# ECE 310 Spring 2026
## Conceptual Review

Corey Snyder

University of Illinois Urbana-Champaign

# Outline

There are seven main units of the course that we will cover in these slides:

1. LTI discrete-time systems, impulse response, convolution (Lectures 1–5)
2. $z$-transform (Lectures 6–11)
3. Discrete-time Fourier transform (Lectures 13–16)
4. Sampling, A/D conversion, D/A conversion (Lectures 17–19, 32)
5. Discrete Fourier transform (Lectures 20–23, 25–26)
6. Digital filter design (Lectures 27–29)
7. Multirate signal processing (Lecture 30–31)

# LTI Discrete-time Systems, Impulse Response, Convolution

# Discrete-time Systems

- A *discrete-time system* is any computational process that maps a discrete-time input, $x[n]$, to a discrete-time output, $y[n]$.
- Let $T$ denote an abstract system or *operator* that maps $x[n]$ to $y[n]$.

$$
\begin{aligned}
x[n] &\overset{T}{\mapsto} y[n] \\
y[n] &= T(x[n])
\end{aligned}
$$

Examples:

$$
\begin{aligned}
y[n] &= x[n] - x[n-1] \\
y[n] &= \text{median}\{x[n], x[n-1], x[n-2]\} \\
y[n] &= \text{SnapchatFilter}(x[n])
\end{aligned}
$$

# Linearity

## Linearity

A discrete-time system $T$ is *linear* if and only if it satisfies the following two conditions:

$$T(ax[n]) = aT(x[n]),\ a \in \mathbb{R} \qquad \text{(Homogeneity)}$$
$$T(x_1[n] + x_2[n]) = T(x_1[n]) + T(x_2[n]) \qquad \text{(Additivity)}$$

which is equivalent to the following single condition:

$$T(ax_1[n] + bx_2[n]) = aT(x_1[n]) + bT(x_2[n]),\ a, b \in \mathbb{R}\ \text{(Superposition)}.$$

In summary, we can prove a system is linear by showing it satisfies homogeneity **and** additivity; **or**, that the system satisfies superposition.

- Refer to Lecture 3 for more details and examples on linearity, time-invariance, causality, and stability!

# Time-invariance

## Time-invariance

A discrete-time system $T$ is *time-invariant* or *shift-invariant* if and only if

$$y[n-n_0] = T(x[n-n_0]), \; n \in \mathbb{Z}$$

for input $x[n]$ and output $y[n]$. In different notation, this may be written as

$$T(x)[n-n_0] = T(x[n-n_0]), \; n \in \mathbb{Z}.$$

In words, this means that shifting the input $x[n]$ by $n_0$ samples leads to a corresponding shift of $n_0$ samples in the output.

# Causality and BIBO Stability

## Causality

A discrete-time system is **causal** if the system output only depends on **present** or **previous** input and output samples. In other words, the system does not depend on future samples.

## BIBO stability

- A discrete-time system $T$ is *bounded-input bounded-output stable* (BIBO stable) if and only if for any $x[n]$ such that $|x[n]| < \beta$ for all $n$, $|T(x[n])| < \alpha$ for all $n$ where $0 \leq \alpha < \infty,\ 0 \leq \beta < \infty$.
- In words, a system is BIBO stable if and only if the system produces a bounded output signal for any bounded input signal. A signal is bounded if it takes a finite value for all indices $n \in \mathbb{Z}$.

# Impulse Response

## Impulse response

The *impulse response* $h[n]$ of a discrete-time system $T$ is the response or output of the system to a unit impulse or Kronecker delta $\delta[n]$ input:

$$h[n] = T(\delta[n])$$

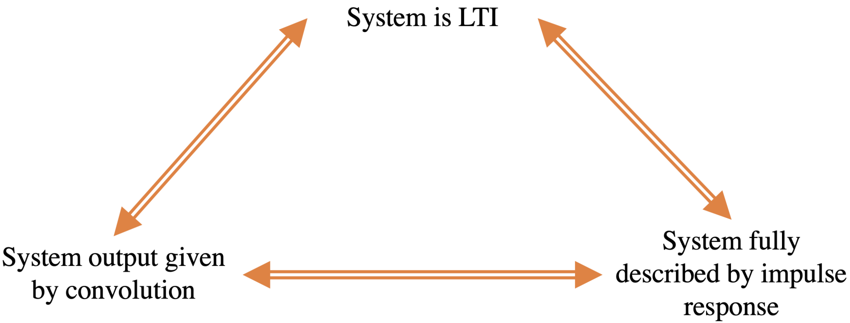
- We may observe the impulse response of any discrete-time system by letting the input signal $x[n] = \delta[n] = \{\cdots, 0, 0, \underset{\uparrow}{1}, 0, 0, \cdots\}$.

# Convolution

## Convolution

The system response $y[n]$ of an LTI system with impulse response $h[n]$ to input signal $x[n]$ is given by the *convolution* of $x[n]$ and $h[n]$, $y[n] = x[n] * h[n]$

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k-\infty}^{\infty} h[k]x[n-k] \tag{1}$$



# Key Properties of Convolution and Impulse Response

- **Commutativity**:

$$x[n] * h_1[n] * h_2[n] = x[n] * h_2[n] * h_1[n] = h_1 * x[n] * h_2[n] = \cdots$$

- **Distributivity**:

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

- **Length**: The convolution of length-$N$ $x[n]$ with length-$M$ $h[n]$ results in the length-$(N + M - 1)$ signal $y[n] = x[n] * h[n]$.

- **Start Point and End Points**: Suppose $x[n]$ has start and end indices (first and last non-zero value locations) $n_0$ and $n_1$, respectively; $h[n]$ has start and end indices $k_0$ and $k_1$; then, $y[n] = x[n] * h[n]$ will have start and end indices:

$$\begin{aligned} \text{Start Index} &= n_0 + k_0 \\ \text{End Index} &= n_1 + k_1 \end{aligned}$$

# Key Properties of Convolution and Impulse Response

- **Identity**:

$$x[n] * \delta[n] = x[n]$$

- **Causality**: An LTI system with impulse response $h[n]$ is causal if and only if

$$h[n] = 0, \;\; n < 0.$$

- **Stability**: An LTI system with impulse response $h[n]$ is BIBO stable if and only if

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty.$$

# Cases for Computing Convolution

Three distinct cases for computing the convolution of two discrete-time signals:

- Refer to Lecture 4 for more details on computing convolution!)

1. Two finite-length signals
   - Apply the table method, matrix-vector multiplication method, or similar methods.

2. Two infinite-length signals
   - Apply the convolution sum or move into the $z$-domain.

3. One finite-length signal and one-infinite length signal
   - Apply LTI system properties, i.e. identity, linearity, time-invariance, to express output as combination of the infinite-length signal.

# Linear Constant Coefficient Difference Equations

## Linear Constant Coefficient Difference Equations

- *Linear Constant Coefficient Difference Equations* (LCCDE) are a popular representation of LTI systems.
- The system output $y[n]$ is a linear combination of system inputs and outputs.

$$y[n] = \sum_{i=1}^{K} b_i y[n-i] + \sum_{j=0}^{M-1} c_j x[n-j], \; 0 \leq K < \infty, \; 1 \leq M < \infty$$

- $K$: number of output or *feedback* terms.
- $M$: number of input terms
- Couple of notes:
  1. Both $K$ and $M$ are finite while $M \geq 1$.
  2. We may express LCCDEs for non-causal systems as well.

# $z$-transform

## $z$-transform

The $z$-transform $X(z)$ of a discrete-time signal $x[n]$ is given by

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

$$x[n] \overset{\mathcal{Z}}{\mapsto} X(z)$$

We must always specify a *region of convergence* to make a $z$-transform unique!

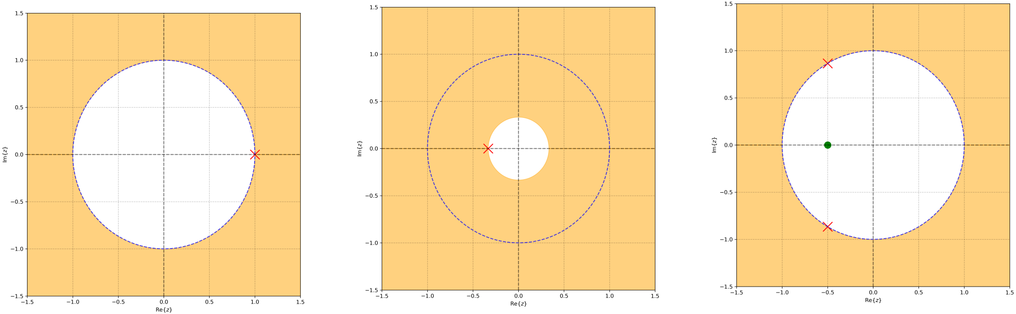
## Region of Convergence

The *region of convergence (ROC)* is the values of $z$ where the $z$-transform sum converges for a given signal.

