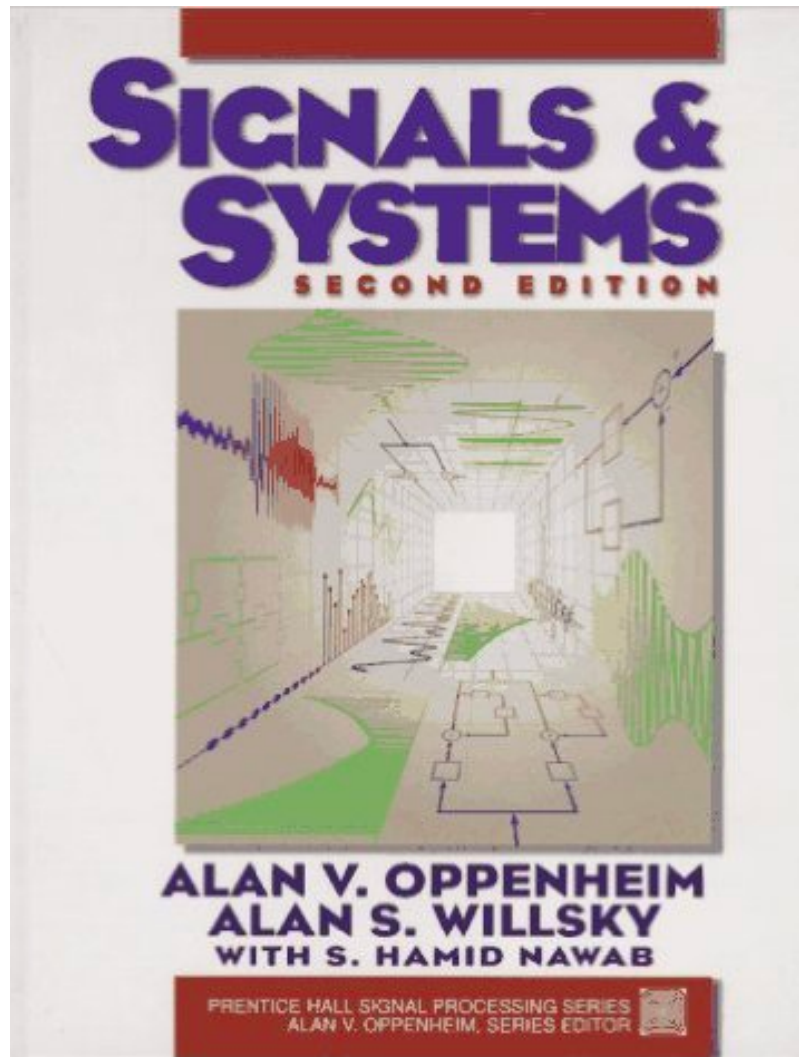


ESE 531: Digital Signal Processing

Lec 10: February 15th, 2018

Practical and Non-integer Sampling, Multi-
rate Sampling

Signals and Systems Review



MIT **OPEN**COURSEWARE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Subscribe to the OCW Newsletter

Home » Courses » Electrical Engineering and Computer Science » Signals and Systems

Signals and Systems

COURSE HOME <

SYLLABUS

CALENDAR

READINGS

LECTURE NOTES

LECTURE VIDEOS

ASSIGNMENTS

EXAMS

DOWNLOAD COURSE MATERIALS

→ **6.003** →

Signals and Systems

modeling — analysis — design

Instructor(s)
Prof. Dennis Freeman

MIT Course Number
6.003

As Taught In
Fall 2011

Level
Undergraduate

[CITE THIS COURSE](#)

Course Features

- > [Video lectures](#)
- > [Online textbooks](#)
- > [Assignments: problem sets with solutions](#)
- > [Subtitles/transcript](#)
- > [Lecture notes](#)
- > [Exams and solutions](#)

Course Description

6.003 covers the fundamentals of signal and system analysis, focusing on representations of discrete-time and continuous-time signals (singularity functions, complex exponentials and geometrics, Fourier representations, Laplace and Z transforms, sampling) and representations of linear, time-invariant systems (difference and differential equations, block diagrams, system functions, poles and zeros, convolution, impulse and step responses, frequency responses). Applications are drawn broadly from engineering and physics, including feedback and control, communications, and signal processing.

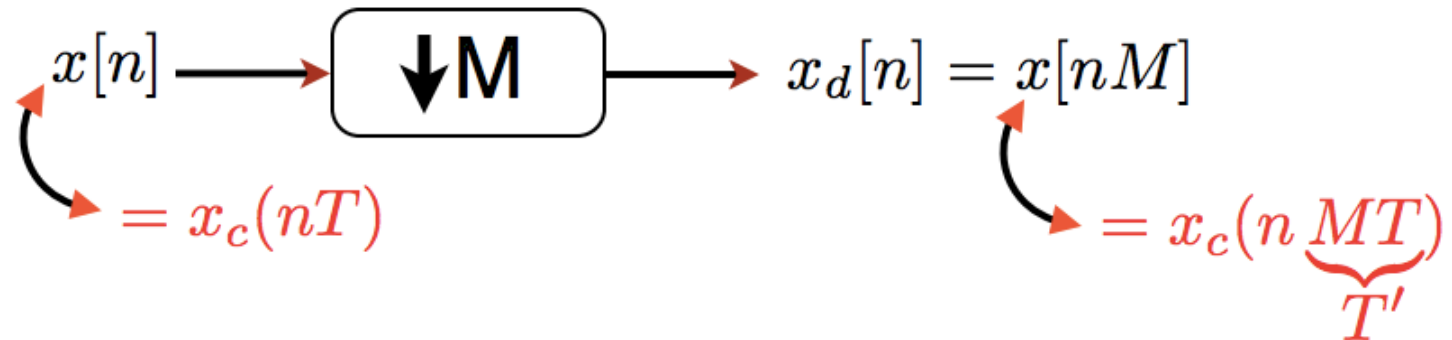


Lecture Outline

- ❑ Review: Downsampling/Upsampling
- ❑ Non-integer Resampling
- ❑ Multi-Rate Processing
 - Interchanging Operations
- ❑ Polyphase Decomposition
- ❑ Multi-Rate Filter Banks

Downsampling

- Definition: Reducing the sampling rate by an integer number

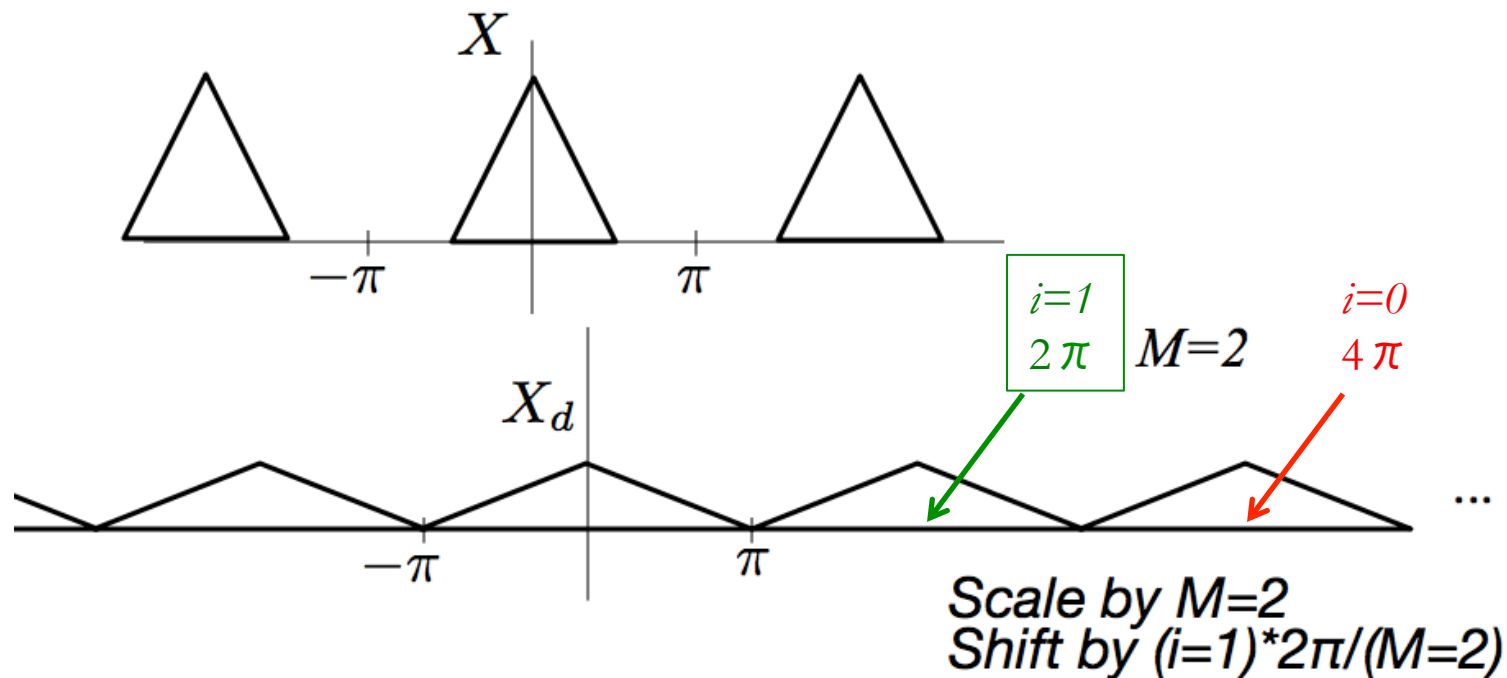


$$X_d(e^{j\omega}) = \frac{1}{M} \sum_{i=0}^{M-1} X(e^{j(\frac{\omega}{M} - \frac{2\pi}{M}i)})$$

stretch by M replicate

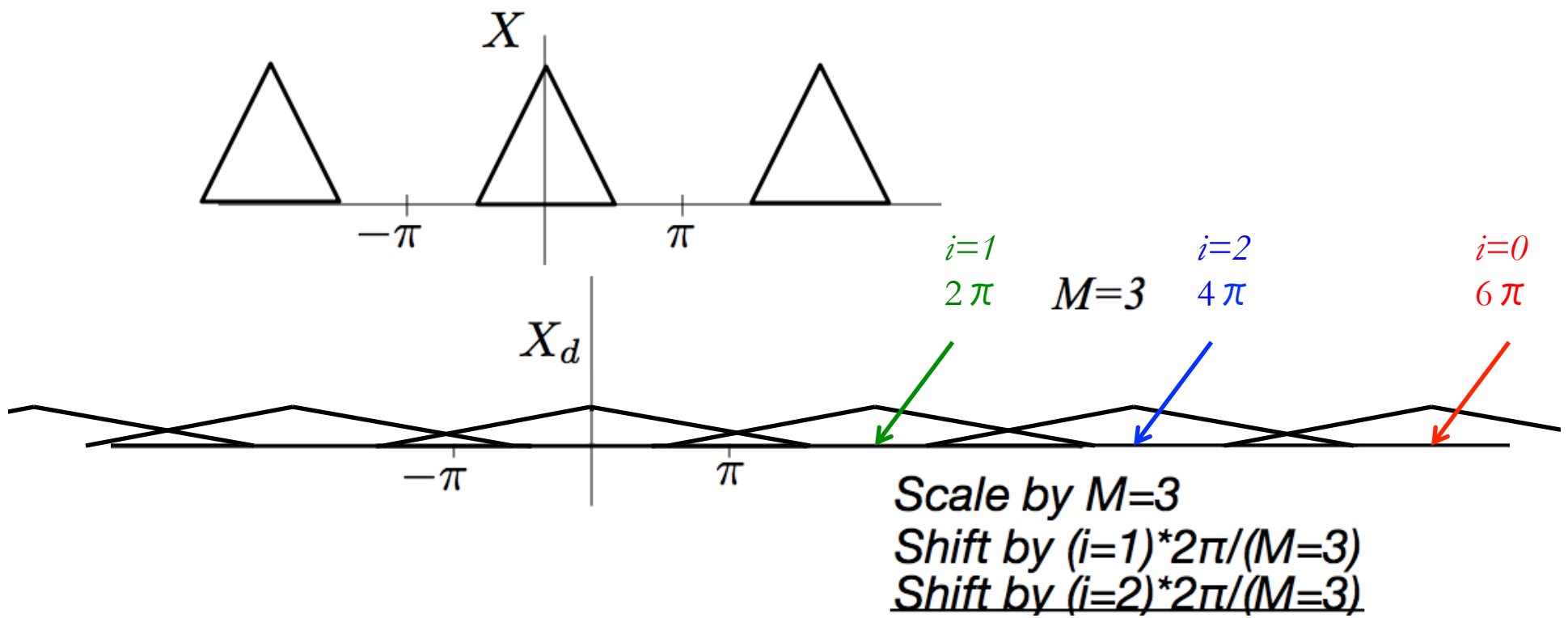
Example

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

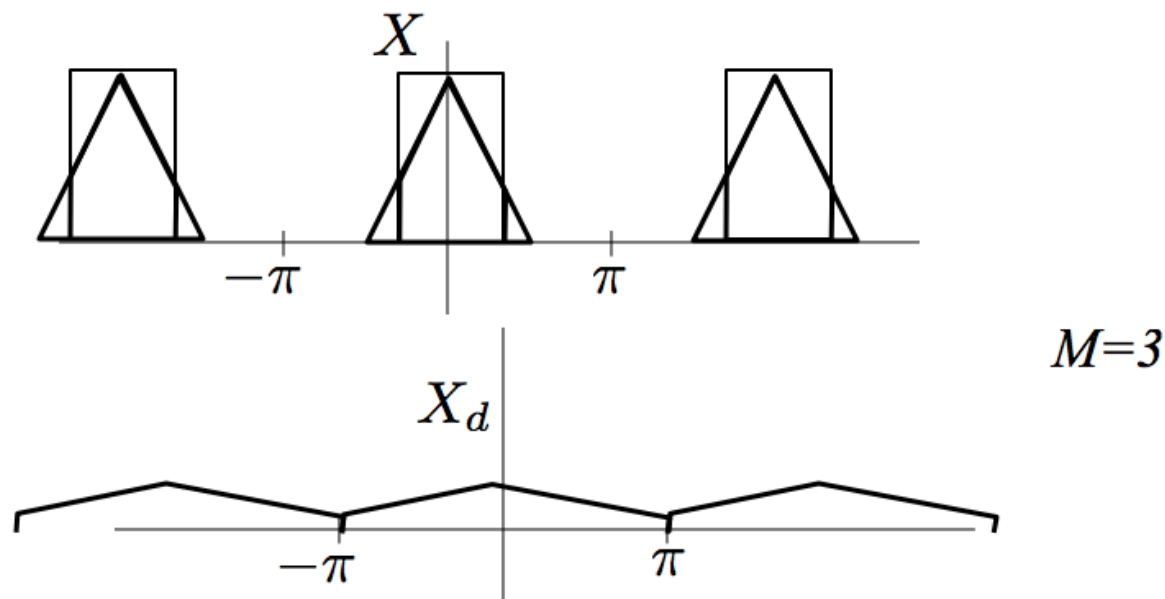
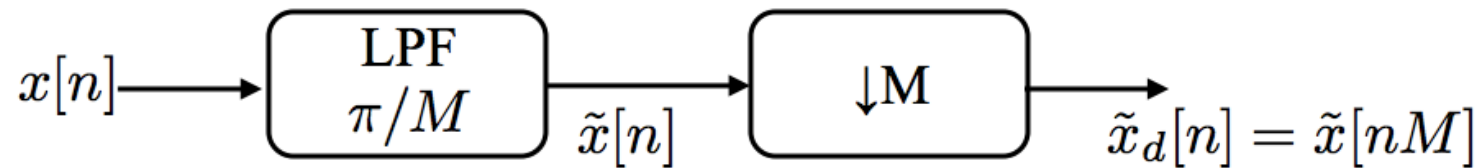


Example

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



Example





Upsampling

- Definition: Increasing the sampling rate by an integer number

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

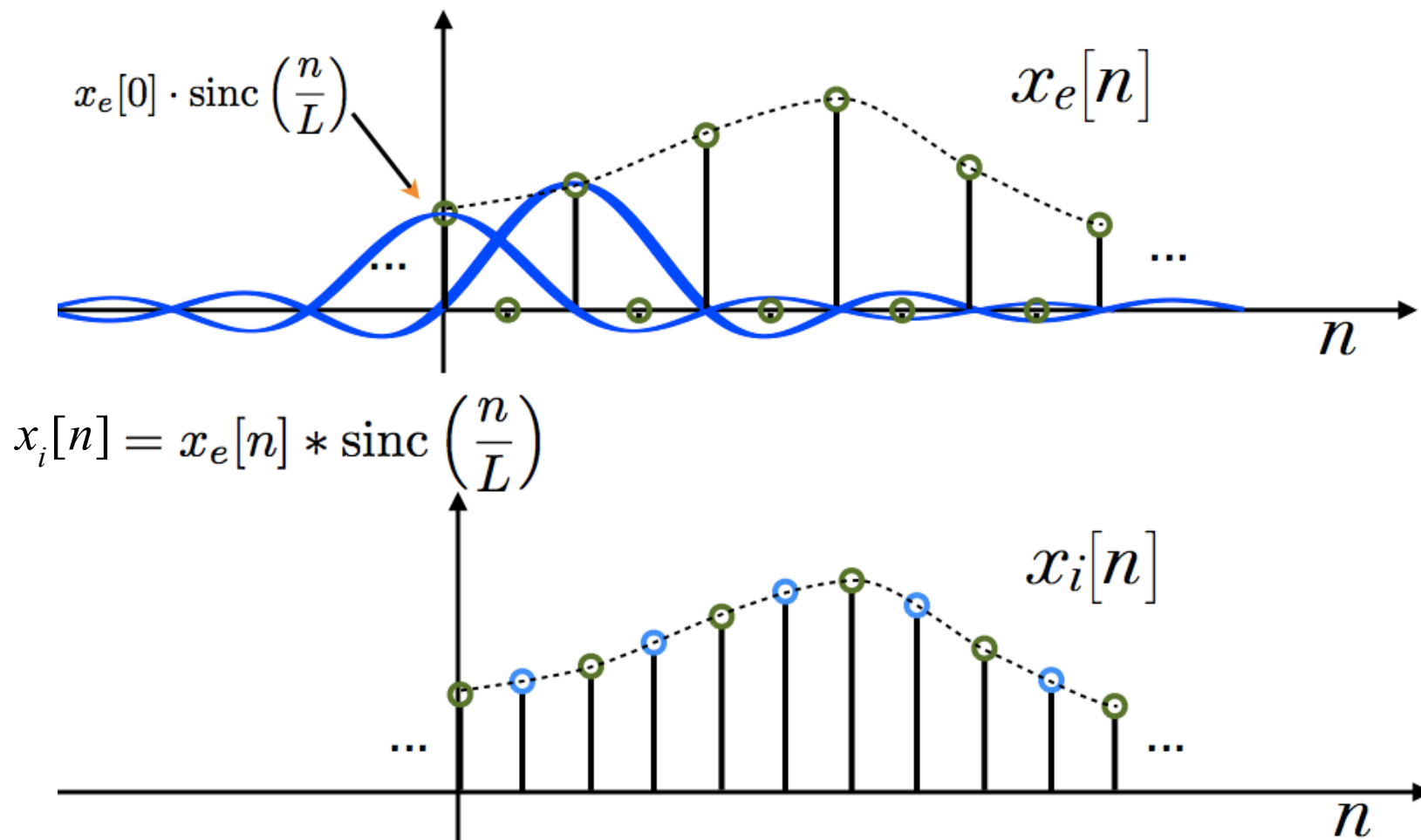
$$x_i[n] = x_c(nT') \quad \text{where } T' = \frac{T}{L} \quad L \text{ integer}$$

