Signal & Systems

Lecture 7: Discrete Fourier Transform

Konstantinos Chatzilygeroudis - costashatz@upatras.gr

Department of Electrical and Computer Engineering
University of Patras

Template made by Panagiotis Papagiannopoulos

Why do only N frequency samples matter?

Assume we observe only a finite-length discrete-time signal:

$$x[n], \qquad n = 0, 1, \ldots, N-1$$

Its DTFT is still a continuous function of frequency:

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n] e^{-j\omega n}$$

Why do only N frequency samples matter?

Assume we observe only a finite-length discrete-time signal:

$$x[n], \qquad n = 0, 1, \ldots, N-1$$

Its DTFT is still a continuous function of frequency:

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n] e^{-j\omega n}$$

But: the signal has only N samples $\Rightarrow$ only N degrees of freedom.

So although $X(e^{j\omega})$ exists for all ω, we cannot obtain infinitely many *independent* spectral quantities from only N numbers.

Key idea

A length-N signal can be completely described by N properly chosen frequency samples.

Why do only N frequency samples matter? (2)

Key idea

A length-N signal can be completely described by N properly chosen frequency samples.

This is the motivation for evaluating the spectrum on the grid

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

Why the grid $\omega_k = \frac{2\pi k}{N}$?

Take the DTFT and sample it at

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

$$X[k] \triangleq X(e^{j\omega_k}) = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

The corresponding complex exponentials are $e^{j\frac{2\pi}{N}kn}$ and these are special because, over $n = 0, \ldots, N-1$, they form an orthogonal set.

$$\sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(k-\ell)n} = \begin{cases} N, & k = \ell \\ 0, & k \neq \ell \end{cases}$$

Consequence

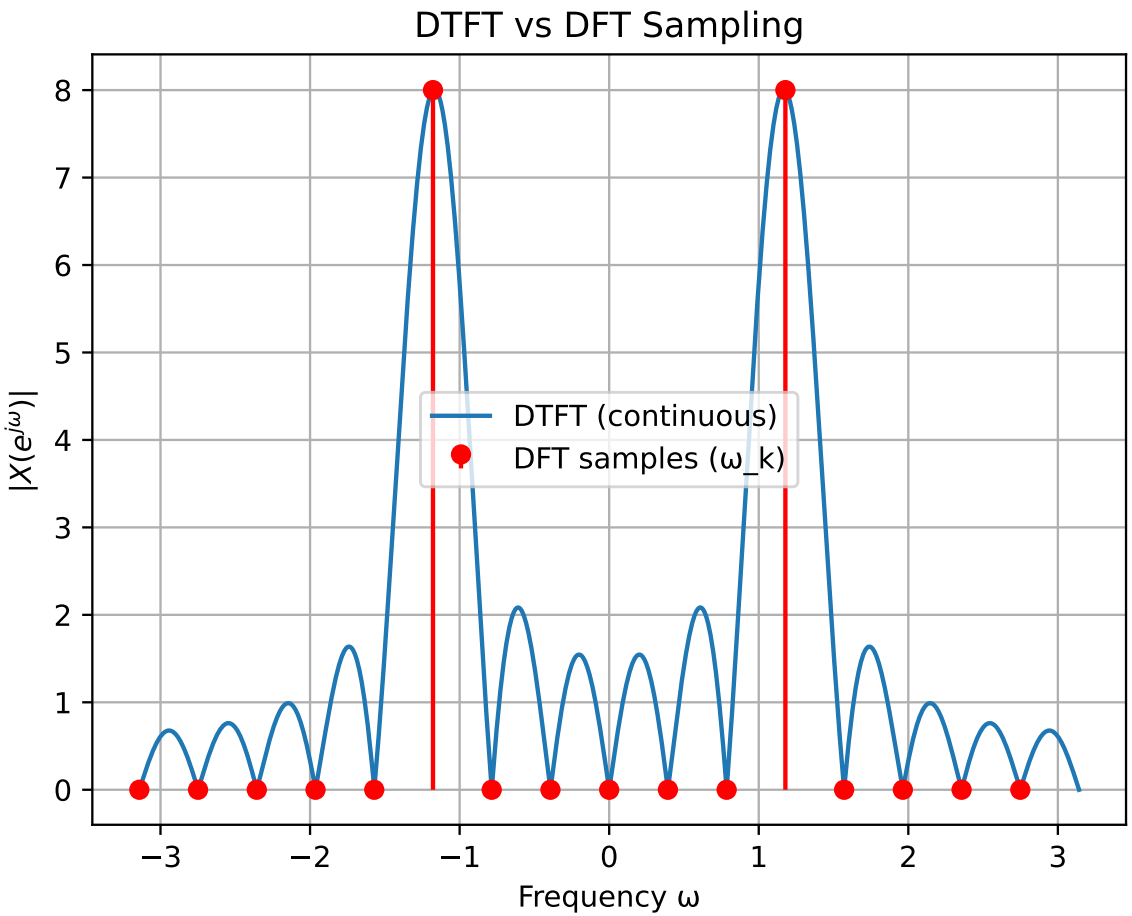
These N frequencies are exactly the independent frequency directions that a length-N signal can be projected onto.

