Signal Processing Story Sheet

by Titus K. Roesler (tkr@mit.edu)

(Last updated: May 14, 2026)

Lecture and Recitation Retrospective

Fourier series: Frequency representations for periodic signals

- **02/03/2026:** Fourier series for continuous-time signals
- **02/05/2026:** Using symmetry to simplify Fourier series calculations
- **02/10/2026:** Fourier series — now with complex exponentials!
- **02/12/2026:** Sampling and aliasing (i.e., discretizing time)
- **02/17/2026:** Presidents' Day
- **02/19/2026:** Fourier series for discrete-time signals

Fourier transforms: Frequency representations for all signals[1]

- **02/24/2026:** Blizzard of 2026
- **02/26/2026:** Fourier transform for continuous-time signals
- **03/03/2026:** Quiz #1
- **03/05/2026:** Fourier transform for discrete-time signals

Systems: Analyzing and designing systems for processing signals

- **03/10/2026:** Systems (e.g., linearity, time-invariance, diff. equations)
- **03/12/2026:** Unit sample/impulse response and convolution
- **03/17/2026:** Frequency response and filtering
- **03/19/2026:** Communication systems (e.g., Fourier transform duality)

Discrete Fourier transform: Good for numerical computation

- **03/31/2026:** Relations among discrete-time Fourier representations
- **04/02/2026:** Frequency resolution, circular convolution, impulse trains
- **04/07/2026:** Fast Fourier transform (FFT) algorithms
- **04/09/2026:** Short-time Fourier transform (i.e., sequence of transforms)
- **04/14/2026:** Speech processing (e.g., source-filter model of speech)
- **04/16/2026:** Quiz #2
- **04/21/2026:** Quiz #2 retrospective

Multidimensional signal processing

- **04/23/2026:** Two-dimensional Fourier transforms
- **04/28/2026:** Two-dimensional Fourier transforms and convolution
- **04/30/2026:** Image processing with the discrete Fourier transform

---

[1] Not all, but a lot.

- **05/05/2026:** Data compression with the discrete cosine transform
- **05/07/2026:** Fourier transforms and magnetic resonance imaging
- **05/12/2026:** Synthetic aperture optics (i.e., Fourier optics)
- **05/15/2026:** Final examination

Mathematics for Engineers

Dimensional Analysis

$$T_0 \text{ (seconds)} \times f_s \text{ (samples / second)} = N_0 \text{ (samples)}$$

$$\omega_0 \text{ (radians / second)} \div f_s \text{ (samples / second)} = \Omega_0 \text{ (radians / sample)}$$

CT cyclical frequency $f = 1/T$	cycles per second or hertz (Hz)
CT angular frequency $\omega = 2\pi f = 2\pi/T$	radians per second
DT cyclical frequency $F = f\Delta = f/f_s = 1/f_sT = 1/N$	cycles per sample
DT angular frequency $\Omega = 2\pi F = 2\pi f\Delta = 2\pi/N$	radians per sample

Geometric Series

$$\sum_{n=0}^{N-1} r^n = \frac{1-r^N}{1-r} \text{ for } |r| < 1 \qquad \sum_{n=0}^{\infty} r^n = \frac{1}{1-r} \text{ for } |r| < 1$$

Binomial Theorem

$$(\alpha+\beta)^N = \binom{N}{0}\alpha^N + \binom{N}{1}\alpha^{N-1}\beta + \cdots + \binom{N}{N-1}\alpha\beta^{N-1} + \binom{N}{N}\beta^N$$

$$\binom{n}{k} \triangleq \frac{n!}{k!\,(n-k)!} \text{ where } n! \triangleq (n)(n-1)(n-2)\cdots(3)(2)(1)$$

