ENGR 76: Information Science and Engineering — Winter 2026 — Lectures 9–10

Sampling, Interpolation, and Digital Representation of Signals

Siddharth Chandak

A continuous-time signal cannot be stored or processed directly by a digital system. To represent such a signal digitally, we must sample it in time, quantize its amplitude, and finally represent the resulting symbols using bits. These notes study this pipeline in stages. We first focus on sampling and interpolation, which address whether it is possible to sample in time while still preserving information. We then study what goes wrong when sampling assumptions are violated. Finally, we introduce quantization, which quantizes amplitude and introduces an unavoidable loss of precision.

1 Sampling

In many applications, signals of interest are defined in continuous time. For example, an audio signal corresponds to air pressure as a function of time, and an electrical signal corresponds to voltage varying continuously over time. However, digital systems cannot store or process continuous-time signals directly. Sampling is the first step in converting a continuous-time signal into a form that can be processed digitally.

1.1 What is sampling?

Definition 1 (Sampling)**.** *Given a continuous-time signal $x(t)$ and a sampling period $T_s > 0$, sampling consists of observing the signal at uniform time instants:*

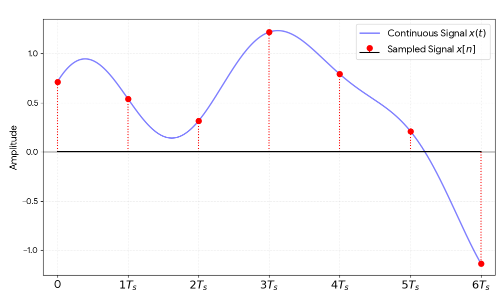
$$\{x(0),\, x(T_s),\, x(2T_s),\, x(3T_s),\, \dots\}.$$

The resulting sequence is called a discrete-time signal*.*

The quantity T_s is called the *sampling period* or *sampling interval*. Equivalently, we define the *sampling frequency*

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

which represents the number of samples taken per second. Importantly, sampling does *not* restrict the range of values the signal can take: the samples $x(kT_s)$ are still real numbers.



Sampling alone does not guarantee that the original signal can be recovered. Since we only observe the signal at isolated time instants, the signal could, in principle, behave arbitrarily between samples. This raises two fundamental questions:

- Is there a class of signals for which the samples uniquely determine the signal?
- What condition does the sampling period T_s (or equivalently, the sampling frequency f_s) need to satisfy for this to hold?

We next answer these questions by restricting our attention to an important class of signals.

1.2 Bandlimited signals

Suppose $x(t)$ is a periodic signal with period T. Recall that it admits a Fourier series representation:

$$x(t) = b_0 + \sum_{j=1}^{\infty} \left(a_j \sin\left(2\pi \frac{j}{T} t\right) + b_j \cos\left(2\pi \frac{j}{T} t\right) \right).$$

Each term corresponds to a sinusoidal component with frequency j/T. It is often convenient to combine the sine and cosine coefficients at a given frequency into a single amplitude

$$A_j = \sqrt{a_j^2 + b_j^2}.$$

The collection $\{A_j\}$ describes how the signal's energy is distributed across frequencies.

Definition 2 (Bandlimited Signal)**.** *A signal is called* bandlimited *if its Fourier series representation contains only finitely many nonzero frequency components. Equivalently, $A_j \neq 0$ for only finitely many values of j.*

Bandlimited signals cannot oscillate arbitrarily fast. Rapid variations in time require high-frequency components; restricting the signal to finitely many frequencies enforces smoothness.

Example 1. *Consider the signal $x(t) = 5 + 2\sin\left(2\pi \frac{10}{T} t\right) - 9\sin\left(2\pi \frac{35}{T} t\right)$. This signal contains only three frequency components and is therefore bandlimited.*

Example 2. *The signal*

$$x(t) = \sum_{j=1}^{\infty} \frac{1}{j} \sin\left(2\pi \frac{j}{T} t\right)$$

contains infinitely many frequency components and is not bandlimited.

