Mitchell Larson
EE 3221
12/20/17
Lab4

1) **Expression used to find the sequence vector for x2**

```
f_B = 880 % 880 Hz tone
x2 = sin(2*pi*f_B*Ts*n);
```

2) **Describe the differences between the two sounds. What causes the difference?**

The sound created by sending the 880 Hz sinusoid to the soundcard is a higher pitch than the 440 Hz sinusoid. The reason that this occurs is due to the frequency of the signal. Higher frequency sounds will have a higher pitch to the human ear, while lower frequency sounds will have a lower pitch.

3) **Explain what [x1 x2 x3] and mean([x1 x2 x3]) do.**

[x1 x2 x3] creates 2 seconds (when default sampling rate of 8192 Hz is used) worth of sound data for each sinusoidal and appends them to each other. This results in a total of 6 seconds of sound that changes tone/frequency every 2 seconds.

Mean([x1 x2 x3]) does something completely different. Rather than appending the sampled data for each sinusoid, the data at every sample iteration is averaged together. This results in sampled data for the average of all 3 sinusoids, which will be 2 seconds long when using the default sampling rate of 8192 Hz

4) **Describe ways in which the spectrum of xs and xc are similar and different**

One way the xs and xc are similar is that at any moment in time, the frequency spectrum would only show a magnitude at a single frequency. This is due to the signal being a sinusoid. The magnitude of the data for a frequency spectrum would also likely be the same or very close. While these characteristics demonstrate some similarities between the frequency spectrums of xc and xs, there are still some important differences. A frequency spectrum for xc would only show a single frequency for the duration of the audio. This is different from xs, which would see 3 different frequencies at different times for the duration of the audio. In addition, the frequency picked up in a spectrum of xc would be unique from the frequencies picked up in a spectrum of xs.

5) **How many possible amplitude levels are possible when 8 bits are used to quantize the signal?**

2^8 = 256 different levels

Mitchell Larson
EE 3221
12/20/17
Lab4

6) Describe what you hear. What might be causing this?

The sound of the actual music didn't sound that much different from the original music, but there was a noticeable static and popping sounds. Additionally, when the music attempts to make small steps in pitch, the steps don't seem to occur. This would be caused by wider steps with less definition. With less distinct levels, some data that should be on separate levels is forced to be on the same level as data it is different from. The static could possibly be due to audio being rounded up to extremes close to 1 or -1 that the audio card may not be good at translating into sound. Another possible explanation is that frequencies that should have no audio are picking up some audio, thus diminishing the quality of the sound (white noise).

