ESE 531: Digital Signal Processing

Lec 24: April 25, 2017

Review

Course Content

- Introduction
- Discrete Time Signals & Systems
- Discrete Time Fourier Transform
- Z-Transform
- Inverse Z-Transform
- Sampling of Continuous Time Signals
- Frequency Domain of Discrete Time Series
- Downsampling/Upsampling
- Data Converters, Sigma Delta Modulation
- Oversampling, Noise Shaping
- Frequency Response of LTI Systems
- Basic Structures for IIR and FIR Systems
- Design of IIR and FIR Filters
- Filter Banks
- Adaptive Filters
- Computation of the Discrete Fourier Transform
- Fast Fourier Transform
- Compressive Sampling

Digital Signal Processing

- Represent signals by a sequence of numbers
    - Sampling and quantization (or analog-to-digital conversion)
- Perform processing on these numbers with a digital processor
    - Digital signal processing
- Reconstruct analog signal from processed numbers
    - Reconstruction or digital-to-analog conversion

- Analog input → analog output
    - Eg. Digital recording music
- Analog input → digital output
    - Eg. Touch tone phone dialing, speech to text
- Digital input → analog output
    - Eg. Text to speech
- Digital input → digital output
    - Eg. Compression of a file on computer

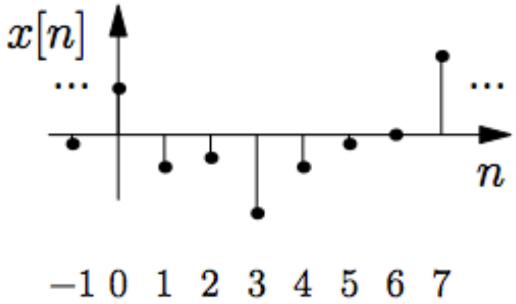
Discrete Time Signals

Signals are Functions

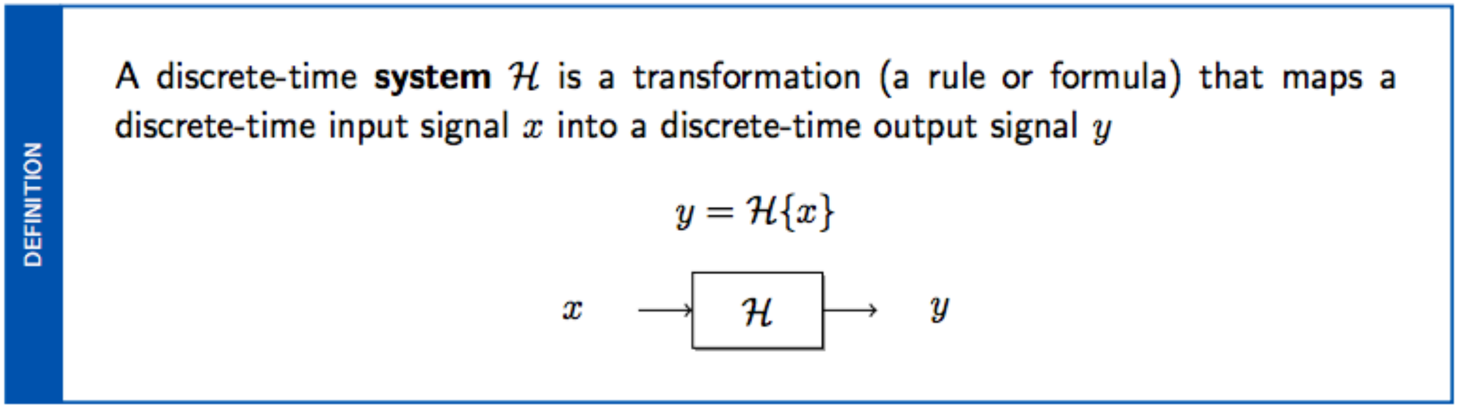
DEFINITION

A **signal** is a function that maps an independent variable to a dependent variable.

- Signal $x[n]$: each value of n produces the value $x[n]$
- In this course, we will focus on **discrete-time** signals:
  - Independent variable is an **integer**: $n \in \mathbb{Z}$ (will refer to as time)
  - Dependent variable is a real or complex number: $x[n] \in \mathbb{R}$ or $\mathbb{C}$



Discrete Time Systems



DEFINITION

A discrete-time **system** $\mathcal{H}$ is a transformation (a rule or formula) that maps a discrete-time input signal x into a discrete-time output signal y

$$y = \mathcal{H}\{x\}$$

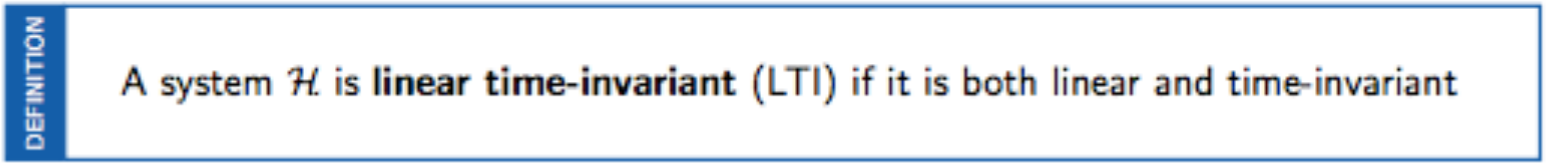
$$x \longrightarrow \boxed{\mathcal{H}} \longrightarrow y$$

- Systems manipulate the information in signals
- Examples:
  - A speech recognition system converts acoustic waves of speech into text
  - A radar system transforms the received radar pulse to estimate the position and velocity of targets
  - A functional magnetic resonance imaging (fMRI) system transforms measurements of electron spin into voxel-by-voxel estimates of brain activity
  - A 30 day moving average smooths out the day-to-day variability in a stock price

System Properties

- Causality
  - y[n] only depends on x[m] for m<=n
- Linearity
  - Scaled sum of arbitrary inputs results in output that is a scaled sum of corresponding outputs
    - $Ax_1[n]+Bx_2[n] \rightarrow Ay_1[n]+By_2[n]$
- Memoryless
  - y[n] depends only on x[n]
- Time Invariance
  - Shifted input results in shifted output
    - $x[n-q] \rightarrow y[n-q]$
- BIBO Stability
  - A bounded input results in a bounded output (ie. max signal value exists for output if max)

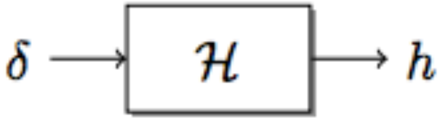
LTI Systems



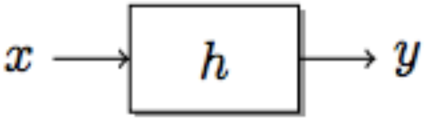
DEFINITION

A system $\mathcal{H}$ is **linear time-invariant** (LTI) if it is both linear and time-invariant

- LTI system can be completely characterized by its impulse response



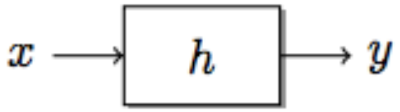
- Then the output for an arbitrary input is a sum of weighted, delay impulse responses



$$y[n] = \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$

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

Convolution



- Convolution formula

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

- To compute the entry $y[n]$ in the output vector y:
  1. **Time reverse** the impulse response vector h and **shift** it n time steps to the right (delay)
  2. Compute the **inner product** between the shifted impulse response and the input vector x
- Repeat for every n

Discrete Time Fourier Transform

DTFT Definition

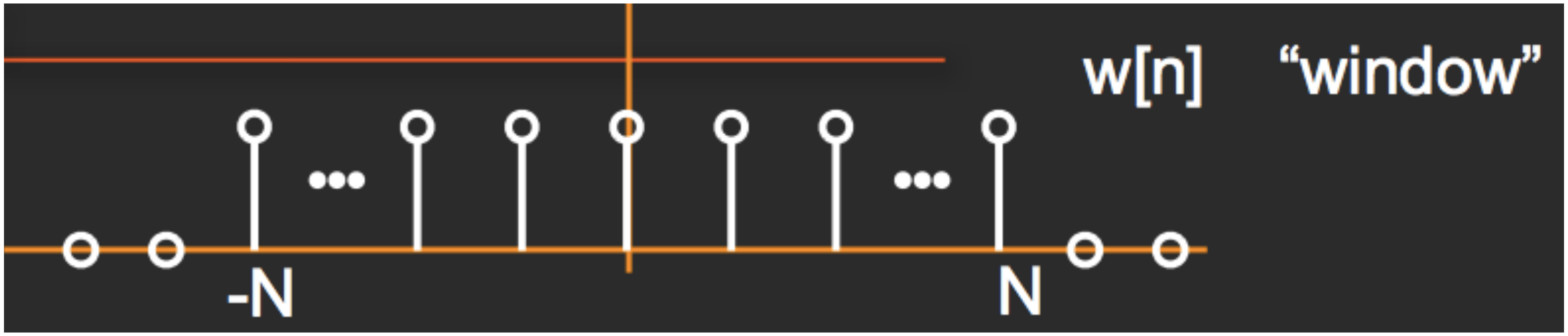
$$X(e^{j\omega}) = \sum_{k=-\infty}^{\infty} x[k] e^{-j\omega k}$$

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

Alternate

$$\boxed{\begin{aligned} X(f) &= \sum_{k=-\infty}^{\infty} x[k] e^{-j2\pi fk} \\ x[n] &= \int_{-0.5}^{0.5} X(f) e^{j2\pi fn} df \end{aligned}}$$

Example: Window DTFT



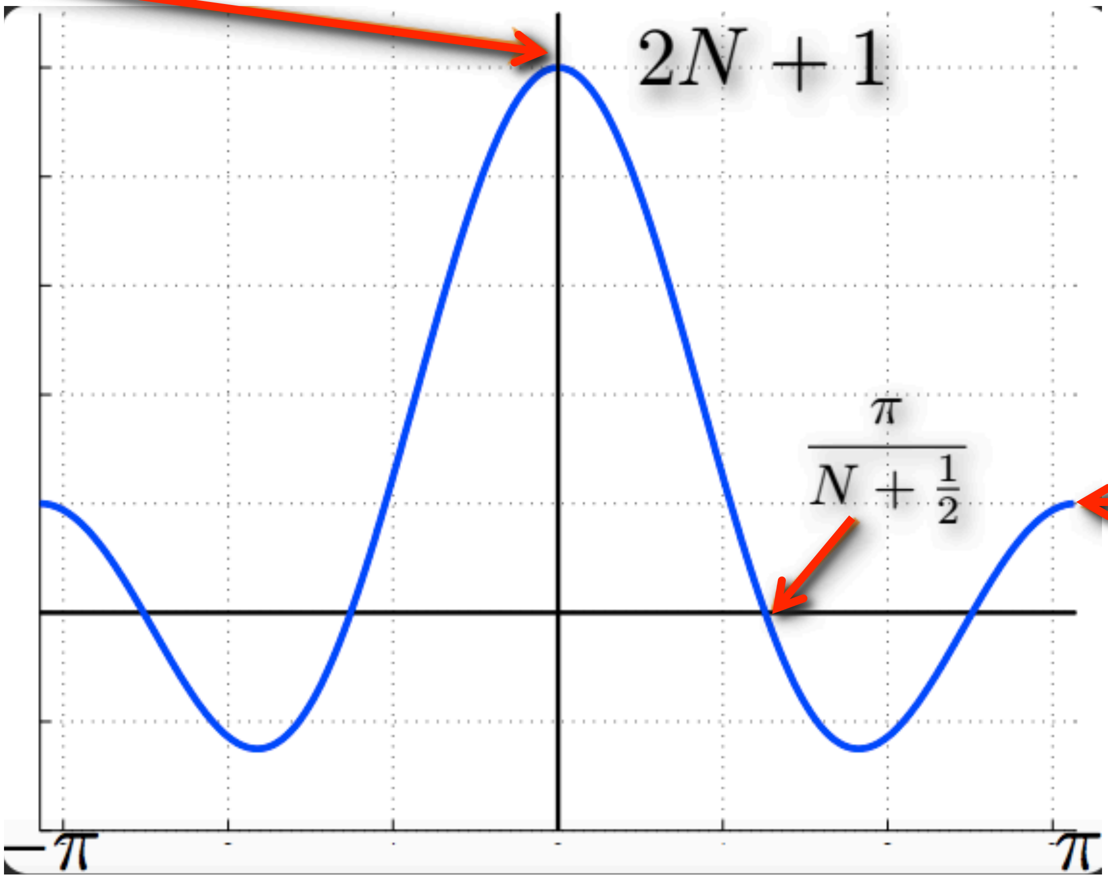
$$W(e^{j\omega}) = \sum_{k=-\infty}^{\infty} w[k] e^{-j\omega k}$$

$$= \sum_{k=-N}^{N} e^{-j\omega k}$$

Example: Window DTFT

$$W(e^{j\omega}) = \frac{\sin\left((N+1/2)\omega\right)}{\sin\left(\omega/2\right)}$$

Also, Σx[n]

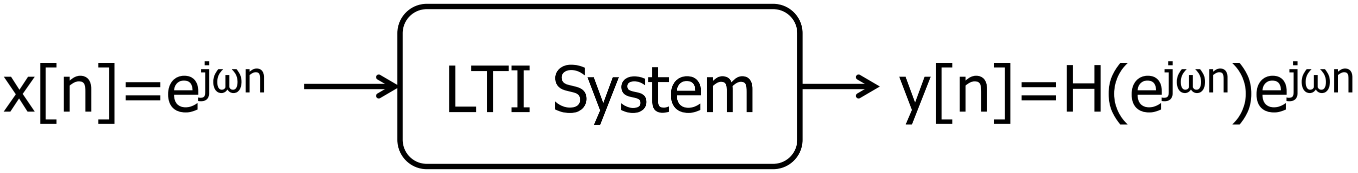


=1 why?

Plot for N=2

LTI System Frequency Response

- Fourier Transform of impulse response



$x[n]=e^{j\omega n}$

LTI System

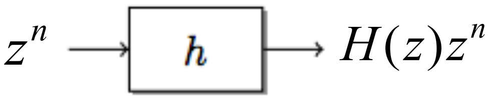
$y[n]=H(e^{j\omega n})e^{j\omega n}$

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega k}$$

z-Transform

- The z-transform generalizes the Discrete-Time Fourier Transform (DTFT) for analyzing infinite-length signals and systems
- Very useful for designing and analyzing signal processing systems
- Properties are very similar to the DTFT with a few caveats

Complex Exponentials as Eigenfunctions

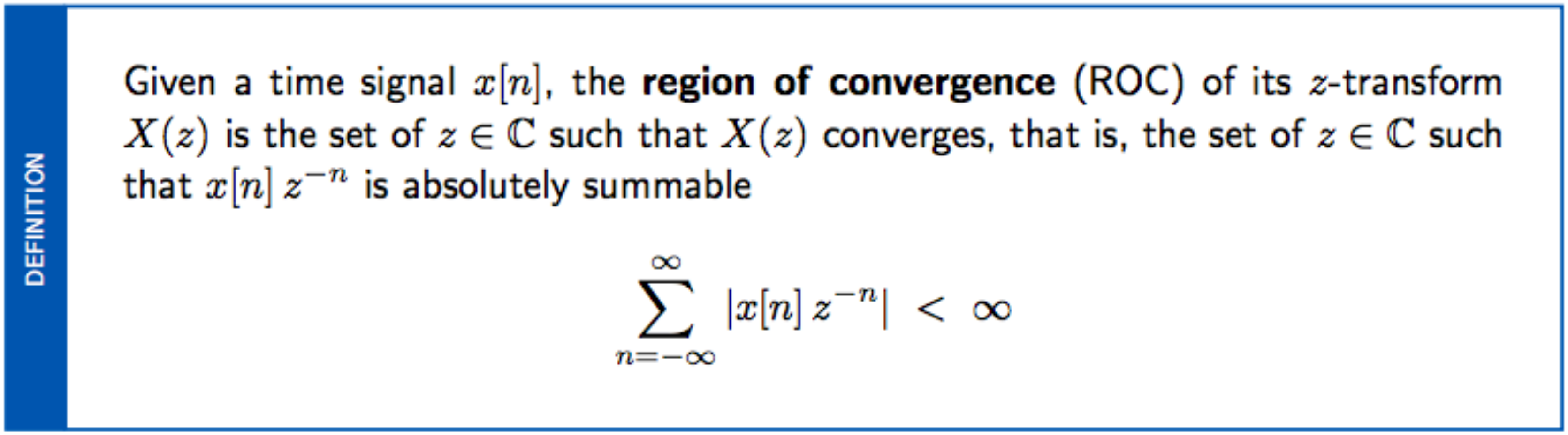


- Prove that complex exponentials are the eigenvectors of LTI systems simply by computing the convolution with input z^n

$$\begin{aligned} z^n * h[n] &= \sum_{m=-\infty}^{\infty} z^{n-m}\, h[m] \;=\; \sum_{m=-\infty}^{\infty} z^n\, z^{-m}\, h[m] \\ &= \left(\sum_{m=-\infty}^{\infty} h[m]\, z^{-m} \right) z^n \\ &= H(z)z^n \end{aligned}$$

$$\boxed{\sum_{n=-\infty}^{\infty} h[n]\, z^{-n} \;=\; H(z)}$$

Region of Convergence (ROC)



DEFINITION

Given a time signal $x[n]$, the **region of convergence** (ROC) of its z-transform $X(z)$ is the set of $z \in \mathbb{C}$ such that $X(z)$ converges, that is, the set of $z \in \mathbb{C}$ such that $x[n]\, z^{-n}$ is absolutely summable

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

Formal Properties of the ROC

- PROPERTY 1:
  - The ROC will either be of the form $0 < r_R < |z|$, or $|z| < r_L < \infty$, or, in general the annulus, i.e., $0 < r_R < |z| < r_L < \infty$.
- PROPERTY 2:
  - The Fourier transform of x[n] converges absolutely if and only if the ROC of the z-transform of x[n] includes the unit circle.
- PROPERTY 3:
  - The ROC cannot contain any poles.
- PROPERTY 4:
  - If x[n] is *a finite-duration sequence*, i.e., a sequence that is zero except in a finite interval $-\infty < N_1 < n < N_2 < \infty$, then the ROC is the entire z-plane, except possibly $z = 0$ or $z = \infty$.

Formal Properties of the ROC

- PROPERTY 5:
  - If x[n] is a *right-sided sequence*, the ROC extends outward from the *outermost* finite pole in X(z) to (and possibly including) z = ∞.
- PROPERTY 6:
  - If x[n] is a *left-sided sequence*, the ROC extends inward from the *innermost* nonzero pole in X(z) to (and possibly including) z=0.
- PROPERTY 7:
  - A *two-sided sequence* is an infinite-duration sequence that is neither right sided nor left sided. If x[n] is a two-sided sequence, the ROC will consist of a ring in the z-plane, bounded on the interior and exterior by a pole and, consistent with Property 3, not containing any poles.
- PROPERTY 8:
  - The ROC must be a connected region.

Inverse z-Transform

- Ways to avoid it:
  - Inspection (known transforms)
  - Properties of the z-transform
  - Power series expansion
  - Partial fraction expansion

Partial Fraction Expansion

- Let

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

- M zeros and N poles at nonzero locations

Partial Fraction Expansion

- If M<N and the poles are 1st order

$$X(z)=\frac{b_0}{a_0}\frac{\prod_{k=1}^{M}(1-c_k z^{-1})}{\prod_{k=1}^{N}(1-d_k z^{-1})}=\sum_{k=1}^{N}\frac{A_k}{1-d_k z^{-1}}$$

- where

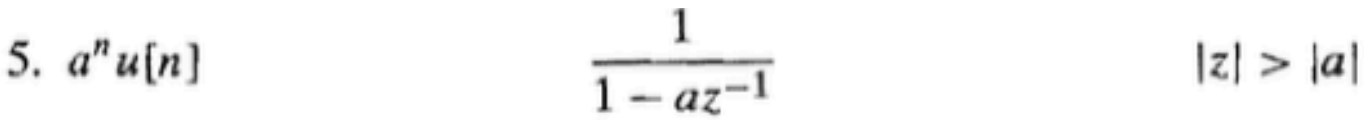
$$A_k=(1-d_k z^{-1})X(z)\Big|_{z=d_k}$$

Example: 2nd-Order z-Transform

- 2nd-order = two poles

Right sided

$$X(z)=\frac{-1}{\left(1-\frac{1}{4}z^{-1}\right)}+\frac{2}{\left(1-\frac{1}{2}z^{-1}\right)},\qquad ROC=\left\{z:\ \frac{1}{2}<|z|\right\}$$



5. $a^n u[n]$	$\dfrac{1}{1-az^{-1}}$	$\lvert z\rvert > \lvert a\rvert$

$$x[n]=-\left(\frac{1}{4}\right)^n u[n]+2\left(\frac{1}{2}\right)^n u[n]$$

Partial Fraction Expansion

- If M≥N and the poles are 1st order

$$X(z) = \sum_{r=0}^{M-N} B_r z^{-r} + \sum_{k=1}^{N} \frac{A_k}{1-d_k z^{-1}}$$

- Where B_k is found by long division

$$A_k = (1-d_k z^{-1})X(z)\Big|_{z=d_k}$$

Example: Partial Fractions

- M=N=2 and poles are first order

$$X(z) = B_0 + \frac{A_1}{1-\frac{1}{2}z^{-1}} + \frac{A_2}{1-z^{-1}}, \qquad ROC = \{z:\ 1<|z|\}$$

$$\begin{array}{r}
2 \\
\frac{1}{2}z^{-2} - \frac{3}{2}z^{-1} + 1 \overline{\big) z^{-2} + 2z^{-1} + 1} \\
\underline{z^{-2} - 3z^{-1} + 2} \\
5z^{-1} - 1
\end{array}$$

$$X(z) = 2 + \frac{-1+5z^{-1}}{(1-\frac{1}{2}z^{-1})(1-z^{-1})}$$

Example: Partial Fractions

- M=N=2 and poles are first order

$$X(z) = 2 - \frac{9}{1 - \frac{1}{2}z^{-1}} + \frac{8}{1 - z^{-1}}, \qquad ROC = \{z : 1 < |z|\}$$

$$x[n] = 2\delta[n] - 9\left(\frac{1}{2}\right)^n u[n] + 8u[n]$$

Power Series Expansion

- Expansion of the z-transform definition

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

$$= \cdots + x[-2]z^{2} + x[-1]z + x[0] + x[1]z^{-1} + x[2]z^{-2} + \cdots$$

Example: Finite-Length Sequence

- Poles and zeros?

$$X(z) = z^2\left(1-\frac{1}{2}z^{-1}\right)(1+z^{-1})(1-z^{-1})$$

$$= z^2 - \frac{1}{2}z - 1 + \frac{1}{2}z^{-1}$$

$$x[n] = \begin{cases} 1, & n=-2 \\ -1/2, & n=-1 \\ -1, & n=0 \\ 1/2, & n=1 \\ 0, & else \end{cases}$$

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

$$= \cdots + x[-2]z^2 + x[-1]z + x[0] + x[1]z^{-1} + x[2]z^{-2} + \cdots$$

Example: Finite-Length Sequence

- Poles and zeros?

$$X(z) = z^2\left(1-\frac{1}{2}z^{-1}\right)(1+z^{-1})(1-z^{-1})$$

$$= z^2 - \frac{1}{2}z - 1 + \frac{1}{2}z^{-1}$$

$$x[n] = \begin{cases} 1, & n=-2 \\ -1/2, & n=-1 \\ -1, & n=0 \\ 1/2, & n=1 \\ 0, & else \end{cases} = \delta[n+2] - \frac{1}{2}\delta[n+1] - \delta[n] + \frac{1}{2}\delta[n-1]$$

4. $\delta[n-m]$ z^{-m}

Difference Equation to z-Transform

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{m=0}^{M} b_m x[n-m]$$

$$y[n] = -\sum_{k=1}^{N} \left(\frac{a_k}{a_0} \right) y[n-k] + \sum_{k=0}^{M} \left(\frac{b_k}{a_0} \right) x[n-m]$$

- Difference equations of this form behave as causal LTI systems
  - when the input is zero prior to n=0
  - Initial rest equations are imposed prior to the time when input becomes nonzero
    - i.e y[-N]=y[-N+1]=…=y[-1]=0

Difference Equation to z-Transform

$$y[n] = -\sum_{k=1}^{N}\left(\frac{a_k}{a_0}\right)y[n-k] + \sum_{k=0}^{M}\left(\frac{b_k}{a_0}\right)x[n-m]$$

$$Y(z) = -\sum_{k=1}^{N}\left(\frac{a_k}{a_0}\right)z^{-k}Y(z) + \sum_{k=0}^{M}\left(\frac{b_k}{a_0}\right)z^{-k}X(z)$$

$$\sum_{k=0}^{N}\left(\frac{a_k}{a_0}\right)z^{-k}Y(z) = \sum_{k=0}^{M}\left(\frac{b_k}{a_0}\right)z^{-k}X(z) \Rightarrow Y(z) = \frac{\sum_{m=0}^{M}\left(b_k\right)z^{-k}}{\sum_{k=0}^{N}\left(a_k\right)z^{-k}}X(z)$$

$$\boxed{H(z) = \frac{\sum_{m=0}^{M}\left(b_k\right)z^{-k}}{\sum_{k=0}^{N}\left(a_k\right)z^{-k}}}$$

Example: 1st-Order System

$$H(z) = \frac{\sum_{m=0}^{M} \left(b_k\right) z^{-k}}{\sum_{k=0}^{N} \left(a_k\right) z^{-k}}$$