Visualization

$$x[n] = \sin(2\pi 3 \frac{n}{N})$$

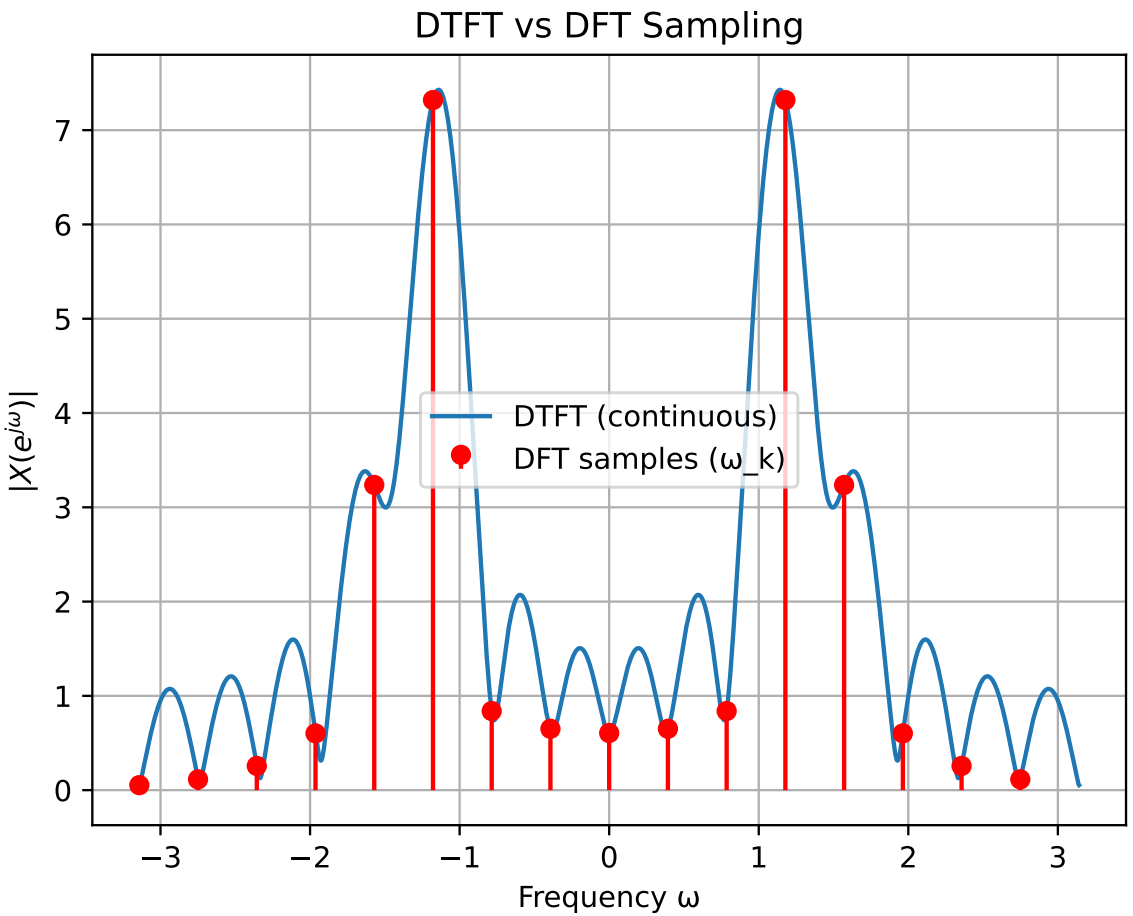
Visualization

$$x[n] = \sin(2\pi 3 \frac{n}{N})$$



Visualization (2)

$$x[n] = \sin(2\pi 3\frac{n}{N}) + \frac{1}{2}\sin(2\pi 3.7\frac{n}{N})$$



From DTFT to DFT: Sampling in Frequency

Recall the DTFT of a finite-length signal $x[n],\ n = 0, \ldots, N-1$:

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n] e^{-j\omega n}$$

Although $X(e^{j\omega})$ is continuous in ω, we only need N independent frequency samples.

We choose the frequency grid:

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

Key idea

The Discrete Fourier Transform (DFT) is obtained by sampling the DTFT at these frequencies.

Definition of the DFT

We define the DFT as:

$$X[k] \triangleq X(e^{j\omega_k}), \quad \omega_k = \frac{2\pi k}{N}$$

Substituting into the DTFT expression:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

Definition

The sequence $\{X[k]\}_{k=0}^{N-1}$ is called the Discrete Fourier Transform (DFT) of $x[n]$.

