# Signal & Systems

## Lecture 5: Sampling, Aliasing and Discrete Time Fourier Transform

Konstantinos Chatzilygeroudis - costashatz@upatras.gr

Department of Electrical and Computer Engineering
University of Patras

Template made by Panagiotis Papagiannopoulos

# Why Sampling?

- The physical world produces **continuous-time signals**
  - sound waves
  - electromagnetic signals
  - biological signals (ECG, EEG)
- Computers and digital systems process **discrete-time signals**
- Therefore we must convert:

$$\text{Continuous-time signal } x(t) \quad \rightarrow \quad \text{Discrete samples } x[n]$$

- **Key Question:**

How many samples do we need to represent a signal without losing information?

# Sampling in Practice

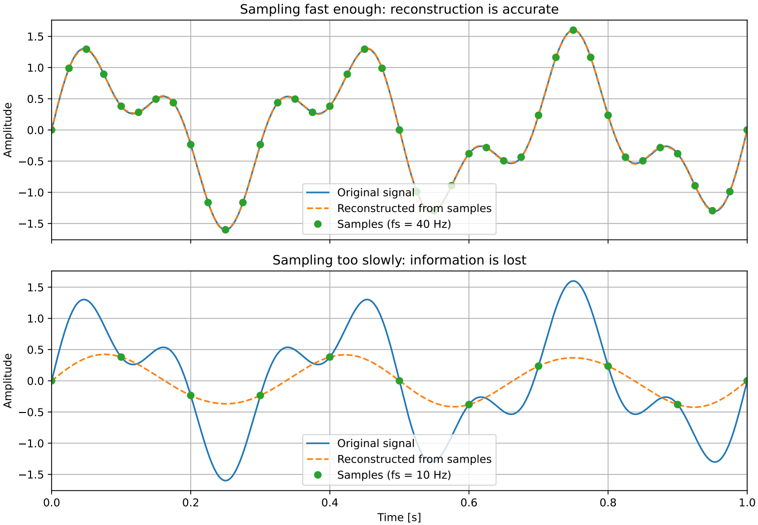
Sampling is everywhere in modern technology:

- **Audio recording**
  - CD audio: $f_s = 44.1$ kHz
- **Digital music and streaming**
- **Medical devices**
  - ECG, EEG
- **Communication systems**
  - software-defined radio
- **Digital cameras and video**

**But what happens if we sample too slowly?**

We lose information!

# Sampling: Motivation



## Sampling a Continuous-Time Signal

- A continuous-time signal is a function of time:

$$x(t)$$

- Digital systems cannot process continuous signals directly.
- Instead, we measure the signal at **discrete time instants**.

### Sampling

Sampling is the process of converting a continuous-time signal into a sequence of discrete samples.

The result is a **discrete-time signal**:

$$x[n]$$

The most common sampling scheme is **uniform sampling**.

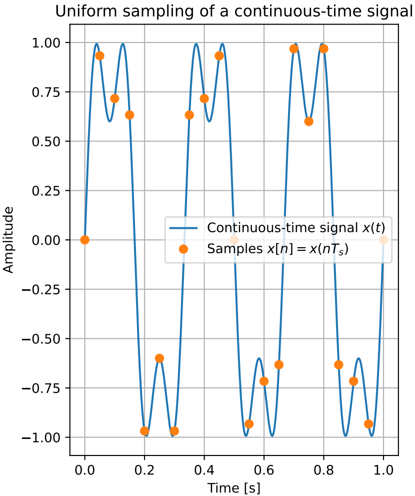
$$x[n] = x(nT_s)$$

- $T_s$ : sampling period
- $f_s = \frac{1}{T_s}$ : sampling frequency
- $n \in \mathbb{Z}$ : sample index

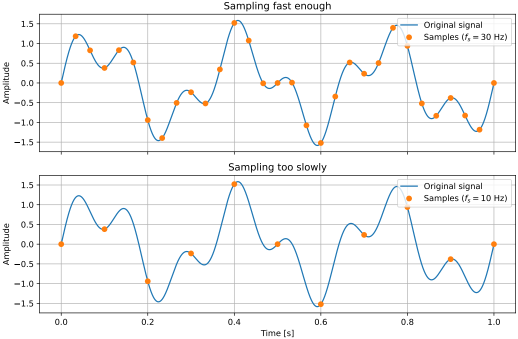
Interpretation:

