Digital Signal Processing

Markus Kuhn

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

http://www.cl.cam.ac.uk/teaching/1112/DSP/

Michaelmas 2011 – Part II

Signals

→ flow of information

→ measured quantity that varies with time (or position)

→ electrical signal received from a transducer
(microphone, thermometer, accelerometer, antenna, etc.)

→ electrical signal that controls a process

Continuous-time signals: voltage, current, temperature, speed, ...

Discrete-time signals: daily minimum/maximum temperature, lap intervals in races, sampled continuous signals, ...

Electronics (unlike optics) can only deal easily with time-dependent signals, therefore spatial signals, such as images, are typically first converted into a time signal with a scanning process (TV, fax, etc.).

Signal processing

Signals may have to be transformed in order to

- $\longrightarrow$ amplify or filter out embedded information
- $\longrightarrow$ detect patterns
- $\longrightarrow$ prepare the signal to survive a transmission channel
- $\longrightarrow$ prevent interference with other signals sharing a medium
- $\longrightarrow$ undo distortions contributed by a transmission channel
- $\longrightarrow$ compensate for sensor deficiencies
- $\longrightarrow$ find information encoded in a different domain

To do so, we also need

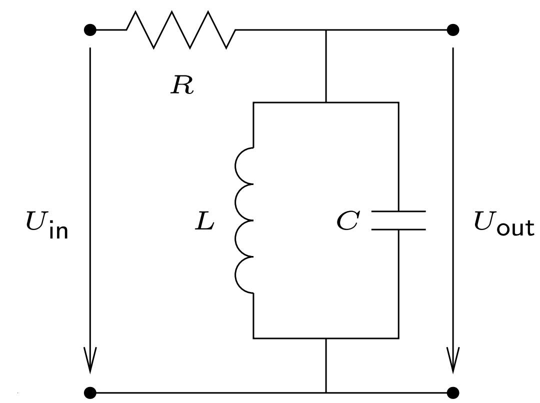
- $\longrightarrow$ methods to measure, characterise, model and simulate transmission channels
- $\longrightarrow$ mathematical tools that split common channels and transformations into easily manipulated building blocks

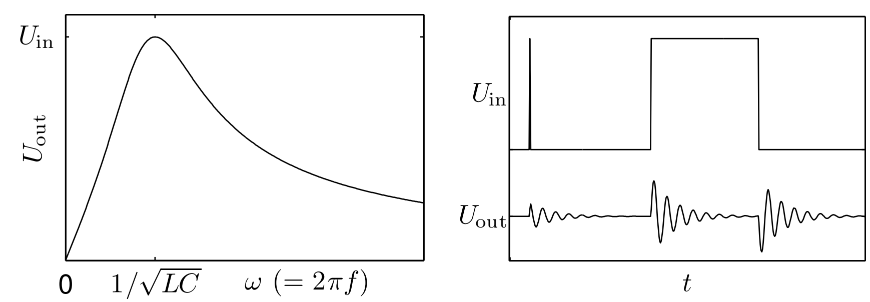
Analog electronics

Passive networks (resistors, capacitors, inductances, crystals, SAW filters), non-linear elements (diodes, . . .), (roughly) linear operational amplifiers

Advantages:

- passive networks are highly linear over a very large dynamic range and large bandwidths
- analog signal-processing circuits require little or no power
- analog circuits cause little additional interference





$$\frac{U_{\text{in}} - U_{\text{out}}}{R} = \frac{1}{L}\int_{-\infty}^{t} U_{\text{out}}\,\mathrm{d}\tau + C\frac{\mathrm{d}U_{\text{out}}}{\mathrm{d}t}$$

Digital signal processing

Analog/digital and digital/analog converter, CPU, DSP, ASIC, FPGA.

Advantages:

- → noise is easy to control after initial quantization
- → highly linear (within limited dynamic range)
- → complex algorithms fit into a single chip
- → flexibility, parameters can easily be varied in software
- → digital processing is insensitive to component tolerances, aging, environmental conditions, electromagnetic interference

But:

- → discrete-time processing artifacts (aliasing)
- → can require significantly more power (battery, cooling)
- → digital clock and switching cause interference

Typical DSP applications

- → communication systems

  modulation/demodulation, channel equalization, echo cancellation

- → consumer electronics

  perceptual coding of audio and video on DVDs, speech synthesis, speech recognition

- → music

  synthetic instruments, audio effects, noise reduction

- → medical diagnostics

  magnetic-resonance and ultrasonic imaging, computer tomography, ECG, EEG, MEG, AED, audiology

- → geophysics

  seismology, oil exploration

- → astronomy

  VLBI, speckle interferometry

- → experimental physics

  sensor-data evaluation

- → aviation

  radar, radio navigation

- → security

  steganography, digital watermarking, biometric identification, surveillance systems, signals intelligence, electronic warfare

- → engineering

  control systems, feature extraction for pattern recognition

Sequences and systems

A *discrete sequence* $\{x_n\}_{n=-\infty}^{\infty}$ is a sequence of numbers

$$\dots, x_{-2}, x_{-1}, x_0, x_1, x_2, \dots$$

where x_n denotes the n-th number in the sequence ($n \in \mathbb{Z}$). A discrete sequence maps integer numbers onto real (or complex) numbers.

We normally abbreviate $\{x_n\}_{n=-\infty}^{\infty}$ to $\{x_n\}$, or to $\{x_n\}_n$ if the running index is not obvious. The notation is not well standardized. Some authors write $x[n]$ instead of x_n, others $x(n)$.

Where a discrete sequence $\{x_n\}$ samples a continuous function $x(t)$ as

$$x_n = x(t_\mathrm{s} \cdot n) = x(n/f_\mathrm{s}),$$

we call t_s the *sampling period* and $f_\mathrm{s} = 1/t_\mathrm{s}$ the *sampling frequency*.

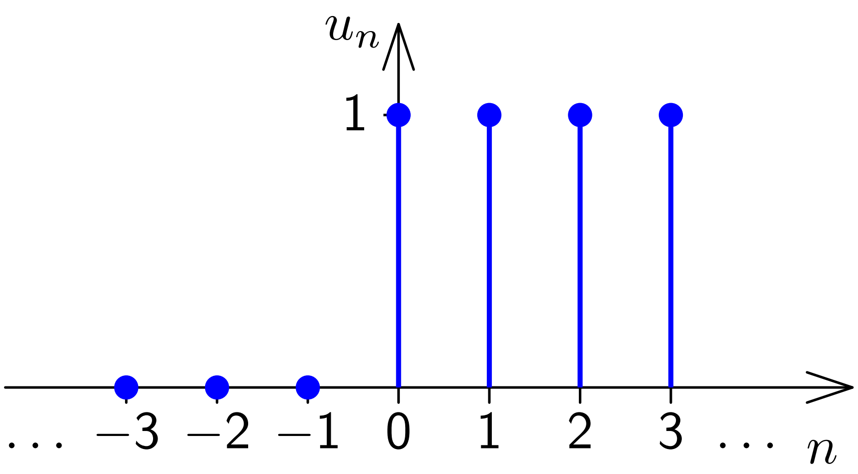
A *discrete system* T receives as input a sequence $\{x_n\}$ and transforms it into an output sequence $\{y_n\} = T\{x_n\}$:

$$\dots, x_2, x_1, x_0, x_{-1}, \dots \longrightarrow \boxed{\begin{array}{c}\text{discrete}\\ \text{system } T\end{array}} \longrightarrow \dots, y_2, y_1, y_0, y_{-1}, \dots$$

Some simple sequences

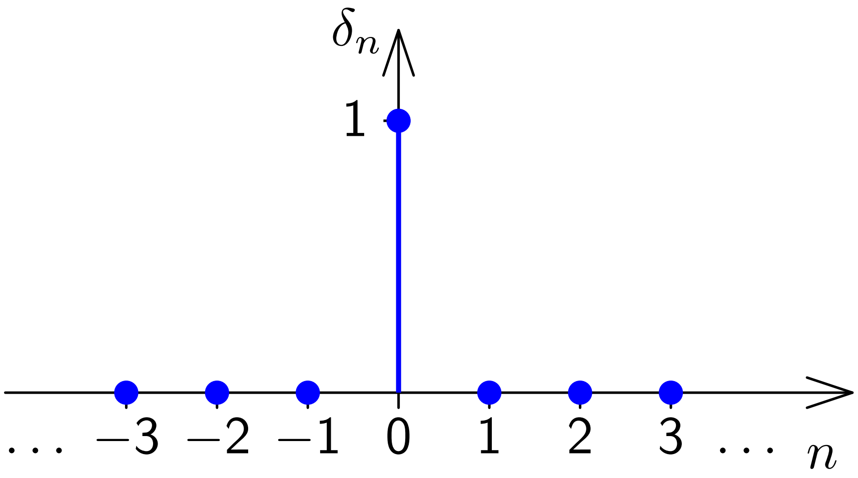
Unit-step sequence:

$$u_n = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$$



Impulse sequence:

$$\begin{aligned} \delta_n &= \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \\ &= u_n - u_{n-1} \end{aligned}$$



Properties of sequences

A sequence $\{x_n\}$ is

$$
\begin{aligned}
\textit{periodic} \quad &\Leftrightarrow \quad \exists k > 0 : \forall n \in \mathbb{Z} : x_n = x_{n+k} \\
\textit{absolutely summable} \quad &\Leftrightarrow \quad \sum_{n=-\infty}^{\infty} |x_n| < \infty \\
\textit{square summable} \quad &\Leftrightarrow \quad \underbrace{\sum_{n=-\infty}^{\infty} |x_n|^2}_{\text{"energy"}} < \infty \quad \Leftrightarrow \quad \text{"energy signal"}
\end{aligned}
$$

$$
0 < \underbrace{\lim_{k\to\infty} \frac{1}{1+2k} \sum_{n=-k}^{k} |x_n|^2}_{\text{"average power"}} < \infty \quad \Leftrightarrow \quad \text{"power signal"}
$$

This energy/power terminology reflects that if U is a voltage supplied to a load resistor R, then $P = UI = U^2/R$ is the power consumed, and $\int P(t)\,\mathrm{d}t$ the energy. It is used even if we drop physical units (e.g., volts) for simplicity in calculations.

Types of discrete systems

A *causal system* cannot look into the future:

$$y_n = f(x_n, x_{n-1}, x_{n-2}, \ldots)$$

A *memory-less system* depends only on the current input value:

$$y_n = f(x_n)$$

A *delay system* shifts a sequence in time:

$$y_n = x_{n-d}$$

T is a *time-invariant system* if for any d

$$\{y_n\} = T\{x_n\} \quad \Longleftrightarrow \quad \{y_{n-d}\} = T\{x_{n-d}\}.$$

T is a *linear system* if for any pair of sequences $\{x_n\}$ and $\{x'_n\}$

$$T\{a \cdot x_n + b \cdot x'_n\} = a \cdot T\{x_n\} + b \cdot T\{x'_n\}.$$

Examples:

The *accumulator system*

$$y_n = \sum_{k=-\infty}^{n} x_k$$

is a causal, linear, time-invariant system with memory, as are the *backward difference system*

$$y_n = x_n - x_{n-1},$$

the *M-point moving average system*

$$y_n = \frac{1}{M} \sum_{k=0}^{M-1} x_{n-k} = \frac{x_{n-M+1} + \cdots + x_{n-1} + x_n}{M}$$

and the *exponential averaging system*

$$y_n = \alpha \cdot x_n + (1 - \alpha) \cdot y_{n-1} = \alpha \sum_{k=0}^{\infty} (1 - \alpha)^k \cdot x_{n-k}.$$

Examples for time-invariant non-linear memory-less systems:

$$y_n = x_n^2, \quad y_n = \log_2 x_n, \quad y_n = \max\{\min\{\lfloor 256 x_n \rfloor, 255\}, 0\}$$

Examples for linear but not time-invariant systems:

$$\begin{aligned} y_n &= \begin{cases} x_n, & n \geq 0 \\ 0, & n < 0 \end{cases} = x_n \cdot u_n \\ y_n &= x_{\lfloor n/4 \rfloor} \\ y_n &= x_n \cdot \Re(\mathrm{e}^{\omega \mathrm{j} n}) \end{aligned}$$

Examples for linear time-invariant non-causal systems:

$$\begin{aligned} y_n &= \frac{1}{2}(x_{n-1} + x_{n+1}) \\ y_n &= \sum_{k=-9}^{9} x_{n+k} \cdot \frac{\sin(\pi k \omega)}{\pi k \omega} \cdot [0.5 + 0.5 \cdot \cos(\pi k/10)] \end{aligned}$$

Convolution

All linear time-invariant (LTI) systems can be represented in the form

$$y_n = \sum_{k=-\infty}^{\infty} a_k \cdot x_{n-k}$$

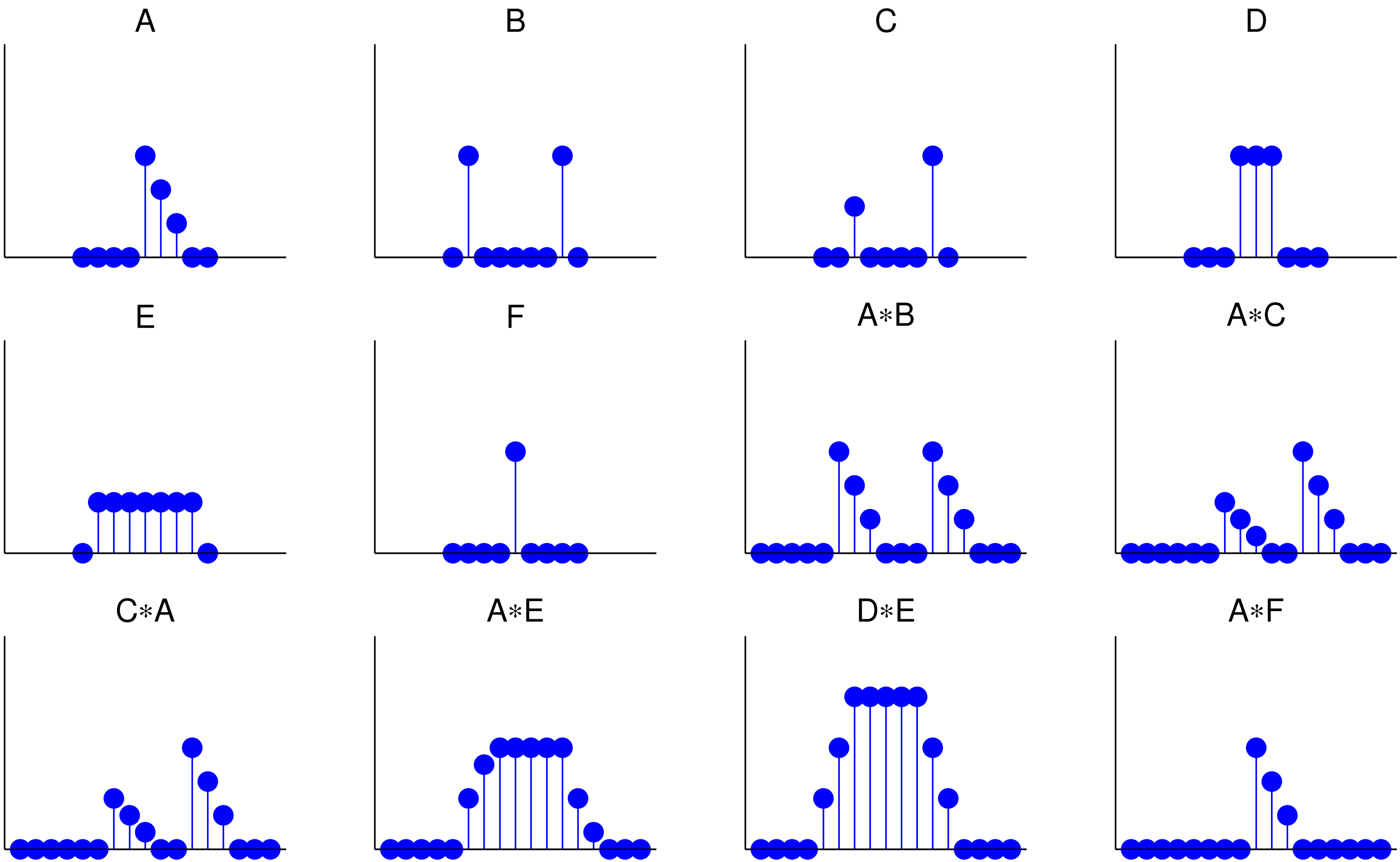
where $\{a_k\}$ is a suitably chosen sequence of coefficients.

This operation over sequences is called *convolution* and defined as

$$\{p_n\} * \{q_n\} = \{r_n\} \quad \Longleftrightarrow \quad \forall n \in \mathbb{Z} : r_n = \sum_{k=-\infty}^{\infty} p_k \cdot q_{n-k}.$$

If $\{y_n\} = \{a_n\} * \{x_n\}$ is a representation of an LTI system T, with $\{y_n\} = T\{x_n\}$, then we call the sequence $\{a_n\}$ the *impulse response* of T, because $\{a_n\} = T\{\delta_n\}$.