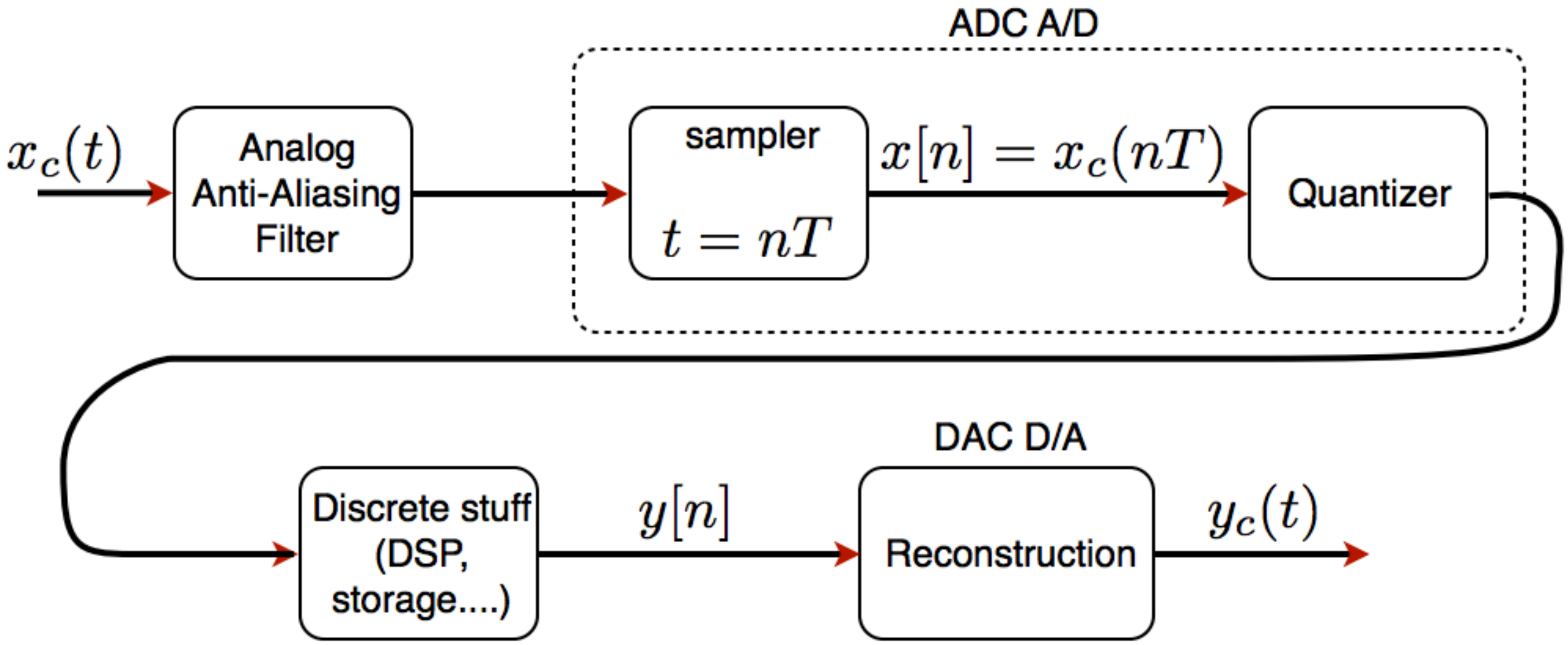
$$y[n] = ay[n-1] + x[n]$$

$$H(z) = \frac{1}{1 - az^{-1}}$$

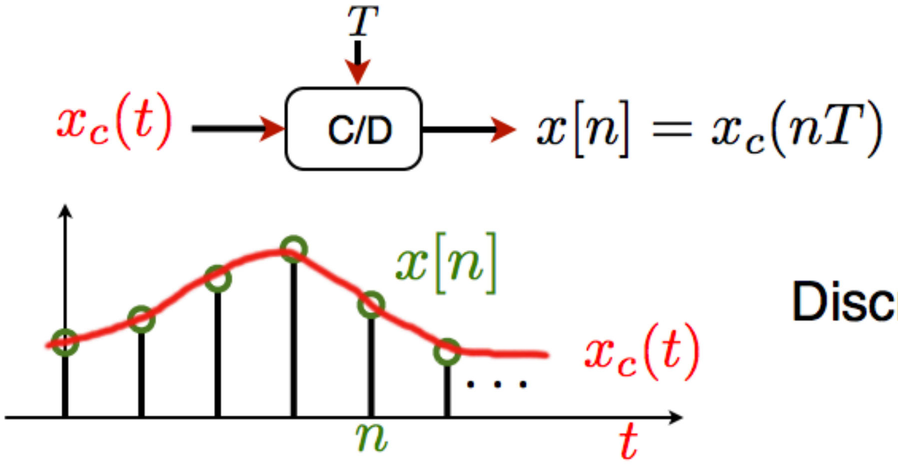
$$h[n] = a^n u[n]$$

Sampling

DSP System


ADC A/D
$x_c(t)$
Analog Anti-Aliasing Filter
sampler
$t = nT$
$x[n] = x_c(nT)$
Quantizer
DAC D/A
Discrete stuff (DSP, storage....)
$y[n]$
Reconstruction
$y_c(t)$

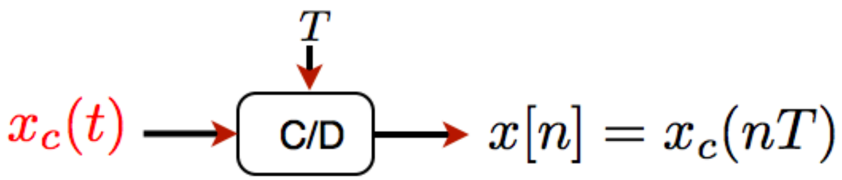
Ideal Sampling Model

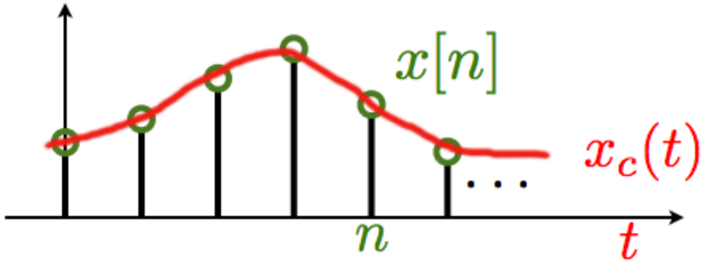


Discrete and Continuous

- Ideal continuous-to-discrete time (C/D) converter
  - T is the sampling period
  - f_s=1/T is the sampling frequency
  - $\Omega_s=2\pi/T$

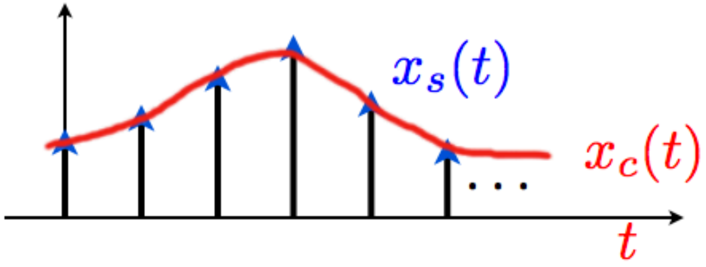
Ideal Sampling Model





Discrete and Continuous

define impulsive sampling:

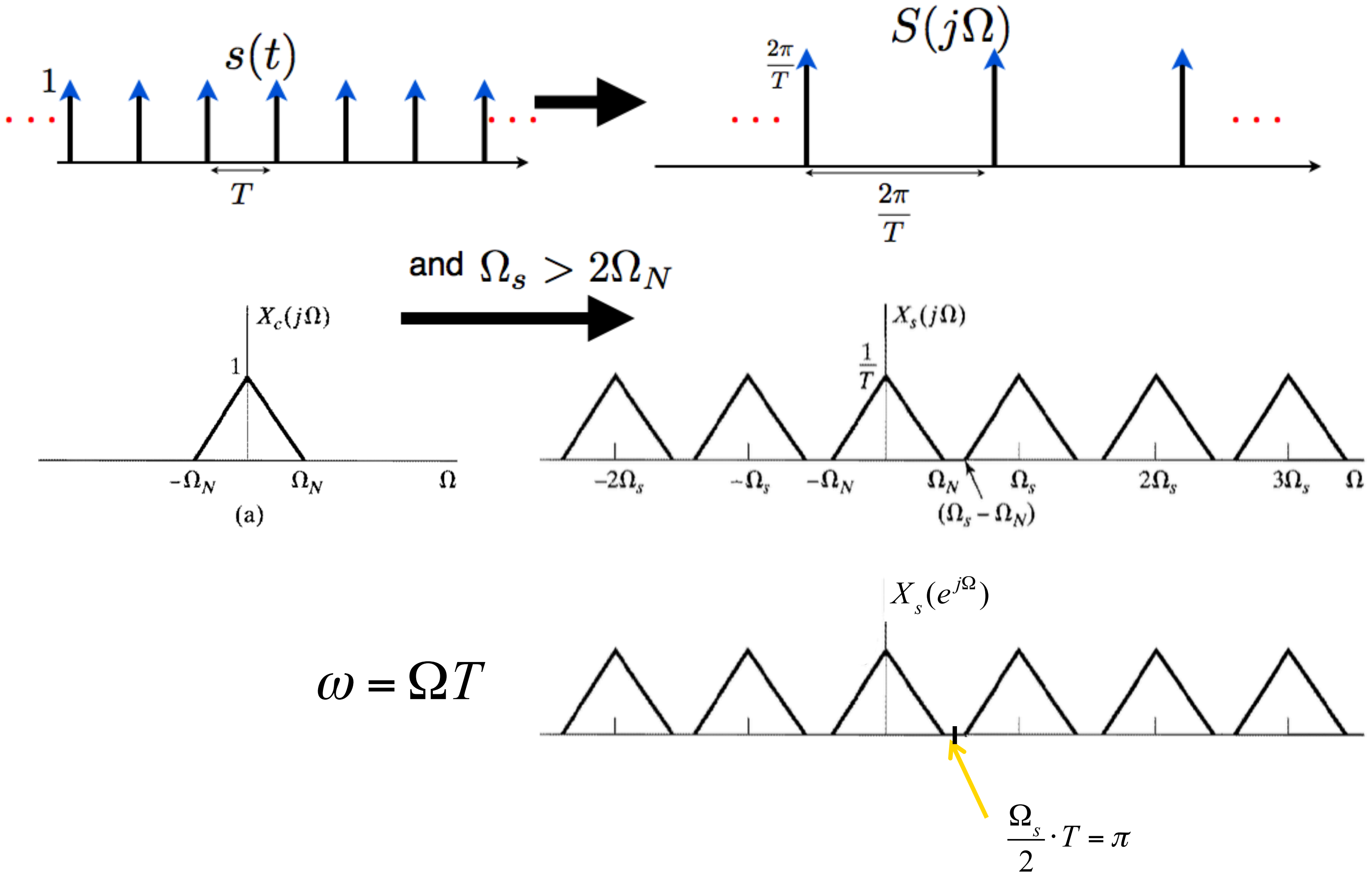


Continuous

$$x_s(t) = \cdots + x_c(0)\delta(t) + x_c(T)\delta(t-T) + \cdots$$

$$x_s(t) = x_c \sum_{n=-\infty}^{\infty} \delta(t-nT)$$

Frequency Domain Analysis

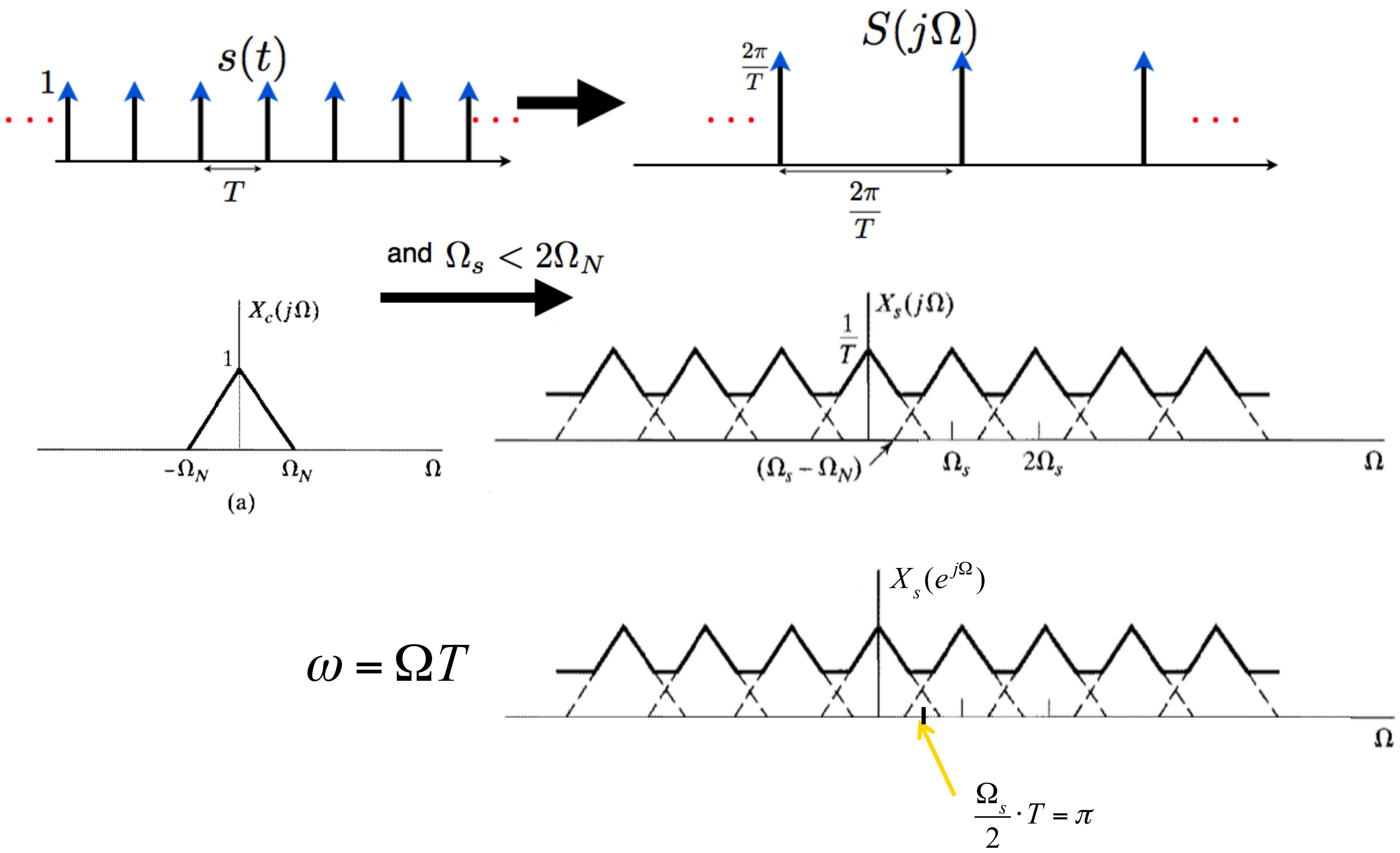


and $\Omega_s > 2\Omega_N$

$\omega = \Omega T$

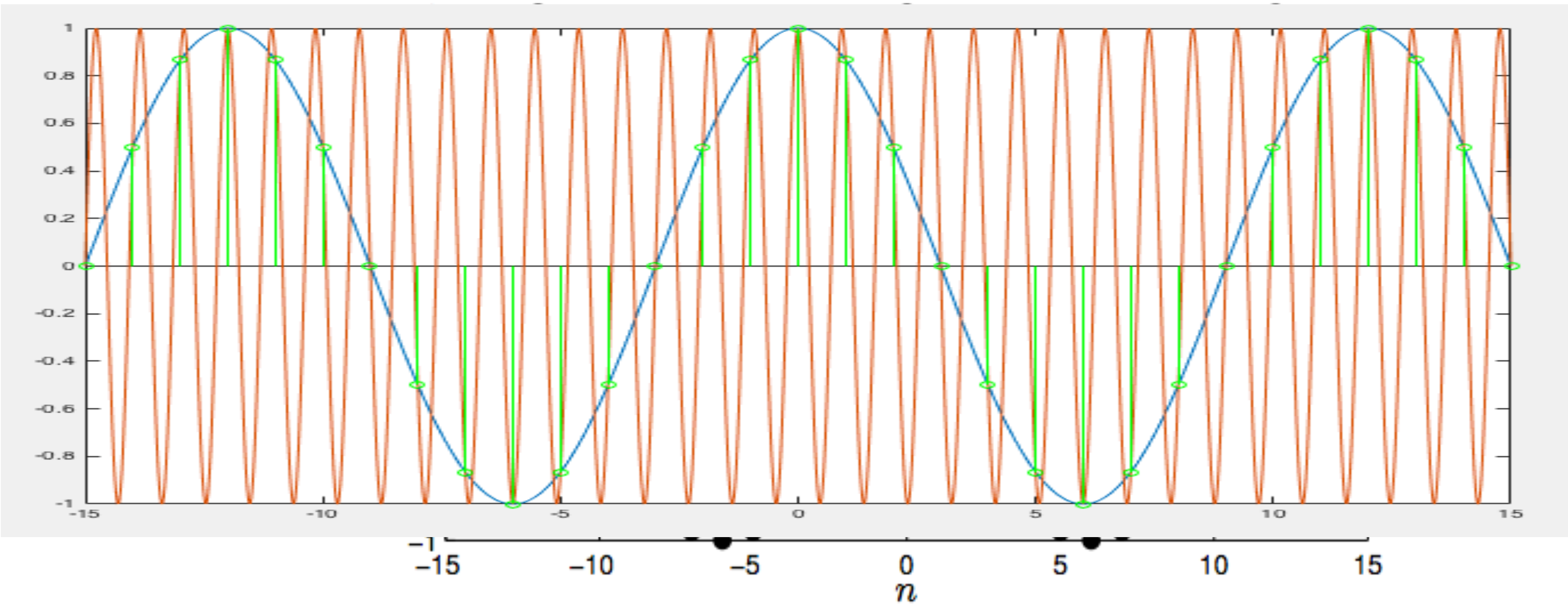
$$\frac{\Omega_s}{2} \cdot T = \pi$$

Frequency Domain Analysis



Aliasing Example

- $x_1[n] = \cos\left(\frac{\pi}{6}n\right)$

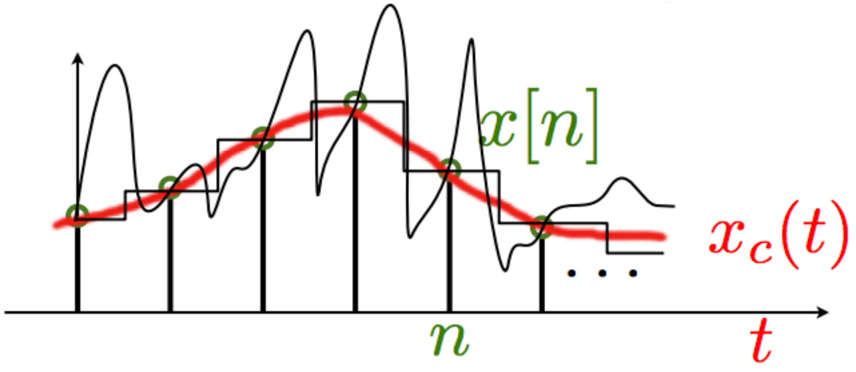


Reconstruction of Bandlimited Signals

- Nyquist Sampling Theorem: Suppose $x_c(t)$ is bandlimited. I.e.

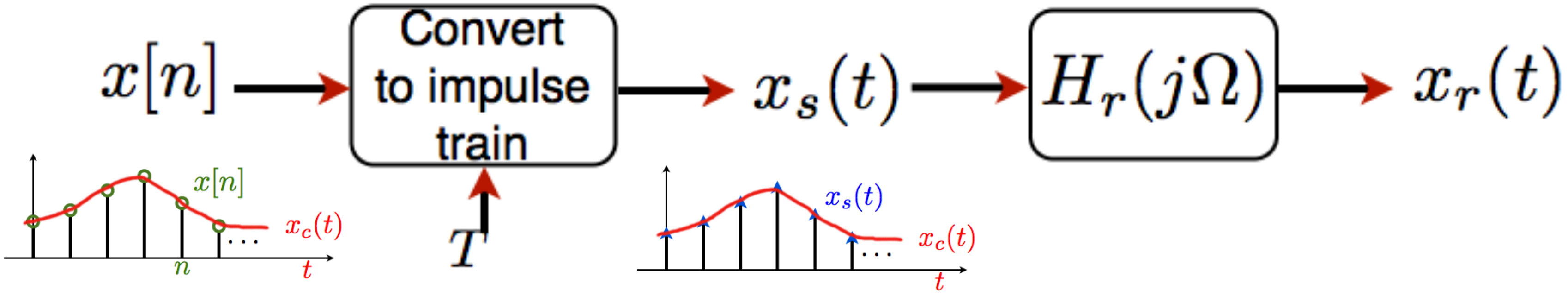
$$X_c(j\Omega) = 0 \;\; \forall \quad |\Omega| \geq \Omega_N$$

- If $\Omega_s \geq 2\Omega_N$, then $x_c(t)$ can be uniquely determined from its samples $x[n]=x_c(nT)$
- Bandlimitedness is the key to uniqueness



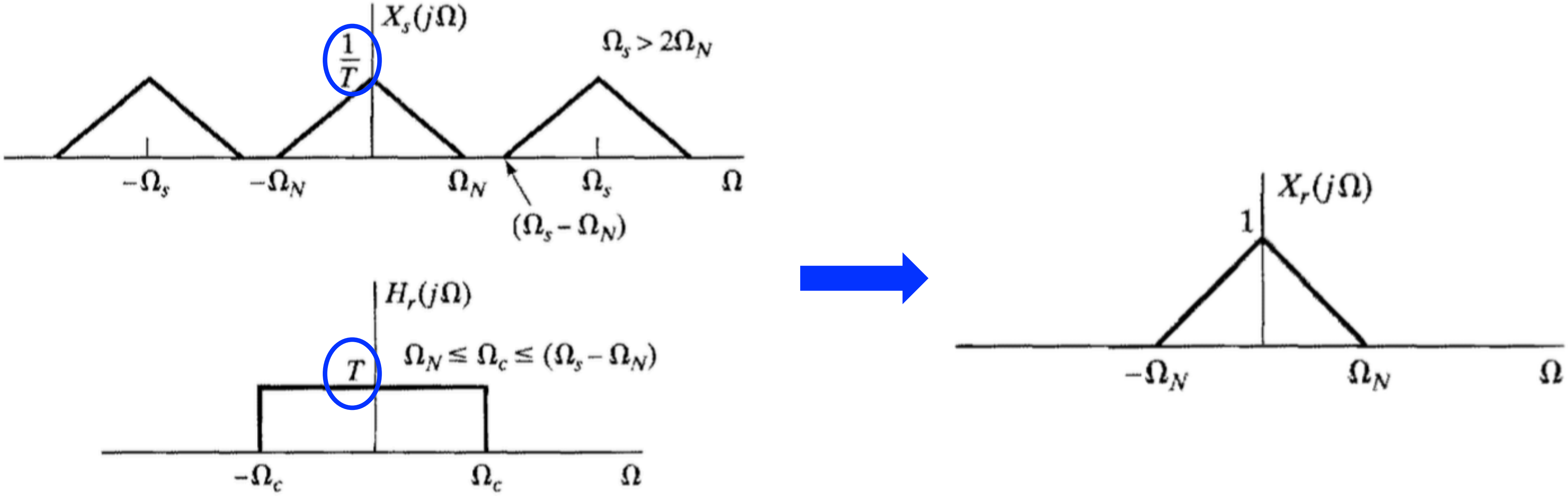
Mulitiple signals go through the samples, but only one is bandlimited within our sampling band

Reconstruction in Frequency Domain



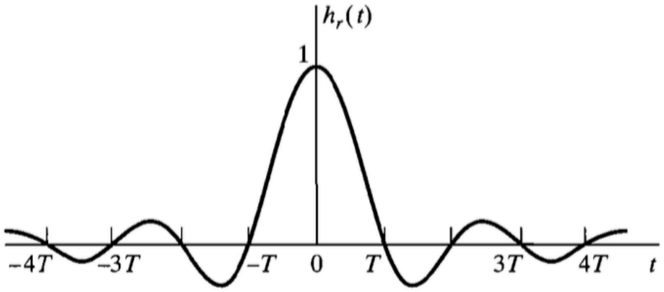
$x[n] \rightarrow$ Convert to impulse train $\rightarrow x_s(t) \rightarrow H_r(j\Omega) \rightarrow x_r(t)$

T

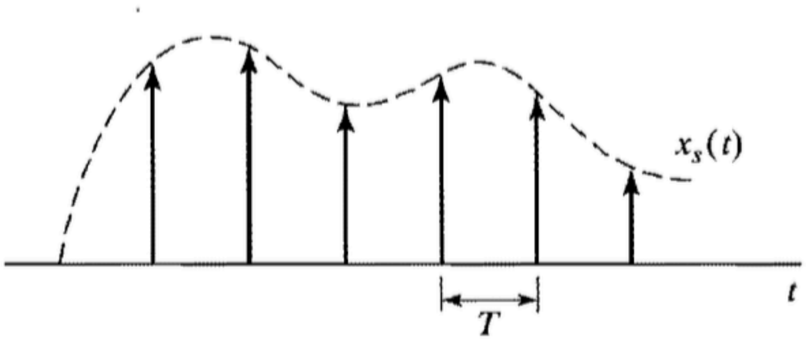


Reconstruction in Time Domain

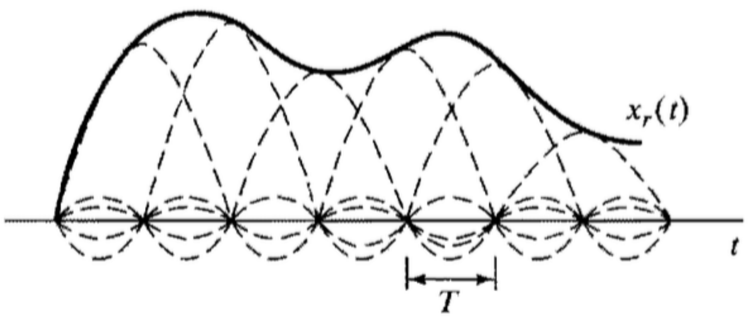
$$
\begin{aligned}
x_r(t) = x_s(t) * h_r(t) &= \left(\sum_n x[n]\delta(t - nT) \right) * h_r(t) \\
&= \sum_n x[n]h_r(t - nT)
\end{aligned}
$$



