Signal & Systems

Lecture 8: Spectral Leakage, Frequency Resolution and Dynamic Range

Konstantinos Chatzilygeroudis - costashatz@upatras.gr

Department of Electrical and Computer Engineering
University of Patras

Template made by Panagiotis Papagiannopoulos

DFT and IDFT

Given a length-N discrete-time sequence $x[n]$, its N**-point Discrete Fourier Transform (DFT)** is

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N}, \qquad k = 0, 1, \ldots, N-1$$

The **Inverse Discrete Fourier Transform (IDFT)** reconstructs the samples:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j2\pi kn/N}, \qquad n = 0, 1, \ldots, N-1$$

Interpretation:

- $x[n]$: signal in the time/sample domain
- $X[k]$: weights of discrete complex exponentials at the DFT frequencies
- The DFT analyzes the signal on the frequency grid $\omega_k = \frac{2\pi k}{N}$

Inverse DFT (IDFT)

Given $X[k]$, the inverse DFT reconstructs the signal:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j2\pi kn/N}$$

Compare with DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N}$$

Differences:

- sign in exponent is reversed
- scaling factor $\frac{1}{N}$

Inverse DFT (IDFT)

Given $X[k]$, the inverse DFT reconstructs the signal:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j2\pi kn/N}$$

Compare with DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N}$$

Differences:

- sign in exponent is reversed
- scaling factor $\frac{1}{N}$

IDFT is very similar to DFT.

Can We Compute IDFT Using FFT?

Observation:

- FFT efficiently computes the DFT
- IDFT has almost the same structure

Key idea:

Can we rewrite the IDFT in terms of a DFT?

Can We Compute IDFT Using FFT?

Observation:

- FFT efficiently computes the DFT
- IDFT has almost the same structure

Key idea:

Can we rewrite the IDFT in terms of a DFT?

Answer: Yes.

We will transform the IDFT into a DFT by using:

- complex conjugation
- scaling

Key Identity for IDFT via FFT

Start from the IDFT:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{j2\pi kn/N}$$

Key Identity for IDFT via FFT

Start from the IDFT:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{j2\pi kn/N}$$

Take complex conjugate:

$$x^*[n] = \frac{1}{N}\sum_{k=0}^{N-1} X^*[k]\, e^{-j2\pi kn/N}$$

Key Identity for IDFT via FFT

Start from the IDFT:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j2\pi kn/N}$$

Take complex conjugate:

$$x^*[n] = \frac{1}{N} \sum_{k=0}^{N-1} X^*[k]\, e^{-j2\pi kn/N}$$

But this is exactly a DFT!

$$x^*[n] = \frac{1}{N} \cdot \mathrm{DFT}\{X^*[k]\}$$

Key Identity for IDFT via FFT

Start from the IDFT:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \, e^{j2\pi kn/N}$$

Take complex conjugate:

$$x^*[n] = \frac{1}{N} \sum_{k=0}^{N-1} X^*[k] \, e^{-j2\pi kn/N}$$

But this is exactly a DFT!

$$x^*[n] = \frac{1}{N} \cdot \mathrm{DFT}\{X^*[k]\}$$

Taking conjugate again:

$$x[n] = \frac{1}{N} \left(\mathrm{DFT}\{X^*[k]\}\right)^*$$

IDFT via FFT (Final Formula)

IDFT using FFT

$$x[n] = \frac{1}{N} \left(\text{FFT}\{X^*[k]\}\right)^*$$

Algorithm:

1. Take complex conjugate of $X[k]$
2. Compute FFT
3. Take complex conjugate of result
4. Divide by N

IDFT can be computed using the same FFT algorithm.

Why This Matters

Practical impact:

- No need for a separate IDFT algorithm
- FFT libraries implement both FFT and IFFT efficiently

Computational complexity:

- DFT: $\mathcal{O}(N^2)$
- FFT / IFFT: $\mathcal{O}(N \log N)$

In practice:

- `fft()` $\rightarrow$ DFT
- `ifft()` $\rightarrow$ IDFT (using same idea internally)

FFT is the core engine for both forward and inverse transforms.

DFT as Analysis on a Frequency Grid

The DFT does *not* evaluate all frequencies. It evaluates only the N frequencies

$$\omega_k = \frac{2\pi k}{N}, \qquad k = 0, \dots, N-1$$

or, in Hz,

$$f_k = \frac{kf_s}{N}$$

Important consequence:

- The spectrum is sampled on a finite grid of frequency bins
- Bin spacing is

$$\Delta f = \frac{f_s}{N}$$

- A sinusoid exactly on a bin is represented very cleanly
- A sinusoid between bins spreads energy across multiple bins

This is the beginning of the answer to:

"Why does my FFT not show a perfect spike?"