Convolution examples



Properties of convolution

For arbitrary sequences $\{p_n\}$, $\{q_n\}$, $\{r_n\}$ and scalars a, b:

→ Convolution is associative

$$(\{p_n\} * \{q_n\}) * \{r_n\} = \{p_n\} * (\{q_n\} * \{r_n\})$$

→ Convolution is commutative

$$\{p_n\} * \{q_n\} = \{q_n\} * \{p_n\}$$

→ Convolution is linear

$$\{p_n\} * \{a \cdot q_n + b \cdot r_n\} = a \cdot (\{p_n\} * \{q_n\}) + b \cdot (\{p_n\} * \{r_n\})$$

→ The impulse sequence (slide 12) is neutral under convolution

$$\{p_n\} * \{\delta_n\} = \{\delta_n\} * \{p_n\} = \{p_n\}$$

→ Sequence shifting is equivalent to convolving with a shifted impulse

$$\{p_{n-d}\} = \{p_n\} * \{\delta_{n-d}\}$$

Proof: all LTI systems just apply convolution

Any sequence $\{x_n\}$ can be decomposed into a weighted sum of shifted impulse sequences:

$$\{x_n\} = \sum_{k=-\infty}^{\infty} x_k \cdot \{\delta_{n-k}\}$$

Let's see what happens if we apply a linear$^{(*)}$ time-invariant$^{(**)}$ system T to such a decomposed sequence:

$$\begin{aligned}
T\{x_n\} &= T\left(\sum_{k=-\infty}^{\infty} x_k \cdot \{\delta_{n-k}\}\right) \stackrel{(*)}{=} \sum_{k=-\infty}^{\infty} x_k \cdot T\{\delta_{n-k}\} \\
&\stackrel{(**)}{=} \sum_{k=-\infty}^{\infty} x_k \cdot \{\delta_{n-k}\} * T\{\delta_n\} = \left(\sum_{k=-\infty}^{\infty} x_k \cdot \{\delta_{n-k}\}\right) * T\{\delta_n\} \\
&= \{x_n\} * T\{\delta_n\} \qquad \text{q.e.d.}
\end{aligned}$$

$\Rightarrow$ The impulse response $T\{\delta_n\}$ fully characterizes an LTI system.

Exercise 1 What type of discrete system (linear/non-linear, time-invariant/non-time-invariant, causal/non-causal, causal, memory-less, etc.) is:

(a) $y_n = |x_n|$

(b) $y_n = -x_{n-1} + 2x_n - x_{n+1}$

(c) $y_n = \prod_{i=0}^{8} x_{n-i}$

(d) $y_n = \frac{1}{2}(x_{2n} + x_{2n+1})$

(e) $y_n = \frac{3x_{n-1} + x_{n-2}}{x_{n-3}}$

(f) $y_n = x_n \cdot \mathrm{e}^{n/14}$

(g) $y_n = x_n \cdot u_n$

(h) $y_n = \sum_{i=-\infty}^{\infty} x_i \cdot \delta_{i-n+2}$

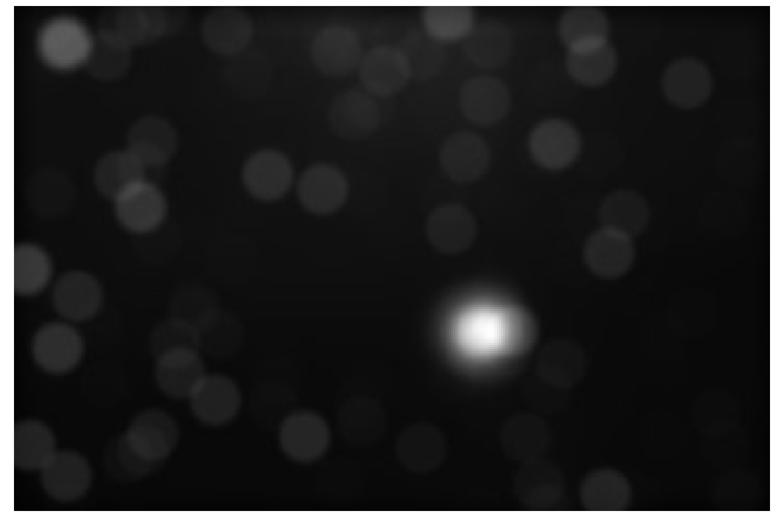
Exercise 2

Prove that convolution is (a) commutative and (b) associative.

Convolution: optics example

If a projective lens is out of focus, the blurred image is equal to the original image convolved with the aperture shape (e.g., a filled circle):

$* \bullet =$

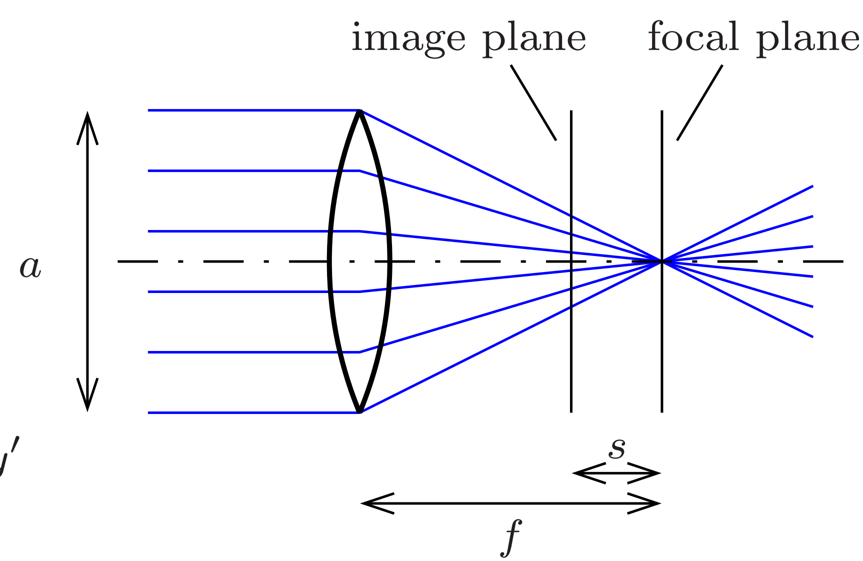


Point-spread function h (disk, $r = \frac{as}{2f}$):

$$h(x,y) = \begin{cases} \frac{1}{r^2\pi}, & x^2+y^2 \le r^2 \\ 0, & x^2+y^2 > r^2 \end{cases}$$

Original image I, blurred image $B = I * h$, i.e.

$$B(x,y) = \iint I(x-x', y-y') \cdot h(x',y') \cdot \mathrm{d}x'\mathrm{d}y'$$



Why are sine waves useful?

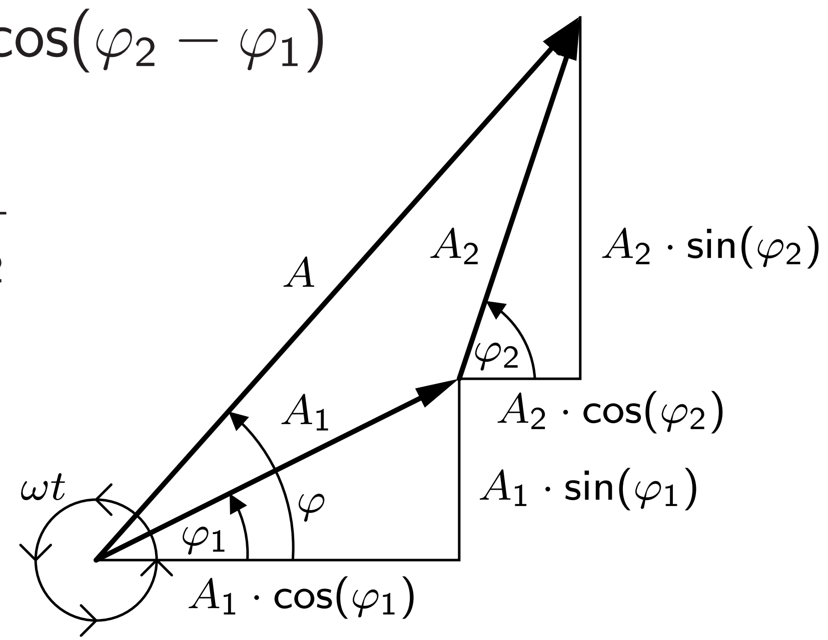
1) Adding together sine waves of equal frequency, but arbitrary amplitude and phase, results in another sine wave of the same frequency:

$$A_1 \cdot \sin(\omega t + \varphi_1) + A_2 \cdot \sin(\omega t + \varphi_2) \;=\; A \cdot \sin(\omega t + \varphi)$$

with

$$A \;=\; \sqrt{A_1^2 + A_2^2 + 2A_1A_2\cos(\varphi_2 - \varphi_1)}$$

$$\tan\varphi \;=\; \frac{A_1\sin\varphi_1 + A_2\sin\varphi_2}{A_1\cos\varphi_1 + A_2\cos\varphi_2}$$



Sine waves of any phase can be formed from sin and cos alone:

$$A \cdot \sin(\omega t + \varphi) = \\ a \cdot \sin(\omega t) + b \cdot \cos(\omega t)$$

with $a = A \cdot \cos(\varphi)$, $b = A \cdot \sin(\varphi)$ and $A = \sqrt{a^2 + b^2}$, $\tan\varphi = \frac{b}{a}$.

Note: Convolution of a discrete sequence $\{x_n\}$ with another sequence $\{y_n\}$ is nothing but adding together scaled and delayed copies of $\{x_n\}$. (Think of $\{y_n\}$ decomposed into a sum of impulses.)

If $\{x_n\}$ is a sampled sine wave of frequency f, so is $\{x_n\} * \{y_n\}$!

$\Longrightarrow$ **Sine-wave sequences form a family of discrete sequences that is closed under convolution with arbitrary sequences.**

The same applies for continuous sine waves and convolution.

2) Sine waves are orthogonal to each other:

$$\int_{-\infty}^{\infty} \sin(\omega_1 t + \varphi_1) \cdot \sin(\omega_2 t + \varphi_2)\, \mathrm{d}t \text{ ``$=$'' } 0$$

$$\Longleftrightarrow \quad \omega_1 \neq \omega_2 \quad \vee \quad \varphi_1 - \varphi_2 = (2k+1)\pi/2 \quad (k \in \mathbb{Z})$$

They can be used to form an orthogonal function basis for a transform.

The term "orthogonal" is used here in the context of an (infinitely dimensional) vector space, where the "vectors" are functions of the form $f : \mathbb{R} \to \mathbb{R}$ (or $f : \mathbb{R} \to \mathbb{C}$) and the scalar product is defined as $f \cdot g = \int_{-\infty}^{\infty} f(t) \cdot g(t)\, \mathrm{d}t$.

Why are exponential functions useful?

Adding together two exponential functions with the same base z, but different scale factor and offset, results in another exponential function with the same base:

$$\begin{aligned} A_1 \cdot z^{t+\varphi_1} + A_2 \cdot z^{t+\varphi_2} &= A_1 \cdot z^t \cdot z^{\varphi_1} + A_2 \cdot z^t \cdot z^{\varphi_2} \\ &= (A_1 \cdot z^{\varphi_1} + A_2 \cdot z^{\varphi_2}) \cdot z^t = A \cdot z^t \end{aligned}$$

Likewise, if we convolve a sequence $\{x_n\}$ of values

$$\ldots, z^{-3}, z^{-2}, z^{-1}, 1, z, z^2, z^3, \ldots$$

$x_n = z^n$ with an arbitrary sequence $\{h_n\}$, we get $\{y_n\} = \{z^n\} * \{h_n\}$,

$$y_n = \sum_{k=-\infty}^{\infty} x_{n-k} \cdot h_k = \sum_{k=-\infty}^{\infty} z^{n-k} \cdot h_k = z^n \cdot \sum_{k=-\infty}^{\infty} z^{-k} \cdot h_k = z^n \cdot H(z)$$

where H(z) is independent of n.

Exponential sequences are closed under convolution with arbitrary sequences. The same applies in the continuous case.

Why are complex numbers so useful?

1) They give us all n solutions ("roots") of equations involving polynomials up to degree n (the " $\sqrt{-1} = \mathrm{j}$ " story).

2) They give us the "great unifying theory" that combines sine and exponential functions:

$$
\begin{aligned}
\cos(\omega t) &= \frac{1}{2}\left(\mathrm{e}^{\mathrm{j}\omega t} + \mathrm{e}^{-\mathrm{j}\omega t}\right) \\
\sin(\omega t) &= \frac{1}{2\mathrm{j}}\left(\mathrm{e}^{\mathrm{j}\omega t} - \mathrm{e}^{-\mathrm{j}\omega t}\right)
\end{aligned}
$$

or

$$
\cos(\omega t + \varphi) = \frac{1}{2}\left(\mathrm{e}^{\mathrm{j}\omega t + \varphi} + \mathrm{e}^{-\mathrm{j}\omega t - \varphi}\right)
$$

or

$$
\begin{aligned}
\cos(\omega n + \varphi) &= \Re(\mathrm{e}^{\mathrm{j}\omega n + \varphi}) = \Re[(\mathrm{e}^{\mathrm{j}\omega})^n \cdot \mathrm{e}^{\mathrm{j}\varphi}] \\
\sin(\omega n + \varphi) &= \Im(\mathrm{e}^{\mathrm{j}\omega n + \varphi}) = \Im[(\mathrm{e}^{\mathrm{j}\omega})^n \cdot \mathrm{e}^{\mathrm{j}\varphi}]
\end{aligned}
$$

Notation: $\Re(a + \mathrm{j}b) := a$ and $\Im(a + \mathrm{j}b) := b$ where $\mathrm{j}^2 = -1$ and $a, b \in \mathbb{R}$.

We can now represent sine waves as projections of a rotating complex vector. This allows us to represent sine-wave sequences as exponential sequences with basis $\mathrm{e}^{\mathrm{j}\omega}$.

A phase shift in such a sequence corresponds to a rotation of a complex vector.

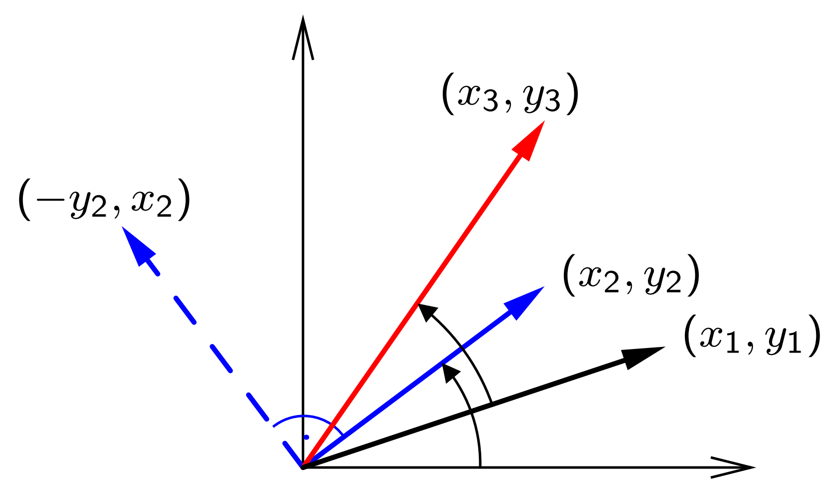
3) Complex multiplication allows us to modify the amplitude and phase of a complex rotating vector using a single operation and value.

Rotation of a 2D vector in (x, y)-form is notationally slightly messy, but fortunately $\mathrm{j}^2 = -1$ does exactly what is required here:

$$\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} = \begin{pmatrix} x_2 & -y_2 \\ y_2 & x_2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_1x_2 - y_1y_2 \\ x_1y_2 + x_2y_1 \end{pmatrix}$$

$$z_1 = x_1 + \mathrm{j}y_1, \quad z_2 = x_2 + \mathrm{j}y_2$$

$$z_1 \cdot z_2 = x_1x_2 - y_1y_2 + \mathrm{j}(x_1y_2 + x_2y_1)$$



Recall: Fourier transform

We define the Fourier integral transform and its inverse as

$$\mathcal{F}\{g(t)\}(f) \quad = \quad G(f) \quad = \quad \int_{-\infty}^{\infty} g(t) \cdot \mathrm{e}^{-2\pi \mathrm{j} f t}\, \mathrm{d}t$$

$$\mathcal{F}^{-1}\{G(f)\}(t) \quad = \quad g(t) \quad = \quad \int_{-\infty}^{\infty} G(f) \cdot \mathrm{e}^{2\pi \mathrm{j} f t}\, \mathrm{d}f$$

Many equivalent forms of the Fourier transform are used in the literature. There is no strong consensus on whether the forward transform uses $\mathrm{e}^{-2\pi \mathrm{j} f t}$ and the backwards transform $\mathrm{e}^{2\pi \mathrm{j} f t}$, or vice versa. The above form uses the *ordinary frequency* f, whereas some authors prefer the *angular frequency* $\omega = 2\pi f$:

$$\mathcal{F}\{h(t)\}(\omega) \quad = \quad H(\omega) \quad = \quad \alpha \int_{-\infty}^{\infty} h(t) \cdot \mathrm{e}^{\mp \mathrm{j} \omega t}\, \mathrm{d}t$$

$$\mathcal{F}^{-1}\{H(\omega)\}(t) \quad = \quad h(t) \quad = \quad \beta \int_{-\infty}^{\infty} H(\omega) \cdot \mathrm{e}^{\pm \mathrm{j} \omega t}\, \mathrm{d}\omega$$

This substitution introduces factors α and β such that $\alpha\beta = 1/(2\pi)$. Some authors set $\alpha = 1$ and $\beta = 1/(2\pi)$, to keep the convolution theorem free of a constant prefactor; others prefer the unitary form $\alpha = \beta = 1/\sqrt{2\pi}$, in the interest of symmetry.

