ESE 531: Digital Signal Processing

Lec 10: February 14th, 2017

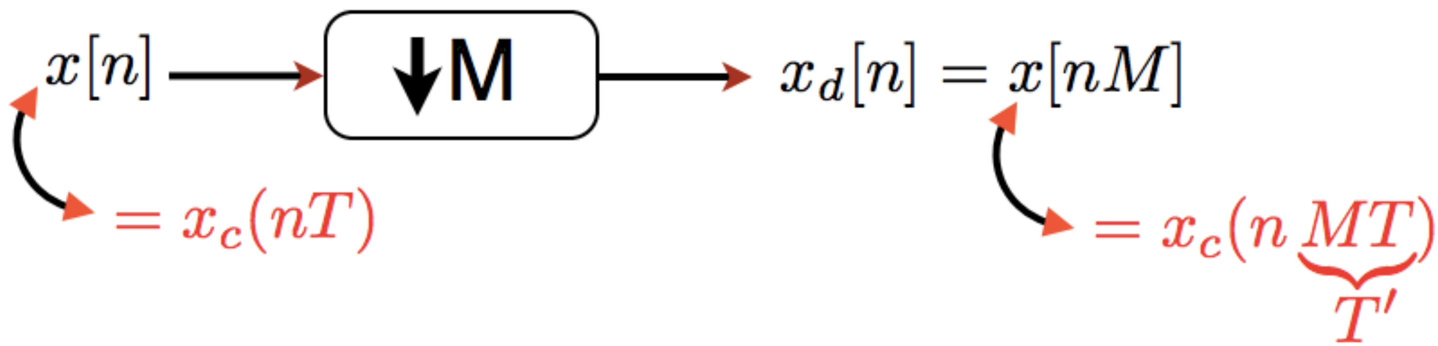
Practical and Non-integer Sampling, Multi-rate Sampling

Lecture Outline

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

Downsampling

- Definition: Reducing the sampling rate by an integer number



Downsampling

$$X_d(e^{j\omega}) = \frac{1}{MT}\sum_k X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi}{MT}k\right)\right)$$

$$= \frac{1}{M}\sum_{i=0}^{M-1} \underbrace{\frac{1}{T}\sum_{r=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi}{MT}i - \frac{2\pi}{T}r\right)\right)}_{X\left(e^{j\left(\frac{\omega}{M} - \frac{2\pi}{M}i\right)}\right)}$$

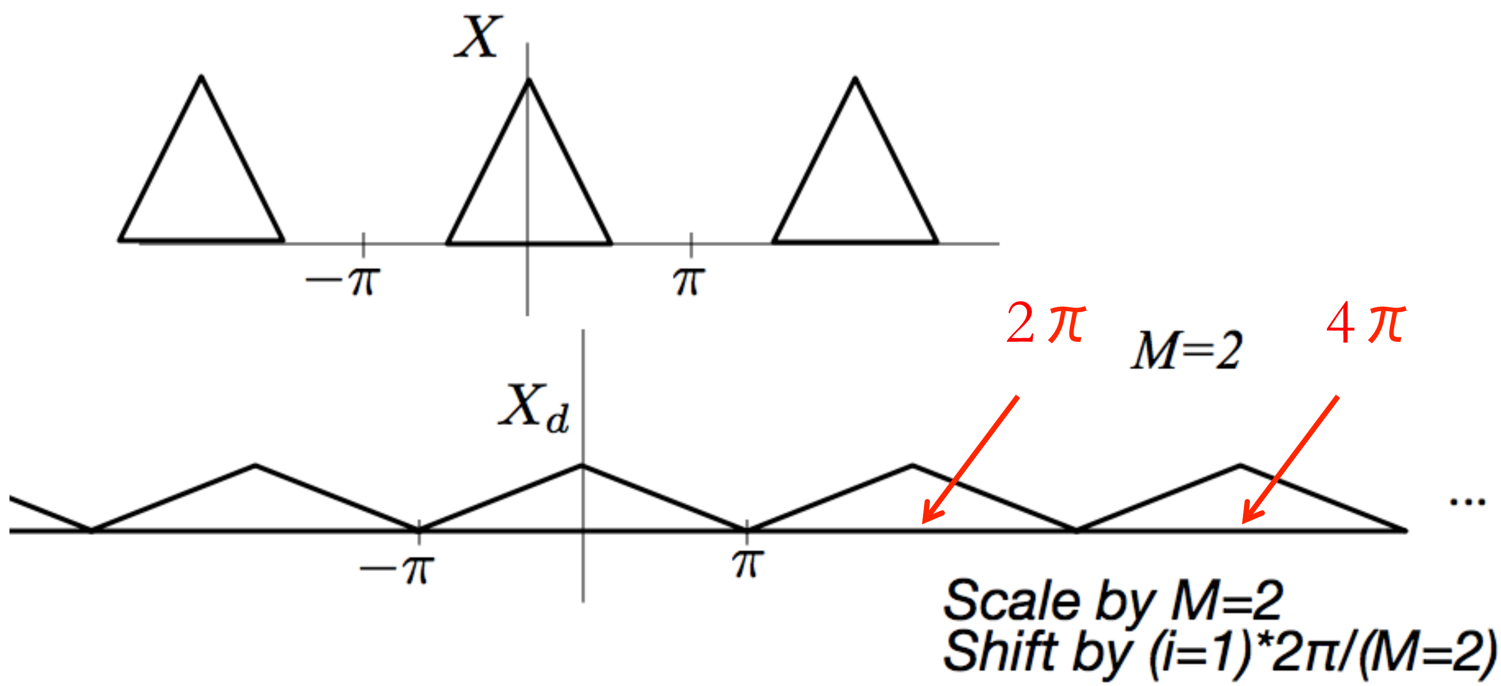
$$X(e^{j\omega}) = \frac{1}{T}\sum_k X_c\left(j\left(\underbrace{\frac{\omega}{T}} - \underbrace{\frac{2\pi}{T}}k\right)\right)$$