Example 3 (Square wave and triangle wave)**.** *Recall the square and triangle waves studied in the previous lectures. These are both* not *bandlimited. A square wave has jump discontinuities. Representing a jump discontinuity requires infinitely many frequency components in its Fourier series. As a result, a square wave is not bandlimited. A triangle wave is continuous but has sharp corners (points where the derivative is discontinuous). These corners also require infinitely many frequency components to represent exactly. Hence, a triangle wave is not bandlimited.*

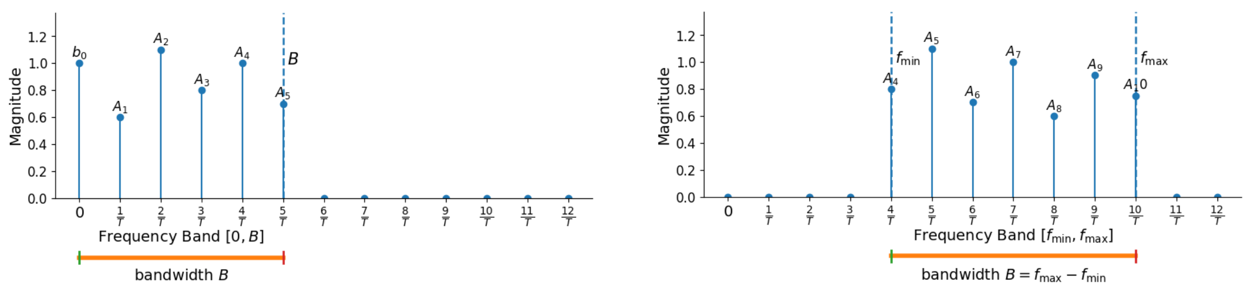
We now define two important subclasses of bandlimited signals.

Definition 3 (Baseband Signal)**.** *A signal is called* baseband *with bandwidth B if all its frequency components lie in the interval* $[0, B]$.

Definition 4 (Passband Signal)**.** *A signal is called* passband *if all its frequency components lie in an interval* $[f_{\min}, f_{\max}]$.

To make this precise, we introduce terminology that describes the range of frequencies present in a signal.

Definition 5 (Frequency band and bandwidth)**.** *The* frequency band *of a signal is the smallest frequency interval that contains all frequencies present in the signal. The* bandwidth *is the width of this interval.*



Example 4. *Consider the signal $x(t) = 15\cos(2\pi 25t) + \sin(2\pi 100t)$. Its frequency components are* 25 *Hz and* 100 *Hz. The frequency band is* $[25, 100]$ *Hz, so the signal is passband with bandwidth* 75 *Hz.*

Example 5. *Consider $x(t) = 12 - \cos(2\pi t) + 18\sin(2\pi 14t)$. The frequency components are* 0*,* 1*, and* 14 *Hz. The frequency band is* $[0, 14]$ *Hz, so the signal is baseband with bandwidth* 14 *Hz.*

Note that whether a signal is baseband or passband depends only on the frequencies present in the signal, not on their amplitudes. Changing the coefficients in front of the sine or cosine terms does not change the frequency components, and hence does not change the frequency band or the bandwidth. For example, the signal $x(t) = 6 + \cos(2\pi t) - 9\sin(2\pi 14t)$ has the same frequency components as the previous example, and therefore the same frequency band and bandwidth, even though the amplitudes are different.

1.3 Representing bandlimited signals over a finite duration

We now show that bandlimited signals have only finitely many degrees of freedom over a finite time interval. Suppose $x(t)$ is a baseband signal with bandwidth B and is modeled as a periodic signal with period T. Since all frequency components satisfy

$$0 \leq \frac{j}{T} \leq B,$$

we must have

$$0 \leq j \leq BT.$$

Thus, the Fourier series truncates:

$$x(t) = b_0 + \sum_{j=1}^{BT} \left(a_j \sin\left(2\pi \frac{j}{T} t\right) + b_j \cos\left(2\pi \frac{j}{T} t\right)\right).$$

The signal is completely determined by the coefficients

$$\{b_0, a_1, b_1, \ldots, a_{BT}, b_{BT}\},$$

which is a total of $2BT + 1$ real numbers.

Key Observation: A baseband signal of bandwidth B and duration T has only $2BT+1$ degrees of freedom.