Properties of the Fourier transform

If

$$x(t) \quad \bullet\!\!-\!\!\circ \quad X(f) \qquad \text{and} \qquad y(t) \quad \bullet\!\!-\!\!\circ \quad Y(f)$$

are pairs of functions that are mapped onto each other by the Fourier transform, then so are the following pairs.

Linearity:

$$ax(t) + by(t) \quad \bullet\!\!-\!\!\circ \quad aX(f) + bY(f)$$

Time scaling:

$$x(at) \quad \bullet\!\!-\!\!\circ \quad \frac{1}{|a|} X\left(\frac{f}{a}\right)$$

Frequency scaling:

$$\frac{1}{|a|} x\left(\frac{t}{a}\right) \quad \bullet\!\!-\!\!\circ \quad X(af)$$

Time shifting:

$$x(t - \Delta t) \quad \bullet\!\!-\!\!\circ \quad X(f) \cdot \mathrm{e}^{-2\pi \mathrm{j} f \Delta t}$$

Frequency shifting:

$$x(t) \cdot \mathrm{e}^{2\pi \mathrm{j} \Delta f t} \quad \bullet\!\!-\!\!\circ \quad X(f - \Delta f)$$

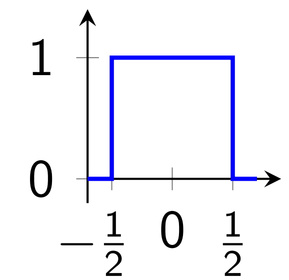
Parseval's theorem (total energy):

$$\int_{-\infty}^{\infty} |x(t)|^2 \mathrm{d}t \quad = \quad \int_{-\infty}^{\infty} |X(f)|^2 \mathrm{d}f$$

Fourier transform example: rect and sinc

The Fourier transform of the "rectangular function"

$$\mathrm{rect}(t) = \begin{cases} 1 & \text{if } |t| < \frac{1}{2} \\ \frac{1}{2} & \text{if } |t| = \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$



is the "(normalized) sinc function"

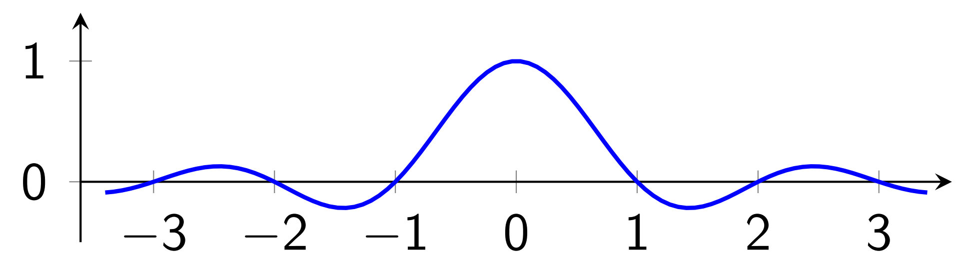
$$\mathcal{F}\{\mathrm{rect}(t)\}(f) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \mathrm{e}^{-2\pi \mathrm{j} f t} \mathrm{d}t = \frac{\sin \pi f}{\pi f} = \mathrm{sinc}(f)$$

and vice versa

$$\mathcal{F}\{\mathrm{sinc}(t)\}(f) = \mathrm{rect}(f).$$

Some noteworthy properties of these functions:

- $\int_{-\infty}^{\infty} \mathrm{sinc}(t)\, \mathrm{d}t = 1 = \int_{-\infty}^{\infty} \mathrm{rect}(t)\, \mathrm{d}t$
- $\mathrm{sinc}(0) = 1 = \mathrm{rect}(0)$
- $\forall n \in \mathbb{Z} \setminus \{0\} : \mathrm{sinc}(n) = 0$



Convolution theorem

Continuous form:

$$\begin{aligned} \mathcal{F}\{(f*g)(t)\} &= \mathcal{F}\{f(t)\}\cdot\mathcal{F}\{g(t)\} \\ \mathcal{F}\{f(t)\cdot g(t)\} &= \mathcal{F}\{f(t)\}*\mathcal{F}\{g(t)\} \end{aligned}$$

Discrete form:

$$\{x_n\}*\{y_n\} = \{z_n\} \quad\Longleftrightarrow\quad X(\mathrm{e}^{\mathrm{j}\omega})\cdot Y(\mathrm{e}^{\mathrm{j}\omega}) = Z(\mathrm{e}^{\mathrm{j}\omega})$$

Convolution in the time domain is equivalent to (complex) scalar multiplication in the frequency domain.

Convolution in the frequency domain corresponds to scalar multiplication in the time domain.

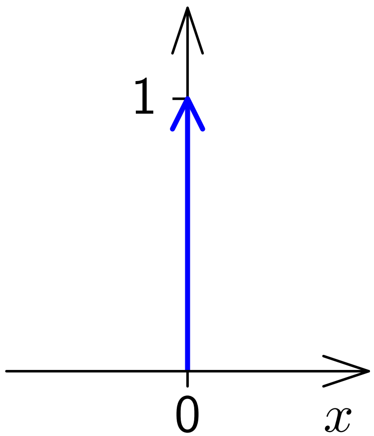
Proof: $z(r)=\int_s x(s)y(r-s)\mathrm{d}s \quad\Longleftrightarrow\quad \int_r z(r)\mathrm{e}^{-\mathrm{j}\omega r}\mathrm{d}r = \int_r\int_s x(s)y(r-s)\mathrm{e}^{-\mathrm{j}\omega r}\mathrm{d}s\mathrm{d}r =$
$\int_s x(s)\int_r y(r-s)\mathrm{e}^{-\mathrm{j}\omega r}\mathrm{d}r\mathrm{d}s = \int_s x(s)\mathrm{e}^{-\mathrm{j}\omega s}\int_r y(r-s)\mathrm{e}^{-\mathrm{j}\omega(r-s)}\mathrm{d}r\mathrm{d}s \overset{t:=r-s}{=}$
$\int_s x(s)\mathrm{e}^{-\mathrm{j}\omega s}\int_t y(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t\mathrm{d}s = \int_s x(s)\mathrm{e}^{-\mathrm{j}\omega s}\mathrm{d}s\cdot\int_t y(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t$. (Same for $\sum$ instead of $\int$.)

Dirac delta function

The continuous equivalent of the impulse sequence $\{\delta_n\}$ is known as Dirac delta function $\delta(x)$. It is a generalized function, defined such that

$$\delta(x) = \begin{cases} 0, & x \neq 0 \\ \infty, & x = 0 \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(x)\,\mathrm{d}x = 1$$



and can be thought of as the limit of function sequences such as

$$\delta(x) = \lim_{n\to\infty} \begin{cases} 0, & |x| \geq 1/n \\ n/2, & |x| < 1/n \end{cases}$$

or

$$\delta(x) = \lim_{n\to\infty} \frac{n}{\sqrt{\pi}}\,\mathrm{e}^{-n^2x^2}$$

The delta function is mathematically speaking not a function, but a *distribution*, that is an expression that is only defined when integrated.

Some properties of the Dirac delta function:

$$\begin{aligned}
\int_{-\infty}^{\infty} f(x)\delta(x-a)\,\mathrm{d}x &= f(a) \\
\int_{-\infty}^{\infty} \mathrm{e}^{\pm 2\pi \mathrm{j} xa}\mathrm{d}x &= \delta(a) \\
\sum_{n=-\infty}^{\infty} \mathrm{e}^{\pm 2\pi \mathrm{j} nxa} &= \frac{1}{|a|}\sum_{n=-\infty}^{\infty} \delta(x-n/a) \\
\delta(ax) &= \frac{1}{|a|}\delta(x)
\end{aligned}$$

Fourier transform:

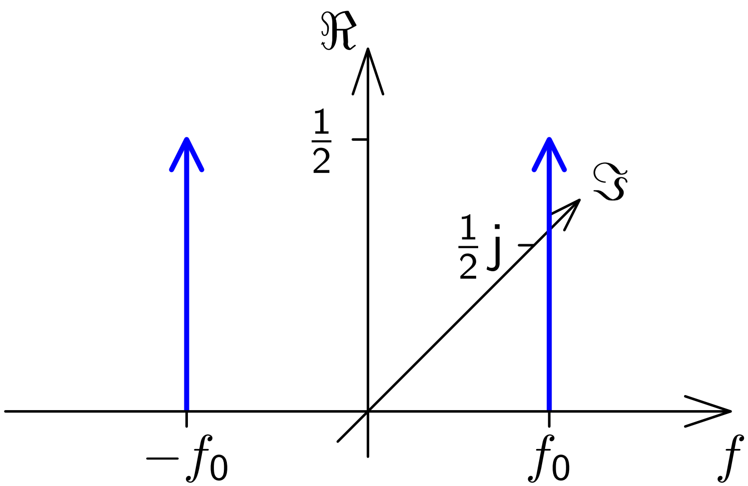
$$\begin{aligned}
\mathcal{F}\{\delta(t)\}(f) &= \int_{-\infty}^{\infty} \delta(t) \cdot \mathrm{e}^{-2\pi \mathrm{j} ft}\,\mathrm{d}t = \mathrm{e}^{0} = 1 \\
\mathcal{F}^{-1}\{1\}(t) &= \int_{-\infty}^{\infty} 1 \cdot \mathrm{e}^{2\pi \mathrm{j} ft}\,\mathrm{d}f = \delta(t)
\end{aligned}$$

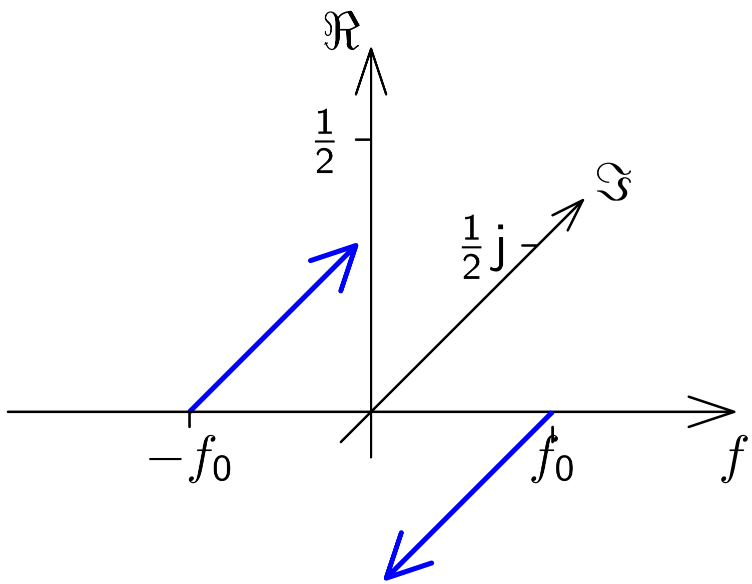
Sine and cosine in the frequency domain

$$\cos(2\pi f_0 t) = \frac{1}{2}\,\mathrm{e}^{2\pi \mathrm{j} f_0 t} + \frac{1}{2}\,\mathrm{e}^{-2\pi \mathrm{j} f_0 t} \qquad \sin(2\pi f_0 t) = \frac{1}{2\mathrm{j}}\,\mathrm{e}^{2\pi \mathrm{j} f_0 t} - \frac{1}{2\mathrm{j}}\,\mathrm{e}^{-2\pi \mathrm{j} f_0 t}$$

$$\mathcal{F}\{\cos(2\pi f_0 t)\}(f) = \frac{1}{2}\delta(f - f_0) + \frac{1}{2}\delta(f + f_0)$$

$$\mathcal{F}\{\sin(2\pi f_0 t)\}(f) = -\frac{\mathrm{j}}{2}\delta(f - f_0) + \frac{\mathrm{j}}{2}\delta(f + f_0)$$





As any $x(t) \in \mathbb{R}$ can be decomposed into sine and cosine functions, the spectrum of any real-valued signal will show the symmetry $X(\mathrm{e}^{\mathrm{j}\omega}) = [X(\mathrm{e}^{-\mathrm{j}\omega})]^*$, where * denotes the complex conjugate (i.e., negated imaginary part).

Fourier transform symmetries

We call a function $x(t)$

$$
\begin{aligned}
\text{odd if } \; x(-t) &= -x(t) \\
\text{even if } \; x(-t) &= x(t)
\end{aligned}
$$

and $\cdot^*$ is the complex conjugate, such that $(a + \mathrm{j}b)^* = (a - \mathrm{j}b)$.

Then

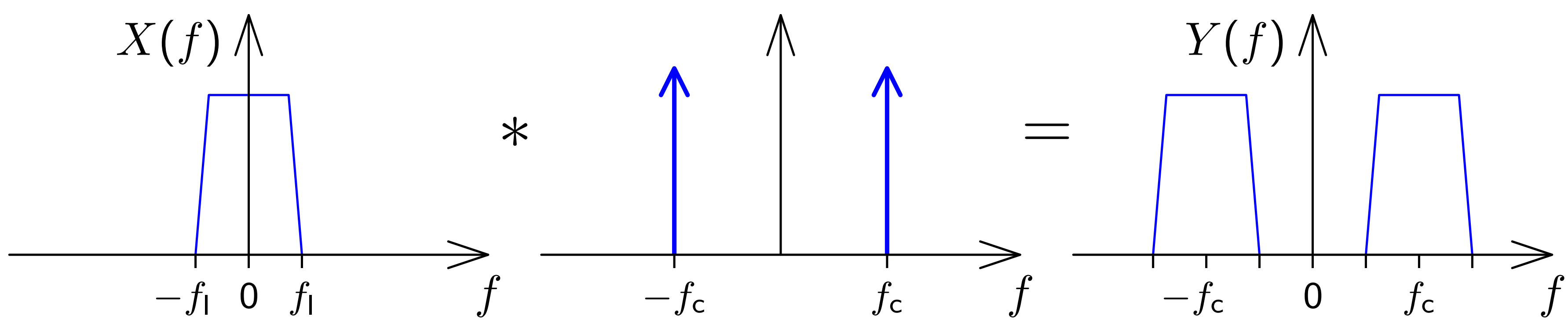
$$
\begin{aligned}
&x(t) \text{ is real} && \Leftrightarrow && X(-f) = [X(f)]^* \\
&x(t) \text{ is imaginary} && \Leftrightarrow && X(-f) = -[X(f)]^* \\
&x(t) \text{ is even} && \Leftrightarrow && X(f) \text{ is even} \\
&x(t) \text{ is odd} && \Leftrightarrow && X(f) \text{ is odd} \\
&x(t) \text{ is real and even} && \Leftrightarrow && X(f) \text{ is real and even} \\
&x(t) \text{ is real and odd} && \Leftrightarrow && X(f) \text{ is imaginary and odd} \\
&x(t) \text{ is imaginary and even} && \Leftrightarrow && X(f) \text{ is imaginary and even} \\
&x(t) \text{ is imaginary and odd} && \Leftrightarrow && X(f) \text{ is real and odd}
\end{aligned}
$$

Example: amplitude modulation

Communication channels usually permit only the use of a given frequency interval, such as 300–3400 Hz for the analog phone network or 590–598 MHz for TV channel 36. Modulation with a carrier frequency f_c shifts the spectrum of a signal $x(t)$ into the desired band.

Amplitude modulation (AM):

$$y(t) = A \cdot \cos(2\pi t f_\mathrm{c}) \cdot x(t)$$



The spectrum of the baseband signal in the interval $-f_\mathrm{l} < f < f_\mathrm{l}$ is shifted by the modulation to the intervals $\pm f_\mathrm{c} - f_\mathrm{l} < f < \pm f_\mathrm{c} + f_\mathrm{l}$.

How can such a signal be demodulated?

Sampling using a Dirac comb

The loss of information in the sampling process that converts a continuous function $x(t)$ into a discrete sequence $\{x_n\}$ defined by

$$x_n = x(t_\mathsf{s} \cdot n) = x(n/f_\mathsf{s})$$

can be modelled through multiplying $x(t)$ by a comb of Dirac impulses

$$s(t) = t_\mathsf{s} \cdot \sum_{n=-\infty}^{\infty} \delta(t - t_\mathsf{s} \cdot n)$$

to obtain the sampled function

$$\hat{x}(t) = x(t) \cdot s(t)$$

The function $\hat{x}(t)$ now contains exactly the same information as the discrete sequence $\{x_n\}$, but is still in a form that can be analysed using the Fourier transform on continuous functions.

