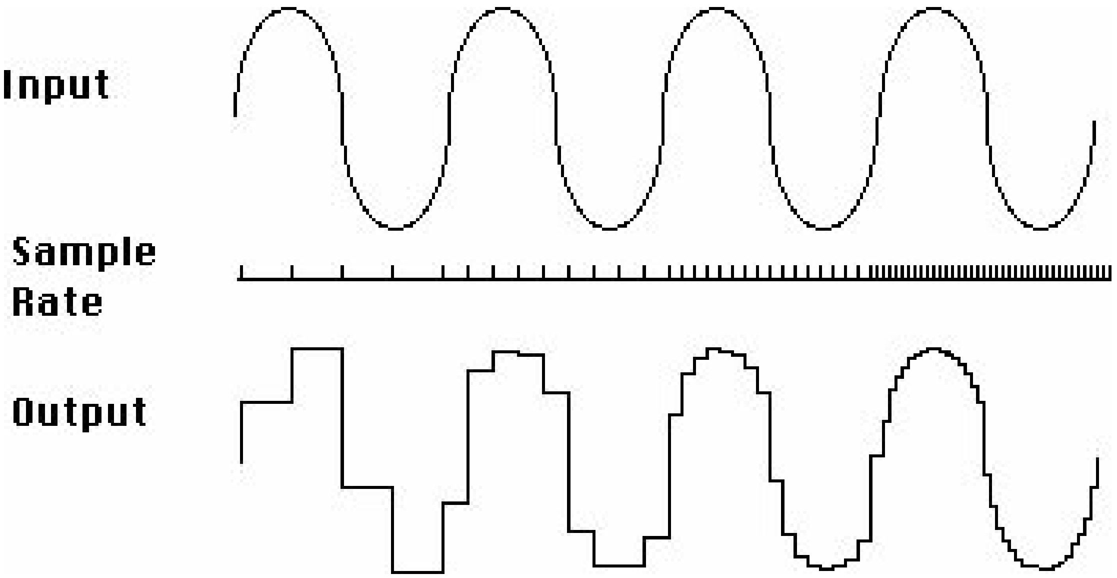
Sampling the Signal

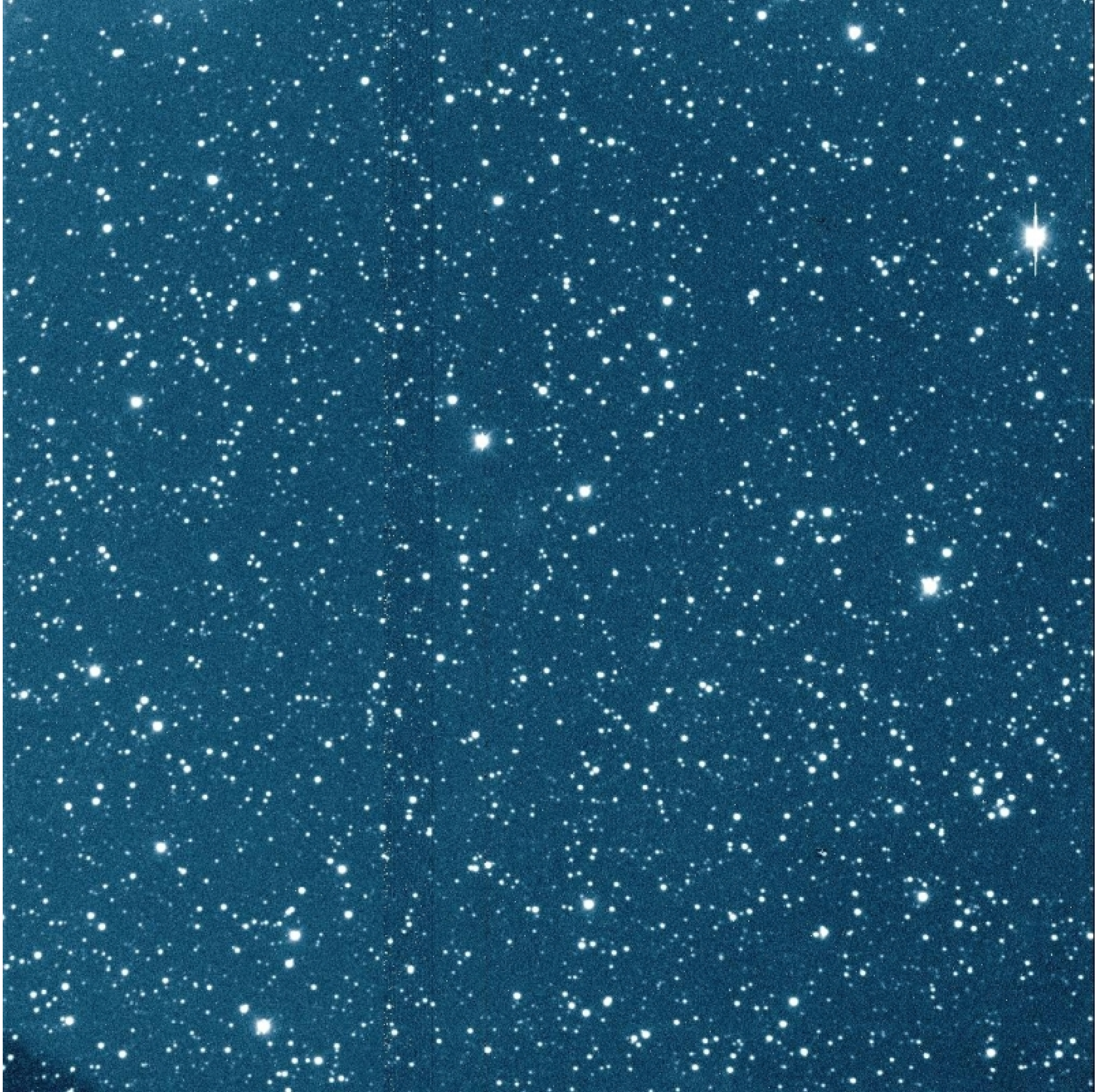


Sampling the Signal

- Generally signals are analog in nature
- Signal has to be:
  - **Digitized** → Sampled and Converted to *digital form*
  - **Stored** → Copies, Redundancy, Compression
  - **Sent** → Description needed: the **header**

An example:

Is **this** data?