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

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

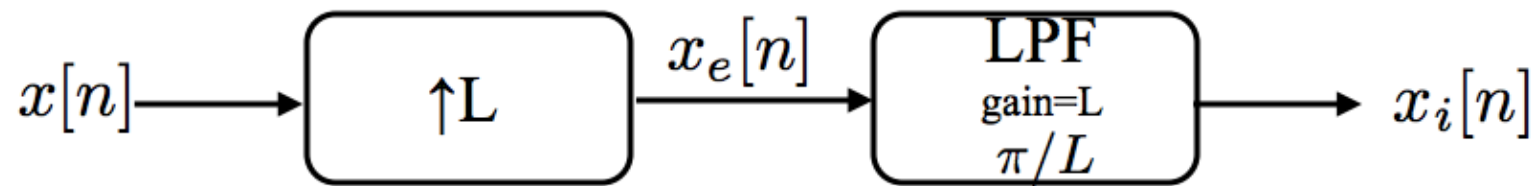
Upsampling

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



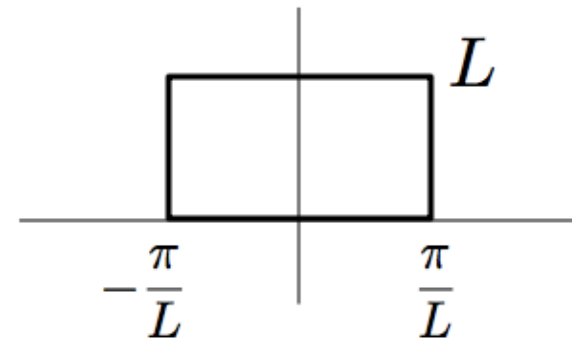
Frequency Domain Interpretation

$$x_i[n] = x_e[n] * \text{sinc}(n/L)$$

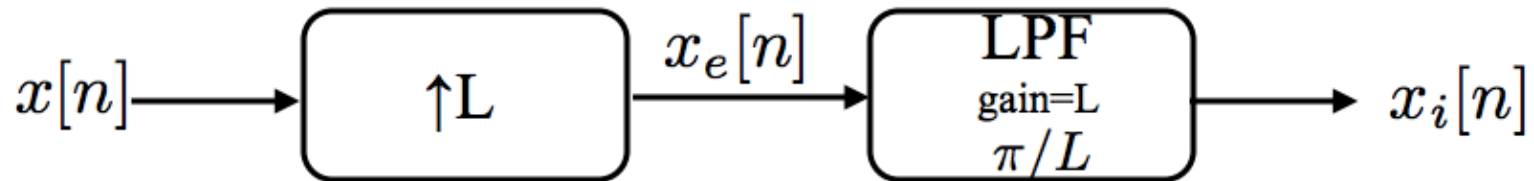


$\text{sinc}(n/L)$

DTFT $\Rightarrow$



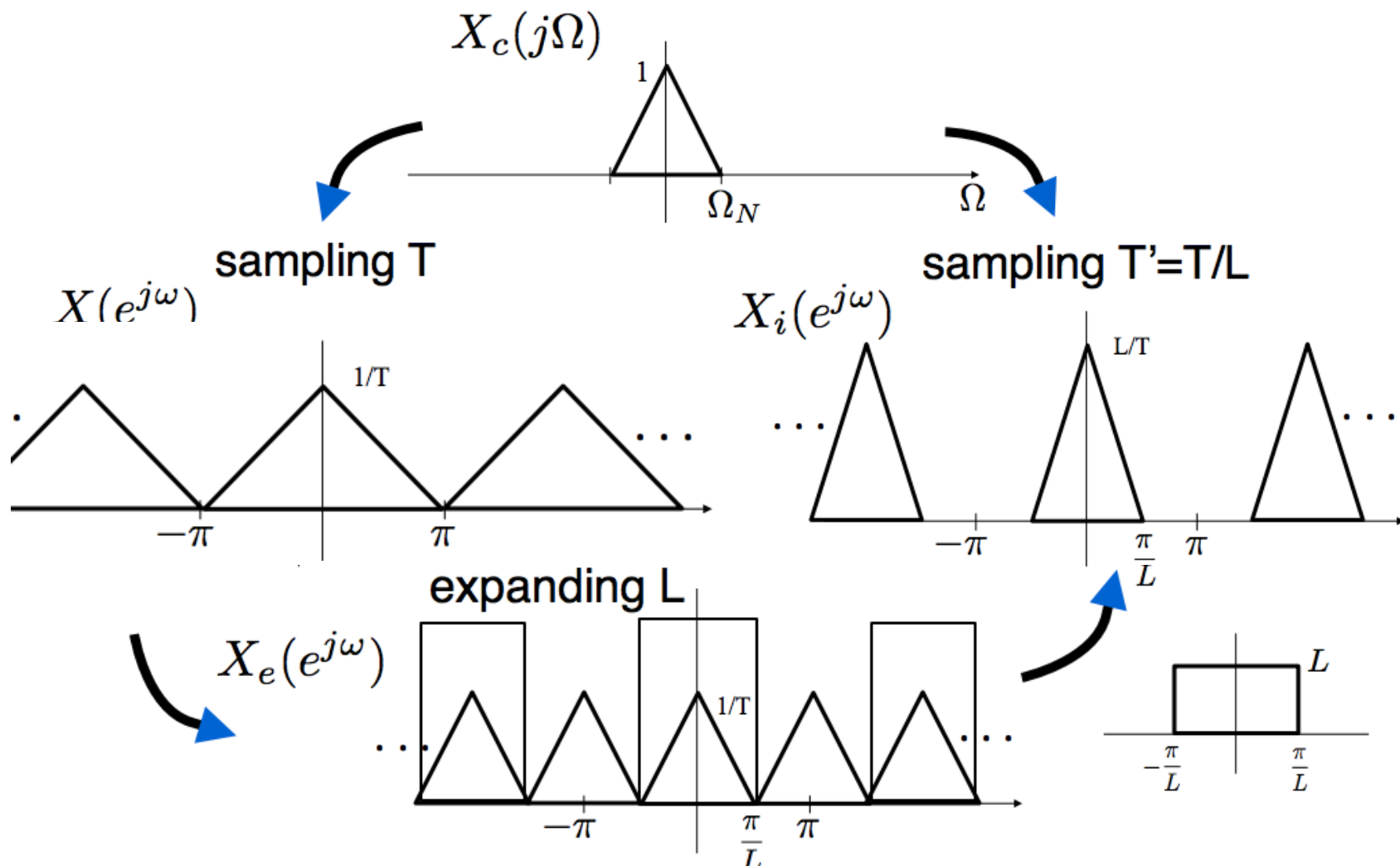
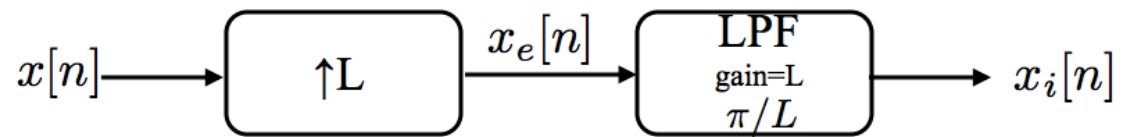
Frequency Domain Interpretation



$$\begin{aligned} X_e(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} \underbrace{x_e[n]}_{\substack{\neq 0 \text{ only for } n=mL \\ (\text{integer } m)}} e^{-j\omega n} \\ &= \sum_{m=-\infty}^{\infty} \underbrace{x_e[mL]}_{=x[m]} e^{-j\omega mL} = X(e^{j\omega L}) \end{aligned}$$

Compress DTFT by a factor of L!

Example



Non-integer Resampling





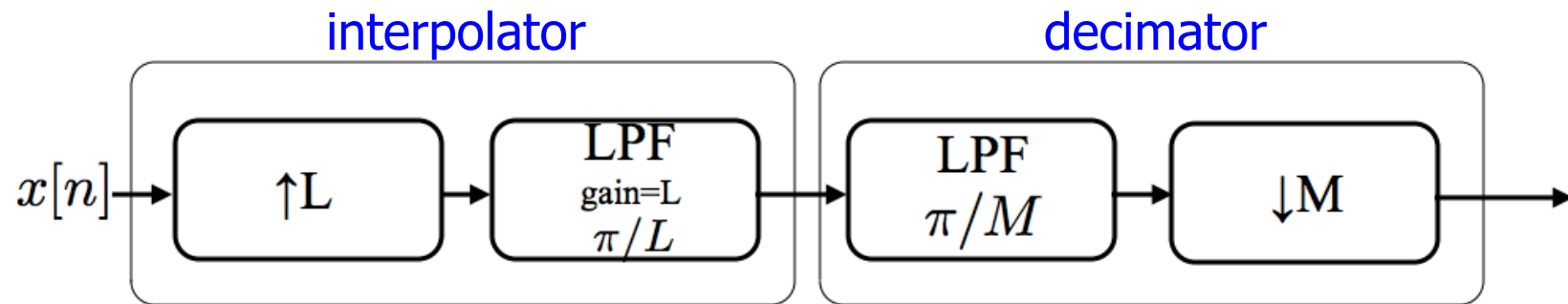
Non-integer Resampling

□ $T' = TM/L$

Non-integer Resampling

□ $T' = TM/L$

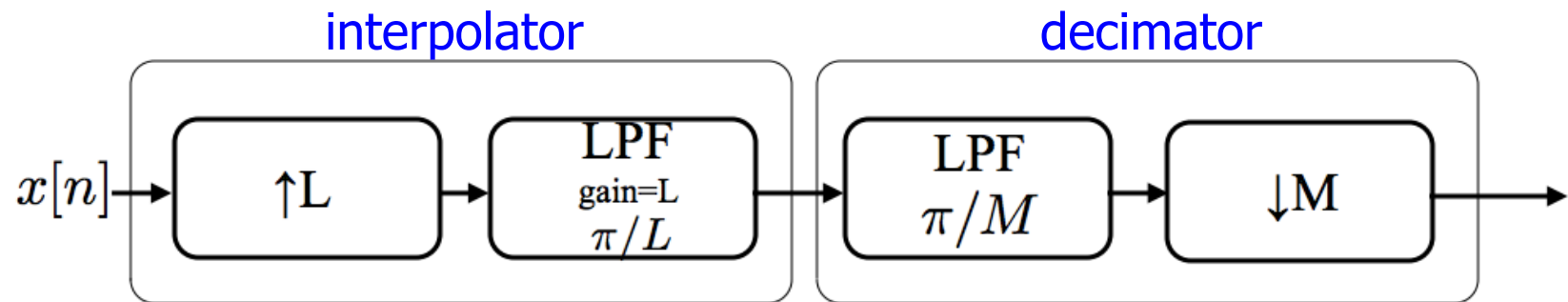
- Upsample by L , then downsample by M



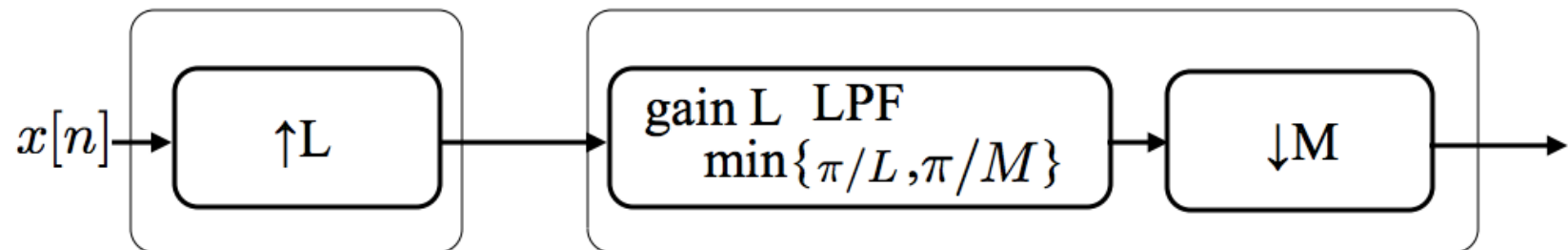
Non-integer Resampling

□ $T' = TM/L$

- Upsample by L , then downsample by M

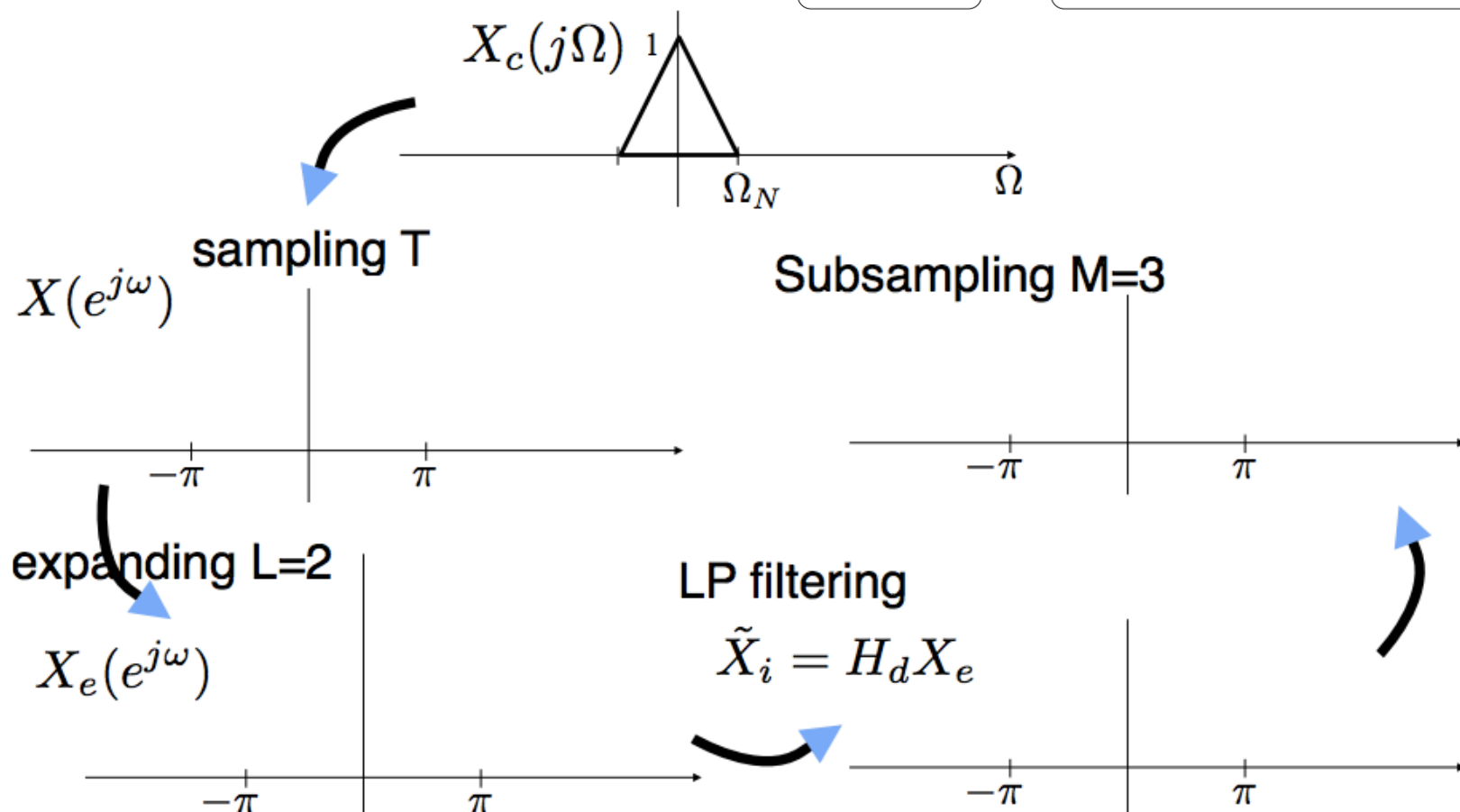
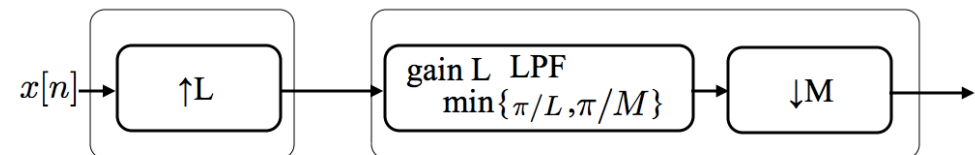