- $x[n]$: time-domain samples (finite length N)
- $X[k]$: frequency samples on a discrete grid

Interpretation of the DFT Samples

Each DFT coefficient corresponds to a specific frequency:

$$\omega_k = \frac{2\pi k}{N}$$

- $X[k]$ measures the presence of frequency ω_k
- Each bin corresponds to a complex exponential: $e^{j\frac{2\pi}{N}kn}$

Important

The DFT does not evaluate all frequencies — only the N frequencies that can be uniquely represented given N samples.

$$\Delta\omega = \frac{2\pi}{N}$$

- Frequency resolution depends on N

Concrete Example: Setup

Let $N = 4$ and:

$$x[n] = [1,\ 0,\ -1,\ 0]$$

We compute the DFT using:

$$X[k] = \sum_{n=0}^{3} x[n] e^{-j\frac{2\pi}{4}kn}$$

Key values:

$$e^{-j\frac{2\pi}{4}} = e^{-j\frac{\pi}{2}} = -j$$

$$e^{-j\pi} = -1, \quad e^{-j\frac{3\pi}{2}} = j$$

We will compute each $X[k]$ as follows:

- Fix k
- Loop over n
- Accumulate terms

Compute $X[0]$

For $k = 0$:

$$X[0] = \sum_{n=0}^{3} x[n] e^{-j\frac{2\pi}{4} \cdot 0 \cdot n}$$

Since $e^0 = 1$:

$$X[0] = x[0] + x[1] + x[2] + x[3]$$

Step-by-step accumulation:

$$\begin{aligned} X[0] &= 1 \\ &\quad + 0 \\ &\quad + (-1) \\ &\quad + 0 \end{aligned}$$

$$X[0] = 0$$

Compute $X[1]$

For $k = 1$:

$$X[1] = \sum_{n=0}^{3} x[n] e^{-j\frac{2\pi}{4}n}$$

Compute each term:

$$\begin{aligned} n = 0: &\quad 1 \cdot 1 = 1 \\ n = 1: &\quad 0 \cdot (-j) = 0 \\ n = 2: &\quad (-1) \cdot (-1) = 1 \\ n = 3: &\quad 0 \cdot j = 0 \end{aligned}$$

Accumulating:

$$X[1] = 1 + 0 + 1 + 0 = 2$$

Compute $X[2]$

For $k = 2$:

$$X[2] = \sum_{n=0}^{3} x[n] e^{-j\pi n}$$

$$e^{-j\pi n} = [1,\ -1,\ 1,\ -1]$$

$$\begin{aligned} X[2] = 1 \cdot 1 \\ + 0 \cdot (-1) \\ + (-1) \cdot 1 \\ + 0 \cdot (-1) \end{aligned}$$

$$X[2] = 1 - 1 = 0$$

Compute $X[3]$

For $k = 3$:

$$X[3] = \sum_{n=0}^{3} x[n] e^{-j\frac{2\pi}{4}3n}$$

Exponential values:

$$[1,\ j,\ -1,\ -j]$$

$$\begin{aligned} X[3] &= 1 \cdot 1 \\ &\quad + 0 \cdot j \\ &\quad + (-1) \cdot (-1) \\ &\quad + 0 \cdot (-j) \end{aligned}$$

$$X[3] = 1 + 1 = 2$$

Thus:

$$X[k] = [0,\ 2,\ 0,\ 2]$$

DFT as Projection onto Basis Functions

Recall the DFT definition:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

We can interpret this as an inner product:

$$X[k] = \langle x[n],\ e^{j\frac{2\pi}{N}kn} \rangle$$

Interpretation

Each DFT coefficient $X[k]$ measures how much of the frequency

$$\omega_k = \frac{2\pi k}{N}$$

is present in the signal.

DFT as Projection onto Basis Functions (2)

Interpretation

Each DFT coefficient $X[k]$ measures how much of the frequency

$$\omega_k = \frac{2\pi k}{N}$$

is present in the signal.

- The complex exponentials act as basis functions
- The DFT computes projections onto these basis functions

The DFT Basis Functions

The basis functions are:

$$\phi_k[n] = e^{j\frac{2\pi}{N}kn}, \qquad k = 0, \ldots, N-1$$

These functions have a key property:

$$\sum_{n=0}^{N-1} \phi_k[n]\, \phi_\ell^*[n] = \begin{cases} N, & k = \ell \\ 0, & k \neq \ell \end{cases}$$

Orthogonality

The basis functions are orthogonal over $n = 0, \ldots, N-1$.

- Each frequency is independent
- No overlap between different bins

Reconstruction from DFT Coefficients

Since the basis functions are orthogonal, we can reconstruct the signal (**Inverse DFT**):

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}kn}$$

Key idea

The DFT coefficients $X[k]$ are the coordinates of $x[n]$ in the frequency basis.

- Time-domain signal = sum of sinusoids
- Each sinusoid weighted by $X[k]$

DFT = change of basis (time → frequency)

IDFT Example

$$x[n] = \frac{1}{4} \sum_{k=0}^{3} X[k] e^{j \frac{2\pi}{4} kn}$$

Using:

$$X[k] = [0,\ 2,\ 0,\ 2]$$

Example for $n = 0$:

$$x[0] = \frac{1}{4}(0 + 2 + 0 + 2) = 1$$

Similarly:

$$x[1] = 0, \quad x[2] = -1, \quad x[3] = 0$$

Conclusion

The inverse DFT exactly reconstructs the signal.