# Pole-Zero Plots

For a $z$-transform $X(z)$,

- The values of $z$ for which $X(z) = 0$ are known as *zeros*.
- The values of $z$ for which $X(z) \to \infty$ are known as *poles*.
- *Pole-zero plots* visualize the ROC, poles, and zeros of a $z$-transform.




(a) $H(z) = \frac{1}{1-z^{-1}}$ (b) $H(z) = \frac{1}{1+\frac{1}{3}z^{-1}}$ (c) $H(z) = \frac{1+\frac{1}{2}z^{-1}}{1+z^{-1}+z^{-2}}$

Figure 1: Example pole-zero plots for (a) $x[n] = u[n]$, (b) $x[n] = \left(-\frac{1}{3}\right)^n u[n]$, and (c) $\cos\left(\frac{2}{3}\pi n\right) u[n]$. The dashed blue circle represents the unit-circle.

# Common $z$-transform Pairs

Table 1: Common $z$-transform pairs.

| $x[n]$ | $X(z)$ | ROC |
|---|---|---|
| $\delta[n]$ | $1$ | All $z$ |
| $u[n]$ | $\frac{1}{1-z^{-1}}$ | $\lvert z\rvert > 1$ |
| $a^n u[n]$ | $\frac{1}{1-az^{-1}}$ | $\lvert z\rvert > \lvert a\rvert$ |
| $-a^n u[-(n+1)]$ | $\frac{1}{1-az^{-1}}$ | $\lvert z\rvert < \lvert a\rvert$ |
| $na^n u[n]$ | $\frac{az^{-1}}{(1-az^{-1})^2}$ | $\lvert z\rvert > \lvert a\rvert$ |
| $-na^n u[-(n+1)]$ | $\frac{az^{-1}}{(1-az^{-1})^2}$ | $\lvert z\rvert < \lvert a\rvert$ |
| $\cos(\omega_0 n)u[n]$ | $\frac{1-\cos(\omega_0)z^{-1}}{1-2\cos(\omega_0)z^{-1}+z^{-2}}$ | $\lvert z\rvert > 1$ |
| $\sin(\omega_0 n)u[n]$ | $\frac{\sin(\omega_0)z^{-1}}{1-2\cos(\omega_0)z^{-1}+z^{-2}}$ | $\lvert z\rvert > 1$ |
| $a^n\cos(\omega_0 n)u[n]$ | $\frac{1-a\cos(\omega_0)z^{-1}}{1-2a\cos(\omega_0)z^{-1}+a^2z^{-2}}$ | $\lvert z\rvert > \lvert a\rvert$ |
| $a^n\sin(\omega_0 n)u[n]$ | $\frac{a\sin(\omega_0)z^{-1}}{1-2a\cos(\omega_0)z^{-1}+a^2z^{-2}}$ | $\lvert z\rvert > \lvert a\rvert$ |

# Summary of Key $z$-transform Properties

| Property | Signal | $z$-transform | ROC |
|---|---|---|---|
| Time-shifting | $x[n-k]$ | $z^{-k}X(z)$ | $R_x$ except $z=0$ or $z=\infty$ |
| Linearity | $ax_1[n]+bx_2[n]$ | $aX_1(z)+bX_2(z)$ | At least $R_{x_1}\cap R_{x_2}$ |
| Convolution | $x_1[n]*x_2[n]$ | $X_1(z)X_2(z)$ | At least $R_{x_1}\cap R_{x_2}$ |
| Differentiation | $nx[n]$ | $-z\frac{dX(z)}{dz}$ | $R_x$ |
| Conjugation | $x^*[n]$ | $X^*(z^*)$ | $R_x$ |
| Time reversal | $x[-n]$ | $X\left(z^{-1}\right)$ | $1/R_x$ |
| Scaling | $a^n x[n]$ | $X\left(z/a\right)$ | $\|a\|R_x$ |
| Real-part | Re$\{x[n]\}$ | $(1/2)[X(z)+X^*(z^*)]$ | At least $R_x$ |
| Imaginary-part | Im$\{x[n]\}$ | $(1/2)[X(z)-X^*(z^*)]$ | At least $R_x$ |

# Computing Inverse $z$-transform: Partial Fraction Decomposition

The $z$-transform for any infinite-length signal may be written in the following form

$$X(z) = \frac{\prod_{k=1}^{N-1}(1 - q_k z^{-1})}{\prod_{k=1}^{M}(1 - p_k z^{-1})} = \sum_{k=1}^{M} \frac{A_k}{1 - p_k z^{-1}}$$

where the inverse $z$-transform of each $\frac{A_k}{1-p_k z^{-1}}$ may be easily found by inspection. To solve for the $A_k$ values, we apply *partial fraction decomposition*.

## Partial Fraction Decomposition

Suppose we have $z$-transform $X(z)$:

1. Factorize the denominator of $X(z)$ like on the left side of the above equation to identify each pole of the system. Set this equal to the right side where each pole has it's own rational expression and $A_k$ coefficient.
2. Multiply both sides by the shared denominator of $X(z)$.
3. Solve for each $A_k$ value by setting $z = p_k$.

# LTI System Response and the $z$-transform

For an LTI system with impulse response $h[n]$, let $H(z)$ denote the system's transfer function. We know thus far that

$$y[n] = x[n] * h[n].$$

We may express the input-output relationship of LTI systems a second way using the convolution property of the $z$-transform:

$$Y(z) = X(z)H(z), \text{ ROC } = \text{ at least } R_x \cap R_h.$$

An important consequence of the convolution property is that we see another way to express the transfer function of an LTI system:

$$H(z) = \frac{Y(z)}{X(z)}.$$

# Transfer Functions and LCCDEs

Recall how we define general Linear Constant Coefficient Difference Equations (LCCDE):

$$y[n] + \sum_{k=1}^{N} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k].$$

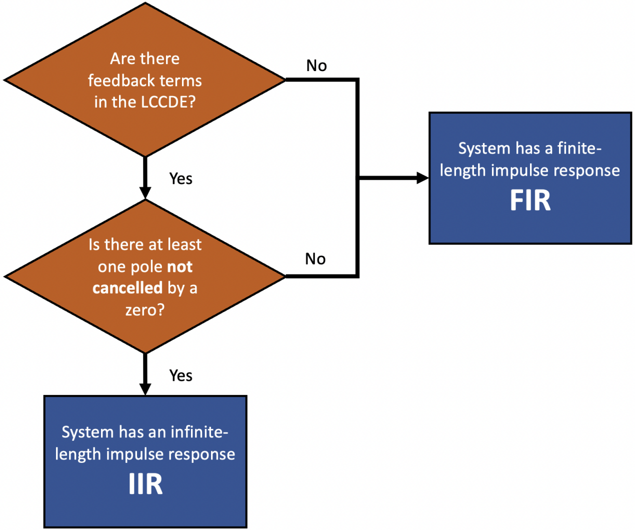
If we take the $z$-transform of both sides, we may obtain the transfer function $H(z)$ of the system.

$$Y(z) + \sum_{k=1}^{N} a_k z^{-k} Y(z) = \sum_{k=0}^{M-1} b_k z^{-k} X(z)$$

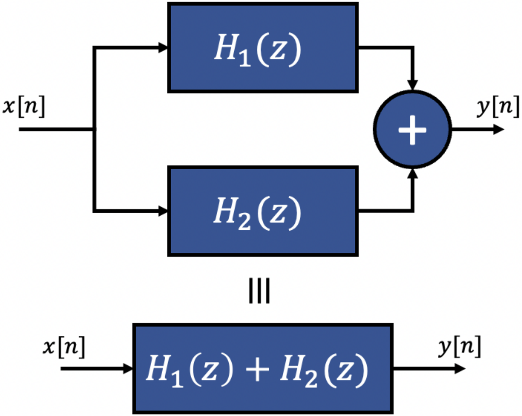
$$\frac{Y(z)}{X(z)} = H(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

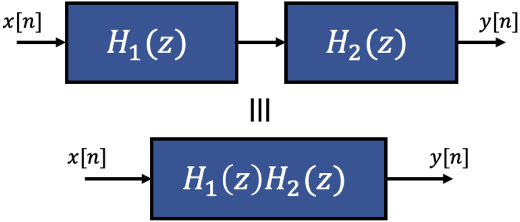
Similarly, we may take the inverse $z$-transform of a transfer function to determine the corresponding LCCDE!

# Characterizing Transfer Functions



# System Algebra



# BIBO Stability in the $z$-domain

- An LTI system is BIBO stable if and only if the ROC of the system's transfer function contains the unit circle, i.e. $|z| = 1$ resides inside the ROC.
  - For example, ROCs including $|z| > \frac{1}{2}; |z| < 2; \frac{1}{3} < |z| < 3$ all represent BIBO stable systems.
- An LTI system with an ROC on the boundary of the unit circle, e.g. $|z| > 1; |z| < 1$ is referred to as *marginally stable* and is still classified as an unstable system.
- Please refer to Lecture 11 for further details regarding ROCs and determining bounded input signals that produce unbounded outputs for unstable/marginally stable systems!

