Digital signal sampling and quantization

Terminologies

Steppy signal: is signal that increases its amplitude in steps.

Noise Floor: is the quietest region of your audio tracks where all you hear is the hiss and hum of electrical noise and whatever ambient sound you have going on in the room.

Headroom: Digital signal processing can result in "clipping", or audio samples that exceed the allowed range. These samples are "clipped" to a maximum or minimum value. Which produces distortion. This is easily avoided by slightly attenuating the signal--to make "headroom" for sample values that may increase as a result of signal processing. Headroom is the amount by which the signal-handling capabilities of an audio system can exceed a designated nominal level.

Dynamic Range: is the ratio between the largest and smallest values that a certain quantity can assume. It is often used in the context of signals, like sound and light. It is mostly measured as a logarithmic base-10 (decibel).

- For analog signal it is the signal to noise-floor ration, the ration between the power of sine wave at unit amplitude and power of the noise floor level.
- For digital signal it is the SNR, the ration between the power of sine wave at unit amplitude and power of the quantization level.

Signals

Signal is defined as a dependent variable that changes subject to independent variable, in many cases and mainly in audio the independent variable is time, and so we will consider that the independent variable is time, which is usually denoted by t, and the dependent variable could be any physical measurement that changes over time - in audio the dependent variable usually is the amplitude of the signal - denoted by x, or x(t) to make the time dependence explicit.

Analog signals are continuous range of value in both the time and amplitude, which leads to an infinite number of values. Since computers can only store a finite number of values, one has to convert the continuous analog waveform into its discrete representation. Digital signals are discrete values in both the time and amplitude. The process of converting analog signal to its digital counterpart is called Digitization, and it involves several steps: Sampling, Quantization, Coding.

Sampling (aka PAM):

The first step in the digitization process, is to obtain measurements of the amplitude of the continuous time signal at regular intervals and record them as numbers. This process in known as Sampling, which is performed by Analog to Digital Converter (DAC for short). The interval by which we take the measurements is known as the Sampling Period denoted by T_s, and its

reciprocal, the sampling frequency or the sampling rate denoted by F_s where $F_s = 1/T_s$ and it is measure in hertz (number of samples per seconds). The result of this process is sequence of numbers. We usually use n to index into this sequence, and so the discrete time signal is denoted by $x[n]$.

Note, In DSP, it is customary to use parenthesis for continuous time signals and brackets for discrete time signals.

With all of that in mind, Sampling basically extracts the values from continuous time signal at regular intervals, which are integers multiples of T_s, such that we can write the relationship between the discrete time and continuous time signal as:

$$x[n] = x(n \cdot T_s) \quad (1)$$

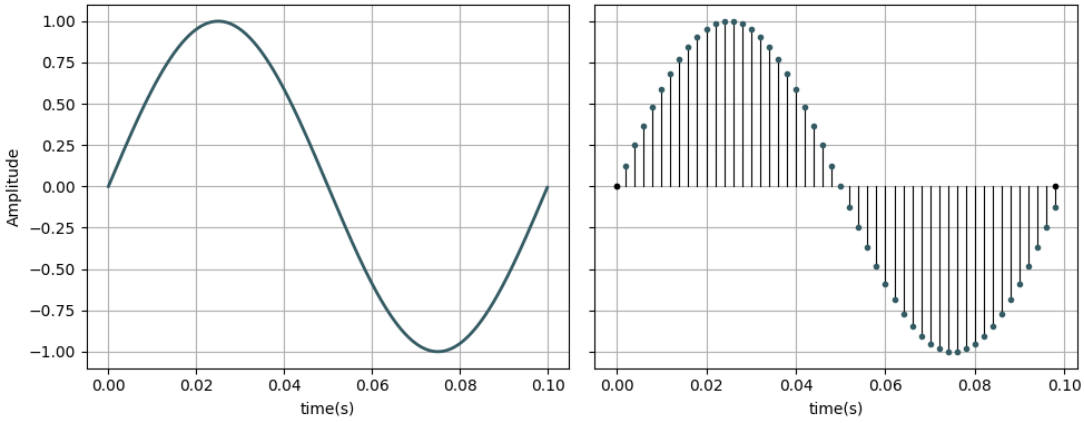

Figure 1 Sampling continuous time signal to discrete signal

Note, the resulting sequence of numbers from the sampling process, can take on any value from continuous range, thus we use the term discrete-time signal instead of digital signal.