DFT as a Matrix Operation

Recall the DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

We can write all equations together:

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W & W^2 & \cdots & W^{N-1} \\ 1 & W^2 & W^4 & \cdots & W^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W^{N-1} & W^{2(N-1)} & \cdots & W^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$$

where:

$$W = e^{-j\frac{2\pi}{N}}$$

Compact Matrix Form

We write:

$$\mathbf{X} = \mathbf{W}\mathbf{x}$$

where:

- $\mathbf{x} \in \mathbb{C}^N$: time-domain vector
- $\mathbf{X} \in \mathbb{C}^N$: frequency-domain vector
- $\mathbf{W}$: DFT matrix

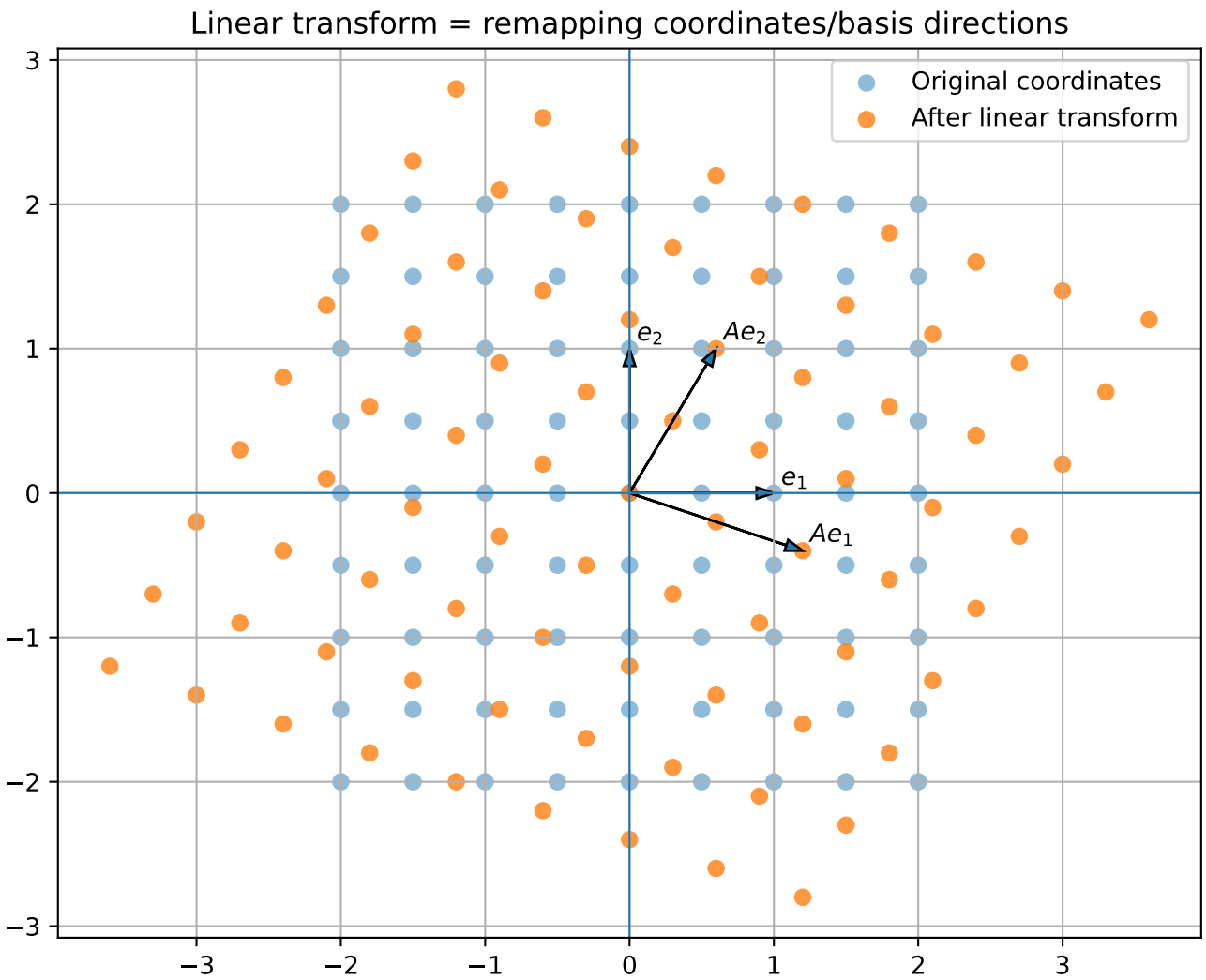
Entries of $\mathbf{W}$:

$$W_{k,n} = e^{-j\frac{2\pi}{N}kn}$$

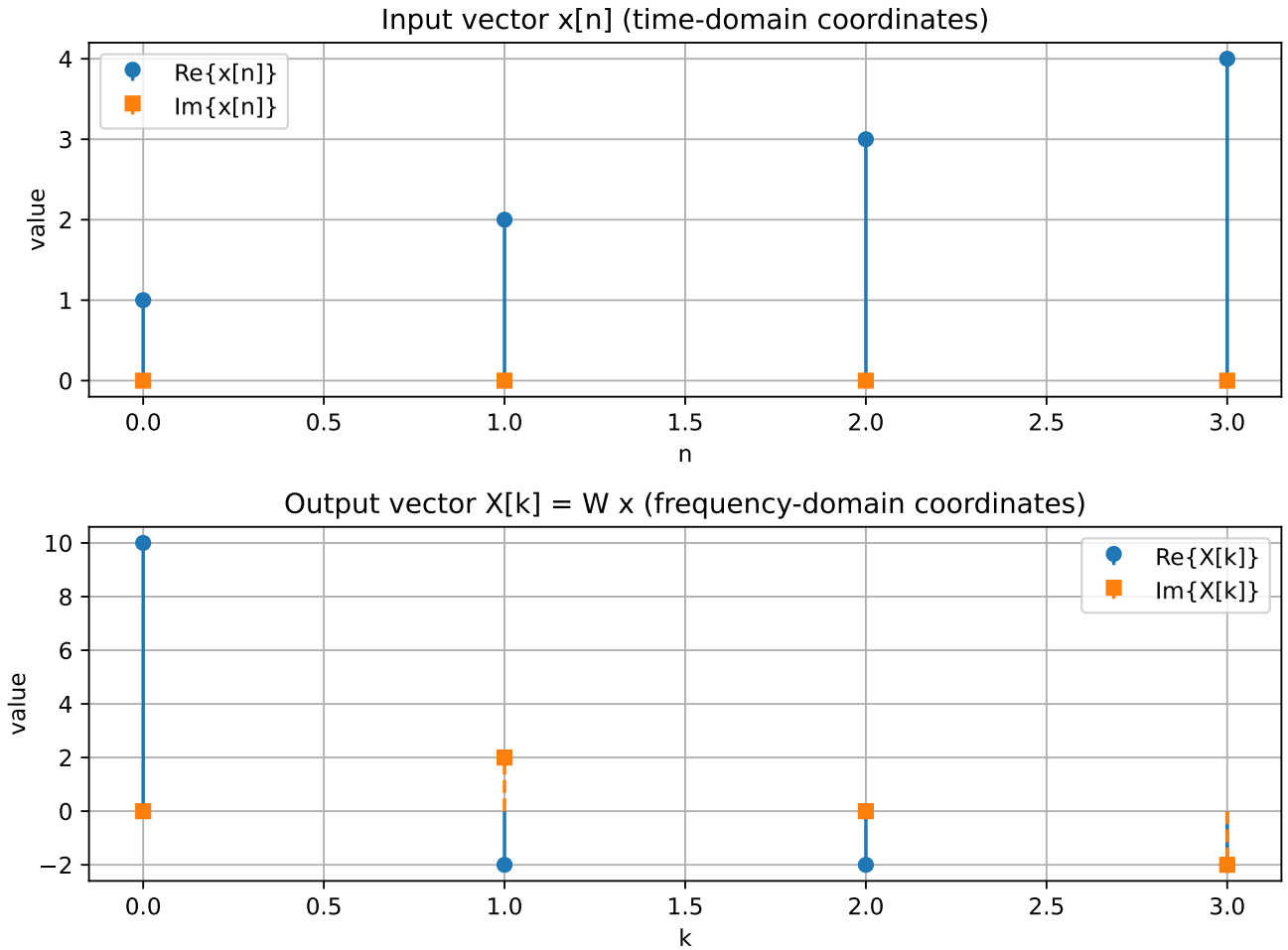
Interpretation

The DFT is a linear transformation from time domain to frequency domain.

DFT as a Linear Mapping



DFT as a Linear Mapping (2)



Computational Cost of the DFT

To compute each $X[k]$:

- Requires N multiplications
- There are N values of k

$$\text{Total cost} \sim N^2$$

Problem

For large N, this becomes very expensive.

- $N = 10^3 \Rightarrow 10^6$ operations
- $N = 10^6 \Rightarrow 10^{12}$ operations

Can we compute the same result faster?

Key Idea Behind the Fast Fourier Transform

The DFT matrix has a lot of structure:

$$W_{k,n} = e^{-j\frac{2\pi}{N}kn}$$

- Repeating patterns
- Symmetries
- Periodicity

Core idea

Exploit this structure to avoid redundant computations.