# Relating BIBO Stability, Causality, and $z$-domain: Summary

| Left-sided | Left-sided | Right-sided | Right-sided |
|---|---|---|---|
| Anti-causal | Non-causal | Non-causal | Causal |
| $n_0 < 0$ | $n_0 \geq 0$ | $n_0 < 0$ | $n_0 \geq 0$ |

| ROC shape of $H(z)$ | Causality Type | Right-sided | Left-sided | Condition for stability |
|---|---|---|---|---|
| $\lvert z\rvert > \lvert p_{\max}\rvert$ | Causal | ✓ | ✗ | $\lvert p_{\max}\rvert < 1$ |
| $\lvert z\rvert < \lvert p_{\min}\rvert$ | Anti-causal | ✗ | ✓ | $\lvert p_{\min}\rvert > 1$ |
| $\lvert p_{\max}\rvert < \lvert z\rvert < \infty$ | Non-causal | ✓ | ✗ | $\lvert p_{\max}\rvert < 1$ |
| $0 < \lvert z\rvert < \lvert p_{\min}\rvert$ | Non-causal | ✗ | ✓ | $\lvert p_{\min}\rvert > 1$ |
| $a < \lvert z\rvert < b$ | Non-causal | ✓ | ✓ | $\lvert p_{\max,h_r}\rvert = a < 1$ and $\lvert p_{\min,h_l}\rvert = b > 1$ |

# Discrete-time Fourier Transform

## Continuous-time Fourier Transform

The continuous-time Fourier transform (CTFT) of the signal $x(t)$ is denoted by $X(\Omega)$. Similarly, we often write $x(t) \overset{\mathcal{F}}{\longleftrightarrow} X(\Omega)$.

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

The CTFT of signal $x(t)$ exists if

$$\int_{-\infty}^{\infty} |x(t)|dt < \infty.$$

## Discrete-time Fourier Transform

The discrete-time Fourier transform (DTFT) of a discrete-time signal $x[n]$ is denoted as $X_d(\omega)$: $x[n] \overset{\mathcal{F}}{\longleftrightarrow} X_d(\omega)$.

$$X_d(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_d(\omega)e^{j\omega n}d\omega$$

The DTFT of the signal $x[n]$ exists if

$$\sum_{n=-\infty}^{\infty} |x[n]| < \infty.$$

# Common DTFT Pairs

| Signal $x[n]$ | DTFT $X_d(\omega)$ |
|---|---|
| $\delta[n]$ | 1 |
| $u[n]$ | $\frac{1}{1-e^{-j\omega}} + \pi\delta(\omega)$ |
| $a^n u[n]$ | $\frac{1}{1-ae^{-j\omega}},\ 0 < \|a\| < 1$ |
| $e^{j\omega_0 n}$ | $2\pi\delta(\omega-\omega_0)$ |
| $\cos(\omega_0 n)$ | $\pi\left[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\right]$ |
| $\sin(\omega_0 n)$ | $-j\pi\left[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)\right]$ |
| $\text{rect}\left(\frac{n-k}{L}\right)$ | $\frac{\sin\left(\frac{L\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}e^{-j\omega k}$ |
| $\text{sinc}(Ln)$ | $\frac{\pi}{L}\text{rect}\left(\frac{\omega}{2L}\right)$ |
| $\text{sinc}^2(Ln)$ | $\frac{\pi}{L}\Delta\left(\frac{\omega}{2L}\right)$ |

# DTFT Properties Table

| Property | Signal | DTFT |
|---|---|---|
| Linearity | $ax_1[n] + bx_2[n]$ | $aX_1(\omega) + bX_2(\omega)$ |
| Time shifting | $x[n-k]$ | $X_d(\omega)e^{-jk\omega}$ |
| Frequency shifting | $e^{j\omega_0 n}x[n]$ | $X_d(\omega - \omega_0)$ |
| Modulation | $x[n]\cos(\omega_0 n)$ | $\frac{1}{2}X_d(\omega-\omega_0) + \frac{1}{2}X_d(\omega+\omega_0)$ |
| Time reversal | $x[-n]$ | $X_d(-\omega)$ |
| Conjugation | $x^*[n]$ | $X_d^*(-\omega)$ |
| Differentiation | $nx[n]$ | $-j\frac{dX_d(\omega)}{d\omega}$ |
| Convolution | $x[n] * h[n]$ | $X_d(\omega)H_d(\omega)$ |
| Windowing | $x[n]w[n]$ | $\frac{1}{2\pi}X_d(\omega) * W_d(\omega)$ |
| Hermitian symmetry | $x[n]$ real | $X_d^*(\omega) = X_d(-\omega)$ |
| Parseval's relation | $\sum_{n=-\infty}^{\infty} \|x[n]\|^2$ | $\frac{1}{2\pi}\int_{2\pi} \|X_d(\omega)\|^2 d\omega$ |

# Highlighting Key Properties

The following properties are especially important when working with the DTFT:

- $2\pi$–periodicity: For any DTFT $X_d(\omega)$,

$$X_d(\omega) = X_d(\omega + 2\pi k),\ \ k \in \mathbb{Z}.$$

- Hermitian symmetry: A signal $x[n] \overset{\mathcal{F}}{\longleftrightarrow} X_d(\omega)$ is real-valued if and only if

$$X_d^*(\omega) = X_d(-\omega)$$

  This is equivalent to

$$|X_d(\omega)| = |X_d(-\omega)| \ \text{ and } \ \angle X_d(\omega) = -\angle X_d(-\omega)$$

- Convolution property:

$$x[n] * h[n] \overset{\mathcal{F}}{\longleftrightarrow} X_d(\omega) H_d(\omega)$$

- Multiplication in time or windowing:

$$x[n]w[n] \overset{\mathcal{F}}{\longleftrightarrow} \frac{1}{2\pi} X_d(\omega) * W_d(\omega)$$

# LTI System Response to Periodic Signals Summary

$$X(z) \longrightarrow \boxed{H(z)} \longrightarrow X(z)H(z)$$

$$X(\omega) \longrightarrow \boxed{H(\omega)} \longrightarrow X(\omega)H(\omega)$$

$$Az^n \longrightarrow \boxed{H(z)} \longrightarrow H(z)Az^n$$

$$Ae^{j\omega_0 n} \longrightarrow \boxed{H(\omega)} \longrightarrow H(\omega_0)Ae^{j\omega_0 n}$$

$$A\cos(\omega_0 n + \theta) \longrightarrow \boxed{H(\omega)} \longrightarrow |H(\omega_0)|A\cos(\omega_0 n + \theta + \angle H(\omega_0))$$

$h[n]$ real-valued

# Magnitude and Phase Response

## Magnitude Response

The *magnitude response* of an LTI system is the magnitude of the system's frequency response for each $\omega \in [-\pi, \pi]$:

$$|H_d(\omega)| = \left|\sqrt{H_d(\omega)H_d^*(\omega)}\right|$$

- Remember that magnitude is always non-negative! $(|H_d(\omega)| \geq 0)$

## Phase Response

The *phase response* of an LTI system is the phase of the system's frequency response for each $\omega \in [-\pi, \pi]$.

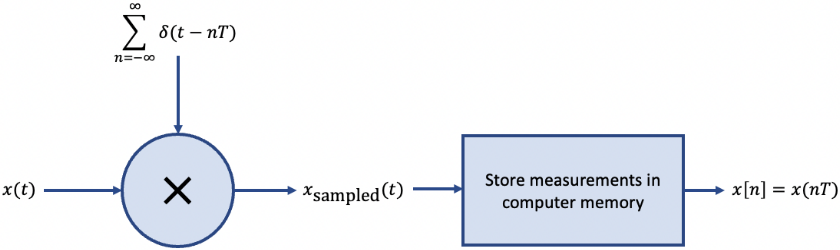
$$\angle H_d(\omega) = \tan^{-1}\left(\frac{\mathrm{Im}\{H_d(\omega)\}}{\mathrm{Re}\{H_d(\omega)\}}\right)$$

- Changes in sign for $H_d(\omega)$ let to $\pm\pi$ jumps in phase, $-1 = e^{\pm j\pi}$!

Please refer to Lecture 16 for examples of determining and plotting frequency, magnitude, and phase responses.

# Sampling, A/D Conversion, D/A Conversion

# Ideal Analog-to-Digital Conversion




$$x[n] = x_a(nT) \qquad \omega = \Omega T \qquad x_a(t) \overset{\mathcal{F}}{\longleftrightarrow} X_a(\Omega)$$

$$X_s(\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a\left(\Omega - \frac{2\pi}{T} k\right) \qquad x_s(t) \overset{\mathcal{F}}{\longleftrightarrow} X_s(\Omega)$$

$$X_d(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a\left(\frac{\omega - 2\pi k}{T}\right) \qquad x[n] \overset{\mathcal{F}}{\longleftrightarrow} X_d(\omega)$$

# Nyquist Criterion

## Nyquist criterion

A bandlimited continuous-time signal with maximum linear frequency $B$ can be sampled at sampling period $T$ to produce $x[n]$ such that $x_a(t)$ may be recovered from $x[n]$ if

$$T < \frac{1}{2B}$$
$$f_s > 2B$$

Otherwise, we will incur *aliasing* when obtaining $x[n]$. The frequency $f_{\text{nyquist}} = 2B$ (units in Hz) is known as the *Nyquist rate*.