$*$

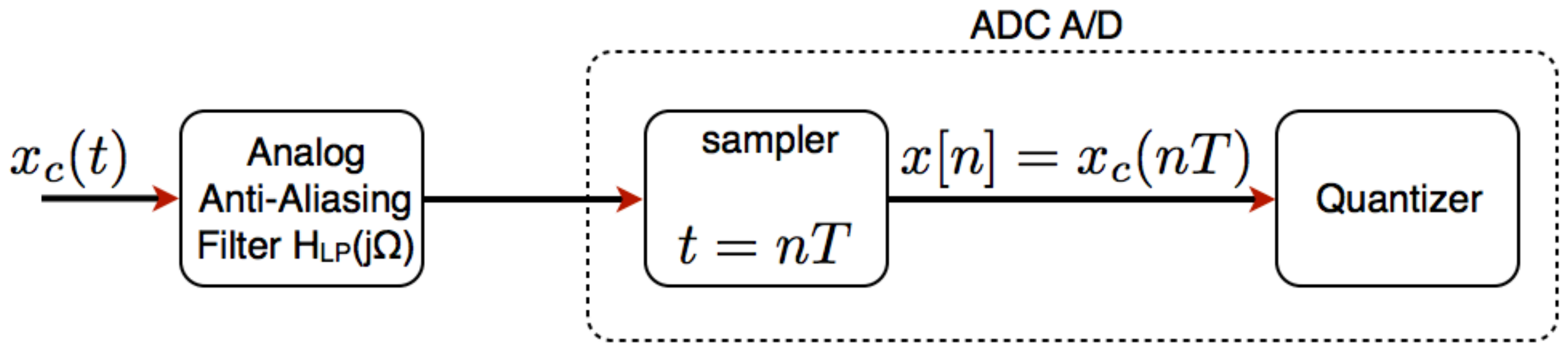


$=$



The sum of "sincs" gives $x_r(t)$ → unique signal that is bandlimited by sampling bandwidth

Anti-Aliasing Filter



ADC A/D

$x_c(t)$

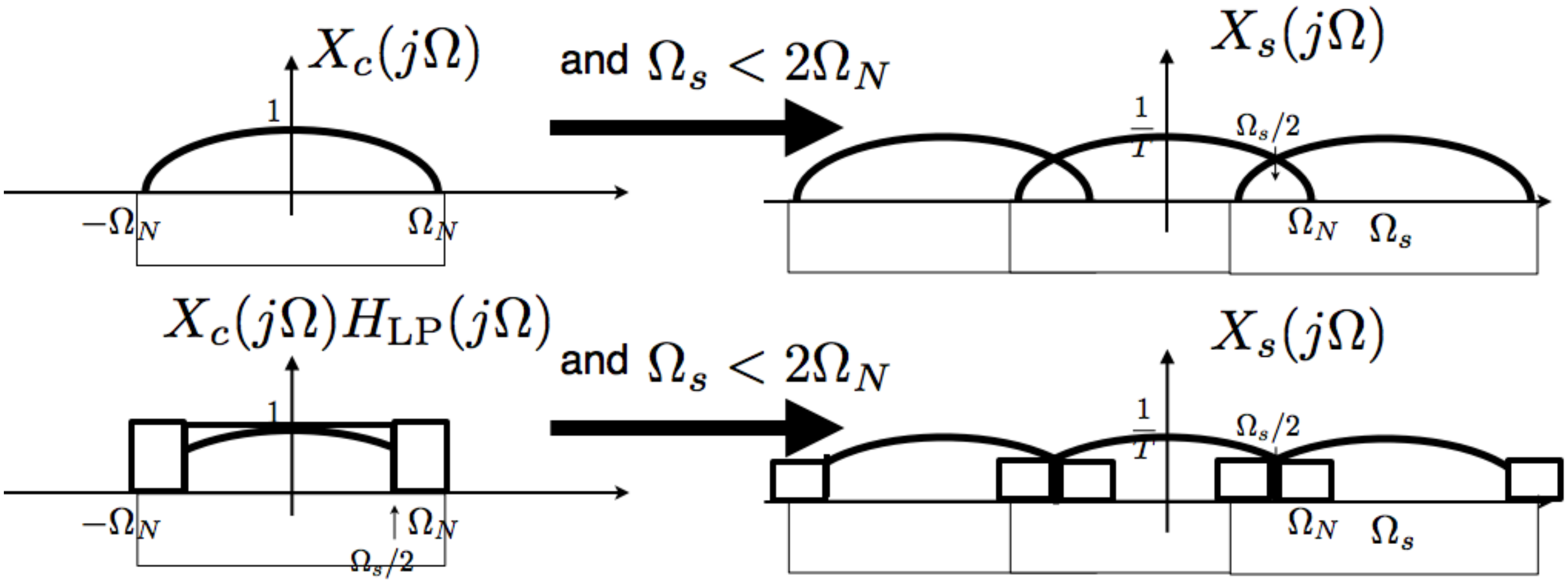
Analog Anti-Aliasing Filter $H_{LP}(j\Omega)$

sampler

$t = nT$

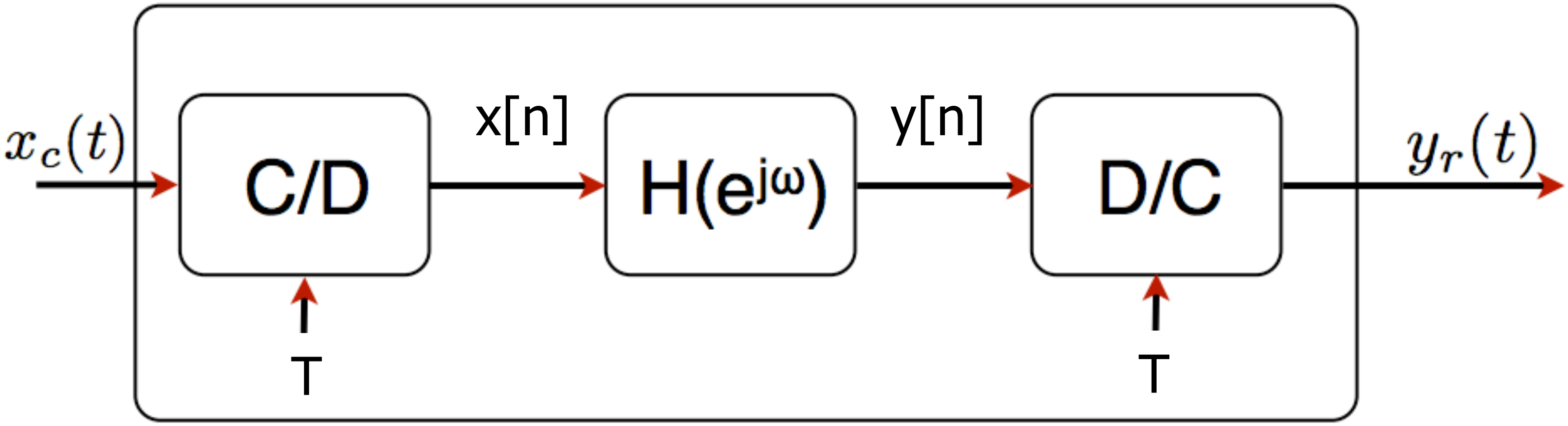
$x[n] = x_c(nT)$

Quantizer



DT and CT processing

Discrete-Time Processing of Continuous Time



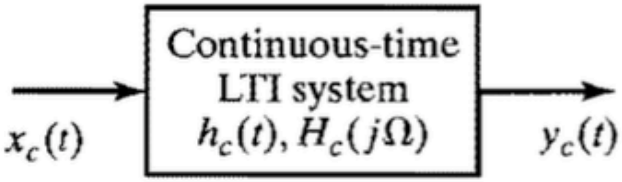
$$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left[j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right] \qquad y_r(t) = \sum_{n=-\infty}^{\infty} y[n]\frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T}$$

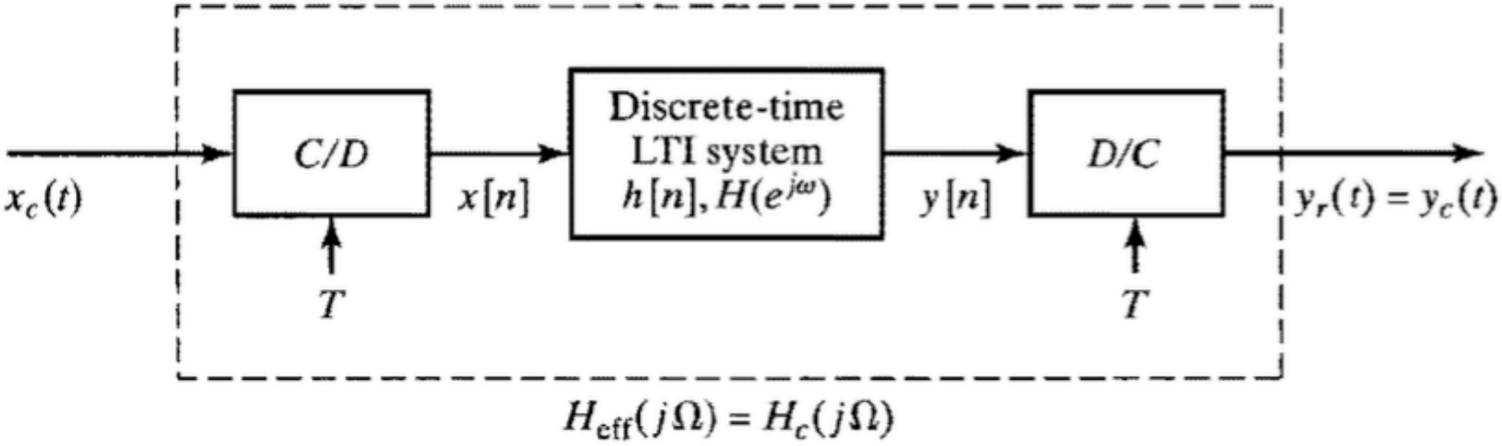
- If $x_c(t)$ is bandlimited by $\Omega_s/T = \pi/T$, then,

$$\frac{Y_r(j\Omega)}{X_c(j\Omega)} = H_{eff}(j\Omega) = \begin{cases} H(e^{j\omega})\Big|_{\omega=\Omega T} & |\Omega| < \Omega_s/T \\ 0 & else \end{cases}$$

Impulse Invariance

- Want to implement continuous-time system in discrete-time





Impulse Invariance

- With $H_c(j\Omega)$ bandlimited, choose

$$H(e^{j\omega}) = H_c(j\omega / T), \qquad |\omega| < \pi$$

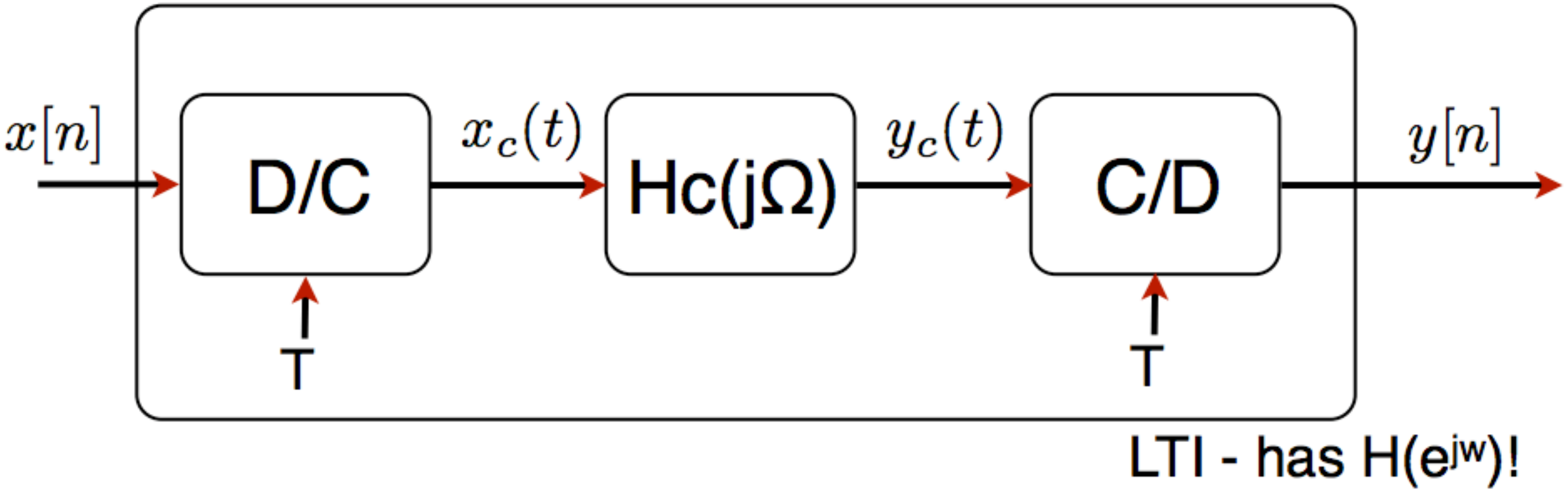
- With the further requirement that T be chosen such that

$$H_c(j\Omega) = 0, \qquad |\Omega| \geq \pi / T$$

$$\boxed{h[n] = Th_c(nT)}$$

Continuous-Time Processing of Discrete-Time

- Useful to interpret DT systems with no simple interpretation in discrete time



$$x_c(t) = \sum_{n=-\infty}^{\infty} x[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T}$$

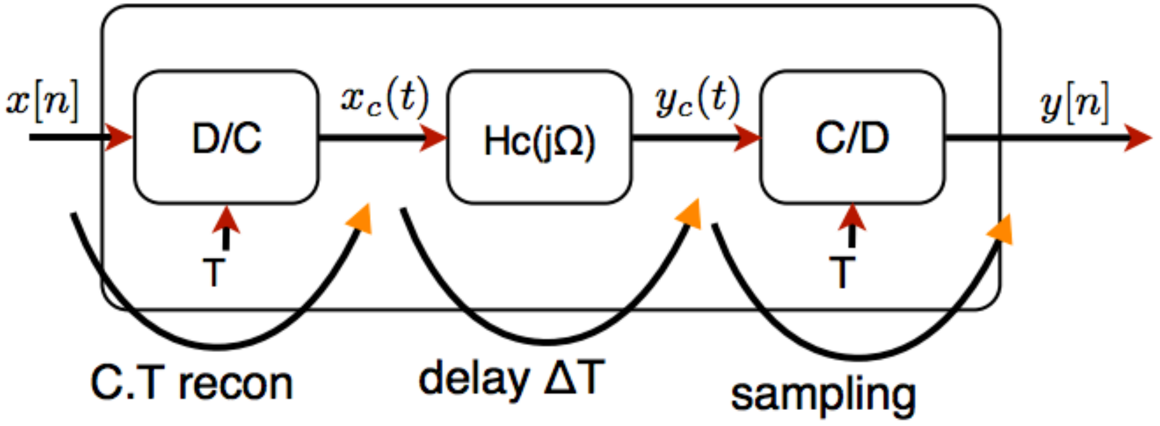
$$y_c(t) = \sum_{n=-\infty}^{\infty} y[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T}$$

Example: Non-integer Delay

- What is the time domain operation when Δ is non-integer? I.e Δ=1/2

$$H(e^{j\omega}) = e^{-j\omega\Delta}$$

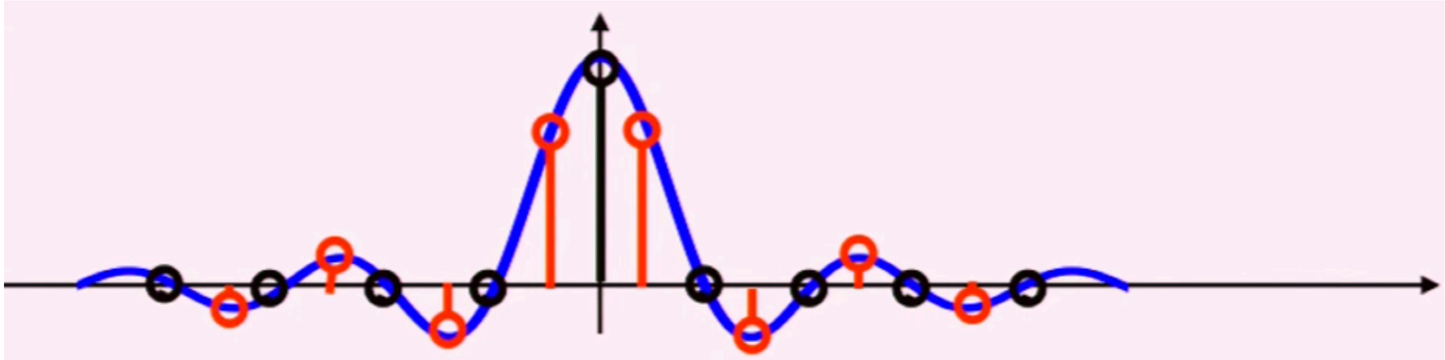
Let: $H_c(j\Omega) = e^{-j\Omega\Delta T}$ delay of ΔT in time



Example: Non-integer Delay

- My delay system has an impulse response of a sinc with a continuous time delay

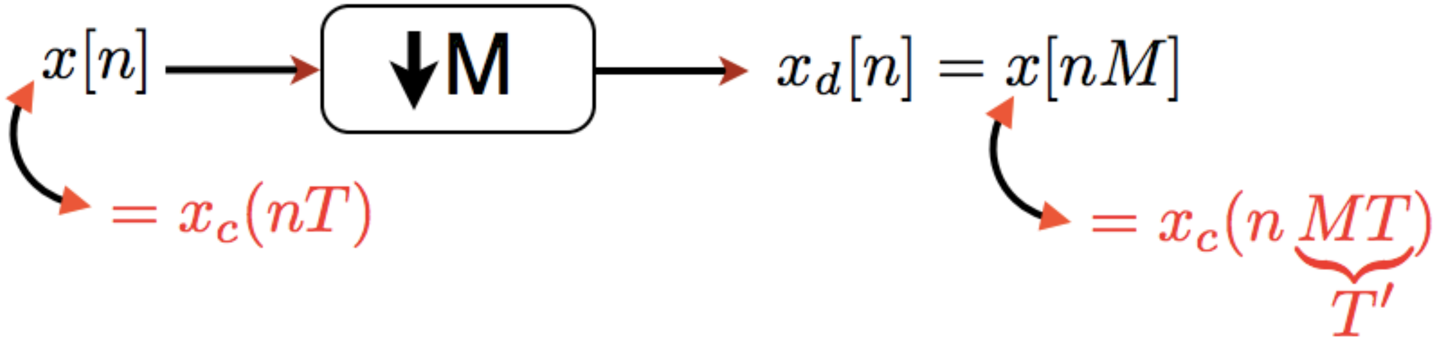
$$h[n] = \mathrm{sinc}(n - \Delta)$$



Rate Sampling

Downsampling

- Definition: Reducing the sampling rate by an integer number



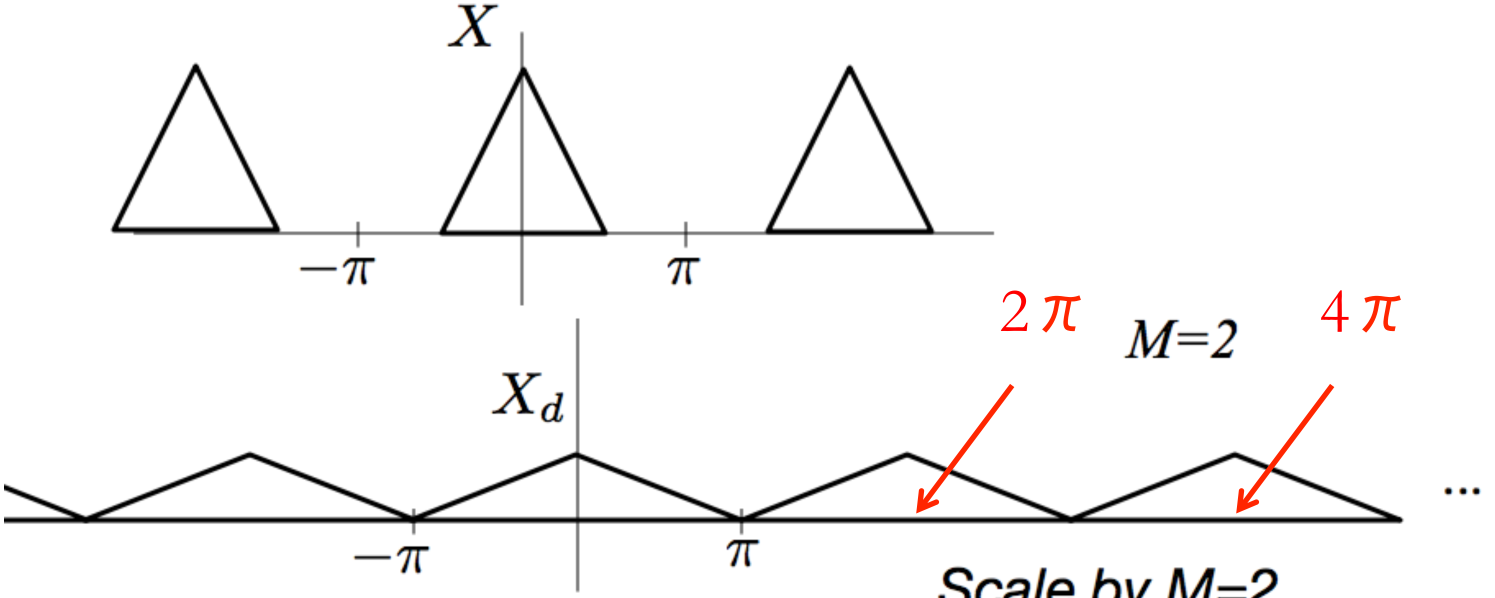
$$x[n] \longrightarrow \boxed{\downarrow M} \longrightarrow x_d[n] = x[nM]$$

$$= x_c(nT)$$

$$= x_c(n\underbrace{MT}_{T'})$$

Example

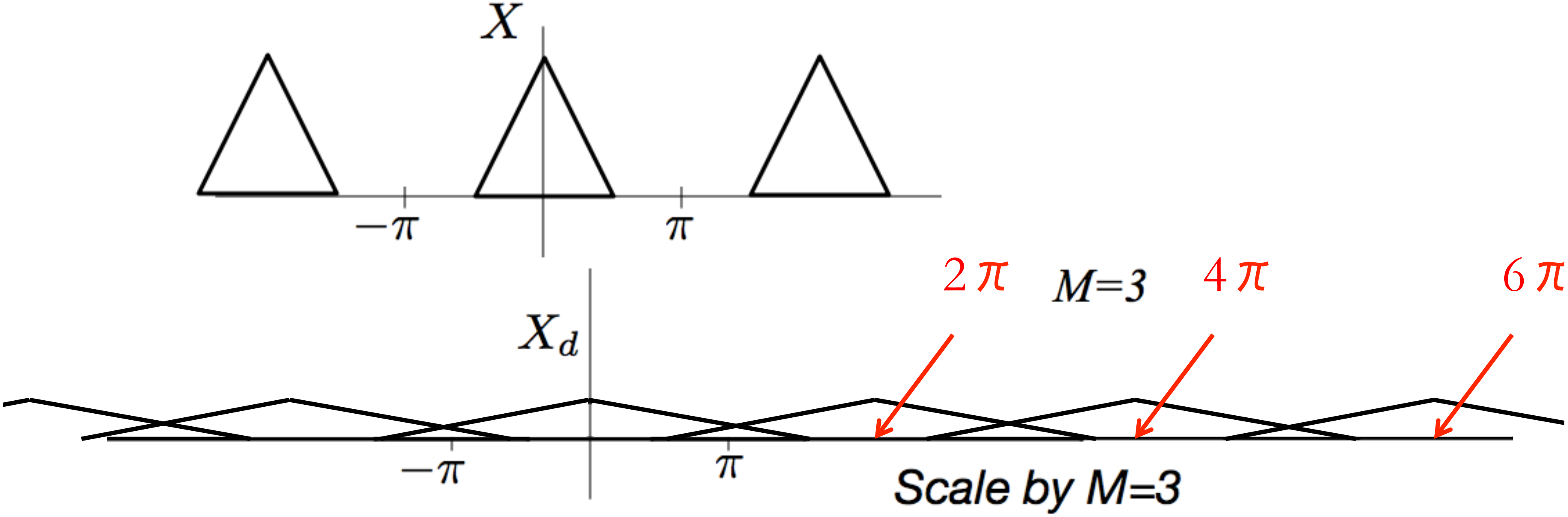
$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j\left(\frac{w}{M}-\frac{2\pi}{M}i\right)}\right)$$



Scale by M=2
Shift by (i=1)*2π/(M=2)

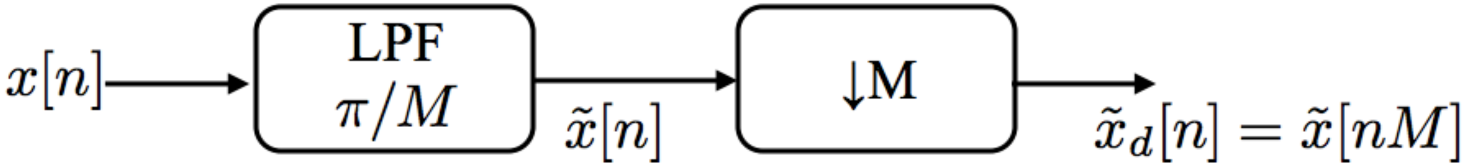
Example

$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j\left(\frac{w}{M} - \frac{2\pi}{M}i\right)}\right)$$

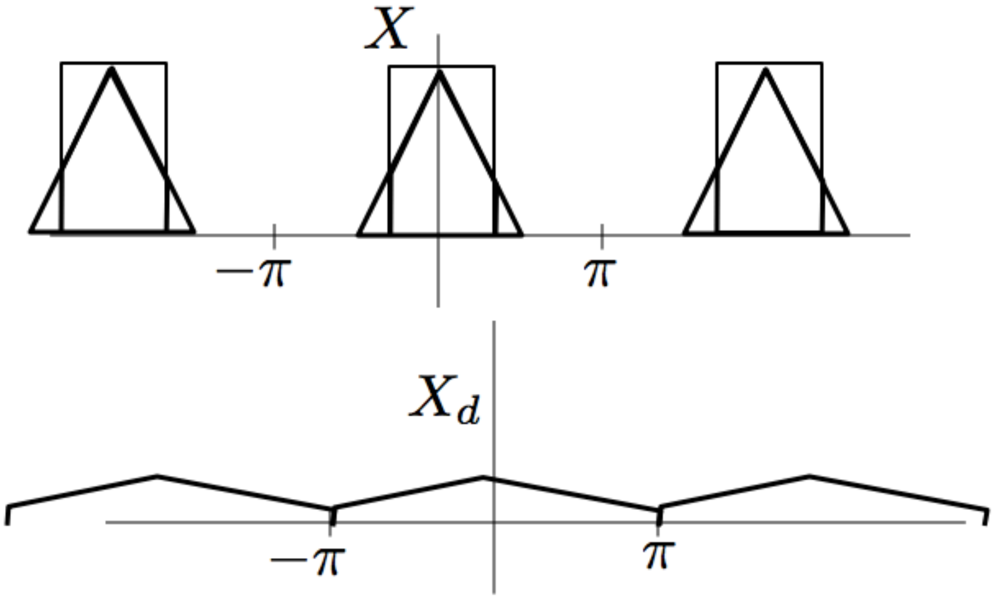


Scale by M=3
Shift by (i=1)*2π/(M=3)
Shift by (i=2)*2π/(M=3)