1.4 Sampling as a system of linear equations

Each sample of the signal provides one linear equation in the unknown Fourier coefficients. For example, let $T = 1$ and $B = 2$. Then

$$x(t) = b_0 + a_1 \sin(2\pi t) + b_1 \cos(2\pi t) + a_2 \sin(4\pi t) + b_2 \cos(4\pi t).$$

Sampling at $t = 0$ gives

$$x(0) = b_0 + b_1 + b_2.$$

Sampling at $t = T_s$ gives

$$x(T_s) = b_0 + a_1 \sin(2\pi T_s) + b_1 \cos(2\pi T_s) + a_2 \sin(4\pi T_s) + b_2 \cos(4\pi T_s).$$

Each additional sample produces another linear equation. To uniquely determine all $2BT + 1$ unknown coefficients, we need at least $2BT + 1$ independent equations. If we sample a signal of duration T every T_s seconds, the total number of samples is

$$\frac{T}{T_s} + 1.$$

To recover all $2BT + 1$ coefficients, we require

$$\frac{T}{T_s} + 1 > 2BT + 1,$$

which simplifies to

$$T_s < \frac{1}{2B} \quad \Longleftrightarrow \quad f_s > 2B.$$

We avoid the case of equality since it corresponds to a critical situation where the system of equations becomes just-determined and may fail to have a unique solution depending on the sampling instants. Requiring a strict inequality ensures that we have strictly more equations than unknowns, and hence guarantees unique recovery of the signal.

Fact 1. *Let $x(t)$ be a baseband signal with bandwidth B. If the sampling frequency satisfies $f_s > 2B$, then the samples $\{x(mT_s)\}$ uniquely determine the signal.*

This fact states that, under the condition $f_s > 2B$, the samples contain all the information about the original signal. In other words, among all bandlimited signals with bandwidth B, there is exactly one signal that is consistent with the given set of samples. Sampling fast enough therefore does not lose information for this class of signals. The importance of this fact lies in the observation that many natural signals of interest are either baseband to begin with or can be made baseband by appropriate preprocessing. For example, the range of frequencies audible to humans is approximately limited to 20 kHz. Audio signals are therefore modeled as baseband signals with bandwidth $B \approx 20$ kHz. Sampling such signals at a rate exceeding $2B$, for instance at 44.1 kHz, guarantees that no information is lost in the sampling process.

This result is surprising for several reasons. A continuous-time signal has infinitely many values, whereas sampling keeps only values at discrete time instants. Between samples, the signal could, in principle, behave arbitrarily. It is not obvious that a finite sampling rate could capture all the information in a signal defined for all real times. What makes this possible is the structure imposed by bandlimitedness. Bandlimited signals are inherently smooth in time and cannot exhibit arbitrarily rapid changes. Sampling at a rate greater than $2B$ captures all the degrees of freedom of such signals, which is why the samples uniquely determine the signal everywhere in time.

This leads to a third fundamental question:

- Given the samples $\{x(mT_s)\}$, how do we explicitly reconstruct the continuous-time signal?

2 Interpolation

So far, we have established that if a baseband signal is sampled at a rate $f_s > 2B$, then the samples uniquely determine the signal. This guarantees that no information is lost in sampling, but it does not yet tell us how to recover the signal explicitly. Interpolation addresses the following question: given the samples $\{x(mT_s)\}_{m\in\mathbb{Z}}$, how do we construct a continuous-time signal $\hat{x}(t)$ that agrees with these samples?

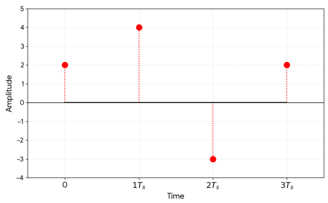
2.1 What is interpolation?

Definition 6 (Interpolation). *Interpolation is the process of constructing a continuous-time signal $\hat{x}(t)$ from a discrete set of samples $\{x(mT_s)\}$ by specifying how the signal behaves between sampling instants.*

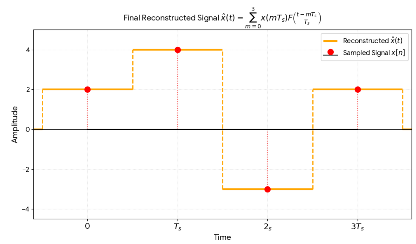
A basic requirement for any interpolation scheme is

$$\hat{x}(mT_s) = x(mT_s) \quad \text{for all integers } m,$$

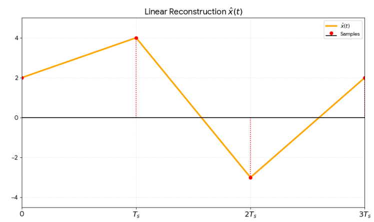
since we know the exact value of the signal at the sampling instants and the reconstructed signal must agree with these values. We first describe two simple interpolation schemes that satisfy the basic interpolation requirement.