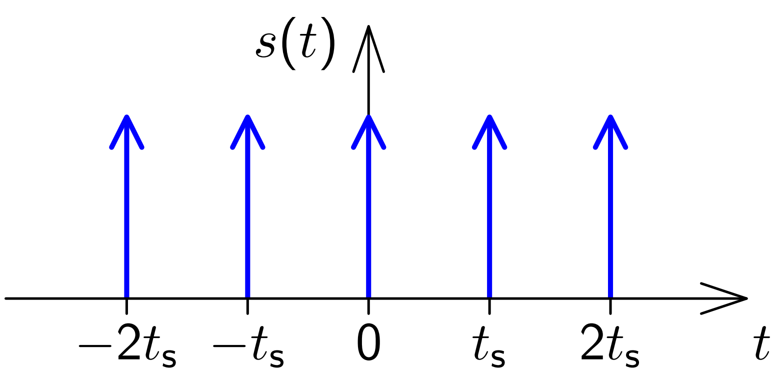
The Fourier transform of a Dirac comb

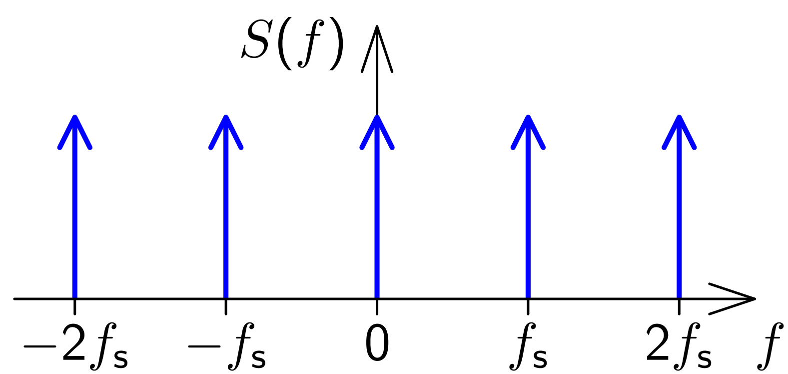
$$s(t) = t_\mathsf{s} \cdot \sum_{n=-\infty}^{\infty} \delta(t - t_\mathsf{s} \cdot n) \quad = \sum_{n=-\infty}^{\infty} \mathrm{e}^{2\pi \mathrm{j} n t / t_\mathsf{s}}$$

is another Dirac comb

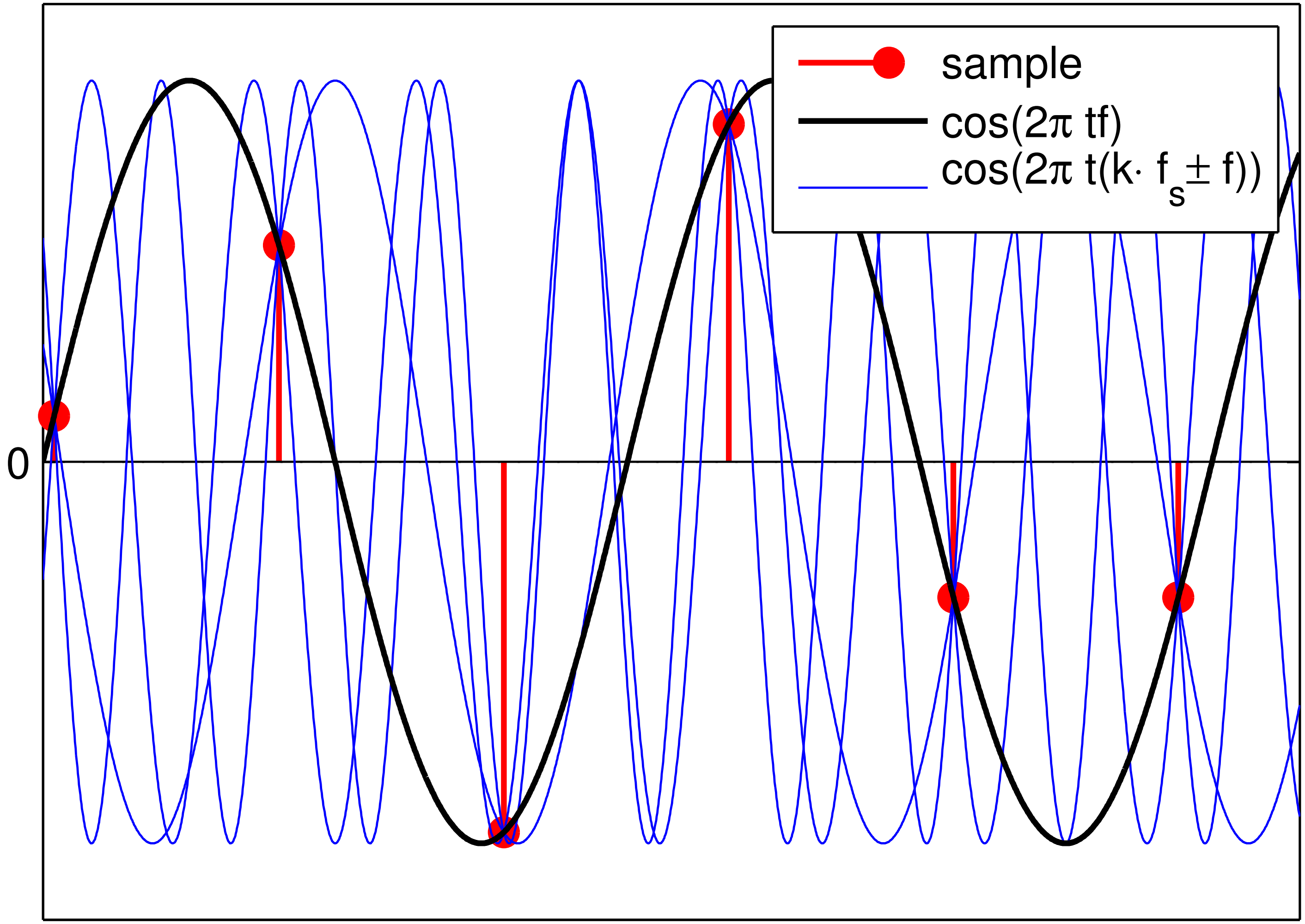
$$S(f) = \mathcal{F}\left\{ t_\mathsf{s} \ \cdot \sum_{n=-\infty}^{\infty} \delta(t - t_\mathsf{s} n) \right\}(f) =$$

$$t_\mathsf{s} \ \cdot \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(t - t_\mathsf{s} n)\, \mathrm{e}^{2\pi \mathrm{j} f t} \mathrm{d}t = \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{t_\mathsf{s}}\right).$$



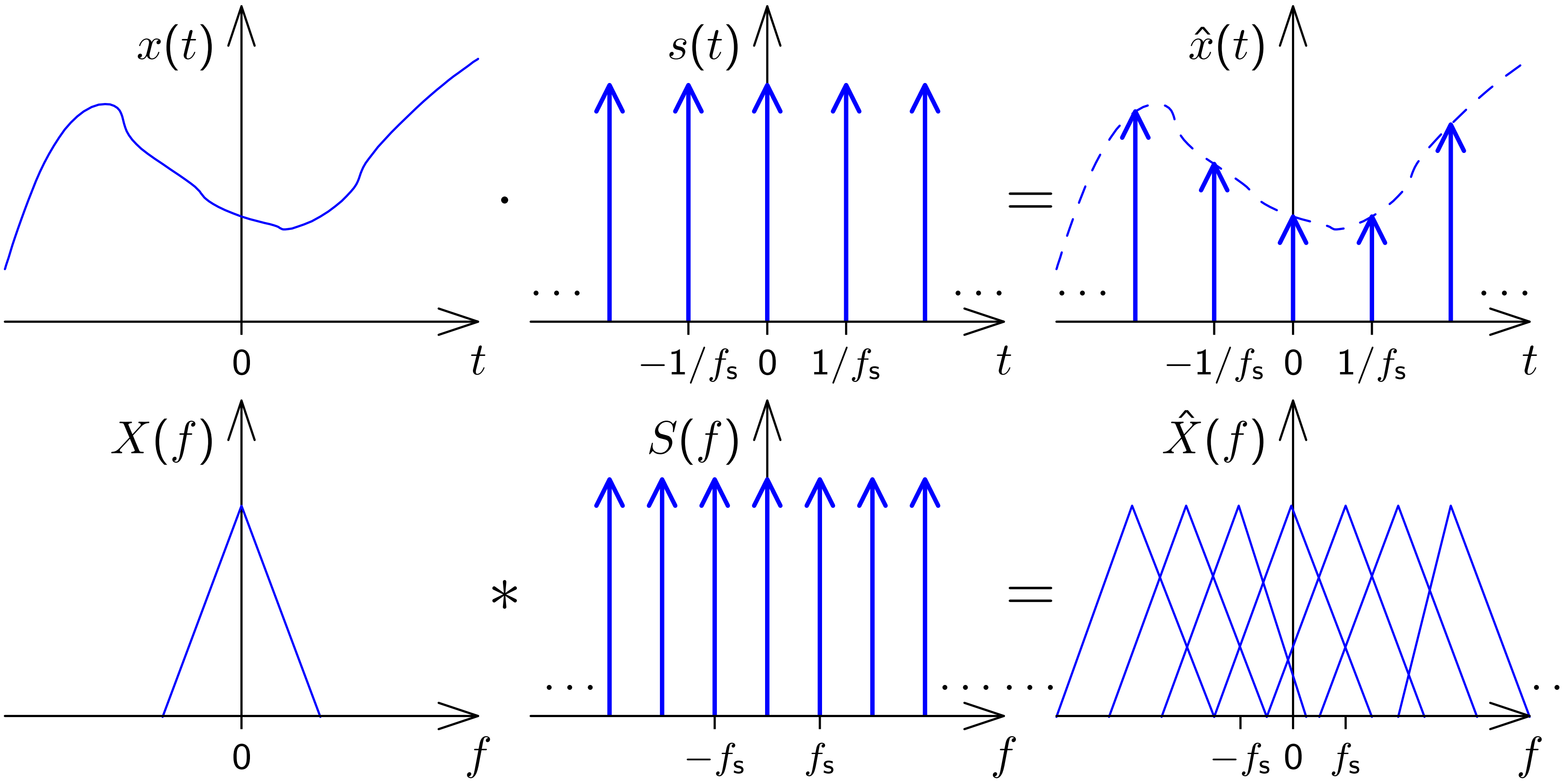


Sampling and aliasing



Sampled at frequency f_{s}, the function $\cos(2\pi t f)$ cannot be distinguished from $\cos[2\pi t(k f_{\mathsf{s}} \pm f)]$ for any $k \in \mathbb{Z}$.

Frequency-domain view of sampling



Sampling a signal in the time domain corresponds in the frequency domain to convolving its spectrum with a Dirac comb. The resulting copies of the original signal spectrum in the spectrum of the sampled signal are called "images".

Discrete-time Fourier transform

The Fourier transform of a sampled signal

$$\hat{x}(t) = t_\mathsf{s} \cdot \sum_{n=-\infty}^{\infty} x_n \cdot \delta(t - t_\mathsf{s} \cdot n)$$

is

$$\mathcal{F}\{\hat{x}(t)\}(f) = \hat{X}(f) = \int_{-\infty}^{\infty} \hat{x}(t) \cdot \mathrm{e}^{-2\pi \mathrm{j} f t} \mathrm{d}t = t_\mathsf{s} \cdot \sum_{n=-\infty}^{\infty} x_n \cdot \mathrm{e}^{-2\pi \mathrm{j} \frac{f}{f_\mathsf{s}} n}$$

Some authors prefer the notation $\hat{X}(\mathrm{e}^{\mathrm{j}\omega}) = \sum_n x_n \cdot \mathrm{e}^{-\mathrm{j}\omega n}$ to highlight the periodicity of $\hat{X}$ and its relationship with the z-transform (slide 103).

The inverse transform is

$$\hat{x}(t) = \int_{-\infty}^{\infty} \hat{X}(f) \cdot \mathrm{e}^{2\pi \mathrm{j} f t} \mathrm{d}f \quad \text{or} \quad x_m = \int_{-f_\mathsf{s}/2}^{f_\mathsf{s}/2} \hat{X}(f) \cdot \mathrm{e}^{2\pi \mathrm{j} \frac{f}{f_\mathsf{s}} m} \mathrm{d}f.$$

Nyquist limit and anti-aliasing filters

If the (double-sided) bandwidth of a signal to be sampled is larger than the sampling frequency f_s, the images of the signal that emerge during sampling may overlap with the original spectrum.

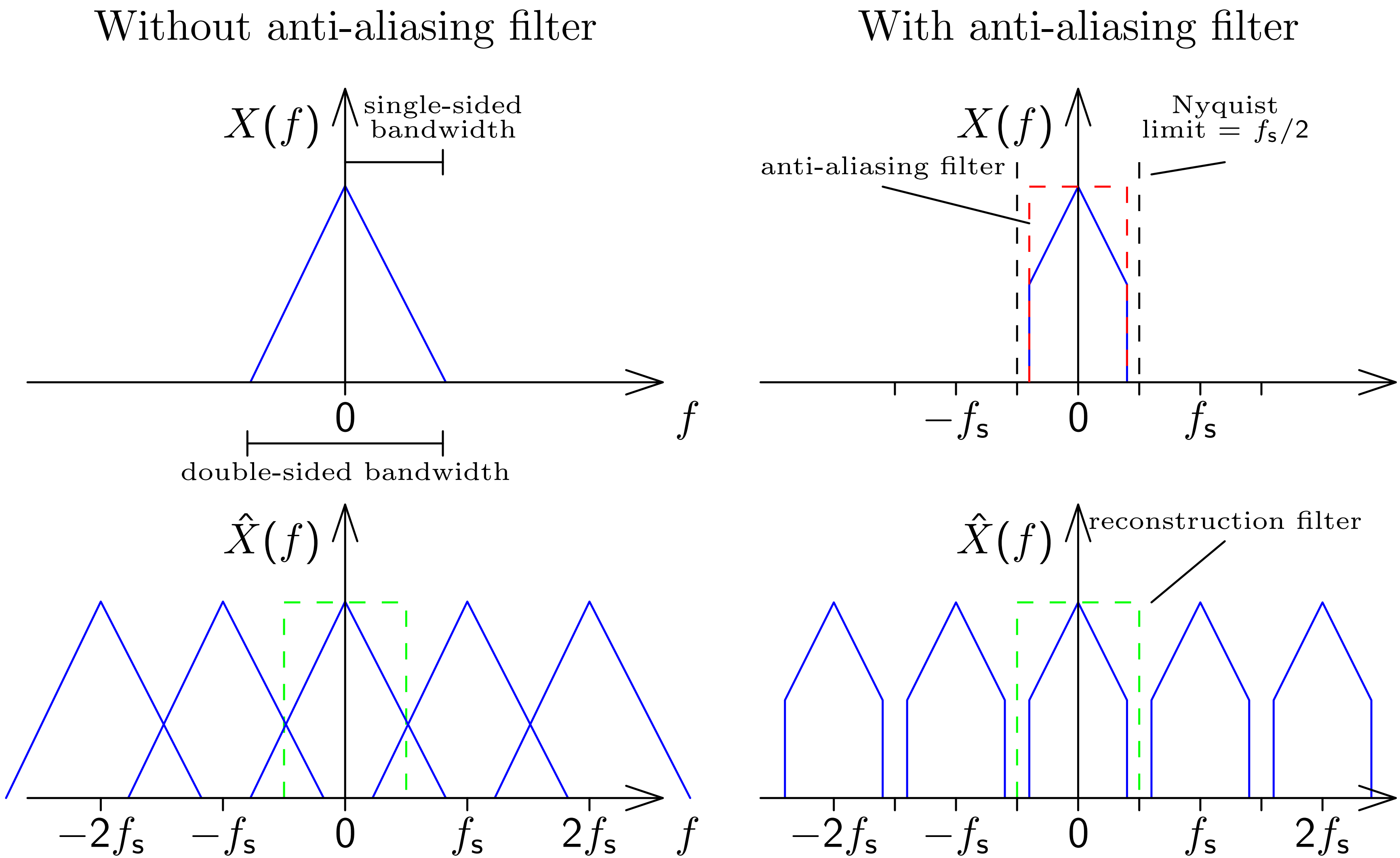
Such an overlap will hinder reconstruction of the original continuous signal by removing the aliasing frequencies with a *reconstruction filter*.

Therefore, it is advisable to limit the bandwidth of the input signal to the sampling frequency f_s before sampling, using an *anti-aliasing filter*.

In the common case of a real-valued base-band signal (with frequency content down to 0 Hz), all frequencies f that occur in the signal with non-zero power should be limited to the interval $-f_\mathrm{s}/2 < f < f_\mathrm{s}/2$.

