorted sampled signal.

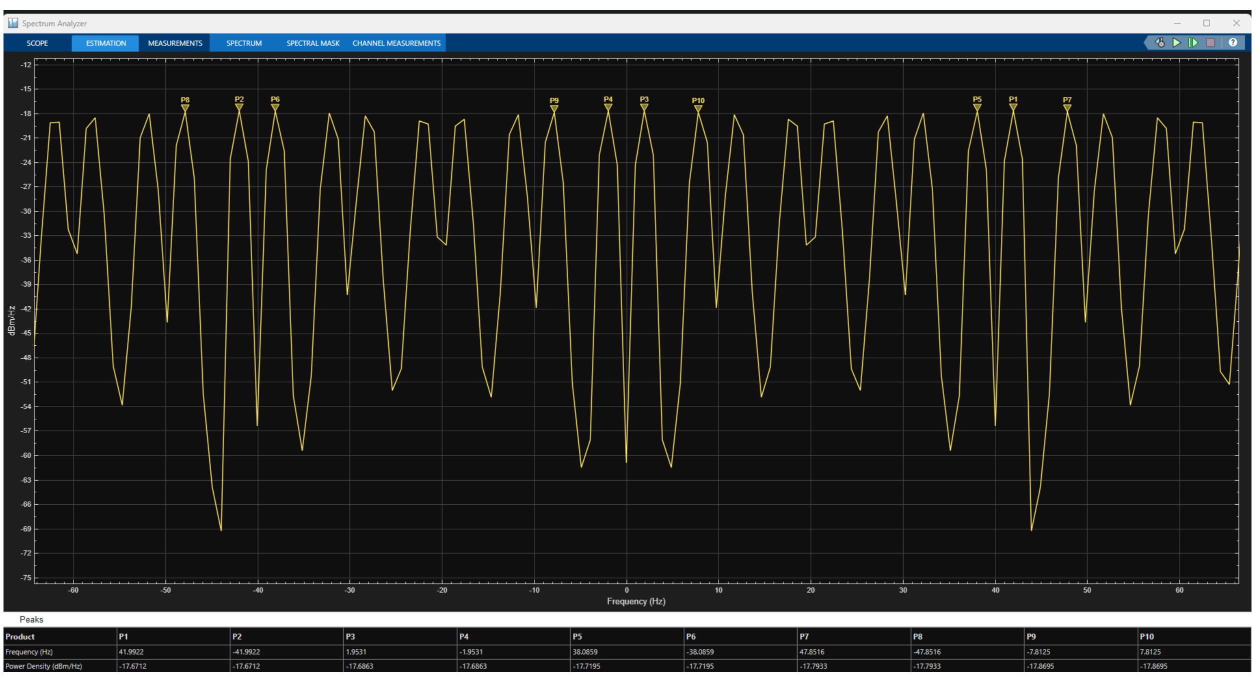


Figure 8, Undersampled Sample Signal Amplitude Spectrum

3. Quantization

3.1 Theory

3.1.1 Quantization Algorithm Design

Quantization is the technique where the values of a sampled signal are mapped to discrete levels based on a step size. Signals in the time-domain transform into a discrete step function where each sample is assigned to a specific point.

When considering a quantization algorithm, we must choose an appropriate number of levels we want there to be in the quantization. The more levels there are, the more precise the quantized signal will be compared to the original signal. A higher precision means that more information will be stored within these levels, so it is important to pick an appropriate number according to how much information our system will be able to handle and process. For example, if a device's limitations allow up to 8 bits per sample for storage, we can set our algorithm to have 2^8 (256) levels. With 4 bits, 2^4 (16) levels. If we use 14 bits for quantization, we will have 2^{14} (16384) levels. We will see in our simulated data that quantization algorithms that quantize to lower and lower bits will limit the precision of the quantized signal compared to the original. The step size $\mathbf{\Delta}$ is the distance between each level, where the range of the dataset of the sampled signal is divided by the bounds within this range.

Once our algorithm specifications are designed, the range is down-shifted so that the initial value is 0. We normalize the range to the step by dividing each value by the step $\mathbf{\Delta}$ and round to the nearest integer. Once this is rounded, we shift our range back to where it was before (up-shift), and we see that each value is on a level according to our quantization specifications.

3.2 Simulated Data

Using MATLAB, we ran two different quantization algorithms, one homemade one where bounds are separated by $\mathbf{\Delta}$, and one using the MATLAB function **quantiz**, where the range is evenly broken into N number of regions. We observe these algorithms through two different mediums in image quantization and audio quantization.