Example



$x[n]$
LPF
π/M
$\tilde{x}[n]$
$\downarrow$M
$\tilde{x}_d[n] = \tilde{x}[nM]$



X
$-\pi$
π
X_d
$-\pi$
π

M=3

Upsampling

- Definition: Increasing the sampling rate by an integer number

$$x[n] = x_c(nT)$$

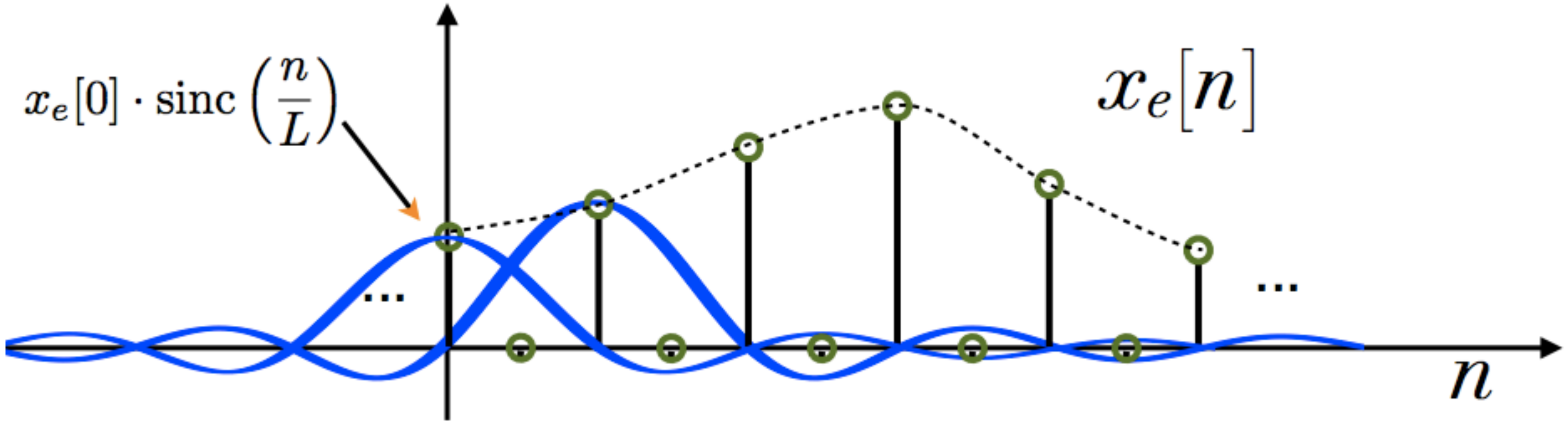
$$x_i[n] = x_c(nT') \quad \text{where } T' = \frac{T}{L} \qquad L \text{ integer}$$

Obtain $x_i[n]$ from $x[n]$ in two steps:

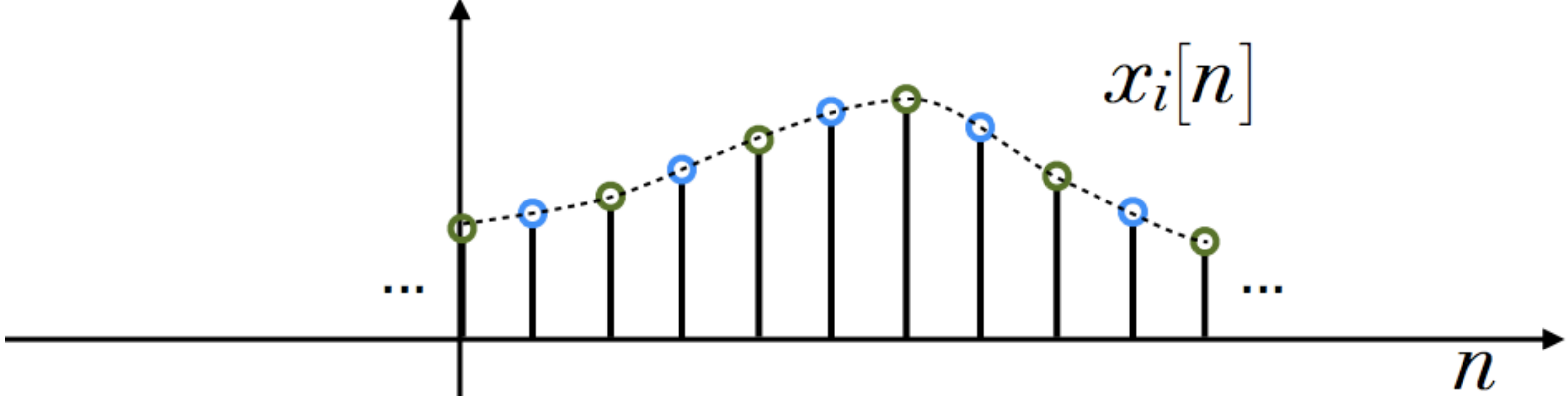
(1) Generate: $x_e[n] = \begin{cases} x[n/L] & n = 0, \ \pm L, \ \pm 2L, \cdots \\ 0 & \text{otherwise} \end{cases}$

Upsampling

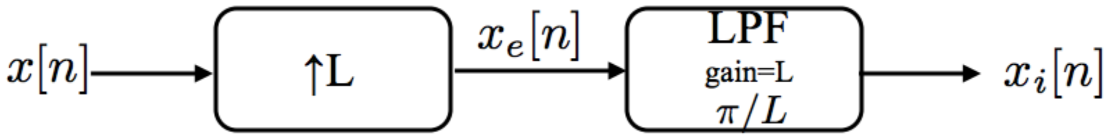
(2) Obtain $x_i[n]$ from $x_e[n]$ by bandlimited interpolation:

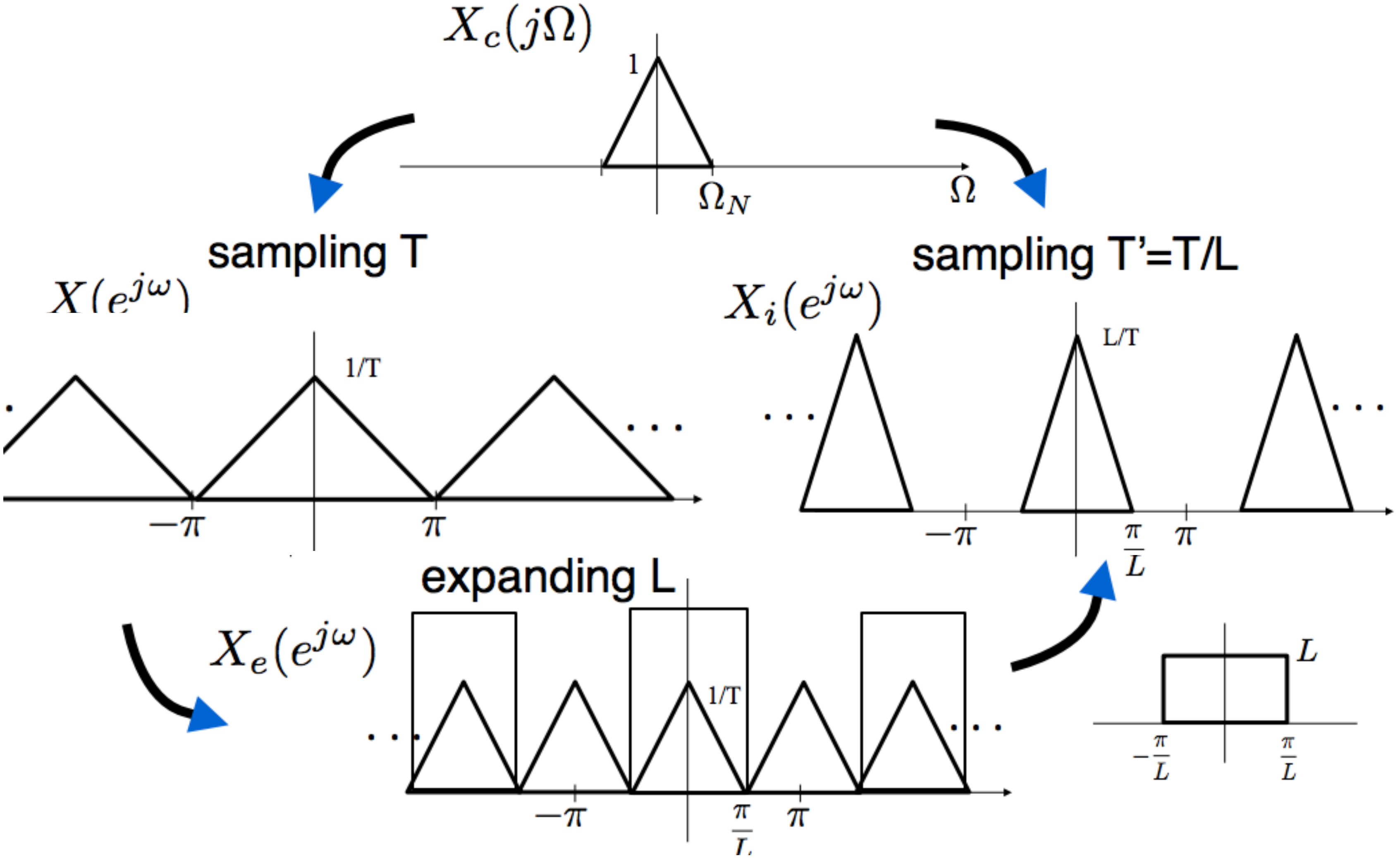


$$x_i[n] = x_e[n] * \mathrm{sinc}\left(\frac{n}{L}\right)$$



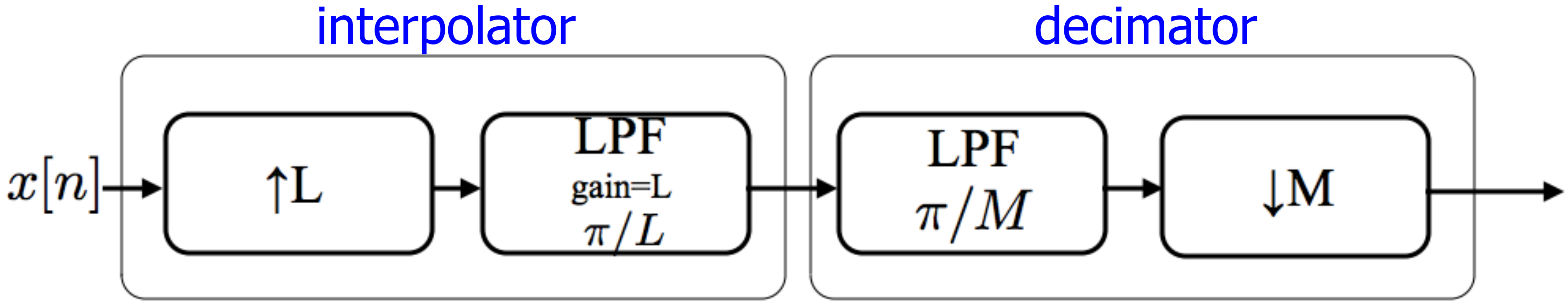
Example


$x[n]$
↑L
$x_e[n]$
LPF
gain=L
π/L
$x_i[n]$

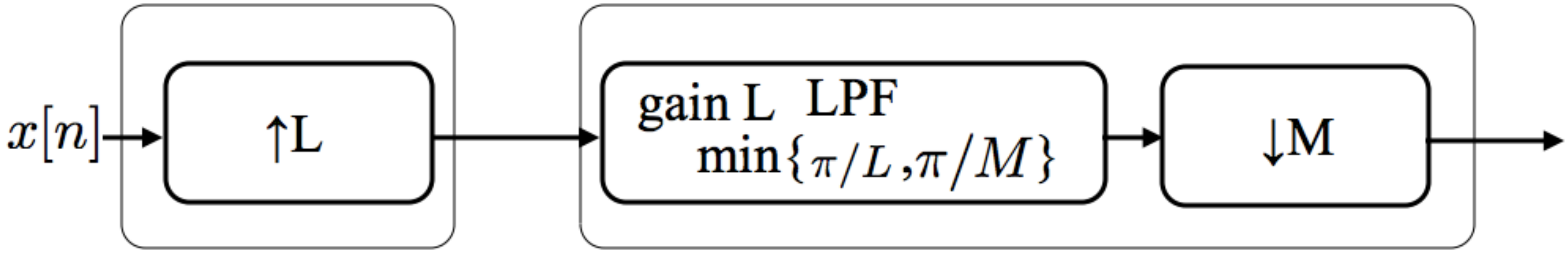

$X_c(j\Omega)$
1
Ω_N
Ω
sampling T
$X(e^{j\omega})$
1/T
$-\pi$
π
sampling T'=T/L
$X_i(e^{j\omega})$
L/T
$-\pi$
$\frac{\pi}{L}$
π
expanding L
$X_e(e^{j\omega})$
1/T
$-\pi$
$\frac{\pi}{L}$
π
L
$-\frac{\pi}{L}$
$\frac{\pi}{L}$

Non-integer Sampling

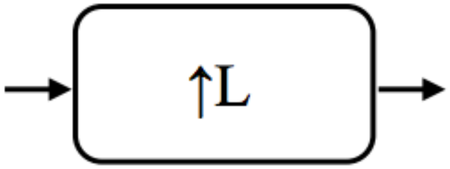
- T'=TM/L
  - Upsample by L, then downsample by M



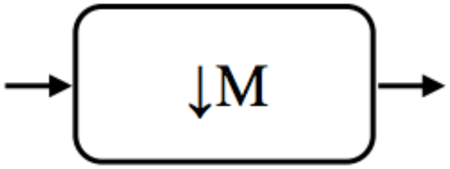
Or,



Interchanging Operations



“expander”



“compressor”

not LTI!

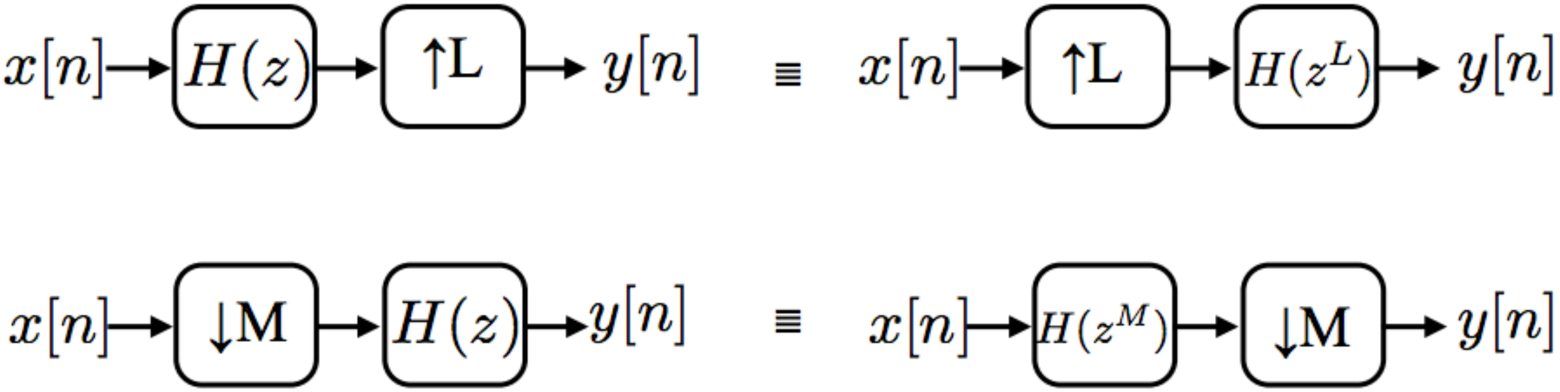
Upsampling
-expanding in time
-compressing in frequency

Downsampling
-compressing in time
-expanding in frequency

Interchanging Operations - Summary

Filter and expander

Expander and expanded filter*



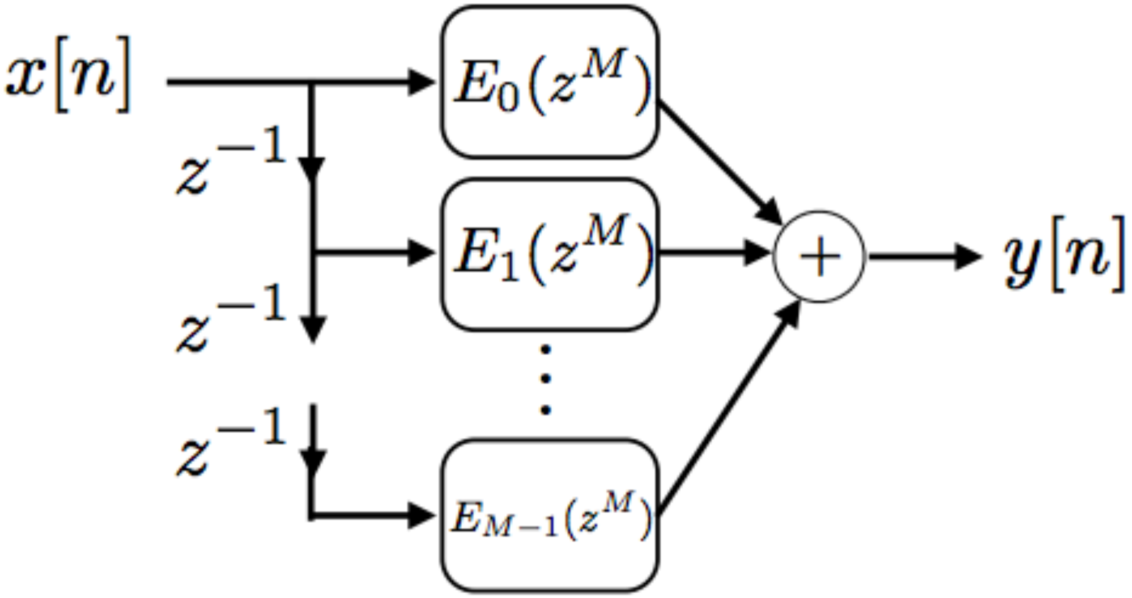
Compressor and filter

Expanded filter* and compressor

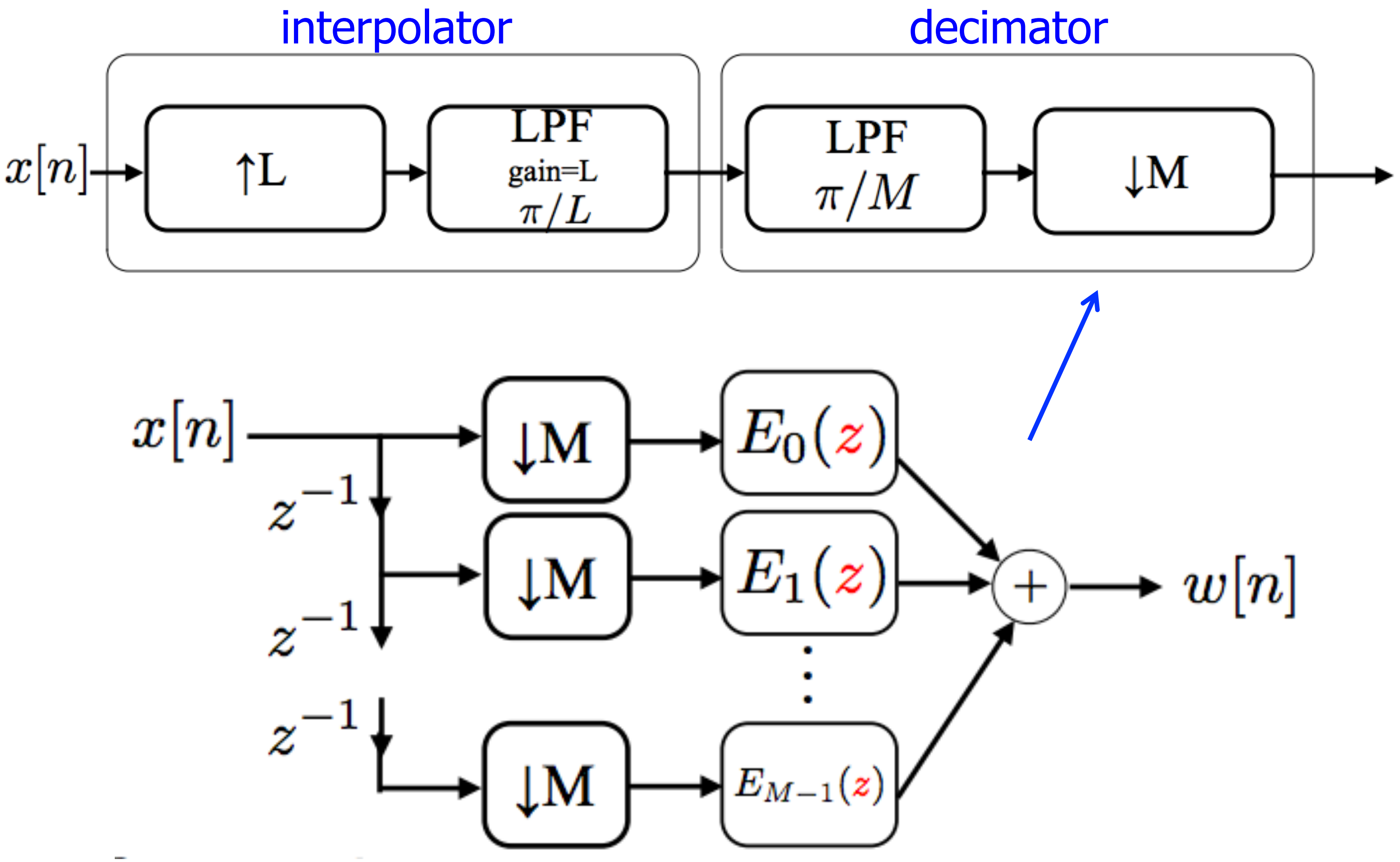
*Expanded filter = expanded impulse response, compressed freq response

Polyphase Decomposition

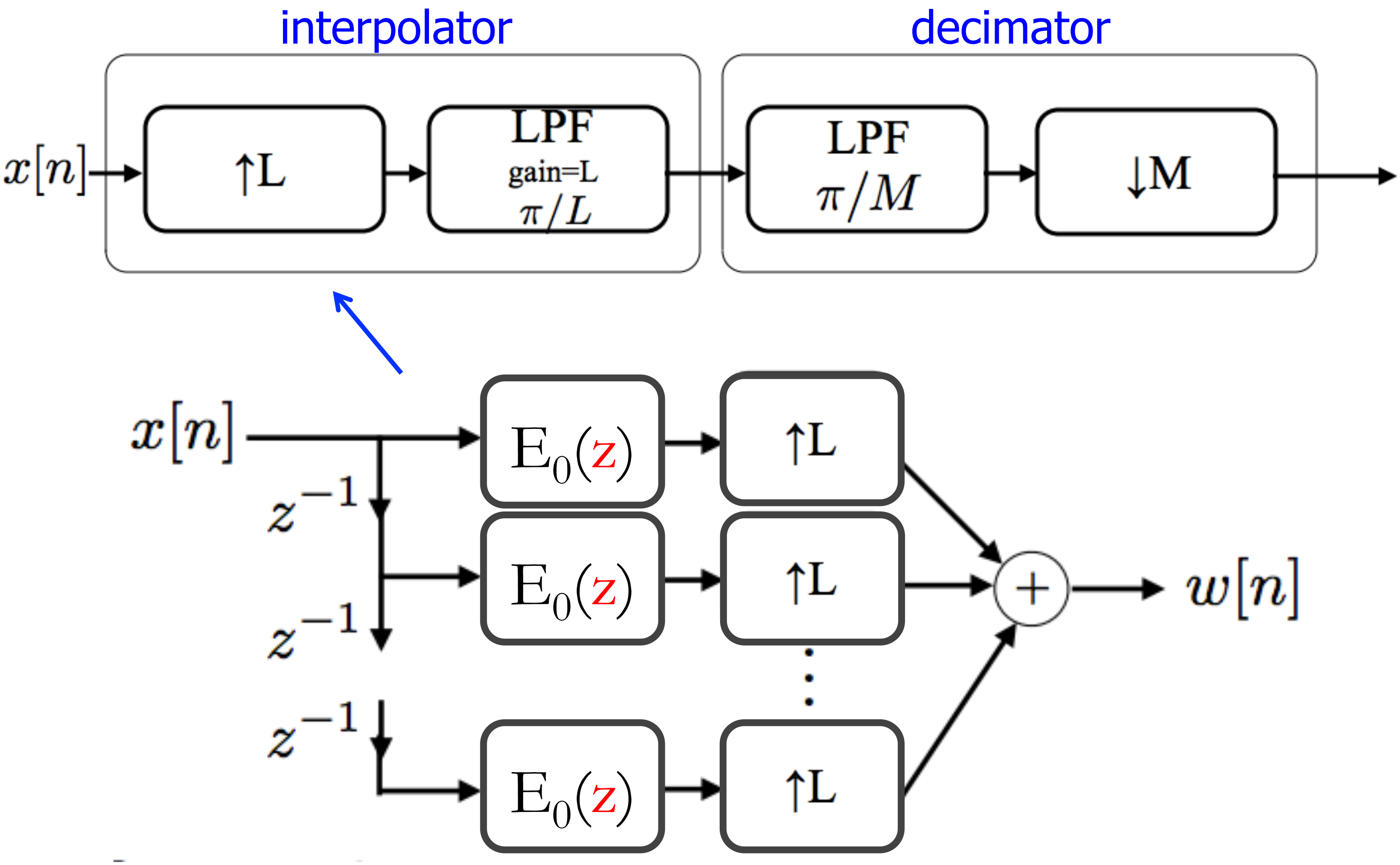
$$H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k}$$



