6.300 Story Sheet (Fall 2024)

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

Semester in Review

Fourier Representations

- **09/05:** Signal Processing
- **09/10:** Sinusoids and Fourier Series
- **09/12:** Continuous-Time Fourier Series (CTFS)
- **09/17:** Sampling and Quantization
- **09/19:** Discrete-Time Fourier Series (DTFS)
- **09/24:** Continuous-Time Fourier Transform (CTFT)
- **09/26:** Discrete-Time Fourier Transform (DTFT)

Signals and Systems

- **10/08:** Linear Time-Invariant (LTI) Systems
- **10/10:** Impulse Response and Convolution
- **10/17:** Frequency Response and Filtering

Digital Signal Processing with the Discrete Fourier Transform

- **10/22:** Discrete Fourier Transform (DFT)
- **10/24:** Spectral Analysis and Circular Convolution with the Discrete Fourier Transform
- **10/29:** Short-Time Fourier Transform (STFT) and Music Information Retrieval (MIR)
- **10/31:** Fast Fourier Transform (FFT)

Applications and Extensions

- **11/12:** Modulation and Communications Systems
- **11/14:** Speech Processing
- **11/19:** 2D Fourier Transforms
- **11/21:** 2D Fourier Transforms
- **11/26:** 2D Convolution and Filtering
- **12/03:** Data Compression and Discrete Cosine Transform (DCT)
- **12/05:** Magnetic Resonance Imaging (MRI)

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 $f_0 = 1/T_0$	cycles per second or hertz (Hz)
CT angular frequency $\omega_0 = 2\pi/T_0 = 2\pi f_0$	radians per second
DT cyclical frequency $f_0/f_s = f_0 T_s = 1/N_0$	cycles per sample
DT angular frequency $\Omega_0 = \omega_0/f_s = 2\pi f_0/f_s = 2\pi f_0 T_s = 2\pi/N_0$	radians per sample

Geometric Series

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

Binomial Theorem

$$(\alpha+\beta)^n = \binom{n}{0}\alpha^n + \binom{n}{1}\alpha^{n-1}\beta + \binom{n}{2}\alpha^{n-2}\beta^2 + \cdots + \binom{n}{n-1}\alpha\beta^{n-1} + \binom{n}{n}\beta^n$$

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

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

$$X(\omega) = \mathrm{Re}\{X(\omega)\} + j\,\mathrm{Im}\{X(\omega)\} = |X(\omega)|\,e^{j\angle X(\omega)}$$

$$|X(\omega)| = \sqrt{\mathrm{Re}\{X(\omega)\}^2 + \mathrm{Im}\{X(\omega)\}^2} \qquad \tan\big(\angle X(\omega)\big) = \frac{\mathrm{Im}\{X(\omega)\}}{\mathrm{Re}\{X(\omega)\}}$$

Fourier-Related Representations

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)

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

2D 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)

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

Fourier Representations: Pairs and Properties

CTFS Pairs (Period T)

$e^{jk_0 \frac{2\pi}{T} t} \longleftrightarrow \delta[k - k_0]$

DTFS Pairs (Period N)

$e^{jk_0 \frac{2\pi}{N} n} \longleftrightarrow \delta[k - k_0]$

CTFT Pairs

$1 \longleftrightarrow 2\pi\delta(\omega)$
$e^{j\omega_0 t} \longleftrightarrow 2\pi\delta(\omega - \omega_0)$
$\delta(t) \longleftrightarrow 1$
$\delta(t - t_0) \longleftrightarrow e^{-j\omega t_0}$
$e^{-\alpha t}u(t) \longleftrightarrow \frac{1}{\alpha + j\omega}$ for $\alpha > 0$

DTFT Pairs

$1 \longleftrightarrow 2\pi\delta(\Omega)$
$e^{j\Omega_0 n} \longleftrightarrow 2\pi\delta(\Omega - \Omega_0)$
$\delta[n] \longleftrightarrow 1$
$\delta[n - n_0] \longleftrightarrow e^{-j\Omega n_0}$
$\alpha^n u[n] \longleftrightarrow \frac{1}{1 - \alpha e^{-j\Omega}}$ for $|\alpha| < 1$

DFT Pairs (N-Point DFT)

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

2D DFT Pairs

$1 \longleftrightarrow \delta[k_r, k_c]$
$e^{jk'_r \frac{2\pi}{R} r} \longleftrightarrow \delta[k_r - k'_r]$
$e^{jk'_c \frac{2\pi}{C} c} \longleftrightarrow \delta[k_c - k'_c]$
$\delta[r, c] \longleftrightarrow \frac{1}{RC}$
$\delta[r - r_0] \longleftrightarrow \frac{1}{R} e^{-jk_r \frac{2\pi}{R} r_0}$
$\delta[c - c_0] \longleftrightarrow \frac{1}{C} e^{-jk_c \frac{2\pi}{C} c_0}$

CTFT Properties

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

DTFT Properties

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

DFT Properties

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

2D DFT Properties

Generalize all properties from 1D to 2D.

$x[r, c] \longleftrightarrow X[k_r, k_c]$
$x[r - r_0, c] \longleftrightarrow e^{-jk_r \frac{2\pi}{R} r_0} X[k_r, k_c]$
$\frac{1}{RC}(x_1 \circledast x_2)[r, c] \longleftrightarrow X_1[k_r, k_c]X_2[k_r, k_c]$
$x_1[r, c]x_2[r, c] \longleftrightarrow (X_1 \circledast X_2)[k_r, k_c]$

From Continuous Time to Discrete Time

Sampling

We *sample* a continuous-time function $x_c(t)$ to get a discrete-time sequence $x_d[n]$. The *sampling interval* or *sampling period* T_s specifies the time, in seconds, between successive samples. Equivalently, if one samples $x_c(t)$ every T_s seconds, the sampling rate is $f_s = 1/T_s$ samples per second, or f_s hertz.

$$x_d[n] \equiv x_c(nT_s)$$

Though this perspective was not emphasized in 6.300, sampling a continuous-time function can be conceptualized as multiplying $x_c(t)$ by a periodic impulse train $p(t)$, with the impulses separated by T_s in time. This perspective elegantly explains the periodic replication of the *spectrum* (i.e., frequency content) of $x_c(t)$ inherent in $x_d[n]$, as multiplication by a periodic impulse train in the time domain corresponds to convolution with yet another periodic impulse train in the frequency domain.