Or,



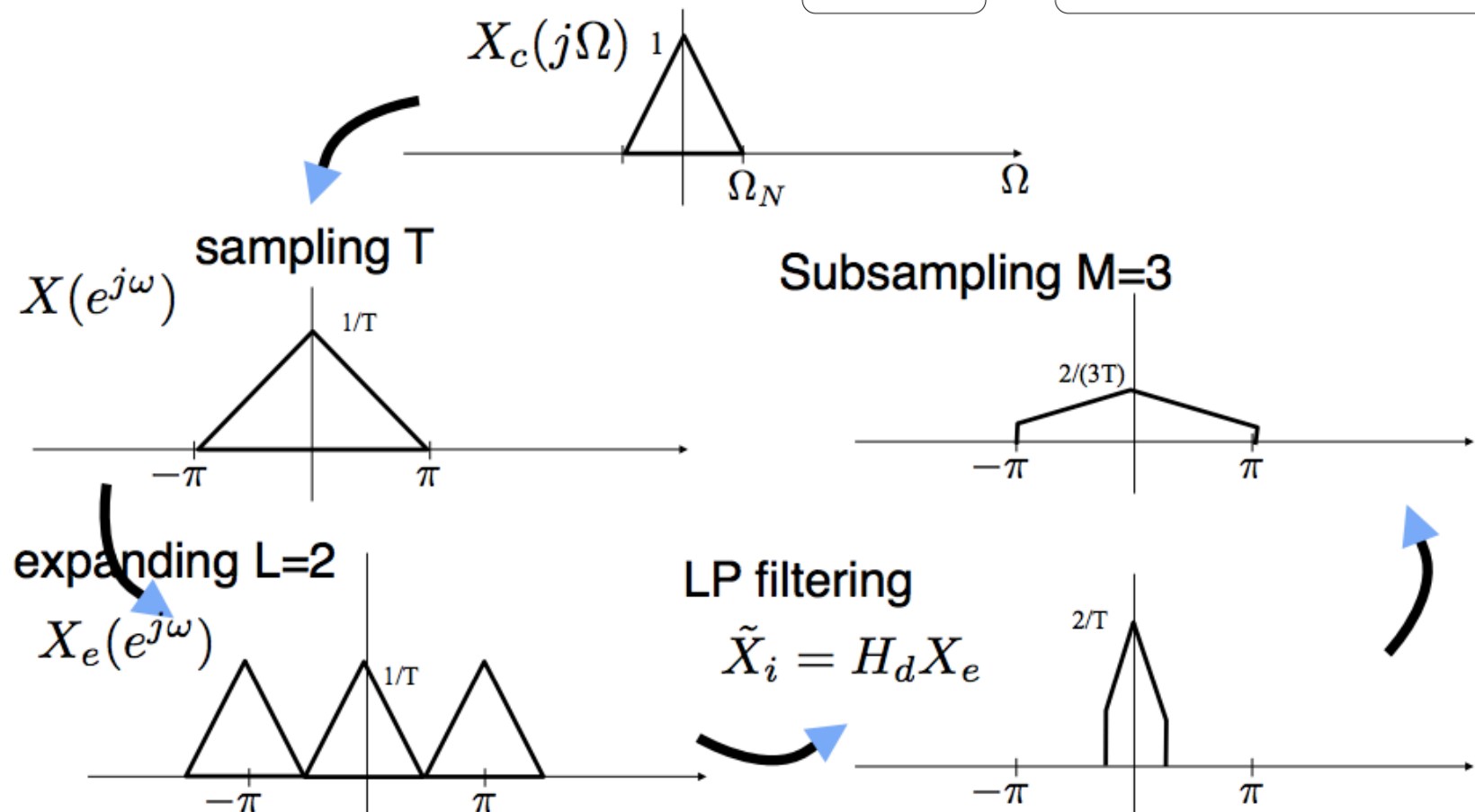
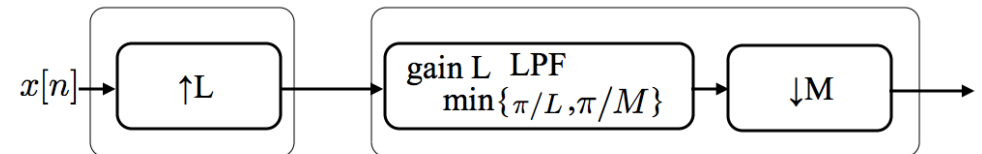
Example

□ $T' = 3/2T \rightarrow L=2, M=3$



Example

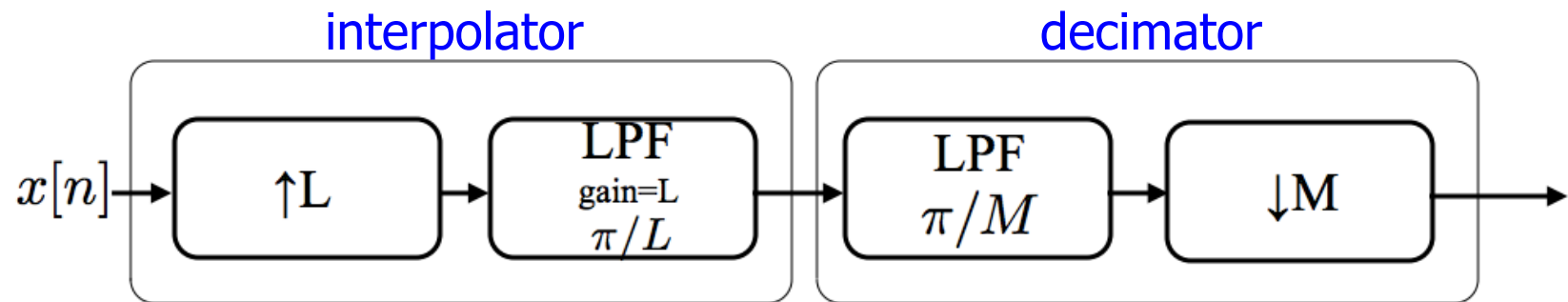
□ $T' = 3/2T \rightarrow L=2, M=3$



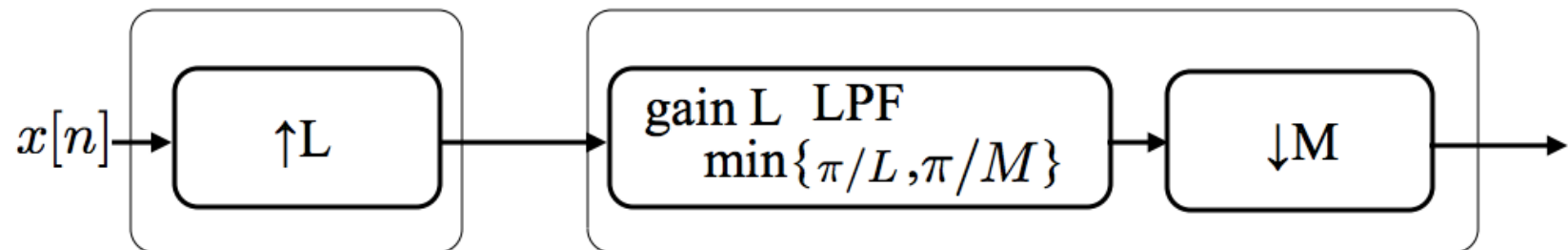
Non-integer Sampling

□ $T' = TM/L$

- Downsample by M , then upsample by L ?



Or,



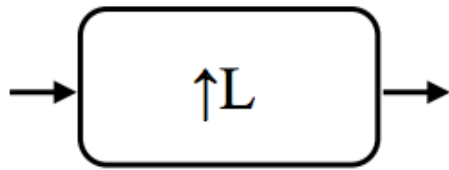


Multi-Rate Signal Processing

- ❑ What if we want to resample by 1.01T?
 - Expand by $L=100$
 - Filter $\pi / 101$ (\$\$\$\$\$)
 - Downsample by $M=101$

- ❑ Fortunately there are ways around it!
 - Called multi-rate
 - Uses compressors, expanders and filtering

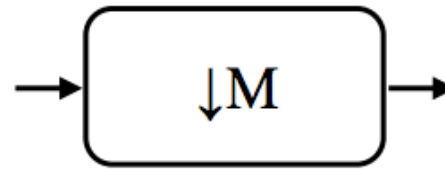
Interchanging Operations



“expander”

Upsampling

- expanding in time
- compressing in frequency



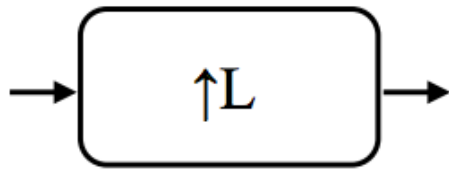
“compressor”

Downsampling

- compressing in time
- expanding in frequency

not LTI!

Interchanging Operations - Expander



“expander”

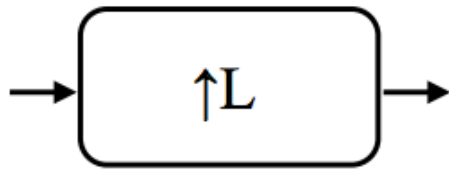
Upsampling

-expanding in time

-compressing in frequency



Interchanging Operations - Expander

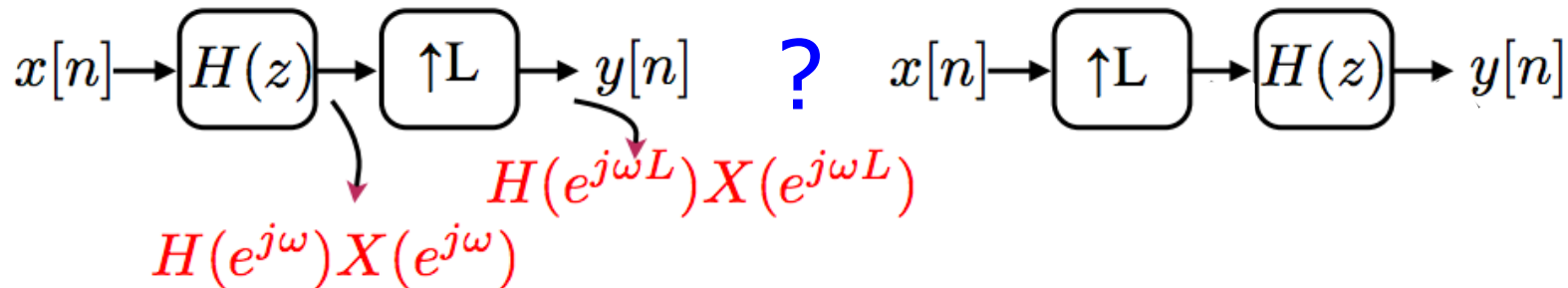


“expander”

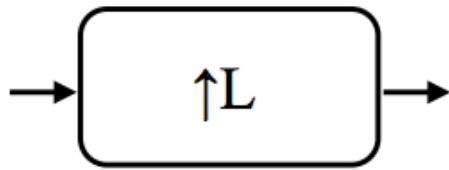
Upsampling

-expanding in time

-compressing in frequency



Interchanging Operations - Expander

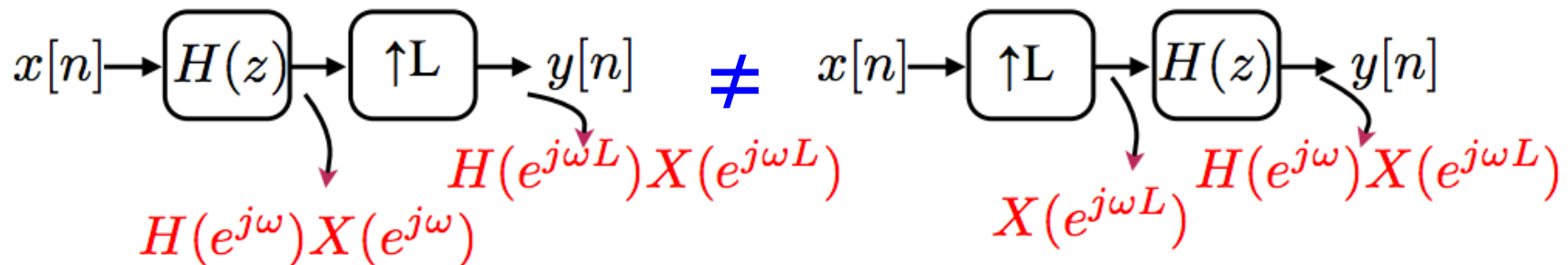


“expander”

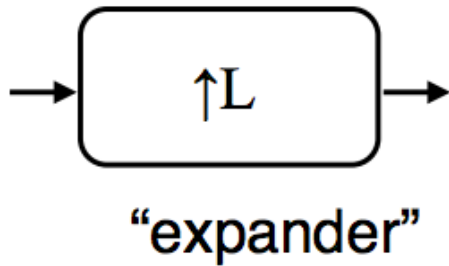
Upsampling

-expanding in time

-compressing in frequency

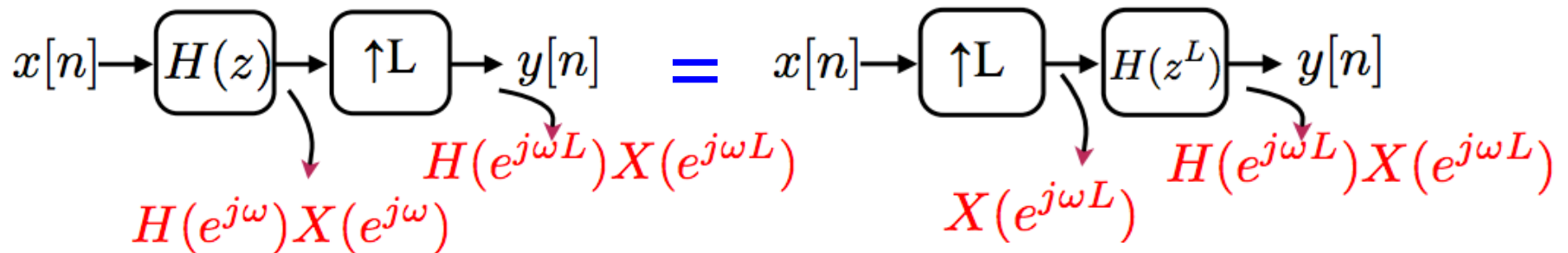


Interchanging Operations - Expander

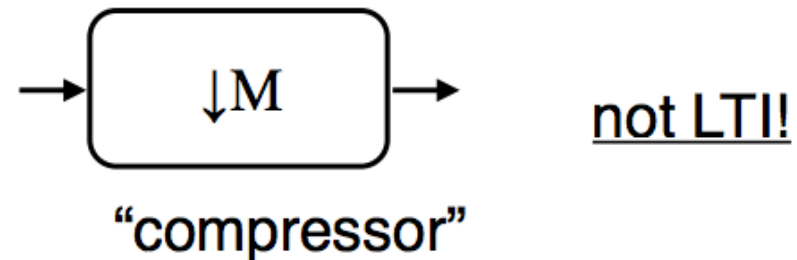


Upsampling

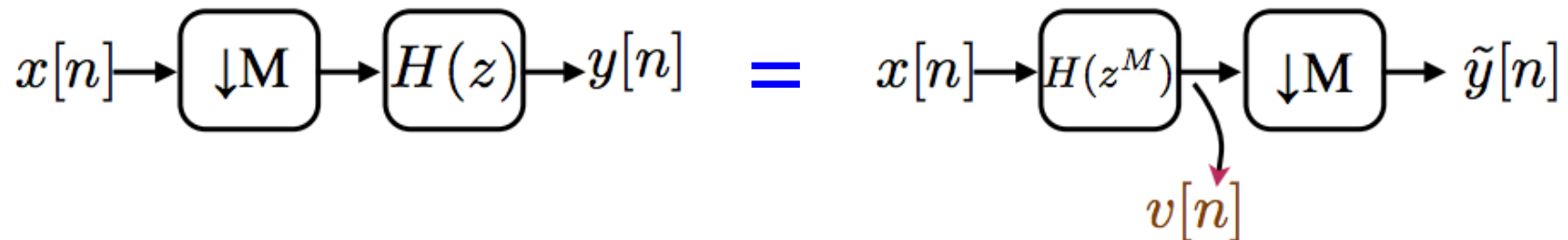
- expanding in time
- compressing in frequency