Instead of computing everything from scratch:

- Break the problem into smaller DFTs
- Reuse intermediate results

Cost reduces from $O(N^2)$ to $O(N \log N)$

FFT Intuition: Divide and Conquer

Split the signal into:

- Even-indexed samples
- Odd-indexed samples

This allows us to write:

$$X[k] = \underbrace{\sum x[2m] e^{-j\frac{2\pi}{N}k(2m)}}_{\text{even part}} + e^{-j\frac{2\pi}{N}k} \underbrace{\sum x[2m+1] e^{-j\frac{2\pi}{N}k(2m)}}_{\text{odd part}}$$

Insight

Each term is a smaller DFT of size $N/2$.

Repeat recursively $\Rightarrow$ Fast Fourier Transform (FFT)

Splitting the DFT: Even and Odd Samples

Start from the DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

Split the sum into even and odd indices:

$$X[k] = \sum_{m=0}^{\frac{N}{2}-1} x[2m] e^{-j\frac{2\pi}{N}k(2m)} + \sum_{m=0}^{\frac{N}{2}-1} x[2m+1] e^{-j\frac{2\pi}{N}k(2m+1)}$$

Factor the exponential in the second term:

$$X[k] = \sum_{m=0}^{\frac{N}{2}-1} x[2m] e^{-j\frac{2\pi}{N}k(2m)} + e^{-j\frac{2\pi}{N}k} \sum_{m=0}^{\frac{N}{2}-1} x[2m+1] e^{-j\frac{2\pi}{N}k(2m)}$$

Defining Smaller DFTs

Define:

$$E[k] \triangleq \sum_{m=0}^{\frac{N}{2}-1} x[2m] e^{-j\frac{2\pi}{N}k(2m)}$$

$$O[k] \triangleq \sum_{m=0}^{\frac{N}{2}-1} x[2m+1] e^{-j\frac{2\pi}{N}k(2m)}$$

Then:

$$X[k] = E[k] + e^{-j\frac{2\pi}{N}k} O[k]$$

Key insight

The DFT of size N is expressed using two smaller DFT-like computations.

Why This Leads to the FFT

Observe that:

$$E[k],\ O[k] \text{ are DFTs of length } \frac{N}{2}$$

- Even-indexed signal: $x[0], x[2], x[4], \ldots$
- Odd-indexed signal: $x[1], x[3], x[5], \ldots$

Divide and Conquer

$$\text{DFT}(N) \rightarrow 2 \times \text{DFT}(N/2)$$

- Repeat recursively
- Reuse computations

This is the core idea behind the FFT

Recursive FFT

```
1  Function FFT(x[n]):
       Input: Sequence x[n], n = 0, ..., N − 1 (with N a power of 2)
       Output: DFT X[k], k = 0, ..., N − 1
2      if N = 1 then
3          return x
       // Split into even and odd indices
4      x_even[m] ← x[2m]
5      x_odd[m] ← x[2m + 1]
       // Recursive FFT calls
6      E[k] ← FFT(x_even)
7      O[k] ← FFT(x_odd)
8      for k ← 0 to N/2 − 1 do
9          W ← e^{−j2πk/N}
10         X[k] ← E[k] + W O[k]
11         X[k + N/2] ← E[k] − W O[k]
12     return X
```

Complexity: $O(N \log_2 N)$ operations ($\log_2 N$ stages, $N/2$ butterflies per stage).

FFT Example: Setup

Let:

$$x[n] = [1,\ 0,\ -1,\ 0], \quad N = 4$$

We already computed the DFT:

$$X[k] = [0,\ 2,\ 0,\ 2]$$

Now we compute the same result using the FFT idea:

- Split into even and odd samples
- Compute smaller DFTs
- Combine results

Step 1: Split Even and Odd Samples

Even-indexed samples:

$$x_e[m] = [x[0], x[2]] = [1,\ -1]$$

Odd-indexed samples:

$$x_o[m] = [x[1], x[3]] = [0,\ 0]$$

We now compute two DFTs of size $N/2 = 2$:

$$E[k] = \text{DFT of } x_e[m] \quad O[k] = \text{DFT of } x_o[m]$$

Step 2: Compute $E[k]$ (DFT of Even Part)

$$x_e[m] = [1,\ -1]$$

DFT of length 2:

$$E[0] = 1 + (-1) = 0$$
$$E[1] = 1 - (-1) = 2$$

$$E[k] = [0,\ 2]$$

Step 3: Compute $O[k]$ (DFT of Odd Part)

$$x_o[m] = [0,\ 0]$$

$$O[0] = 0 + 0 = 0$$

$$O[1] = 0 - 0 = 0$$

$$O[k] = [0,\ 0]$$

Step 4: Combine Results

Use:

$$X[k] = E[k] + W_N^k O[k]$$

$$X[k + N/2] = E[k] - W_N^k O[k]$$

where:

$$W_N = e^{-j\frac{2\pi}{N}} = e^{-j\frac{2\pi}{4}}$$

Compute for $k = 0, 1$:

For $k = 0$:

$$X[0] = E[0] + O[0] = 0$$

$$X[2] = E[0] - O[0] = 0$$

For $k = 1$:

$$X[1] = E[1] + W_4^1 O[1] = 2$$

$$X[3] = E[1] - W_4^1 O[1] = 2$$

FFT Result

$$X[k] = [0,\ 2,\ 0,\ 2]$$

Observation

Same result as the direct DFT computation.