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \qquad x_d[n] = (x_c \cdot p)(nT_s) \qquad X_d(\Omega) = (X_c * P)(\Omega)\Big|_{\Omega=\omega T_s}$$

Sampling leads to periodic replication of the spectrum of $x(t)$. All DTFTs are 2π-periodic.

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 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` online. The application visualizes the aliasing that occurs when a continuous-time signal is sampled by an analog-to-digital converter, or ADC.

Linear Time-Invariant Systems

Linearity

Together, additivity and homogeneity imply linearity.

Additivity

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

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

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

Homogeneity

Scaling the input scales the output correspondingly.

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

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

Linearity

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

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

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

Time-Invariance

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

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

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

Causality

This property is not one we've examined in 6.300, but it will arise in subsequent classes concerning signals and systems. A system is said to be *causal* if the output at any given time depends only on:

- current or previous values of the input; and
- previous values of the output.

For example, the system defined by $y[n] = y[n-1] + x[n]$ is causal, whereas the system defined by $y[n] = x[n+1]$ is not causal.

Analogously, a signal $x[n]$ is said to be causal if $x[n] = 0$ for $n < 0$. That is, a causal signal could be the impulse response of a causal LTI system.

Representations of Linear Time-Invariant Systems I

Differential Equation (CT) / Difference Equation (DT)

Derivatives in continuous time play an analogous role to sample delays and advances in discrete time.

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

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

The difference equation representation motivates implementing causal systems using recursion.

$$y[n] = \sum_{k=1}^{M_y} a_k y[n-k] + \sum_{k=0}^{M_x} b_k x[n-k]$$

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 $\delta[n]$.

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

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

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.

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

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

For causal systems, $h(t) = 0$ for $t < 0$ and $h[n] = 0$ for $n < 0$.

$$x(t) \to \boxed{\mathcal{H}} \to y(t) = (x*h)(t) = \int_{0}^{\infty} h(\tau)x(t-\tau)d\tau$$

$$x[n] \to \boxed{\mathcal{H}} \to y[n] = (x*h)[n] = \sum_{k=0}^{\infty} h[k]x[n-k] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + \cdots$$

Representations of Linear Time-Invariant Systems II

Frequency Response

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

The beauty and significance of this characterization arises from the fact that 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.

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

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.

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

Convolution

Linear 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 of linear convolution.

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

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

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.

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

Convolution with Impulses

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

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

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

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

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

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

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

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

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

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

Convolution by Table

Linear 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	

Circular Convolution

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

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 DFT

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 $x_w[n]$.

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

The DFT returns N (scaled) equally-spaced samples of the DTFT over the interval $[0, 2\pi]$. 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.

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

Linear 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 $h[n]$.

Spectral Analysis with the Discrete Fourier Transform

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.

Practice Problems: Sampling and Reconstruction

#1-1: Spectrum of a Continuous-Time Signal

Consider the periodic continuous-time signal $x_c(t)$ defined below.

$$x_c(t) = \cos\left(\frac{\pi}{2}t\right)$$

Sketch $X_c(\omega)$, the continuous-time Fourier transform of $x_c(t)$, over $-10\pi \leq \omega \leq 10\pi$.

- If $X_c(\omega)$ is complex-valued, plot the magnitude $|X_c(\omega)|$ and phase $\angle X_c(\omega)$ separately.
- For full credit, clearly label the axes and all key parameters.

#1-2: Sampling with an Impulse Train

We sample the continuous-time signal $x_c(t)$ once per second to produce the discrete-time signal $x_d[n]$. To sample $x_c(t)$, we multiply $x_c(t)$ by a periodic impulse train $p(t)$ to produce $(x_c \cdot p)(t)$.

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t-n) = \cdots + \delta(t+2) + \delta(t+1) + \delta(t) + \delta(t-1) + \delta(t-2) + \cdots$$

Sketch $P(\omega)$, the continuous-time Fourier transform of $p(t)$, over $-10\pi \leq \omega \leq 10\pi$.

- If $P(\omega)$ is complex-valued, plot the magnitude $|P(\omega)|$ and phase $\angle P(\omega)$ separately.
- For full credit, clearly label the axes and all key parameters.

Hint: $p(t)$ is periodic! Compute the Fourier series coefficients $P[k]$, and then relate the Fourier series coefficients $P[k]$ to the Fourier transform $P(\omega)$.

#1-3: Spectrum of a Sampled Signal

We express the discrete-time signal $x_d[n]$ as

$$x_d[n] = (x_c \cdot p)(n)$$

for integer n.

Sketch $X_d(\Omega)$, the discrete-time Fourier transform of $x_d[n]$, over $-10\pi \leq \Omega \leq 10\pi$.

- If $X_d(\Omega)$ is complex-valued, plot the magnitude $|X_d(\Omega)|$ and phase $\angle X_d(\Omega)$ separately.
- For full credit, clearly label the axes and all key parameters.

Practice Problems: Sampling and Reconstruction (cont.)

#1-4: Reconstruction via Zero-Order Hold

We reconstruct a continuous-time signal $x_r(t)$ from the samples of $x_d[n]$. This is done by way of a *zero-order hold*: We take a single sample of the discrete-time signal $x_d[n]$ and hold that value constant for one second, until repeating this procedure with the next sample.

We can formulate this zero-order hold as a two-step procedure.

- Express the samples of $x_d[n]$ as impulses spaced out in continuous time.

$$x_d(t) = \sum_n x_d[n]\delta(t-n)$$

- Convolve $x_d(t)$ with $h(t)$, a rectangular pulse of unit length, to produce $x_r(t) = (x_d * h)(t)$.

$$h(t) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Sketch $H(\omega)$, the frequency response corresponding to $h(t)$, over $-10\pi \leq \omega \leq 10\pi$.