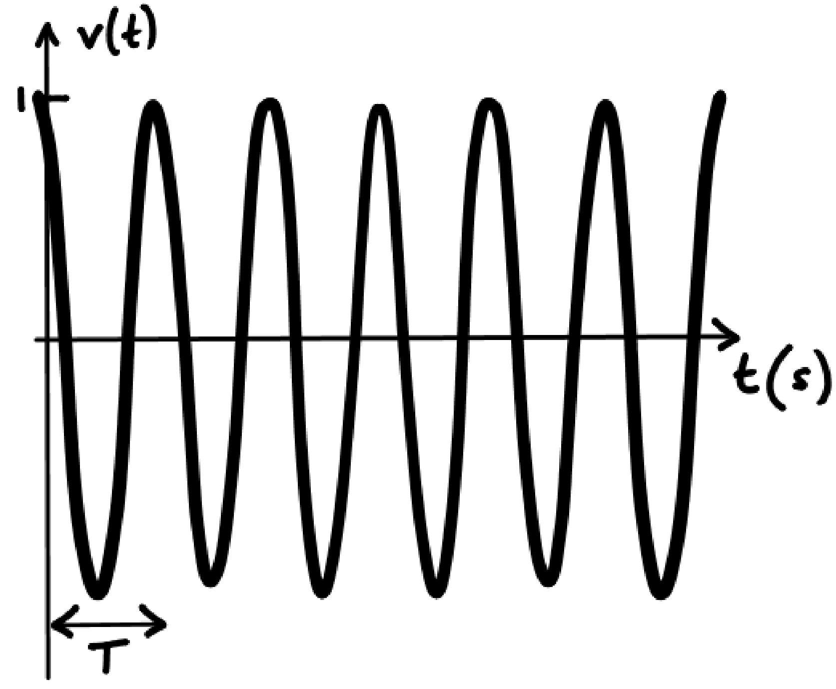


- The photo
- The digital photo
- The values
- The “ordered” values

The ANALOG Signal

- An analog signal exists throughout a continuous interval of time and/or takes on a continuous range of values.
- Example: A sinusoidal signal (also called a pure tone) has these properties.

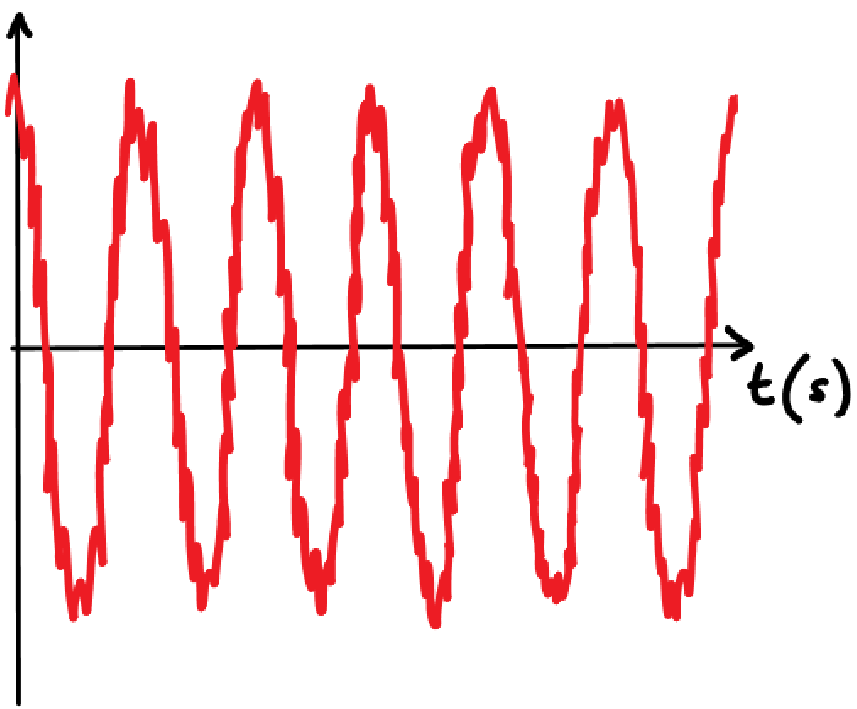
This signal $v(t)=\cos(2\pi ft)$ could be a perfect analog recording of a pure tone of frequency f Hz. If $f=440$ Hz, this tone is the musical note A above middle C, to which orchestras often tune their instruments.
The period $T=1/f$ is the duration of one full oscillation.



The RECORDED Signal

- In reality, electrical recordings suffer from noise that unavoidably degrades the signal. The more a recording is transferred from one analog format to another, the more it loses fidelity to the original.

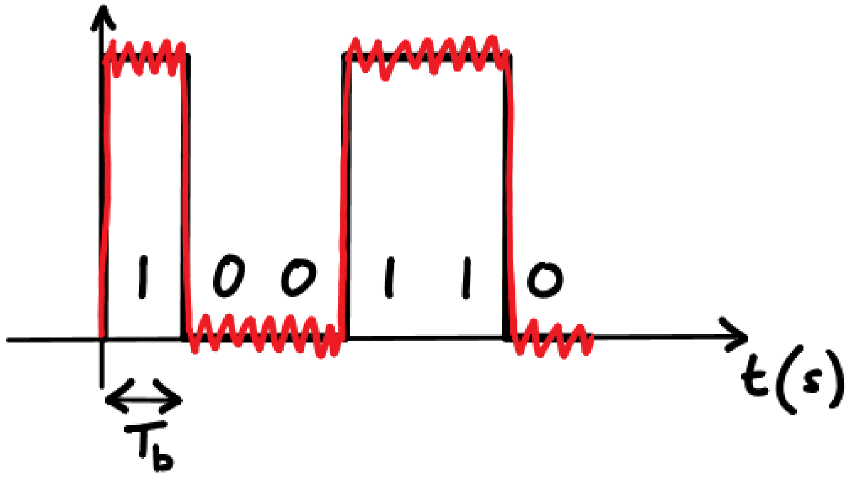
Noise degrades the sinusoidal signal.
It is **often impossible** to recover the original signal exactly from the noisy version.



The DIGITAL Signal

- A digital signal is a sequence of discrete symbols.
- If these symbols are zeros and ones, we call them bits.
- Digital signal is neither continuous in time nor continuous in its range of values and, therefore, cannot perfectly represent arbitrary analog signals.
- On the other hand, digital signals are resilient against noise.

Consider a digital signal 100110 converted to an analog signal for radio transmission. The received signal suffers from noise, but given sufficient bit duration T_b, it is still easy to read off the original sequence 100110 perfectly.