(a) Samples



(b) Nearest Neighbor Interpolation



(c) Linear Interpolation

Nearest-neighbor interpolation. One of the simplest interpolation schemes is nearest-neighbor interpolation. In this method, the reconstructed signal is held constant between sampling instants, taking the value of the nearest sample. Given samples $\{x(mT_s)\}$, the reconstructed signal $\hat{x}_{\text{nn}}(t)$ is defined by

$$\hat{x}_{\text{nn}}(t) = x(mT_s) \quad \text{for } t \in \left[mT_s - \tfrac{T_s}{2}, \, mT_s + \tfrac{T_s}{2}\right).$$

Nearest-neighbor interpolation satisfies the basic interpolation requirement, but the reconstructed signal has jump discontinuities at the midpoints between samples.

Linear interpolation Another common interpolation method is linear interpolation, where consecutive samples are connected by straight line segments. Given samples $\{x(mT_s)\}$, the reconstructed signal $\hat{x}_{\text{lin}}(t)$ is defined by

$$\hat{x}_{\text{lin}}(t) = x(mT_s) + \frac{x((m+1)T_s) - x(mT_s)}{T_s}\,(t - mT_s), \quad t \in [mT_s, (m+1)T_s].$$

Linear interpolation also satisfies $\hat{x}(mT_s) = x(mT_s)$ for all m, and produces a continuous signal.

Although nearest-neighbor and linear interpolation both match the samples exactly, neither can produce perfect reconstruction of a bandlimited signal. Nearest-neighbor interpolation introduces jump discontinuities, while linear interpolation introduces sharp corners at the sampling instants. Both of these features require infinitely many frequency components to represent, and hence the resulting signals are not bandlimited. Since the original signal is bandlimited, neither interpolation scheme can recover it exactly, regardless of how fast the signal is sampled.

2.2 General interpolation framework

We now describe a general framework that includes all interpolation schemes discussed so far. An interpolated signal can be written as

$$\hat{x}(t) = \sum_{m=-\infty}^{\infty} x(mT_s)\, F\left(\frac{t - mT_s}{T_s}\right),$$

where $F(t)$ is called the *interpolation function.*

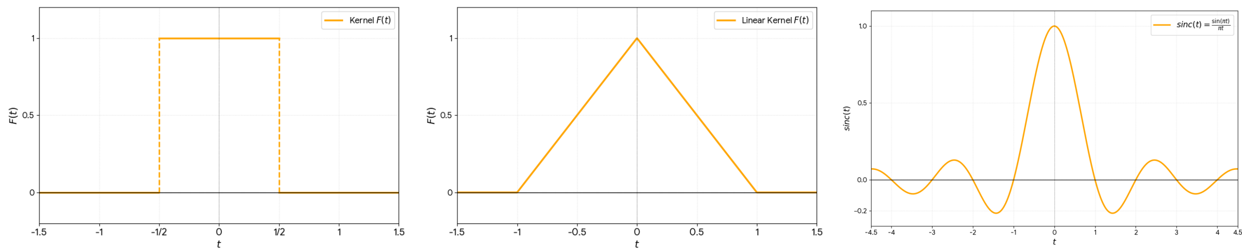
Definition 7 (Valid interpolation function)**.** *A function $F(t)$ is a valid interpolation function if*

- $F(0) = 1$, *and*
- $F(k) = 0$ *for all nonzero integers k.*

To see why these conditions are required, evaluate $\hat{x}(t)$ at a sampling instant $t = mT_s$:

$$\hat{x}(mT_s) = \sum_{k=-\infty}^{\infty} x(kT_s)\, F(m-k).$$

2.2.1 Understanding the interpolation formula



(a) NN Interpolation Function (b) Linear Interpolation Function (c) Sinc Interpolation Function