The upper limit $f_\mathrm{s}/2$ for the single-sided bandwidth of a baseband signal is known as the "Nyquist limit".

Nyquist limit and anti-aliasing filters



Anti-aliasing and reconstruction filters both suppress frequencies outside $|f| < f_s/2$.

Reconstruction of a continuous band-limited waveform

The ideal anti-aliasing filter for eliminating any frequency content above $f_s/2$ before sampling with a frequency of f_s has the Fourier transform

$$H(f) = \begin{cases} 1 & \text{if } |f| < \frac{f_s}{2} \\ 0 & \text{if } |f| > \frac{f_s}{2} \end{cases} = \mathrm{rect}(t_s f).$$

This leads, after an inverse Fourier transform, to the impulse response

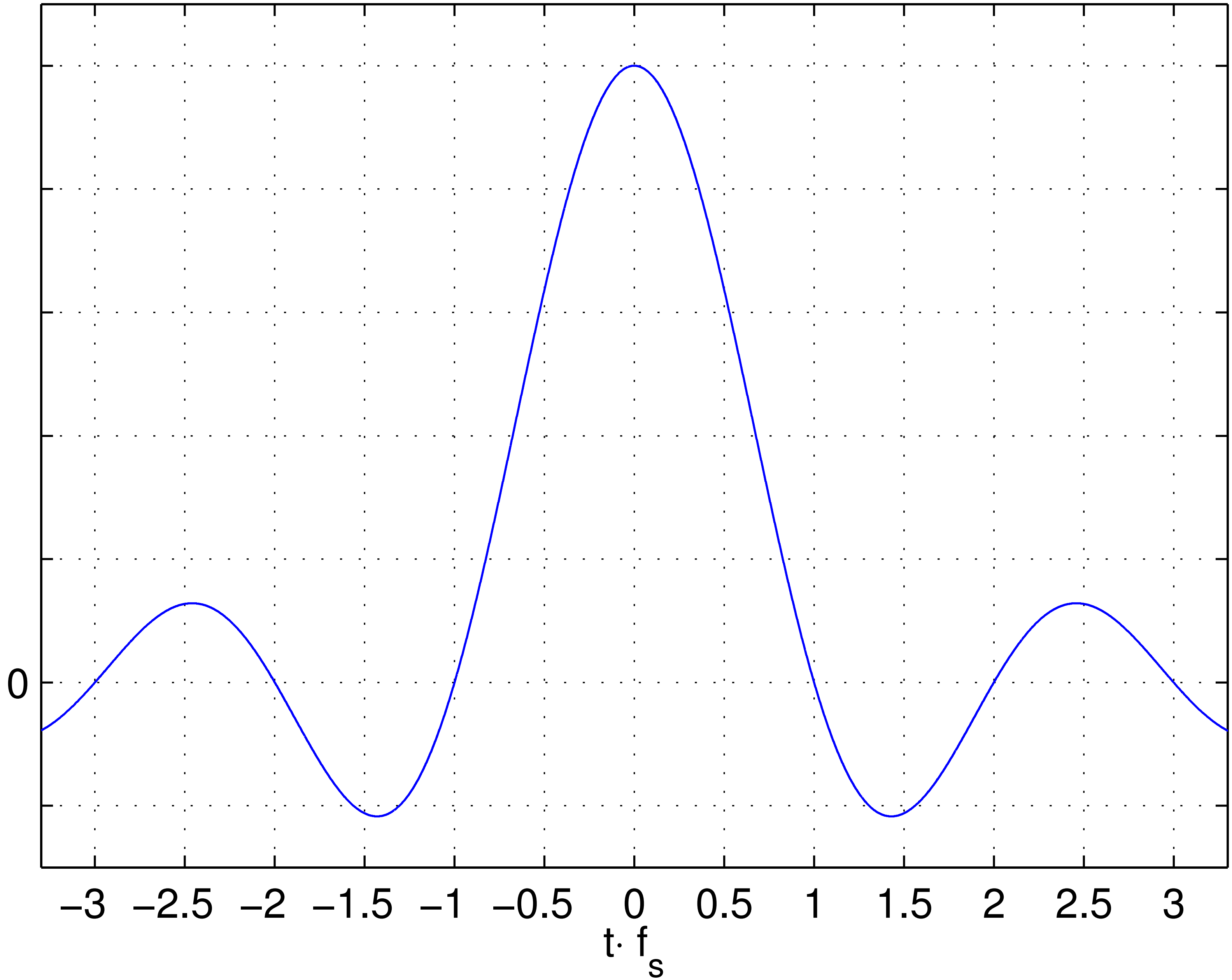
$$h(t) = f_s \cdot \frac{\sin \pi t f_s}{\pi t f_s} = \frac{1}{t_s} \cdot \mathrm{sinc}\left(\frac{t}{t_s}\right).$$

The original band-limited signal can be reconstructed by convolving this with the sampled signal $\hat{x}(t)$, which eliminates the periodicity of the frequency domain introduced by the sampling process:

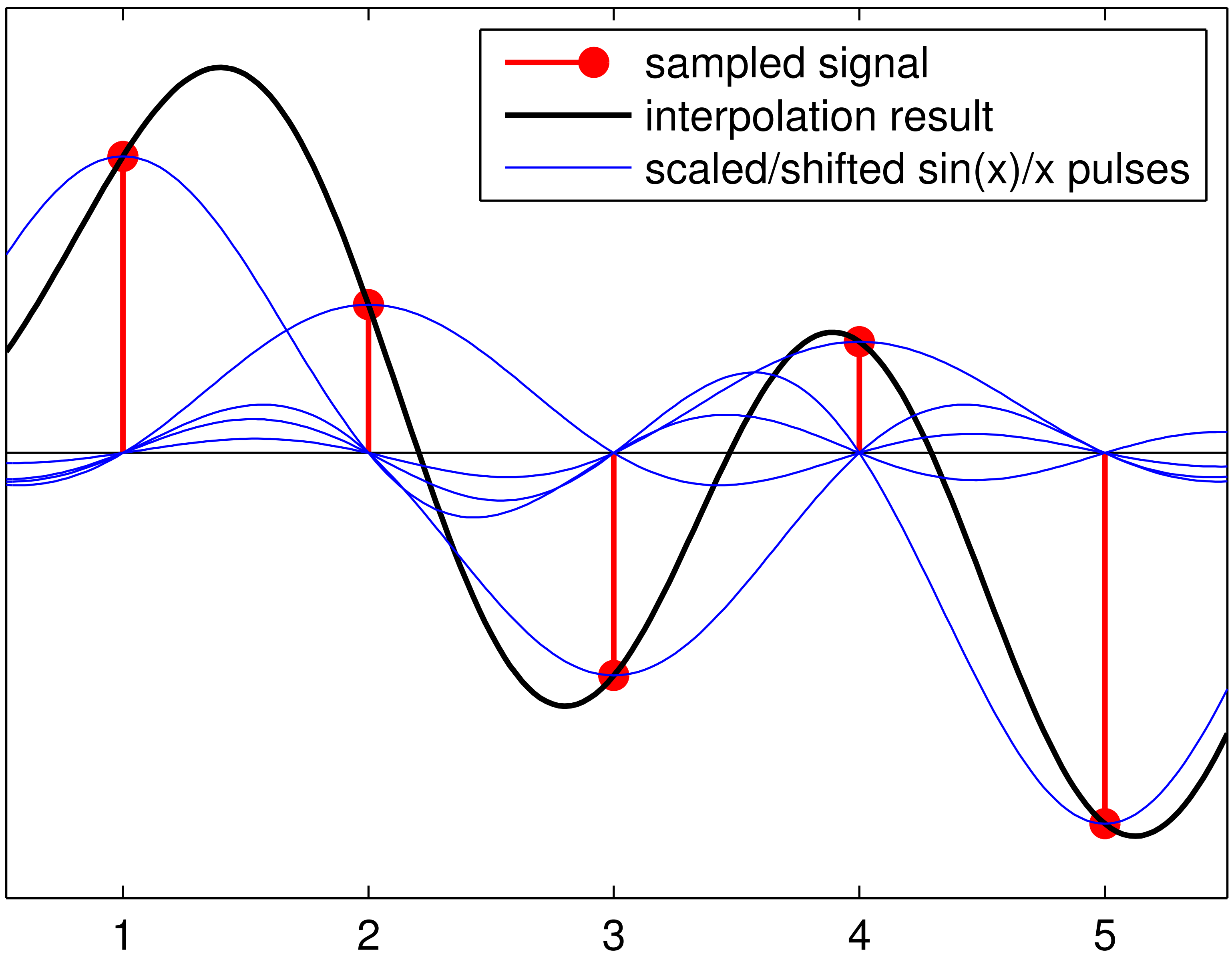
$$x(t) = h(t) * \hat{x}(t)$$

Note that sampling $h(t)$ gives the impulse function: $h(t) \cdot s(t) = \delta(t)$.

Impulse response of ideal low-pass filter with cut-off frequency $f_s/2$:

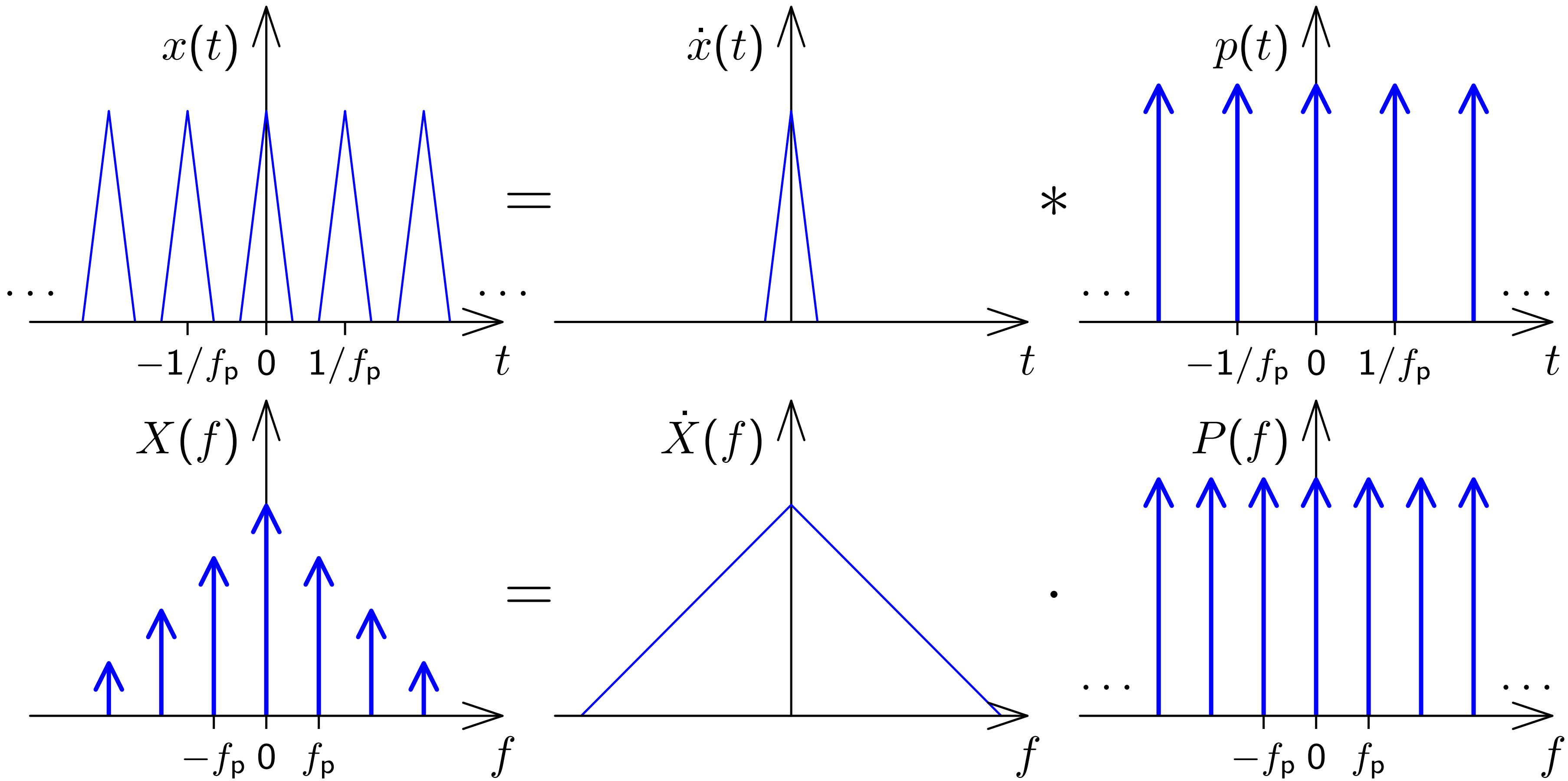


Reconstruction filter example



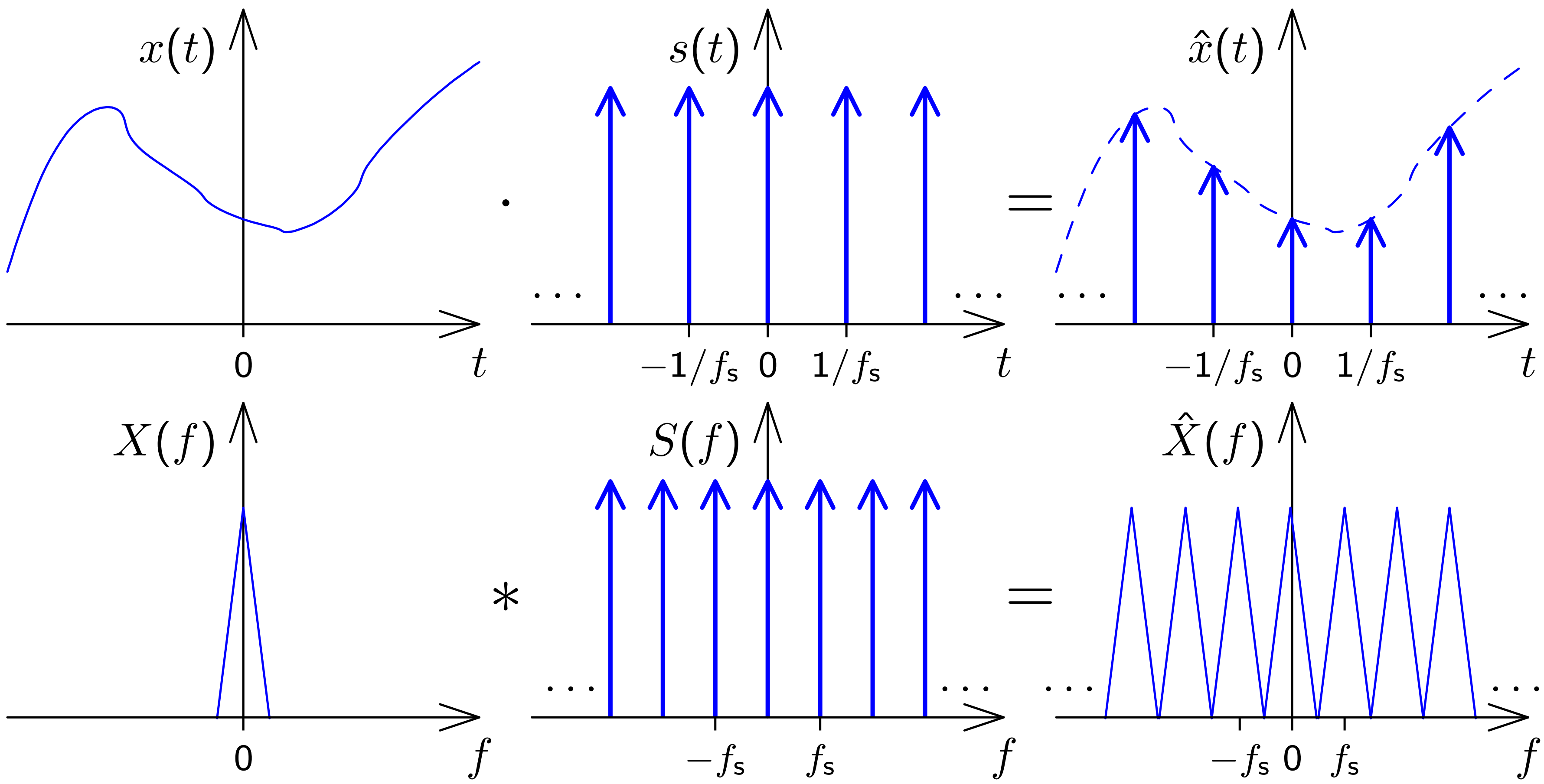
Spectrum of a periodic signal

A signal $x(t)$ that is periodic with frequency f_p can be factored into a single period $\dot{x}(t)$ convolved with an impulse comb $p(t)$. This corresponds in the frequency domain to the multiplication of the spectrum of the single period with a comb of impulses spaced f_p apart.



Spectrum of a sampled signal

A signal $x(t)$ that is sampled with frequency f_s has a spectrum that is periodic with a period of f_s.