3.2.1 Image Quantization

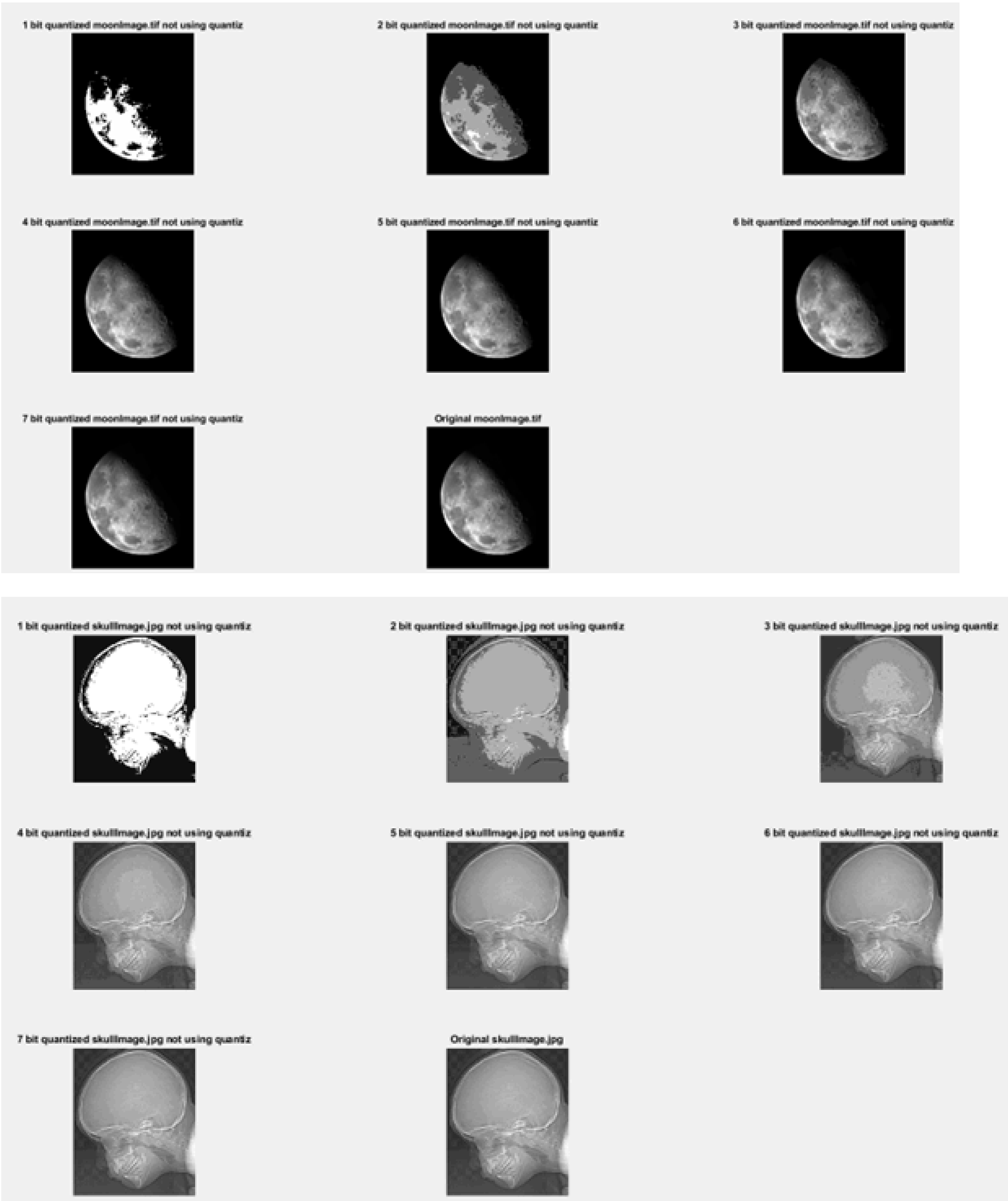


Figure 9, Image Quantization of Varying Bit Sizes, Homemade Algorithm

Let's examine the 1-bit quantized image. In both the moon and skull images, we see that due to the fact that 1-bit quantization contains two available levels, each pixel will either take on black or white (in the discrete sense, 1 or 0). With this in mind, as we increase the size of bit quantization, we can see that different color options become available. In image quantization, the storage of quantized data appears as colors within each pixel. The 7-bit quantized image looks very similar to the original image, an 8-bit quantized image, due to the fact that the step size in 7-bit quantization is similar to the step size of 8-bit quantization. Each value rounded to one of the bounds of the levels in 7-bit quantization will be quite similar to the values rounded to a bound of the levels in 8-bit quantization.