The resulting values are called samples. Note that the term sample refers to a single number that represent the signal amplitude at a specific point in time, this is different from what musicians mean when talking about samples, they usually refer to short recording of audio.

Note: the two most common sampling rate for audio are 44.1 Khz and 48 Khz.

The sampling frequency determines how detailed our representation of the analog signal will be captured in the digital domain. Since high frequencies create more details than low frequencies, we need high frequency sampling rate to capture those details.

Some people like think of the sampling process as being similar to taking picture frames at regular intervals to make a digital video, this analogy isn't quite accurate, but it is somehow helpful simplification.

The process of sampling is sometimes called PAM which stands for Pulse Amplitude Modulation, the reason is that when you sample a signal the original continuous time signal modulates the pulses amplitude - printing the amplitude of the continuous time signal in the pulses -, resulting pulses that mirror the amplitude of the original signal at that specific time.

The Sampling Theorem:

The purpose of digitization is to capture the original signal as close as possible - process it maybe -, and reconstruct it back. One extremely important rule for doing this process right comes from Claude Shannon work in 1949, the sampling theorem states the conditions under which a continuous time signal can be reconstructed back exactly from its samples, and also defines the interpolation algorithm which should be used to achieve this exact interpolation - which I will talk about next.

In loose terms, the sampling theorem states that the original continuous time signal can be reconstructed back exactly from its samples, only, and only if the highest frequency present in the original single (denoted by F_h) is less then half of the sampling frequency:

$$F_s < \frac{F_s}{2} \tag{2}$$

when this condition is met, we can reconstruct the original signal back.

Note: Half of the sampling frequency is known as the Nyquist frequency, in honor of Harry Nyquist who also contributed to sampling theory.

Note: In some books, instead of using strictly greater than they use greater or equal for the sampling theorem rule above, which is WRONG, it has to be strictly greater than. And that's because when you sample a sine wave that has frequency exactly equal to the Nyquist frequency (half the sampling rate), you may be able to reconstruct faithfully the original sine wave back only and only if the samples-instants happen to coincide with the maxima of the sine-wave, but when the sample-instants happen to coincide with zero crossing of the sine wave, you will capture nothing.

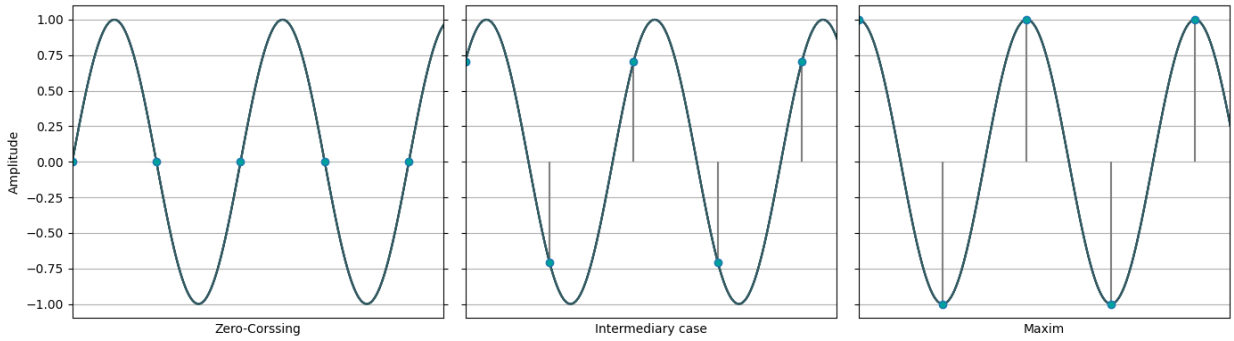

Figure 2 Sampling a signal at exactly the Nyquist limit, could happen perfectly if the signal maximum and minimum points happen to coincide with the samples-instants (right), other than that we get inaccurate sampling (middle), or worst capture nothing if samples-instants coincide with zero-crossing points (left).