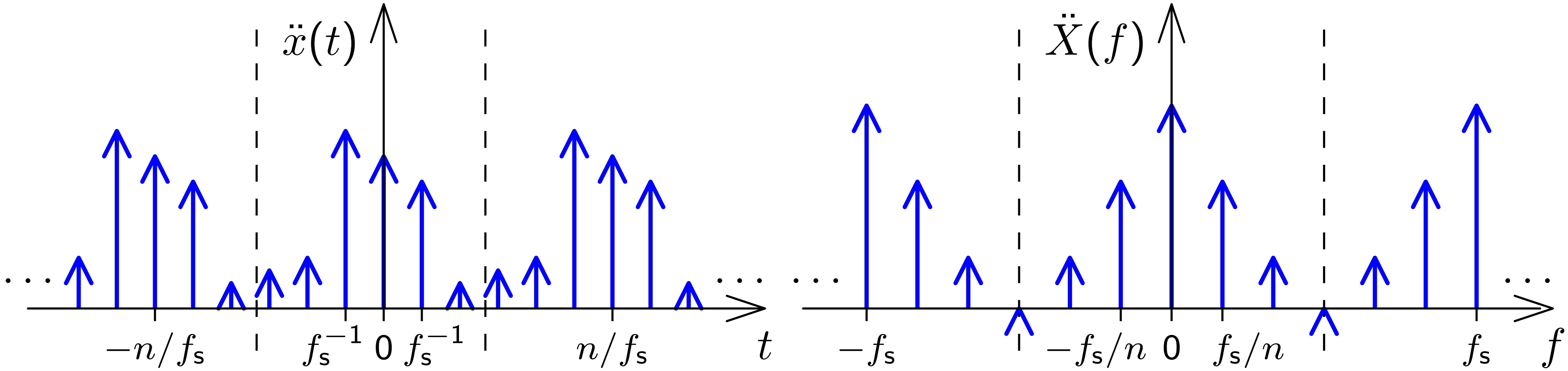


Continuous vs discrete Fourier transform

- Sampling a **continuous** signal makes its spectrum **periodic**
- A **periodic** signal has a **sampled** spectrum

We **sample** a signal $x(t)$ with f_s, getting $\hat{x}(t)$. We take n consecutive samples of $\hat{x}(t)$ and **repeat** these periodically, getting a new signal $\ddot{x}(t)$ with period n/f_s. Its spectrum $\ddot{X}(f)$ is **sampled** (i.e., has non-zero value) at frequency intervals f_s/n and **repeats** itself with a period f_s.

Now both $\ddot{x}(t)$ and its spectrum $\ddot{X}(f)$ are **finite** vectors of length n.



Discrete Fourier Transform (DFT)

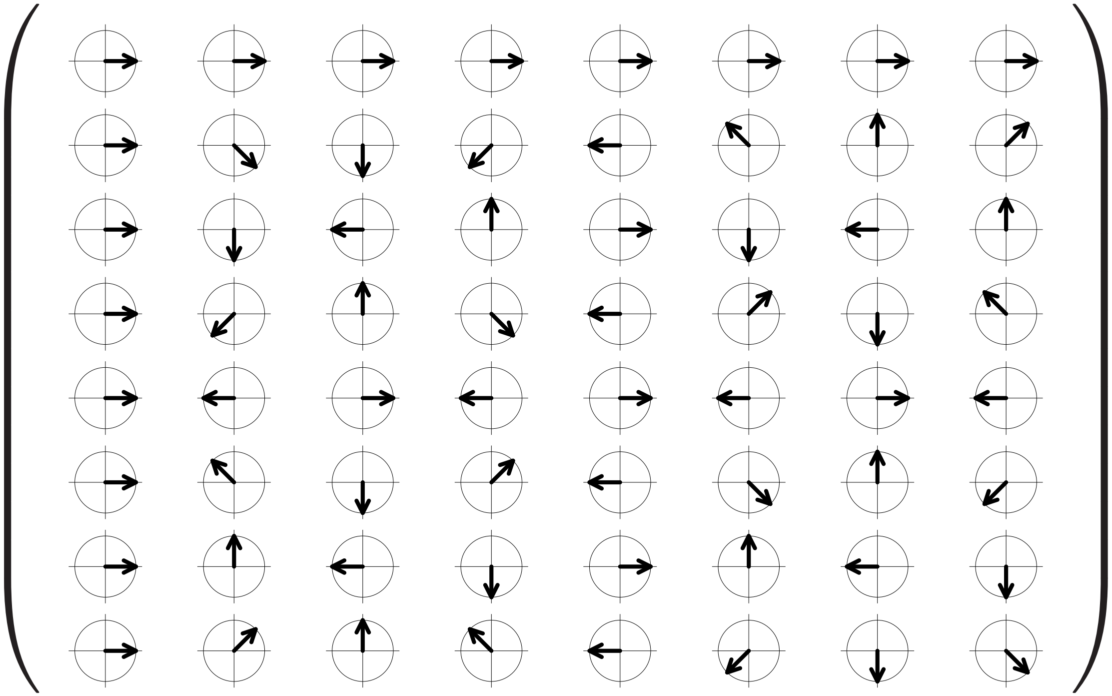
$$X_k = \sum_{i=0}^{n-1} x_i \cdot \mathrm{e}^{-2\pi\mathrm{j}\frac{ik}{n}} \qquad x_k = \frac{1}{n} \cdot \sum_{i=0}^{n-1} X_i \cdot \mathrm{e}^{2\pi\mathrm{j}\frac{ik}{n}}$$

The n-point DFT multiplies a vector with an $n \times n$ matrix

$$F_n = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \mathrm{e}^{-2\pi\mathrm{j}\frac{1}{n}} & \mathrm{e}^{-2\pi\mathrm{j}\frac{2}{n}} & \mathrm{e}^{-2\pi\mathrm{j}\frac{3}{n}} & \cdots & \mathrm{e}^{-2\pi\mathrm{j}\frac{n-1}{n}} \\ 1 & \mathrm{e}^{-2\pi\mathrm{j}\frac{2}{n}} & \mathrm{e}^{-2\pi\mathrm{j}\frac{4}{n}} & \mathrm{e}^{-2\pi\mathrm{j}\frac{6}{n}} & \cdots & \mathrm{e}^{-2\pi\mathrm{j}\frac{2(n-1)}{n}} \\ 1 & \mathrm{e}^{-2\pi\mathrm{j}\frac{3}{n}} & \mathrm{e}^{-2\pi\mathrm{j}\frac{6}{n}} & \mathrm{e}^{-2\pi\mathrm{j}\frac{9}{n}} & \cdots & \mathrm{e}^{-2\pi\mathrm{j}\frac{3(n-1)}{n}} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \mathrm{e}^{-2\pi\mathrm{j}\frac{n-1}{n}} & \mathrm{e}^{-2\pi\mathrm{j}\frac{2(n-1)}{n}} & \mathrm{e}^{-2\pi\mathrm{j}\frac{3(n-1)}{n}} & \cdots & \mathrm{e}^{-2\pi\mathrm{j}\frac{(n-1)(n-1)}{n}} \end{pmatrix}$$

$$F_n \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{n-1} \end{pmatrix}, \qquad \frac{1}{n} \cdot F_n^* \cdot \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{n-1} \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

Discrete Fourier Transform visualized



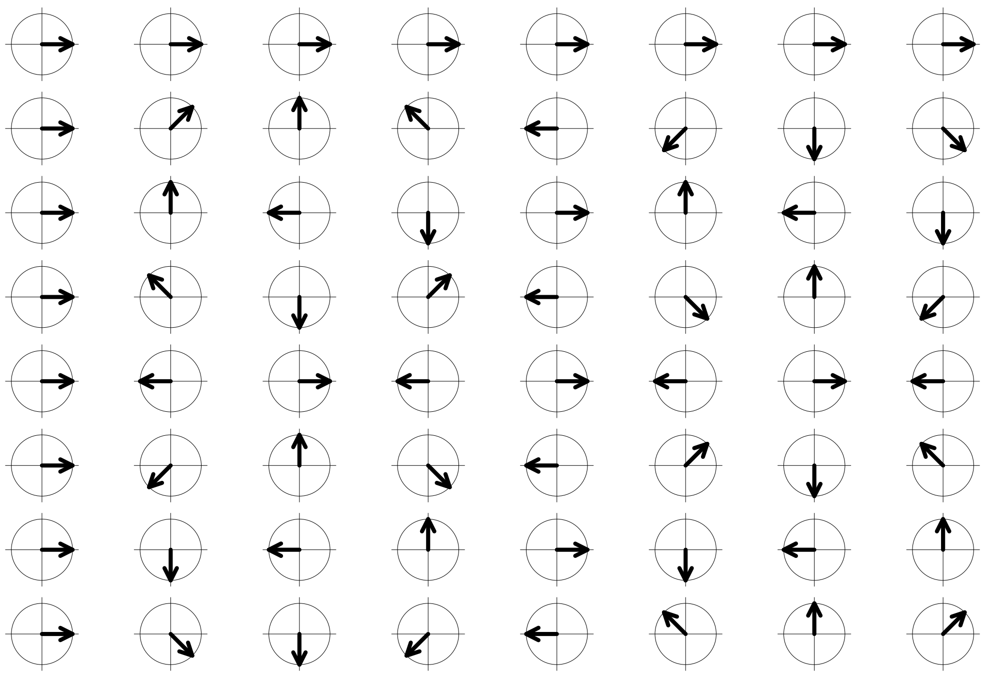
$$\cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{pmatrix}$$

The n-point DFT of a signal $\{x_i\}$ sampled at frequency f_s contains in the elements X_0 to $X_{n/2}$ of the resulting frequency-domain vector the frequency components 0, f_s/n, $2f_\mathrm{s}/n$, $3f_\mathrm{s}/n$, ..., $f_\mathrm{s}/2$, and contains in X_{n-1} downto $X_{n/2}$ the corresponding negative frequencies. Note that for a real-valued input vector, both X_0 and $X_{n/2}$ will be real, too.

Why is there no phase information recovered at $f_\mathrm{s}/2$?

Inverse DFT visualized

$$\frac{1}{8} \cdot$$



$$\cdot \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$$

Fast Fourier Transform (FFT)

$$
\begin{aligned}
\left(\mathcal{F}_n\{x_i\}_{i=0}^{n-1}\right)_k &= \sum_{i=0}^{n-1} x_i \cdot \mathrm{e}^{-2\pi\mathrm{j}\frac{ik}{n}} \\
&= \sum_{i=0}^{\frac{n}{2}-1} x_{2i} \cdot \mathrm{e}^{-2\pi\mathrm{j}\frac{ik}{n/2}} \;+\; \mathrm{e}^{-2\pi\mathrm{j}\frac{k}{n}} \sum_{i=0}^{\frac{n}{2}-1} x_{2i+1} \cdot \mathrm{e}^{-2\pi\mathrm{j}\frac{ik}{n/2}} \\
&= \begin{cases}
\left(\mathcal{F}_{\frac{n}{2}}\{x_{2i}\}_{i=0}^{\frac{n}{2}-1}\right)_k + \mathrm{e}^{-2\pi\mathrm{j}\frac{k}{n}} \cdot \left(\mathcal{F}_{\frac{n}{2}}\{x_{2i+1}\}_{i=0}^{\frac{n}{2}-1}\right)_k, & k < \frac{n}{2} \\
\left(\mathcal{F}_{\frac{n}{2}}\{x_{2i}\}_{i=0}^{\frac{n}{2}-1}\right)_{k-\frac{n}{2}} + \mathrm{e}^{-2\pi\mathrm{j}\frac{k}{n}} \cdot \left(\mathcal{F}_{\frac{n}{2}}\{x_{2i+1}\}_{i=0}^{\frac{n}{2}-1}\right)_{k-\frac{n}{2}}, & k \geq \frac{n}{2}
\end{cases}
\end{aligned}
$$

The DFT over n-element vectors can be reduced to two DFTs over $n/2$-element vectors plus n multiplications and n additions, leading to $\log_2 n$ rounds and $n \log_2 n$ additions and multiplications overall, compared to n^2 for the equivalent matrix multiplication.

A high-performance FFT implementation in C with many processor-specific optimizations and support for non-power-of-2 sizes is available at `http://www.fftw.org/`.

Efficient real-valued FFT

The symmetry properties of the Fourier transform applied to the discrete Fourier transform $\{X_i\}_{i=0}^{n-1} = \mathcal{F}_n\{x_i\}_{i=0}^{n-1}$ have the form

$$\begin{aligned} \forall i: x_i &= \Re(x_i) &\iff \forall i: X_{n-i} &= X_i^* \\ \forall i: x_i &= \mathrm{j}\cdot\Im(x_i) &\iff \forall i: X_{n-i} &= -X_i^* \end{aligned}$$

These two symmetries, combined with the linearity of the DFT, allows us to calculate two real-valued n-point DFTs

$$\{X_i'\}_{i=0}^{n-1} = \mathcal{F}_n\{x_i'\}_{i=0}^{n-1} \qquad \{X_i''\}_{i=0}^{n-1} = \mathcal{F}_n\{x_i''\}_{i=0}^{n-1}$$

simultaneously in a single complex-valued n-point DFT, by composing its input as

$$x_i = x_i' + \mathrm{j}\cdot x_i''$$

and decomposing its output as

$$X_i' = \frac{1}{2}(X_i + X_{n-i}^*) \qquad X_i'' = \frac{1}{2}(X_i - X_{n-i}^*)$$

To optimize the calculation of a single real-valued FFT, use this trick to calculate the two half-size real-value FFTs that occur in the first round.

Fast complex multiplication

Calculating the product of two complex numbers as

$$(a + \mathrm{j}b) \cdot (c + \mathrm{j}d) = (ac - bd) + \mathrm{j}(ad + bc)$$

involves four (real-valued) multiplications and two additions.

The alternative calculation

$$(a + \mathrm{j}b) \cdot (c + \mathrm{j}d) = (\alpha - \beta) + \mathrm{j}(\alpha + \gamma) \quad \text{with} \quad \begin{array}{rcl} \alpha & = & a(c + d) \\ \beta & = & d(a + b) \\ \gamma & = & c(b - a) \end{array}$$

provides the same result with three multiplications and five additions.

The latter may perform faster on CPUs where multiplications take three or more times longer than additions.

This “Karatsuba multiplication” is most helpful on simpler microcontrollers. Specialized signal-processing CPUs (DSPs) feature 1-clock-cycle multipliers. High-end desktop processors use pipelined multipliers that stall where operations depend on each other.

FFT-based convolution

Calculating the convolution of two finite sequences $\{x_i\}_{i=0}^{m-1}$ and $\{y_i\}_{i=0}^{n-1}$ of lengths m and n via

$$z_i = \sum_{j=\max\{0,i-(n-1)\}}^{\min\{m-1,i\}} x_j \cdot y_{i-j}, \qquad 0 \le i < m+n-1$$

takes mn multiplications.

Can we apply the FFT and the convolution theorem to calculate the convolution faster, in just $O(m \log m + n \log n)$ multiplications?