The interpolation formula

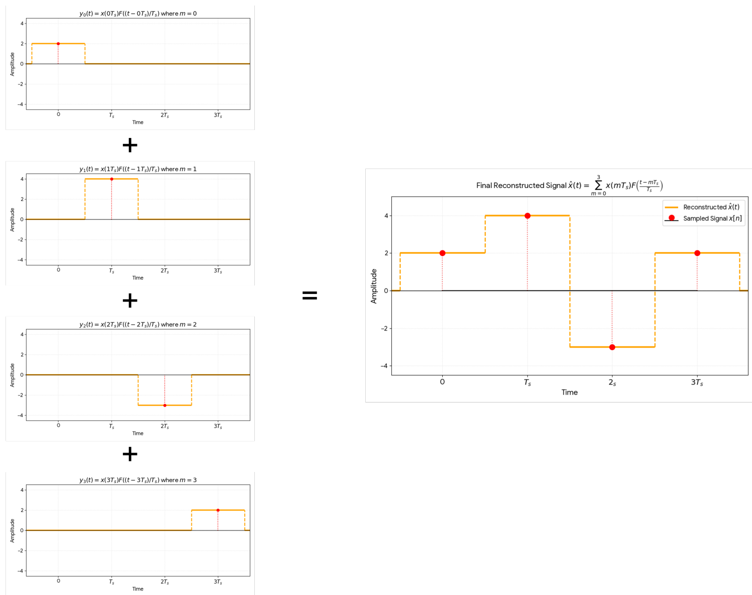
$$\hat{x}(t) = \sum_{m=-\infty}^{\infty} x(mT_s)\, F\left(\frac{t-mT_s}{T_s}\right)$$

can be understood as a superposition of contributions from all samples. Each sample value $x(mT_s)$ contributes to the reconstructed signal at time t, with its influence weighted by the function $F\left(\frac{t-mT_s}{T_s}\right)$. When t is close to mT_s, this weight is large; when t is far from mT_s, the weight is small or zero, depending on the choice of F.

The function $F(\cdot)$ should be interpreted as a *kernel* that describes how a single sample influences nearby time instants. The argument $\frac{t-mT_s}{T_s}$ has two important effects. First, subtracting mT_s shifts the kernel so that it is centered at the sampling instant $t = mT_s$. Second, dividing by T_s scales time so that the kernel has the same shape relative to the sampling interval, regardless of the actual sampling rate.

Equivalently, $F\left(\frac{t}{T_s}\right)$ describes the influence of a sample located at time 0, while $F\left(\frac{t-mT_s}{T_s}\right)$ is the same kernel shifted to be centered at mT_s. Interpolation therefore consists of placing identical, shifted copies of the kernel at every sampling instant, scaling each copy by the corresponding sample value, and summing the resulting functions (a worked out example for the nearest neighbor case is presented below).

Different interpolation schemes correspond to different choices of the kernel F. These choices determine how far each sample's influence extends in time, as well as the smoothness and frequency content of the reconstructed signal.



2.3 Sinc Interpolation

One might attempt to construct a valid interpolation function using a polynomial. To force zeros at all nonzero integers, we would need to include factors corresponding to every integer:

$$F(t) = (1-t)(1+t)\left(1-\frac{t}{2}\right)\left(1+\frac{t}{2}\right)\left(1-\frac{t}{3}\right)\left(1+\frac{t}{3}\right)\cdots$$

This infinite product produces a function that has zeros at all nonzero integers while remaining equal to 1 at $t = 0$.

The infinite product above corresponds to a well-known function.

Definition 8 (Sinc function)**.** *The sinc function is defined as*

$$sinc(t) = \frac{\sin(\pi t)}{\pi t},$$

with the convention $sinc(0) = 1$*.*

The sinc function satisfies

$$\text{sinc}(0) = 1 \quad \text{and} \quad \text{sinc}(k) = 0 \text{ for all nonzero integers } k,$$

and hence is a valid interpolation function. Using sinc interpolation, the reconstructed signal is

$$\hat{x}(t) = \sum_{m=-\infty}^{\infty} x(mT_s)\,\text{sinc}\left(\frac{t - mT_s}{T_s}\right).$$

We now state the full sampling theorem, which combines the existence result from the sampling section with the explicit reconstruction formula.