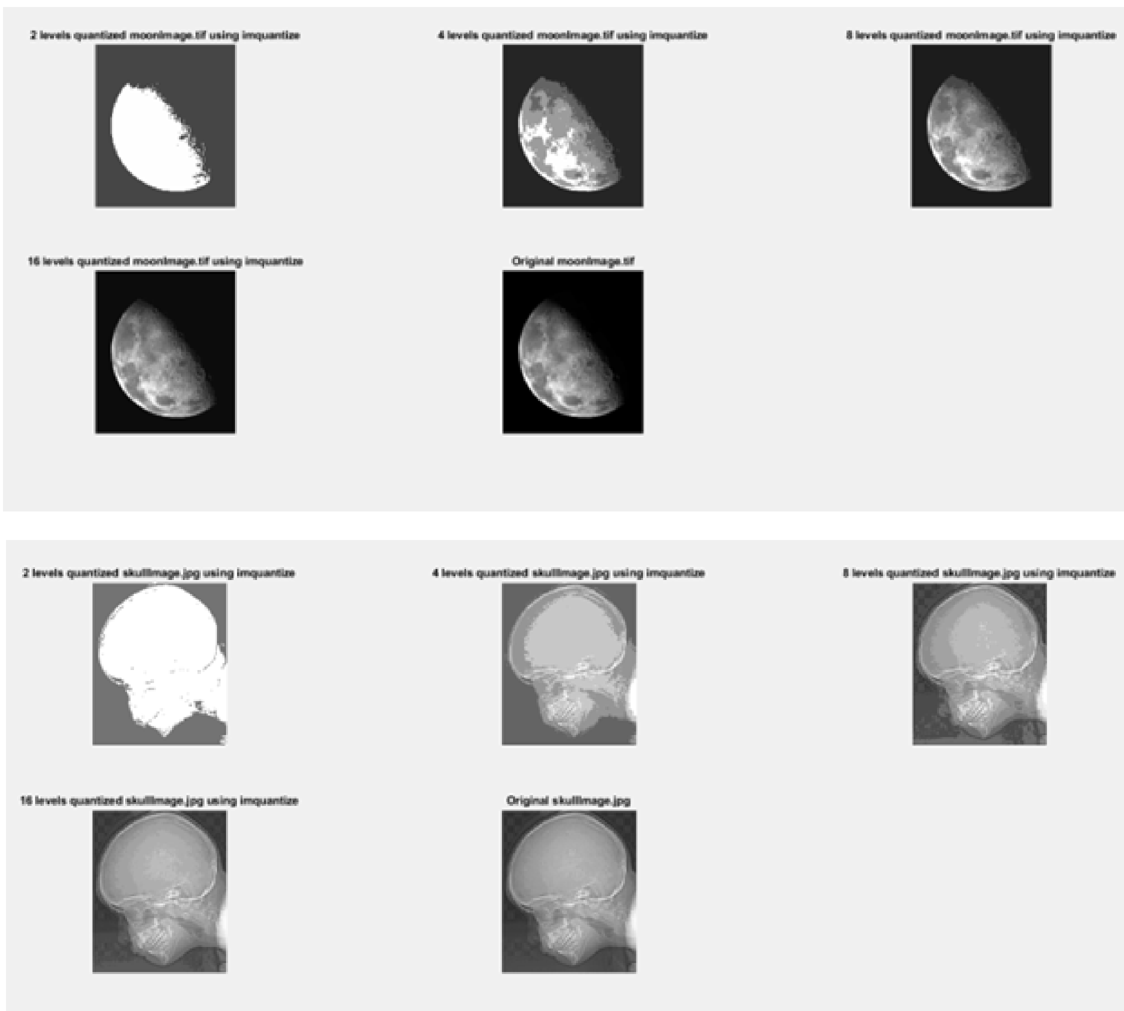


Figure 10, Image Quantization of Varying Bit Sizes, "quantiz" Function

An interesting thing to note when comparing the 1-bit quantizations between the homemade algorithm and the algorithm using the **quantiz** function is that the homemade algorithm actually offers more precision in the quantization. In the 1-bit quantized moon images, the moon craters are clearly visible shaded with black in the homemade algorithm, while these craters are not visible in the quantization algorithm using **quantiz**. However, in the 2-bit, 3-bit, and 4-bit quantizations, it is clear that more information is retained in the quantization process using the **quantiz** function. This is especially apparent in audio quantization.

3.2.2 Audio Quantization

In audio quantization, the values of each sample are rounded to discrete values, like image quantizing. When played, the lower bit quantized audio will be more distorted and homogenized.

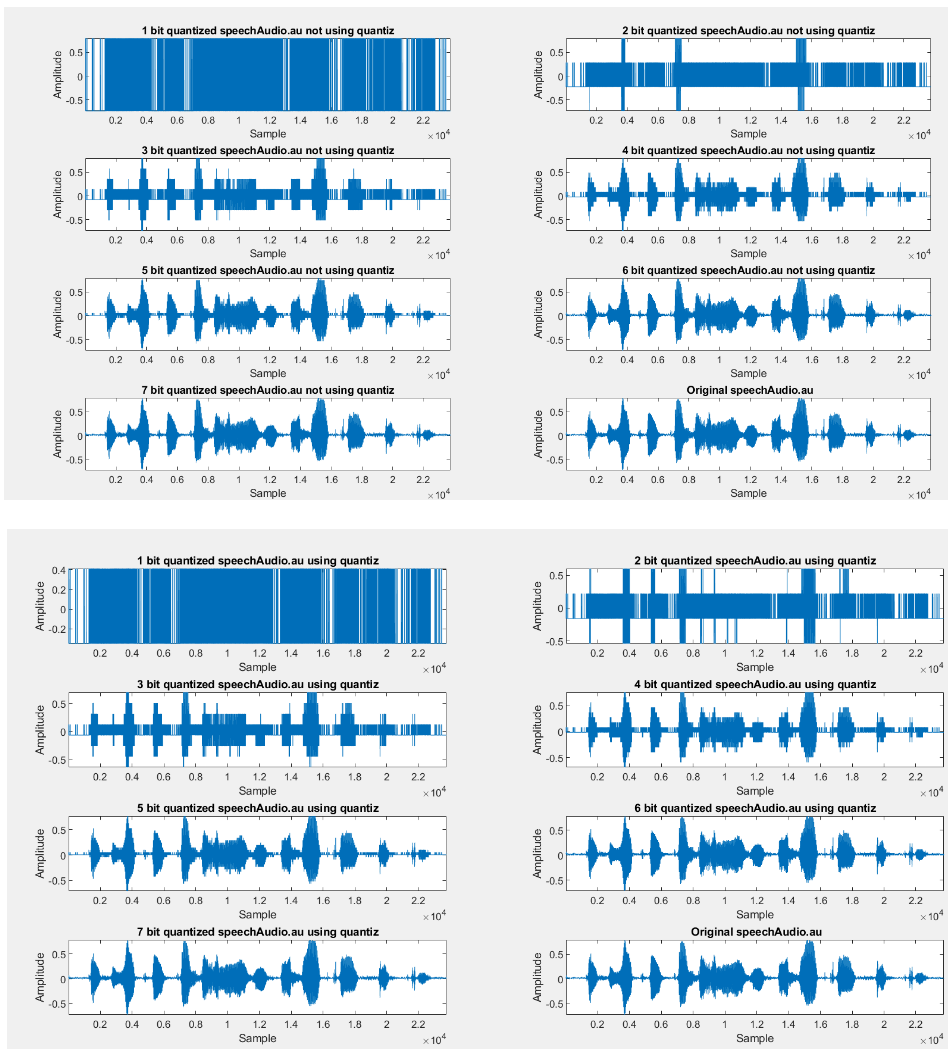


Figure 11, Audio Quantization of 'speechAudio.au', Homemade vs *quantiz* Algorithms

Between the homemade algorithm and the **quantiz** function, we can see that the 1-bit quantized signal appears the same in both algorithms; there are clearly two levels, representing 0 and 1. There is a lot of distortion due to the fact that much of the audio signal has been quantized to the normalized '1' value. Working backwards from the original audio file, we see that as we lower our bit-depth quantization, the audio file appears more and more "blocky" by sample number. The **quantiz** function does a better job at keeping the integrity of the original signal at higher bit-depth quantizations, the most apparent at 2-bit quantization. The same can be observed when we run these algorithms for different audio files.