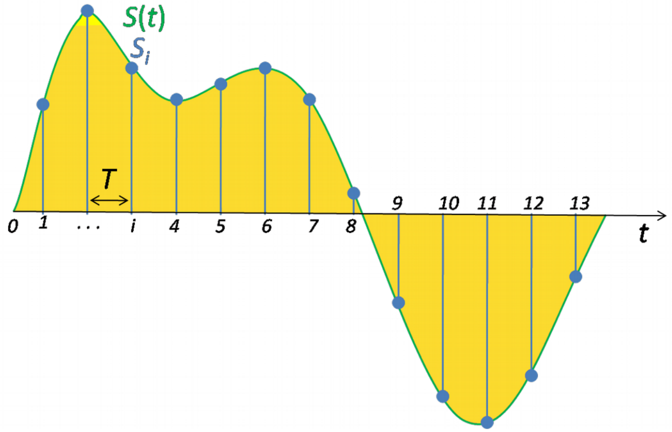
The DIGITAL Signal

- Digital signals can be stored on digital media (E.g. CD) and manipulated on digital systems
- This digital technology enables a variety of digital processing unavailable to analog systems.
  - Music signal encoded on a media includes additional data used for digital error correction. In case the media is scratched and some of the digital signal becomes corrupted, the digital player may still be able to reconstruct the missing bits exactly from the error correction data.
- To protect the integrity of the data despite being stored on a damaged device, it is common to convert **analog signals** to **digital signals** using steps called **sampling** and **quantization**.

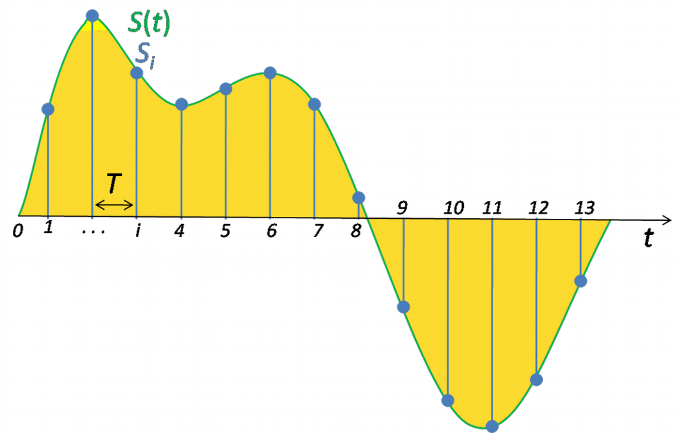
Sampling the Signal

- Generally signal are analog in nature
-
- Signal has to be:
  - **Digitized** → **Sampled** and Quantized to *digital form*
  - **Stored** → Copies, Redundancy, Compression
  - **Sent** → Description needed: the **header**

In **signal** processing, **sampling** is the reduction of a continuous-time signal to a discrete-time signal. A common example is the conversion of a sound wave (a continuous signal) to a **sequence of samples** (a discrete-time signal). A **sample** is a value or set of values at a point in time and/or space.



In **signal** processing, **sampling** is the reduction of a continuous-time signal to a discrete-time signal. A common example is the conversion of a sound wave (a continuous signal) to a **sequence of samples** (a discrete-time signal). A **sample** is a value or set of values at a point in time and/or space.



$$S(t) = \cos(2\pi t)$$

$$S_i = \Sigma S(t)\, \delta(t-iT)$$

$$S_i = \begin{cases} S(t) & t=iT \\ 0 & \text{elsewhere} \end{cases}$$

The sampling frequency is $f_s=1/T$ Hz.

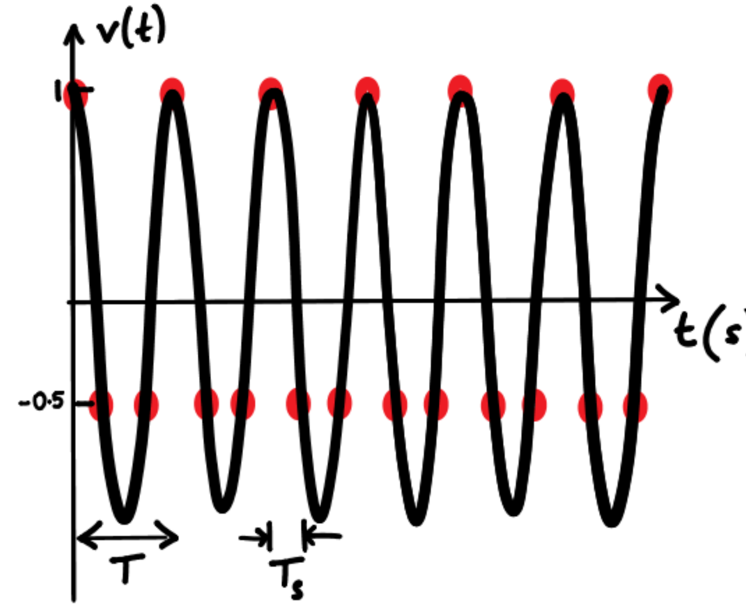
The sampling frequency could also be stated in terms of radians, denoted by ω_s

.

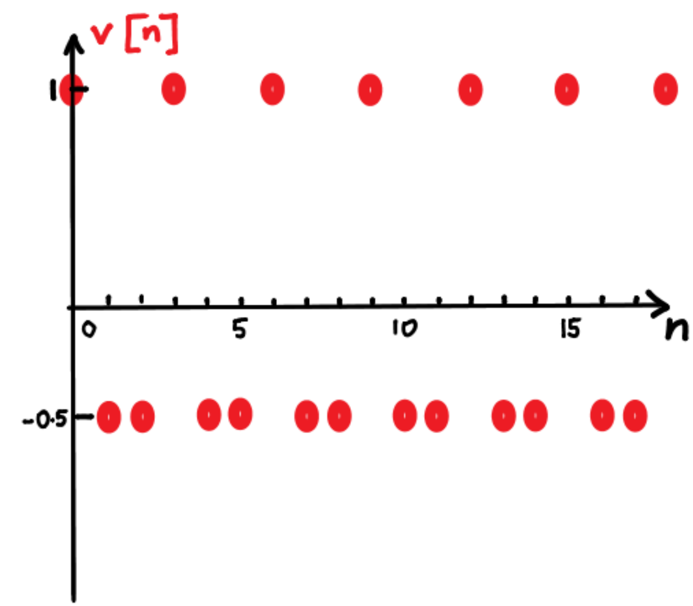
https://www.youtube.com/watch?v=cPhgFM-0WIE

The SAMPLED Signal

The signal $v(t)=\cos(2\pi ft)$ is sampled **uniformly** with 3 **sampling intervals** within each signal period T.
Therefore, the sampling interval $Ts=T/3$ and the sampling rate $f_s=3f$.
Another way see that $f_s=3f$ is to notice that there are three samples in every signal period T.