- We measure the signal every $T_s$ seconds.
- This produces a sequence of numbers:

$$x[0], x[1], x[2], \ldots$$



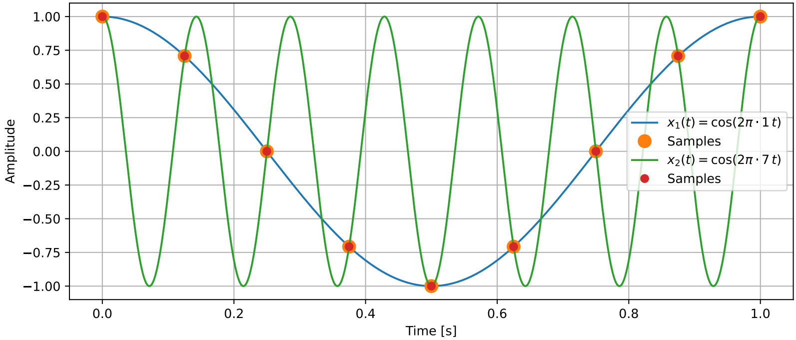
# Effect of the Sampling Rate




**Sampling fast enough:** the samples capture the waveform well.
**Sampling too slowly:** important variations are missed.

This motivates the question:
**How large must $f_s$ be to avoid losing information?**

# Effect of the Sampling Rate (2)



# Mathematical Model of Sampling

Sampling can be modeled mathematically using an **impulse train**:

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

The sampled signal is obtained by multiplication:

$$x_s(t) = x(t)\, s(t)$$

Therefore,

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\, \delta(t - nT_s)$$

- Each sample becomes a weighted impulse.
- This representation is useful for frequency-domain analysis.

# Spectrum of the Sampling Impulse Train

Sampling multiplies the signal by an impulse train:

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

Its Fourier transform is also an impulse train:

$$S(\omega) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_s)$$

where $\omega_s = \frac{2\pi}{T_s}$ is the **sampling angular frequency**.

### Key idea

Sampling introduces periodic impulses in the frequency domain.

## Spectrum of the Sampling Impulse Train (2)

We compute its Fourier transform:

$$S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt$$

Substitute the impulse train:

$$S(\omega) = \int_{-\infty}^{\infty} \left( \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \right) e^{-j\omega t} dt$$

Using the sifting property of the delta function:

$$S(\omega) = \sum_{n=-\infty}^{\infty} e^{-j\omega n T_s}$$

This infinite exponential sum corresponds to a periodic impulse train:

$$S(\omega) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_s)$$

The sampled signal is

$$x_s(t) = x(t)s(t)$$

Multiplication in time corresponds to convolution in frequency:

$$X_s(\omega) = \frac{1}{2\pi} X(\omega) * S(\omega)$$

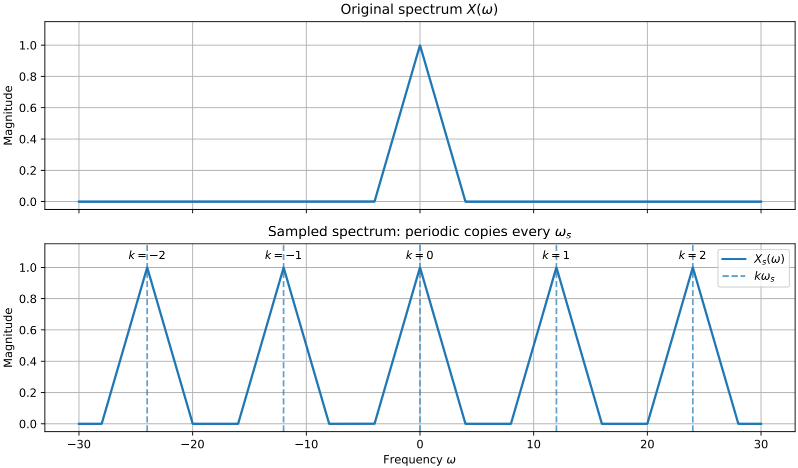
Substituting the spectrum of the impulse train:

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)$$

### Interpretation

Sampling creates **periodic copies of the spectrum** spaced by the sampling frequency.

# Sampling Creates Periodic Copies in Frequency



# Ideal Reconstruction Filter

To recover the original signal, we apply a low-pass filter. **Why?**

# Ideal Reconstruction Filter

To recover the original signal, we apply a low-pass filter. **Why?**

$$H(\omega) = \begin{cases} T_s & |\omega| < \omega_s/2 \\ 0 & \text{otherwise} \end{cases}$$

This filter:

- keeps the baseband spectrum
- removes all replicated copies

After filtering:

$$X(\omega) = H(\omega)X_s(\omega)$$

### Result

The original spectrum is recovered.