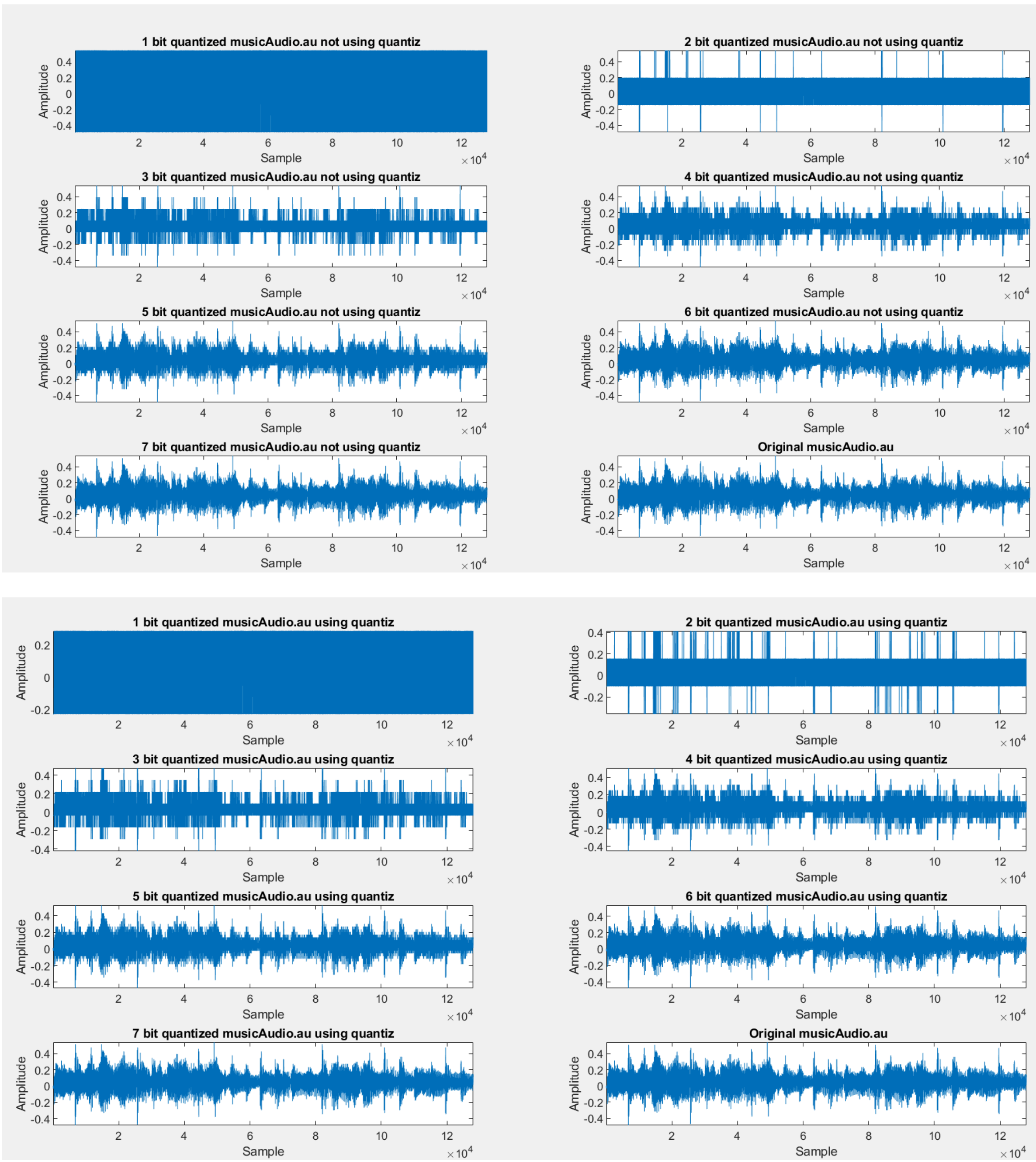


Figure 12, Audio Quantization of 'musicAudio.au', Homemade vs *quantiz* Algorithms