- If $H(\omega)$ is complex-valued, plot the magnitude $|H(\omega)|$ and phase $\angle H(\omega)$ separately.
- For full credit, clearly label the axes and all key parameters.

#1-5: Spectrum of Reconstructed Signal

Picking up from **(d)**, the reconstructed continuous-time signal $x_r(t)$ may be expressed as

$$x_r(t) = (x_d * h)(t).$$

Sketch $X_r(\omega)$, the continuous-time Fourier transform of $x_r(t)$, over $-10\pi \leq \omega \leq 10\pi$.

- If $X_r(\omega)$ is complex-valued, plot the magnitude $|X_r(\omega)|$ and phase $\angle X_r(\omega)$ separately.
- For full credit, clearly label the axes and all key parameters.

Practice Problems: LTI Systems

#2: Moving Average

Consider the causal discrete-time LTI system with the following unit-sample response.

$$h[n] = \delta[n] + 2\delta[n-1] + \delta[n-2]$$

#2-1: Sketch $h[n]$. Before performing any computations, do you expect this system to act more like a low-pass filter or more like a high-pass filter? Briefly justify your intuition.

#2-2: Determine an expression for the frequency response $H(\Omega)$ corresponding to $h[n]$. Factor $H(\Omega)$ as $H(\Omega) = A(\Omega)e^{j\phi(\Omega)}$ for a real and symmetric $A(\Omega)$ and a real and anti-symmetric $\phi(\Omega)$.

#2-3: On separate axes, sketch the magnitude $|H(\Omega)|$ and phase $\angle H(\Omega)$ of $H(\Omega)$. Was your guess from **#2-1** correct?

#2-4: For this system, determine a linear difference equation with constant coefficients that relates the input $x[n]$ to the output $y[n]$.

#3: Discrete-Time Differentiators

Consider the discrete-time LTI system $\mathcal{S}_1$ defined by the following difference equation.

$$y_1[n] = x[n] - x[n-1]$$

#3-1: Before performing any computations, do you expect this system to act more like a low-pass filter or more like a high-pass filter? Briefly justify your intuition.

#3-2: Determine an expression for the unit-sample response $h_1[n]$. Sketch $h_1[n]$.

#3-3: Determine an expression for the frequency response $H_1(\Omega)$ corresponding to $h_1[n]$.

#3-4: On separate axes, sketch the magnitude $|H_1(\Omega)|$ and phase $\angle H_1(\Omega)$ of $H_1(\Omega)$. Was your guess from **#3-1** correct?

Now, consider the discrete-time LTI system $\mathcal{S}_2$ defined by the following difference equation.

$$y_2[n] = x[n+1] - 2x[n] + x[n-1]$$

#3-5: Before performing any computations, do you expect this system to act more like a low-pass filter or more like a high-pass filter? Briefly justify your intuition.

#3-6: Determine an expression for the unit-sample response $h_2[n]$. Sketch $h_2[n]$.

#3-7: Determine an expression for the frequency response $H_2(\Omega)$ corresponding to $h_2[n]$.

#3-8: On separate axes, sketch the magnitude $|H_2(\Omega)|$ and phase $\angle H_2(\Omega)$ of $H_2(\Omega)$. Was your guess from **#3-5** correct?

#3-9: Compare and contrast system $\mathcal{S}_1$ with system $\mathcal{S}_2$.

Practice Problems: DFT and Convolution

#4: Computing the DFT

Suppose that $x[n]$ is a length-L sequence. We want to compute $X_N[k]$, the N-point DFT of $x[n]$, where $N < L$. Consider the following two methods for computing $X_N[k]$.

- **Method #1:** Compute $X(\Omega)$, the DTFT of $x[n]$. Sample $X(\Omega)$ at $\Omega_k = 2\pi k/N$ for $k \in \{0, 1, 2, \ldots, N-1\}$ and scale by $1/N$ to produce $X_N[k]$.
- **Method #2:** Directly compute the N-point DFT from the first N samples of $x[n]$ using the DFT analysis formula.

Will the two methods described above always produce the same $X_N[k]$? Why or why not?

#5: Pythonic Convolution

```
» x = np.random.randn(L) # Random length-L sequence; L is a positive integer
» h = np.random.randn(P) # Random length-P sequence; P is a positive integer
» f = np.convolve(x, h, mode='full') # Compute linear convolution of x and h
```

#5-1: What is `len(f)`?

```
» X = np.fft.fft(x, L) # L-point DFT of x
» H = np.fft.fft(h, L) # L-point DFT of h; assume L > P
» G = L * np.multiply(X, H) # Multiply L-point DFTs elementwise; scale by L
» g = np.fft.ifft(G, L) # L-point inverse DFT
```

#5-2: What is `len(g)`? Is `len(g)` the same as `len(f)`? Why or why not?

#5-3: Determine the smallest positive integer `n1` and largest positive integer `n2` for which the expression `f[n1:n2] == g[n1:n2]` will always return `True`.

#6: Linear Convolution vs. Circular Convolution

Suppose that $x[n] = 0$ and $h[n] = 0$ for $n < 0$. We convolve $x[n]$ and $h[n]$ to produce $(x * h)[n]$.

In addition, suppose that we compute the DTFTs of $x[n]$ and $h[n]$ and sample them at $\Omega_k = 2\pi k/5$ for $k \in \{0, 1, 2, 3, 4\}$ to produce the respective 5-point DFTs $X_5[k]$ and $H_5[k]$. We then compute the 5-point inverse DFT of $5X_5[k]H_5[k]$.

n	0	1	2	3	4	5	6	7	8	9
$(x * h)[n]$	4	3	7	7	0	A	B	C	D	E
$\text{IDFT}_5\{5X_5[k]H_5[k]\}$	4	3	14	13	1	4	3	14	13	1

#6-1: Determine appropriate values for the constants A, B, C, D, and E.

#6-2: Give a few choices of $x[n]$ and $h[n]$ that produce $(x * h)[n]$.