# Perfect Reconstruction in Time Domain

The inverse Fourier transform of the ideal low-pass filter gives

# Perfect Reconstruction in Time Domain

The inverse Fourier transform of the ideal low-pass filter gives

$$h(t) = \text{sinc}\left(\frac{t}{T_s}\right)$$

The reconstructed signal is

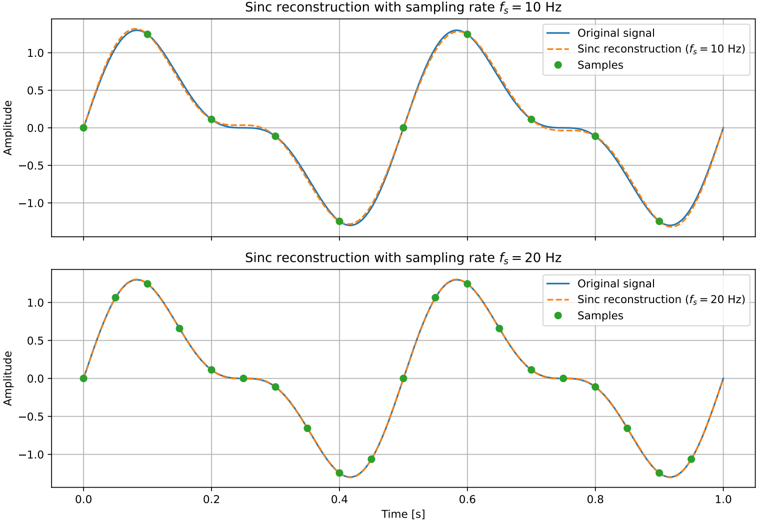
$$x(t) = \sum_{n=-\infty}^{\infty} x[n]\,\text{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

- Each sample generates a shifted sinc function
- The sum of all sinc functions reconstructs the signal

### Key idea

Perfect reconstruction is achieved using **sinc interpolation**.

# Sinc Interpolation



# Practical Issues with Sinc Reconstruction

The ideal reconstruction formula assumes:

1. **Infinite support:** $\text{sinc}(t)$ extends to infinity.
2. **Infinite number of samples:** The sum runs from $-\infty$ to $\infty$.
3. **Perfectly bandlimited signals:** Real signals rarely have exact bandwidth limits.

# Practical Reconstruction Methods

In practice we approximate ideal reconstruction.

Common approaches:

- **Finite-length sinc interpolation**
  - truncate the sinc
  - use a limited number of neighboring samples
- **Windowed sinc filters**
  - multiply sinc by a window
  - improves stability
- **Simpler interpolation methods**
  - zero-order hold (DACs)
  - linear interpolation
  - spline interpolation

# Zero-Order Hold Reconstruction

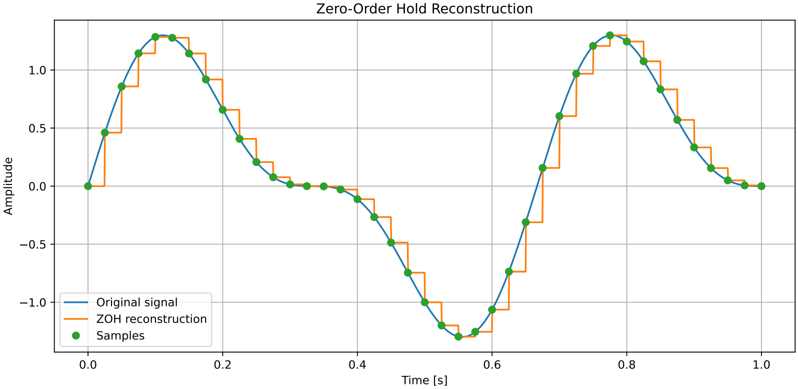
A common practical reconstruction method is **zero-order hold**.
Each sample is held constant for one sampling period.

$$x(t) = x[n] \quad \text{for} \quad nT_s \leqslant t < (n+1)\,T_s$$

Result:

- simple hardware implementation
- produces a staircase signal
- usually followed by an analog low-pass filter

# Zero-Order Hold Reconstruction - Example



# Practical Sinc Reconstruction

Ideal reconstruction formula:

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

**Problem:**

- The sinc function has **infinite support**
- The sum requires **infinitely many samples**
- Real signals are only observed over a **finite interval**

**Practical solution: truncated sinc interpolation**

$$x(t) \approx \sum_{n=n_0-K}^{n_0+K} x[n] \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

- Use only a **finite number of neighboring samples**
- Typical values: $K = 10$–$50$

## Key idea

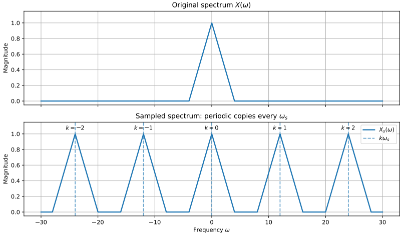
Perfect reconstruction uses infinite sinc functions. Practical systems approximate this using a finite window.

Sampling creates periodic copies of the spectrum:

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)$$

If the sampling frequency is sufficiently large:

- the spectral copies do not overlap
- the original spectrum can be isolated



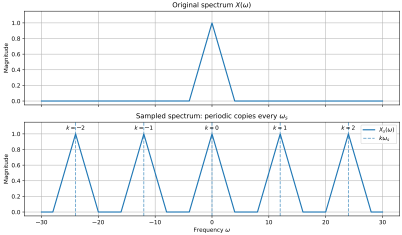
# When Can We Reconstruct a Signal?

Sampling creates periodic copies of the spectrum:

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)$$

If the sampling frequency is sufficiently large:

- the spectral copies do not overlap
- the original spectrum can be isolated




## Key question

How large must the sampling frequency be?

# Nyquist-Shannon Sampling Theorem

## Sampling Theorem

If a signal is bandlimited to frequency $B$,

$$X(\omega) = 0 \quad \text{for } |\omega| > \omega_B$$

then the signal can be perfectly reconstructed if

$$\omega_s > 2\omega_B$$

or equivalently

$$f_s > 2B$$

- $f_s$: sampling frequency
- $B$: signal bandwidth

# Nyquist Frequency

The **Nyquist frequency** is defined as

$$f_N = \frac{f_s}{2}$$

Interpretation:

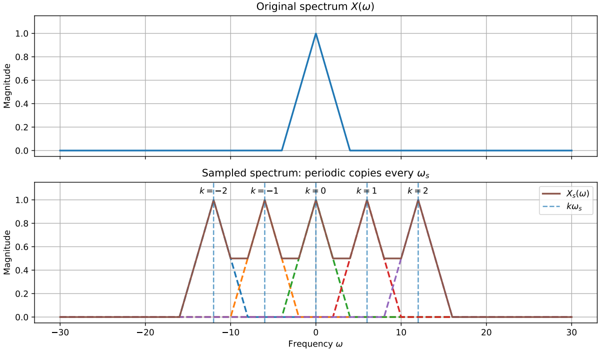
- The Nyquist frequency is the **highest frequency** that can be represented without aliasing.
- To avoid spectral overlap:

$$B < f_N$$

**Equivalent statement of the sampling theorem**

The sampling rate must be at least twice the signal bandwidth.

If the sampling frequency is too small ($f_s < 2B$), then the spectral copies overlap.




## Aliasing

Different frequencies become indistinguishable after sampling.

# Aliasing Breaks Reconstruction

A sinusoid at 7 Hz is sampled at

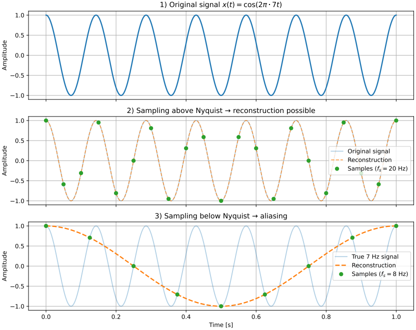
$$f_s = 8 \text{ Hz}$$

Since the Nyquist frequency is

$$f_N = \frac{f_s}{2} = 4 \text{ Hz},$$

the signal is undersampled. Its samples are identical to those of a lower-frequency sinusoid:

$$f_{\text{alias}} = |7 - 8| = 1 \text{ Hz}$$




### Key idea

Different continuous-time signals produce the same samples. Therefore exact reconstruction is impossible.

After sampling we obtain a discrete-time signal:

$$x[n] = x(nT_s)$$

We would like a frequency representation of this signal (or any general discrete-time signal). For continuous-time signals we used the Fourier transform:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

For discrete-time signals we use the **Discrete-Time Fourier Transform (DTFT)**.