Fact 2 (Shannon–Nyquist Sampling Theorem)**.** *Let $x(t)$ be a baseband signal with bandwidth B. If the sampling frequency satisfies*

$$f_s > 2B,$$

then $x(t)$ can be perfectly reconstructed from its samples $\{x(mT_s)\}_{m\in\mathbb{Z}}$ using sinc interpolation:

$$x(t) = \sum_{m=-\infty}^{\infty} x(mT_s)\, \mathit{sinc}\left(\frac{t - mT_s}{T_s}\right) \quad \textit{for all } t.$$

3 Aliasing and the Stroboscopic Effect

When the sampling frequency satisfies $f_s > 2B$, we saw that the samples $\{x(mT_s)\}$ uniquely determine the original bandlimited signal and allow perfect reconstruction using sinc interpolation. We now examine what happens when this condition is not satisfied.

Suppose that $x(t)$ is a baseband signal with bandwidth B, but the sampling frequency satisfies $f_s < 2B$. We can still form the sinc interpolation of the samples,

$$\hat{x}(t) = \sum_{m=-\infty}^{\infty} x(mT_s)\, \mathrm{sinc}\left(\frac{t - mT_s}{T_s}\right),$$

and this reconstructed signal will still match the samples exactly: $\hat{x}(mT_s) = x(mT_s) \quad$ for all integers m. However, the key difference is that the reconstructed signal $\hat{x}(t)$ is always bandlimited to the interval $\left[0, \frac{f_s}{2}\right]$, regardless of the bandwidth of the original signal $x(t)$.

Fact 3. *Let $x(t)$ be bandlimited to $[0, B]$, and let*

$$\hat{x}(t) = \sum_{m=-\infty}^{\infty} x(mT_s)\, \mathit{sinc}\left(\frac{t - mT_s}{T_s}\right).$$

Then $\hat{x}(t)$ is bandlimited to $\left[0, \frac{f_s}{2}\right]$. Moreover,

- *If $f_s > 2B$, then $\hat{x}(t) = x(t)$ for all t.*
- *If $f_s < 2B$, then $\hat{x}(t) \neq x(t)$ in general, even though $\hat{x}(mT_s) = x(mT_s)$ for all m.*

Intuition. Sinc interpolation always produces a signal that agrees with the samples and is bandlimited to $\left[0, \frac{f_s}{2}\right]$. When the sampling frequency is high enough, this bandlimit is wide enough to represent the original signal, and perfect reconstruction is possible. When the sampling frequency is too low, the reconstruction is forced to lie in a smaller frequency band than the original signal. As a result, the reconstructed signal is the *best bandlimited signal consistent with the samples*, but it cannot equal the original signal.

This phenomenon is called *aliasing*. High-frequency content in the original signal is forced to appear as lower-frequency behavior in the reconstructed signal because the reconstruction is constrained by the sampling frequency. An interactive visualization showing how the reconstructed signal changes as a function of the sampling frequency can be found here.

3.1 The stroboscopic effect

A visually intuitive manifestation of aliasing is the *stroboscopic effect*. This effect occurs when a continuously moving object is observed only at discrete time instants.

When the observation rate is sufficiently high, the apparent motion inferred from the samples matches the true motion of the object. When the observation rate is too low, the sampled positions can be consistent with a different motion altogether. As a result, the object may appear to move more slowly, remain stationary, or even move in the opposite direction. The stroboscopic effect arises for the same reason as aliasing in signals: sampling too slowly constrains the reconstructed motion to have a limited temporal bandwidth. Motion that varies faster than this limit is forced to appear as a slower, aliased motion that matches the observed samples.

An interactive visualization of the stroboscopic (wagon-wheel) effect can be found here.

4 Quantization and Dequantization

Sampling converts a continuous-time signal into a discrete-time sequence, but the sampled values $x(kT_s)$ are still real numbers. Digital systems cannot store real numbers with infinite precision. To represent these values using a finite number of bits, we must further restrict the set of values they are allowed to take. This process is known as quantization.

Definition 9 (Quantization)**.** *Quantization is the process of mapping a real-valued number to one of finitely many discrete levels.*

Quantization operates on the amplitude of the signal, not on time. While sampling determines *when* the signal is observed, quantization determines *how precisely* each observed value is represented.

Uniform quantization. The simplest and most commonly studied form of quantization is *uniform quantization*. Suppose the signal values are known to lie in an interval $[a, b]$. In uniform quantization, this interval is divided into N equal-width bins, each of width $\Delta = (b - a)/N$.