# Aliasing

## Aliasing

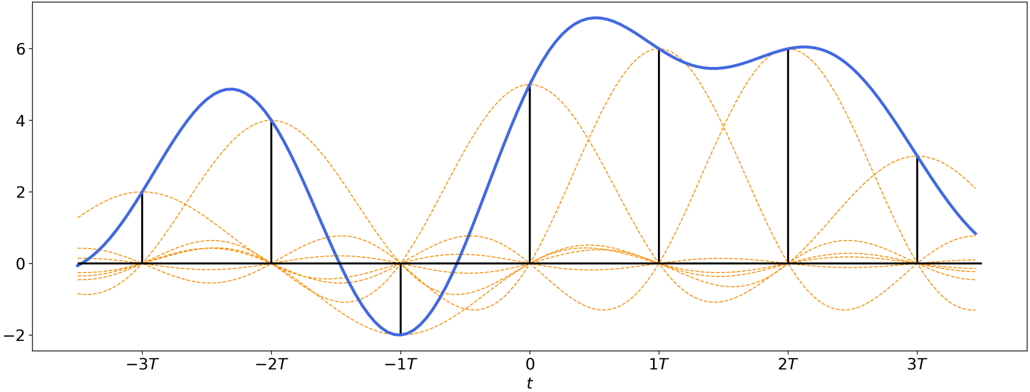
- Bandlimited analog signals $x_a(t)$ sampled *above* the Nyquist rate may be perfectly recovered from $x[n]$ via ideal digital-to-analog conversion.
- Otherwise, i.e. sampling below the Nyquist rate, the sampled signal $x[n]$ will have an incorrect, alternative appearance or *alias*. The resulting $x[n]$ is said to exhibit *aliasing*.

Please refer to Lecture 19 for a detailed example demonstrating the aliasing effect with sampling a single sinusoid.

# Ideal D/A Conversion: Time-domain

Recall that ideal digital-to-analog (D/A) conversion is achieved by sinc interpolation.

$$x_a(t) = \sum_{n=-\infty}^{\infty} x[n]\mathrm{sinc}\left(\frac{\pi}{T}(t - nT)\right)$$

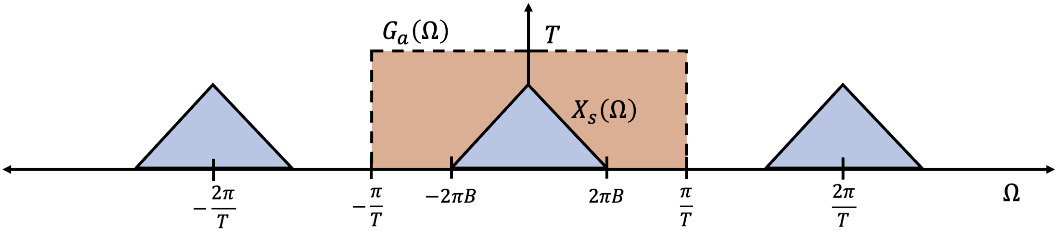


# Ideal D/A Conversion: Frequency-domain

$$X_a(\Omega) = X_s(\Omega) G_a(\Omega)$$

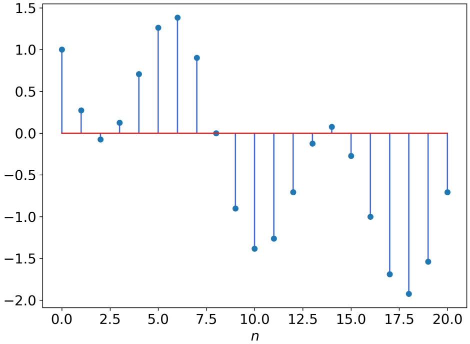
$$X_s(\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a\left(\Omega - \frac{2\pi}{T} k\right)$$

$$G_a(\Omega) = T \mathrm{rect}\left(\frac{\Omega}{\frac{2\pi}{T}}\right)$$

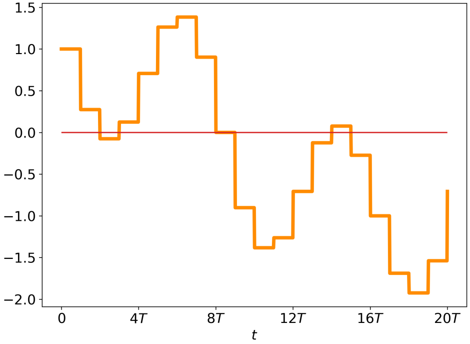


Ideal sinc interpolation is impractical since it is (1) non-causal and (2) not BIBO stable. As a practical alternative, we consider the *zero-order hold* (ZOH) D/A converter.

$$g_{\text{zoh}}(t) = \begin{cases} 1, & 0 \leq t \leq T \\ 0, & \text{else} \end{cases}, \qquad x_r(t) = \sum_{n=-\infty}^{\infty} x[n] g_{\text{zoh}}(t - nT)$$




(a) $x[n]$



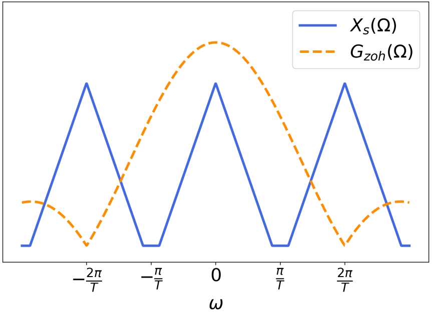

(b) $x_r(t) = \sum_{n=-\infty}^{\infty} x[n] g_{\text{zoh}}(t - nT)$

# Practical D/A: Zero-order Hold

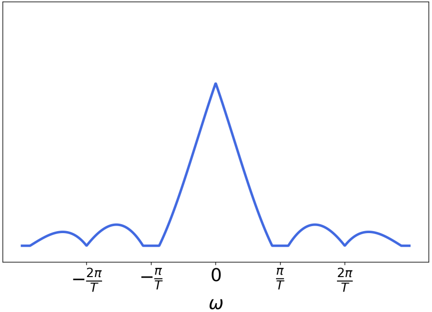
In the frequency domain, we have

$$g_{\text{zoh}}(t) = \begin{cases} 1, & 0 \leq t \leq T \\ 0, & \text{else} \end{cases} \overset{\mathcal{F}}{\longleftrightarrow} G_{\text{zoh}}(\Omega) = Te^{-j\frac{\Omega T}{2}} \operatorname{sinc}\left(\frac{\Omega T}{2}\right)$$

$$X_r(\Omega) = X_s(\Omega) G_{\text{zoh}}(\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a\left(\Omega - \frac{2\pi}{T}k\right) G_{\text{zoh}}(\Omega)$$




(a) $|X_s(\Omega)|$ and $|G_{\text{zoh}}(\Omega)|$




(b) $|X_r(\Omega)|$

# Compensation Filter

$x_r(t) \neq x_a(t)$ after applying a zero-order hold:

- In time, we only have a piecewise constant approximation of $x_a(t)$.
- In frequency, we see droop in the true $X_a(\Omega)$ and spurious frequencies in $X_r(\Omega)$.

These issues are corrected by an analog *compensation* or *reconstruction filter*, $H_r(\Omega)$ such that

$$|H_r(\Omega)| = \begin{cases} \frac{1}{\text{sinc}\left(\frac{\Omega T}{2}\right)}, & |\Omega| < \frac{\pi}{T} \\ 0, & |\Omega| \geq \frac{\pi}{T} \end{cases}$$

.

Please refer to Lecture 32 for more details and examples regarding the transition bandwidth of the compensation filter after oversampling or upsampling.

# Discrete Fourier Transform

## Discrete Fourier transform (DFT)

The discrete Fourier transform of a length–$N$ discrete-time signal is denoted by $X[k]$.

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

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

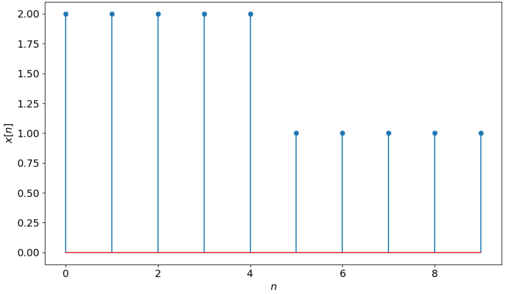
- Note that $X[k]$ is defined for $0 \leq k \leq N-1$!

Recall that we can also zero-pad $x[n]$ such that $X[k]$ evaluates more points of the DTFT of $x[n]$. Please refer to Lecture 20 for further explanation and examples of zero-padding!