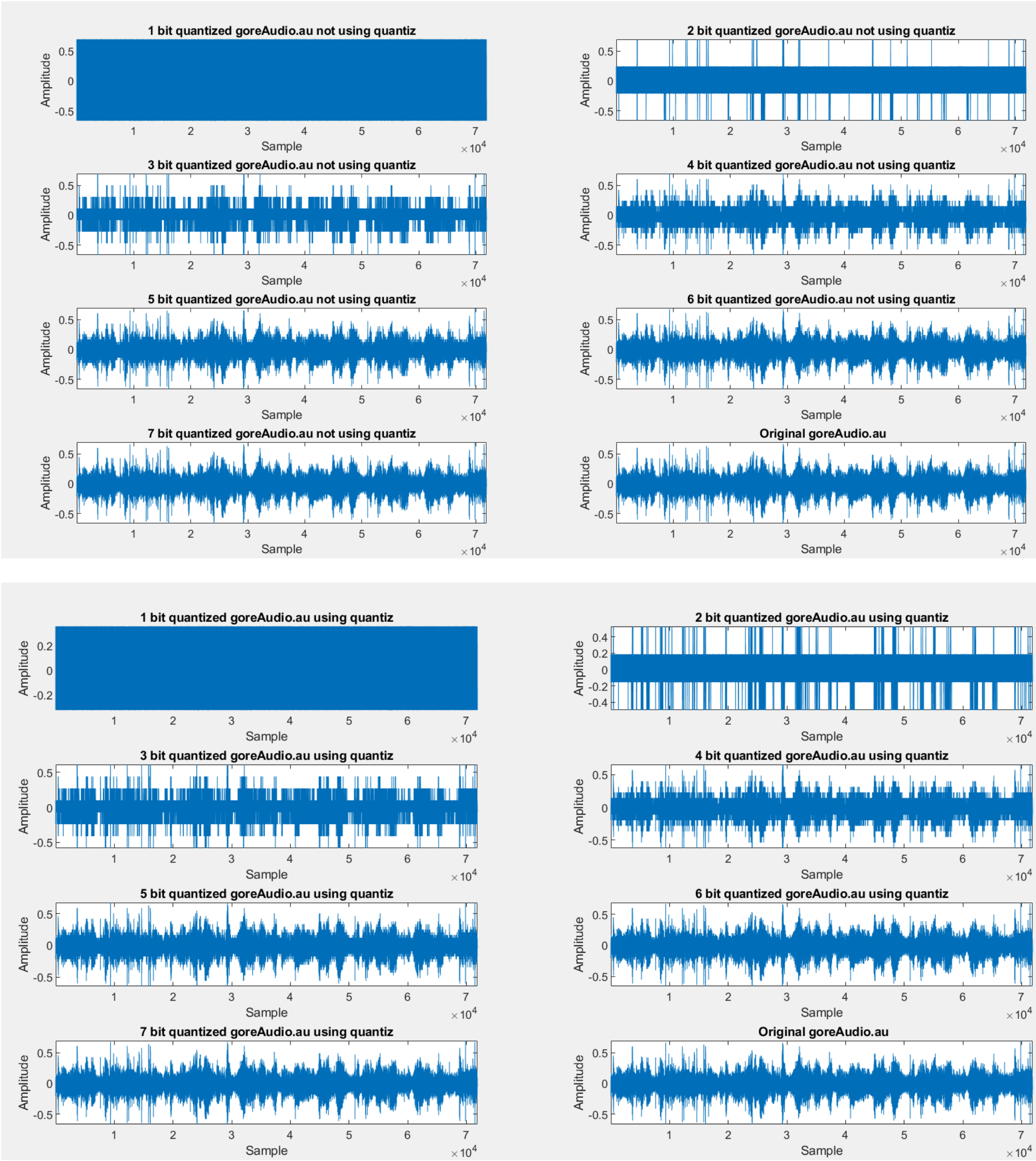


Figure 13, Audio Quantization of ‘goreAudio.au’, Homemade vs *quantiz* Algorithms

4. Error Analysis & Signal-to-Noise Ratio (SNR)

4.1 Theory

4.1.1 Error

The approximative nature of quantization dictates that quantized values have some amount of error–that is, the difference between the actual value of the original signal and the quantized value of the quantized signal. The minimum value of the error is half of the step, $\Delta/2$. By using a histogram, we can plot all error values from a quantized signal and analyze correlation or randomness of the quantization error. As counter-intuitive as it is, a more random error is more desirable in quantization algorithms, as highly correlated error in quantized datasets implies a bias in the data and thus a systematic error. A signal's "randomness" is considered white noise. When we quantize the signal, this white noise may or may not affect the integrity of the quantization. We must look for a high factor of randomness in our error as this may account for any possible noise in the original signal, which means that the data is unbiased and therefore reliable as a quantization algorithm.

4.1.2 Correlation

There are two different correlation schemes we will be utilizing to examine the accuracy of a quantized signal. Self-correlated data is when a dataset is lined up and compared to a shifted version of itself, where the distance of the shift is known as lag. This measurement takes the values of one sample and compares it to another value of another sample. When we self-correlate the quantization error, we often observe a high correlation value. This is indicative of a systematic error and can make it hard to separate the quantized signal from the unwanted noise. Cross-correlated data measures the quantized signal against the original signal at different sample points. High cross-correlation between the two signals indicates a common pattern.

4.1.3 Signal-to-Noise Ratio

The Power Signal-to-Noise ratio (SNR) of a quantized signal can be useful in the quantization analysis as a way to judge the quality of the signal. We can characterize this ratio with the equation

$$PSNR = \frac{P_y}{P_E},$$

where P_y is the power of the quantized signal and P_E is the power of the noise in the signal. Bluntly put, a good SNR for our purposes must be large, with the signal being a large value and the noise being a small value. Recall that the power of a signal can be defined by

$$P_X = \frac{1}{L} \sum_{n=1}^{L} x^2(n),$$

where L is the length of signal *x(n)*. With this, we can plot a rate-distortion curve. These curves compare the distortion with the bit rate(s) of the quantized signal. Bit-rate is a measure of bits per second where the bit-depth of quantization is multiplied with the sampling rate. For example, suppose we have a signal sampled at 8kHz. The bit-rate of the quantized signal for 16 quantization bits will be 128kbps, as the sampling frequency is multiplied by the number of bits.