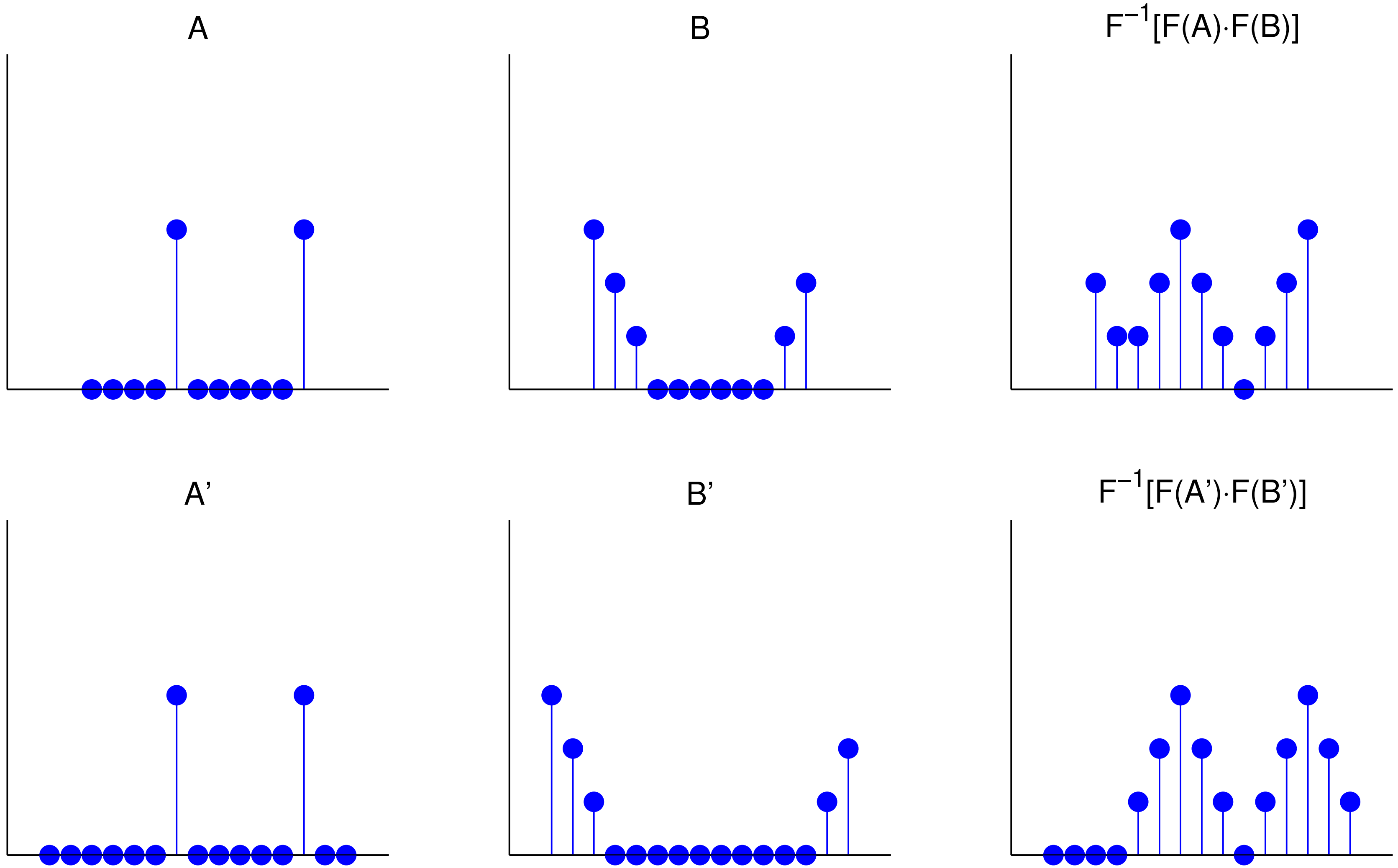
$$\{z_i\} = \mathcal{F}^{-1}\left(\mathcal{F}\{x_i\} \cdot \mathcal{F}\{y_i\}\right)$$

There is obviously no problem if this condition is fulfilled:

$\{x_i\}$ and $\{y_i\}$ are periodic, with equal period lengths

In this case, the fact that the DFT interprets its input as a single period of a periodic signal will do exactly what is needed, and the FFT and inverse FFT can be applied directly as above.

In the general case, measures have to be taken to prevent a wrap-over:

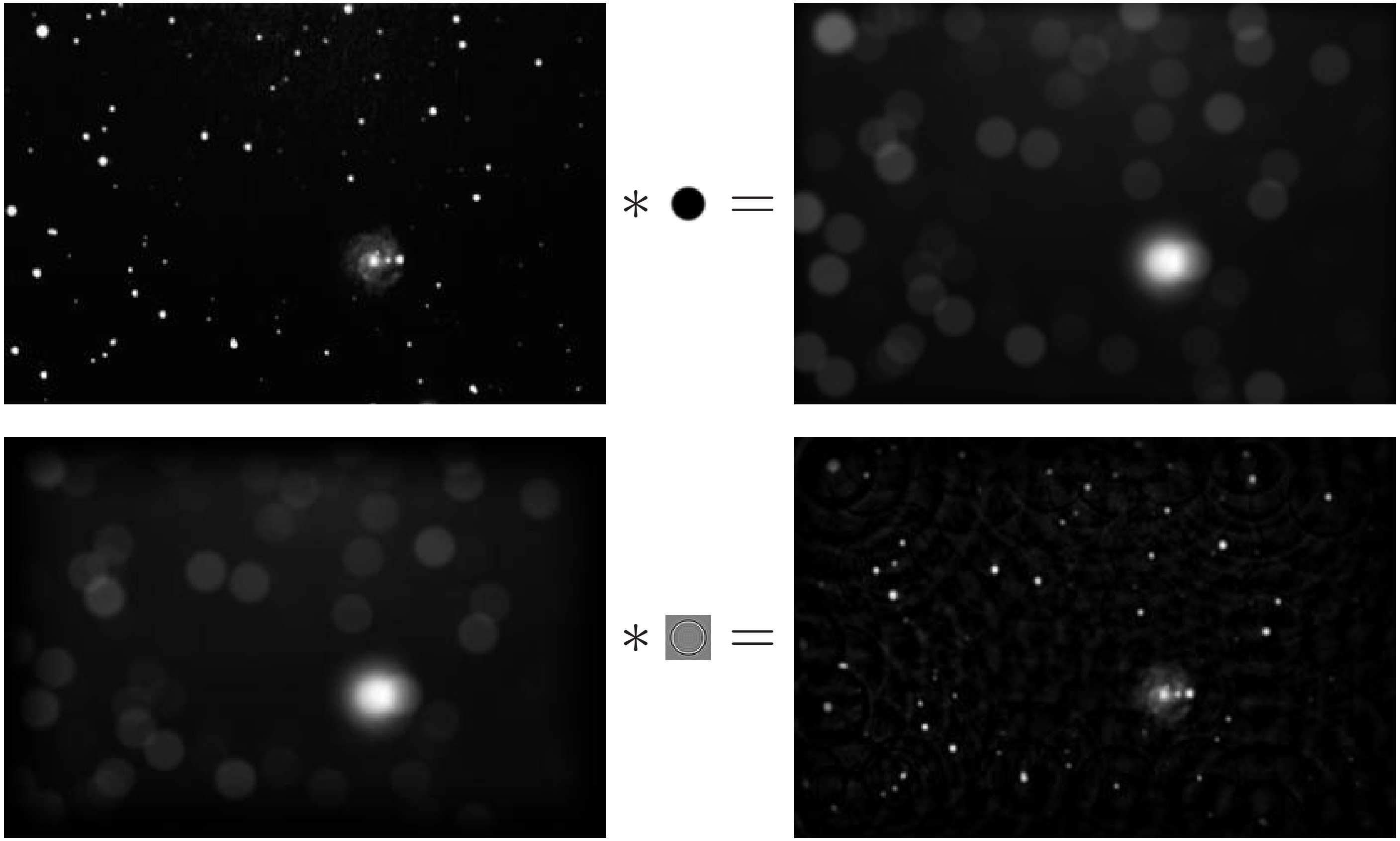


Both sequences are padded with zero values to a length of at least $m+n-1$. This ensures that the start and end of the resulting sequence do not overlap.

Deconvolution

A signal $u(t)$ was distorted by convolution with a known impulse response $h(t)$ (e.g., through a transmission channel or a sensor problem). The "smeared" result $s(t)$ was recorded.

Can we undo the damage and restore (or at least estimate) $u(t)$?



The convolution theorem turns the problem into one of multiplication:

$$\begin{aligned} s(t) &= \int u(t-\tau)\cdot h(\tau)\cdot \mathrm{d}\tau \\ s &= u * h \\ \mathcal{F}\{s\} &= \mathcal{F}\{u\}\cdot\mathcal{F}\{h\} \\ \mathcal{F}\{u\} &= \mathcal{F}\{s\}/\mathcal{F}\{h\} \\ u &= \mathcal{F}^{-1}\{\mathcal{F}\{s\}/\mathcal{F}\{h\}\} \end{aligned}$$

In practice, we also record some noise $n(t)$ (quantization, etc.):

$$c(t) = s(t) + n(t) = \int u(t-\tau)\cdot h(\tau)\cdot \mathrm{d}\tau + n(t)$$

Problem – At frequencies f where $\mathcal{F}\{h\}(f)$ approaches zero, the noise will be amplified (potentially enormously) during deconvolution:

$$\tilde{u} = \mathcal{F}^{-1}\{\mathcal{F}\{c\}/\mathcal{F}\{h\}\} = u + \mathcal{F}^{-1}\{\mathcal{F}\{n\}/\mathcal{F}\{h\}\}$$

Typical workarounds:

- $\rightarrow$ Modify the Fourier transform of the impulse response, such that $|\mathcal{F}\{h\}(f)| > \epsilon$ for some experimentally chosen threshold ϵ.
- $\rightarrow$ If estimates of the signal spectrum $|\mathcal{F}\{s\}(f)|$ and the noise spectrum $|\mathcal{F}\{n\}(f)|$ can be obtained, then we can apply the "Wiener filter" ("optimal filter")

$$W(f) = \frac{|\mathcal{F}\{s\}(f)|^2}{|\mathcal{F}\{s\}(f)|^2 + |\mathcal{F}\{n\}(f)|^2}$$

before deconvolution:

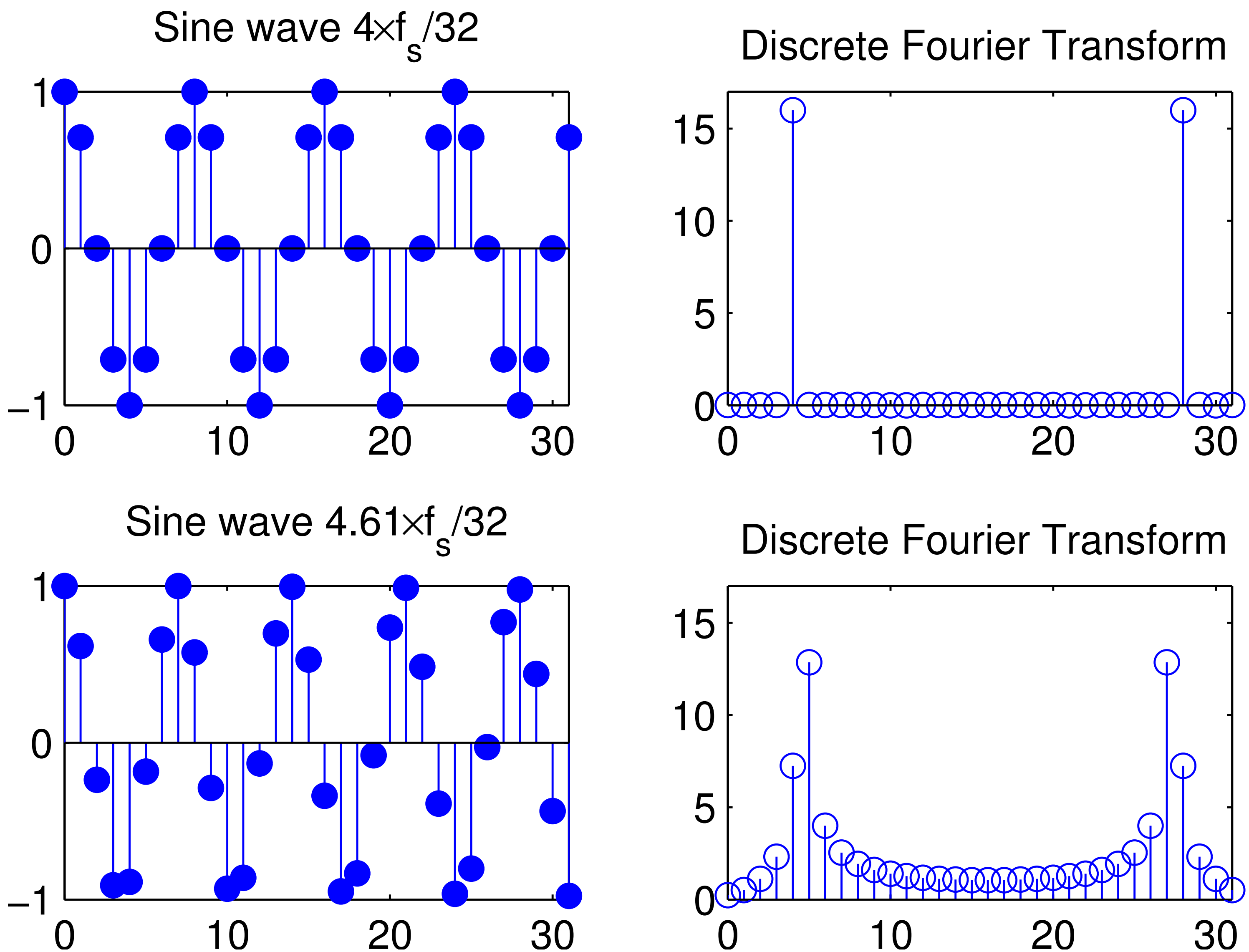
$$\tilde{u} = \mathcal{F}^{-1}\{W \cdot \mathcal{F}\{c\}/\mathcal{F}\{h\}\}$$

Exercise 13 Use MATLAB to deconvolve the blurred stars from slide 28.

The files `stars-blurred.png` with the blurred-stars image and `stars-psf.png` with the impulse response (point-spread function) are available on the course-material web page. You may find the MATLAB functions `imread`, `double`, `imagesc`, `circshift`, `fft2`, `ifft2` of use.

Try different ways to control the noise (see above) and distortions near the margins (windowing). [The MATLAB image processing toolbox provides ready-made "professional" functions `deconvwnr`, `deconvreg`, `deconvlucy`, `edgetaper`, for such tasks. Do not use these, except perhaps to compare their outputs with the results of your own attempts.]

Spectral estimation



We introduced the DFT as a special case of the continuous Fourier transform, where the input is sampled *and periodic*.

If the input is sampled, but not periodic, the DFT can still be used to calculate an approximation of the Fourier transform of the original continuous signal. However, there are two effects to consider. They are particularly visible when analysing pure sine waves.

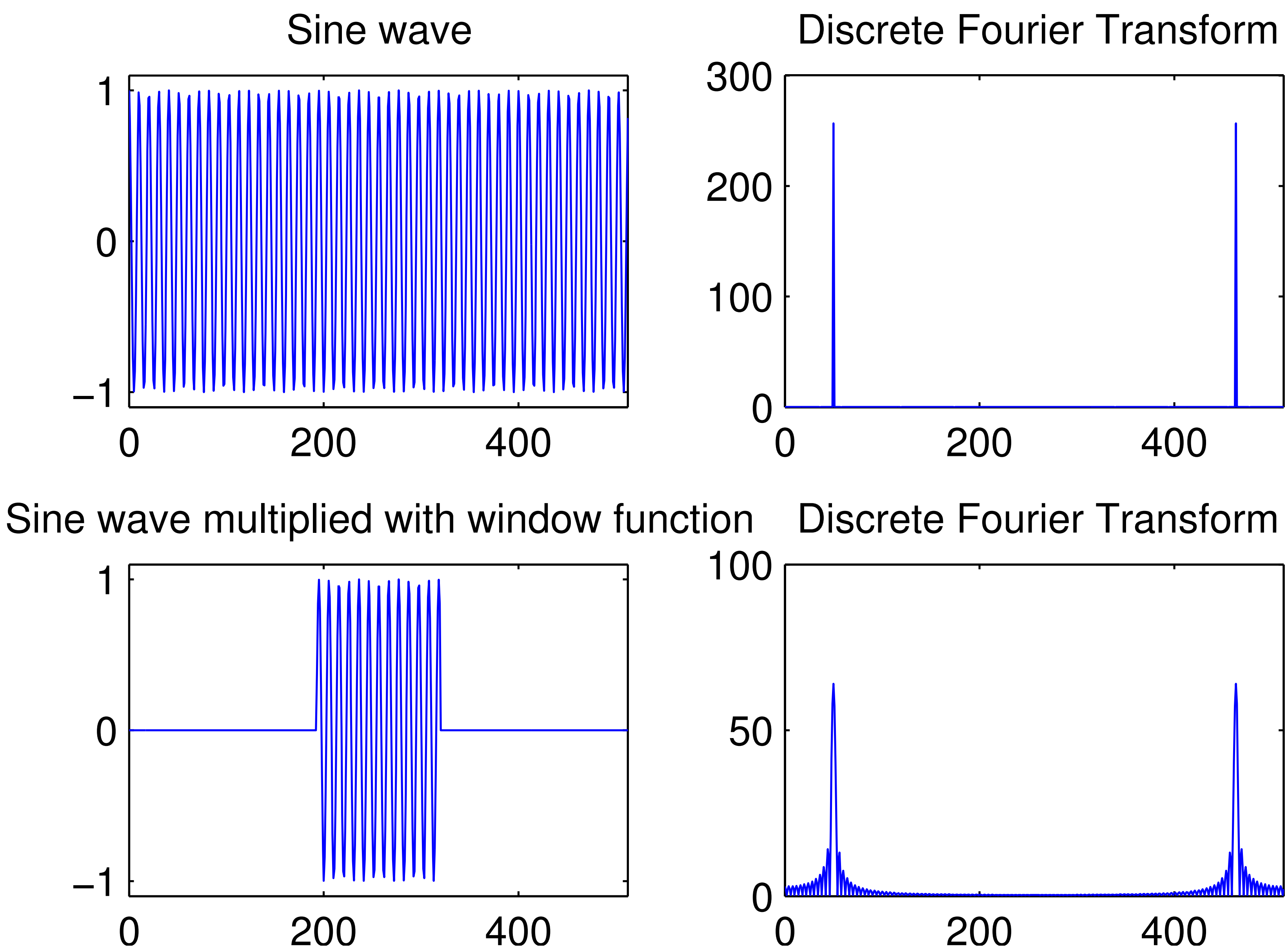
Sine waves whose frequency is a multiple of the base frequency (f_{s}/n) of the DFT are identical to their periodic extension beyond the size of the DFT. They are, therefore, represented exactly by a single sharp peak in the DFT. All their energy falls into one single frequency "bin" in the DFT result.