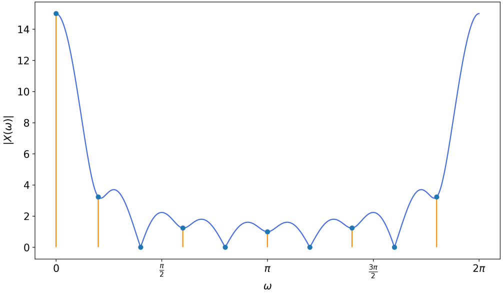
# Relationship between DTFT and DFT

Given a finite-length signal $x[n] \stackrel{\mathcal{F}}{\longleftrightarrow} X_d(\omega)$, the DFT evaluates the DTFT at discrete frequencies $\omega_k = \frac{2\pi k}{N}$ for $0 \leq k \leq N-1$.

$$X[k] = X_d(\omega_k) = X_d\left(\frac{2\pi k}{N}\right), \; 0 \leq k \leq N-1$$




(a) Finite-length $x[n]$




(b) $X_d(\omega)$ and values of $X[k]$

# Periodicity of the DFT

For a length–$N$ signal $x[n]$ and its DFT $X[k]$:

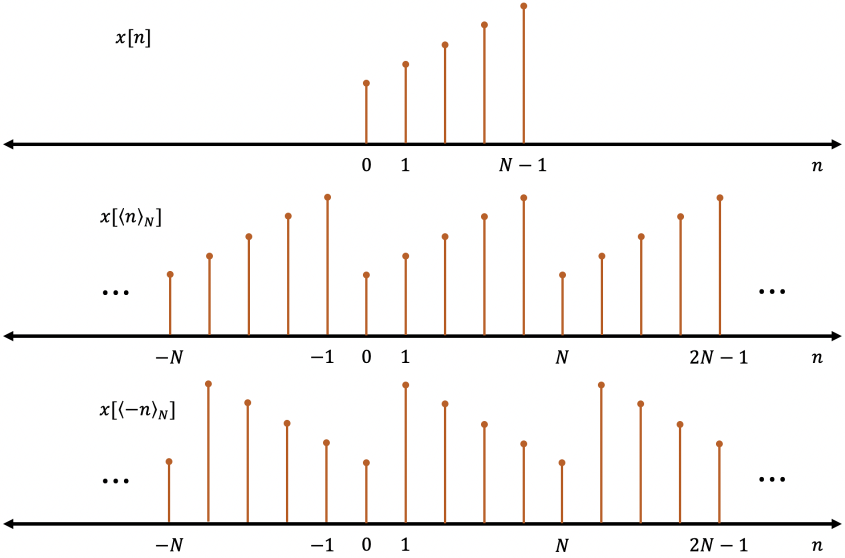
- $N$**-periodicity in frequency**:

$$X[k] = X[k + mN] = X[\langle k \rangle_N]$$

- $N$**-periodicity in time:**

$$x[n] = x[n + mN] = x[\langle n \rangle_N]$$

# Periodic Extension



# Summary of DFT Properties

Many other DFT properties follow from DTFT properties except with **circularity**, e.g. circular time shifting, circular time reversal.

| Property | Length-$N$ $x[n]$ | DFT $X[k]$ |
|---|---|---|
| Linearity | $ax_1[n] + bx_2[n]$ | $aX_1[k] + bX_2[k]$ |
| Circular time shifting | $x[\langle n-m\rangle_N]$ | $e^{-j\frac{2\pi mk}{N}} X[k]$ |
| Circular frequency shifting | $e^{j\frac{2\pi mn}{N}} x[n]$ | $X[\langle k-m\rangle_N]$ |
| Circular modulation | $\cos\left(\frac{2\pi m}{N} n\right) x[n]$ | $\frac{1}{2}X[\langle k+m\rangle_N] + \frac{1}{2}X[\langle k-m\rangle_N]$ |
| Circular time reversal | $x[\langle -n\rangle_N]$ | $X[\langle -k\rangle_N]$ |
| Conjugation | $x^*[n]$ | $X^*[\langle -k\rangle_N]$ |
| Duality | $X[n]$ | $Nx[\langle -k\rangle_N]$ |
| Circular convolution | $x[n] \circledast h[n]$ | $X[k]H[k]$ |
| Windowing | $x[n]w[n]$ | $\frac{1}{N} X[k] \circledast W[k]$ |
| Parseval's theorem | $\sum_{n=0}^{N-1} x[n]y^*[n]$ | $\frac{1}{N}\sum_{k=0}^{N-1} X[k]Y^*[k]$ |
| Parseval's relation | $\sum_{n=0}^{N-1} \lvert x[n]\rvert^2$ | $\frac{1}{N}\sum_{k=0}^{N-1} \lvert X[k]\rvert^2$ |

# Circular Convolution

## Circular Convolution Property

For length–$N$ finite-length sequences $x[n]$ and $h[n]$,

$$x[n] \circledast h[n] \overset{\text{DFT}}{\longleftrightarrow} X[k]H[k]$$

where

$$x[n] \circledast h[n] = \sum_{m=0}^{N-1} x[m]h[\langle n-m\rangle_N] = \sum_{m=0}^{N-1} h[m]x[\langle n-m\rangle_N]$$

- Circular convolution is performed between equal-length sequences. If one sequence is shorter, we zero-pad this sequence to match the length of the longer sequence.
- We sometimes denote circular convolution with $\circledast_N$ to denote that both sequences are padded to length–$N$ as necessary.

Please refer to Lecture 21 for examples of computing circular convolution!

# Spectral Analysis Problem

## Spectral analysis

Spectral analysis is a common application of digital signal processing where we seek to identify unknown sinusoidal components of a finite-length signal.

$$x[n] = \sum_{m=1}^{M} A_m \cos(\omega_m n), \; 0 \le n \le N-1$$

Unknowns include:

- Number of spectral components $M$.
- Frequencies $\omega_m, \; 0 \le m \le M-1$.
- Amplitudes $A_m, \; 0 \le m \le M-1$.

# DTFT of Finite-length Sinusoid

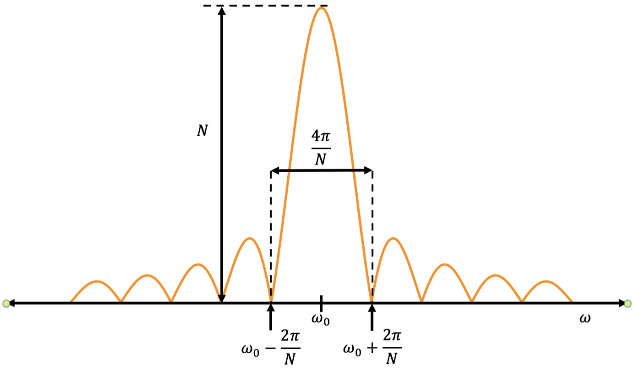
Let $x[n] = A\cos(\omega_0 n),\ 0 \leq n \leq N-1.$

Then,

$$X_d(\omega) = \frac{A}{2}C(\omega - \omega_0) + \frac{A}{2}C(\omega + \omega_0),$$

where

$$C(\omega) = e^{-j\frac{\omega}{2}(N-1)}\left(\frac{\sin\left(\frac{N}{2}\omega\right)}{\sin\left(\frac{1}{2}\omega\right)}\right)$$

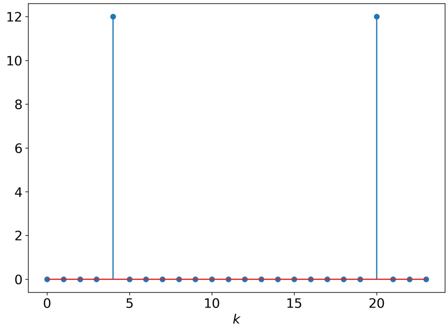



Magnitude spectrum of $C(\omega - \omega_0)$.

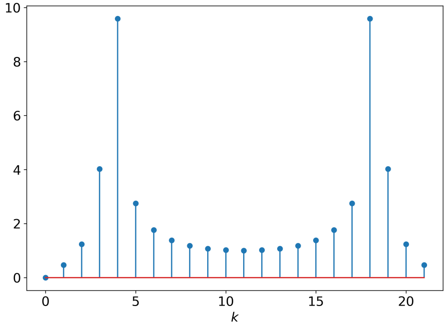
# DFT of Finite-length Sinusoid

Consider two sinusoidal signals: one with an integer number of cycles ($x_1[n]$) and another with a non-integer number of cycles ($x_2[n]$).

$$x_1[n] = \cos\left(\frac{\pi}{3}n\right), 0 \le n \le 23, \qquad x_2[n] = \cos\left(\frac{\pi}{3}n\right), 0 \le n \le 21$$




(a) $X_1[k]$




(b) $X_2[k]$

# Main Lobe Separation Criteria

One of the two key challenges in spectral analysis is the ability to resolve frequencies that are close to one another, i.e. the main lobes of each component may overlap.