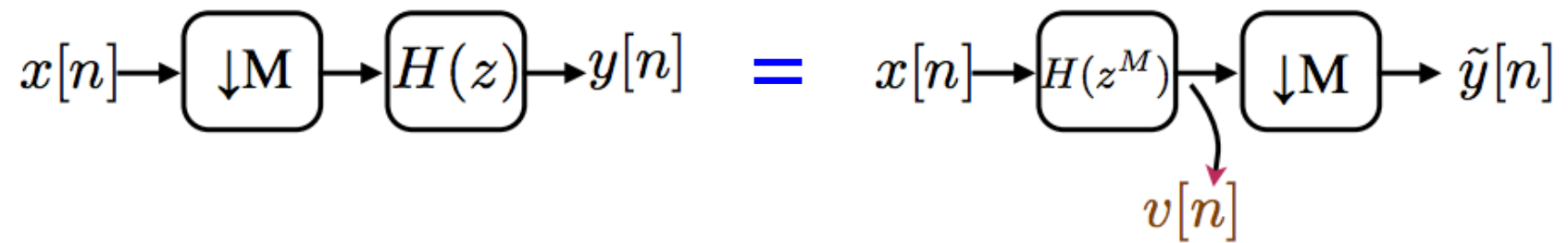
Interchanging Operations - Compressor



Downsampling
-compressing in time
-expanding in frequency

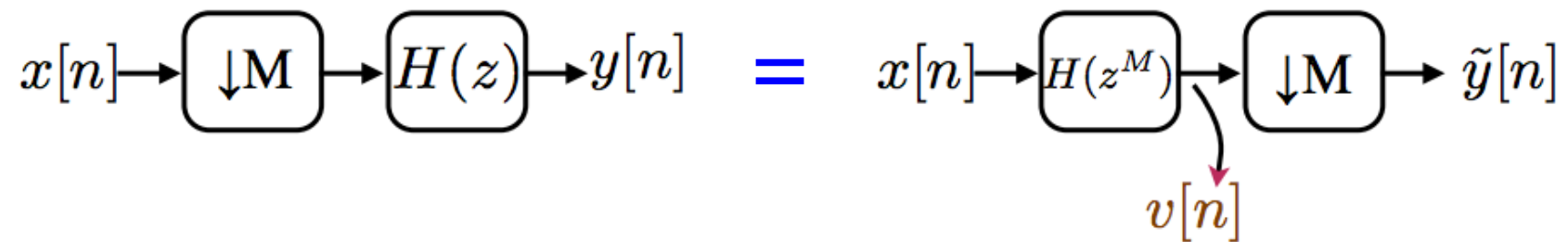


Interchanging Operations - Compressor



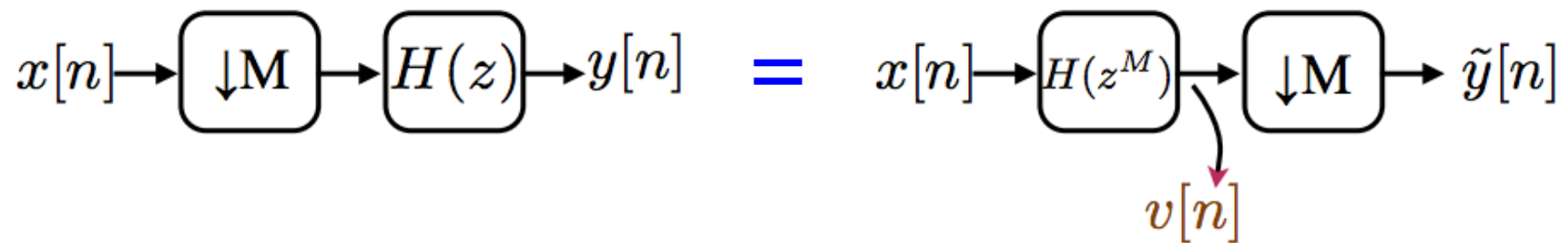
$$Y(e^{j\omega}) = H(e^{j\omega}) \left(\frac{1}{M} \sum_{i=0}^{M-1} X \left(e^{j(\frac{\omega}{M} - \frac{2\pi i}{M})} \right) \right)$$

Interchanging Operations - Compressor



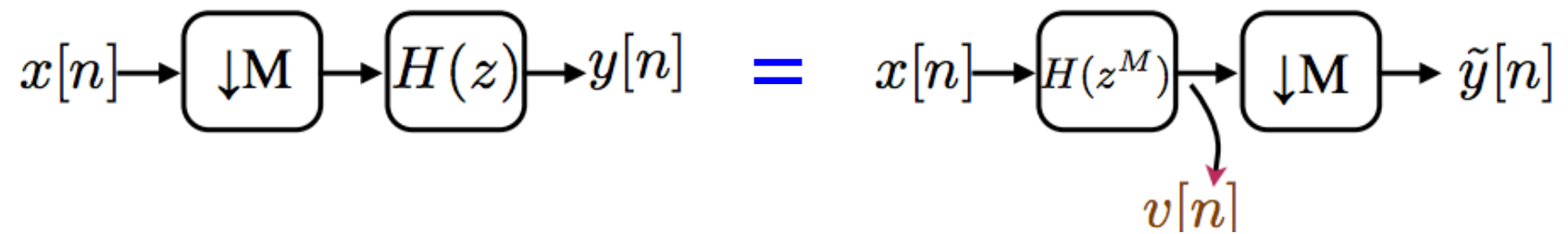
$$\begin{aligned}
 Y(e^{j\omega}) &= H(e^{j\omega}) \left(\frac{1}{M} \sum_{i=0}^{M-1} X \left(e^{j(\frac{\omega}{M} - \frac{2\pi i}{M})} \right) \right) \\
 &= \frac{1}{M} \sum_{i=0}^{M-1} \underbrace{H \left(e^{j(\omega - 2\pi i)} \right)}_{H(e^{j\omega})} X \left(e^{j(\frac{\omega}{M} - \frac{2\pi i}{M})} \right)
 \end{aligned}$$

Interchanging Operations - Compressor



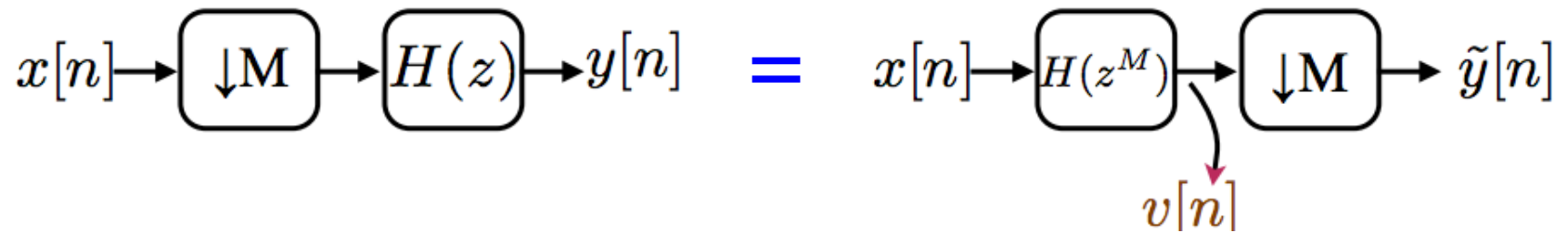
$$\begin{aligned}
 Y(e^{j\omega}) &= H(e^{j\omega}) \left(\frac{1}{M} \sum_{i=0}^{M-1} X \left(e^{j(\frac{\omega}{M} - \frac{2\pi i}{M})} \right) \right) \\
 &= \frac{1}{M} \sum_{i=0}^{M-1} \underbrace{H \left(e^{j(\omega - 2\pi i)} \right)}_{H(e^{j\omega})} X \left(e^{j(\frac{\omega}{M} - \frac{2\pi i}{M})} \right) \\
 &= \frac{1}{M} \sum_{i=0}^{M-1} H \left(e^{jM(\frac{\omega}{M} - \frac{2\pi i}{M})} \right) X \left(e^{j(\frac{\omega}{M} - \frac{2\pi i}{M})} \right)
 \end{aligned}$$

Interchanging Operations - Compressor



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

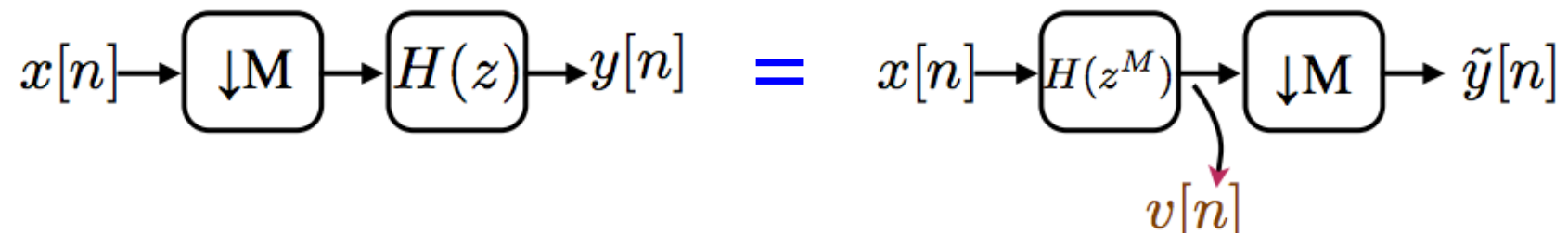
Interchanging Operations - Compressor



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

$$\tilde{Y}(e^{j\omega}) = \frac{1}{M} \sum_{i=0}^{M-1} V\left(e^{j\left(\frac{\omega}{M} - \frac{2\pi i}{M}\right)}\right)$$

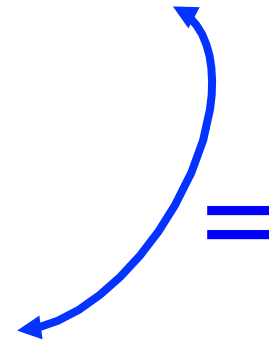
Interchanging Operations - Compressor



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

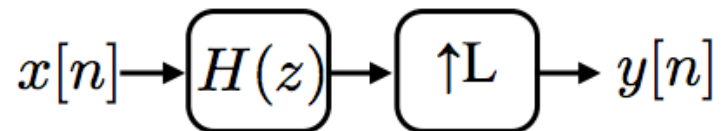
$$V(e^{j\omega}) = H(e^{j\omega M})X(e^{j\omega})$$

$$\tilde{Y}(e^{j\omega}) = \frac{1}{M} \sum_{i=0}^{M-1} V\left(e^{j\left(\frac{\omega}{M} - \frac{2\pi i}{M}\right)}\right)$$

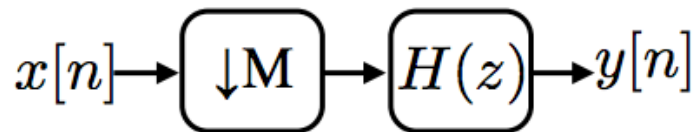
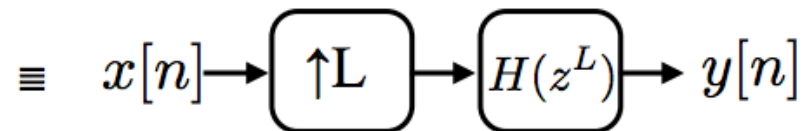


Interchanging Operations - Summary

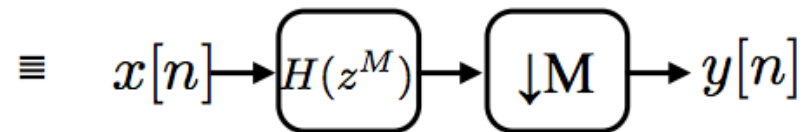
Filter and expander



Expander and expanded filter*



Compressor and filter



Expanded filter* and compressor

*Expanded filter = expanded impulse response, compressed freq response



Multi-Rate Signal Processing

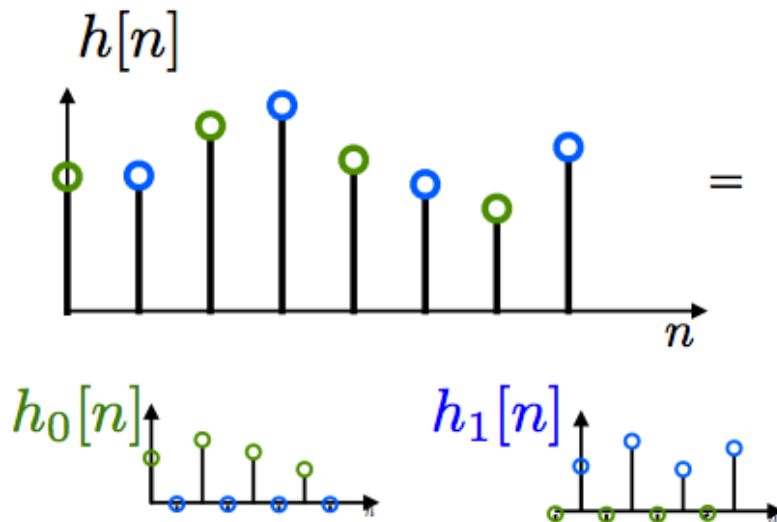
- ❑ What if we want to resample by 1.01T?
 - Expand by $L=100$
 - Filter $\pi/101$ (\$\$\$\$\$)
 - Downsample by $M=101$

- ❑ Fortunately there are ways around it!
 - Called multi-rate
 - Uses compressors, expanders and filtering

Polyphase Decomposition

- We can decompose an impulse response (of our filter) to:

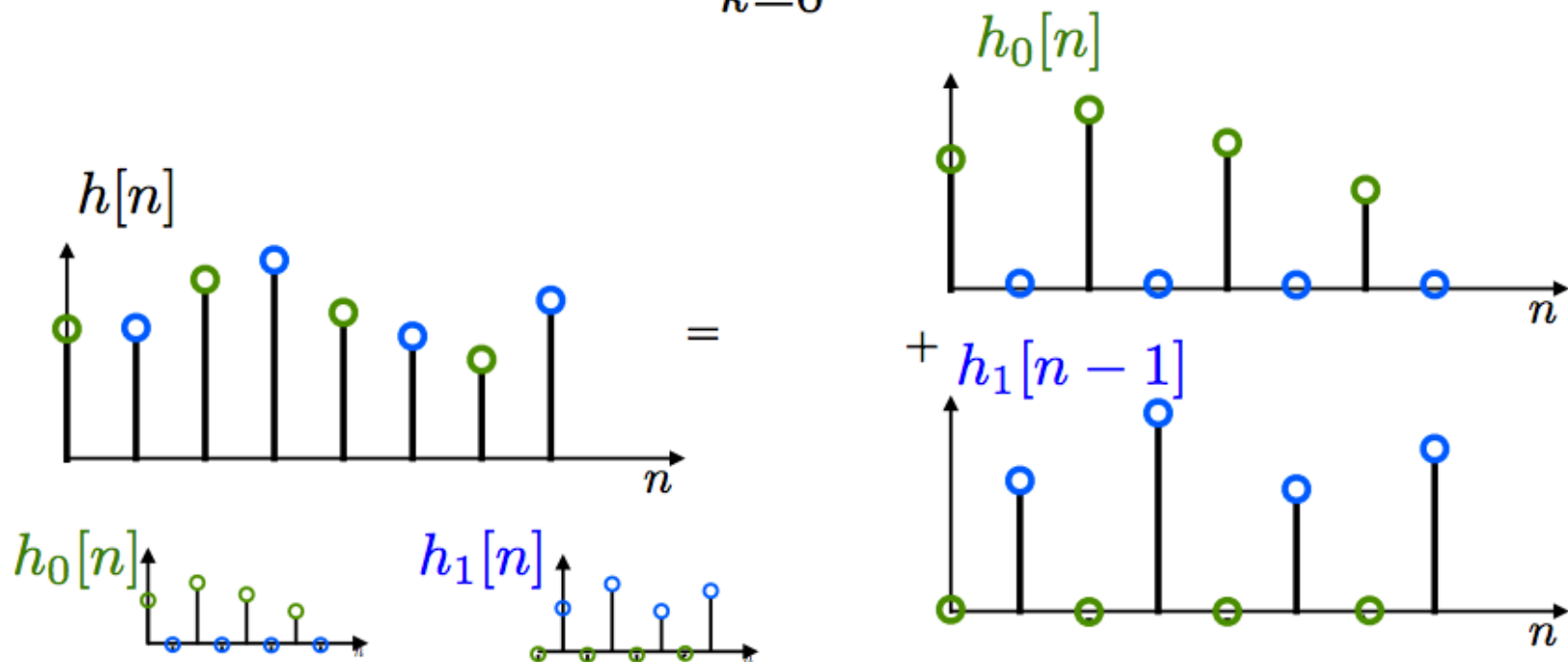
$$h[n] = \sum_{k=0}^{M-1} h_k[n - k]$$



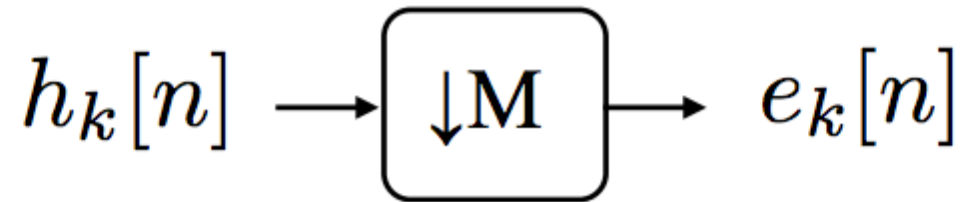
Polyphase Decomposition

- We can decompose an impulse response (of our filter) to:

$$h[n] = \sum_{k=0}^{M-1} h_k[n - k]$$



Polyphase Decomposition

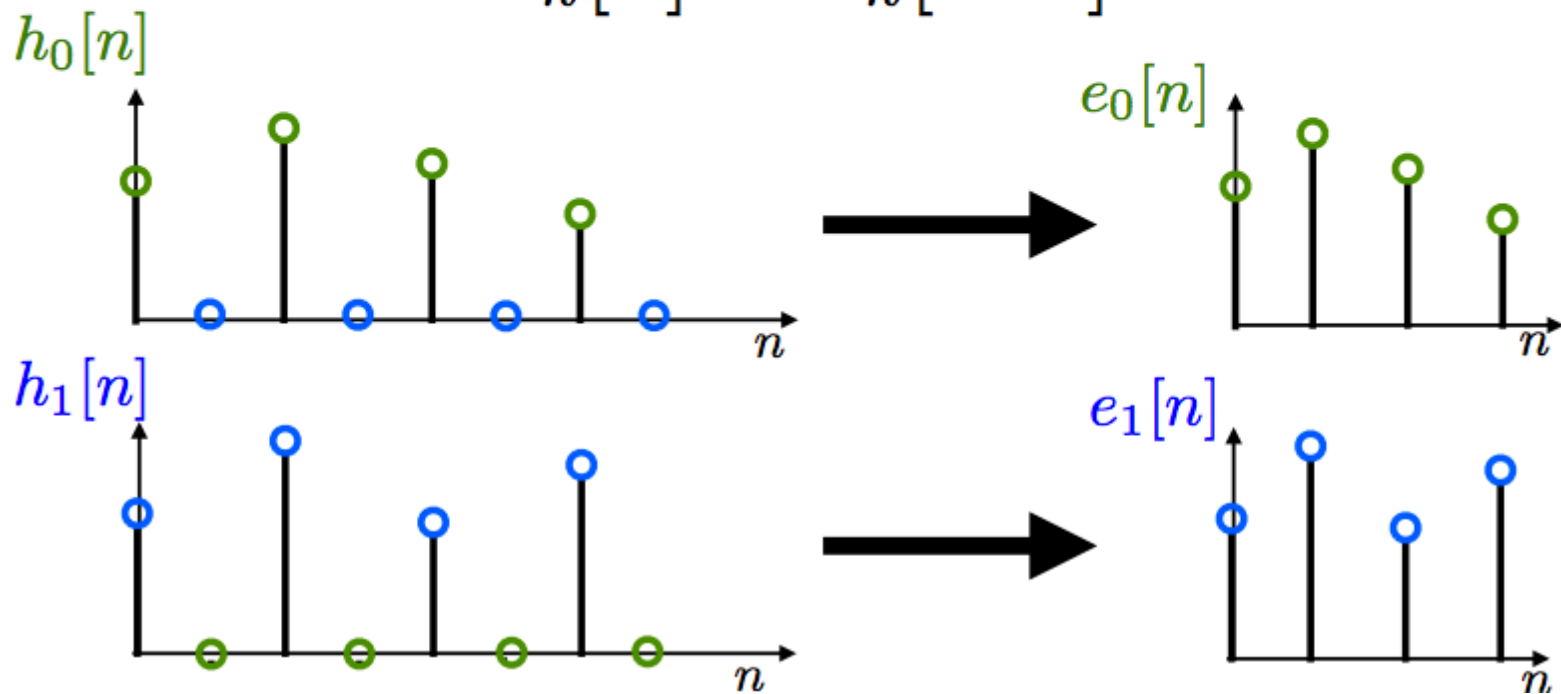


$$e_k[n] = h_k[nM]$$

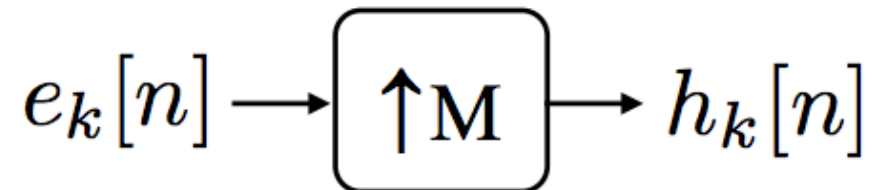
Polyphase Decomposition

$$h_k[n] \rightarrow \boxed{\downarrow M} \rightarrow e_k[n]$$

$$e_k[n] = h_k[nM]$$



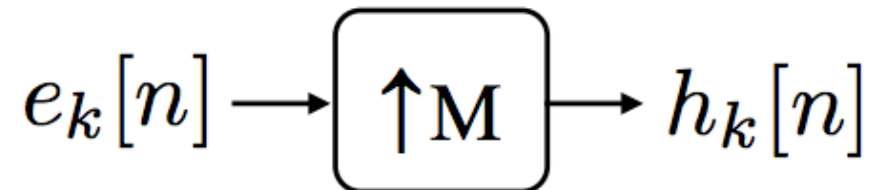
Polyphase Decomposition



recall upsampling $\Rightarrow$ scaling

$$H_k(z) = E_k(z^M)$$

Polyphase Decomposition



recall upsampling $\Rightarrow$ scaling

$$H_k(z) = E_k(z^M)$$

Also, recall:

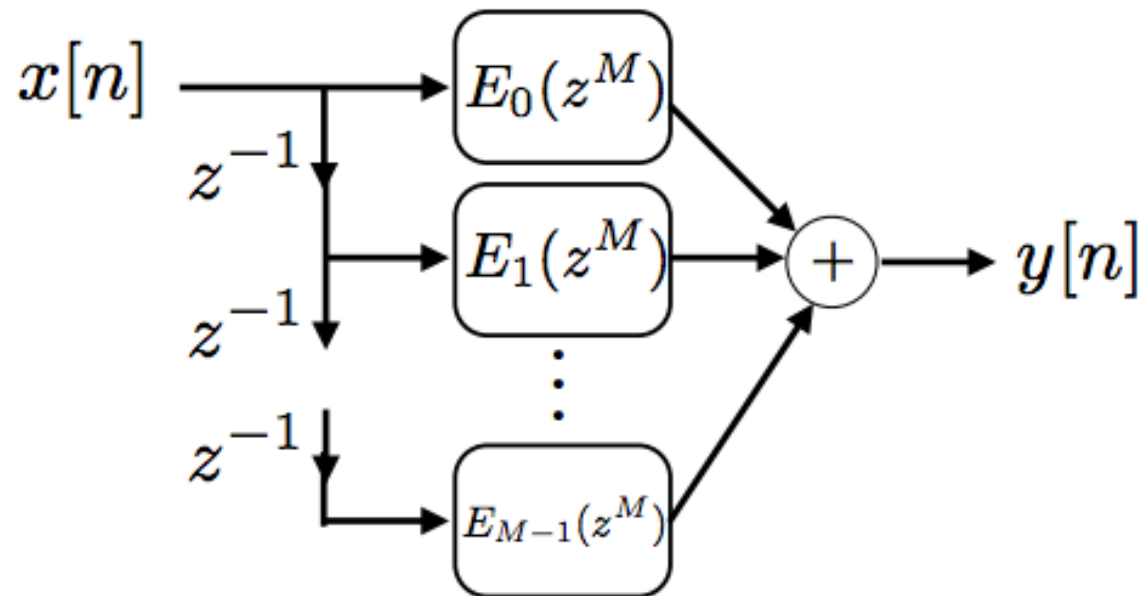
$$h[n] = \sum_{k=0}^{M-1} h_k[n - k]$$

So,

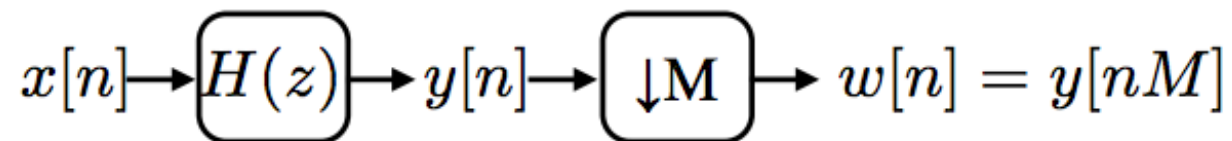
$$H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k}$$

Polyphase Decomposition

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



Polyphase Implementation of Decimation

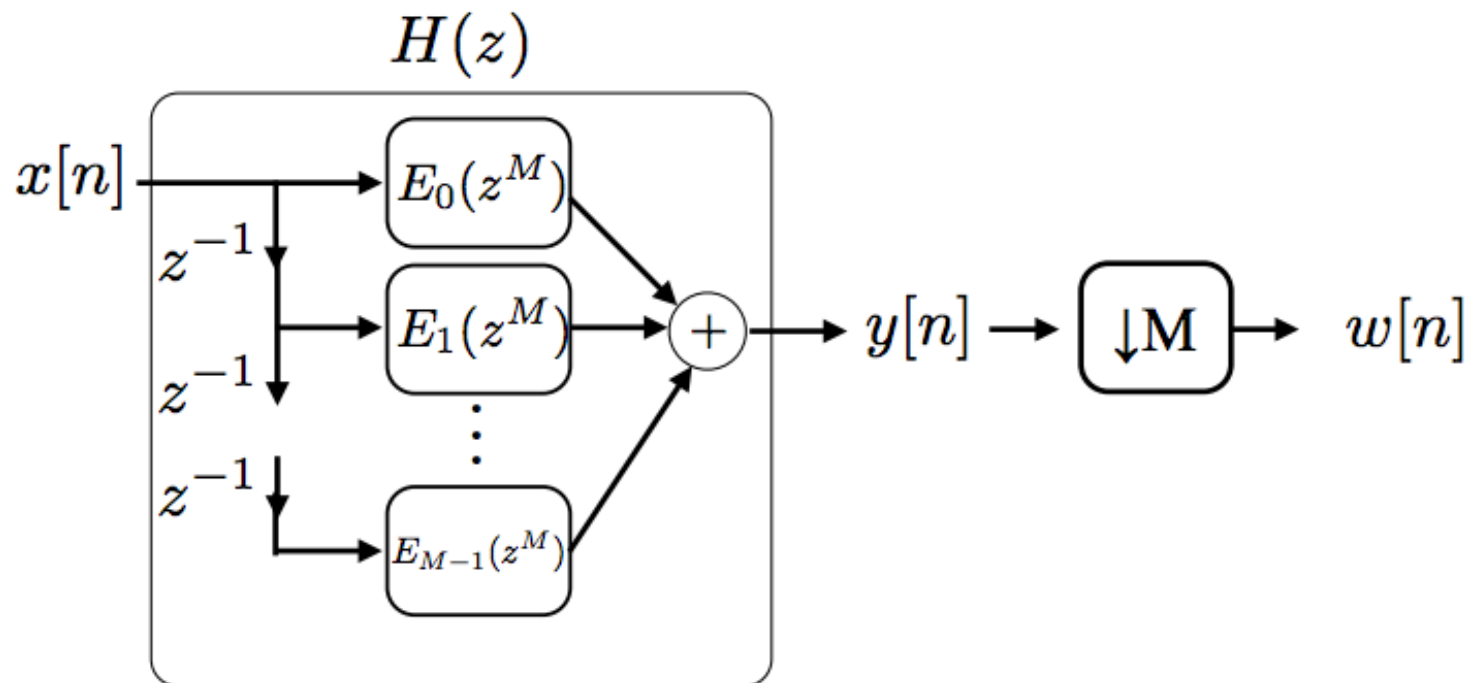


□ Problem:

- Compute all $y[n]$ and then throw away -- wasted computation!
- For FIR length $N \rightarrow N$ mults/unit time

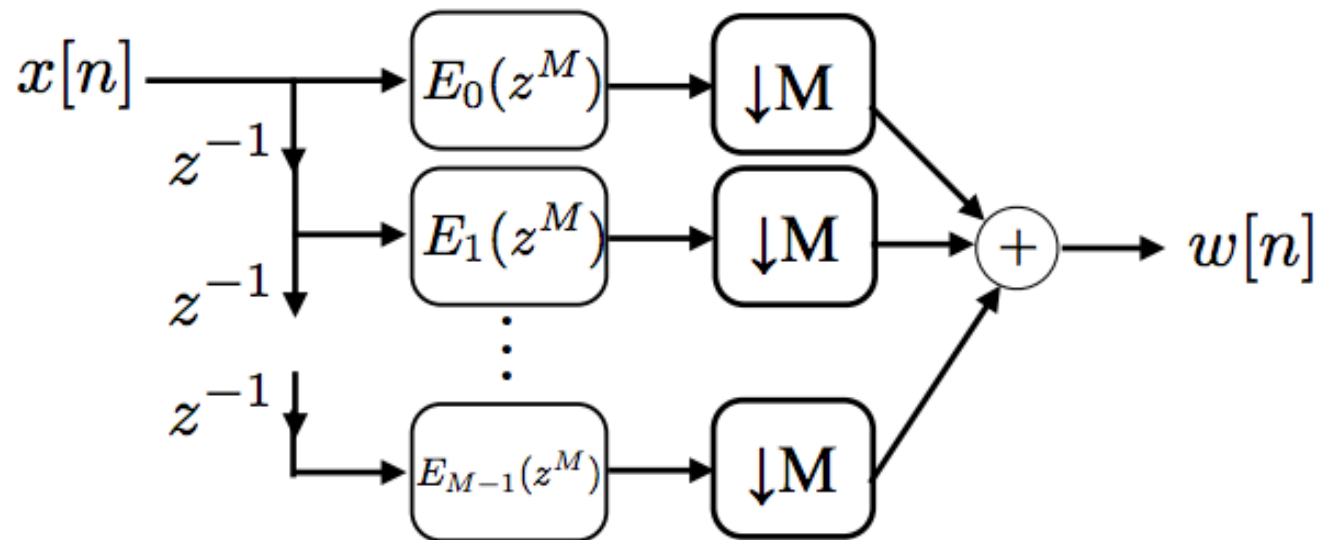
Polyphase Implementation of Decimation

$$x[n] \rightarrow H(z) \rightarrow y[n] \rightarrow \downarrow M \rightarrow w[n] = y[nM]$$



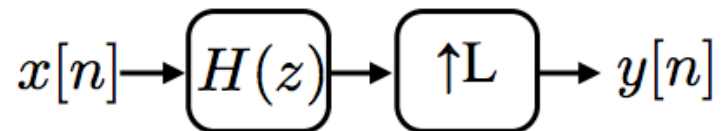
Polyphase Implementation of Decimation

$$x[n] \rightarrow H(z) \rightarrow y[n] \rightarrow \downarrow M \rightarrow w[n] = y[nM]$$

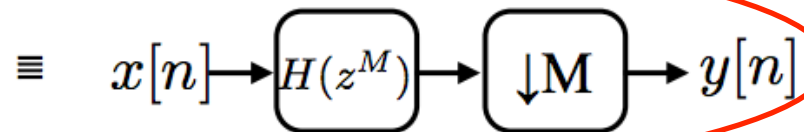
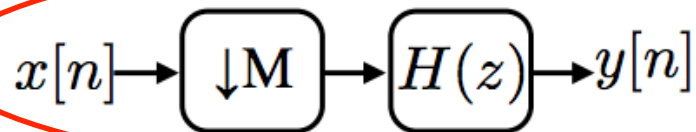
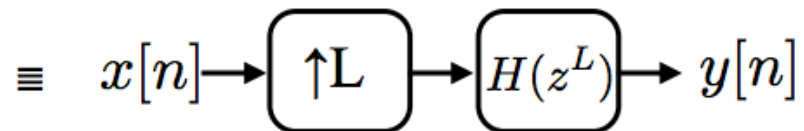