Complex Variables

Imaginary Unit

$$j = \sqrt{-1} \qquad j^2 = -1 \qquad j^3 = -j \qquad j^4 = 1$$

Euler's Formula

$$e^{j\theta} = \cos(\theta) + j\,\sin(\theta) \qquad \cos(\theta) = \frac{e^{j\theta}+e^{-j\theta}}{2} \qquad \sin(\theta) = \frac{e^{j\theta}-e^{-j\theta}}{2j}$$

Complex Variables in Rectangular and Polar Coordinates

$$z = \mathrm{Re}\{z\} + j\,\mathrm{Im}\{z\} = |z|\,e^{j\angle z} \qquad |z| = \sqrt{\mathrm{Re}\{z\}^2 + \mathrm{Im}\{z\}^2} \qquad \tan(\angle z) = \frac{\mathrm{Im}\{z\}}{\mathrm{Re}\{z\}}$$

Fourier Series and Transforms

Continuous-Time Fourier Series (CTFS)

$$X[k] = \frac{1}{T}\int_{t_0}^{t_0+T} x(t)e^{-jk\frac{2\pi}{T}t}dt \qquad x(t) = \sum_{k=-\infty}^{\infty} X[k]e^{jk\frac{2\pi}{T}t}$$

Discrete-Time Fourier Series (DTFS)

$$X[k] = \frac{1}{N}\sum_{n=n_0}^{n_0+N-1} x[n]e^{-jk\frac{2\pi}{N}n} \qquad x[n] = \sum_{k=k_0+N}^{k_0+N-1} X[k]e^{jk\frac{2\pi}{N}n}$$

Continuous-Time Fourier Transform (CTFT)

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \qquad x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega$$

Discrete-Time Fourier Transform (DTFT)

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

Discrete Fourier Transform (DFT)

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

Two-Dimensional Discrete Fourier Transform (2D DFT)

$$X[k_r,k_c] = \frac{1}{RC}\sum_{r=0}^{R-1}\sum_{c=0}^{C-1} x[r,c]e^{-j(k_r\frac{2\pi}{R}r+k_c\frac{2\pi}{C}c)} \qquad x[r,c] = \sum_{k_r=0}^{R-1}\sum_{k_c=0}^{C-1} X[k_r,k_c]e^{j(k_r\frac{2\pi}{R}r+k_c\frac{2\pi}{C}c)}$$

Discrete Cosine Transform (DCT)

$$X_C[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\cos\left(\frac{\pi k}{N}\left(n+\frac{1}{2}\right)\right) \qquad x[n] = X_C[0] + 2\sum_{k=1}^{N-1} X_C[k]\cos\left(\frac{\pi k}{N}\left(n+\frac{1}{2}\right)\right)$$

CTFS Pairs (period T)

$\delta(t-t_0) \iff \frac{1}{T}e^{-jk\frac{2\pi}{T}t_0}$
$e^{jk_0\frac{2\pi}{T}t} \iff \delta[k-k_0]$

DTFS Pairs (period N)

$\delta[n-n_0] \iff \frac{1}{N}e^{-jk\frac{2\pi}{N}n_0}$
$e^{jk_0\frac{2\pi}{N}n} \iff \delta[k-k_0]$

CTFT Pairs

$\delta(t-t_0) \iff e^{-j\omega t_0}$
$e^{j\omega_0 t} \iff 2\pi\delta(\omega-\omega_0)$
$e^{-\sigma t}u(t) \iff 1/(\sigma+j\omega)$ for $\sigma > 0$

DTFT Pairs

$\delta[n-n_0] \iff e^{-j\Omega n_0}$
$e^{j\Omega_0 n} \iff 2\pi\delta(\Omega-\Omega_0)$
$r^n u[n] \iff 1/(1-re^{-j\Omega})$ for $|r| < 1$

DFT Pairs (size N)

$\delta[n-n_0] \iff \frac{1}{N}e^{-jk\frac{2\pi}{N}n_0}$
$e^{jk_0\frac{2\pi}{N}n} \iff \delta[k-k_0]$

2D DFT Pairs (size $R \times C$)