4.2 Simulated Data

4.2.1 Quantization Error

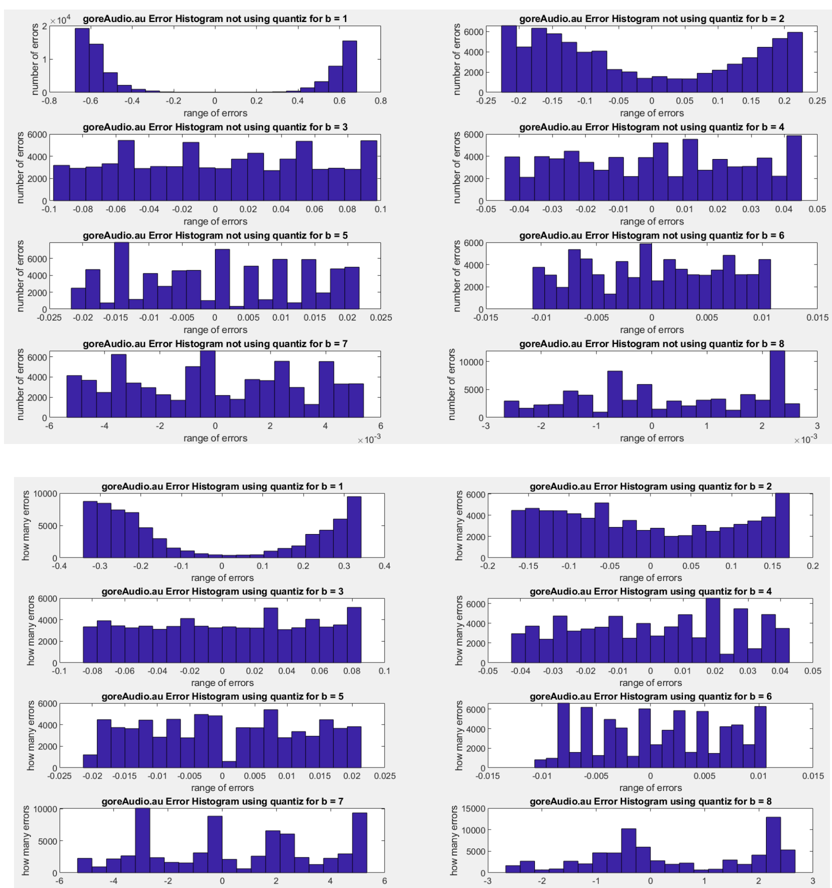
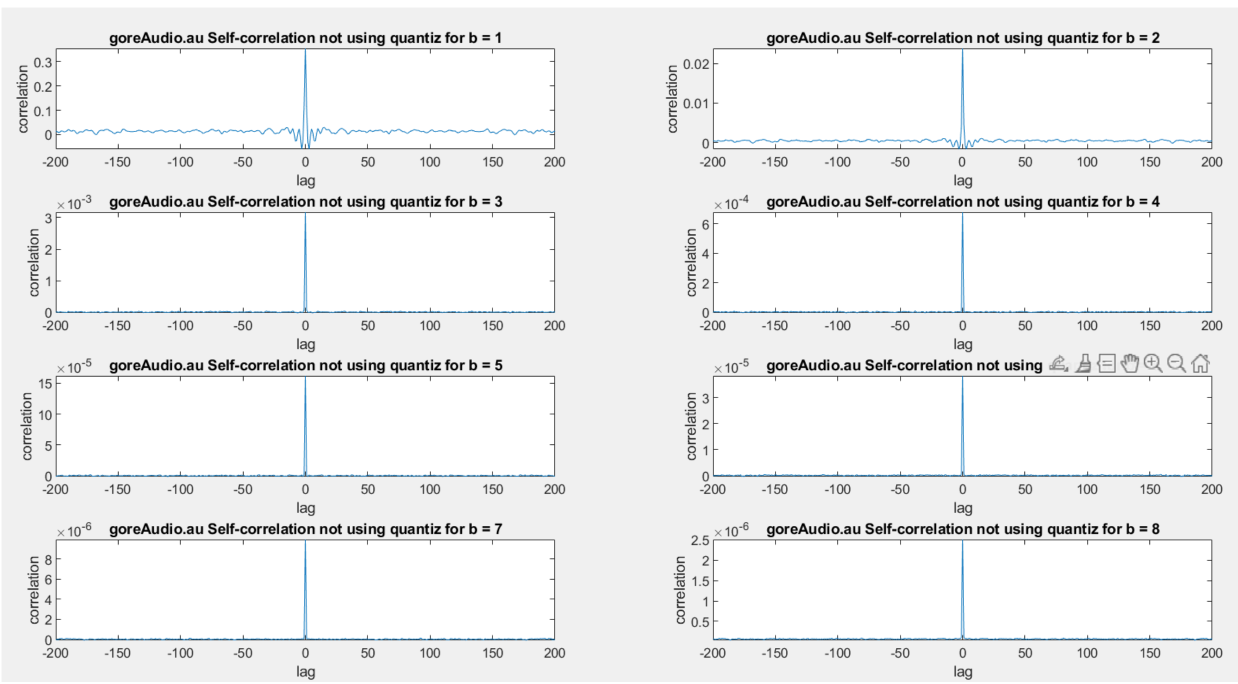


Figure 14, Quantization Error Histogram of ‘goreAudio.au’ for Both Algorithms

This histogram plots show that randomness increases with the bit-depth. In the 1-bit quantized dataset, we see that the histogram behaves under a pattern, with the number of errors rising on each side of the range. This suggests the inaccuracy of the 1-bit quantized dataset compared to the original signal and thus indicates a systematic error. This supports the concept that lower and lower bit-depth quantization will do more “damage” to the original signal. In the 7-bit quantization, there appears to be a higher factor of randomness in the error histogram in the algorithm using the **quantiz** function, further supporting the higher accuracy in quantization compared to the homemade algorithm at higher bit-depths.

