Big Idea (Week 2): Discrete Sample and Quantization

How do we process continuous information on a digital computer that represents and processes discrete data?

A "sound" is a continuous acoustic (local atmospheric pressure) waveform. Digital audio engineering must deal with acquiring and storing in the computer's finite and discrete memory representation of those continuous waveforms amenable to processing (minimally: internal copying and reconstruction via external speakers).

Contemporary digital audio technology uses "pulse code modulation" (PCM), the process of converting acoustic sound pressure waves into analog voltage signals using analog electro-mechanical components, and then sampling those voltage signals in time and value using analog-to-digital (A2D) electronics. The question now arises as to what level of fidelity to the original sound can be expected from this electromechanically transformed and double-digitally (both in time and value) sampled signal. That is, how much noise does a particular sampling rate or quantization interval introduces compared to the original signal? More formally, we introduce the notion of a quantization function,

$$\mathrm{Quantize}_L(x) = \mathrm{Round}(L \cdot x)/L, \tag{1}$$

where Round is the function that rounds real numbers to the nearest integer.

Using this formalism, we can define the PCM function as a composition of quantization steps as follows. A sampler quantizes time to convert a continuous signal $s(t)$ to one varying discretely between constant intervals of duration T, $s_T(t)$ defined as

$$s_T(t) = \mathrm{Sample}_T[s(t)] = s[\mathrm{Quantize}_{1/T}(t)] = s[T \cdot \mathrm{Round}(t/T)]. \tag{2}$$

The PCM function follows the time sampled signal with a quantization in value to some level L, yielding $s_{(L,T)}(t)$, defined as

$$s_{(L,T)}(t) = PCM_{(L,T)}[s(t)] = \mathrm{Quantize}_L\left(\mathrm{Sample}_T[s(t)]\right) = \mathrm{Quantize}_L\left(s[\mathrm{Quantize}_{1/T}(t)]\right). \tag{3}$$

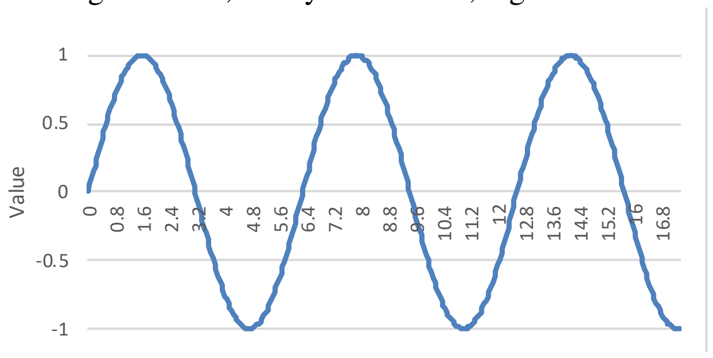
The *noise* introduced in sampling and quantization can then be formally defined as:

$$n(t) = s(t) - s_{(L,T)}(t) \tag{4}$$

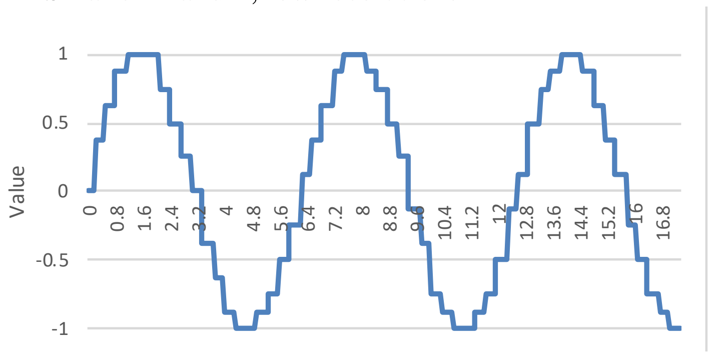
Note that we define *noise* formally and quantitatively here as the difference between our intended (ideal) signal and the signal we actual received (represent, reconstruct). This may not exactly match your intuitive notion of noise, but we will see that it is often not far off.

Since there are an uncountably infinite number of $s(t)$'s, but only a countable number of $s_{L,T}(t)$'s, this double digital sampling process always discards information, introducing non-zero noise. However, if we have a *model* for the structure that exists within $s(t)$, or, at least, the structure that we care about in $s(t)$, and this structure is appropriately restricted, we can represent $s(t)$ perfectly—or, at least, so that the noise, $n(t)$, is irrelevant to our application. Considering what humans can actually hear, it turns out that a finite representation is "Good Enough" such that we can define reasonably values of L and T for audio signals of interest for human hearing.

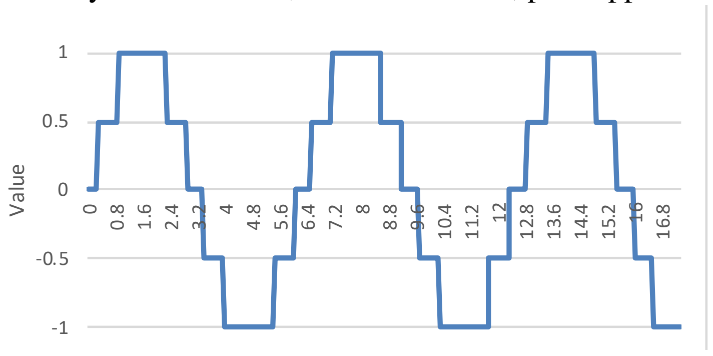
Large L and T, nearly continuous, high resolution:



Smaller L and T, low resolution:



Very small L and T, lower resolution, poor approximation:


Value