Polyphase Implementation of Decimator



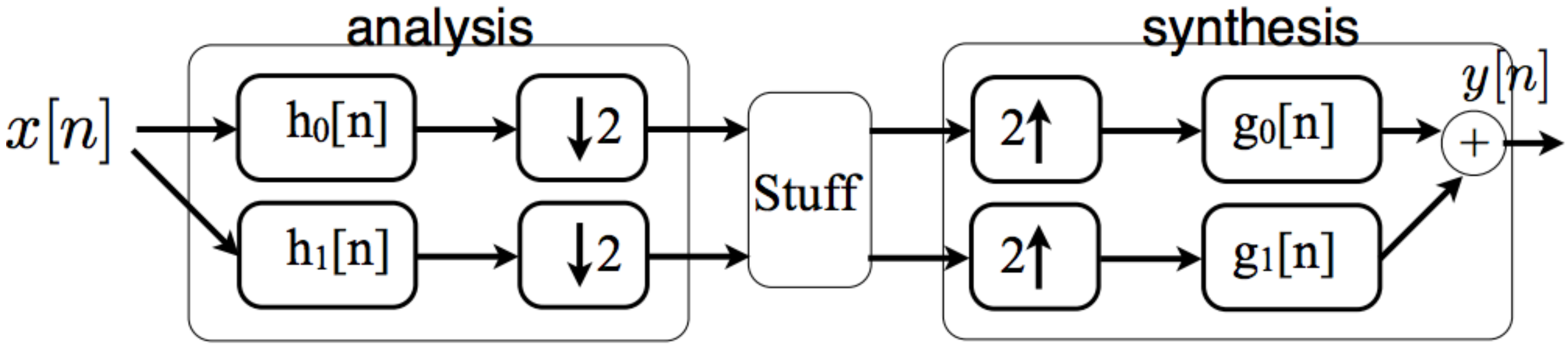
Polyphase Implementation of Interpolation



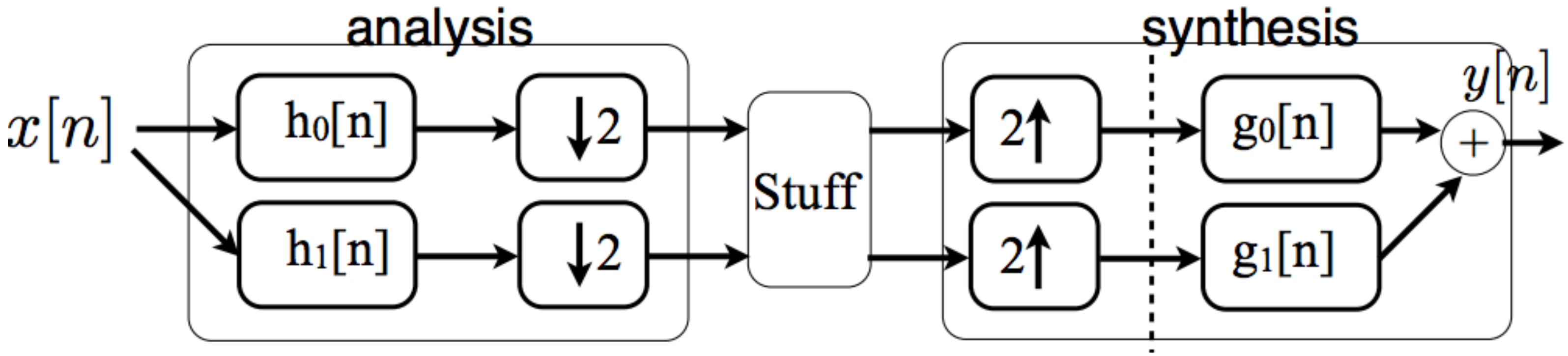
Multi-Rate Filter Banks

- Use filter banks to operate on a signal differently in different frequency bands
  - To save computation, reduce the rate after filtering

- $h_0[n]$ is low-pass, $h_1[n]$ is high-pass
  - Often $h_1[n]=e^{j\pi n}h_0[n]$ ← shift freq resp by π



Perfect Reconstruction non-Ideal Filters

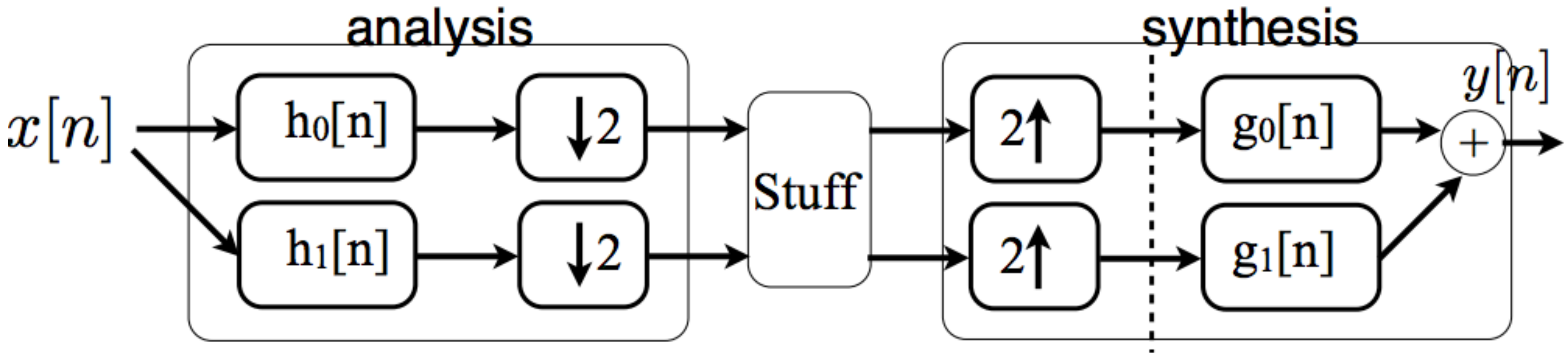


$$
\begin{aligned}
Y(e^{j\omega}) &= \frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j\omega}) + G_1(e^{j\omega})H_1(e^{j\omega})\right] X(e^{j\omega}) \\
&+\frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j(\omega-\pi)}) + G_1(e^{j\omega})H_1(e^{j(\omega-\pi)})\right] X(e^{j(\omega-\pi)})
\end{aligned}
$$

need to cancel!

aliasing

Quadrature Mirror Filters

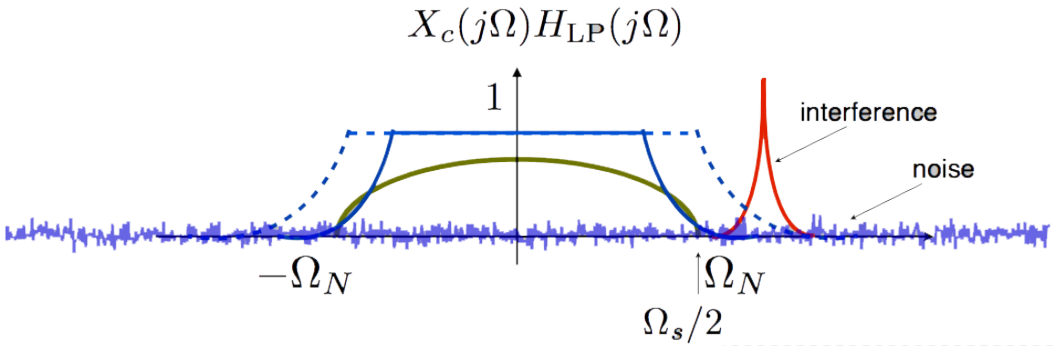


Quadrature mirror filters

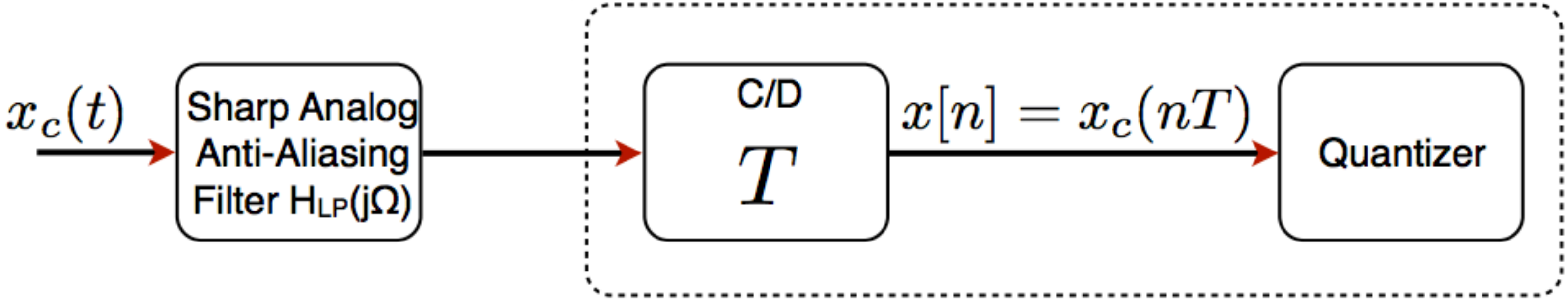
$$\begin{aligned} H_1(e^{j\omega}) &= H_0(e^{j(\omega-\pi)}) \\ G_0(e^{j\omega}) &= 2H_0(e^{j\omega}) \\ G_1(e^{j\omega}) &= -2H_1(e^{j\omega}) \end{aligned}$$

Data Converters

Oversampled ADC

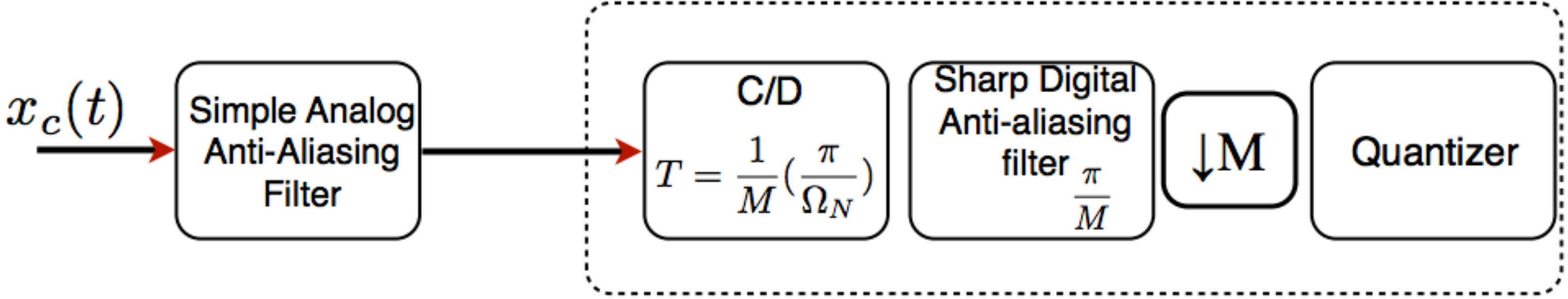


ADC A/D



$x_c(t)$ → Sharp Analog Anti-Aliasing Filter $H_{LP}(j\Omega)$ → C/D T → $x[n] = x_c(nT)$ → Quantizer

Oversampled ADC A/D



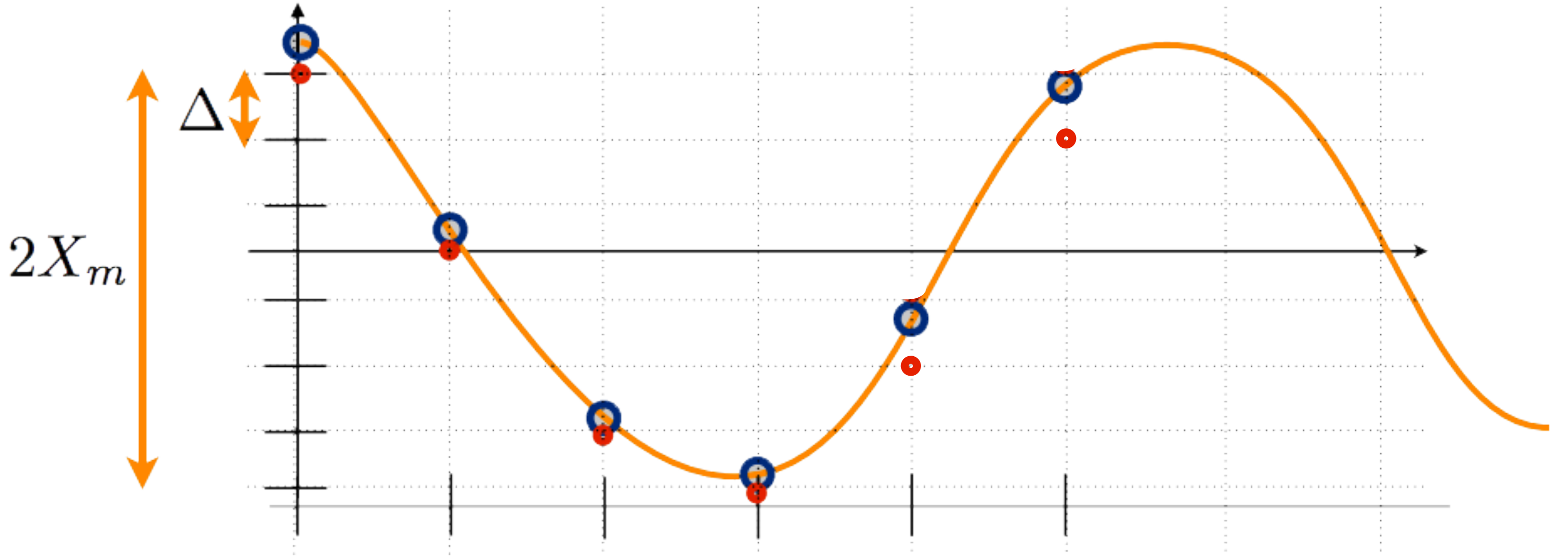
$x_c(t)$ → Simple Analog Anti-Aliasing Filter → C/D $T = \frac{1}{M}(\frac{\pi}{\Omega_N})$ → Sharp Digital Anti-aliasing filter $\frac{\pi}{M}$ → $\downarrow$M → Quantizer

Sampling and Quantization

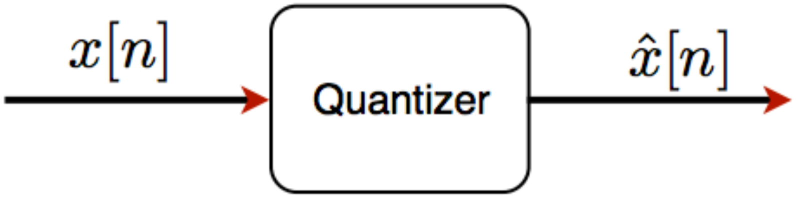
for 2's complement with B+1 bits $-1 \leq \hat{x}_B[n] < 1$

$$\Delta = \frac{2X_m}{2^{B+1}} = \frac{X_m}{2^B}$$

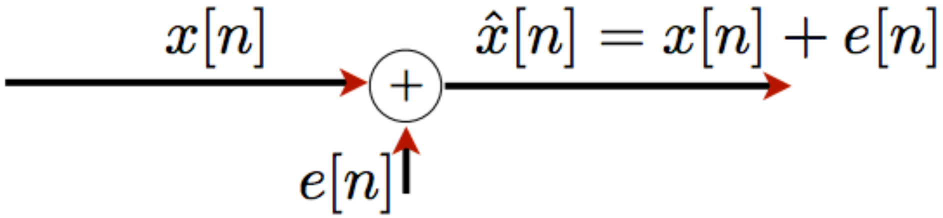
$$\hat{x}[n] = X_m \hat{x}_B[n]$$



Quantization Error



- Model quantization error as noise



- In that case:

$$-\Delta/2 \leq e[n] < \Delta/2$$

$$(-X_m - \Delta/2) < x[n] \leq (X_m - \Delta/2)$$

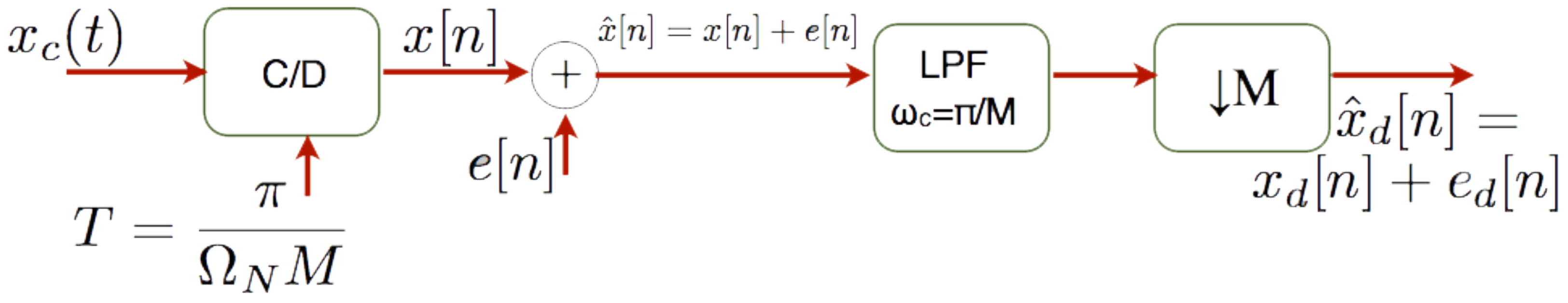
Signal-to-Quantization-Noise Ratio

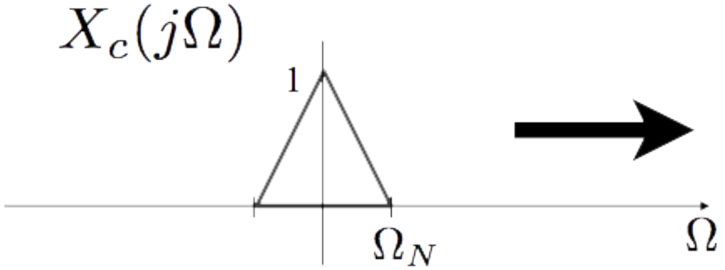
- Assuming full-scale sinusoidal input, we have

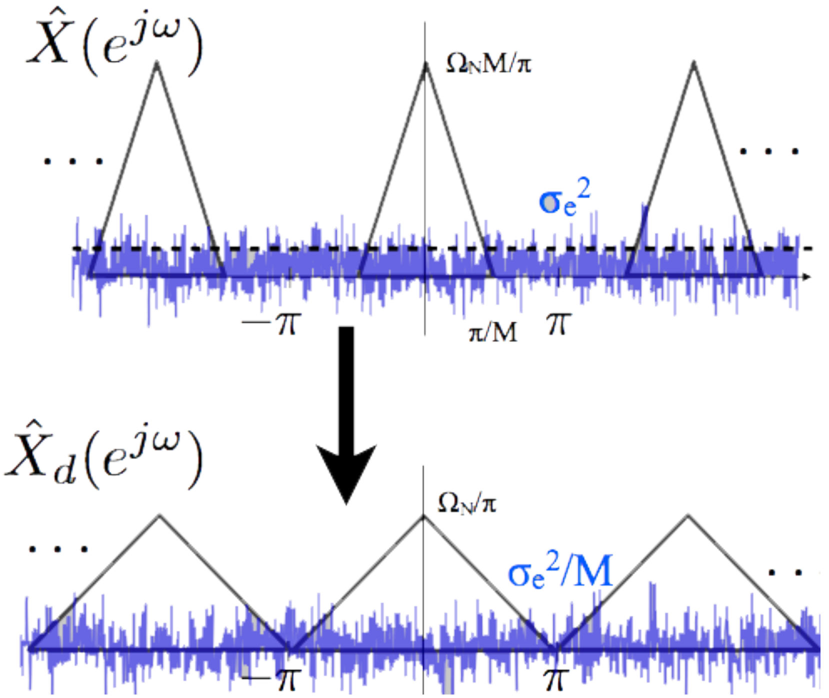
$$SQNR = \frac{P_{sig}}{P_{qnoise}} = \frac{\frac{1}{2}\left(\frac{2^B \Delta}{2}\right)^2}{\frac{\Delta^2}{12}} = 1.5 \times 2^{2B} = 6.02B + 1.76 \;\; dB$$

B (Number of Bits)	SQNR
8	50dB
12	74dB
16	98dB
20	122dB

Quantization Noise with Oversampling

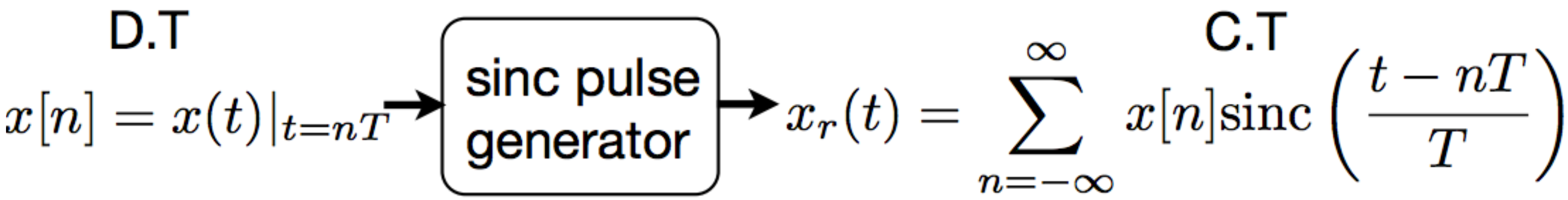






$$\mathrm{SNR}_Q = 6.02B + 10.8 - 20\log_{10}\left(\frac{X_m}{\sigma_x}\right) + \boxed{10\log_{10} M}$$

Practical DAC



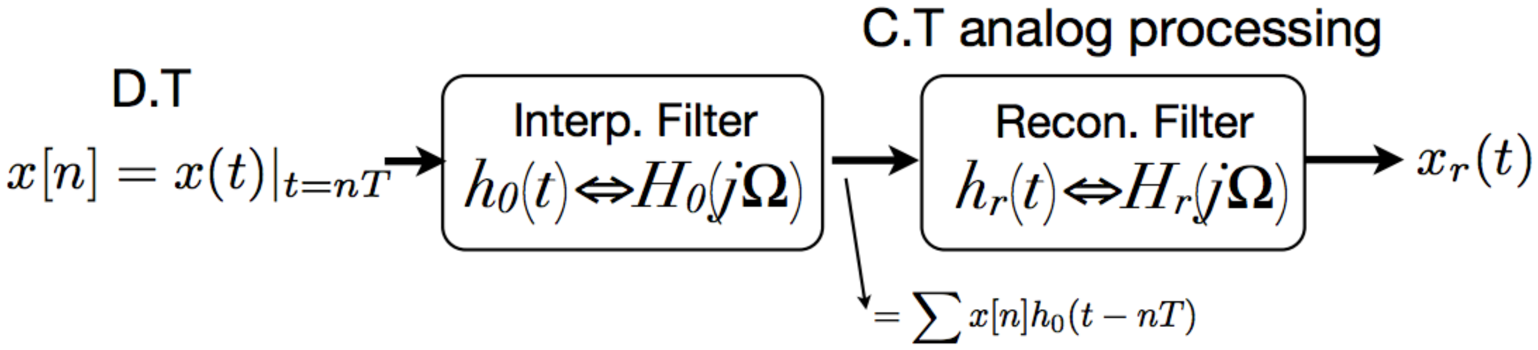
D.T

$$x[n] = x(t)|_{t=nT} \rightarrow \boxed{\text{sinc pulse generator}} \rightarrow x_r(t) = \sum_{n=-\infty}^{\infty} x[n]\text{sinc}\left(\frac{t-nT}{T}\right)$$

C.T

- Scaled train of sinc pulses
- Difficult to generate sinc → Too long!