Sine waves with other frequencies, which do not match exactly one of the output frequency bins of the DFT, are still represented by a peak at the output bin that represents the nearest integer multiple of the DFT's base frequency. However, such a peak is distorted in two ways:

$\longrightarrow$ Its amplitude is lower (down to 63.7%).

$\longrightarrow$ Much signal energy has "leaked" to other frequencies.

Windowing


Sine wave
Discrete Fourier Transform
Sine wave multiplied with window function
Discrete Fourier Transform
1
0
−1
0
200
400
300
200
100
0
0
200
400
1
0
−1
0
200
400
100
50
0
0
200
400

The reason for the leakage and scalloping losses is easy to visualize with the help of the convolution theorem:

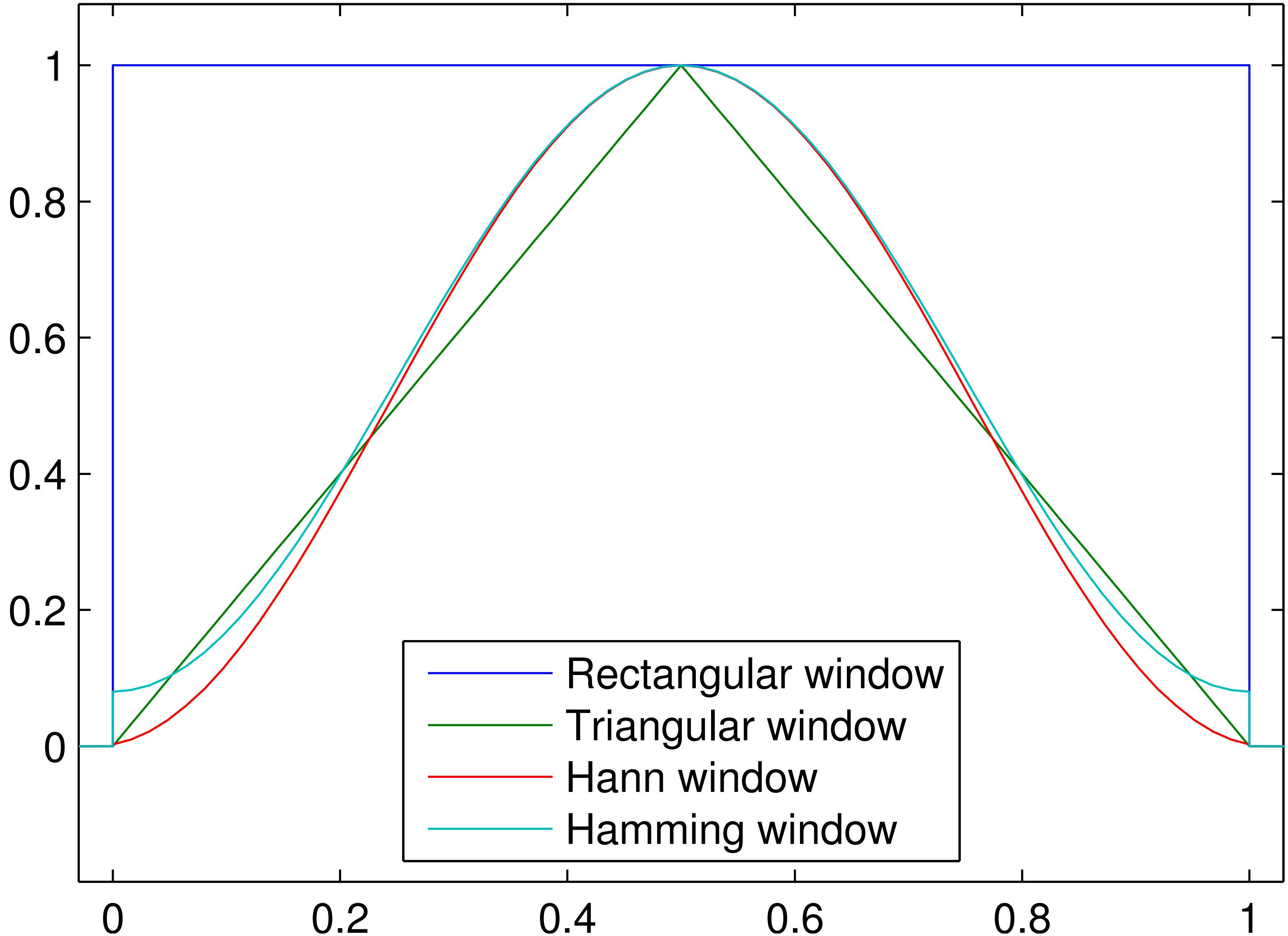
The operation of cutting a sequence of the size of the DFT input vector out of a longer original signal (the one whose continuous Fourier spectrum we try to estimate) is equivalent to multiplying this signal with a rectangular function. This destroys all information and continuity outside the "window" that is fed into the DFT.

Multiplication with a rectangular window of length T in the time domain is equivalent to convolution with $\sin(\pi f T)/(\pi f T)$ in the frequency domain.

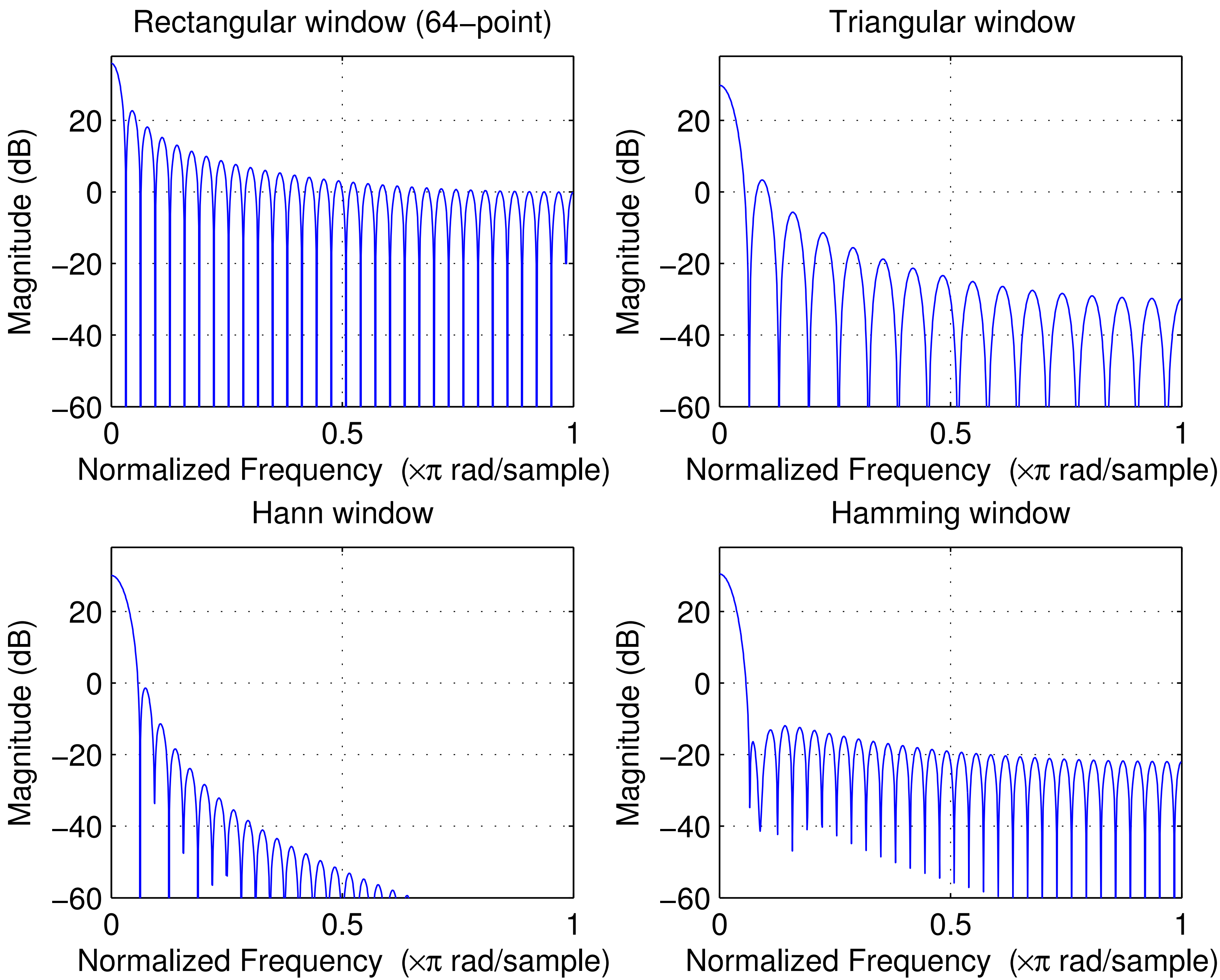
The subsequent interpretation of this window as a periodic sequence by the DFT leads to sampling of this convolution result (sampling meaning multiplication with a Dirac comb whose impulses are spaced f_s/n apart).

Where the window length was an exact multiple of the original signal period, sampling of the $\sin(\pi f T)/(\pi f T)$ curve leads to a single Dirac pulse, and the windowing causes no distortion. In all other cases, the effects of the convolution become visible in the frequency domain as leakage and scalloping losses.

Some better window functions



All these functions are 0 outside the interval [0,1].



Rectangular window (64−point)
Triangular window
Hann window
Hamming window
Magnitude (dB)
Normalized Frequency (×π rad/sample)
20
0
−20
−40
−60
0
0.5
1

Numerous alternatives to the rectangular window have been proposed that reduce leakage and scalloping in spectral estimation. These are vectors multiplied element-wise with the input vector before applying the DFT to it. They all force the signal amplitude smoothly down to zero at the edge of the window, thereby avoiding the introduction of sharp jumps in the signal when it is extended periodically by the DFT.

Three examples of such window vectors $\{w_i\}_{i=0}^{n-1}$ are:

Triangular window (Bartlett window):

$$w_i = 1 - \left|1 - \frac{i}{n/2}\right|$$

Hann window (raised-cosine window, Hanning window):

$$w_i = 0.5 - 0.5 \times \cos\left(2\pi\frac{i}{n-1}\right)$$

Hamming window:

$$w_i = 0.54 - 0.46 \times \cos\left(2\pi\frac{i}{n-1}\right)$$

Zero padding increases DFT resolution

The two figures below show two spectra of the 16-element sequence

$$s_i = \cos(2\pi \cdot 3i/16) + \cos(2\pi \cdot 4i/16), \qquad i \in \{0, \ldots, 15\}.$$

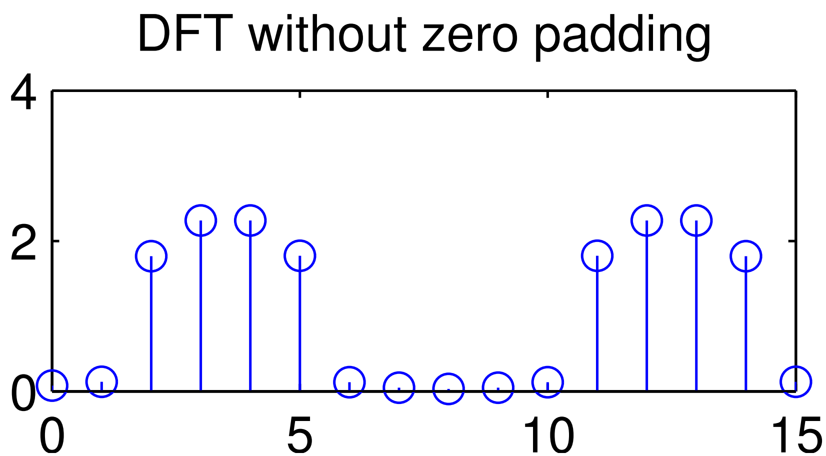
The left plot shows the DFT of the windowed sequence

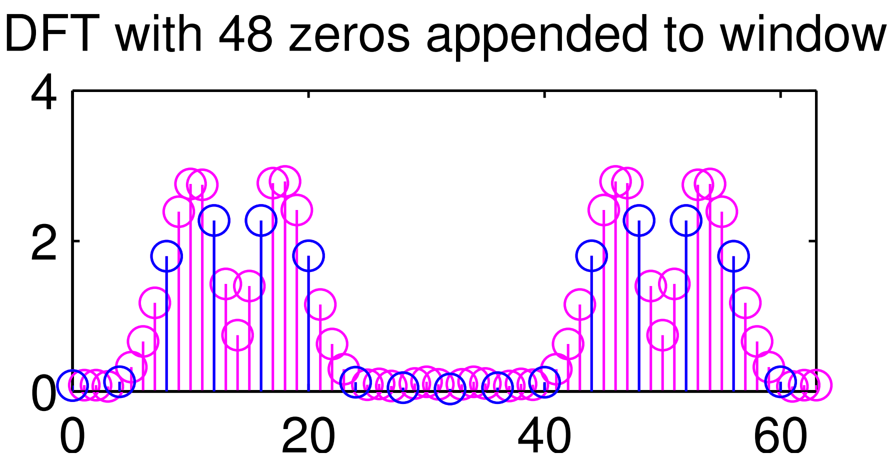
$$x_i = s_i \cdot w_i, \qquad i \in \{0, \ldots, 15\}$$

and the right plot shows the DFT of the zero-padded windowed sequence

$$x'_i = \begin{cases} s_i \cdot w_i, & i \in \{0, \ldots, 15\} \\ 0, & i \in \{16, \ldots, 63\} \end{cases}$$

where $w_i = 0.54 - 0.46 \times \cos\left(2\pi i/15\right)$ is the Hamming window.





Applying the discrete Fourier transform to an n-element long real-valued sequence leads to a spectrum consisting of only $n/2+1$ discrete frequencies.

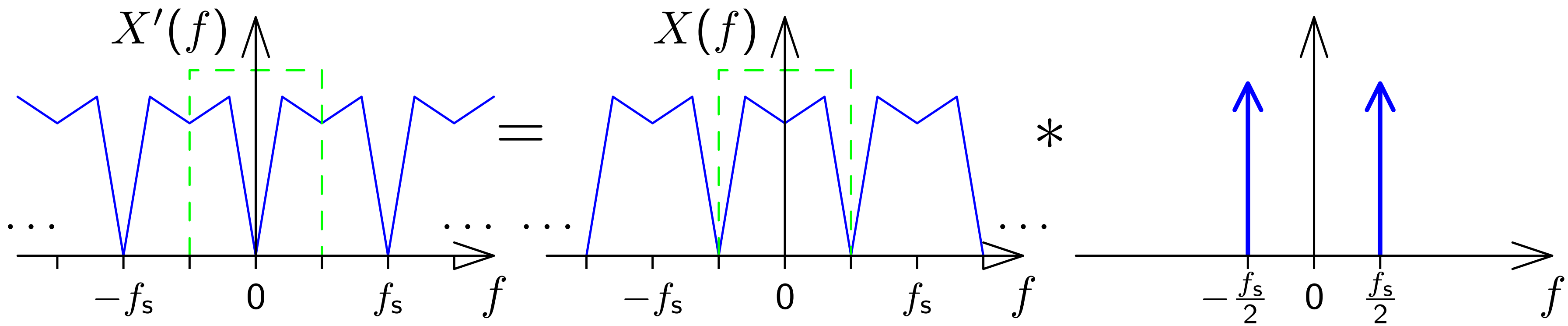
Since the resulting spectrum has already been distorted by multiplying the (hypothetically longer) signal with a windowing function that limits its length to n non-zero values and forces the waveform smoothly down to zero at the window boundaries, appending further zeros outside the window will not distort the signal further.

The frequency resolution of the DFT is the sampling frequency divided by the block size of the DFT. Zero padding can therefore be used to increase the frequency resolution of the DFT.

Note that zero padding does *not* add any additional information to the signal. The spectrum has already been "low-pass filtered" by being convolved with the spectrum of the windowing function. Zero padding in the time domain merely samples this spectrum blurred by the windowing step at a higher resolution, thereby making it easier to visually distinguish spectral lines and to locate their peak more precisely.

Frequency inversion

In order to turn the spectrum $X(f)$ of a real-valued signal x_i sampled at f_s into an inverted spectrum $X'(f) = X(f_\mathsf{s}/2 - f)$, we merely have to shift the periodic spectrum by $f_\mathsf{s}/2$:



This can be accomplished by multiplying the sampled sequence x_i with $y_i = \cos \pi f_\mathsf{s} t = \cos \pi i$, which is nothing but multiplication with the sequence

$$\dots, 1, -1, 1, -1, 1, -1, 1, -1, \dots$$

So in order to design a discrete high-pass filter that attenuates all frequencies f outside the range $f_\mathsf{c} < |f| < f_\mathsf{s}/2$, we merely have to design a low-pass filter that attenuates all frequencies outside the range $-f_\mathsf{c} < f < f_\mathsf{c}$, and then multiply every second value of its impulse response with -1.