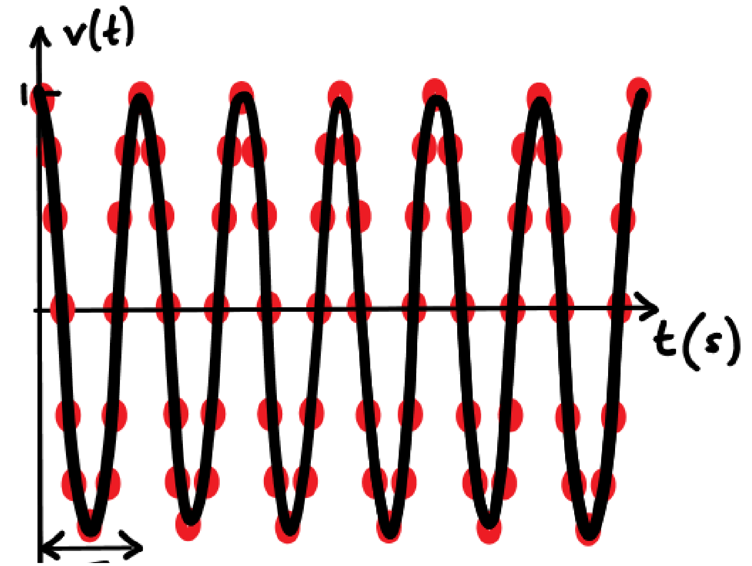
The samples are shown as the sequence $v[n]$ indexed by integer values of n.



If a sinusoidal signal is sampled with a **high sampling rate**, the original signal can be recovered exactly by connecting the samples together in a smooth way (called ideal low pass filtering).

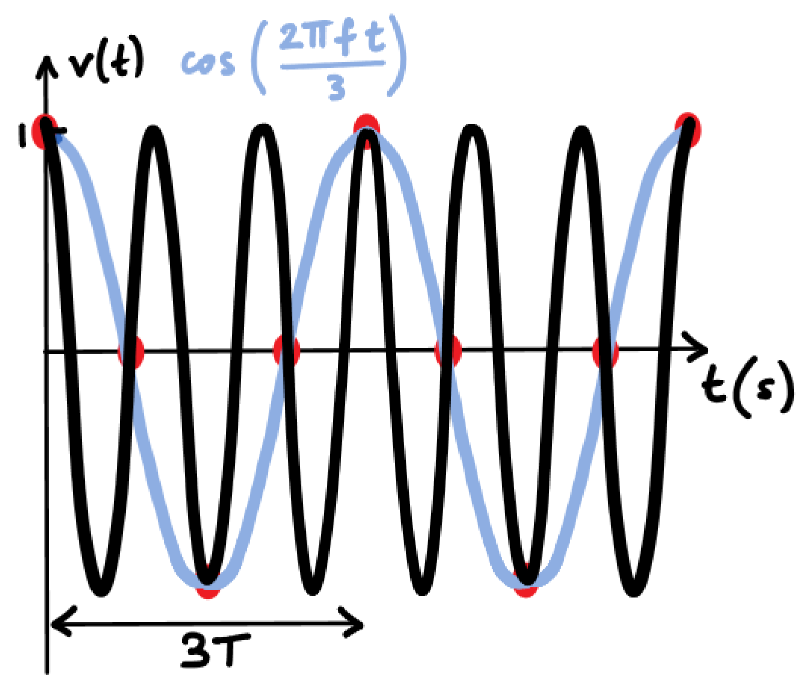
The signal $v(t)=\cos(2\pi ft)$ is sampled **uniformly** with 12 **sampling intervals** within each signal period T.
Therefore, the sampling interval $Ts=T/12$ and the sampling rate $f_s=12f$.

The original signal $v(t)$ **can be recovered** from the samples by connecting them together smoothly.



The signal $v(t)=\cos(2\pi ft)$ is sampled **uniformly** with 4 **sampling intervals** within every 3 signal periods.
Therefore, the sampling rate $f_s=(4/3)f$.

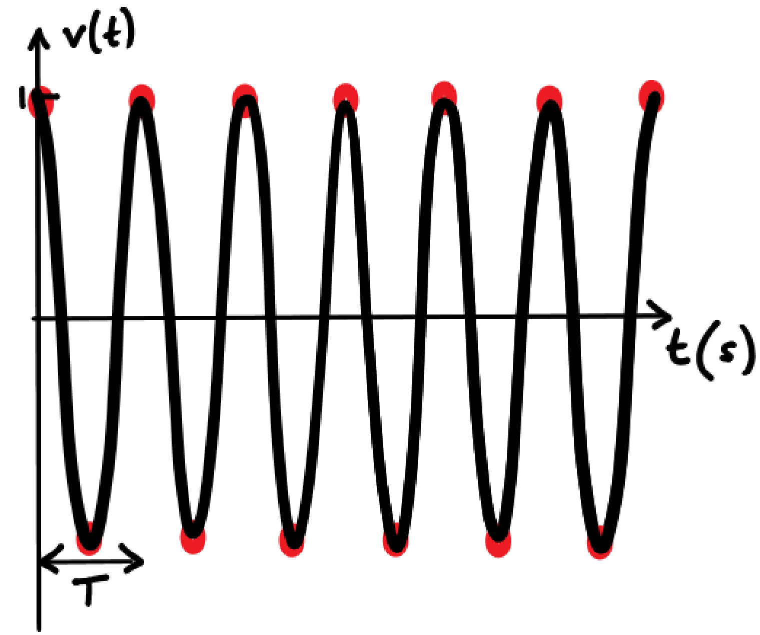
Notice that a different sinusoid $\cos(2\pi ft/3)$ with lower frequency $f/3$ also fits the samples.
Attempting to recover $v(t)=\cos(2\pi ft)$ by ideal low pass filtering instead produces $\cos(2\pi ft/3)$ since the latter has a lower frequency.
So, the sampling rate $f_s=(4/3)f$ is insufficient to recover $v(t)$ from the samples.



For which values of sampling rate f_s can we sample and then perfectly recover a sinusoidal signal $v(t)=\cos(2\pi ft)$?