$\delta[r-r_0, c-c_0] \iff \frac{1}{RC}e^{-jk_r\frac{2\pi}{R}r_0}e^{-jk_r\frac{2\pi}{C}c_0}$
$\delta[r-r_0] \iff \frac{1}{R}e^{-jk_r\frac{2\pi}{R}r_0}\delta[k_c]$
$\delta[c-c_0] \iff \frac{1}{C}e^{-jk_c\frac{2\pi}{C}c_0}\delta[k_r]$
$e^{j(k_r'\frac{2\pi}{R}r+k_c'\frac{2\pi}{C}c)} \iff \delta[k_r-k_r']\delta[k_c-k_c']$
$e^{jk_r'\frac{2\pi}{R}r} \iff \delta[k_r-k_r']\delta[k_c]$
$e^{jk_c'\frac{2\pi}{C}c} \iff \delta[k_r]\delta[k_c-k_c']$

CTFT Properties

$x(t-t_0) \iff e^{-j\omega t_0}X(\omega)$
$x(t)e^{j\omega_0 t} \iff X(\omega-\omega_0)$
$x(-t) \iff X(-\omega)$
$x(\alpha t) \iff |\alpha|^{-1}X(\alpha^{-1}\omega)$
$dx(t)/dt \iff j\omega X(\omega)$
$t\,x(t) \iff j\,dX(\omega)/d\omega$
$(x_1 * x_2)(t) \iff X_1(\omega)X_2(\omega)$
$x_1(t)x_2(t) \iff \frac{1}{2\pi}(X_1 * X_2)(\omega)$

DTFT Properties

$x[n-n_0] \iff e^{-j\Omega n_0}X(\Omega)$
$x[n]e^{j\Omega_0 n} \iff X(\Omega-\Omega_0)$
$x[-n] \iff X(-\Omega)$
$x[\alpha n] \iff X(\alpha^{-1}\Omega)$
$n\,x[n] \iff j\,dX(\Omega)/d\Omega$
$(x_1 * x_2)[n] \iff X_1(\Omega)X_2(\Omega)$
$x_1[n]x_2[n] \iff \frac{1}{2\pi}(X_1 * X_2)(\Omega)$

DFT Properties

$x[n-n_0] \iff e^{-jk\frac{2\pi}{N}n_0}X[k]$
$x[n]e^{jk_0\frac{2\pi}{N}n} \iff X[k-k_0]$
$x[-n] \iff X[-k]$
$x[\alpha n] \iff X[\alpha^{-1}k]$
$\frac{1}{N}(x_1 \circledast x_2)[n] \iff X_1[k]X_2[k]$
$x_1[n]x_2[n] \iff (X_1 \circledast X_2)[k]$

2D DFT Properties

Generalize from 1D to 2D.

$x[r-r_0, c-c_0] \iff e^{-j(k_r\frac{2\pi}{R}r_0+k_c\frac{2\pi}{C}c_0)}X[k_r,k_c]$
$x[n]e^{j(k_r'\frac{2\pi}{R}r+k_c'\frac{2\pi}{C}c)} \iff X[k_r-k_r', k_c-k_c']$
$x[-r,-c] \iff X[-k_r,-k_c]$
$x[\alpha r, \beta c] \iff X[\alpha^{-1}k_r, \beta^{-1}k_c]$
$\frac{1}{RC}(x_1 \circledast x_2)[r,c] \iff X_1[k_r,k_c]X_2[k_r,k_c]$
$x_1[r,c]x_2[r,c] \iff (X_1 \circledast X_2)[k_r,k_c]$

Sampling and Aliasing

Sampling

Sampling a continuous-time signal $x(t)$ yields a discrete-time signal $x[n]$. The *sampling interval* or *sampling period* Δ specifies the time, in seconds, between successive samples. Equivalently, if one samples $x(t)$ every Δ seconds, the sampling rate is $f_s = 1/\Delta$ samples per second.