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

stretch by M (↑ $\frac{\omega}{M}$) replicate (↑ $\frac{2\pi}{M}i$)

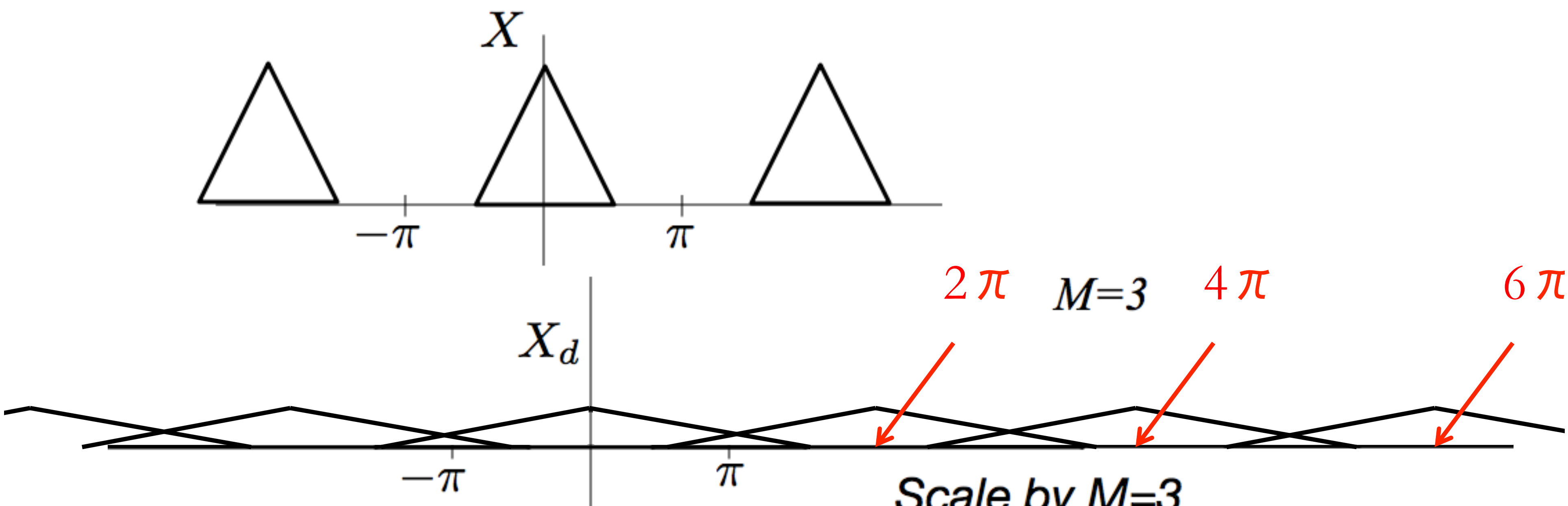
Example

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



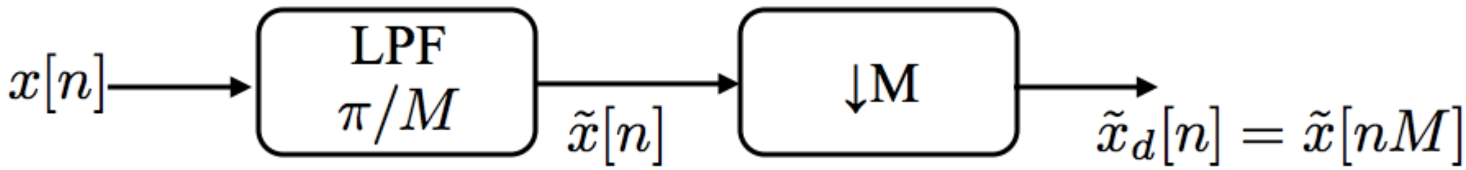
Example

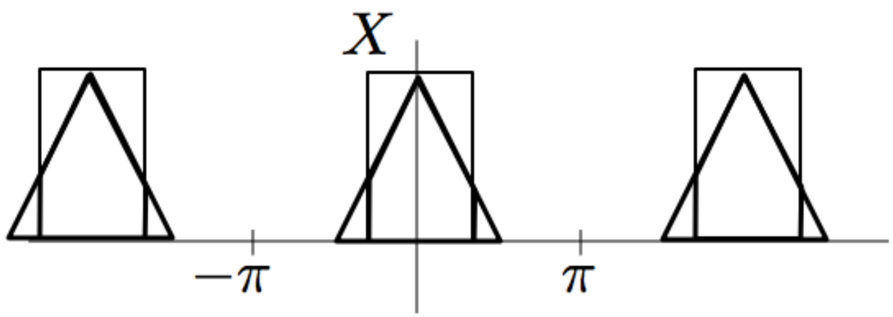