Mitchell Larson
EE 3221
12/20/17
Lab4

7) Frequency spectrum comparison of bit sampling

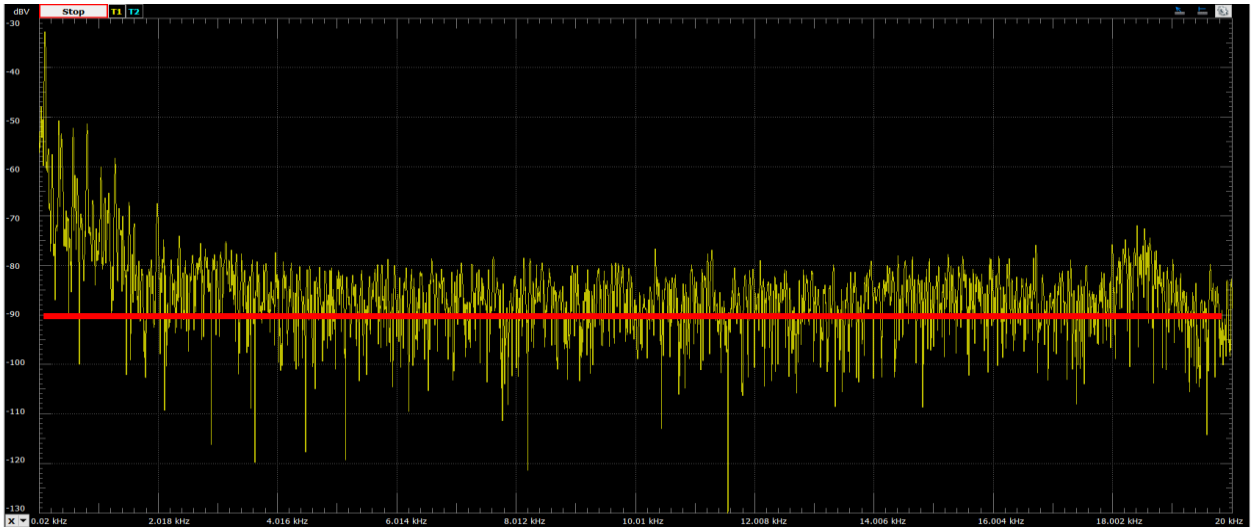

Figure 1 - Frequency spectrum of original sound - 16 bits

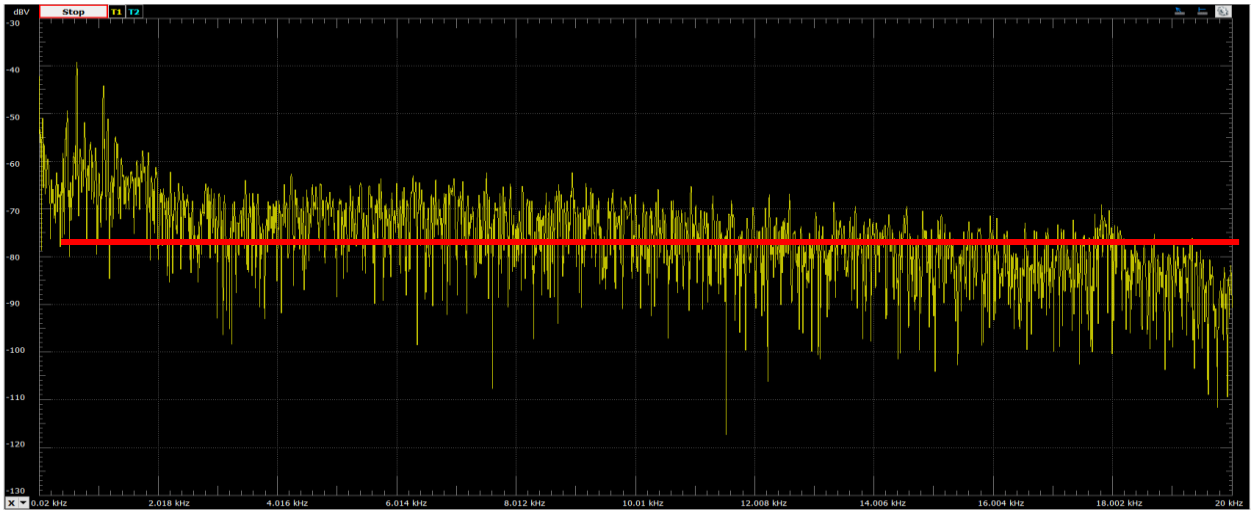

Figure 2 - Frequency spectrum of original sound - 4 bits

Comparing the frequency spectrum for the original sound and the 4-bit sound reveals that there is significantly more signal strength at higher frequencies when less bits are used. The higher frequency noise averages to -90 dB in the original sound, but -80 dB in the 4-bit sound. This extra signal strength should not exist, and would be considered white noise.

Mitchell Larson
EE 3221
12/20/17
Lab4

8) Describe, qualitatively, what happens when fewer bits are used to represent the samples

When fewer bits are used to represent the samples, the audio quality noticeably depreciates. When 16 bits were used to represent the audio samples, the audio sounded very good at moderate volume levels; there was little cracking and static noise. However, if fewer bits are used, the cracking and static becomes much more abundant. This phenomenon is known as "white noise", and it is caused by the quantization error that can be observed in the spectrum plots above (average noise moves from -90 dB to -80 dB). With fewer bits used to represent audio data, the "unwanted" signal at higher frequencies becomes much stronger, and there is a very noticeable audible impact on the quality of the sound.