Interchanging Operations - Summary

Filter and expander



Expander and expanded filter

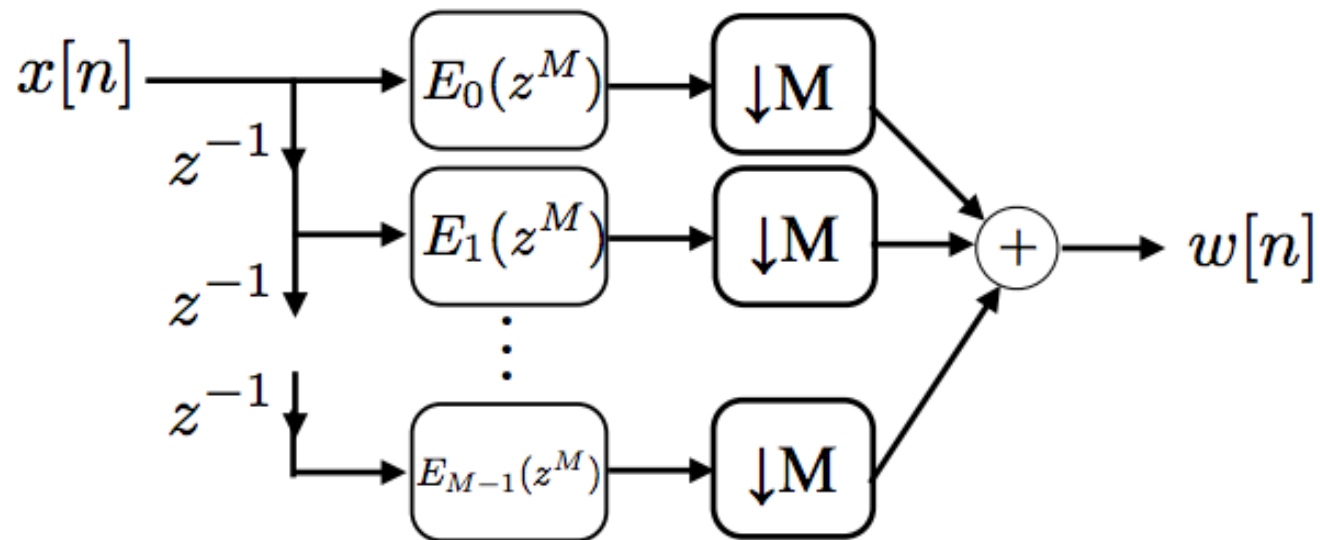


Compressor and filter

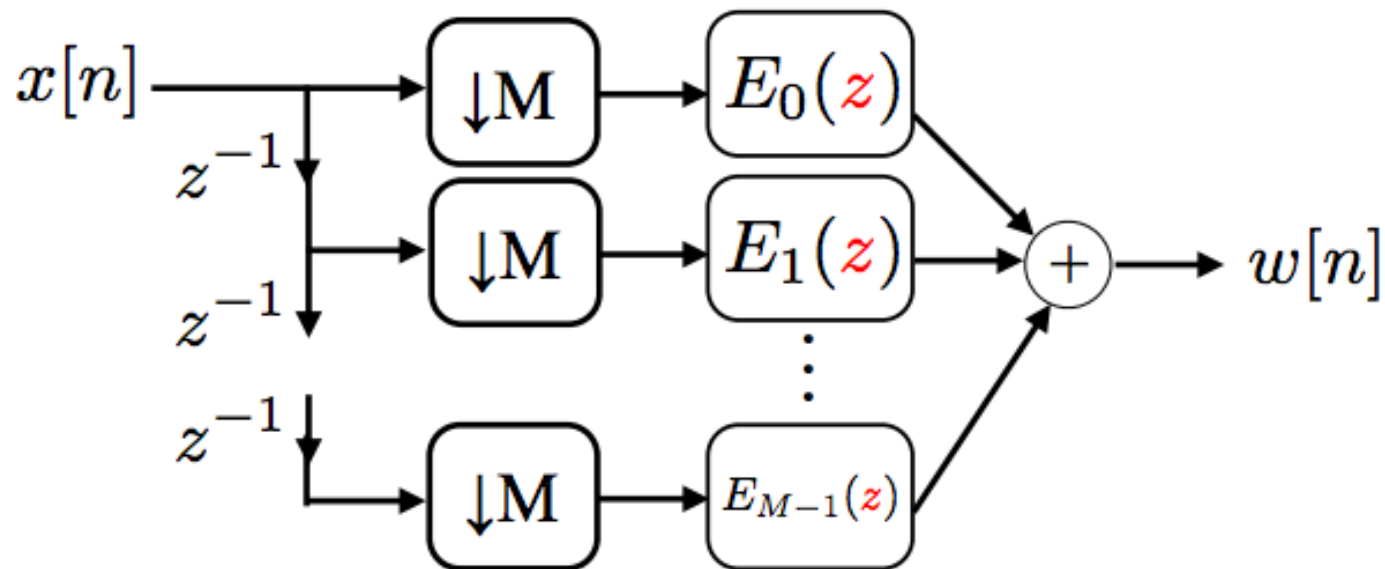
Expanded filter and compressor

Polyphase Implementation of Decimation

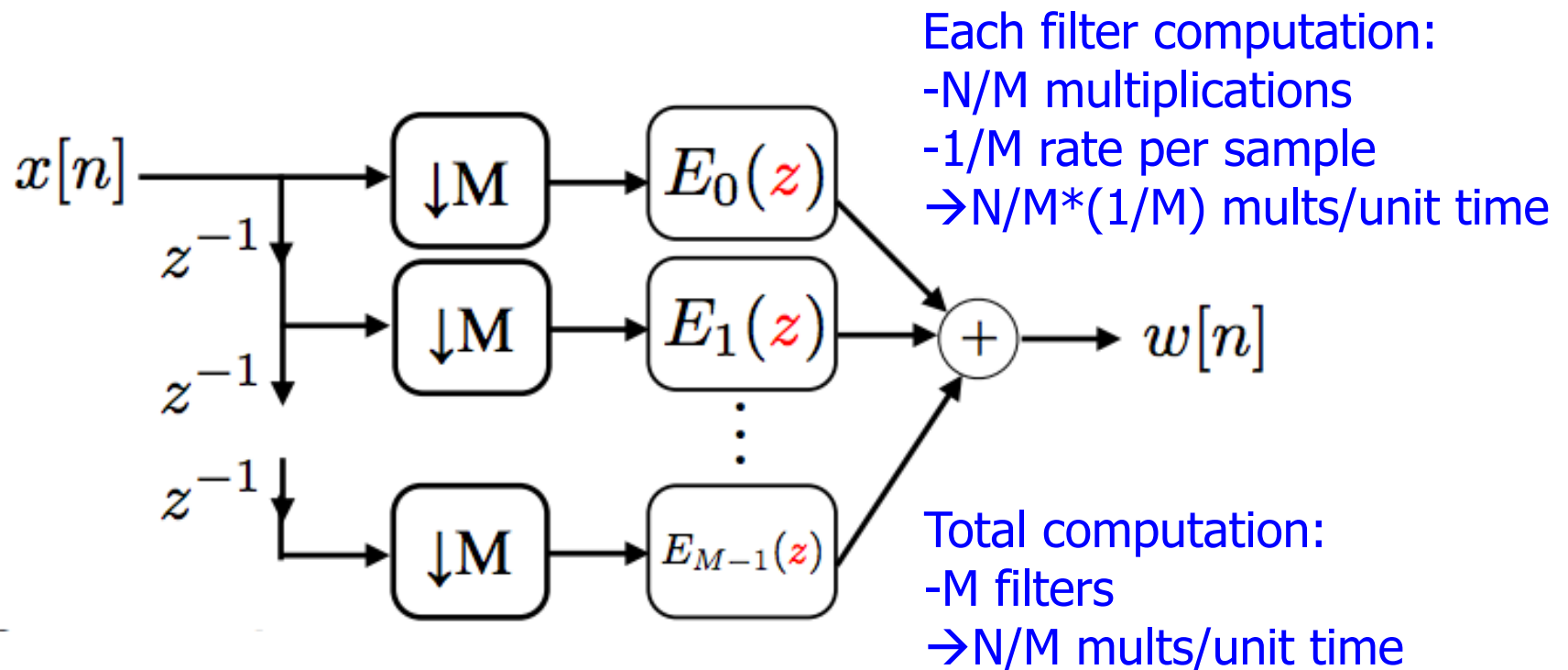
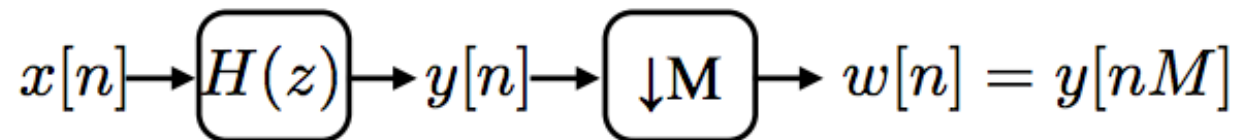
$$x[n] \rightarrow H(z) \rightarrow y[n] \rightarrow \downarrow M \rightarrow w[n] = y[nM]$$



Polyphase Implementation of Decimation



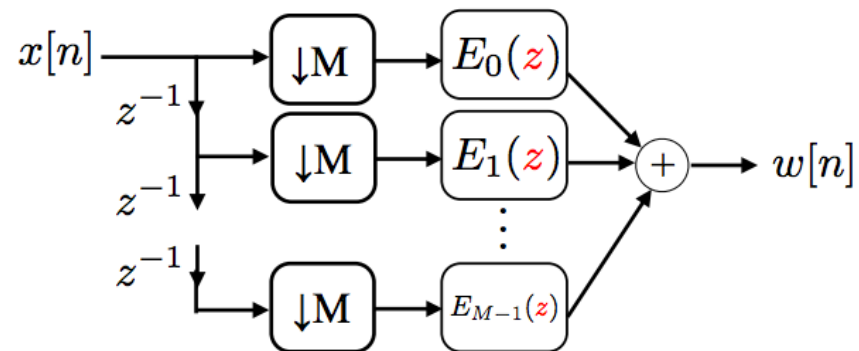
Polyphase Implementation of Decimation



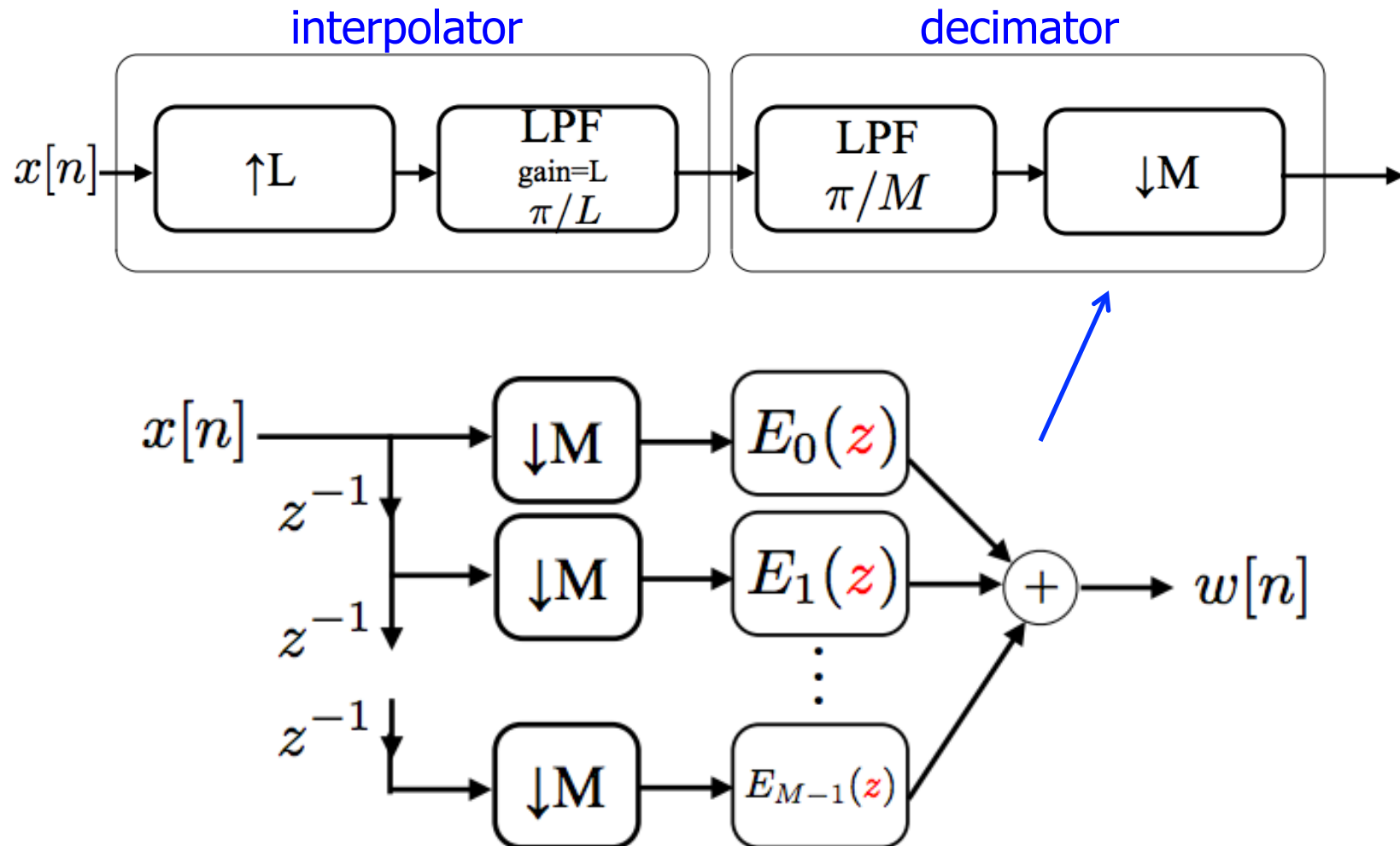
Multi-Rate Signal Processing

- ❑ What if we want to resample by $1.01T$?
 - Expand by $L=100$
 - Filter $\pi/101$ (\$\$\$\$\$)
 - Downsample by $M=101$

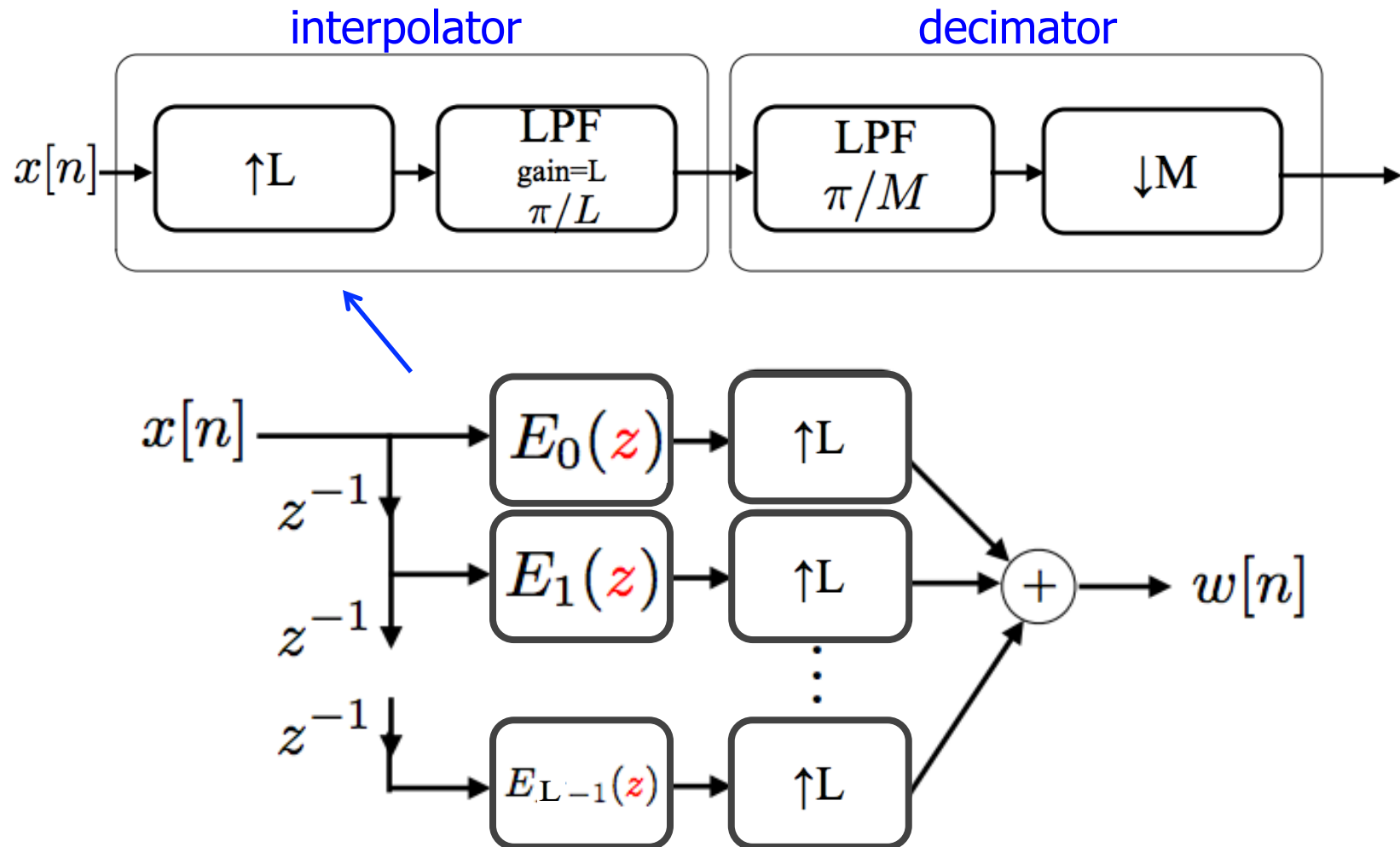
- ❑ Fortunately there are ways around it!
 - Called multi-rate
 - Uses compressors, expanders and filtering



Polyphase Implementation of Decimator



Polyphase Implementation of Interpolation



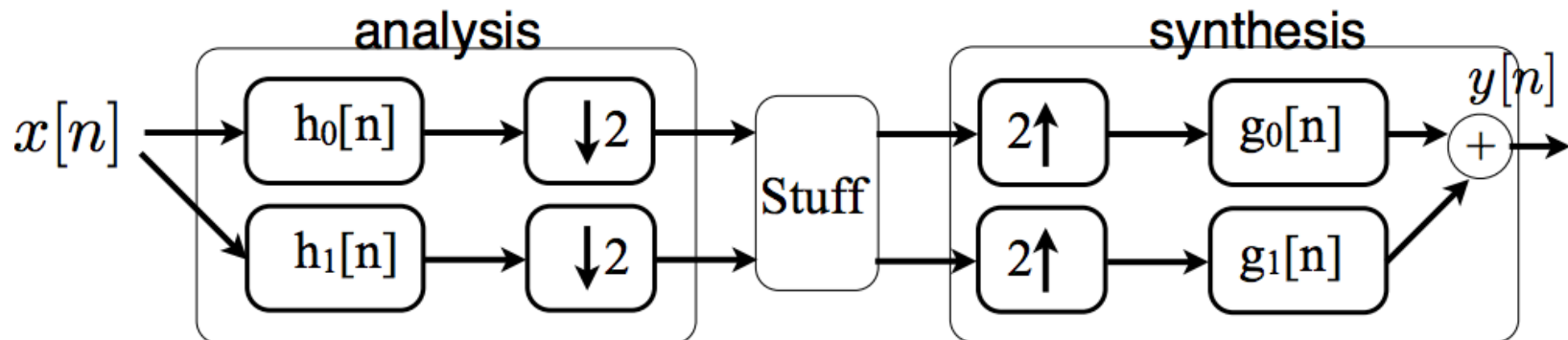


Multi-Rate Filter Banks

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

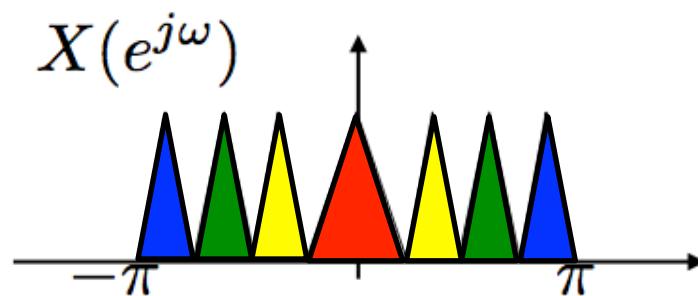
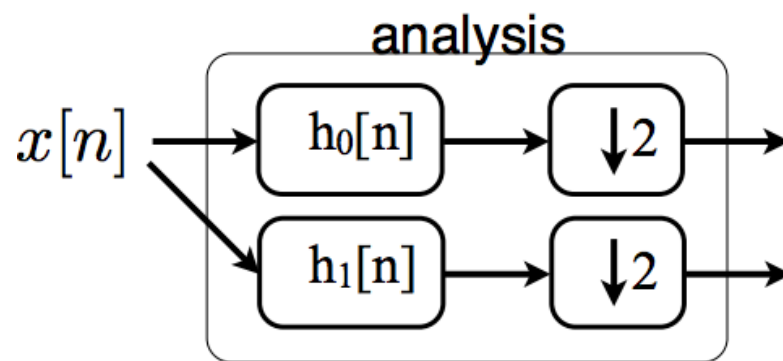
Multi-Rate Filter Banks

- ❑ 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 $h_1[n] = e^{j\pi n} h_0[n] \leftarrow$ shift freq resp by π