Practice Problems: Practical Processing

Alyssa P. Hacker is transmitting signals to Ben Bitdiddle. Help Ben process these signals!
Each sub-problem may be completed independently and with few, if any, calculations. For full credit, justify your answers. Keep your explanations concise.

#7-1: Modulation and Sub-Nyquist Sampling

$$x_c(t) \longrightarrow \boxed{\text{CT/DT Converter}} \longrightarrow x[n]$$

Alyssa P. Hacker is sending Ben Bitdiddle a real-valued signal $x_c(t)$.

$$x_c(t) = x_\ell(t)\cos\Big((2\pi \times 2750)t\Big)$$

$X_\ell(\omega)$ is symmetric about $\omega = 0$ and bandlimited such that $X_\ell(\omega) = 0$ for $|\omega| \geq 2\pi \times 250$.
Ben wants to process $x_c(t)$ in discrete time, so he needs to sample $x_c(t)$ to produce $x[n]$. Ben's CT/DT converter can only sample at a rate of $2\pi \times 500$ radians per second, though. Is it possible for Ben to sample $x_c(t)$ fast enough to produce $x[n]$ without allowing aliasing to distort the spectrum? If so, explain a procedure Ben could employ. If not, explain why. (For full credit, you must justify your answer.)

#7-2: Multirate Processing

Ben's computer is awfully slow. To reduce the number of mathematical operations his computer will need to perform during processing, Ben wonders if he can process a discrete-time signal at a lower rate than he sampled at. To that end, Ben considers the *decimation* and *expansion* operations defined below. (R_d and R_e denote positive integers.)

Decimation

$$x[n] \rightarrow \boxed{\downarrow R_d} \rightarrow x_d[n] = x[nR_d]$$

Expansion

$$x[n] \rightarrow \boxed{\uparrow R_e} \rightarrow x_e[n] = \begin{cases} x\left[\frac{n}{R_e}\right] & n \in \{0, R_e, 2R_e, 3R_e, \dots\} \\ 0 & \text{otherwise} \end{cases}$$

$$x[n] \rightarrow \boxed{\boxed{\downarrow R_d} \rightarrow \boxed{\uparrow R_e}} \rightarrow \hat{x}[n]$$

Consider the system shown directly above. Ben claims that, as long as $R_d = R_e$, it will always be the case that $\hat{x}[n] = x[n]$ for any $x[n]$. Is Ben's assessment correct? Why or why not? Explain. (For full credit, you must justify your answer.)

Practice Problems: Practical Processing (continued)

#7-3: Real-Time Processing with Block Convolution

Alyssa is transmitting a long signal of unknown length. Ben wants to perform real-time processing as he receives the samples of $x[n]$, so he decides to employ the overlap-add method.

- Ben breaks $x[n]$ up into non-overlapping segments of length L.
- Ben processes each segment by convolving it with $h[n]$, a unit-sample response of length P, where $P < L$.
- Ben adds successive segments. M points of the previous segment overlap with (i.e., "spill over into") each segment.

Ben doesn't remember what the value for M, the overlap between successive segments, ought to be. Using your knowledge of convolution, help Ben determine a value for M in terms of L and P. (For full credit, you must justify your answer.)

#7-4: Fast Convolution with the DFT

Ben has heard how efficient FFT algorithms for computing the DFT are, and he wonders if performing his processing in the frequency domain would be worthwhile.

- Using FFT algorithms, compute the N-point DFTs of $x[n]$ and $h[n]$: $X_N[k]$ and $H_N[k]$.
- Multiply $X_N[k]$ and $H_N[k]$ elementwise to produce $Y_N[k]$.
- Compute the N-point inverse DFT of $Y_N[k]$ to produce $y[n]$.

However, Ben also knows that multiplying DFTs in the frequency domain corresponds to circularly convolving $x[n]$ and $h[n]$ in the time domain. Ben concludes that it is impractical to use the DFT to perform filtering, as this method will always lead to unwanted circular convolution artifacts.

Is Ben's conclusion correct? If so, why? If not, explain how Ben can prevent circular convolution artifacts from distorting his processed signal. (For full credit, you must justify your answer.)

#7-5: Ideal Filtering with the DFT

Regardless of what you told him in **(d)**, Ben decides to go ahead with performing his processing in the frequency domain. Ben excitedly claims that he can perform "ideal filtering" — that is, process signals using "ideal" filters with arbitrarily narrow transition bands — using the DFT, and he shows you a plot of his ideal low-pass filter.

Is Ben correct that the DFT enables "ideal filtering," or is he mistaken? Explain. (For full credit, you must justify your answer.)

Practice Problems: Practical Processing (continued)

#7-6: Spectral Analysis with the DFT

Ben analyzes the DFT of a segment of $x[n]$. Ben claims that, if $X[k] = 0$ for all $k \in \{0, 1, \ldots, N-1\}$, then $X(\Omega) = 0$ for all $\Omega \in [0, 2\pi]$.

Is Ben's claim true? If so, explain why. If not, refute his claim. (For full credit, you must justify your answer.)

#7-7: Zero-Padding and Frequency Resolution

Ben recalls that the frequency resolution of a DFT is given by

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

where f_s denotes the sampling rate in hertz used to sample the continuous-time signal, and N denotes the length of the DFT computed. Ben plans to zero-pad $x[n]$ to an arbitrarily-long length and then compute a DFT using a large value of N. Zero-padding, he claims, will give him arbitrarily-fine frequency resolution, so he will be able to resolve two peaks in the underlying DTFT no matter how close they are.

Is Ben correct? Explain. (For full credit, you must justify your answer.)

Beyond 6.300

Signal Processing

6.301: Signals, Systems, and Inference — L. Zheng, P. Hagelstein

Covers signals, systems and inference in communication, control and signal processing. Input-output and state-space models of linear systems driven by deterministic and random signals. Time-domain and transform-domain representations of discrete-time signals. State feedback and observers. Probabilistic models. Random processes, correlation functions, power spectral density, and spectral factorization. Least-mean-square error estimation and Wiener filtering. Hypothesis testing, detection, and matched filters. **Prerequisites:** 6.300; 6.370, 6.380, or 18.05. **Recommended:** 18.06 or equivalent.