Practical DAC



C.T analog processing

D.T

$x[n] = x(t)|_{t=nT}$

Interp. Filter

$h_0(t) \Leftrightarrow H_0(j\Omega)$

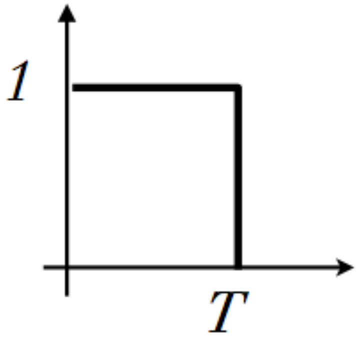
Recon. Filter

$h_r(t) \Leftrightarrow H_r(j\Omega)$

$x_r(t)$

$= \sum x[n] h_0(t - nT)$

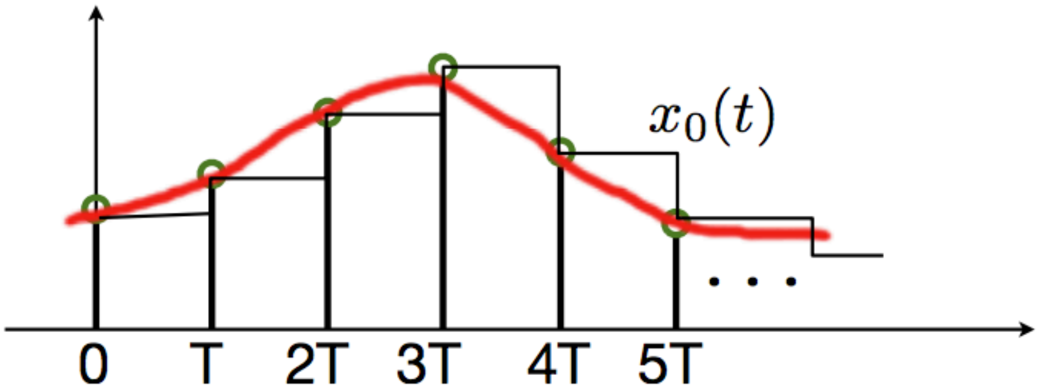
- $h_0(t)$ is finite length pulse → easy to implement
- For example: zero-order hold



$$H_0(j\Omega) = Te^{-j\Omega\frac{T}{2}}\text{sinc}(\frac{\Omega}{\Omega_s})$$

Practical DAC

Zero-Order-Hold interpolation

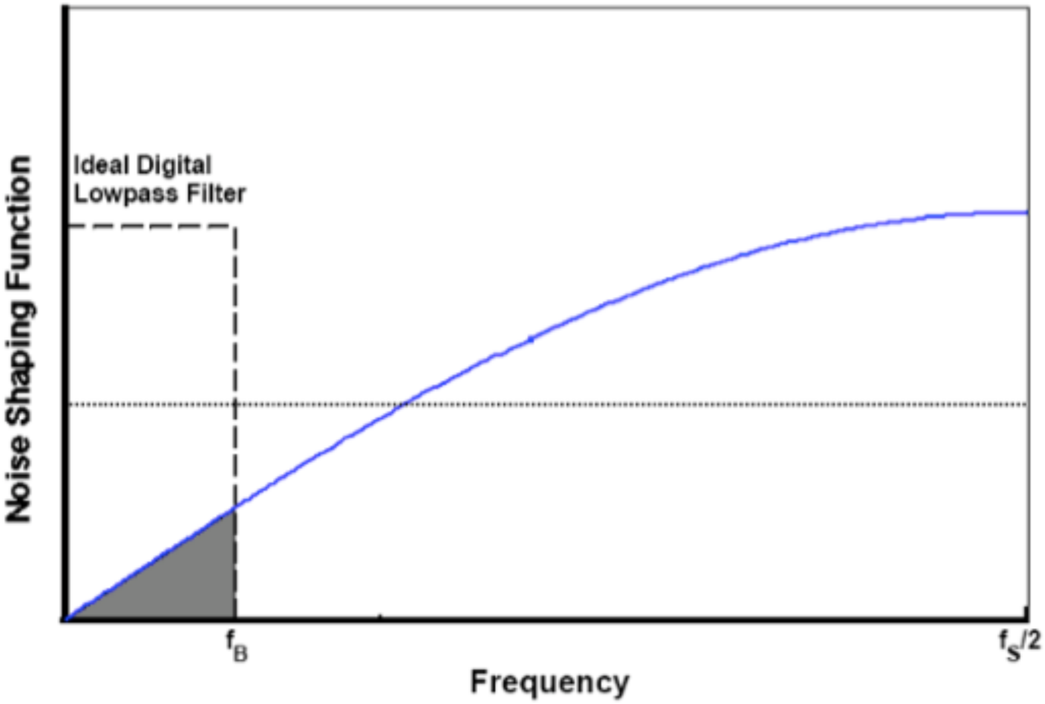


$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n]h_0(t - nT) = h_0(t) * x_s(t)$$

Taking a FT:

$$\begin{aligned} X_(j\Omega) &= H_0(j\Omega)X_s(j\Omega) \\ &= H_0(j\Omega)\frac{1}{T}\sum_{k=-\infty}^{\infty} X(j(\Omega - k\Omega_s)) \end{aligned}$$

Noise Shaping



- Idea: "Somehow" build an ADC that has most of its quantization noise at high frequencies
- Key: Feedback

Frequency Response of Systems

Frequency Response of LTI System

$$Y\left(e^{j\omega}\right) = H\left(e^{j\omega}\right)X\left(e^{j\omega}\right)$$

- We can define a magnitude response

$$\left|Y\left(e^{j\omega}\right)\right| = \left|H\left(e^{j\omega}\right)\right|\left|X\left(e^{j\omega}\right)\right|$$

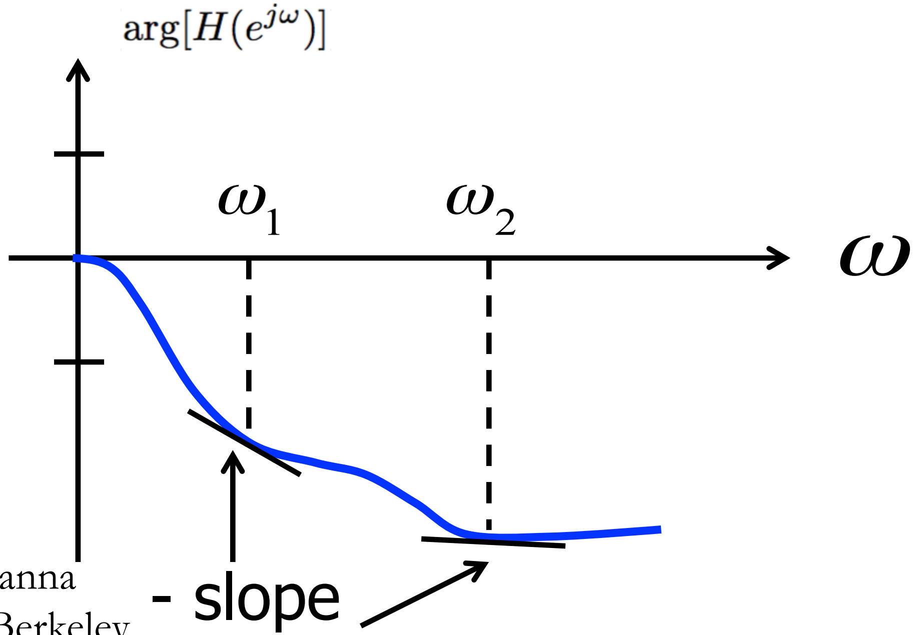
- And a phase response

$$\angle Y\left(e^{j\omega}\right) = \angle H\left(e^{j\omega}\right) + \angle X\left(e^{j\omega}\right)$$

Group Delay

- General phase response at a given frequency can be characterized with group delay, which is related to phase

$$\text{grd}[H(e^{j\omega})] = -\frac{d}{d\omega}\{\arg[H(e^{j\omega})]\}$$



LTI System

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

Example: $y[n] = x[n] + 0.1y[n-1]$

Stable and causal if all poles inside unit circle

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \ldots + a_N z^{-N}} = \frac{b_0}{a_0} \frac{\prod_{k=1}^{M}(1 - c_k z^{-1})}{\prod_{k=1}^{N}(1 - d_k z^{-1})}$$

- Transfer function is not unique without ROC
  - If diff. eq represents LTI and causal system, ROC is region outside all singularities
  - If diff. eq represents LTI and stable system, ROC includes unit circle in z-plane

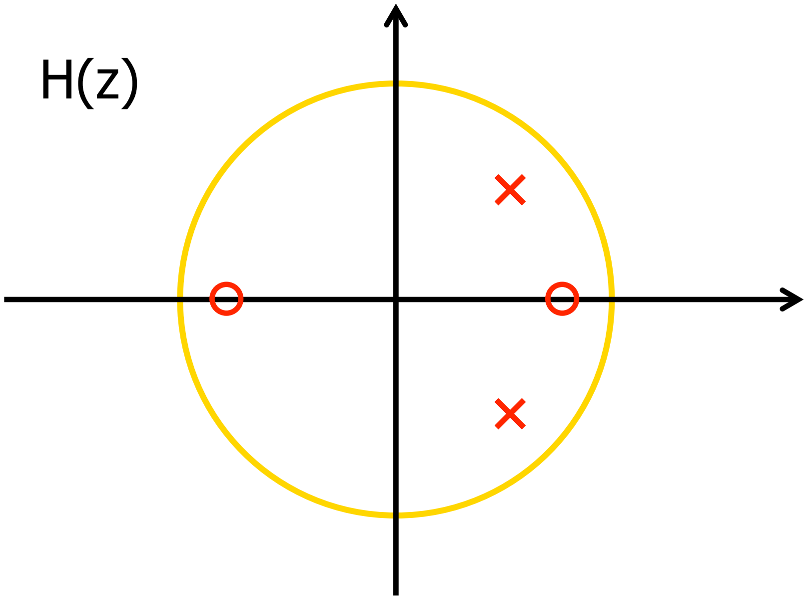
General All-Pass Filter

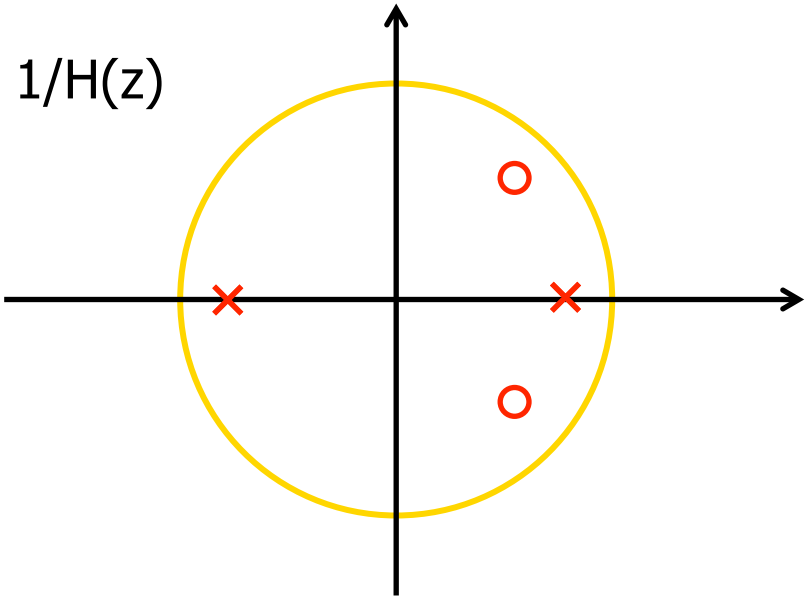
- d_k=real pole, e_k=complex poles paired w/ conjugate, e_k^*

$$H_{\mathrm{ap}}(z) = A \prod_{k=1}^{M_r} \frac{z^{-1} - d_k}{1 - d_k z^{-1}} \prod_{k=1}^{M_c} \frac{(z^{-1} - e_k^*)(z^{-1} - e_k)}{(1 - e_k z^{-1})(1 - e_k^* z^{-1})}$$

Minimum-Phase Systems

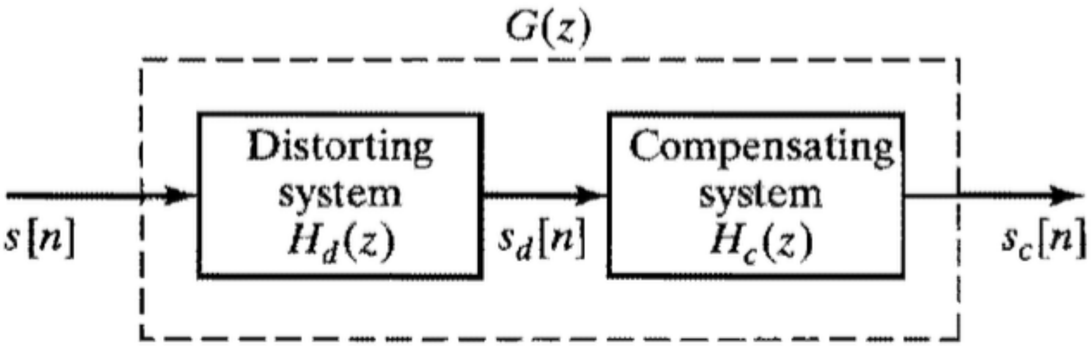
- Definition: A stable and causal system H(z) (i.e. poles inside unit circle) whose inverse 1/H(z) is also stable and causal (i.e. zeros inside unit circle)
  - All poles and zeros inside unit circle





Min-Phase Decomposition Purpose

- Have some distortion that we want to compensate for:



- If $H_d(z)$ is min phase, easy:
  - $H_c(z)=1/H_d(z)$ ← also stable and causal
- Else, decompose $H_d(z)=H_{d,min}(z)\ H_{d,ap}(z)$
  - $H_c(z)=1/H_{d,min}(z) \rightarrow H_d(z)H_c(z)=H_{d,ap}(z)$
    - Compensate for magnitude distortion

Generalized Linear Phase

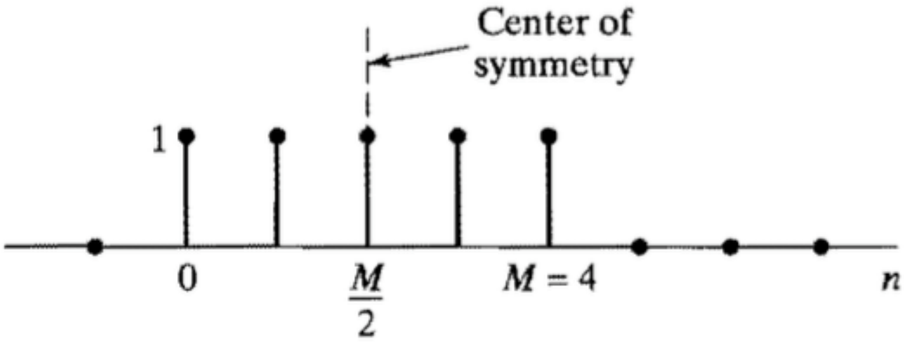
- An LTI system has generalized linear phase if frequency response $H(e^{j\omega})$ can be expressed as:

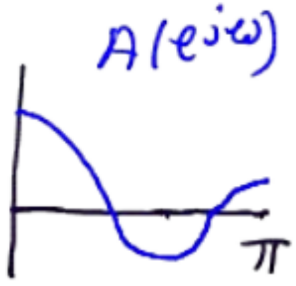
$$\boxed{H(e^{j\omega}) = A(\omega)e^{-j\omega\alpha + j\beta}, \quad |\omega| < \pi}$$

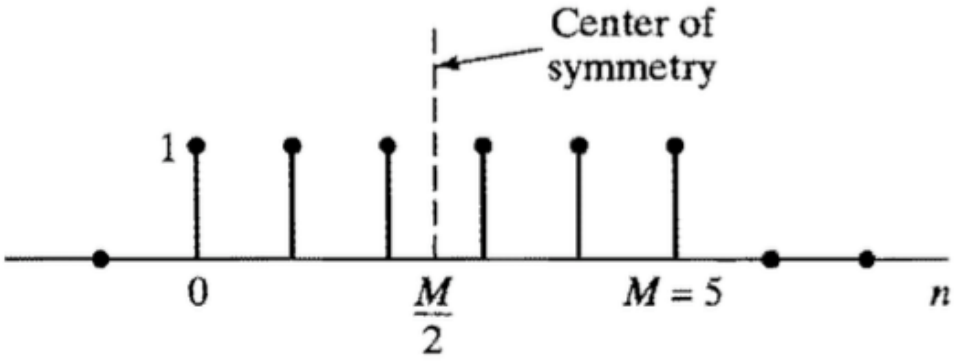
- Where A(ω) is a real function.

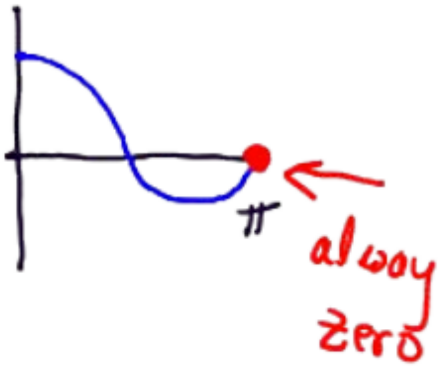
- What is the group delay?

FIR GLP: Type I and II

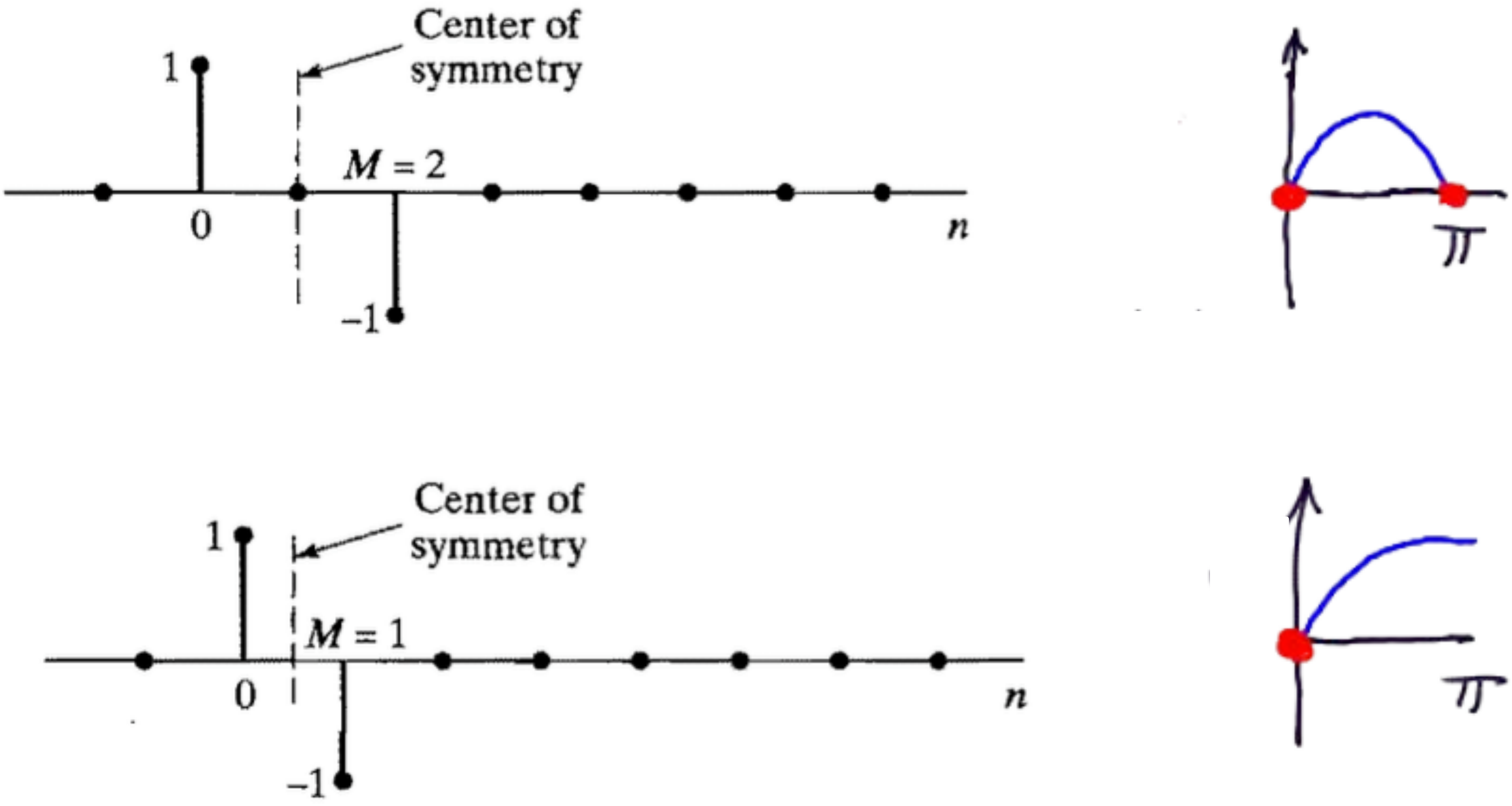








FIR GLP: Type III and IV



Zeros of GLP System

- FIR GLP System Function

$$H(z) = \sum_{n=0}^{M} h[n]z^{-n}$$

Real system → zeros occur in conjugate-reciprocal groups of 4

$$(1 - re^{j\theta}z^{-1})(1 - re^{-j\theta}z^{-1})(1 - r^{-1}e^{j\theta}z^{-1})(1 - r^{-1}e^{-j\theta}z^{-1})$$

- If zero is on unit circle (r=1)

$$(1 - e^{j\theta}z^{-1})(1 - e^{-j\theta}z^{-1}).$$

- If zero is real and not on unit circle (θ =0)

$$(1 \pm rz^{-1})(1 \pm r^{-1}z^{-1}).$$

FIR Filter Design

FIR Design by Windowing

- Given desired frequency response, $H_d(e^{j\omega})$, find an impulse response

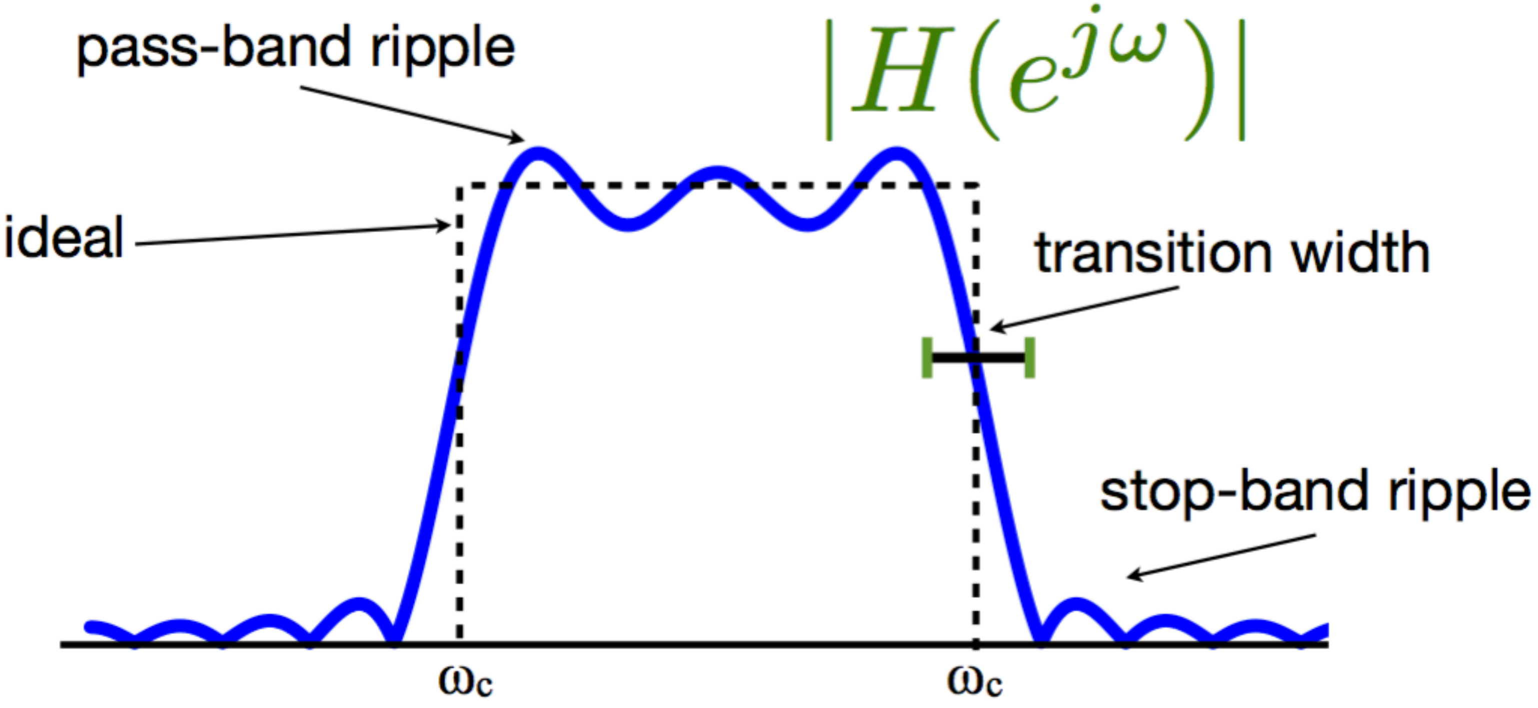
$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega$$

ideal

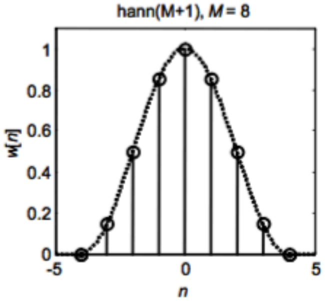
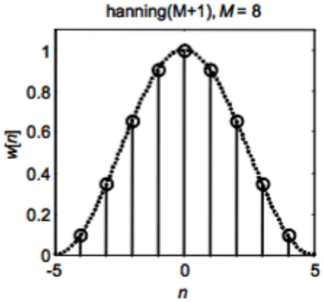
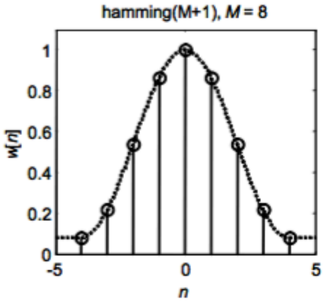
- Obtain the Mth order causal FIR filter by truncating/windowing it

$$h[n] = \begin{Bmatrix} h_d[n]w[n] & 0 \leq n \leq M \\ 0 & \text{otherwise} \end{Bmatrix}$$

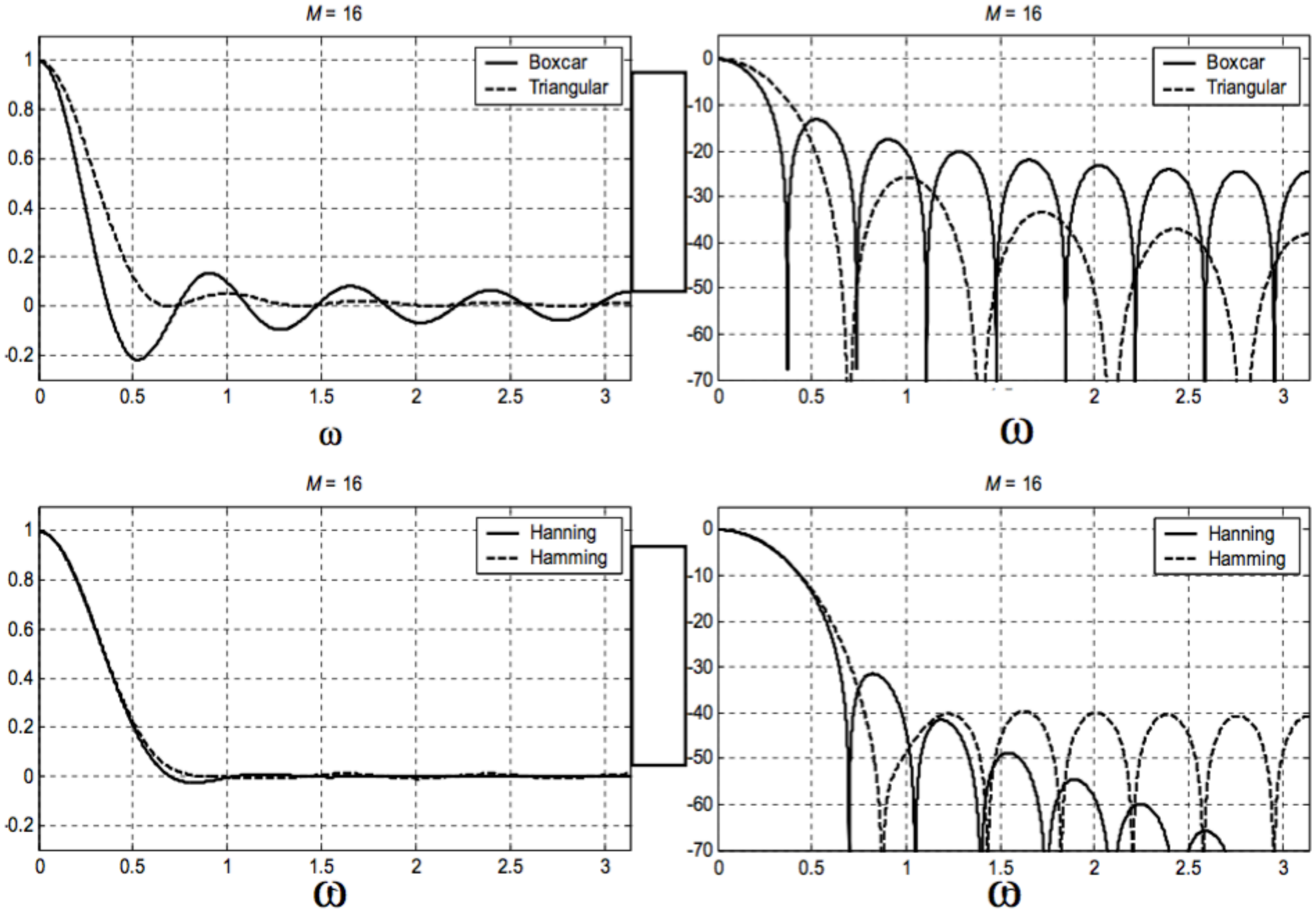
FIR Design by Windowing



Tapered Windows

Name(s)	**Definition**	**MATLAB Command**	**Graph ($M = 8$)**
Hann	$w[n]=\begin{cases}\frac{1}{2}\left[1+\cos\left(\frac{\pi n}{M/2}\right)\right] & \lvert n\rvert \le M/2 \\ 0 & \lvert n\rvert > M/2\end{cases}$	`hann(M+1)`	 hann(M+1), $M = 8$
Hanning	$w[n]=\begin{cases}\frac{1}{2}\left[1+\cos\left(\frac{\pi n}{M/2+1}\right)\right] & \lvert n\rvert \le M/2 \\ 0 & \lvert n\rvert > M/2\end{cases}$	`hanning(M+1)`	 hanning(M+1), $M = 8$
Hamming	$w[n]=\begin{cases}0.54+0.46\cos\left(\frac{\pi n}{M/2}\right) & \lvert n\rvert \le M/2 \\ 0 & \lvert n\rvert > M/2\end{cases}$	`hamming(M+1)`	 hamming(M+1), $M = 8$