$$x[n] \equiv x(n\Delta)$$

Aliasing

Sampling leads to throwing away information. How much information are we able to throw away without misrepresenting the underlying continuous-time signal? The Nyquist-Shannon sampling theorem gives us a definite answer. If a continuous-time signal $x(t)$ is *bandlimited* to ω_c in frequency (i.e., $X(\omega) = 0$ for $|\omega| > \omega_c$), then $2\omega_c$ is the minimum sampling rate that avoids aliasing. *Aliasing* is the unwanted distortion of a signal's spectrum due to the aforementioned periodic replication, which is caused by *undersampling*. Check out https://www.analog.com/en/resources/interactive-design-tools/frequency-folding-tool.html, which visualizes the aliasing that occurs when a continuous-time signal is sampled.

Linear, Time-Invariant Systems

Additivity

If you add the inputs, you add the corresponding outputs.

$$x_1(t) \to \boxed{\mathcal{H}} \to y_1(t) \text{ and } x_2(t) \to \boxed{\mathcal{H}} \to y_2(t) \implies x_1(t) + x_2(t) \to \boxed{\mathcal{H}} \to y_1(t) + y_2(t)$$

$$x_1[n] \to \boxed{\mathcal{H}} \to y_1[n] \text{ and } x_2[n] \to \boxed{\mathcal{H}} \to y_2[n] \implies x_1[n] + x_2[n] \to \boxed{\mathcal{H}} \to y_1[n] + y_2[n]$$

Homogeneity

Scaling the input scales the output correspondingly.

$$x(t) \to \boxed{\mathcal{H}} \to y(t) \implies \alpha x(t) \to \boxed{\mathcal{H}} \to \alpha y(t) \text{ for some constant } \alpha$$

$$x[n] \to \boxed{\mathcal{H}} \to y[n] \implies \alpha x[n] \to \boxed{\mathcal{H}} \to \alpha y[n] \text{ for some constant } \alpha$$

Linearity

A linear combination of inputs yields a linear combination of the respective outputs.

$$\alpha_1 x_1(t) + \alpha_2 x_2(t) + \cdots \rightarrow \boxed{\mathcal{H}} \rightarrow \alpha_1 y_1(t) + \alpha_2 y_2(t) + \cdots$$

$$\alpha_1 x_1[n] + \alpha_2 x_2[n] + \cdots \rightarrow \boxed{\mathcal{H}} \rightarrow \alpha_1 y_1[n] + \alpha_2 y_2[n] + \cdots$$

Together, additivity and homogeneity imply linearity.

Time-Invariance

Delaying or advancing the input delays or advances the output correspondingly.

$$x(t) \rightarrow \boxed{\mathcal{H}} \rightarrow y(t) \implies x(t-t_0) \rightarrow \boxed{\mathcal{H}} \rightarrow y(t-t_0) \text{ for any } t_0$$

$$x[n] \rightarrow \boxed{\mathcal{H}} \rightarrow y[n] \implies x[n-n_0] \rightarrow \boxed{\mathcal{H}} \rightarrow y[n-n_0] \text{ for any integer } n_0$$

Differential Equation (CT) / Difference Equation (DT)

Derivatives and integrals play an analogous role to delays and advances.

$$\sum_k b_k \frac{d^k y(t)}{dt^k} = \sum_k a_k \frac{d^k x(t)}{dt^k} \qquad \left(\text{e.g., } m\frac{d^2 y(t)}{dt^2} + b\frac{dy(t)}{dt} + ky(t) = x(t)\right)$$

$$\sum_k a_k y[n-k] = \sum_k b_k x[n-k] \qquad \left(\text{e.g., } y[n] = \alpha y[n-1] + x[n]\right)$$

Impulse Response (CT) / Unit-Sample Response (DT)

The impulse response (CT) or unit-sample response (DT) enable us to characterize an LTI system by a single signal — the system's response to an impulse $\delta(t)$ or unit sample $\delta[n]$.