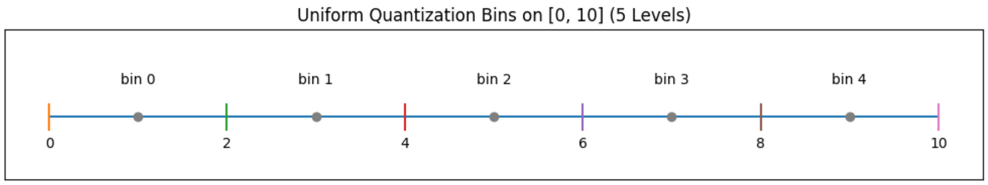
Quantization index representation. After quantization, each sample is represented by a discrete symbol (the bin index). These indices can then be encoded using bits. If there are N quantization levels, then each index can be represented using $\log_2 N$ bits. This makes explicit the tradeoff introduced by quantization: increasing N improves precision but increases the number of bits required per sample, while decreasing N reduces the number of bits but increases distortion.

4.1 Dequantization.

Quantized values are not directly useful for reconstruction, since they only identify bins, not actual signal values. Dequantization assigns each quantization index a representative real value.

Definition 10 (Dequantization)**.** *Dequantization maps a quantized symbol back to a representative real value, typically chosen as the midpoint of the corresponding quantization bin.*

For uniform quantization, if a sample is mapped to a bin spanning $[a + i\Delta,\ a + (i + 1)\Delta)$, a common choice of reconstruction value is $a + (i + \frac{1}{2})\Delta$.



Example 6 (Uniform quantization with $N = 5$ bins)**.** *Suppose a real-valued sample X lies in the interval $[0, 10]$ and we apply uniform quantization with $N = 5$ bins. The bin width is $\Delta = (10 - 0)/5 = 2$. The five bins are $[0, 2)$, $[2, 4)$, $[4, 6)$, $[6, 8)$, and $[8, 10]$. Consider the sample value $X = 6.3$. Since $6.3 \in [6, 8)$, the quantizer outputs the bin index 3. Dequantization maps this index to the midpoint of the bin, so the reconstructed value is $\hat{X} = 7$.*

Quantization error. The difference between the original sample value and its dequantized value is called the *quantization error*. For uniform quantization, this error is bounded in magnitude by $\Delta/2$. Unlike sampling (under the Nyquist condition), quantization is inherently lossy. Once a value has been quantized, the exact original value cannot be recovered. Reducing the number of quantization levels further increases the loss.

5 From Continuous-Time Signals to Bits and Back

We now summarize the complete digital signal representation pipeline.

Forward direction (digitization).

1. **Sampling:** Convert a continuous-time signal $x(t)$ into a discrete-time sequence $\{x(kT_s)\}$.
2. **Quantization:** Map each real-valued sample to one of finitely many discrete levels.
3. **Coding:** Represent the resulting discrete symbols using bits, exploiting statistical structure when possible (e.g., Huffman coding).

Backward direction (reconstruction).

1. **Decoding:** Recover discrete symbols from the bitstream.
2. **Dequantization:** Map symbols back to representative real values.
3. **Interpolation:** Construct a continuous-time signal from the dequantized samples.

Sampling and interpolation determine whether converting a continuous-time signal into a sequence indexed by time preserves all the information in the original signal. For bandlimited signals sampled at a rate $f_s > 2B$, sampling followed by sinc interpolation is *lossless*: the original signal can be recovered exactly, with no loss of information.

Quantization plays a fundamentally different role. It limits the precision with which each sample value is represented by mapping real-valued amplitudes to a finite set of levels. Unlike sampling, quantization is inherently *lossy*. Once a value has been quantized, the exact original value cannot be recovered, regardless of how sophisticated the reconstruction method is. Increasing the number of quantization levels reduces the distortion, but it can never eliminate it entirely.

The final step is representation using bits. Once the samples have been quantized to a finite alphabet, source coding techniques are used to represent these discrete symbols as bit sequences. This step is *lossless*: the quantized symbols can be recovered exactly from the bitstream. Thus, in the digital signal representation pipeline, any irreversible loss of information comes solely from quantization, not from sampling, or coding (as long as sampling frequency satisfies the Nyquist condition).