- Direct DFT: compute everything from scratch
- FFT: reuse smaller DFTs

Less computation, same output

Another FFT Example

Let:

$$x[n] = [1,\ 2,\ 3,\ 4], \quad N = 4$$

We compute the DFT using the FFT approach:

- Split into even and odd samples
- Compute smaller DFTs
- Combine using twiddle factors

Step 1: Even and Odd Parts

Even-indexed samples:

$$x_e[m] = [x[0], x[2]] = [1,\ 3]$$

Odd-indexed samples:

$$x_o[m] = [x[1], x[3]] = [2,\ 4]$$

We compute:

$$E[k] = \text{DFT of } x_e[m], \quad O[k] = \text{DFT of } x_o[m]$$

Step 2: Compute $E[k]$

$$x_e[m] = [1,\ 3]$$

DFT (length 2):

$$E[0] = 1 + 3 = 4$$

$$E[1] = 1 - 3 = -2$$

$$E[k] = [4,\ -2]$$

Step 3: Compute $O[k]$

$$x_o[m] = [2,\ 4]$$

$$O[0] = 2 + 4 = 6$$

$$O[1] = 2 - 4 = -2$$

$$O[k] = [6,\ -2]$$

Step 4: Twiddle Factors

$$W_4 = e^{-j\frac{2\pi}{4}} = e^{-j\frac{\pi}{2}} = -j$$

So:

$$W_4^0 = 1, \quad W_4^1 = -j$$

Combination formulas

$$X[k] = E[k] + W_4^k O[k]$$

$$X[k+2] = E[k] - W_4^k O[k]$$

Step 5: Combine

$$X[0] = E[0] + O[0] = 4 + 6 = 10$$

$$X[2] = E[0] - O[0] = 4 - 6 = -2$$

$$X[1] = E[1] + W_4^1 O[1] = -2 + (-j)(-2) = -2 + 2j$$

$$X[3] = E[1] - W_4^1 O[1] = -2 - 2j$$

FFT Result

$$X[k] = [10,\ -2 + 2j,\ -2,\ -2 - 2j]$$

What we see

- Both $E[k]$ and $O[k]$ contribute
- Twiddle factors introduce complex rotation

This is the full FFT structure in action

Frequency Bins: What does k mean?

Recall the DFT frequencies:

$$\omega_k = \frac{2\pi k}{N}, \quad k = 0, \ldots, N-1$$

In terms of physical frequency (Hz):

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

where f_s is the sampling frequency.

Interpretation

Each index k corresponds to a specific frequency f_k.

- $k = 0 \rightarrow$ DC (constant component)
- $k = 1 \rightarrow$ lowest nonzero frequency
- $k = N/2 \rightarrow$ highest frequency (Nyquist)

Frequency Resolution

Spacing between frequency bins:

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

Key idea

The frequency grid is uniform.

- Larger $N \rightarrow$ finer frequency resolution
- Smaller $N \rightarrow$ coarser grid

Resolution depends on signal length, not the FFT algorithm

What does $X[k]$ measure?

Recall:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

Interpretation

$X[k]$ measures how much of the frequency

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

is present in the signal.

- Magnitude $|X[k]| \rightarrow$ strength (amplitude)
- Phase $\angle X[k] \rightarrow$ phase of that component

When a Frequency Matches a Bin

Suppose:

$$x[n] = \sin\left(2\pi \frac{k_0}{N} n\right)$$

with $k_0 \in \mathbb{Z}$.

Then:

- Energy appears only at bin k_0
- Clean, sharp peak

Key idea

If the signal frequency matches the grid, the DFT is perfectly localized.

When the Frequency is Not on the Grid

Suppose:

$$x[n] = \sin\left(2\pi \frac{k_0 + \delta}{N} n\right)$$

with $k_0 \in \mathbb{Z}$, and $\delta \in \mathbb{R}$.

- Frequency does not match any bin
- Energy spreads across multiple bins

Observation

The DFT cannot represent arbitrary frequencies exactly.

This is spectral leakage!