For intermediary cases, you will capture sinusoids with wrong amplitude. Thus, perfect reconstruction of the original signal is not guaranteed for $F_s < F_s/2$. In fact, most of the high frequencies won't be captured accurately, but will be reconstructed correctly (or interpolated correctly, using sophisticated interpolated methods).

If the sampling theorem ignored and signal was sampled with sampling frequency less than two times the highest frequency present in the original signal, we get what is known as Aliasing, which is artifact of under-sampling, a type of distortion that can only digitally.

Aliasing

The first thing that ADC does before sampling the analog signal, is to filter out all the frequencies above the Nyquist frequency. If the high frequencies are not filtered out, any frequency higher than the Nyquist frequency will be flipped or reflected into (or over) the original signal as distortion (aka artifact). The filter that filters out the frequencies above the Nyquist frequency is known as Anti-Aliasing Filter. Nearly all ADC nowadays, implement high quality Anti-Aliasing-filters, so that aliasing isn't be problem that you need to care about. With a good ADC aliasing will only happen either intentionally as special effect, or accidentally with a poorly designed DSP software.

To answer why aliasing occurs in digital systems, is far simpler to do it mathematically, when sampling a signal (detailed discussion of aliasing is in its own article)

Quantization

As I mentioned before, the resulting sequence of numbers from the sampling process, can take on any value from continuous range. Thus, their representation would require an infinite number of digits (bits) to represent them in the digital domain. Because we have to store these values digitally in finite storage space, we need to represent each number with a finite number of digits (bits). That's where quantization take place. Quantization is the process of assigning the arbitrary values (of the discrete-time signal) to the nearest defined amplitude levels (which are a limited number of possibilities along the amplitude axis, defined by the Bit-Depth). The number

of amplitude possibilities (levels) that the quantization process defines, is denoted nu N_q , and the distance between each consecutive levels is denoted by q.

Note: Clipping of high and low values take place at the quantization process - assuming that the digital system also treats ±1 as the highest possible values. Any values larger than +1 or less then -1, is mapped to +1 and -1 respectively.

Note: Quantization is sometimes referred to as descritizing the amplitude values.

Just like sampling process defines the resolution along the time (x) axis with the sampling rate, the quantization process defines the resolution along the amplitude (y) axis with the Bit-Depth. Bit-Depth is the number of bits that will be used to represent each value in the amplitude (y) axis. common values for audio are 16 bits (16bit/44.1Khz is the standard for digital CDs), and 24 bit for modern DAW, 32, 64 also are use, and even 128 bits.

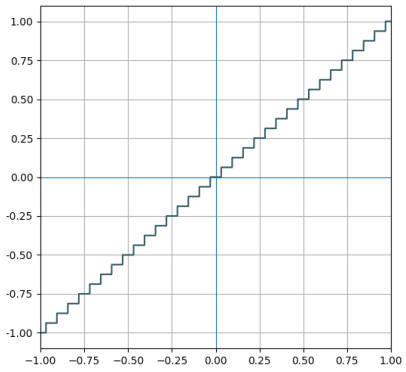

Figure 3 Characteristic of line a quantizer, input from a continue range x mapped to discrete levels.

Quantization noise:

when forcing an arbitrary signal value x to its closest quantization level x_q, the latter can be seen as x plus some error. We will denote that error as e_q (for quantization error) and so we have:

$$x_q = x + e_q \Leftrightarrow e_q = x - x_q \qquad (3)$$

The quantization error (Fig. 2) is restricted to the range, $-q/2 \ldots + q/2$, That means the error will never be greater than half of the quantization step size - except for extremely high or low values.