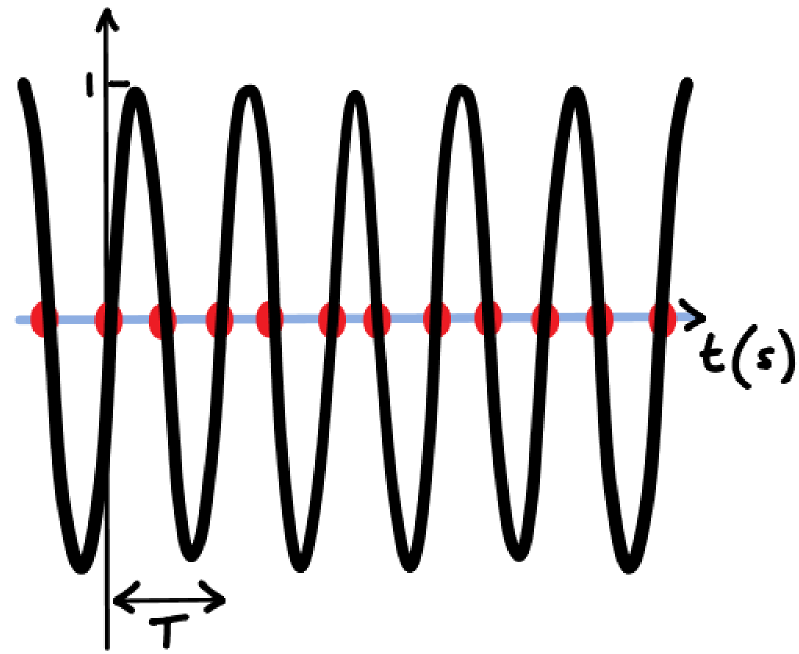
The signal $v(t)=\cos(2\pi ft)$ is sampled **uniformly** with 2 **sampling intervals** within each signal period T.
Therefore, the sampling interval $Ts=T/2$ and the sampling rate $f_s=2f$.

Since there is a sample at every peak and trough of the sinusoid, there is no lower frequency sinusoid that fits these samples. Therefore, $v(t)$ **can be recovered exactly** from the samples by ideal low pass filtering.



The signal $v(t)=\cos(2\pi ft)$ is sampled **uniformly** with 2 **sampling intervals** within each signal period T.
Therefore, the sampling interval $Ts=T/2$ and the sampling rate $f_s=2f$.

Since all the samples are at the zero crossings, ideal low pass filtering produces a **zero signal** instead of recovering the sinusoid.



The Nyquist-Shannon sampling theorem

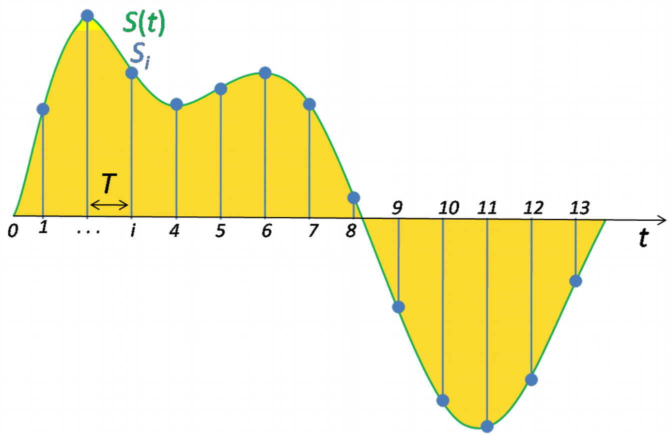
- The Nyquist-Shannon sampling theorem states that the sampling rate for exact recovery of a signal composed of a sum of sinusoids is larger than twice the maximum frequency of the signal. This rate is called the Nyquist sampling rate $f_{Nyquist}$.
  - $f_s > f_{Nyquist} = 2f_{max}$
- For example, if the signal is
  - $7+5\cos(2\pi 440t)+3\sin(2\pi 880t)$
  - then the sampling rate f_s should be chosen to be **larger than**
  - $f_{Nyquist}=2(880)=1760$ Hz.

The Nyquist-Shannon sampling theorem

- NOTE that, no matter how fast we sample, there may exist sinusoids of a sufficiently high frequency for which the sampling rate we are using is too low. So, even where it seems we have no problem recovering the sinusoid, we can't be sure that the true sinusoid is not one of a much higher frequency and we are not sampling fast enough.
- The way around this problem is to assume from the outset that the sinusoids under consideration will have a frequency no more than f_{max}.
- Then, as long as we sample faster than $2f_{max}$, we will be able to recover the original sinusoid exactly.
- → **Bandlimited** signal

We will see that in a few lessons!

In **signal** processing, **sampling** is the reduction of a continuous-time signal to a discrete-time signal. A common example is the conversion of a sound wave (a continuous signal) to a **sequence of samples** (a discrete-time signal). A **sample** is a value or set of values at a point in time and/or space.



The type of sampling mentioned above is sometimes referred to as “ideal” sampling.
In practice, there are usually two non-ideal effects.
One effect is that the sensor obtaining the samples can’t pick off a value at a single time. Instead, some averaging or integration over a small interval occurs, so that the sample actually represents the average value of the analog signal in some interval. This effect can be modeled as a **CONVOLUTION**.

We will see that in a few lessons!

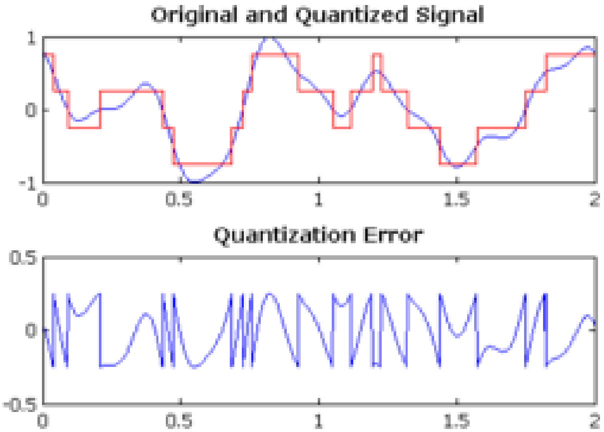
The second non-ideal effect is **quantization** noise. Whether averaged or not, the actual sample value obtained will rarely be the exact value of the underlying analog signal at some time. Noise in the samples is often modeled as adding (usually small) random values to the samples.

We will see that in a few slides!