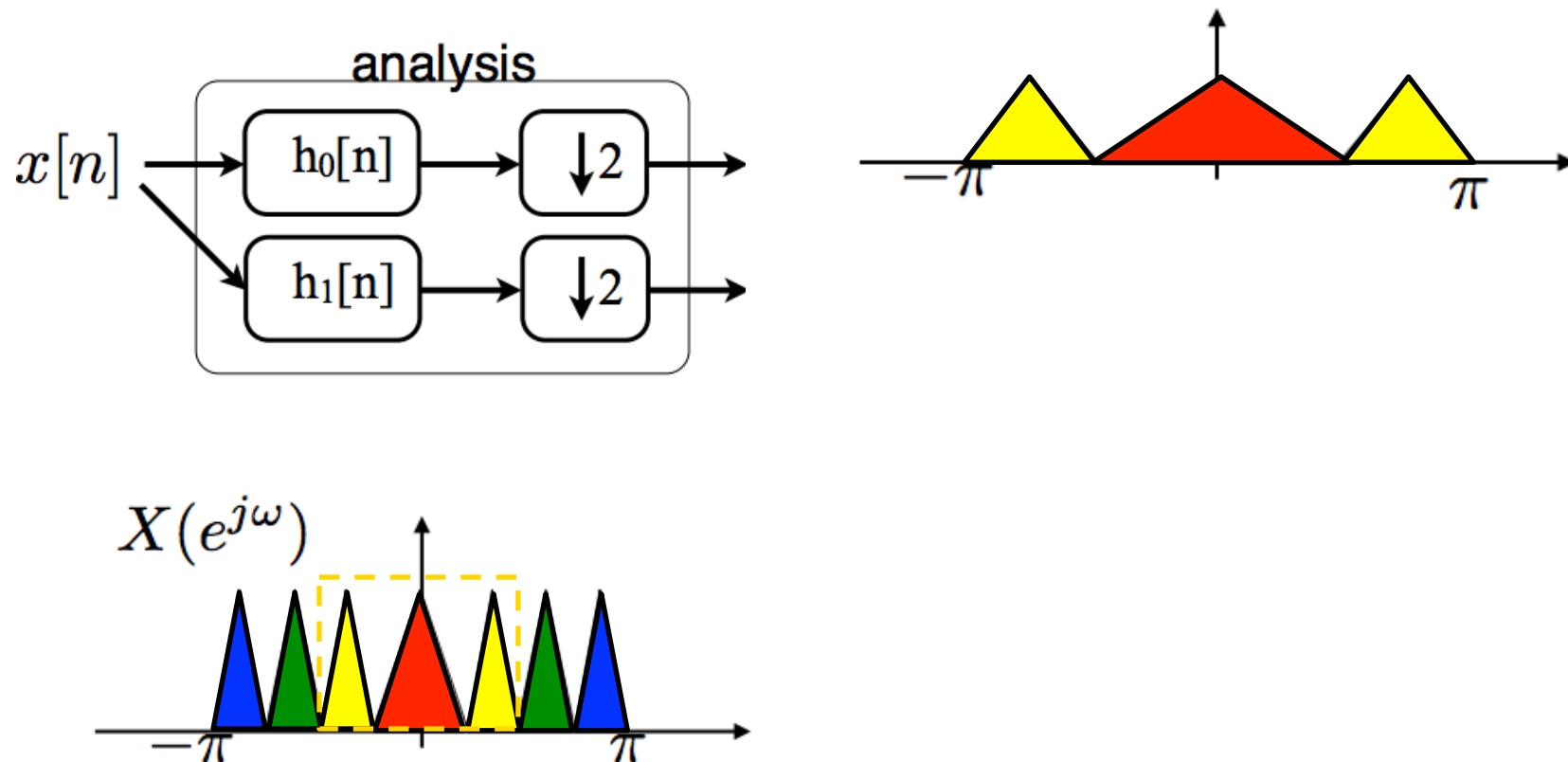
Multi-Rate Filter Banks

- Assume h_0, h_1 are ideal low/high pass with $\omega_c = \pi/2$



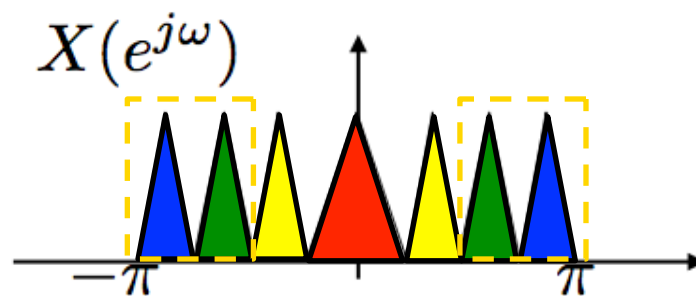
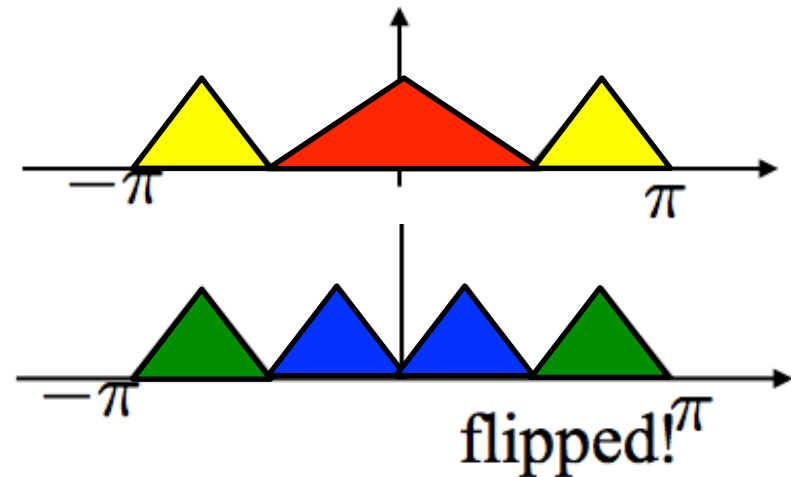
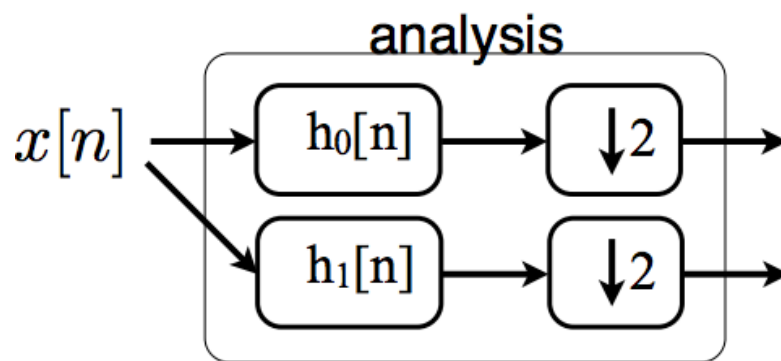
Multi-Rate Filter Banks

- Assume h_0, h_1 are ideal low/high pass with $\omega_c = \pi/2$



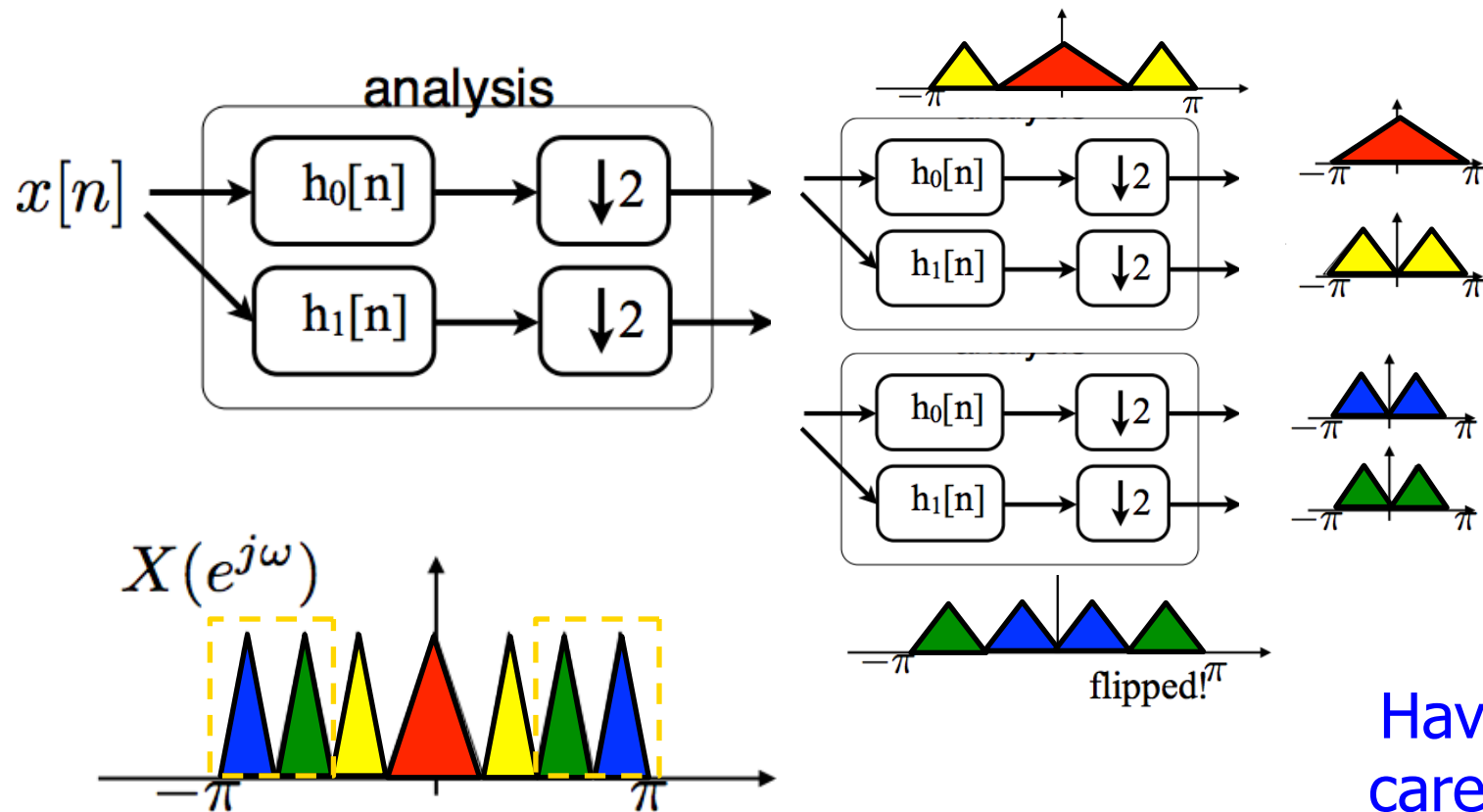
Multi-Rate Filter Banks

- Assume h_0, h_1 are ideal low/high pass with $\omega_c = \pi/2$



Multi-Rate Filter Banks

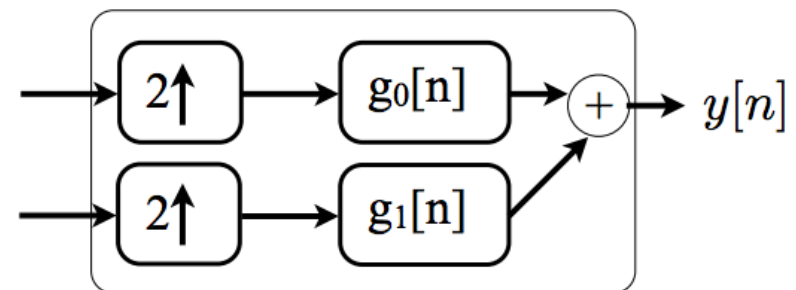
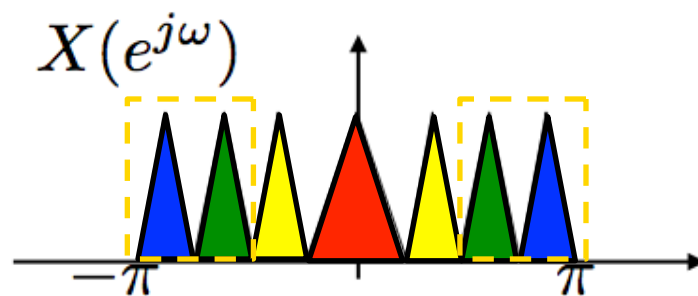
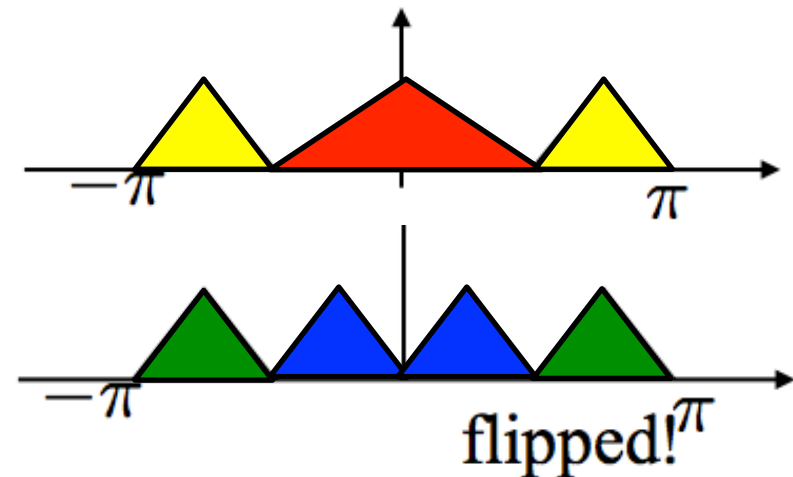
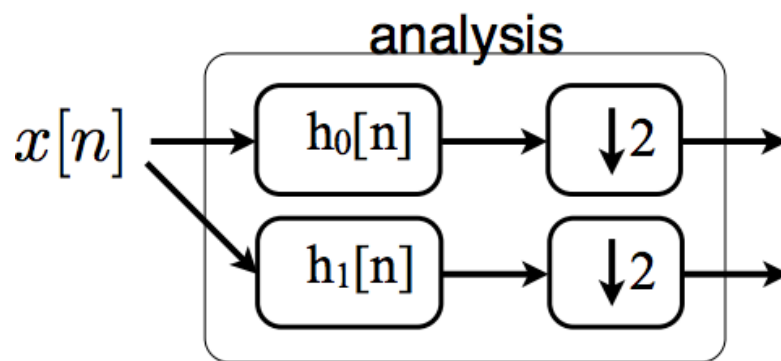
- Assume h_0, h_1 are ideal low/high pass with $\omega_c = \pi/2$



Have to be
careful with
order!

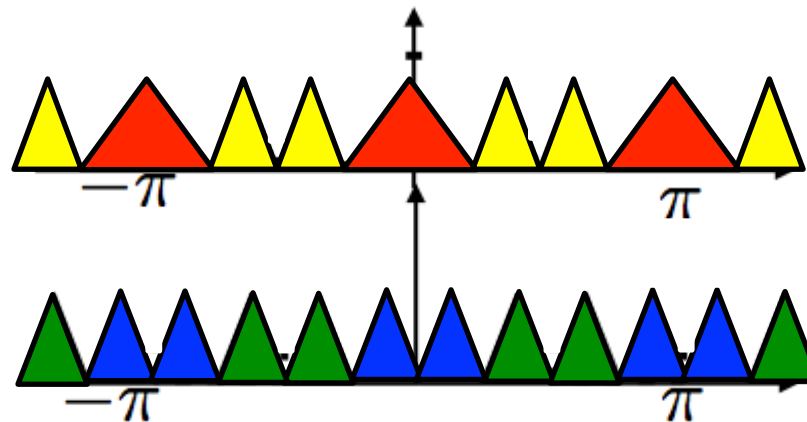
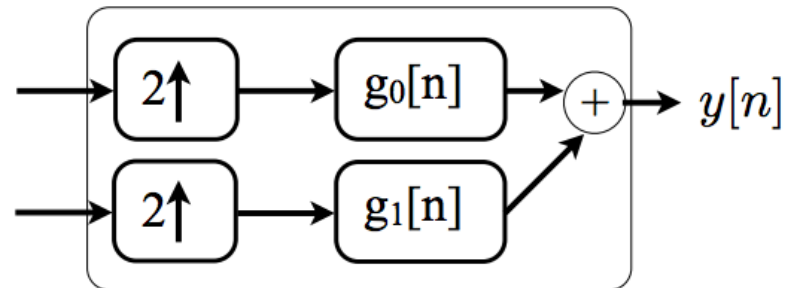
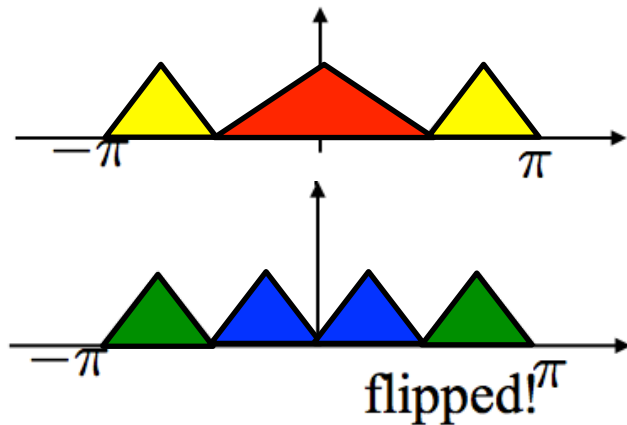
Multi-Rate Filter Banks