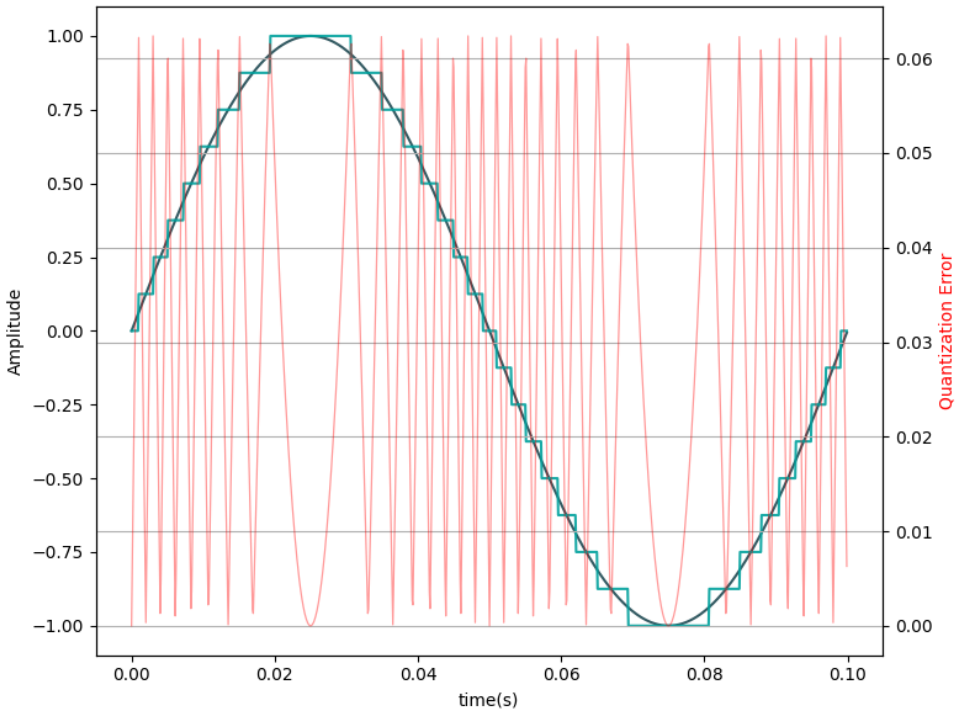

Figure 4 Quantization error

Now, the quantization error (treated as a discrete time signal $e_q[n]$) will manifest itself as additive white noise (that is why it's called quantization noise) with equal probability for all error values in the range $-q/2 \ldots + q/2$. this assumption is usually proceeded with the assumption that the discrete-time signal to be quantized has no relationship to the quantization process (or generally the original signal has no relationship to the sampling process), which means that the quantization error is inevitable. Mathematically we can view the error signal as a random signal with a uniform probability density function between $-q/2 \ldots + q/2$. That is:

$$P(e) = \begin{Bmatrix} \frac{1}{q} & for \ -\frac{q}{2} \leq e < \frac{q}{2} \\ 0 & otherwise \end{Bmatrix} \tag{4}$$

Wirdrow's quantization theorem, provides more rigorous Mathematical justification for why the quantization error should be treated as uniform distribution of white noise. But that discussion is a bit too far for the purpose of this article, I will leave that to you if you want to pursue more information on that topic.

We define the SNR (Signal to Noise Ration) of system as the ratio between the RMS (root Means Square) of the desired signal and the noise, expressed in Decibels:

$$SNR = 20\, Log_{10}\left(\frac{x_{rms}}{e_{rms}}\right) \tag{5}$$

Where, the RMS value is the square root of the (average) power of the signal, - below I described the power and the RMS of a signal in both the digital and analog domain. Denoting the power of the signal as P_x, and the power of the error as P_e, we can write the SNR as:

$$SNR = 10\,Log_{10}\left(\frac{P_s}{P_e}\right) \tag{6}$$

The power of a quantization error signal is given by the variance of the associated continuous variable, which comes out as:

$$p_e = \int_{-\frac{q}{2}}^{\frac{q}{2}} \frac{1}{q} e^2 \cdot de = \frac{1}{3q} e^3 \Big|_{-\frac{q}{2}}^{\frac{q}{2}} = \frac{q^2}{12} \tag{7}$$

q : quantization step-size.
N_q : Number of quantization levels.
e_q : quantization error.

Taking a sine wave at unit amplitude, as the desired signal, which has the power of:

$$P_x = \frac{1}{2} \tag{8}$$

putting this, into the expression of the SNR described power, Equation (6)

$$SNR = 10\,Log_{10}\left(\frac{\frac{1}{2}}{\frac{q^2}{12}}\right) = 10\,Log_{10}\left(\frac{6}{q^2}\right) \tag{9}$$

The above equation can be used to calculate the SNR between a sine wave with unite amplitude, and the quantization error, when quantizing with a step size of q.