4.2.2 Correlation



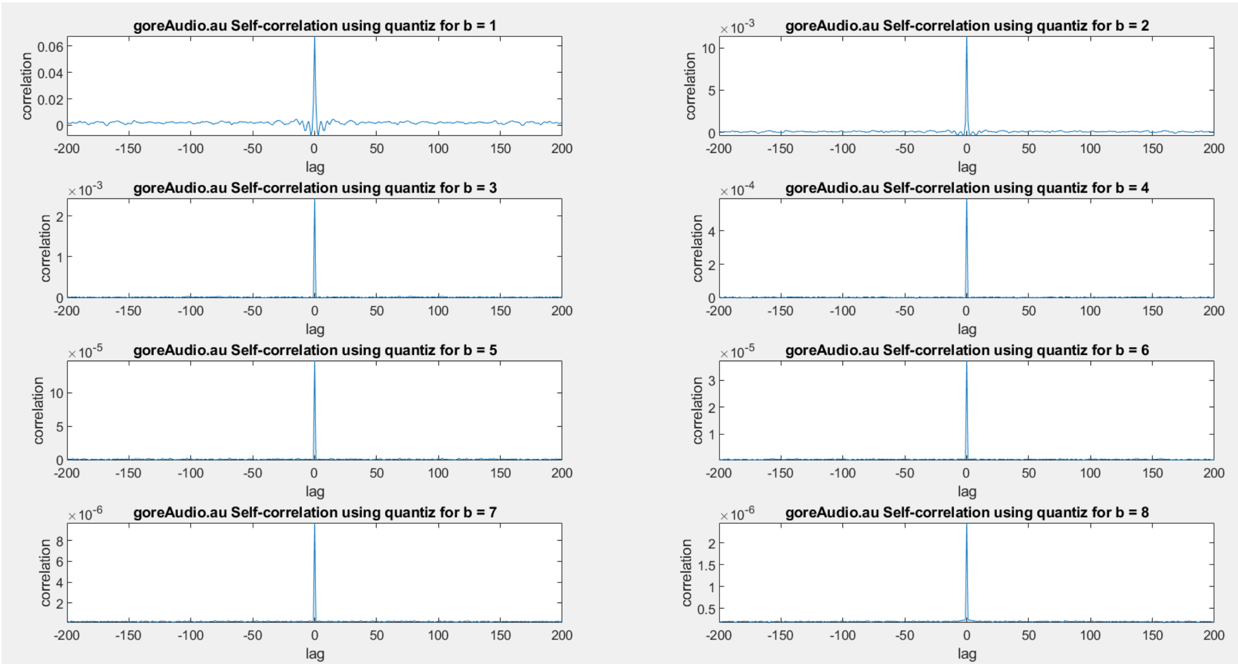


Figure 15, Self-correlation Graphs of Varying Bit-Depths for 'goreAudio.au' for Both Algorithms

The self-correlation graphs show that the quantized datasets all have high correlation when the lag is 0. This is behaving as expected as the error dataset is compared to itself. As lag approaches infinity or negative infinity, we can see the correlation between the error dataset and the lagged data set increase in randomness. The similarities between the error and the time-shifted error decrease as bit-depth quantization increases. Take a look at the y-axis scale for correlation at 7-bit quantization compared to 1-bit quantization. The correlation is *extremely* low in cases of a nonzero lag, scaled to 10^{-6}.