- Assume h_0, h_1 are ideal low/high pass with $\omega_c = \pi/2$



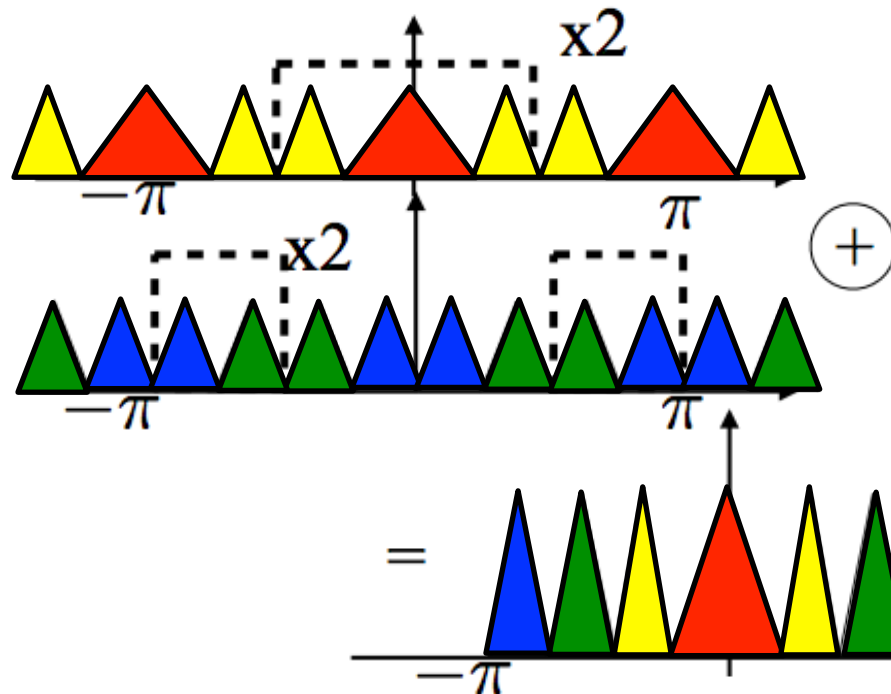
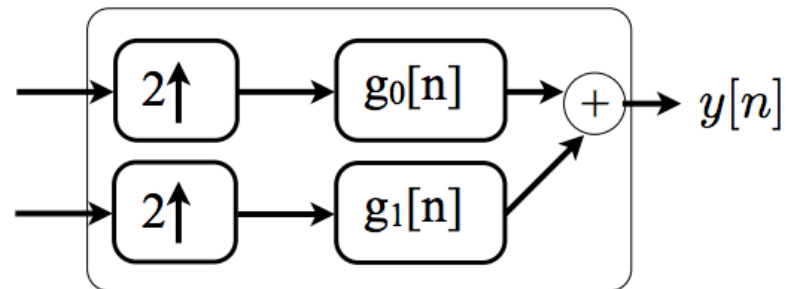
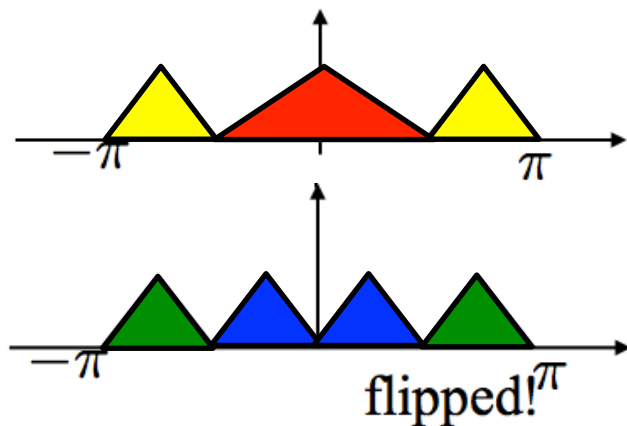
Multi-Rate Filter Banks

- Assume h_0, h_1 are ideal low/high pass with $\omega_c = \pi/2$



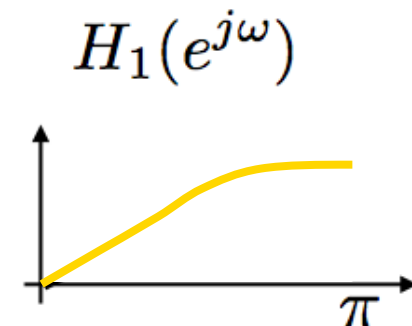
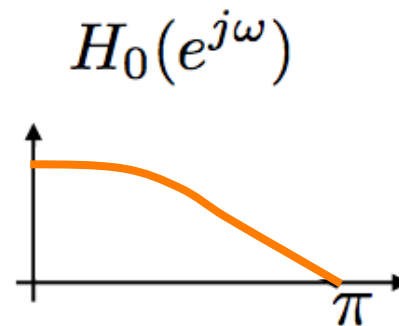
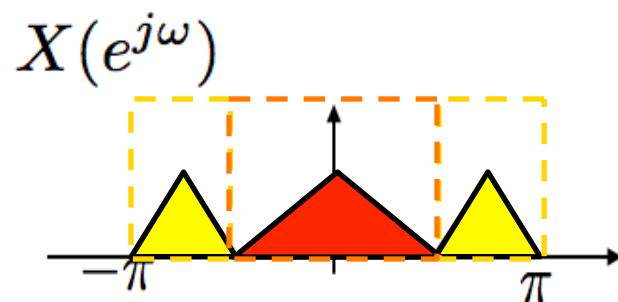
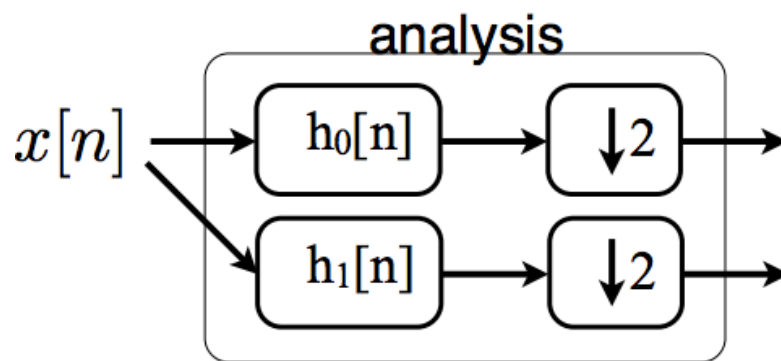
Multi-Rate Filter Banks

- Assume h_0, h_1 are ideal low/high pass with $\omega_c = \pi/2$



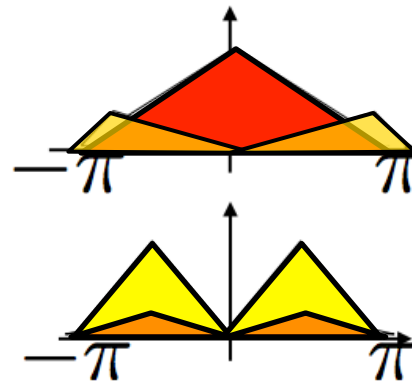
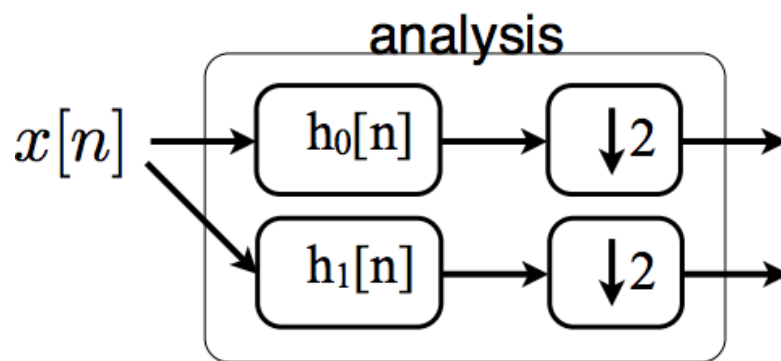
Multi-Rate Filter Banks

- h_0, h_1 are **NOT** ideal low/high pass

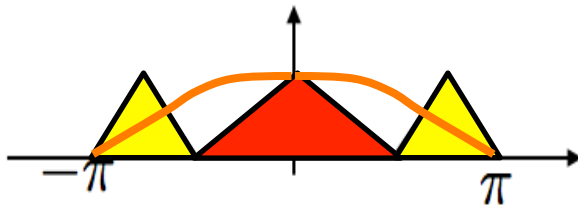


Non Ideal Filters

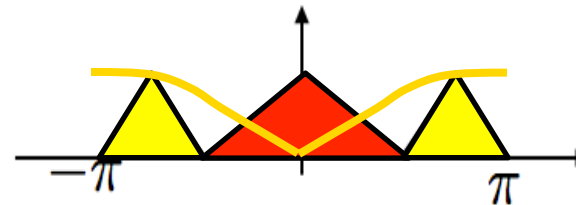
- h_0, h_1 are **NOT** ideal low/high pass



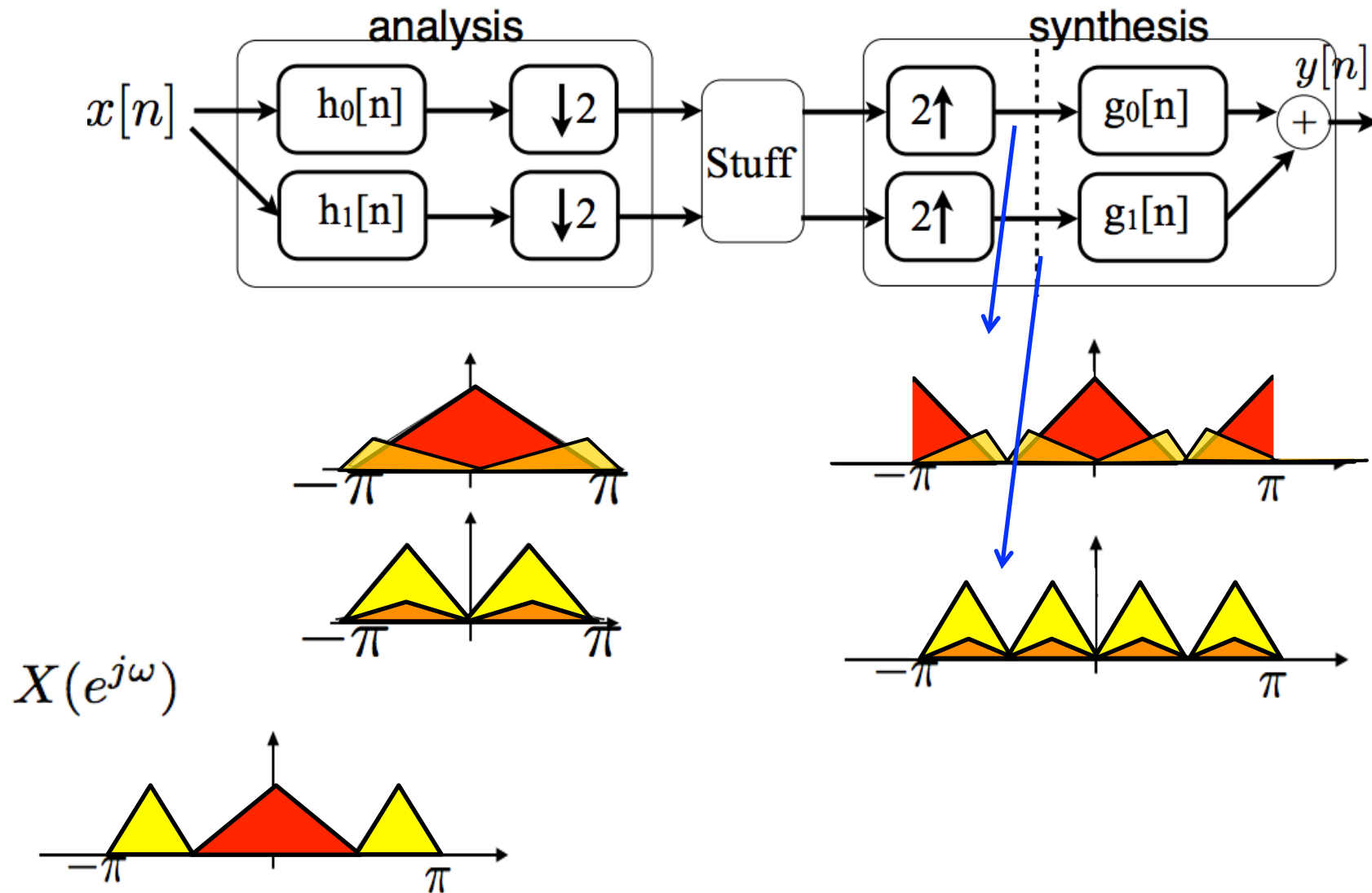
$$X(e^{j\omega})$$



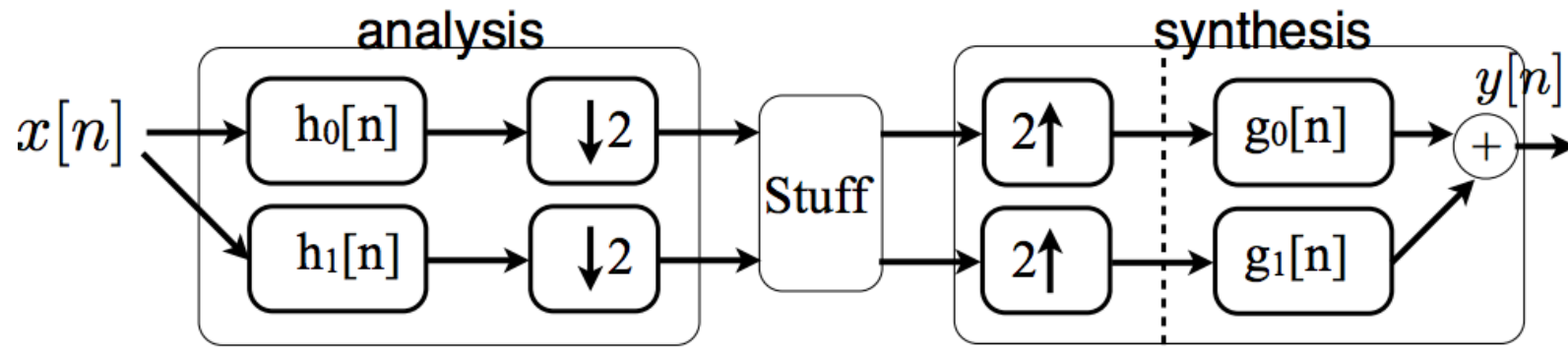
$$X(e^{j\omega})$$



Non Ideal Filters



Perfect Reconstruction non-Ideal Filters

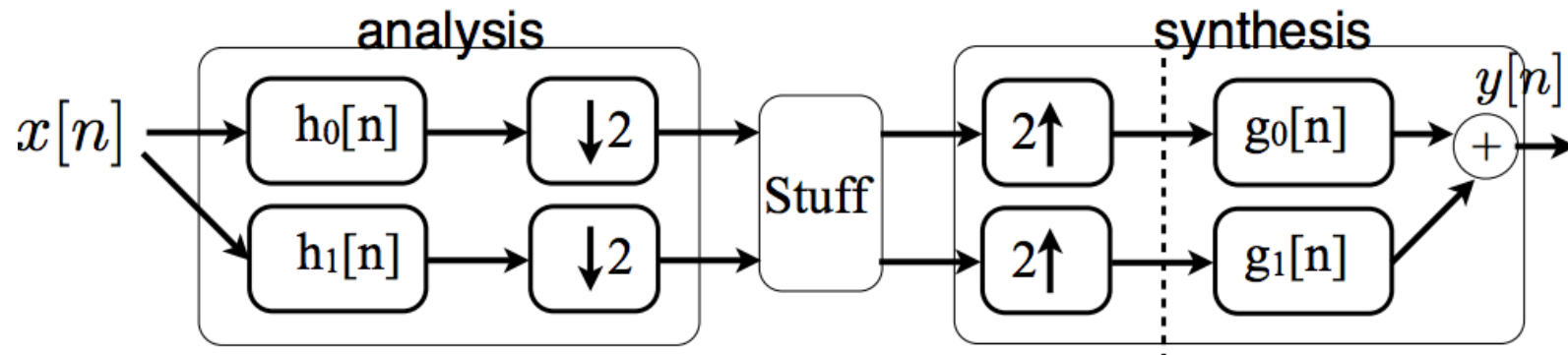


$$\begin{aligned}
 Y(e^{j\omega}) &= \frac{1}{2} [G_0(e^{j\omega})H_0(e^{j\omega}) + G_1(e^{j\omega})H_1(e^{j\omega})] X(e^{j\omega}) \\
 &\quad + \frac{1}{2} [G_0(e^{j\omega})H_0(e^{j(\omega-\pi)}) + G_1(e^{j\omega})H_1(e^{j(\omega-\pi)})] X(e^{j(\omega-\pi)})
 \end{aligned}$$

↑
↑

need to cancel!
aliasing

Quadrature Mirror Filters



Quadrature mirror filters

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

Perfect Reconstruction non-Ideal Filters

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

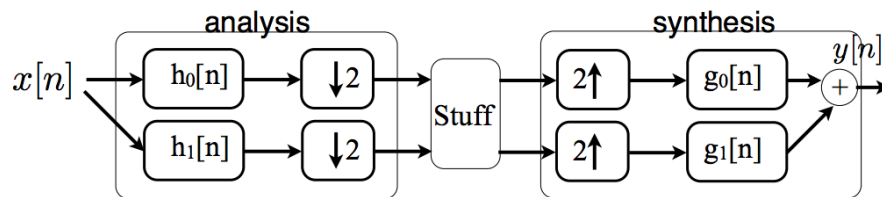
↑
need to cancel!

↑
aliasing

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

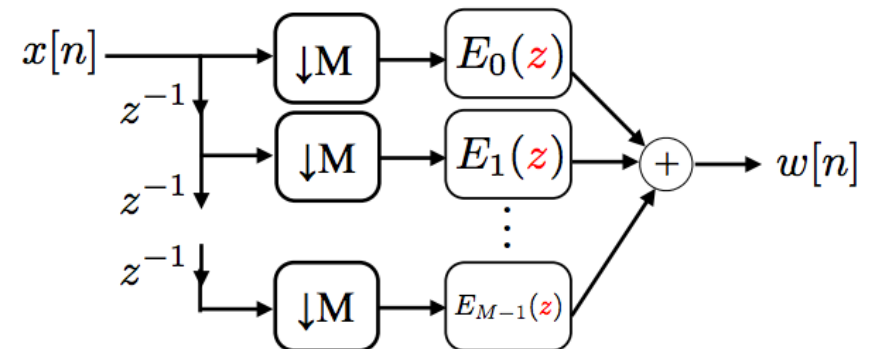
Big Ideas

- ❑ Downsampling/Upsampling
- ❑ Practical Interpolation
- ❑ Non-integer Resampling
- ❑ Multi-Rate Processing
 - Interchanging Operations
- ❑ Polyphase Decomposition
- ❑ Multi-Rate Filter Banks



$$x[n] \rightarrow H(z) \rightarrow \uparrow L \rightarrow y[n] \quad \equiv \quad x[n] \rightarrow \uparrow L \rightarrow H(z^L) \rightarrow y[n]$$

$$x[n] \rightarrow \downarrow M \rightarrow H(z) \rightarrow y[n] \quad \equiv \quad x[n] \rightarrow H(z^M) \rightarrow \downarrow M \rightarrow y[n]$$





Admin

- ❑ HW 4 due Sunday