## Lobe separation criteria

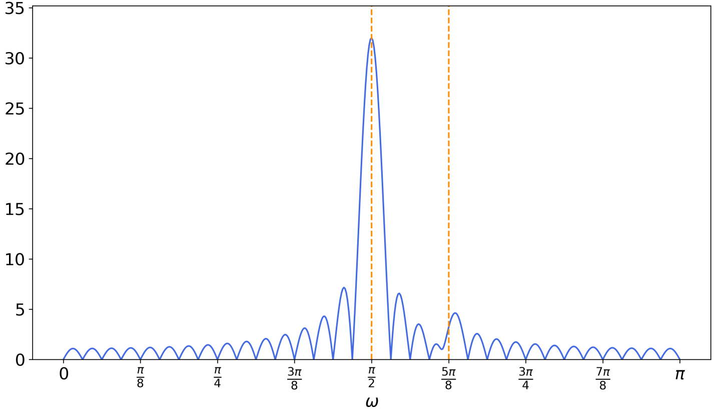
For a given pair of spectral components with frequencies $\omega_1$ and $\omega_2$ in a length–$N$ signal, we have two main lobe separation criteria.

- Full lobe separation:
$$N > \frac{4\pi}{|\omega_1 - \omega_2|}$$
- Half lobe separation:
$$N > \frac{2\pi}{|\omega_1 - \omega_2|}$$

# Side Lobe Reduction

Consider a case where two components have very different magnitudes.

$$s[n] = \cos(0.5\pi n) + \frac{1}{10}\cos(0.625\pi n),\ \ 0 \le n \le 63$$




Magnitude spectrum $|S(\omega)|$. Orange dashed lines identify $\omega_1$ and $\omega_2$.

# Side Lobe Reduction

The other key challenge is spectral components having large differences in magnitudes; thus, the side lobes of one component may mask the main lobe of smaller components. Recall that the DTFT of finite-length signals can be described as the product of an infinite-length signal with a *window function*.

$$
\begin{aligned}
s[n] &= \cos(0.5\pi n) + \frac{1}{10}\cos(0.625\pi n), \; 0 \le n \le 63 \\
s[n] &= \underbrace{\left(\cos(0.5\pi n) + \frac{1}{10}\cos(0.625\pi n)\right)}_{x[n]} \underbrace{(u[n] - u[n-64])}_{w[n]} \\
&= x[n]w[n]
\end{aligned}
$$

Thus,

$$
s[n] = x[n]w[n] \stackrel{\mathcal{F}}{\longleftrightarrow} S_d(\omega) = \frac{1}{2\pi} X_d(\omega) * W_d(\omega)
$$

$S_d(\omega)$ is then shifted and scaled copies of the window function's DTFT $W_d(\omega)$!

# Window Function Tradeoff

The rectangular and Hamming windows present a tradeoff between *main lobe width* and *side lobe attenuation/height*. Ideally, we want narrow main lobes, i.e. easier to separate spectral components, and low side lobes, i.e. less likely for side lobes to hide other components.

| | Rectangular window | Hamming window |
|---|---|---|
| Main Lobe Width | $\frac{4\pi}{N}$ | $\frac{8\pi}{N}$ |
| Side Lobe Attenuation | -13 dB | -42 dB |

# Fast Fourier Transform

## Fast Fourier transform

The *fast Fourier transform* (FFT) is a class of algorithms that compute the discrete Fourier transform of a length–$N$ signal with $\mathcal{O}(N \log N)$ computational (time) complexity.

## Twiddle factor

The *twiddle factor* is a convenient choice of notation for expressing the DFT and examining FFT algorithms.

- $W = e^{-j2\pi}$, $W_N = e^{-j\frac{2\pi}{N}}$
- $W^k = e^{-j2\pi k}$, $W_N^k = e^{-j\frac{2\pi k}{N}}$

Thus, we may write the DFT using the twiddle factor.

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

# Summary of Decimation-in-time FFT

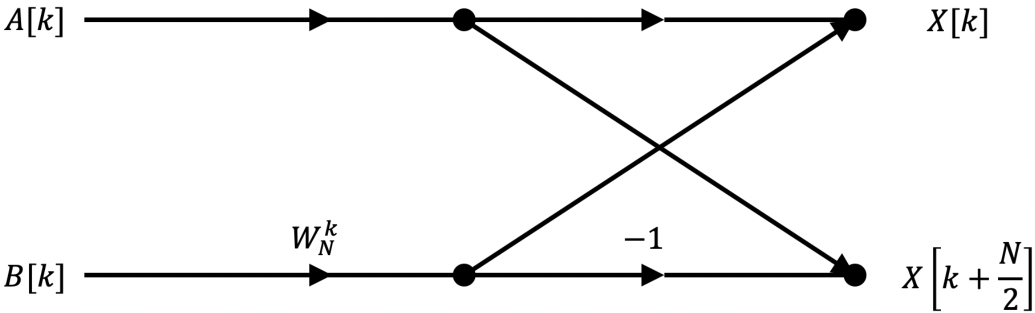
For length–$N$ signal $x[n]$:

1. Recursively divide $x[n]$ in half by the even and odd indices
2. Repeat Step 1 until base case of length–1 sequences where the DFT is trivial.
3. Merge pairs of length–$\frac{L}{2}$ DFTs to length–$L$. **Merge DFTs that came from splitting the same sequence.** Repeat Step 3 until the merging length $L = N$.

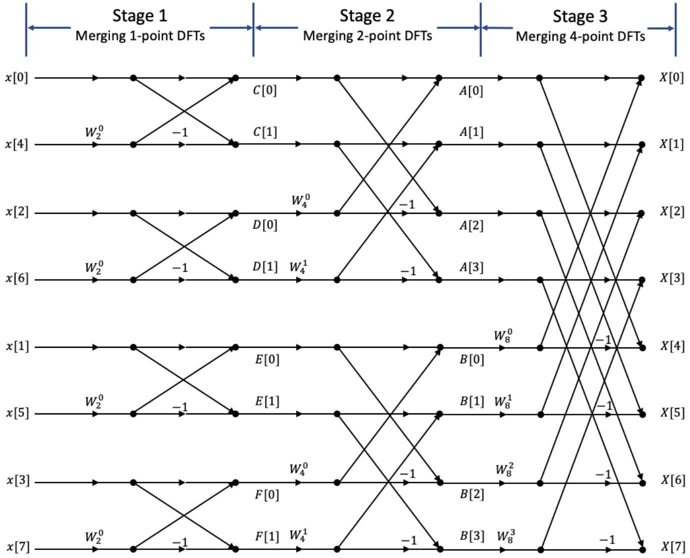
$$C[k] = A[k] + W_L^k B[k],\ 0 \leq k \leq \frac{L}{2} - 1$$

$$C\left[k + \frac{L}{2}\right] = A[k] - W_L^k B[k],\ 0 \leq k \leq \frac{L}{2} - 1$$

*Butterfly diagrams* are a popular visualization for FFT algorithms to demonstrate the merging equations.



# Length–8 DFT Butterfly Diagram



# Fast Linear Convolution Algorithm

Let $x[n]$ and $h[n]$ be length–$N$ and length–$L$ signals, respectively. The convolution sum will require $\mathcal{O}(N^2)$ multiply-add operations. We may instead compute $y[n] = x[n] * h[n]$ in $\mathcal{O}(N \log N)$ complexity in the following steps utilizing the DFT.

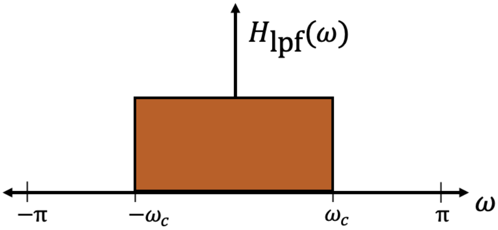
1. Zero-pad $x[n]$ and $h[n]$ each to length–$(N + L - 1)$ to create $x_{\text{zp}}[n]$ and $h_{\text{zp}}[n]$.
   **Note:** We may zero-pad to $2^k \geq N + L - 1$ to accommodate the use of a radix–2 FFT.
   - Cost=$\mathcal{O}(1)$ (constant time, does not depend on length)
2. Take the DFT of $x_{\text{zp}}[n]$ and $h_{\text{zp}}[n]$ using the FFT to obtain $X_{\text{zp}}[k]$ and $H_{\text{zp}}[k]$.
   - Cost=$\mathcal{O}(N \log N)$
3. Multiply $X_{\text{zp}}[k]$ and $H_{\text{zp}}[k]$ element-wise to compute $Y_{\text{zp}}[k]$.
   - Cost=$\mathcal{O}(N)$
4. Compute the inverse DFT of $Y_{\text{zp}}[k]$ using the FFT to obtain the output $y[n]$.
   - Cost=$\mathcal{O}(N \log N)$

# Digital Filter Design

# Canonical Ideal Digital Filters

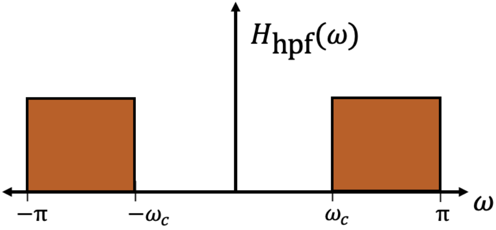
Low-pass Filter (LPF)

$$H_{\mathrm{lpf}}(\omega) = \begin{cases} 1, & |\omega| \leq \omega_c \\ 0, & \omega_c < |\omega| \leq \pi \end{cases}$$




High-pass Filter (HPF)