9) Plot of x, xq, and e

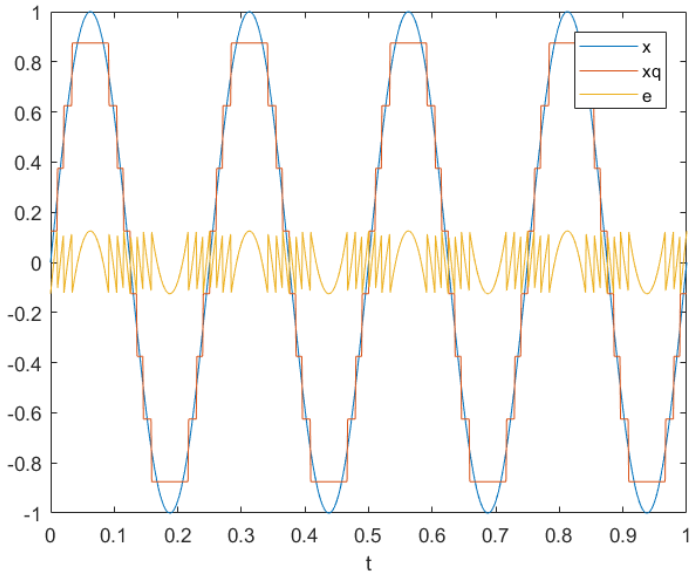

Figure 3 - Plot of x, xq, and e

Using the minimum and maximum functions in matlab, the range of the quantization error, e, was determined to be -0.125 to 0.125. It turns out that the size of this range is the size of the quantization steps shown in the plot above. This makes sense since the quantization error is maximized when the line for x and xq cross. At that point, the quantization level is +/- 0.125 away from the actual value. Data is quantized to the level that is the closest to it. This means the data can be quantized up or down.

Mitchell Larson
EE 3221
12/20/17
Lab4

10) Simulated SQNR values

Number of bits	Simulated SQNR (dB)	Theoretical SQNR(dB)
2	**11.8**	**13.8**
4	**25.1**	**25.8**
8	**49.9**	**49.9**
12	**73.5**	**74.0**
16	**98.4**	**98.1**

As the number of bits is increased, the quantization error, e, quickly decreases because of the decibel scale. Once the number of bits was increased to 8, the plot of x, xq, and e showed e essentially being a flat line on the x-axis. In addition to this, it seems that as the number of bits is increased, the simulated and theoretical SQNR values close in on each other. With only 2 bits, there is a significant difference between the simulated SQNR value and the theoretical SQNR value, but at 16 bits, the difference is negligible. Essentially, to ensure quality data you should make the power of the signal much greater than the power of the quantization error by increasing the number of bits to represent the data.

Mitchell Larson
EE 3221
12/20/17
Lab4

11) 3-bit truncating quantizer

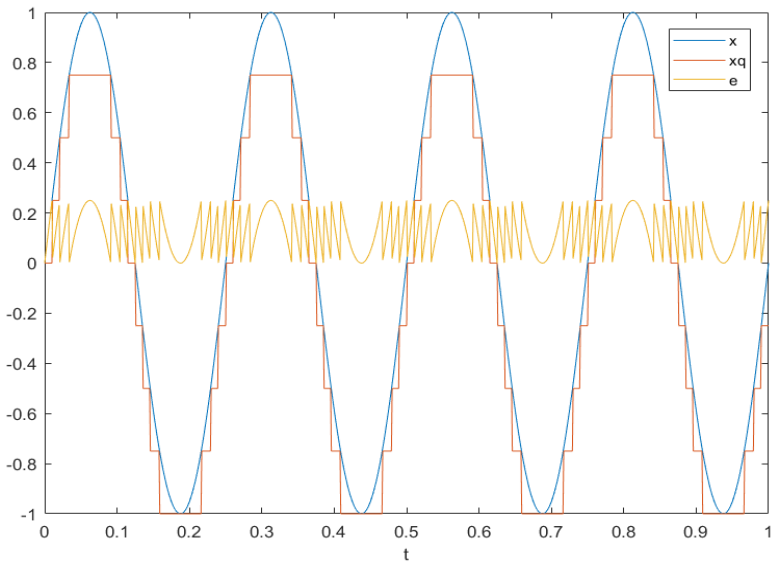

Figure 4 - 3 bit truncating quantizer with error shown

Using a 3-bit truncating quantizer, the quantization error, e, ranges from 0 to 0.25. The size of this range is the size of the steps shown in the plot above. Unlike the rounding truncator, the quantization error is always positive. This is because rather than data being quantized to the level closest to the true value, they are always quantized to the next lowest level. As a result, the difference between the actual value and the quantized value will always be positive (with the maximum error being the size of the step in a sinusoidal signal).

12) SQNR using 8-bit truncating quantizer

When an 8-bit truncating quantizer is used, the SQNR becomes 43.9, which is significantly lower than the SQNR value of 49.9 when using a rounding quantizer. The most likely reason that this occurs is due to the truncation essentially offsetting the quantization level by -0.125 in comparison to the rounding quantizer. While the size of the range of e remains the same, a comparison of the plots shows that there is a significant difference. Using a rounding quantizer produces a quantized signal that is closer to the true shape of the original signal. Therefore, the SQNR value is much higher when using the rounding truncator. When reducing the quantizer definition to 2 bits and then making another comparison, the difference becomes even more clear. A rounding quantizer does a much better job at approximating the signal.