# Discrete-Time Fourier Transform

The DTFT of a discrete-time signal $x[n]$ is

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

The inverse transform is

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega$$

- $x[n]$: discrete-time signal
- $X(e^{j\omega})$: frequency representation
- $\omega$: digital angular frequency

# Periodicity of the DTFT

The DTFT is periodic in frequency.

$$X(e^{j(\omega+2\pi)}) = X(e^{j\omega})$$

**Key property**

The spectrum of a discrete-time signal repeats every $2\pi$.

# Periodicity of the DTFT

The DTFT is periodic in frequency.

$$X(e^{j(\omega+2\pi)}) = X(e^{j\omega})$$

### Key property

The spectrum of a discrete-time signal repeats every $2\pi$.

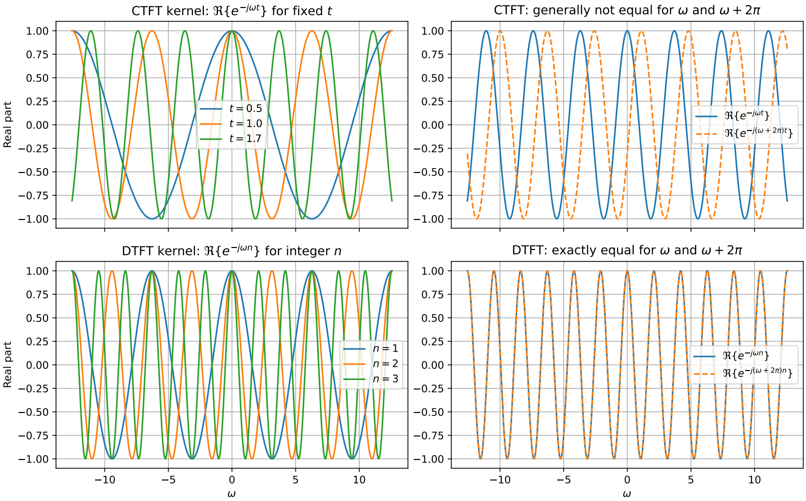
This is because:

$$e^{-j(\omega+2\pi)n} = e^{-j\omega n}$$

and

$$e^{-j2\pi n} = 1$$

for integer $n$.


CTFT kernel: ℜ{e^{−jωt}} for fixed t
CTFT: generally not equal for ω and ω + 2π
DTFT kernel: ℜ{e^{−jωn}} for integer n
DTFT: exactly equal for ω and ω + 2π
Real part
ω
t = 0.5
t = 1.0
t = 1.7
ℜ{e^{−jωt}}
ℜ{e^{−j(ω+2π)t}}
n = 1
n = 2
n = 3
ℜ{e^{−jωn}}
ℜ{e^{−j(ω+2π)n}}

# Digital Frequency

The DTFT uses normalized angular frequency: $\omega$ (radians per sample). If the signal is sampled, then we can get a relation to the physical frequency $f$:

$$\omega = 2\pi \frac{f}{f_s}$$

Important points:

- The frequency range is

$$-\pi \leqslant \omega \leqslant \pi$$

- This corresponds to

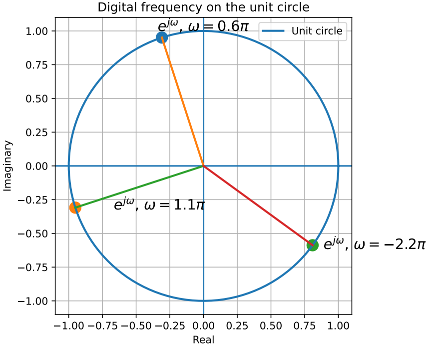
$$-\frac{f_s}{2} \leqslant f \leqslant \frac{f_s}{2}$$

# Digital Frequency and the Unit Circle

The DTFT evaluates the signal spectrum at $X(e^{j\omega})$. This corresponds to points on the **unit circle** in the complex plane:

$$e^{j\omega} = \cos(\omega) + j\sin(\omega)$$

- As $\omega$ varies, the point moves around the unit circle.
- One full revolution corresponds to $\omega = 2\pi$.



# Thank you

- **Any Questions?**
- **Office Hours**:
  - **Mon & Tue** *(09:00-11:00)*
  - 24/7 by email (`costashatz@upatras.gr`, subject: *ECE_SS_AM*)
- **Material and Announcements**

*Laboratory of Automation & Robotics*