6.302: Fundamentals of Music Processing — E. Egozy

Analyzes recorded music in digital audio form using advanced signal processing and optimization techniques to understand higher-level musical meaning. Windowing, feature extraction, discrete and short-time Fourier transforms, chromagrams, and onset detection. Analysis methods including dynamic time warping, dynamic programming, self-similarity matrices, and matrix factorization. Variety of applications, including event classification, audio alignment, chord recognition, structural analysis, tempo and beat tracking, content-based audio retrieval, and audio decomposition. **Prerequisites:** 6.300; 21M.051.

6.700: Discrete-Time Signal Processing — J. Ward

Control

6.310: Dynamical System Modeling and Control Design — J. White

Biomedical

6.480: Biomedical Systems — Modeling and Inference — L. Lewis et al.
6.880: Biomedical Signal and Image Processing — R. Alam

Communications

6.741: Principles of Digital Communication — M. Médard

Natural Language Processing and Speech

6.861: Quantitative Methods for Natural Language Processing — J. Andreas, J. Glass
6.862: Spoken Language Processing — J. Glass

Computational Imaging

6.C27: Computational Imaging — Physics and Algorithms — G. Barbastathis, S. You

Practice Problems: Sampling and Reconstruction

#1-1: Spectrum of a Continuous-Time Signal

Consider the periodic continuous-time signal $x_c(t)$ defined below.

$$x_c(t) = \cos\left(\frac{\pi}{2}t\right)$$

Sketch $X_c(\omega)$, the continuous-time Fourier transform of $x_c(t)$, over $-10\pi \leq \omega \leq 10\pi$.

- If $X_c(\omega)$ is complex-valued, plot the magnitude $|X_c(\omega)|$ and phase $\angle X_c(\omega)$ separately.
- For full credit, clearly label the axes and all key parameters.

Express $x_c(t)$ as a sum of conjugate complex exponentials.

$$x_c(t) = \frac{1}{2}e^{-j\frac{\pi}{2}t} + \frac{1}{2}e^{j\frac{\pi}{2}t}$$

Compute the CTFT.

$$X_c(\omega) = \pi\delta(\omega + \pi/2) + \pi\delta(\omega - \pi/2)$$

$X_c(\omega)$ is real and non-negative, so $|X_c(\omega)| = X_c(\omega)$ and $\angle X_c(\omega) = 0$ for all ω.

Practice Problems: Sampling and Reconstruction (cont.)

#1-2: Sampling with an Impulse Train

We sample the continuous-time signal $x_c(t)$ once per second to produce the discrete-time signal $x_d[n]$. To sample $x_c(t)$, we multiply $x_c(t)$ by a periodic impulse train $p(t)$ to produce $(x_c \cdot p)(t)$.

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t-n) = \cdots + \delta(t+2) + \delta(t+1) + \delta(t) + \delta(t-1) + \delta(t-2) + \cdots$$

Sketch $P(\omega)$, the continuous-time Fourier transform of $p(t)$, over $-10\pi \leq \omega \leq 10\pi$.

- If $P(\omega)$ is complex-valued, plot the magnitude $|P(\omega)|$ and phase $\angle P(\omega)$ separately.
- For full credit, clearly label the axes and all key parameters.

Hint: $p(t)$ is periodic! Compute the Fourier series coefficients $P[k]$, and then relate the Fourier series coefficients $P[k]$ to the Fourier transform $P(\omega)$.

Compute the Fourier series coefficients $P[k]$. Relate $P[k]$ to the Fourier transform $P(\omega)$.

$$P[k] = \int_0^1 \delta(t) e^{-j2\pi kt} dt = 1 \text{ for all } k$$

Each Fourier series coefficient is an impulse in frequency. (Don't forget to multiply by 2π.)

$$P(\omega) = 2\pi \sum_{k=-\infty}^{\infty} \delta(\omega - 2\pi k)$$

Practice Problems: Sampling and Reconstruction (cont.)

#1-3: Spectrum of a Sampled Signal

We express the discrete-time signal $x_d[n]$ as

$$x_d[n] = (x_c \cdot p)(n)$$

for integer n.

Sketch $X_d(\Omega)$, the discrete-time Fourier transform of $x_d[n]$, over $-10\pi \leq \Omega \leq 10\pi$.

- If $X_d(\Omega)$ is complex-valued, plot the magnitude $|X_d(\Omega)|$ and phase $\angle X_d(\Omega)$ separately.
- For full credit, clearly label the axes and all key parameters.

Multiplication in the time domain corresponds to convolution in the frequency domain.

$$x_d[n] = (x_c \cdot p)(n) \leftrightarrow X_d(\Omega) = \frac{1}{2\pi}(X_c * P)(\omega)\Big|_{\omega=\Omega}$$

Sampling results in periodic replication of the continuous-time spectrum every 2π radians per sample.

$$X_c(\omega) = \pi\delta(\omega + \pi/2) + \pi\delta(\omega - \pi/2)$$

$$X_d(\Omega) = \pi\delta\Big((\Omega + \pi/2) \bmod 2\pi\Big) + \pi\delta\Big((\Omega - \pi/2) \bmod 2\pi\Big)$$

Practice Problems: Sampling and Reconstruction (cont.)

#1-4: Reconstruction via Zero-Order Hold

We reconstruct a continuous-time signal $x_r(t)$ from the samples of $x_d[n]$. This is done by way of a *zero-order hold*: We take a single sample of the discrete-time signal $x_d[n]$ and hold that value constant for one second, until repeating this procedure with the next sample.

We can formulate this zero-order hold as a two-step procedure.

- Express the samples of $x_d[n]$ as impulses spaced out in continuous time.

$$x_d(t) = \sum_n x_d[n]\delta(t-n)$$

- Convolve $x_d(t)$ with $h(t)$, a rectangular pulse of unit length, to produce $x_r(t) = (x_d * h)(t)$.

$$h(t) = \begin{cases} 1 & 0 \le t \le 1 \\ 0 & \text{otherwise} \end{cases}$$

Sketch $H(\omega)$, the frequency response corresponding to $h(t)$, over $-10\pi \le \omega \le 10\pi$.