Why My FFT Looks "Wrong"

We often expect:

$$\text{pure tone} \implies \text{one FFT spike}$$

But in practice, the FFT often shows:

- energy spread across several bins
- peak location slightly offset from the true frequency
- amplitude smaller than expected
- nearby weak tones hidden by a strong one

Usually the FFT is not wrong. The issue is that our *measurement setup* is limited:

1. we observe only a finite-length record
2. we analyze only discrete frequency bins
3. the tone may not lie exactly on that frequency grid

Finite data + discrete frequency grid = leakage and limited resolution

Reason 1: We Never Observe an Infinite Signal

A perfect sinusoid is conceptually infinite in time, but in practice we record only

$$n = 0, 1, \dots, N-1$$

So the measured signal is really

$$x_{\text{obs}}[n] = x[n]\, w_R[n]$$

where $w_R[n]$ is a rectangular window:

$$w_R[n] = \begin{cases} 1, & 0 \leqslant n \leqslant N-1 \\ 0, & \text{otherwise} \end{cases}$$

Meaning:

- We are cutting out a finite segment of the signal
- This time truncation changes the spectrum
- Even a single pure tone no longer looks ideally concentrated

The FFT is applied to the truncated signal, not the ideal infinite one.

Reason 2: Time Truncation Causes Spectral Leakage

Multiplication in time corresponds to convolution in frequency:

$$x_{\text{obs}}[n] = x[n]\,w_R[n] \quad \Longrightarrow \quad X_{\text{obs}}(e^{j\omega}) = X(e^{j\omega}) * W_R(e^{j\omega})$$

For an ideal sinusoid:

- the ideal spectrum is concentrated at one frequency

After truncation by the rectangular window:

- that energy is convolved with the window spectrum
- energy spreads into neighboring frequencies

This spreading is called spectral leakage.

Intuition

Cutting a signal abruptly creates discontinuities at the edges.
Discontinuities require many frequency components to represent.

Reason 3: The Tone May Fall Between DFT Bins

The DFT checks only frequencies

$$f_k = \frac{kf_s}{N}$$

Suppose the true sinusoid frequency is f_0.

Case A: bin-centered tone

$$f_0 = f_k \quad \text{for some } k$$

Then the FFT looks very clean.

Case B: off-bin tone

$$f_0 \neq f_k \quad \text{for all } k$$

Then no single bin matches the sinusoid exactly, so the energy appears in multiple bins.

Key point

The FFT does not say the signal contains many tones.
It says the finite observation cannot represent that tone perfectly on the chosen frequency grid.

A Better Mental Model

The FFT is best viewed as answering the question:

"How much of each DFT basis frequency is present in this finite data record?"

It is **not** directly answering:

"What is the exact continuous-frequency content of the ideal infinite signal?"

So when the spectrum looks "wrong," ask:

- Did I use a short record?
- Is the tone off the bin grid?
- Am I seeing leakage?
- Is my bin spacing too coarse?

The FFT is correct for the finite data you gave it.

Preview: What We Will Study Next

The question "Why is my FFT wrong?" leads to two major ideas:

1. **Spectral leakage**
   - caused by finite-length observation
   - worsens when tones are not bin-centered
2. **Frequency resolution**
   - determined by bin spacing

$$\Delta f = \frac{f_s}{N}$$

   - improved by increasing the record length N

These explain why:

- a single tone may smear
- two close tones may merge
- weak tones may disappear near strong ones

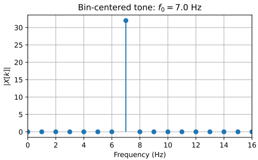
Why My FFT Looks "Wrong": Three Typical Cases

Case 1: Bin-centered tone

A sinusoid falls exactly on a DFT bin:

$$f_0 = k\frac{f_s}{N}$$

- energy concentrated at one bin
- very little spreading
- "clean" FFT peak

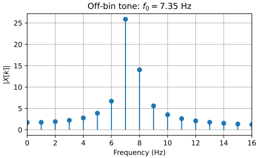


Case 2: Off-bin tone

A sinusoid falls between DFT bins:

$$f_0 \neq k\frac{f_s}{N}$$

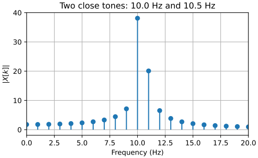
- energy spreads to nearby bins
- peak looks wider
- this is spectral leakage



Case 3: Two close tones

Two nearby frequencies are present.

- short record: peaks merge
- longer record: peaks separate
- resolution depends on N



Same FFT tool, different signal-record conditions.

What Changed Across These Cases?