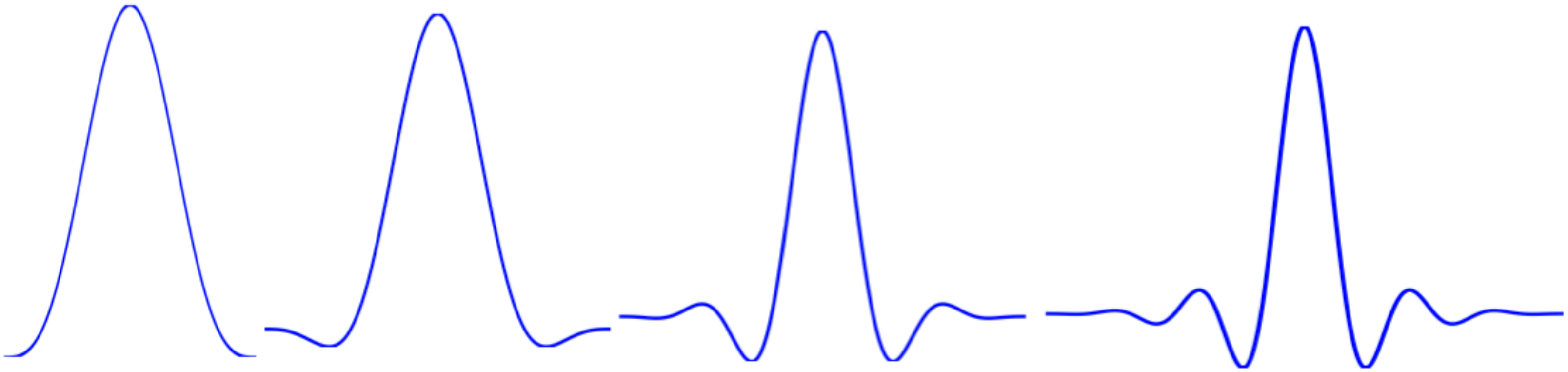
Tradeoff – Ripple vs. Transition Width



Characterization of Filter Shape

Time-Bandwidth Product, a unitless measure

T(BW) = $(M+1)\omega/2\pi$ ⇒ also, total # of zero crossings

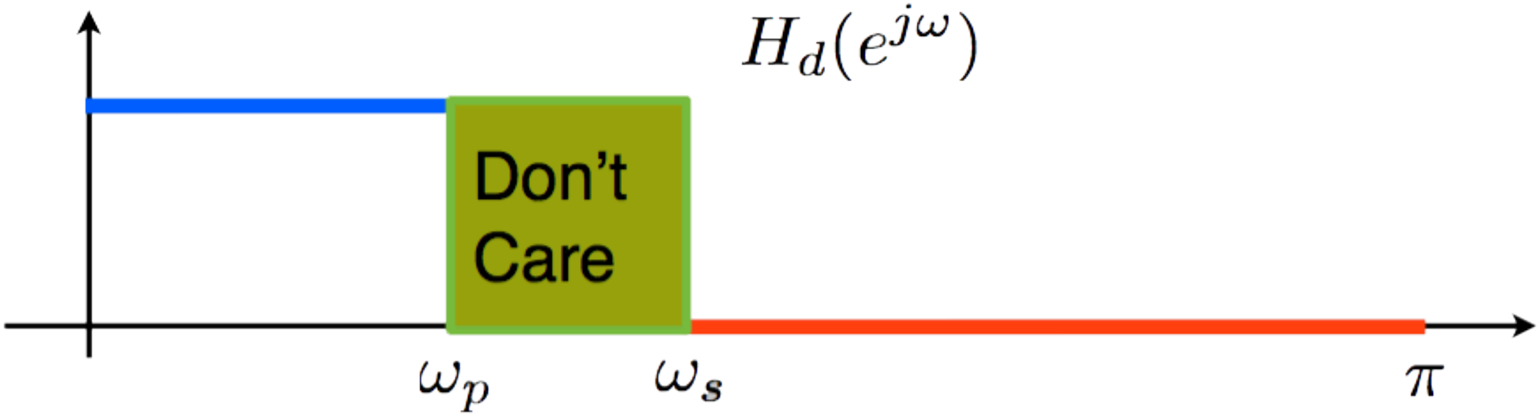


TBW=2 TBW=4 TBW=8 TBW=12

Larger TBW ⇒ More of the "sinc" function

hence, frequency response looks more like a rect function

Optimality



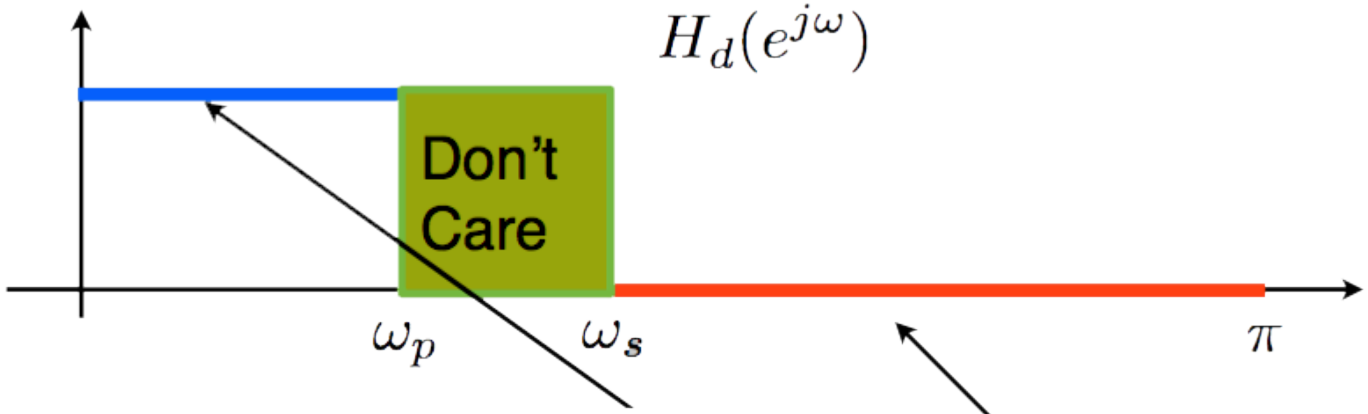
- Least Squares:

$$\text{minimize} \quad \int_{\omega \in \text{care}} |H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

- Variation: Weighted Least Squares:

$$\text{minimize} \quad \int_{-\pi}^{\pi} W(\omega)|H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

Least-Squares Linear Phase Filter



Given M, ω_P, ω_s find the best LS filter:

$$A = \begin{bmatrix} 1 & \cdots & 2\cos(\frac{M}{2}\omega_1) \\ \vdots & & \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_p) \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_s) \\ \vdots & & \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_P) \end{bmatrix}$$

$$b = [1, 1, \cdots, 1, 0, 0, \cdots, 0]^T$$

Least-Squares

$$\text{argmin}_{\tilde{h}} \quad ||A\tilde{h} - b||^2$$

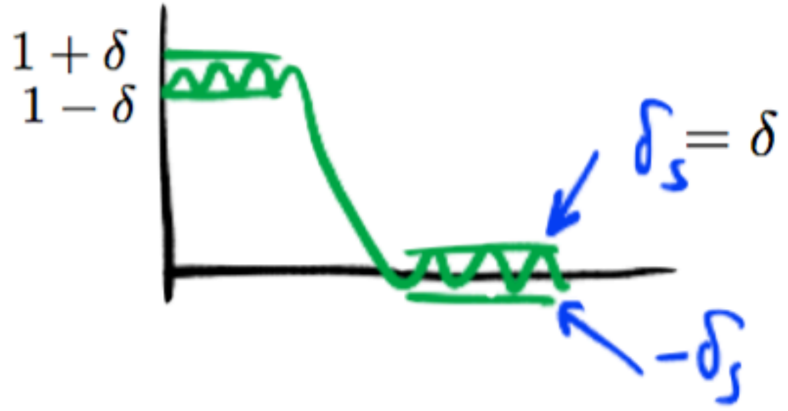
Solution:

$$\tilde{h} = (A^*A)^{-1}A^*b$$

- Result will generally be non-symmetric and complex valued.
- However, if $\tilde{H}(e^{j\omega})$ is real, $\tilde{h}[n]$ should have symmetry!

Min-Max Ripple Design

- Recall, $\tilde{H}(e^{j\omega})$ is symmetric and real
- Given ω_p, ω_s, M, find δ, $\tilde{h}_+$



$$\begin{array}{lll} \text{minimize} & \delta & \\ \text{Subject to :} & & \\ & 1-\delta \le \tilde{H}(e^{j\omega_k}) \le 1+\delta & 0 \le \omega_k \le \omega_p \\ & -\delta \le \tilde{H}(e^{j\omega_k}) \le \delta & \omega_s \le \omega_k \le \pi \\ & \delta > 0 & \end{array}$$

- Formulation is a linear program with solution δ, $\tilde{h}_+$
- A well studied class of problems

IIR Filter Design

IIR Filter Design

- Transform continuous-time filter into a discrete-time filter meeting specs
  - Pick suitable transformation from s (Laplace variable) to z (or t to n)
  - Pick suitable analog $H_c(s)$ allowing specs to be met, transform to $H(z)$

- We've seen this before… impulse invariance

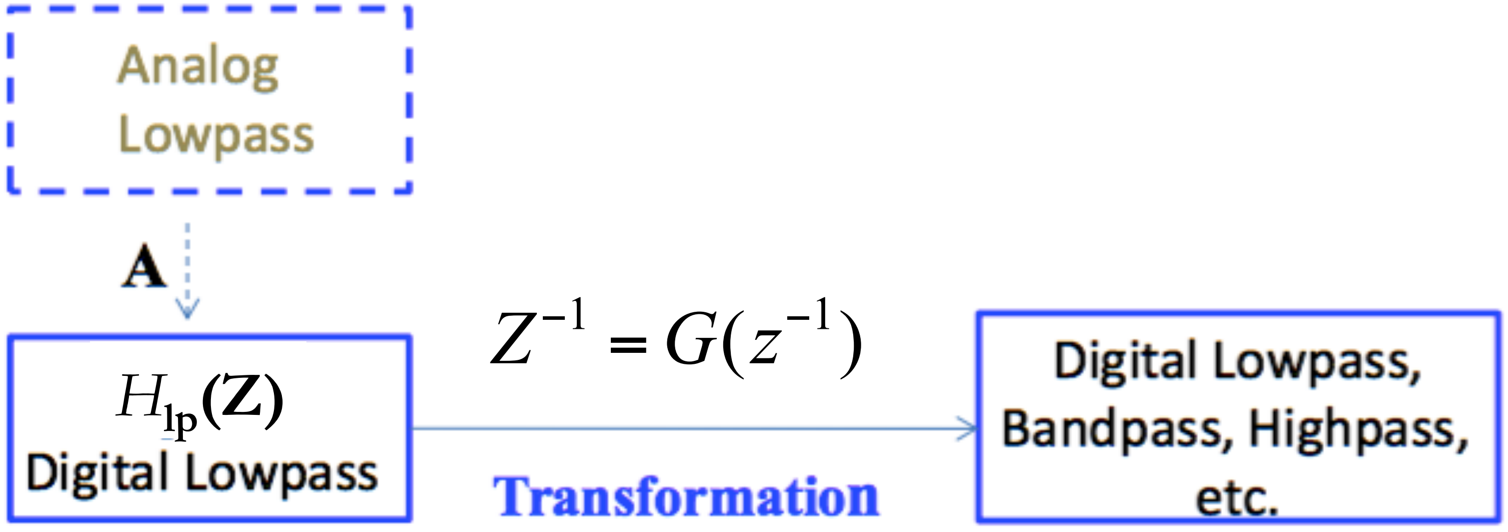
Bilinear Transformation

- The technique uses an algebraic transformation between the variables s and z that maps the entire jΩ-axis in the s-plane to one revolution of the unit circle in the z-plane.

$$s = \frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right);$$

$$H(z) = H_c\left(\frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)\right).$$

Transformation of DT Filters



Analog Lowpass

A

$H_{lp}(\mathbf{Z})$ Digital Lowpass

$Z^{-1} = G(z^{-1})$

Transformation

Digital Lowpass, Bandpass, Highpass, etc.

- Map Z-plane→z-plane with transformation G

$$H(z) = H_{\mathrm{lp}}(Z)\Big|_{Z^{-1}=G(z^{-1})}$$

General Transformations

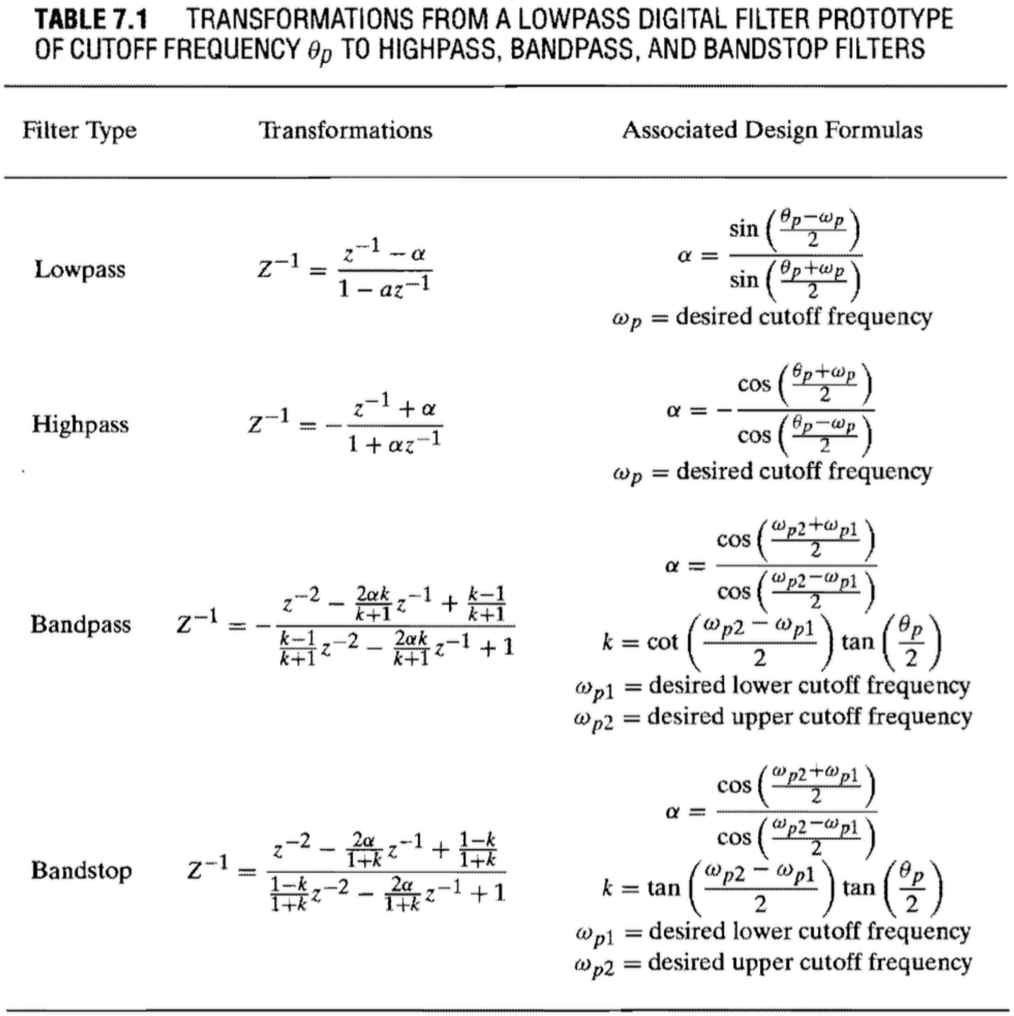
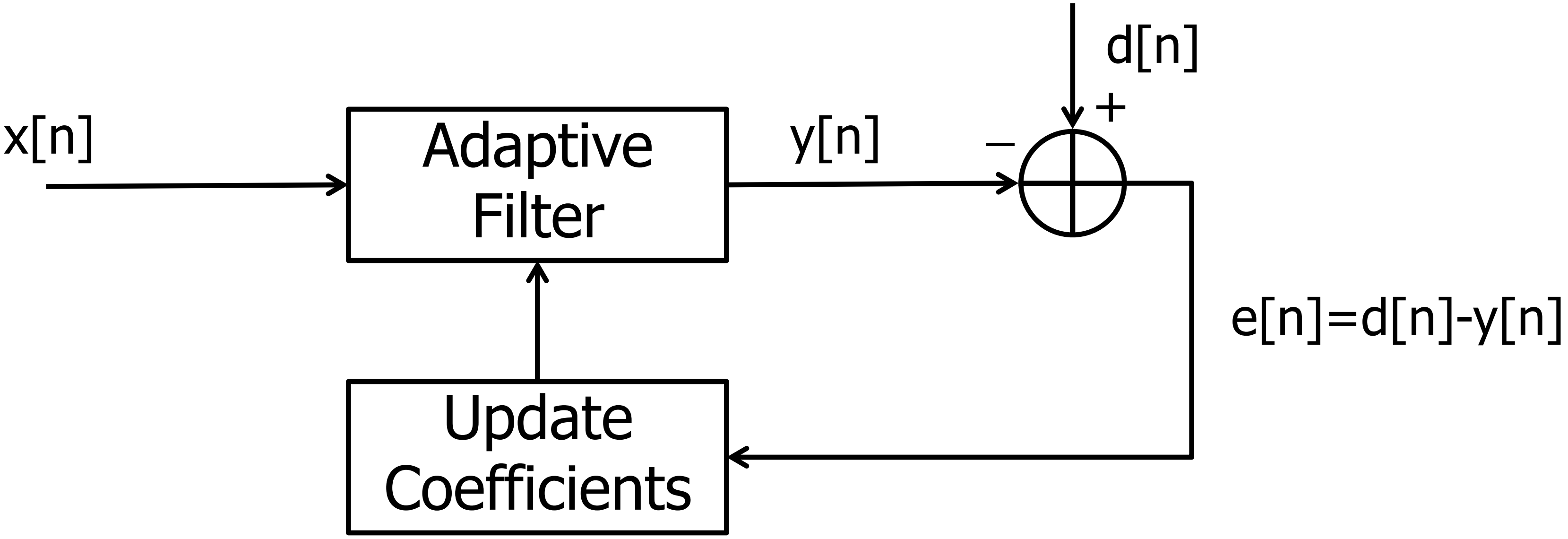


TABLE 7.1 TRANSFORMATIONS FROM A LOWPASS DIGITAL FILTER PROTOTYPE OF CUTOFF FREQUENCY θ_p TO HIGHPASS, BANDPASS, AND BANDSTOP FILTERS

Filter Type	Transformations	Associated Design Formulas
Lowpass	$Z^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$	$\alpha = \frac{\sin\left(\frac{\theta_p - \omega_p}{2}\right)}{\sin\left(\frac{\theta_p + \omega_p}{2}\right)}$ ω_p = desired cutoff frequency
Highpass	$Z^{-1} = -\frac{z^{-1} + \alpha}{1 + \alpha z^{-1}}$	$\alpha = -\frac{\cos\left(\frac{\theta_p + \omega_p}{2}\right)}{\cos\left(\frac{\theta_p - \omega_p}{2}\right)}$ ω_p = desired cutoff frequency
Bandpass	$Z^{-1} = -\frac{z^{-2} - \frac{2\alpha k}{k+1} z^{-1} + \frac{k-1}{k+1}}{\frac{k-1}{k+1} z^{-2} - \frac{2\alpha k}{k+1} z^{-1} + 1}$	$\alpha = \frac{\cos\left(\frac{\omega_{p2} + \omega_{p1}}{2}\right)}{\cos\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right)}$ $k = \cot\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right) \tan\left(\frac{\theta_p}{2}\right)$ ω_{p1} = desired lower cutoff frequency ω_{p2} = desired upper cutoff frequency
Bandstop	$Z^{-1} = \frac{z^{-2} - \frac{2\alpha}{1+k} z^{-1} + \frac{1-k}{1+k}}{\frac{1-k}{1+k} z^{-2} - \frac{2\alpha}{1+k} z^{-1} + 1}$	$\alpha = \frac{\cos\left(\frac{\omega_{p2} + \omega_{p1}}{2}\right)}{\cos\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right)}$ $k = \tan\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right) \tan\left(\frac{\theta_p}{2}\right)$ ω_{p1} = desired lower cutoff frequency ω_{p2} = desired upper cutoff frequency

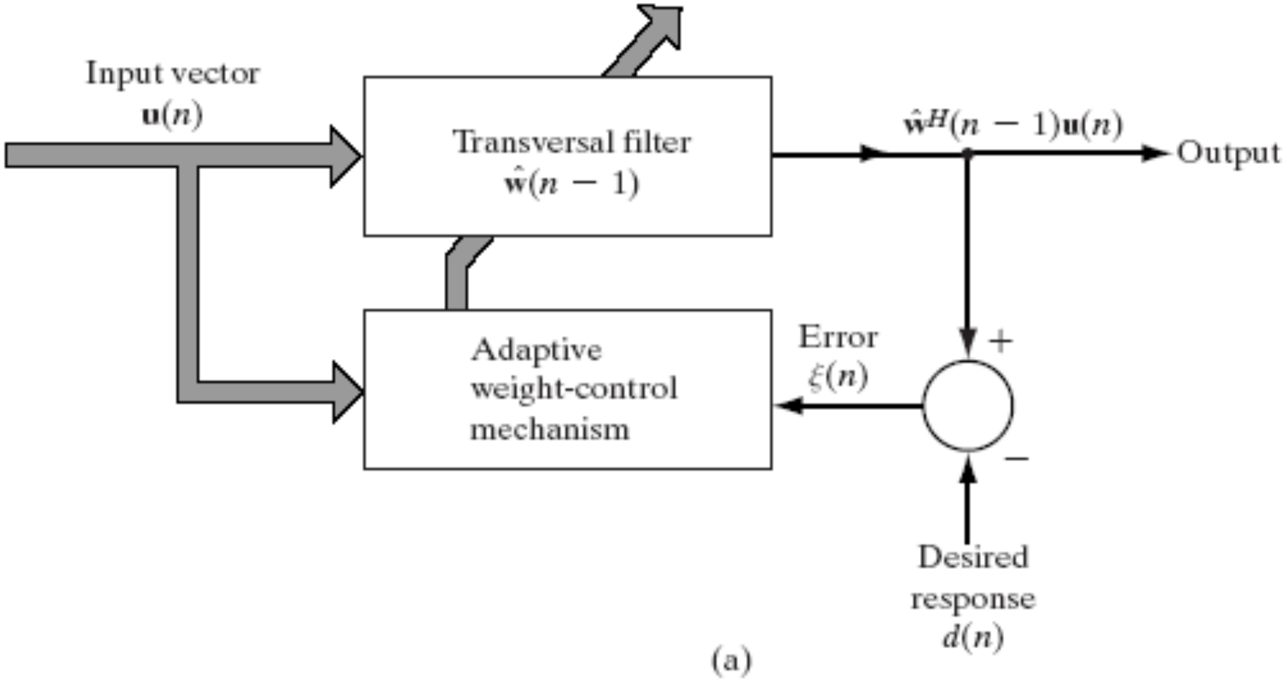
Adaptive Filters

- An adaptive filter is an adjustable filter that processes in time
  - It adapts…



Least-Mean-Square (LMS) Algorithm

- The LMS Algorithm consists of two basic processes
  - Filtering process
    - Calculate the output of FIR filter by convolving input and taps
    - Calculate estimation error by comparing the output to desired signal
  -



Discrete Fourier Transform

Discrete Fourier Transform

- The DFT

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}$$ **Inverse DFT, synthesis**

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}$$ **DFT, analysis**

- It is understood that,

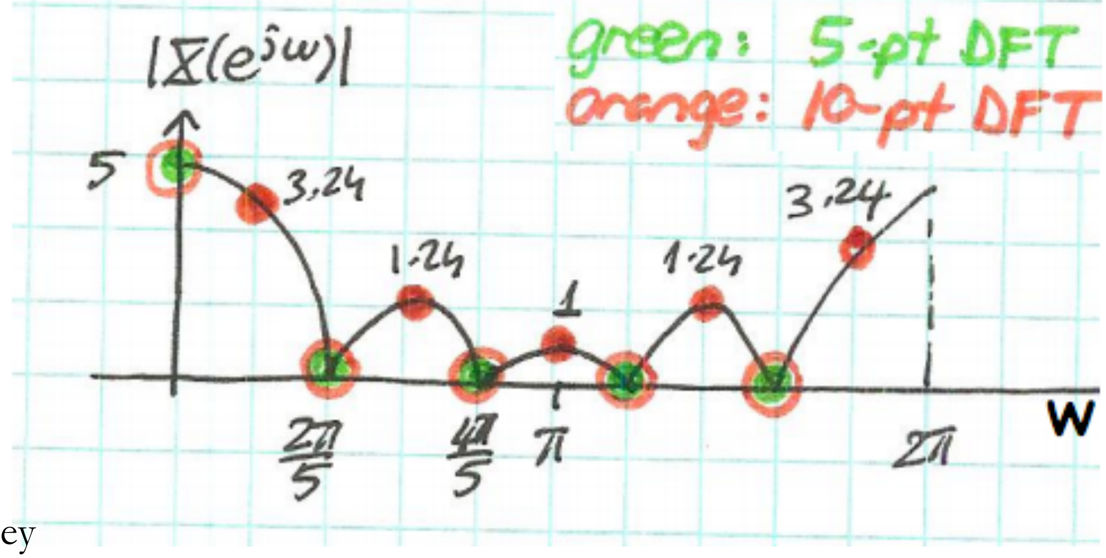
$$\begin{aligned} x[n] &= 0 \quad \text{outside } 0 \leq n \leq N-1 \\ X[k] &= 0 \quad \text{outside } 0 \leq k \leq N-1 \end{aligned}$$