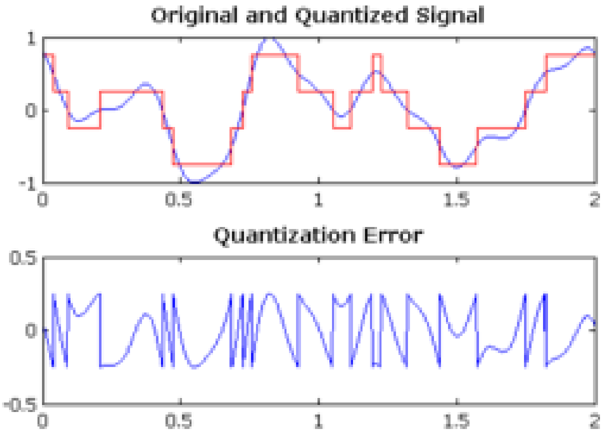
Converting the Signal

- Generally signals are analog in nature
-
- Signal has to be:
  - **Digitized** → Sampled and **Quantized** to *digital form*
  - **Stored** → Copies, Redundancy, Compression
  - **Sent** → Description needed: the **header**

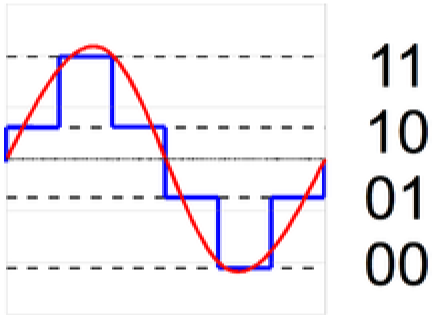
Quantization, in mathematics and digital **signal** processing, is the process of mapping input values from a large set (often a continuous set) to output values in a (**countable**) smaller **set**. Rounding and truncation are typical examples of quantization processes.



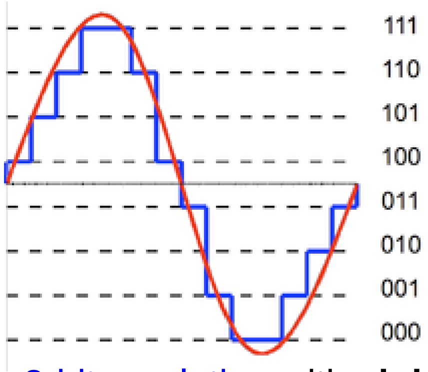
Quantization, in mathematics and digital **signal** processing, is the process of mapping input values from a large set (often a continuous set) to output values in a (**countable**) smaller **set**. Rounding and truncation are typical examples of quantization processes.



A sequence of samples like Sj is not a digital signal because the sample values can potentially take on a **continuous range** of values.
In order to complete the **analog to digital conversion**, each sample value is mapped to a discrete level (represented by a sequence of bits) in a process called quantization. In a B-bit quantizer, each quantization level is represented with B bits, so that the number of levels equals 2^B

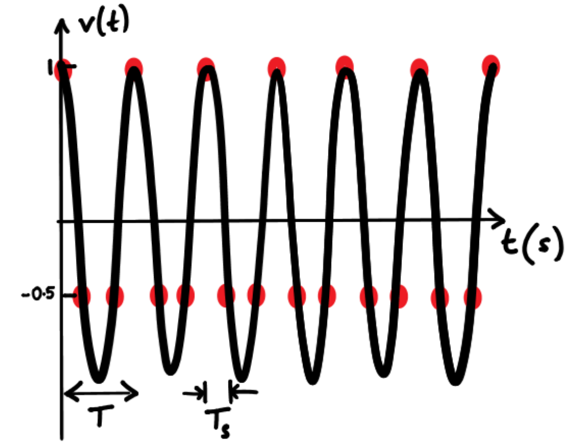


2-bit resolution with **four** levels of quantization compared to analog.

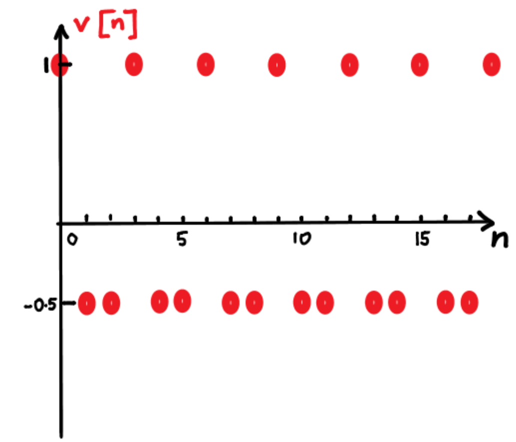


3-bit resolution with **eight** levels of quantization compared to analog.

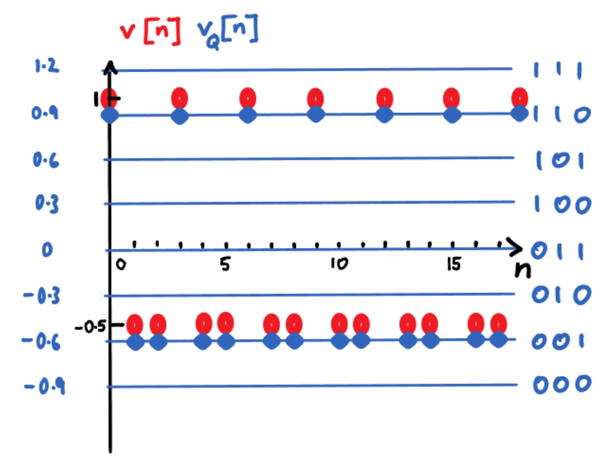
The original signal $\mathrm{v(t)}$ is sampled **uniformly** with sampling rate $\mathrm{f_s}$.



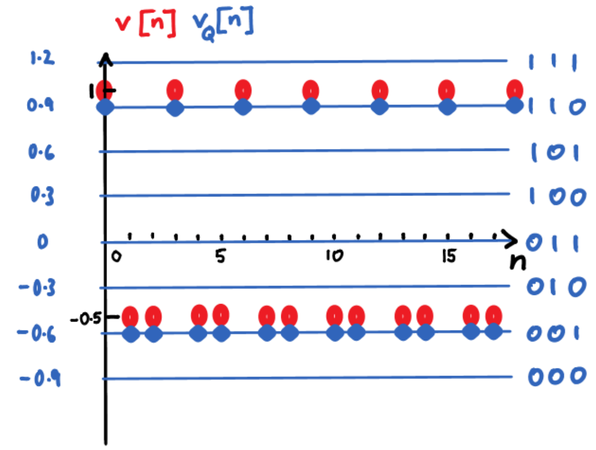
The samples are shown as the sequence $\mathrm{v[n]}$.