Now, before plugging definitive values in the previous equation. I should explain how we calculate q. Because computers represent numbers using bits, we use b to represent the number of bits, we can represent 2^b distinct values. Because the amplitude oscillates between positive and negative values, we need to represent both. So, using b bits, we can represent 2^{b-1} positive values, and 2^{b-1} negative values (half of 2^b bit pattern will be used for positive values, the other half for the negative values), But we have not represented the zero value in this setup yet, to do this we "steal" one bit pattern from the positive range, now with that we can only represent $2^{b-1} - 1$ positive values. With all of that in mind, the step size is:

$$q = \frac{1}{2^{b-1}} \tag{10}$$

Now, we have all everything we need to calculate the SNR for a signal with b bits as:

$$SNR = 10\log_{10}\left(\frac{6}{\left(\frac{1}{2^{b-1}}\right)^2}\right) = 10\log_{10}\left(6 \cdot 2^{(2b-2)}\right) \tag{11}$$

as a rule of thumb:

$$SNR = 6.b \tag{12}$$

This says, each bit increases the signal-to-noise ration by 6dB. The exact formula gives an SNR of roughly 98 dB for 16 bits, and SNR of 146 dB for 24 bits (as opposed to 96, and 144 respectively for the thumb-rule).

Note: The SNR calculation we have performed so far was for a sine wave with a unit amplitude, which has a power of $\frac{1}{2}$ *and RMS of* $\frac{1}{\sqrt{2}} = 0.707$ *..., and this is quite a bit of optimistic assumption.*

Practical signals usually have far less power and RMS than sine wave with unite amplitude. Thus, we would get a lower SNR for most practical signals.

In practical scenarios, when recording a signal, we have to make a tradeoff between the headroom and SNR. Large Bit-Depth offers a larger dynamic range, which reduce the problem of the tradeoff between the headroom and SNR, but at the expense of more storage space.

Interpolation:

Whereas the continuous time analog signal x(t) is defined at all points in time, the discrete time digital signal is only defined for times which are multiples of T_s. To reconstruct a continuous signal from samples, we must somehow 'guess' what the intermediary values of the signal in between samples. Interpolation is the process of 'guessing' the signal values at arbitrary instant of time. Thereby the interpolation process creates a continuous time single, and can be seen as the inverse process of sampling.

Note: Interpolation is only necessary for certain type of Processing, like oversampling. The interpolation (reconstruction of the smooth continuous time signal from the discrete time version), is done by DAC.

Ideally, we would want our interpolated signal matches as close as possible the original analog signal. The crudest of all interpolation schemes, is piecewise constant interpolation - just take the value of the previous sample extend it till the next sample. The interpolated single will have stair-step like shape. The next simple interpolation scheme is linear interpolation (aka connect the dot interpolation). Whereas, the samples values are connected with straight lines. The figure shows reconstructed continuous signal, linear (middle) interpolation.

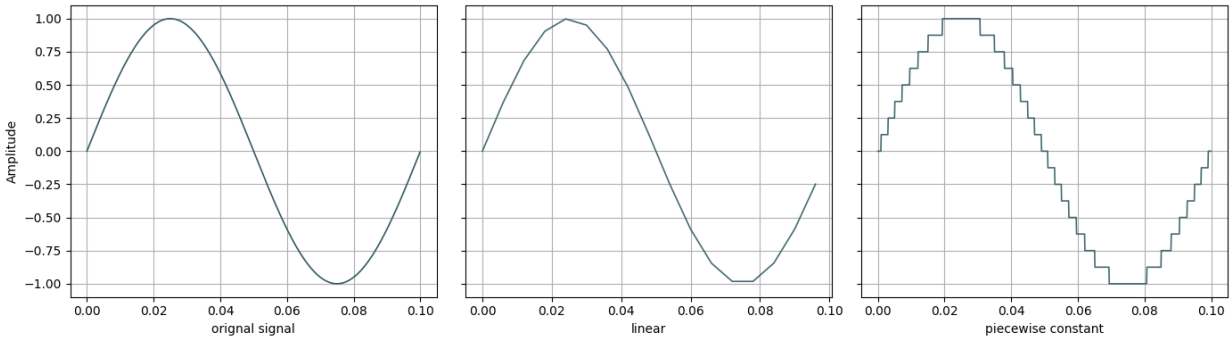

Figure 5 Signal Interpolation

These basic interpolations scheme inevitably produce some error (Figure 6), More sophisticated interpolation methods (such as higher order polynomial interpolation, quadratic, cubic, quartic, pentic, etc.) yield a smoother and more faithful reconstruction.