- If $H(\omega)$ is complex-valued, plot the magnitude $|H(\omega)|$ and phase $\angle H(\omega)$ separately.
- For full credit, clearly label the axes and all key parameters.

Compute the Fourier transform of $h(t)$. As $h(t)$ is a rectangular pulse in time, the corresponding Fourier transform is that of a sinc.

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{-j\omega t}dt = \int_0^1 e^{-j\omega t}dt = \left.\frac{e^{-j\omega t}}{-j\omega}\right|_0^1 = \frac{1}{-j\omega}\left(e^{-j\omega} - 1\right)$$

This may be simplified further.

$$H(\omega) = \frac{e^{-j\omega/2}}{-j\omega}\left(e^{-j\omega/2} - e^{j\omega/2}\right) = \frac{e^{-j\omega/2}}{-j\omega}\left(-2j\sin(\omega/2)\right) = \frac{\sin(\omega/2)}{(\omega/2)}e^{-j\omega/2}$$

The magnitude $|H(\omega)|$ is the pointwise absolute value of $H(\omega)$. The phase $\angle H(\omega)$ is a line through the origin with a slope of $-1/2$, with abrupt jumps of π whenever $H(\omega) < 0$. (We wrap the phase such that $\angle H(\omega) \in [-\pi, \pi]$ for all ω.)

Practice Problems: Sampling and Reconstruction (cont.)

#1-5: Spectrum of Reconstructed Signal

Picking up from **(d)**, the reconstructed continuous-time signal $x_r(t)$ may be expressed as

$$x_r(t) = (x_d * h)(t).$$

Sketch $X_r(\omega)$, the continuous-time Fourier transform of $x_r(t)$, over $-10\pi \leq \omega \leq 10\pi$.

- If $X_r(\omega)$ is complex-valued, plot the magnitude $|X_r(\omega)|$ and phase $\angle X_r(\omega)$ separately.
- For full credit, clearly label the axes and all key parameters.

$$X_r(\omega) = X_d(\omega)H(\omega) = |X_d(\omega)||H(\omega)|e^{j\angle H(\omega)}$$

Practice Problems: LTI Systems

#2: Moving Average

Consider the causal discrete-time LTI system with the following unit-sample response.

$$h[n] = \delta[n] + 2\delta[n-1] + \delta[n-2]$$

#2-1: Sketch $h[n]$. Before performing any computations, do you expect this system to act more like a low-pass filter or more like a high-pass filter? Briefly justify your intuition.

#2-2: Determine an expression for the frequency response $H(\Omega)$ corresponding to $h[n]$. Factor $H(\Omega)$ as $H(\Omega) = A(\Omega)e^{j\phi(\Omega)}$ for a real and symmetric $A(\Omega)$ and a real and anti-symmetric $\phi(\Omega)$.

$$H(\Omega) = 1 + 2e^{-j\Omega} + e^{-j2\Omega} = e^{-j\Omega}\Big(2 + e^{j\Omega} + e^{-j\Omega}\Big) = \Big(2 + 2\cos(\Omega)\Big)e^{-j\Omega} = A(\Omega)e^{j\phi(\Omega)}$$

#2-3: On separate axes, sketch the magnitude $|H(\Omega)|$ and phase $\angle H(\Omega)$ of $H(\Omega)$. Was your guess from **(a)** correct?

$|A(0)| = 4$ and $|A(\pm\pi)| = 0 \rightarrow$ low-pass filter

#2-4: For this system, determine a linear difference equation with constant coefficients that relates the input $x[n]$ to the output $y[n]$.

$$y[n] = x[n] + 2x[n-1] + x[n-2]$$

Practice Problems: LTI Systems (continued)

#3: Discrete-Time Differentiators

Consider the discrete-time LTI system $\mathcal{S}_1$ defined by the following difference equation.

$$y_1[n] = x[n] - x[n-1]$$

#3-1: Before performing any computations, do you expect this system to act more like a low-pass filter or more like a high-pass filter? Briefly justify your intuition.

#3-2: Determine an expression for the unit-sample response $h_1[n]$. Sketch $h_1[n]$.

$$x[n] = \delta[n] \implies y_1[n] = h_1[n] = \delta[n] - \delta[n-1]$$

#3-3: Determine an expression for the frequency response $H_1(\Omega)$ corresponding to $h_1[n]$.

$$H_1(\Omega) = 1 - e^{-j\Omega} = e^{-j\Omega/2}\left(e^{j\Omega/2} - e^{-j\Omega/2}\right) = 2j\sin(\Omega/2)e^{-j\Omega/2} = 2\sin(\Omega/2)e^{j(\pi/2-\Omega/2)}$$

#3-4: On separate axes, sketch the magnitude $|H_1(\Omega)|$ and phase $\angle H_1(\Omega)$ of $H_1(\Omega)$. Was your guess from **(a)** correct?

$|H_1(0)| = 0$ and $|H_1(\pm\pi)| = 2 \rightarrow$ high-pass filter

Now, consider the discrete-time LTI system $\mathcal{S}_2$ defined by the following difference equation.

$$y_2[n] = x[n+1] - 2x[n] + x[n-1]$$

#3-5: Before performing any computations, do you expect this system to act more like a low-pass filter or more like a high-pass filter? Briefly justify your intuition.

#3-6: Determine an expression for the unit-sample response $h_2[n]$. Sketch $h_2[n]$.

$$x[n] = \delta[n] \implies y_2[n] = h_2[n] = \delta[n+1] - 2\delta[n] + \delta[n-1]$$

#3-7: Determine an expression for the frequency response $H_2(\Omega)$ corresponding to $h_2[n]$.