DFT vs DTFT

- Back to example

$$X[k] = \sum_{n=0}^{4} W_{10}^{nk}$$

$$= e^{-j\frac{4\pi}{10}k}\frac{\sin(\frac{\pi}{2}k)}{\sin(\frac{\pi}{10}k)}$$



Circular Convolution

- For $x_1[n]$ and $x_2[n]$ with length N

$$x_1[n] \circledast_N x_2[n] \leftrightarrow X_1[k] \cdot X_2[k]$$

  - Very useful!! (for linear convolutions with DFT)

Linear Convolution via Circular Convolution

- Zero-pad x[n] by P-1 zeros

$$x_{\text{zp}}[n] = \begin{cases} x[n] & 0 \leq n \leq L-1 \\ 0 & L \leq n \leq L+P-2 \end{cases}$$

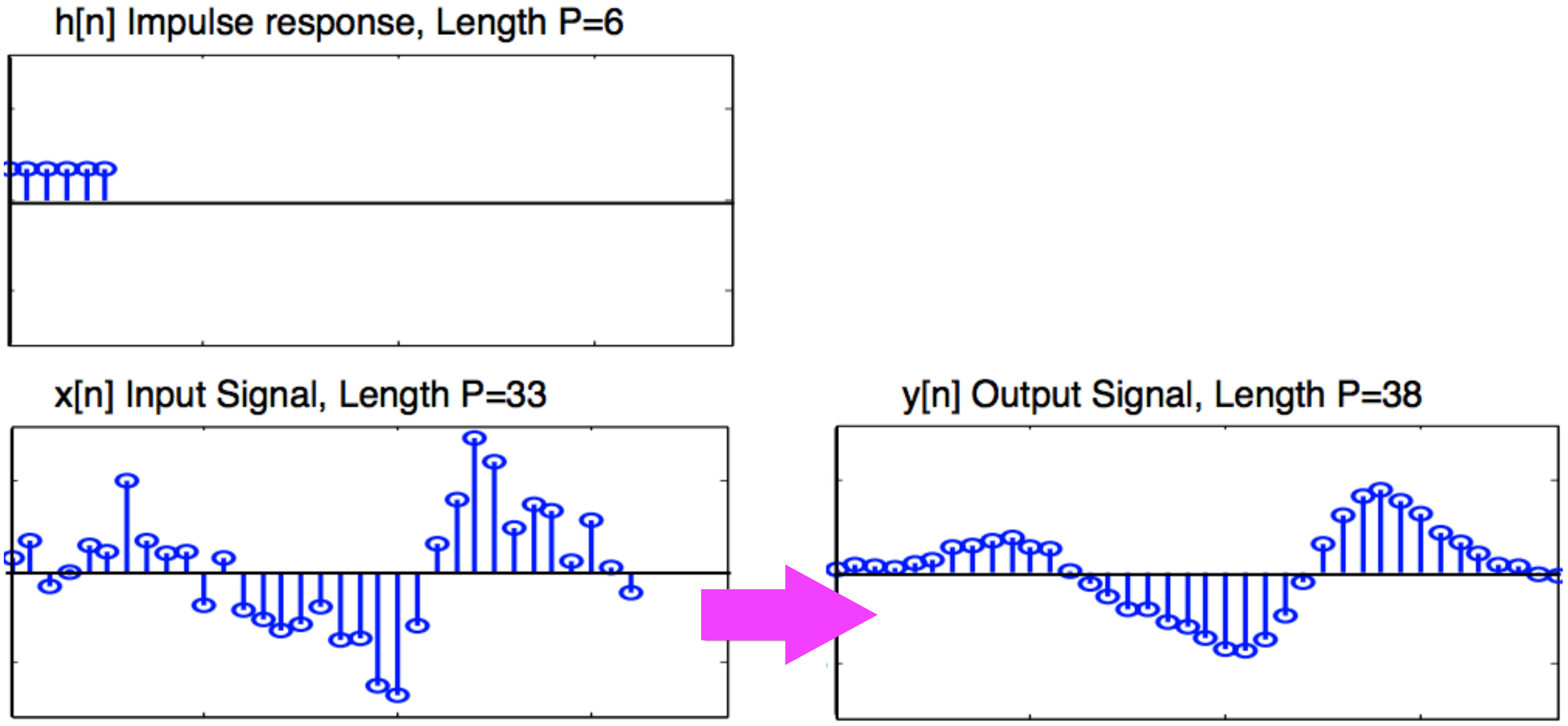
- Zero-pad h[n] by L-1 zeros

$$h_{\text{zp}}[n] = \begin{cases} h[n] & 0 \leq n \leq P-1 \\ 0 & P \leq n \leq L+P-2 \end{cases}$$

- Now, both sequences are length M=L+P-1

Block Convolution

Example:



h[n] Impulse response, Length P=6

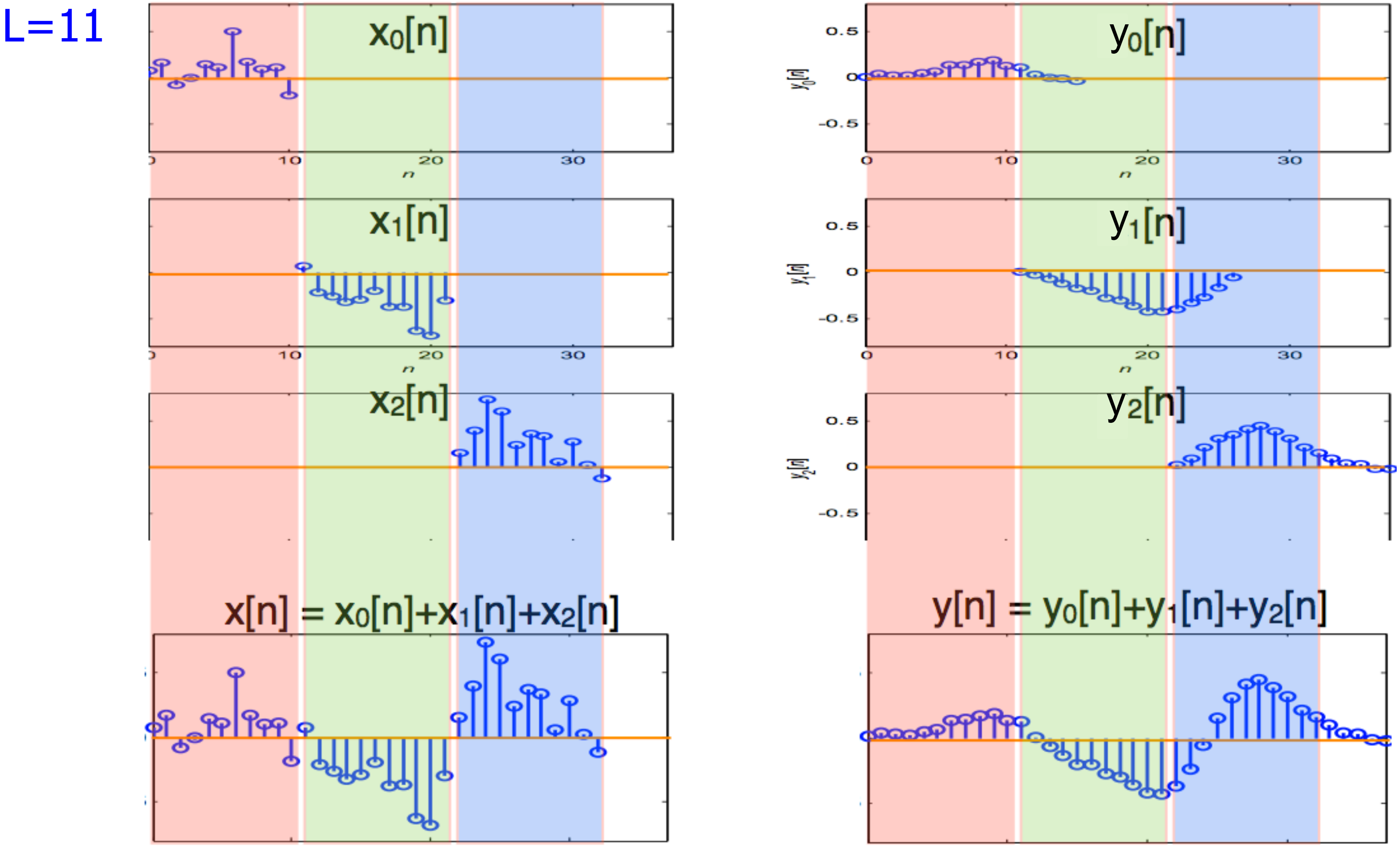
x[n] Input Signal, Length P=33

y[n] Output Signal, Length P=38

Example of Overlap-Add

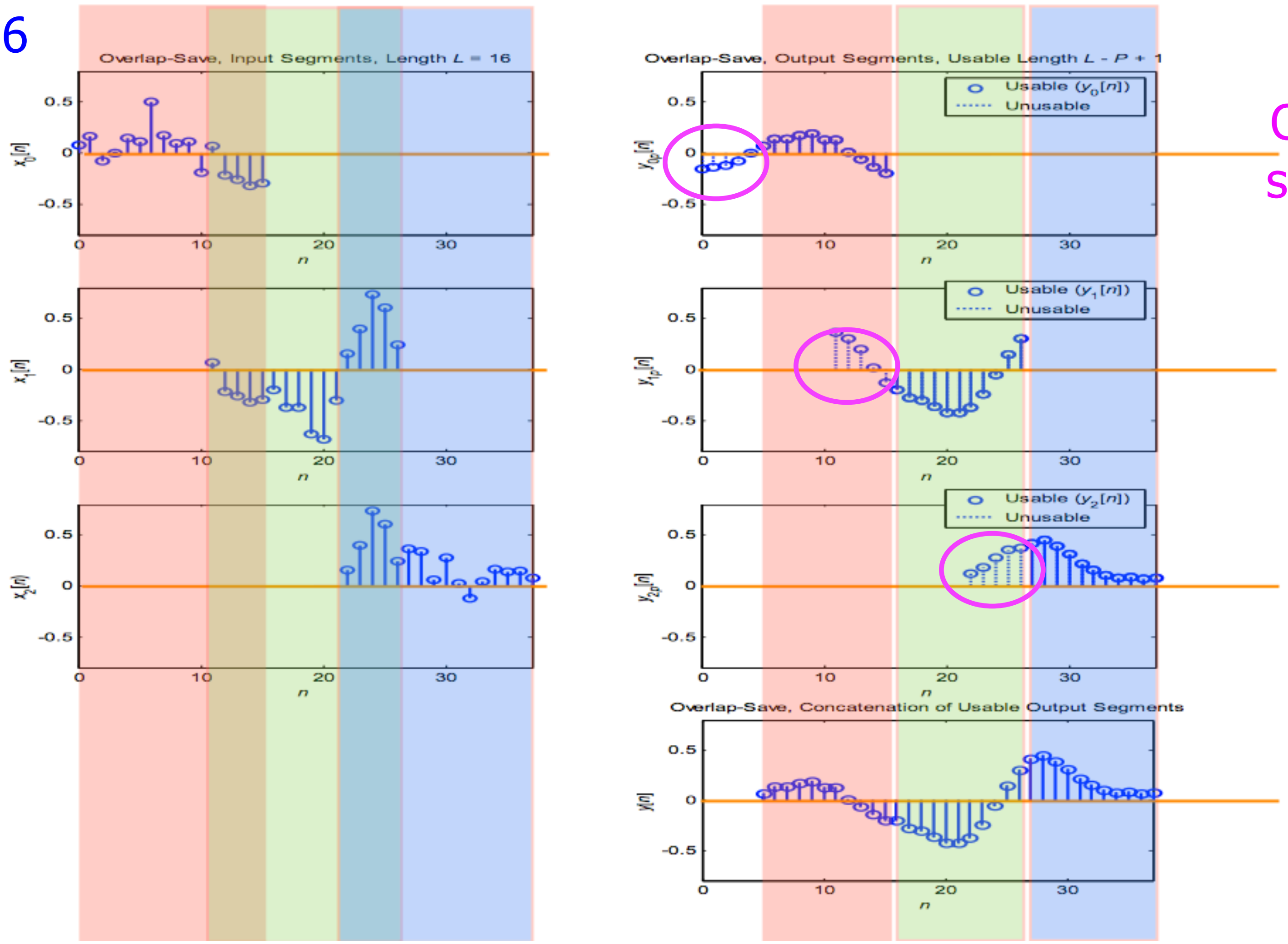
L+P-1=16

L=11



Example of Overlap-Save

L+P-1=16



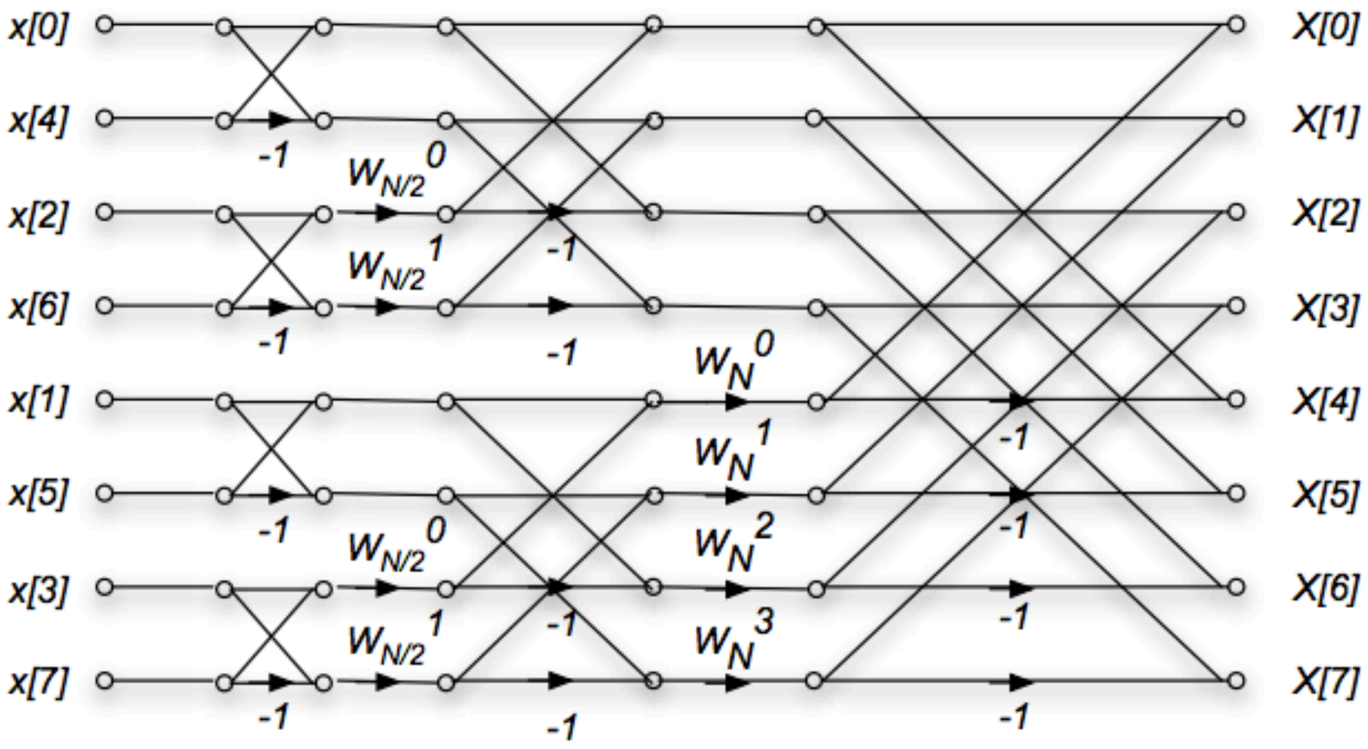
P-1=5
Overlap
samples

Adapted from M. Lustig, EECS Berkeley

FFT

Decimation-in-Time FFT

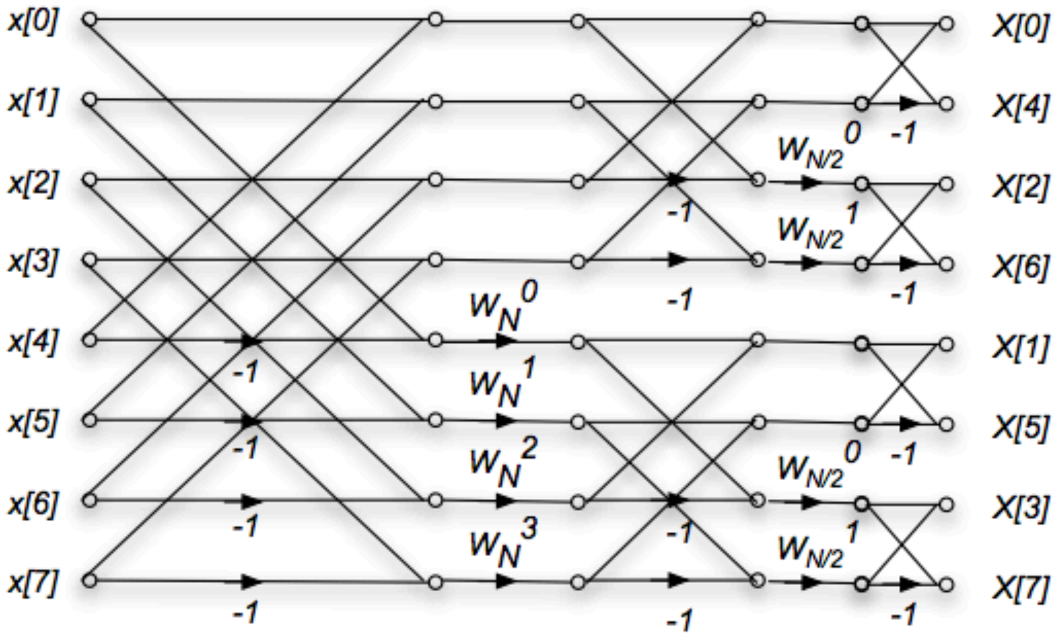
Combining all these stages, the diagram for the 8 sample DFT is:



- $3=\log_2(N)=\log_2(8)$ stages
- $4=N/2=8/2$ multiplications in each stage
  - 1st stage has trivial multiplication

Decimation-in-Frequency FFT

The diagram for and 8-point decimation-in-frequency DFT is as follows



This is just the decimation-in-time algorithm reversed!
The inputs are in normal order, and the outputs are bit reversed.

Big Ideas

- Fast Fourier Transform
  - Enable computation of an N-point DFT (or DFT^{-1}) with the order of just $N \cdot \log_2 N$ complex multiplications.
  - Most FFT algorithms decompose the computation of a DFT into successively smaller DFT computations.
    - Decimation-in-time algorithms
    - Decimation-in-frequency
  - Historically, power-of-2 DFTs had highest efficiency
  - Modern computing has led to non-power-of-2 FFTs with high efficiency
  - Sparsity leads to reduce computation on order $K \cdot \log N$

Circular Conv. as Linear Conv. w/ Aliasing

$$x_{3p}[n] = \begin{cases} \sum_{r=-\infty}^{\infty} x_3[n - rN], & 0 \le n \le N-1, \\ 0, & \text{otherwise}, \end{cases}$$

- Therefore

$$x_{3p}[n] = x_1[n] \text{Ⓝ} x_2[n]$$

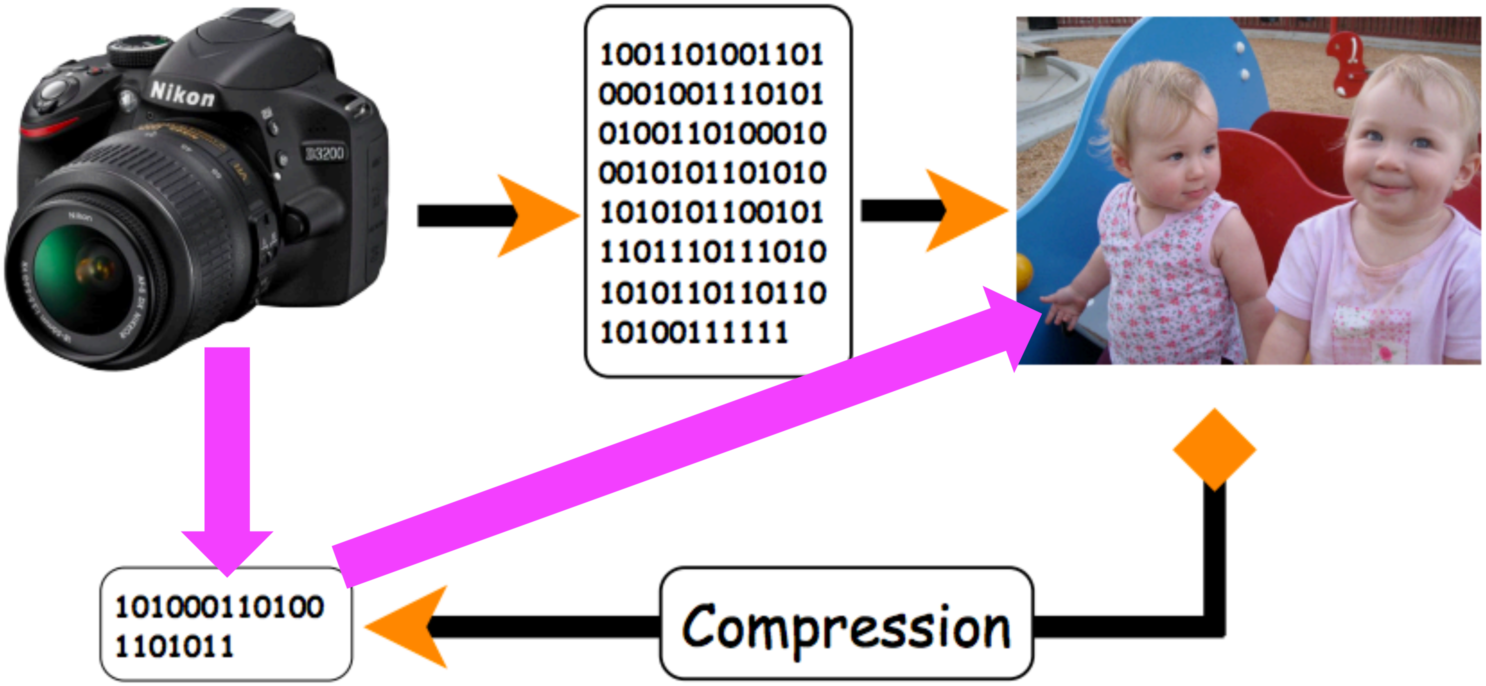
- The N-point circular convolution is the sum of linear convolutions shifted in time by N

Big Ideas

- Discrete Fourier Transform (DFT)
  - For finite signals assumed to be zero outside of defined length
  - N-point DFT is sampled DTFT at N points
  - Useful properties allow easier linear convolution
- Fast Convolution Methods
  - Use circular convolution (i.e DFT) to perform fast linear convolution
    - Overlap-Add, Overlap-Save
  - Circular convolution is linear convolution with aliasing
- Fast Fourier Transform
  - Enable computation of an N-point DFT (or DFT^{-1}) with the order of just $N \cdot \log_2 N$ complex multiplications.
- Design DSP methods to minimize computations!

Compressive Sensing/Sampling

- Standard approach
  - First collect, then compress
    - Throw away unnecessary data



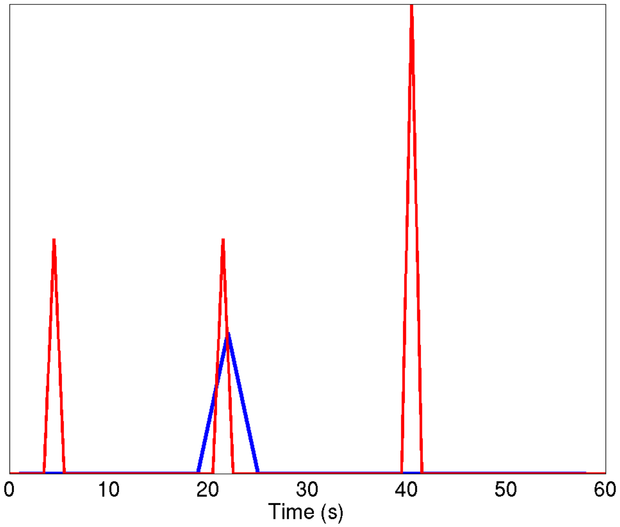
1001101001101
0001001110101
0100110100010
0010101101010
1010101100101
1101110111010
1010110110110
10100111111

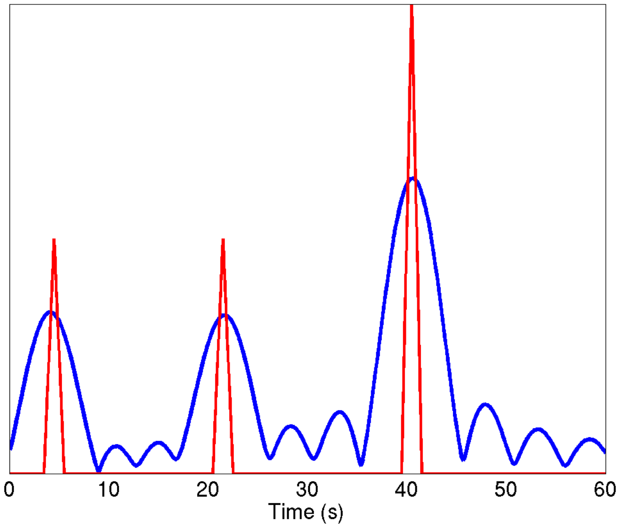
Compression

101000110100
1101011

Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in some cases exactly recover


Undersampled in time


Undersampled in frequency
(reconstructed in time with IFFT)

Requires sparsity and incoherent sampling

Final Exam

- Final – 5/3
  - Location TBD
  - Starts at exactly 3:00pm, ends at exactly 5:00pm (120 minutes)
  - Closed book
  - Cumulative – covers entire course
    - Except data converters, noise shaping, compressive sampling
  - Data/Equation sheet provided by me
    - Similar to midterm sheet and old final sheet
  - 8.5x11 cheat sheet allowed
  - Review session by Shlesh on 5/2, time TBD
  - Old exams posted
  - Calculators allowed, no smart phones
- Keep an eye on Piazza for office hour additions