- In Cases 1 and 2, the signal contains only *one* sinusoid
- The difference is whether the sinusoid aligns with the DFT grid
- In Case 3, the issue is not only leakage, but also **frequency resolution**

Main lessons:

1. Finite data causes spectral leakage
2. The DFT uses a discrete frequency grid
3. Two close tones require a sufficiently long record to be resolved

$$\Delta f = \frac{f_s}{N}$$

Larger $N \Rightarrow$ finer frequency grid $\Rightarrow$ better resolution.

What is Frequency Resolution?

Question: When can we distinguish two nearby frequencies?

Suppose the signal contains:

$$x[n] = \sin(2\pi f_1 n/f_s) + \sin(2\pi f_2 n/f_s)$$

Goal:

- Can we see *two distinct peaks* in the FFT?
- Or do they merge into one?

Definition:

Frequency Resolution

The ability to distinguish two closely spaced frequencies in the spectrum.

Resolution is not about computation — it is about available data.

Time Duration vs Frequency Resolution

Let the observation duration be:

$$T = \frac{N}{f_s}$$

Then:

$$\Delta f = \frac{1}{T}$$

Key insight:

- Longer observation time $\rightarrow$ better frequency resolution
- Short signals $\rightarrow$ poor resolution

Intuition

To estimate frequency accurately, we need to observe many cycles.

Intuition: Why Longer Signals Help

Short record:

- few cycles observed
- hard to distinguish frequencies
- peaks appear wide and overlapping

Long record:

- many cycles observed
- frequency estimate becomes sharper
- peaks become narrow and separable

Mental picture

Short signal $\rightarrow$ blurry spectrum
Long signal $\rightarrow$ sharp spectrum

When Resolution is Not Enough

Suppose:

$$f_1 = 10 \text{ Hz}, \quad f_2 = 10.5 \text{ Hz}$$

If:

$$\Delta f = 1 \text{ Hz}$$

Then:

- both tones fall into nearly the same bin region
- FFT shows one broadened peak
- we incorrectly conclude: "only one frequency exists"

Lack of resolution can hide real signal components.

Question: When are two frequencies resolvable?

Rayleigh Criterion (informal)

Two sinusoids are resolvable if their frequency separation is on the order of the main-lobe width.

For a rectangular window:

$$\text{minimum separation} \approx \Delta f = \frac{f_s}{N}$$

Meaning:

- If $|f_1 - f_2| < \Delta f \rightarrow$ peaks merge
- If $|f_1 - f_2| \gtrsim \Delta f \rightarrow$ peaks can be separated

Rayleigh limit defines the boundary between “merged” and “resolved”.

Important: Resolution vs Grid Spacing

Common misconception:

- "Resolution = bin spacing Δf"

More accurate view:

- Δf defines sampling of the spectrum
- Resolution depends on:
  - signal length
  - window shape (next lecture!)

Example:

- Zero-padding $\rightarrow$ smoother plot (more points)

Important: Resolution vs Grid Spacing

Common misconception:

- "Resolution = bin spacing Δf"

More accurate view:

- Δf defines sampling of the spectrum
- Resolution depends on:
  - signal length
  - window shape (next lecture!)

Example:

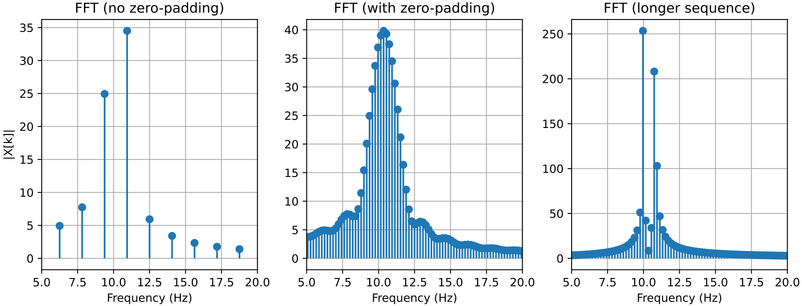
- Zero-padding $\rightarrow$ smoother plot (more points)
- But does *not* improve true resolution

Resolution is about information, not interpolation.

Zero-padding Example

$$x[n] = \sin(2\pi \frac{f_1 n}{f_s}) + \sin(2\pi \frac{f_2 n}{f_s})$$

with $f_1 = 10, f_2 = 10.8$, and $f_s = 100$.



Why Frequency Resolution Matters

In practice:

- Detecting closely spaced tones
- Identifying harmonics
- Audio analysis (notes close in pitch)
- Radar / sonar target separation

Problems caused by poor resolution:

- two signals appear as one
- incorrect frequency estimates
- weak signals hidden by strong ones

How to improve resolution:

- increase N (longer recording)
- choose appropriate window (next lecture!)

Better resolution = better insight into the signal.

What Limits Resolution in Practice?

We now understand:

- finite data $\rightarrow$ spectral leakage
- limited duration $\rightarrow$ limited resolution

New question:

Can we reduce leakage without losing too much resolution?