$H_2(\Omega) = 2\cos(\Omega) - 2$, which is a high-pass filter: $|H_2(0)| = 0$ and $|H_2(\pm\pi)| = 4$

#3-8: On separate axes, sketch the magnitude $|H_2(\Omega)|$ and phase $\angle H_2(\Omega)$ of $H_2(\Omega)$. Was your guess from **(e)** correct?

#3-9: Compare and contrast system $\mathcal{S}_1$ with system $\mathcal{S}_2$.

Practice Problems: DFT and Convolution

#4: Computing the DFT

Suppose that $x[n]$ is a length-L sequence. We want to compute $X_N[k]$, the N-point DFT of $x[n]$, where $N < L$. Consider the following two methods for computing $X_N[k]$.

- **Method #1:** Compute $X(\Omega)$, the DTFT of $x[n]$. Sample $X(\Omega)$ at $\Omega_k = 2\pi k/N$ for $k \in \{0, 1, 2, \ldots, N-1\}$ and scale by $1/N$ to produce $X_N[k]$.
- **Method #2:** Directly compute the N-point DFT from the first N samples of $x[n]$ using the DFT analysis formula.

Will the two methods described above always produce the same $X_N[k]$? Why or why not?

No. Taking N samples of the DTFT (after computing a potentially-infinite summation) is not the same as computing an N-point DFT using the DFT analysis formula.

#5: Pythonic Convolution

```
» x = np.random.randn(L) # Random length-L sequence; L is a positive integer
» h = np.random.randn(P) # Random length-P sequence; P is a positive integer
» f = np.convolve(x, h, mode='full') # Compute linear convolution of x and h
```

#5-1: What is `len(f)`?

```
len(f) = L + P - 1
```

```
» X = np.fft.fft(x, L) # L-point DFT of x
» H = np.fft.fft(h, L) # L-point DFT of h; assume L > P
» G = L * np.multiply(X, H) # Multiply L-point DFTs elementwise; scale by L
» g = np.fft.ifft(G, L) # L-point inverse DFT
```

#5-2: What is `len(g)`? Is `len(g)` the same as `len(f)`? Why or why not?

```
len(g) = L
```

#5-3: Determine the smallest positive integer `n1` and largest positive integer `n2` for which the expression `f[n1:n2] == g[n1:n2]` will always return `True`.

Discard the first $P-1$ samples that time-aliased from circular convolution. Keep the rest.

Practice Problems: DFT and Convolution (continued)

#6: Linear Convolution vs. Circular Convolution

Suppose that $x[n] = 0$ and $h[n] = 0$ for $n < 0$. We convolve $x[n]$ and $h[n]$ to produce $(x * h)[n]$.

In addition, suppose that we compute the DTFTs of $x[n]$ and $h[n]$ and sample them at $\Omega_k = 2\pi k/5$ for $k \in \{0, 1, 2, 3, 4\}$ to produce the respective 5-point DFTs $X_5[k]$ and $H_5[k]$. We then compute the 5-point inverse DFT of $5X_5[k]H_5[k]$.

n	0	1	2	3	4	5	6	7	8	9
$(x * h)[n]$	4	3	7	7	0	A	B	C	D	E
$\text{IDFT}_5\{5X_5[k]H_5[k]\}$	4	3	14	13	1	4	3	14	13	1

#6-1: Determine appropriate values for the constants A, B, C, D, and E.

$A = 0$, $B = 0$, $C = 7$, $D = 6$, and $E = 1$

#6-2: Give a few choices of $x[n]$ and $h[n]$ that produce $(x * h)[n]$.

e.g., $x[n] = \delta[n]$ and $h[n] = (x * h)[n]$ as printed above (or vice versa)

Practice Problems: Practical Processing

Alyssa P. Hacker is transmitting signals to Ben Bitdiddle. Help Ben process these signals!
Each sub-problem may be completed independently and with few, if any, calculations. For full credit, justify your answers. Keep your explanations concise.

#7-1: Modulation and Sub-Nyquist Sampling

$$x_c(t) \longrightarrow \boxed{\text{CT/DT Converter}} \longrightarrow x[n]$$

Alyssa P. Hacker is sending Ben Bitdiddle a real-valued signal $x_c(t)$.

$$x_c(t) = x_\ell(t)\cos\big((2\pi \times 2750)t\big)$$

$X_\ell(\omega)$ is symmetric about $\omega = 0$ and bandlimited such that $X_\ell(\omega) = 0$ for $|\omega| \geq 2\pi \times 250$.
Ben wants to process $x_c(t)$ in discrete time, so he needs to sample $x_c(t)$ to produce $x[n]$. Ben's CT/DT converter can only sample at a rate of $2\pi \times 500$ radians per second, though. Is it possible for Ben to sample $x_c(t)$ fast enough to produce $x[n]$ without allowing aliasing to distort the spectrum? If so, explain a procedure Ben could employ. If not, explain why. (For full credit, you must justify your answer.)

Yes, Ben can perform "sub-Nyquist sampling." The gist is to modulate the signal down to baseband and apply a low-pass filter to prevent aliased copies from distorting the spectrum during sampling.

- Two copies of $X_\ell(\omega)$ are centered about $|\omega_c| = 2\pi \times 2750$ radians per sample. Consider $\cos(\omega_c t)$ to be the carrier signal and $x_\ell(t)$ to be the modulation signal: $x_c(t) = x_\ell(t)\cos(\omega_c t)$.
- Multiply $x_c(t)$ by the carrier signal $\cos(\omega_c t)$ to center $X_\ell(\omega)$ at $\omega = 0$.
- Apply a low-pass filter with cut-off frequency $|\omega_\ell| = 2\pi \times 250$ to prevent aliasing.
- Sample at the rate of $2\pi \times 500$ radians per second.

Practice Problems: Practical Processing (continued)

#7-2: Multirate Processing

Ben's computer is awfully slow. To reduce the number of mathematical operations his computer will need to perform during processing, Ben wonders if he can process a discrete-time signal at a lower rate than he sampled at. To that end, Ben considers the *decimation* and *expansion* operations defined below. (R_d and R_e denote positive integers.)