Overlaid on the samples $\mathrm{v[n]}$ is a **3-bit resolution quantization** with 8 uniformly spaced quantization levels.
The **quantization approximates** each sample value in $\mathrm{v[n]}$ to its nearest level value, producing the quantized sequence $\mathrm{vQ[n]}$.
The sequence $\mathrm{vQ[n]}$ can be written as a sequence of bits using the 3-bit representations shown on the right.



Overlaid on the samples $\mathrm{v}[\mathrm{n}]$ is a **3-bit resolution quantization** with 8 uniformly spaced quantization levels.
The **quantization approximates** each sample value in $\mathrm{v}[\mathrm{n}]$ to its nearest level value, producing the quantized sequence $\mathrm{vQ}[\mathrm{n}]$.
The sequence $\mathrm{vQ}[\mathrm{n}]$ can be written as a sequence of bits using the 3-bit representations shown on the right.



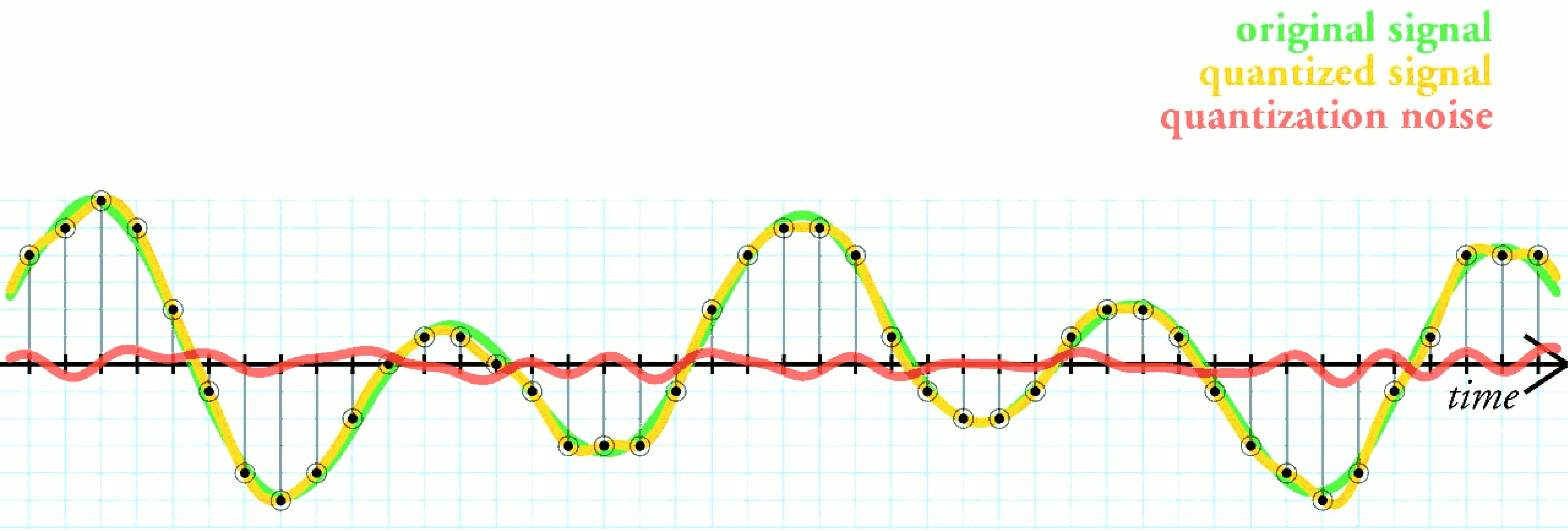
Note that quantization introduces a **quantization error** between the samples and their quantized versions given by $\mathrm{e}[\mathrm{n}]=\mathrm{v}[\mathrm{n}]-\mathrm{vQ}[\mathrm{n}]$.

If a sample lies between quantization levels, the **maximum absolute quantization error** $|\mathrm{e}[\mathrm{n}]|$ is **half of the spacing** between those levels.

For the quantization in the upper figure, the maximum error between levels is 0.15 since the spacing is uniformly 0.3.

Note, however, that if the sample **overshoots** the highest level or undershoots the lowest level by more than 0.15, the absolute quantization error will be larger than 0.15.

Example of quantization error



The simplest way to quantize a signal is to choose the digital amplitude value closest to the original analog amplitude.
This example shows the original analog signal (green), the quantized signal (black dots), the signal reconstructed from the quantized signal (yellow) and the difference between the original signal and the reconstructed signal (red). The difference between the original signal and the reconstructed signal is the quantization error and, in this simple quantization scheme, is a deterministic function of the input signal.

Example of quantization error

As an example, **rounding a real number x to the nearest integer value** forms a very basic type of quantizer: a uniform one.

A typical uniform quantizer with a quantization step size equal to some value Δ can be expressed as

$$Q(x) = \Delta \cdot \text{floor} \left(x/\Delta + 1/2 \right)$$

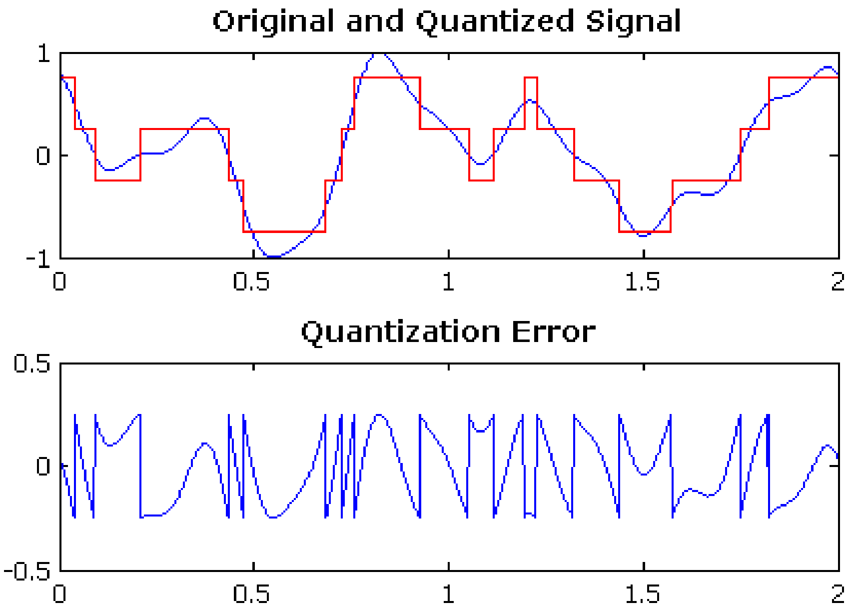
For simple rounding to the nearest integer, the step size Δ is equal to 1.
With $\Delta = 1$ or with Δ equal to any other integer value, this quantizer has real-valued inputs and integer-valued outputs, although this property is not a necessity (a quantizer may also have an integer input domain and may also have non-integer output values).