$$\delta(t) \rightarrow \boxed{\mathcal{H}} \rightarrow h(t) = \int_{-\infty}^{\infty} h(\tau)\delta(t-\tau)d\tau$$

$$\delta[n] \rightarrow \boxed{\mathcal{H}} \rightarrow h[n] = \sum_{k=-\infty}^{\infty} h[k]\delta[n-k]$$

The time-domain output of an LTI system is given by the convolution of the input with the system's impulse or unit-sample response.

$$x(t) \rightarrow \boxed{\mathcal{H}} \rightarrow y(t) = (x * h)(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$$

$$x[n] \rightarrow \boxed{\mathcal{H}} \rightarrow y[n] = (x * h)[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

Frequency Response

The frequency response is a frequency-domain characterization of a system.

Complex exponentials are eigenfunctions of linear time-invariant systems. That is, if the input to an LTI system is a linear combination of complex exponentials, the output of the system is a linear combination of complex exponentials, each scaled by a complex constant — the corresponding eigenvalues. The frequency response tells us these eigenvalues.

Moreover, Fourier transforms enable us to represent signals (for which Fourier transforms exist) as linear combinations of complex exponentials. By expanding a input to an LTI system in the eigenbasis — a linear combination of complex exponentials — we are readily able to compute the system's output.

$$e^{j\omega t} \rightarrow \boxed{\mathcal{H}} \rightarrow H(\omega)e^{j\omega t} = |H(\omega)|e^{j(\omega t + \angle H(\omega))}$$

$$e^{j\Omega n} \rightarrow \boxed{\mathcal{H}} \rightarrow H(\Omega)e^{j\Omega n} = |H(\Omega)|e^{j(\Omega n + \angle H(\Omega))}$$

$$X(\omega) \rightarrow \boxed{\mathcal{H}} \rightarrow Y(\omega) = H(\omega)X(\omega)$$

$$X(\Omega) \rightarrow \boxed{\mathcal{H}} \rightarrow Y(\Omega) = H(\Omega)X(\Omega)$$

This perspective enables us to characterize an LTI system as a *filter* that shapes a signal's spectrum.

Equivalent Representations of LTI Systems

Consider the LTI system represented by the following difference equation.

$$\sum_{k=-\infty}^{\infty} a_k y[n-k] = \sum_{k=-\infty}^{\infty} b_k x[n-k]$$

Computing the DTFT yields the frequency response.

$$\sum_{k=-\infty}^{\infty} a_k e^{-jk\Omega} Y(\Omega) = \sum_{k=-\infty}^{\infty} b_k e^{-jk\Omega} X(\Omega) \iff H(\Omega) = \frac{Y(\Omega)}{X(\Omega)} = \frac{\sum_{k=-\infty}^{\infty} b_k e^{-jk\Omega}}{\sum_{k=-\infty}^{\infty} a_k e^{-jk\Omega}}$$

The impulse response is given by the inverse DTFT of the frequency response. $H(\Omega)$ is a rational polynomial that one can decompose into a sum of simpler rational polynomials (e.g., via partial fractions) with simpler inverse DTFTs.

$$h[n] = \frac{1}{2\pi}\int_{2\pi} H(\Omega)e^{j\Omega n}d\Omega$$

e.g., $h[n] = \frac{1}{2\pi}\int_{2\pi}\left(\frac{1}{1-\frac{1}{2}e^{-j\Omega}} + \frac{1}{1+\frac{1}{3}e^{-j\Omega}}\right)e^{j\Omega n}d\Omega = \left(\frac{1}{2}\right)^n u[n] + \left(-\frac{1}{3}\right)^n u[n]$ (perhaps)

Convolution

Convolution

To perform a convolution is to *convolve* two functions or sequences. To *convolute* is to make something more complicated.