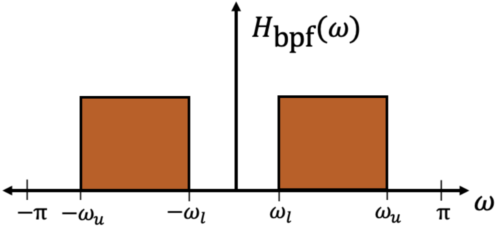
$$H_{\mathrm{hpf}}(\omega) = \begin{cases} 0, & |\omega| \leq \omega_c \\ 1, & \omega_c < |\omega| \leq \pi \end{cases}$$



# Canonical Ideal Digital Filters

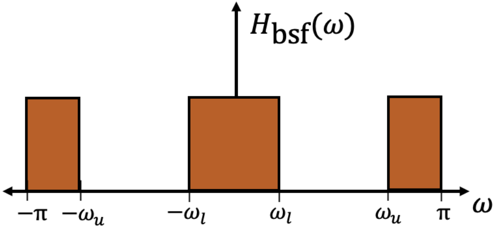
## Band-pass Filter (BPF)

$$H_{\text{bpf}}(\omega) = \begin{cases} 1, & \omega_l \le |\omega| \le \omega_u \\ 0, & \text{otherwise} \end{cases}$$

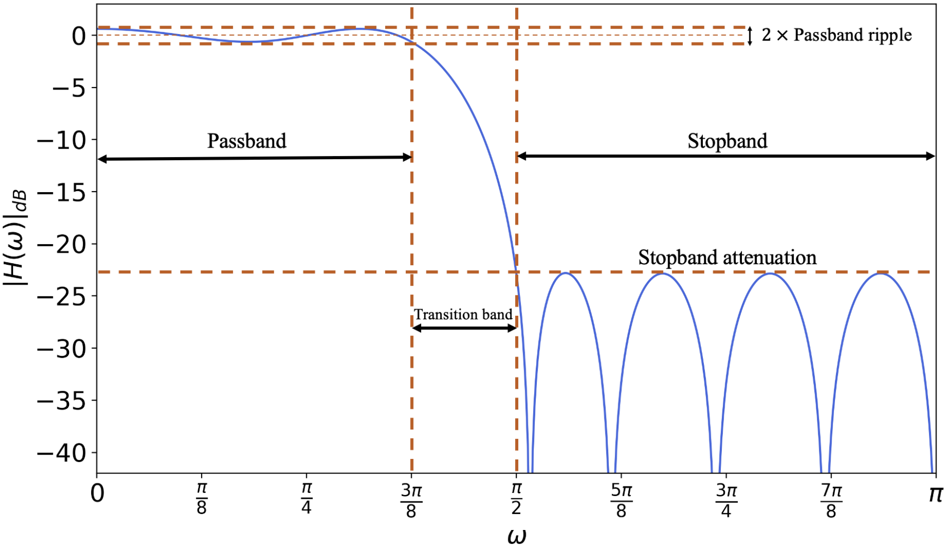



## Band-stop Filter (BSF)

$$H_{\text{bsf}}(\omega) = \begin{cases} 0, & \omega_l \le |\omega| \le \omega_u \\ 1, & \text{otherwise} \end{cases}$$



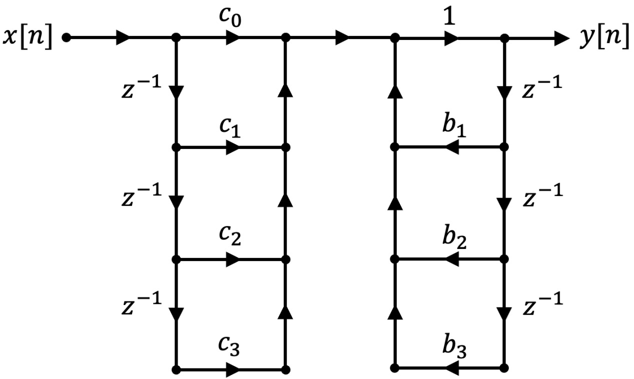
# Practical Digital Filter



# Block Diagrams: Direct Form I

Let our digital filter be given by

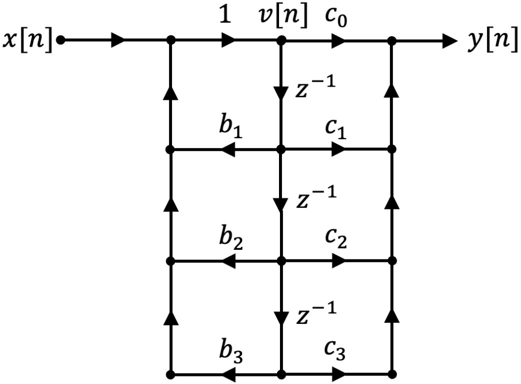
$$y[n] = b_1 y[n-1] + b_2 y[n-2] + b_3 y[n-3] + c_0 x[n] + c_1 x[n-1] + c_2 x[n-2] + c_3 x[n-3].$$



# Block Diagrams: Direct Form II

Let our digital filter be given by

$$y[n] = b_1 y[n-1] + b_2 y[n-2] + b_3 y[n-3] + c_0 x[n] + c_1 x[n-1] + c_2 x[n-2] + c_3 x[n-3].$$



# Group Delay

- The *group delay* $\tau_{\text{gd}}$ of a phase response is given by

$$\tau_{\text{gd}} = -\frac{d\angle H_d(\omega)}{d\omega}$$

  and expresses how an LTI system will delay each frequency component in a signal.
- LTI systems with linear phase responses have *uniform group delay* and delay all frequencies by the same amount.
  - These systems have phase responses of the form $\angle H_d(\omega) = k\omega$.
  - The derivative of the phase response should be constant!
- LTI systems with non-uniform group delay shift different frequency components by different amounts
  - These systems have a phase responses with a non-constant derivative.

# Possible Filter Structures

Summary of the four types of linear-phase FIR filters. We also note whether the frequency response is necessarily zero at $\omega = 0$ or $\omega = \pi$. These conditions may prohibit one of the four canonical filter types from being possible for a given filter type.

| Filter type | Symmetry | Length | $H_d(0)$ | $H_d(\pi)$ | Possible filters |
|---|---|---|---|---|---|
| Type-I | Even | Odd | Can be non-zero | Can be non-zero | LP, HP, BP, BS |
| Type-II | Even | Even | Can be non-zero | Always zero | LP, BP |
| Type-III | Odd | Odd | Always zero | Always zero | BP |
| Type-IV | Odd | Even | Always zero | Can be non-zero | BP, HP |

# Linear Phase vs. Generalized Linear Phase

## Generalized linear phase

Let $H_d(\omega) = R(\omega)e^{j(-\alpha\omega+\beta)}$ be the frequency response of a GLP filter where $R(\omega)$ is the real-valued amplitude response.

An FIR filter is said to have *generalized linear phase* (GLP) if its phase response is linear and differentiable almost everywhere with the same constant slope except for a finite number of constant offsets in phase. These constant offsets may come from

- $\pm\pi$ jumps in phase due to sign changes in $R(\omega)$.
- $\beta \neq 0$, i.e. for Type-III and Type-IV filters, $\beta = \frac{\pi}{2}$ due to the factor of $j$.

Linear phase vs. Generalized linear phase:

- **(Strictly) Linear Phase**: $\angle H_d(\omega) = -\alpha\omega$
- **Generalized Linear Phase**: $\angle H_d(\omega) = -\alpha\omega + \beta(\omega)$
  - $\beta(\omega)$ represents piece-wise constant adjustments in phase, e.g. from $\pm\pi$ jumps.

# Window Method

## Window method for FIR filter design

The window method designs a length–$N$ FIR filter in the following three steps.

1. Derive the ideal (infinite-length) impulse response $d[n]$ corresponding to the desired ideal frequency response $D(\omega)$, i.e. take the inverse DTFT of $D(\omega)$.
2. Shift the ideal impulse response $d[n]$ by $\alpha = (N-1)/2$ to obtain $g[n] = d[n-\alpha]$.
3. Apply window function $w[n]$ to $g[n]$ to finally obtain $h[n] = g[n]w[n]$.

Consider applying this method for a low-pass filter with cutoff $\omega_c$

1. $d[n] = \dfrac{1}{2\pi}\displaystyle\int_{-\pi}^{\pi} D(\omega)e^{j\omega n}d\omega = \dfrac{1}{2\pi}\displaystyle\int_{-\omega_c}^{\omega_c} e^{j\omega n}d\omega = \dfrac{1}{\pi n}\dfrac{e^{j\omega_c n} - e^{-j\omega_c n}}{j2} = \dfrac{\omega_c}{\pi}\mathrm{sinc}(\omega_c n)$
2. $g[n] = \frac{\omega_c}{\pi}\mathrm{sinc}(\omega_c(n-\alpha))$
3. $h[n] = \frac{\omega_c}{\pi}\mathrm{sinc}(\omega_c(n-\alpha))w[n] = \frac{\omega_c}{\pi}\mathrm{sinc}(\omega_c(n-\alpha)),\ 0 \le n \le N-1$

# Window Functions

There are lots of window functions!