Periodicity of the Spectrum

The DFT is periodic:

$$X[k+N] = X[k]$$

- Frequencies wrap around
- High-frequency bins correspond to negative frequencies

Example

$$k = N-1 \;\leftrightarrow\; f = -\frac{f_s}{N}$$

Summary: Interpreting Frequency Bins

- Each bin k corresponds to:

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

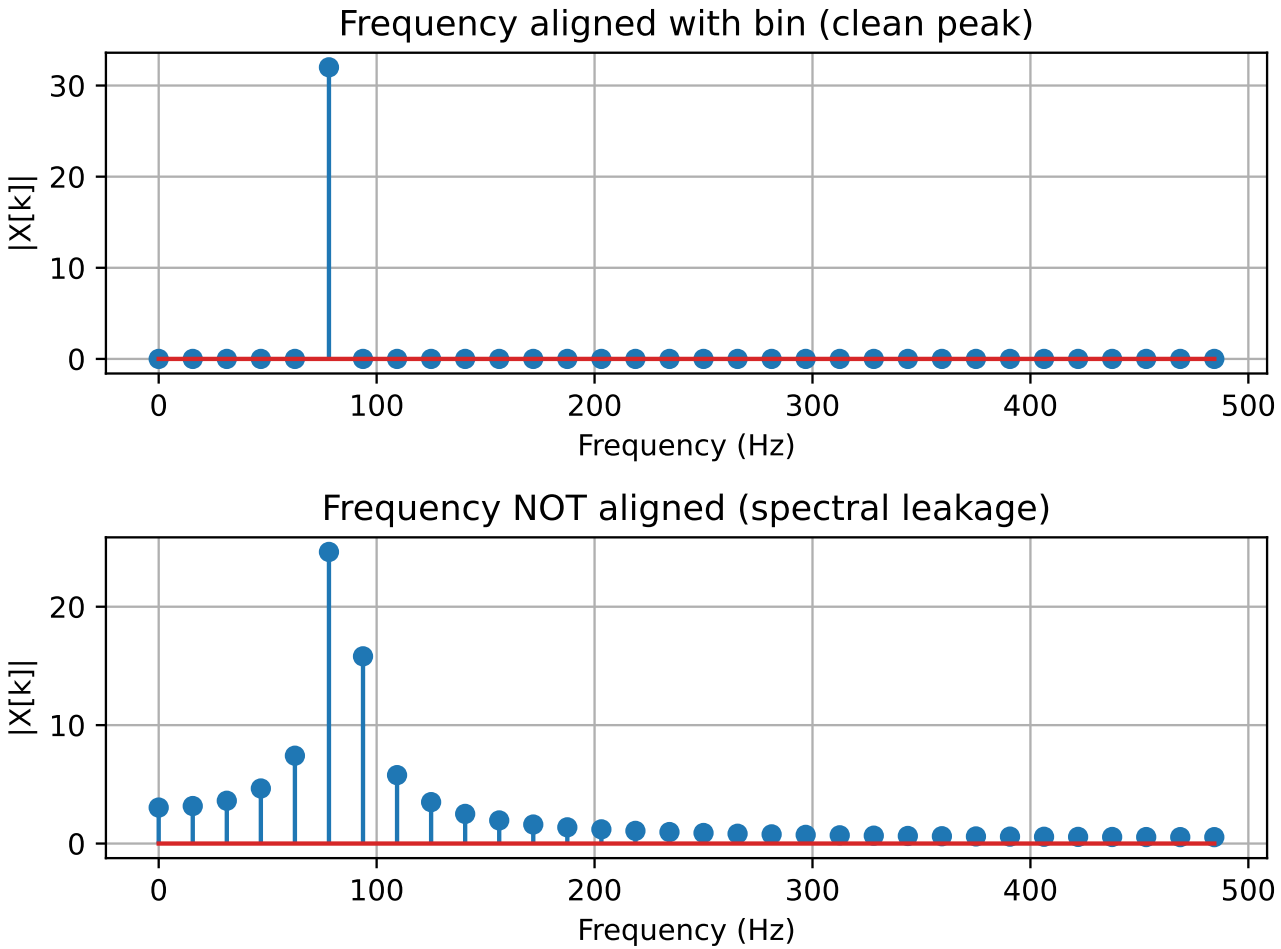
- Spacing:

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

- $X[k]$ measures amplitude and phase at that frequency
- Only grid frequencies are represented exactly

DFT = discrete sampling of frequency content

Visualization



Normalization in the DFT

Recall the DFT pair:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \quad , \quad x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}$$

Important

The scaling factor $\frac{1}{N}$ can be placed in different ways.

- There is no single universal convention
- Different fields and software use different normalizations

Convention 1: Standard (Analysis/Synthesis)

Forward DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

Inverse DFT:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}$$

Interpretation

- Forward transform: no scaling
- Inverse transform: divides by N

- Very common in signal processing

Convention 2: Symmetric Normalization

Forward DFT:

$$X[k] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

Inverse DFT:

$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}$$

Interpretation

- Energy is preserved between domains
- Transform becomes unitary

- Common in mathematics and theoretical analysis

Convention 3: Engineering / Software

Typical implementation:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}$$

In practice

- `fft()` $\rightarrow$ no scaling
- `ifft()` $\rightarrow$ includes $\frac{1}{N}$

- Most common in real-world applications

Why Normalization Matters

The choice of scaling affects:

- Amplitude of $X[k]$
- Energy interpretation
- Comparisons between signals

Common pitfall

A peak of magnitude 100 might mean:

- true amplitude 100 or
- true amplitude $100/N$

Always check the normalization before interpreting amplitudes

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