$(x * h)(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$

$(h * x)(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$

$(x * h)[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$

$(h * x)[n] = \sum_{m=-\infty}^{\infty} h[m]x[n-m]$

Associativity of Convolution

$\big(x * (h_1 * h_2)\big)(t) = \big((x * h_1) * h_2\big)(t)$

$\big(x * (h_1 * h_2)\big)[n] = \big((x * h_1) * h_2\big)[n]$

Commutativity of Convolution

$(x * h)(t) = (h * x)(t)$

$(x * h)[n] = (h * x)[n]$

Distributivity of Convolution

$\big(x * (h_1 + h_2)\big)(t) = (x * h_1)[n] + (x * h_2)(t)$

$\big(x * (h_1 + h_2)\big)[n] = (x * h_1)[n] + (x * h_2)[n]$

Circular Convolution

Circular convolution inherits the properties above.

$(x \circledast h)[n] = \sum_{m=0}^{N-1} x[m]h\big[\big((n-m) \bmod N\big)\big]$

$(h \circledast x)[n] = \sum_{m=0}^{N-1} h[m]x\big[\big((n-m) \bmod N\big)\big]$

You can perform circular convolution as follows.

- Perform linear convolution.
- Wrap the result into a length-N interval. (This last step is *time-aliasing* — analogous to the more familiar frequency-aliasing.)
- Periodically extend every N samples.

Multiplication of N-point DFTs corresponds to time-domain circular convolution.

$\frac{1}{N}(x \circledast h)[n] \iff X_N[k]H_N[k]$

Convolution with Unit Impulses and Unit Samples

$h(t) = \delta(t) \longrightarrow (x * h)(t) = x(t)$

$h[n] = \delta[n] \longrightarrow (x * h)[n] = x[n]$

$h(t) = \delta(t - t_0)$

$(x * h)(t) = x(t - t_0)$

$h[n] = \delta[n - n_0]$

$(x * h)[n] = x[n - n_0]$

$h(t) = \sum_k \alpha_k \delta(t - t_k)$

$(x * h)(t) = \sum_k \alpha_k x(t - t_k)$

$h[n] = \sum_k \alpha_k \delta[n - k]$

$(x * h)[n] = \sum_k \alpha_k x[n - k]$

Using a Table for Convolution

$$x[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2]$$

$$h[n] = \delta[n] + 2\delta[n-2] + \delta[n-4]$$

$$(x * h)[n] = x[n] + 2x[n-2] + x[n-4]$$

n	0	1	2	3	4	5	6	7
$x[n]$	1	2	3					
$2x[n-2]$			2	4	6			
$x[n-4]$					1	2	3	
$(x * h)[n]$	1	2	5	4	7	2	3	

Using a Table for Circular Convolution

$$4X_4[k]H_4[k] \iff (x \circledast h)[n]$$

n	0	1	2	3	4	5	6	7
$(x * h)[n]$	1	2	5	4	7	2	3	
$(x \circledast h)[n]$	8	4	8	4	8	4	8	4

Discrete Fourier Transform

Motivation for the Discrete Fourier Transform

We may represent periodic discrete-time signals as a sum of harmonically-related (complex) sinusoids via DTFS. Periodicity is a rather restrictive property, however, as many signals of interest are not periodic. (For a signal to be periodic, it must repeat forever — it must be infinitely long!) The DTFT is a Fourier representation for aperiodic discrete-time signals, but the definition involves an infinite summation, and the spectrum — a function of the continuous variable Ω — is continuous. In search of a discrete-time Fourier transform-like representation that is amenable for digital computation, we turned to the DFT.

DFT: Discrete in Time, Discrete in Frequency

- finite-length signals ($x_w[n] = x[n]w[n]$)
- discrete in time (indexed by integer n)
- discrete in frequency (indexed by integer k)

Relation to DTFS and DTFT

The DFT is equivalent to the DTFS of an N-periodic extension of $x_w[n] = x[n]w[n]$. From this perspective, sharp discontinuities in the periodic extension of $x_w[n]$ lead to spurious high-frequency content spread across the spectrum of $x_w[n]$.