Answer: use better windows.

This is the topic of the next lecture.

What is Dynamic Range?

Question: Can we detect a weak signal next to a strong one?

Dynamic Range

The ability to distinguish small spectral components in the presence of large ones.

Example:

- strong tone at 10 Hz
- weak tone at 10.5 Hz

Can we see the weak tone?

Dynamic range determines what we can detect, not just what exists.

Why Dynamic Range is Limited

Main reason: spectral leakage

A strong tone spreads energy into nearby frequencies:

- creates *side lobes*
- raises the "floor" of the spectrum

Effect:

- weak tones can be masked
- they may become invisible in the FFT

Key idea

It is not enough to have resolution — we also need low leakage.

Dynamic Range Example

Signal:

$$x[n] = \sin(2\pi f_1 n) + 0.01 \sin(2\pi f_2 n)$$

Observations:

- strong tone dominates the spectrum
- leakage creates a wide “skirt”
- weak tone may disappear inside it

Result:

- both tones exist
- but FFT may show only one

Detection is limited by dynamic range.

Dynamic Range (Formal Definition)

Definition

Dynamic range (DR) is the ratio between the largest and smallest detectable spectral components.

In amplitude:

$$\mathrm{DR} = \frac{A_{\mathrm{max}}}{A_{\mathrm{min}}}$$

In decibels (dB):

$$\mathrm{DR}_{\mathrm{dB}} = 20 \log_{10}\left(\frac{A_{\mathrm{max}}}{A_{\mathrm{min}}}\right)$$

Interpretation:

- Large DR $\rightarrow$ can detect very weak signals near strong ones
- Small DR $\rightarrow$ weak signals are masked

Dynamic range is about detectability, not resolution.

How Do We Estimate $A_{\min}$?

In practice:

- compute FFT magnitude
- identify the spectral floor

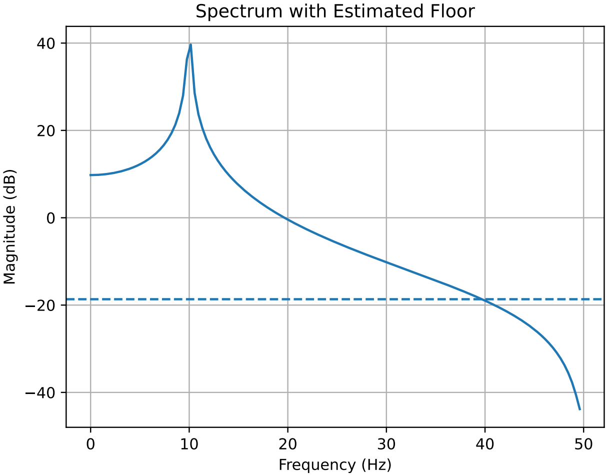
$$A_{\min} \approx \text{floor level of } |X[k]|$$

Dynamic range:

$$\mathrm{DR}_{\mathrm{dB}} \approx \text{peak} - \text{floor}$$

Detectability is limited by the spectral floor.

DR Estimation Example



Resolution vs Dynamic Range

Resolution:

- can we separate close frequencies?

Dynamic range:

- can we see weak signals near strong ones?

Important distinction

You may have enough resolution, but still miss a weak tone due to leakage.

Good spectral analysis requires both resolution and dynamic range.

How Do We Improve Dynamic Range?

We have seen:

- leakage limits dynamic range
- rectangular window has strong side lobes

Idea:

Use better windows to reduce leakage

Windowing trades resolution for improved dynamic range.

This is the focus of the next lecture.

Does Increasing N Improve Dynamic Range?

Important question:
Does a longer signal improve DR?

Answer: Not directly.

What increasing N does:

- improves frequency resolution
- narrows the main lobe

What it does NOT do:

- does not reduce side-lobe levels (in dB)
- does not fundamentally improve leakage floor

Conclusion:

- Resolution $\uparrow$ with N
- Dynamic range $\approx$ determined by window

N controls resolution, window controls dynamic range.

Subtle Point: N Helps Indirectly

Although N does not change side-lobe levels:

It still helps in practice:

- narrower main lobe
- leakage energy spreads over a smaller bandwidth

Effect:

- nearby tones may become easier to detect
- but weak signals far below side-lobes remain hidden

Takeaway

Increasing N improves separation, but not the leakage floor level.

Resolution vs Dynamic Range (Final Picture)

Frequency Resolution

$$\Delta f = \frac{f_s}{N}$$

- controlled by N
- determines ability to separate close tones

Dynamic Range

$$\mathrm{DR}_{\mathrm{dB}} \approx \text{side-lobe suppression}$$

- controlled by window choice
- determines ability to detect weak tones

Good spectral analysis requires BOTH resolution and dynamic range.

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