- **Rectangular window**

$$w[n] = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & \text{else} \end{cases}$$

- **Hamming window**

$$w[n] = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), & 0 \le n \le N-1 \\ 0, & \text{else} \end{cases}$$

- **Bartlett (triangular) window**

$$w[n] = \begin{cases} \frac{2n}{N-1}, & 0 \le n \le \frac{N-1}{2} \\ \frac{2-2n}{N-1}, & \frac{N-1}{2} \le n \le N-1 \\ 0, & \text{else} \end{cases}$$

- **Hann window**

$$w[n] = \begin{cases} \frac{1}{2} - \frac{1}{2}\cos\left(\frac{2\pi n}{N-1}\right), & 0 \le n \le N-1 \\ 0, & \text{else} \end{cases}$$

The choice of window function trades off the filter transition bandwidth vs. stopband attenuation. Regardless of window function, the transition bandwidth decreases as filter length increases. The stopband attenuation is fixed by the choice of window function.

# Designing Other Filters from Low-pass Filters

Recall that we may design other canonical filter types from an appropriately designed low-pass filter.

- High-pass filters:
  1. Take the complement of a low-pass filter, i.e. $D_{\mathrm{hpf}}(\omega) = 1 - D_{\mathrm{lpf}}(\omega)$.
  2. Modulate an appropriate low-pass filter with $\cos(\pi n) = (-1)^n$.
- Band-pass filters:
  1. Modulate a low-pass filter with $\cos(\omega_0 n)$ for an appropriate low-pass filter and choice of $\omega_0$
- Band-stop filters:
  1. Design an appropriate bandpass filter $D_{\mathrm{bpf}}(\omega)$ then take the complement, i.e. $D_{\mathrm{bsf}}(\omega) = 1 - D_{\mathrm{bpf}}(\omega)$.
  2. Add a low-pass filter and high-pass filter together.

Refer to Lecture 29 for further details and illustrations!

# Multirate Signal Processing

# Downsampling: Time-Domain

## Downsampling in the time-domain

A discrete-time signal $x[n]$ is *downsampled* by an integer factor $D$ by taking every $D$'th sample in $x[n]$.

$$y[n] = x[Dn]$$

- Reduces length of $x[n]$ by factor of $D$.
- Reduces the implicit sampling rate by $\frac{1}{D}$ when performing D/A conversion.

For example, let

$$x[n] = \{\underset{\uparrow}{1}, 2, 3, 4, 5, 6, 7, 8, 9\}.$$

If we downsample $x[n]$ by $D = 3$, we obtain

$$y[n] = \{\underset{\uparrow}{1}, 4, 7\}.$$

# Downsampling: Frequency-Domain

## Downsampling: frequency domain

A discrete-time signal $x[n] \overset{\mathcal{F}}{\longleftrightarrow} X_d(\omega)$ downsampled by factor $D$ will produce $y[n] = x[Dn] \overset{\mathcal{F}}{\longleftrightarrow} Y_d(\omega)$ where

$$Y_d(\omega) = \frac{1}{D} \sum_{k=0}^{D-1} X_d\left(\frac{\omega - 2\pi k}{D}\right).$$

- The height of $X_d(\omega)$ is reduced by $\frac{1}{D}$.
- The spectrum $X_d(\omega)$ is stretched by $D$.
- The summation ensures that $Y_d(\omega)$ satisfies $2\pi$–periodicity of the DTFT.

The above formula is equivalent to (1) reducing the height of $X_d(\omega)$ by $\frac{1}{D}$ then (2) stretching each spectral copy by $D$ **about its center**.

# Anti-aliasing Filter

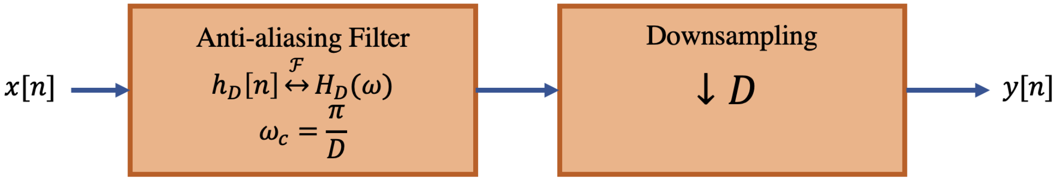
Downsampling a discrete-time signal may cause aliasing as the DTFT spectrum stretches and overlaps between adjacent spectral copies.

## Anti-aliasing filter

To prevent aliasing, we place an *anti-aliasing filter* $H_D(\omega)$ **before** performing the downsampling operation. The structure of this filter is

$$H_D(\omega) = \begin{cases} 1, & |\omega| \le \frac{\pi}{D} \\ 0, & \text{otherwise} \end{cases}.$$

# Decimator




We may refer to the entire downsampling system composed of an anti-aliasing filter followed by a downsampler as a *decimator* system.

# Upsampling: Time-Domain

## Upsampling: time-domain

A discrete-time signal $x[n]$ is *upsampled* by an integer factor $U$ by inserting $U-1$ zeros after each sample in $x[n]$.

$$y[n] = \begin{cases} x\left[\frac{n}{U}\right], & n = 0, U, 2U, \ldots \\ 0, & \text{else} \end{cases}.$$

- Increases length of $x[n]$ by factor of $U$.
- Increases the implicit sampling rate by $U$ when performing D/A conversion.

For example, let

$$x[n] = \{\underset{\uparrow}{1}, 2, 3\}.$$

If we upsample $x[n]$ by $U=3$, we obtain

$$y[n] = \{\underset{\uparrow}{1}, 0, 0, 2, 0, 0, 3, 0, 0\}.$$

# Upsampling: Frequency-Domain

## Upsampling: frequency domain

A discrete-time signal $x[n] \stackrel{\mathcal{F}}{\longleftrightarrow} X_d(\omega)$ upsampled by a factor of $U$ will produce $y[n] \stackrel{\mathcal{F}}{\longleftrightarrow} Y_d(\omega)$ where

$$Y_d(\omega) = X_d(U\omega)$$

- The spectrum $X_d(\omega)$ is compressed by a factor of $U$.

We will see $U$ copies of the original $X_d(\omega)$ spectrum between $[-\pi, \pi]$ in the upsampled spectrum $Y_d(\omega)$.

# Interpolation Filter

Upsampling a discrete-time signal causes $U-1$ additional spectral copies to appear from $[-\pi, \pi]$. We also have $U-1$ "blank" samples between each original $x[n]$ sample.

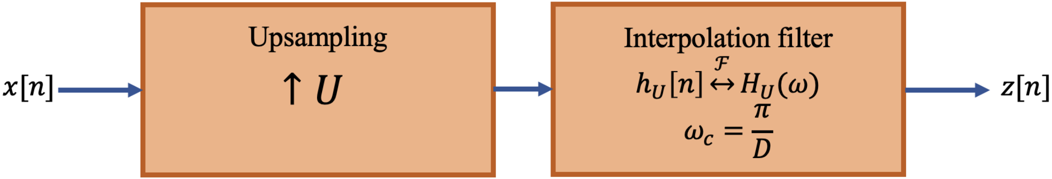
## Interpolation filter

The *interpolation filter* $H_U(\omega)$ removes the additional $U-1$ spectral copies in the DTFT of the upsampled signal

$$H_U(\omega) = \begin{cases} U, & |\omega| \leq \frac{\pi}{U} \\ 0, & \text{otherwise} \end{cases}.$$

- The interpolation filter comes **after** the upsampling operation.
- The interpolation filter is equivalent to performing sinc interpolation in the time-domain to "fill in the blanks" of the blank samples.
- The filter has magnitude $U$ to maintain the height of the original $x[n]$ samples.

# Interpolator




We may refer to the entire upsampling system composed of an upsampler followed by an interpolation filter as an *interpolator* system.

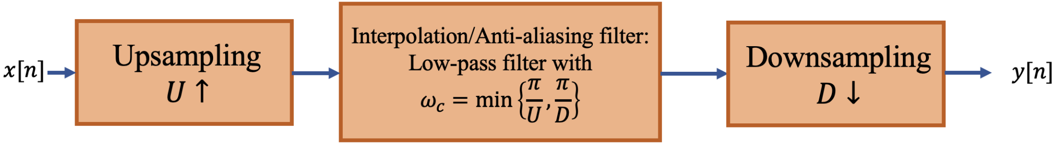
# Conversion by Non-Integer Factor

## Non-Integer Rate Conversion

We may achieve non-integer rate conversion by combining upsampling and downsampling operations. Upsampling by $U$ and downsampling by $D$ will lead to the sampling rate changing by a factor of $\frac{U}{D}$. Such a system proceeds as follows:

1. Upsample by $U$.
2. Apply low-passs filter that is the combination of interpolation and anti-aliasing filters. Thus, it has height $U$ and cutoff frequency $\omega_c = \min\left\{\frac{\pi}{D}, \frac{\pi}{U}\right\}$.
3. Downsample by $D$.

# Conversion by Non-Integer Factor




Cascaded rate conversion scheme for rate conversion by non-integer factor $R = \frac{U}{D}$.