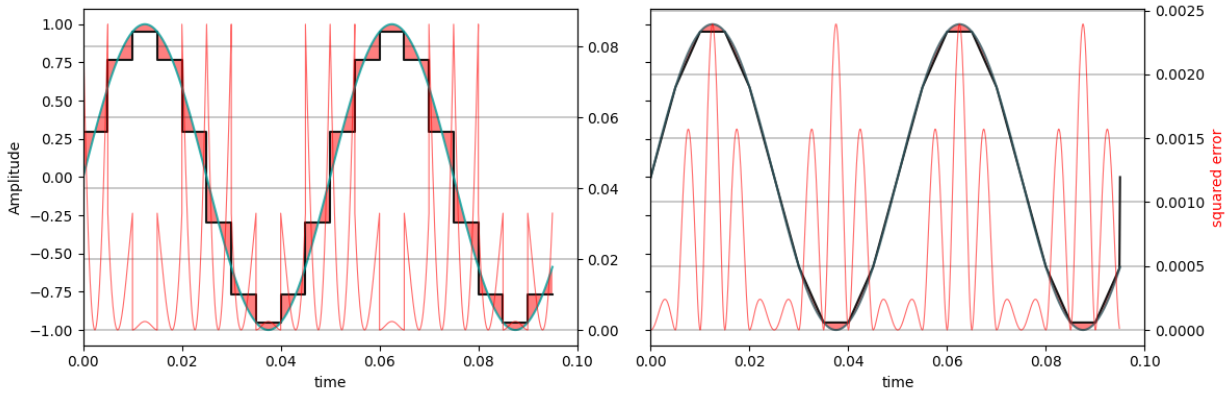

Figure 6 Interpolation Error, piece-wise interpolation (left), linear interpolation (right)

More Details About the Natural Interpolation:

To actually hear digital audio, we must run through a digital to analog converter (DAC) the analog output of the DAC can be then send to a power amplifier, then to a speaker or headphone.

All DACs no matter their esoteric differences, they all do the same thing, they go to each sample and produce an analog voltage that matches that sample's measurement at the appropriate time based on the sample rate, those voltages go through a reconstruction filter so that the final output is a smooth analog waveform.

Contrary to what you might see in your DAW when zooming in on signal waveform, the signal is not made out of blocky stairstep, or jagged connect the dots line. The waveform shown by the software, will be either piecewise constant or linear interpolation, just for a visual shortcut to save the CPU power - Note, some software will perform more sophisticated interpolation methods that will produce a smoother representation of the waveform.

As an analogy to understand why a reconstruct filter can reproduce a smooth waveform that at least closely matches the original signal. We would hold a string to a vertical board using a thumbtack on each end, we will treat the thumbtacks as sample points, most audio software will show these points on the screen either as stairstep of straight line, but the actual sampled audio is more analogous to the curve of the string, which happens naturally due to the laws of physics. Each thumbtack is like voltage the DAC produces per sample, and the string (or more precisely the characteristics of the string and the law of physics), is like the reconstruction filter, which turns the individual voltages into a continuous waveform.

In the case of the thumbtack and the string, the factors that define the natural curve of the string, are the length of the string, the position of the thumbtack and gravity. In the case of digital audio, the factors that define the smooth analog output of the DAC are the sampling rate, the measurement of the samples, and the internal working within the reconstruction filter.

Now, for more practical proof, if you have a oscilloscope that can be plugged to your computer, compare the shape of the waveform of 20 Khz sine wave (at 44.1Khz sampling rate), on the screen of the audio software (that uses either piecewise constant or linear interpolation) and the screen of the oscilloscope. You will see that the shape of the waveform on your screen won't look anything like a sine wave.

Power vs RMS of Digital Signal:

- The RMS: is the root square of the sum of all samples squared, divided by their total number.
- The Power: is the sum of all samples squared, divided by their total number.

Power vs RMS of Continuous Signal:

- Power: is integral of the signal squared divided by the signal duration
- RMS: is root square of the integral of the signal squared divided by the signal duration.