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

The DFT returns N (scaled) equally-spaced samples of the DTFT over the interval $\Omega \in [0, 2\pi]$. Increasing the DFT length N yields more samples of the DTFT, which are spaced more closely together.

$$X[k] = \frac{1}{N}X_w\left(\frac{2\pi k}{N}\right)$$

Circular Convolution

Multiplication of N-point DFTs corresponds to time-domain circular convolution.

$$\frac{1}{N}(x \circledast h)[n] \iff X_N[k]H_N[k]$$

Non-Circular Convolution with the DFT

The linear convolution of a length-L signal $x[n]$ with a length-P signal $h[n]$ is a length-$(L+P-1)$ signal $(x*h)[n]$. To use the DFT (implemented via FFT algorithms, for example) to perform fast convolutions, we want to eliminate time-aliasing artifacts from circular convolution.

- Zero-pad $x[n]$ and $h[n]$ each to a length of $(L+P-1)$ samples.
- Compute the $(L+P-1)$-point DFTs $X[k]$ and $H[k]$. Multiply the DFTs: $X[k]H[k]$.
- Compute the inverse DFT of $X[k]H[k]$ to obtain the linear convolution of $x[n]$ with $h[n]$.

Windowing

To obtain a length-L data sequence from an indefinite-length discrete-time sequence $x[n]$, multiply by a length-L window function $w[n]$ such that $w[n]=0$ for $n \notin \{0,1,\ldots,L-1\}$ in order to zero out values of $x[n]$ outside the window. This produces the finite-length windowed sequence $x_w[n]=x[n]w[n]$.

Windowing: Time Domain

Apply the length-L window function $w[n]$ to an indefinite-length discrete-time sequence $x[n]$. This produces the length-L data sequence $x_w[n]=x[n]w[n]$.

$$x_w[n]=x[n]w[n]$$

Windowing: Frequency Domain

Multiplication in the time domain corresponds to convolution in the frequency domain. That is, multiplying $x[n]$ by the window function $w[n]$ corresponds to convolving the DTFT of $x[n]$ with the frequency response of $w[n]$.

$$X_w(\Omega)=\frac{1}{2\pi}(X*H)(\Omega)$$

Ideally, the frequency response of $w[n]$ would approximate an idealized impulse, which would prevent egregious spectral leakage, or "frequency-domain smearing." A number of window functions are used in practice. We have examined rectangular, triangular, and Hann windows in 6.300. The latter two windows, also referred to as Bartlett and Hanning windows, respectively, "taper off" near the ends of the interval with the intent of suppressing sidelobes in the frequency response.

Windowing: Discretized Frequency Domain

We may conceptualize the DFT as N equally-spaced samples of the DTFT of $x_w[n]$ over the interval $[0, 2\pi]$, where N is the length of the DFT. In general, the DFT length N need not equal not the window length L, though we have often set or implicitly assumed $N = L$ in 6.300.

$$X_w[k] = \frac{1}{N} X_w\left(\frac{2\pi k}{N}\right)$$

The scaling factor of $1/N$ is not conceptually significant, but it is necessary for us to stay consistent with our Fourier transform conventions as formulated in 6.300.

It may be shown that the continuous-time cyclical frequency f_k corresponding to DFT index k is given by $f_k = k\Delta f$, where $\Delta f = (f_s/N)$ denotes the spacing, in hertz, between adjacent DFT indices. Zero-padding ($N > L$) does not enable one to distinguish arbitrarily-close spectral peaks, however. Increasing N increases the number of frequency samples of the DTFT but does not change the underlying frequency response of the window function $w[n]$, which smears the spectrum of $x[n]$. The width of the frequency response of $w[n]$ is inversely proportional to the window length L, which is not the same as the DFT length N.