The essential property of a quantizer is that it has a **countable set of possible output values** that has fewer members than the set of possible input values. The members of the set of output values may have integer, rational, or real values (or even other possible values as well, in general – such as vector values or complex numbers).

When the quantization step size is small (relative to the variation in the signal being measured), it can be demonstrated that the mean squared error produced by such a rounding operation will be $o(\Delta^2/12)$.

More on quantization error

- A common assumption for the analysis of quantization error is that it affects a signal processing system in a similar manner to that of additive white noise: having negligible correlation with the signal and an approximately flat power spectral density.
- The additive noise model is commonly used for the analysis of quantization error effects in digital filtering systems, and it can be very useful in such analysis. It is valid in cases of high resolution quantization (small Δ relative to the signal strength).
- However, additive noise behaviour is not always a valid assumption, and care should be taken to avoid assuming that this model always applies.
- Actually, the quantization error is deterministically related to the signal rather than being independent of it. Therefore, periodic signals can create periodic quantization noise.

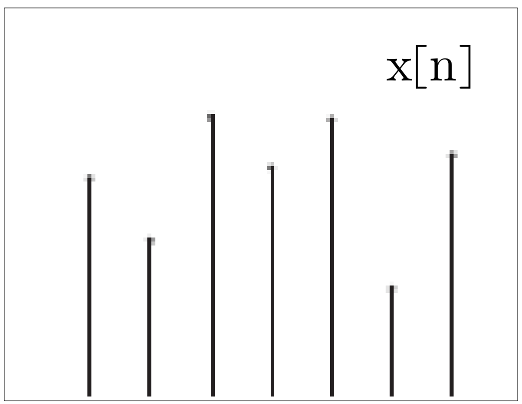


We will see that in a few lessons!

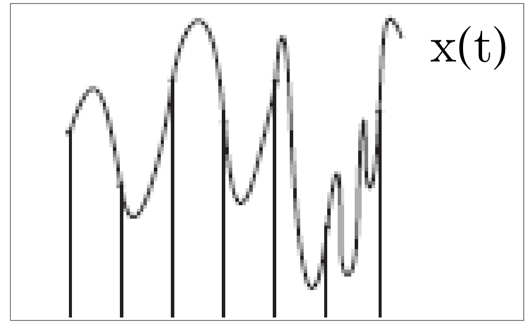
Brief Resume

- Sampling converts an analog signal (function of time) into a discrete-time signal (sequence of real numbers). Quantization replaces each real number with an approximation from a finite set of discrete values (levels). This is necessary for storage and processing by numerical methods.
- Most commonly, these discrete values are represented as **fixed-point words** or **floating-point words**. Common word-lengths are **8-bit** (256 levels), **16-bit** (65,536 levels), **32-bit** (4.3 billion levels), and so on, though any number of quantization levels is possible (not just powers of two).
- Quantizing a sequence of numbers produces a sequence of quantization errors which is sometimes modeled as an additive random signal called quantization noise because of its stochastic behavior. The more levels a quantizer uses, the lower is its quantization noise power.
- In general, both processes lose some information. So **discrete-valued signals** are only an approximation of the **continuous-valued discrete-time signal**, which is itself only an approximation of the original **continuous-valued continuous-time signal**. But both types of approximation errors can, in theory, be made arbitrarily small by good design.

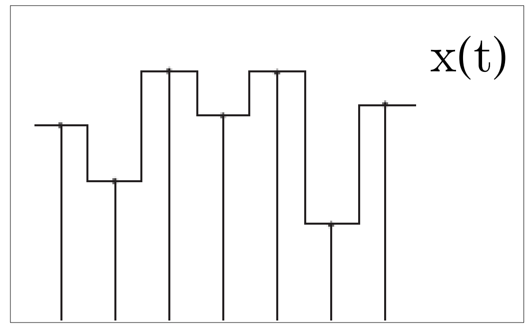
How to construct a continuous-time signal given discrete-time samples?



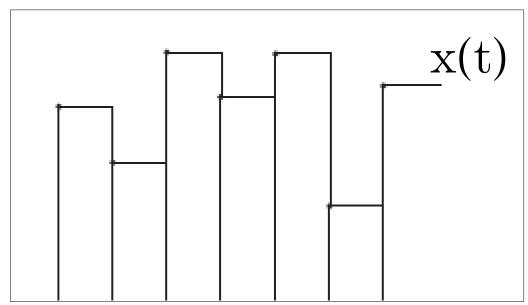
Suppose we are given a set of samples $x[n]$ that we know came from some continuous-time signal $x(t)$. We also know the sampling rate T, so that we know $x(nT) = x[n]$.



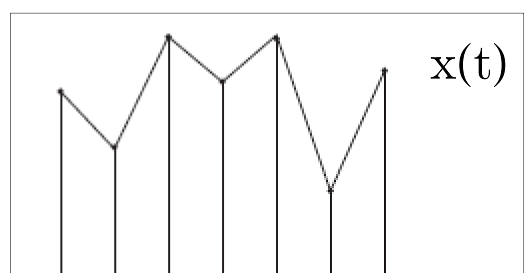
My own reconstruction



Nearest-neighbor reconstruction



Zero-order hold reconstruction



First-order hold reconstruction

The CD capacity

- An audio CD holds up to 74 minutes, 33 seconds of sound, just enough for a complete mono recording of Ludwig von Beethoven's Ninth Symphony at probably the slowest pace it has ever been played.
- CDs use a sampling rate of 44.1 kHz with 16-bit quantization for each sample. When the CD was first introduced in 1983, every 8 bits of digital signal data were encoded as 17 bits of signal and error correction data together. Given that 8 bits are 1 byte and that 2^{20} bytes are 1 megabyte (MB), we calculate that the capacity of a compact disc is about 800 MB.

Duration of the analog signal =(74 min)(60smin)+33s=4473 s
Samples in signal data =(4473 s)(44100samples/s)=197300000 samples
Bits of digital signal data =(197300000 samples)(16bits/sample)=3156000000 bits
Bytes of digital signal data =(3156000000 bits)(1byte/8bits)=394500000 bytes
MB of digital signal data =(394500000 bytes)(1MB/1024bytes)=376.2 MB
MB of signal and error correction data =(376.2 MB)(17bits/8bits)=799.5 MB