Decimation

$$x[n] \rightarrow \boxed{\downarrow R_d} \rightarrow x_d[n] = x[nR_d]$$

Expansion

$$x[n] \rightarrow \boxed{\uparrow R_e} \rightarrow x_e[n] = \begin{cases} x\left[\frac{n}{R_e}\right] & n \in \{0, R_e, 2R_e, 3R_e, \dots\} \\ 0 & \text{otherwise} \end{cases}$$

$$x[n] \rightarrow \boxed{\boxed{\downarrow R_d} \rightarrow \boxed{\uparrow R_e}} \rightarrow \hat{x}[n]$$

Consider the system shown directly above. Ben claims that, as long as $R_d = R_e$, it will always be the case that $\hat{x}[n] = x[n]$ for any $x[n]$. Is Ben's assessment correct? Why or why not? Explain. (For full credit, you must justify your answer.)

No, Ben's assessment is not correct. In general, $\hat{x}[n] \neq x[n]$ for any $x[n]$. Decimation by a factor of R_d in discrete time corresponds to sampling at a fraction $(1/R_d)$ of the initial sampling rate. This may lead to aliasing, which expansion cannot undo.

Practice Problems: Practical Processing (continued)

#7-3: Real-Time Processing with Block Convolution

Alyssa is transmitting a long signal of unknown length. Ben wants to perform real-time processing as he receives the samples of $x[n]$, so he decides to employ the overlap-add method.

- Ben breaks $x[n]$ up into non-overlapping segments of length L.
- Ben processes each segment by convolving it with $h[n]$, a unit-sample response of length P, where $P < L$.
- Ben adds successive segments. M points of the previous segment overlap with (i.e., "spill over into") each segment.

Ben doesn't remember what the value for M, the overlap between successive segments, ought to be. Using your knowledge of convolution, help Ben determine a value for M in terms of L and P. (For full credit, you must justify your answer.)

The convolution of a length-L segment with a length-P segment is a length-$(L+P-1)$ segment, so each successive segment should overlap the preceding segment by $M = P - 1$ samples.

#7-4: Fast Convolution with the DFT

Ben has heard how efficient FFT algorithms for computing the DFT are, and he wonders if performing his processing in the frequency domain would be worthwhile.

- Using FFT algorithms, compute the N-point DFTs of $x[n]$ and $h[n]$: $X_N[k]$ and $H_N[k]$.
- Multiply $X_N[k]$ and $H_N[k]$ elementwise to produce $Y_N[k]$.
- Compute the N-point inverse DFT of $Y_N[k]$ to produce $y[n]$.

However, Ben also knows that multiplying DFTs in the frequency domain corresponds to circularly convolving $x[n]$ and $h[n]$ in the time domain. Ben concludes that it is impractical to use the DFT to perform filtering, as this method will always lead to unwanted circular convolution artifacts.

Is Ben's conclusion correct? If so, why? If not, explain how Ben can prevent circular convolution artifacts from distorting his processed signal. (For full credit, you must justify your answer.)

If $x[n]$ is of length L and $h[n]$ is of length P, the linear convolution $(x * h)[n]$ will be of length $L + P - 1$. The unwanted artifacts of circular convolution arise from time-aliasing the output to lie within a length-N interval, where $N < L+P-1$. To prevent these artifacts from arising, we can zero-pad $x[n]$ and $h[n]$ both to length $N = L + P - 1$ and compute N-point DFTs. Ultimately, this will yield the length-$(L+P-1)$ linear convolution of $x[n]$ and $h[n]$ — without the unwanted circular convolution artifacts.

For even modest-length signals, zero-padding the time-domain signals and performing filtering using appropriately-long DFTs is faster than directly evaluating the convolution summation.

Practice Problems: Practical Processing (continued)

#7-5: Ideal Filtering with the DFT

Regardless of what you told him in **(d)**, Ben decides to go ahead with performing his processing in the frequency domain. Ben excitedly claims that he can perform "ideal filtering" — that is, process signals using "ideal" filters with arbitrarily narrow transition bands — using the DFT, and he shows you a plot of his ideal low-pass filter.

Is Ben correct that the DFT enables "ideal filtering," or is he mistaken? Explain. (For full credit, you must justify your answer.)

> The DFT does not enable "ideal filtering." The illusion of designing an "ideal filter" comes from looking at a select few frequency-samples of the underlying DTFT. Inspection of the DTFT would reveal that the filter designed is not, in fact, "ideal."

#7-6: Spectral Analysis with the DFT

Ben analyzes the DFT of a segment of $x[n]$. Ben claims that, if $X[k] = 0$ for all $k \in \{0, 1, \ldots, N-1\}$, then $X(\Omega) = 0$ for all $\Omega \in [0, 2\pi]$.

Is Ben's claim true? If so, explain why. If not, refute his claim. (For full credit, you must justify your answer.)

> Ben's claim — that, essentially, "if you can't see it, it's not there" — is false. The DFT may only sample at nulls of the underlying DTFT, for instance. Ben cannot make such a strong claim from inspecting only the DFT of $x[n]$.

Practice Problems: Practical Processing (continued)

#7-7: Zero-Padding and Frequency Resolution

Ben recalls that the frequency resolution of a DFT is given by

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

where f_s denotes the sampling rate in hertz used to sample the continuous-time signal, and N denotes the length of the DFT computed. Ben plans to zero-pad $x[n]$ to an arbitrarily-long length and then compute a DFT using a large value of N. Zero-padding, he claims, will give him arbitrarily-fine frequency resolution, so he will be able to resolve two peaks in the underlying DTFT no matter how close they are.

Is Ben correct? Explain. (For full credit, you must justify your answer.)

Ben is not correct. Zero-padding in time corresponds to interpolation between DTFT samples in the discrete-frequency domain and does not necessarily allow one to distinguish between two closely-spaced peaks in frequency. (The length of the window used is inversely proportional to the mainlobe width of the window's frequency response, which is one key factor that impacts one's ability to resolve closely-spaced peaks. The window length is not necessarily the same as the DFT length.) More simply said, you can't just add "zero information" in time to get arbitrarily-fine resolution in frequency.