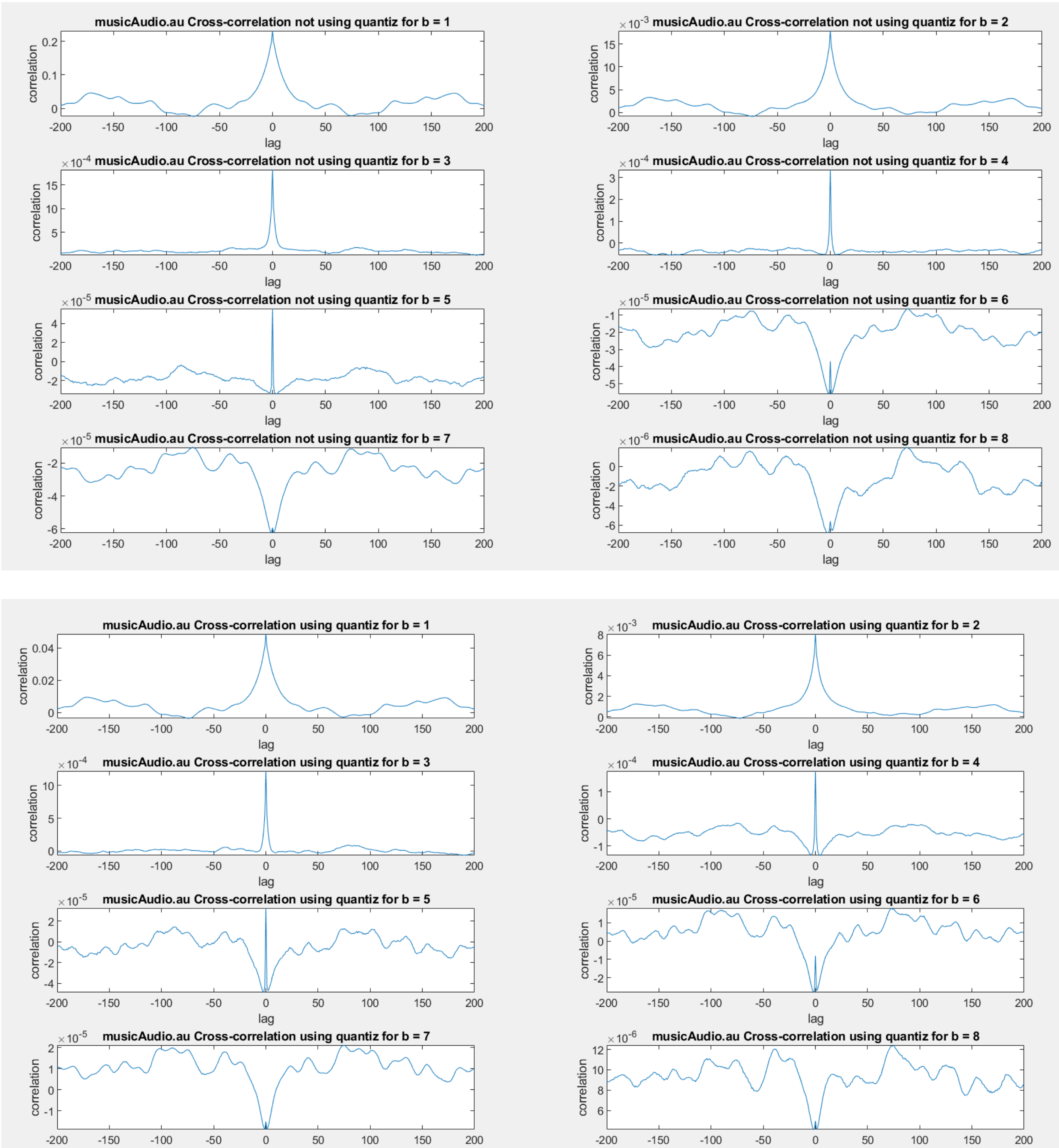


Figure 16, Cross-correlation Graphs of Varying Bit-Depths for 'musicAudio.au' for Both Algorithms

In cross-correlation, where the quantized dataset is compared to the original signal's, we can observe the correlation associated with a quantized dataset's to a shifted original dataset. At 0 lag for the quantization of audio file 'musicAudio.au,' we see that there is a low correlation, suggesting a lack of bias in the quantization and thus higher accuracy of quantization.

4.2.3 Distortion Curves

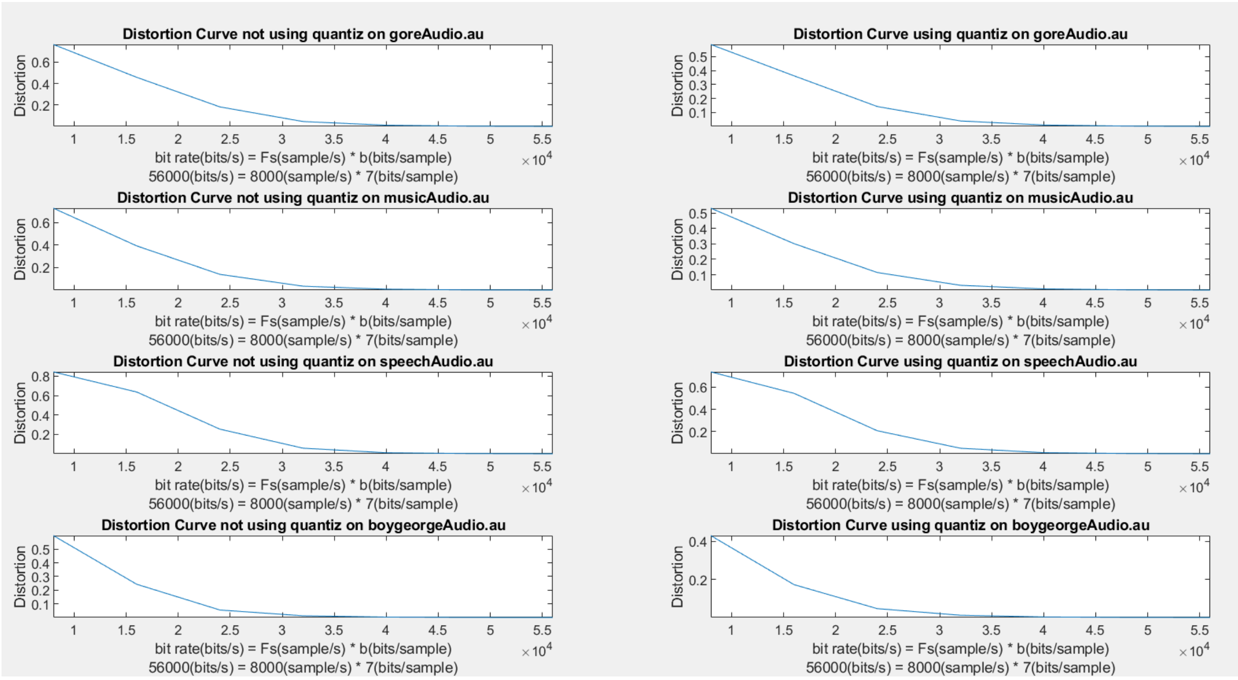


Figure 17, Rate-Distortion Curves for 4 Audio Files

As mentioned earlier, we can compare the rate of distortion in the quantization process to the bit-rate. These graphs show the rate-distortion curve, where as our bit rate increases, there is a decreasing factor in distortion (5-bit quantization produces a 40kbps bit rate, 6-bit produces 48kbps, 7-bit produces 56kbps). Higher bit rates have an exceedingly low distortion rate.

5. Conclusion

Over the course of 3 activities, we have furthered our exploration into the realm of signal processing, studying how to digitize analog signals. We can represent continuous signals in a discrete manner by taking measurements at a set interval. The Nyquist-Shannon sampling theorem ensures that we may capture an accurate sample and avoid distortion and aliasing of the signal. Quantization takes a sampled signal and maps each value to a discrete level, rounding out the value and approximating it to the original signal. The more levels there are, the more precise the quantization system is. This is supported from the increased signal-to-noise ratio in an algorithm. One must consider the storage constraints while deciding an appropriate bit-depth quantization in order to maximize the signal quality. The quantization of images and audio alike can be perceived as a technique of compression, where the set number of levels to a quantization may limit the media's quality of complexion. This difference is known as the quantization error, which can be used to observe the distortion of any quantization algorithm. A low randomness in error of a quantization algorithm suggests that there is a systematic bias present in the quantization and may not be desirable in representing information. Sampling and quantization is a fundamental set of techniques in the signal processing industry. In the age of information, storage efficiency is important, and it is desirable to find the most optimal way to communicate and transfer data while keeping the integrity and quality of the information.