$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j\left(\frac{w}{M} - \frac{2\pi}{M} i\right)}\right)$$



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

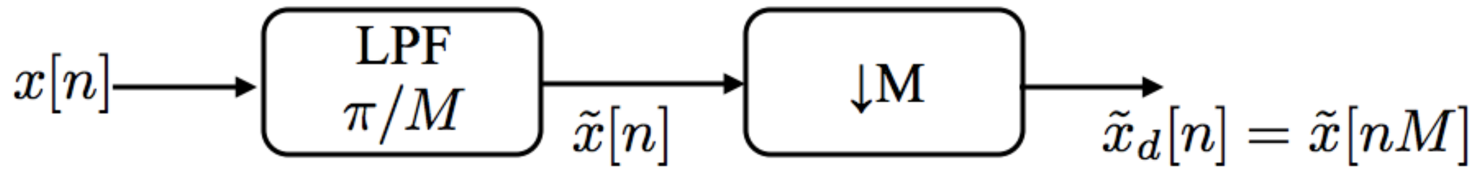
Example



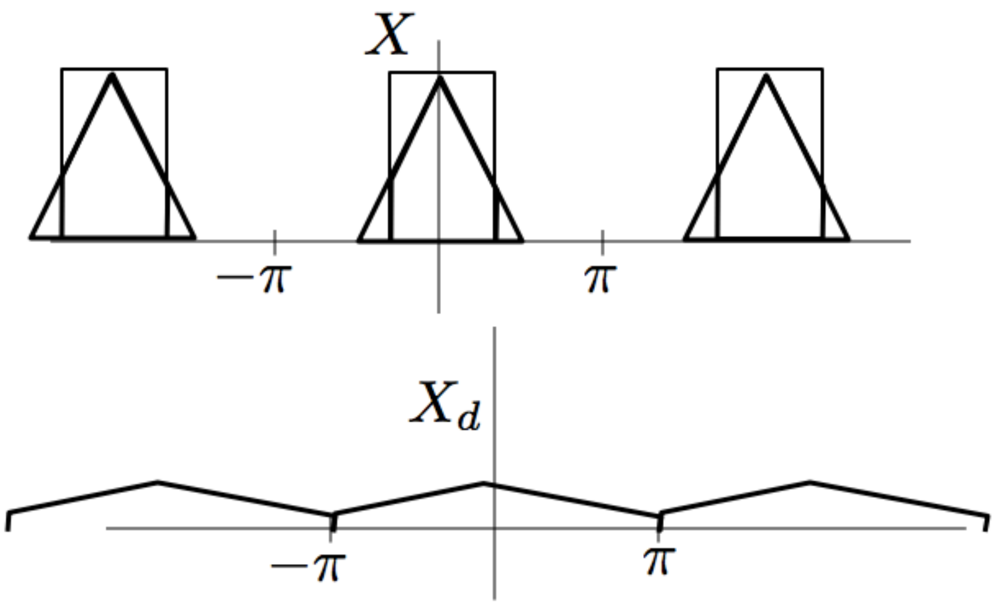


$M=3$

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

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

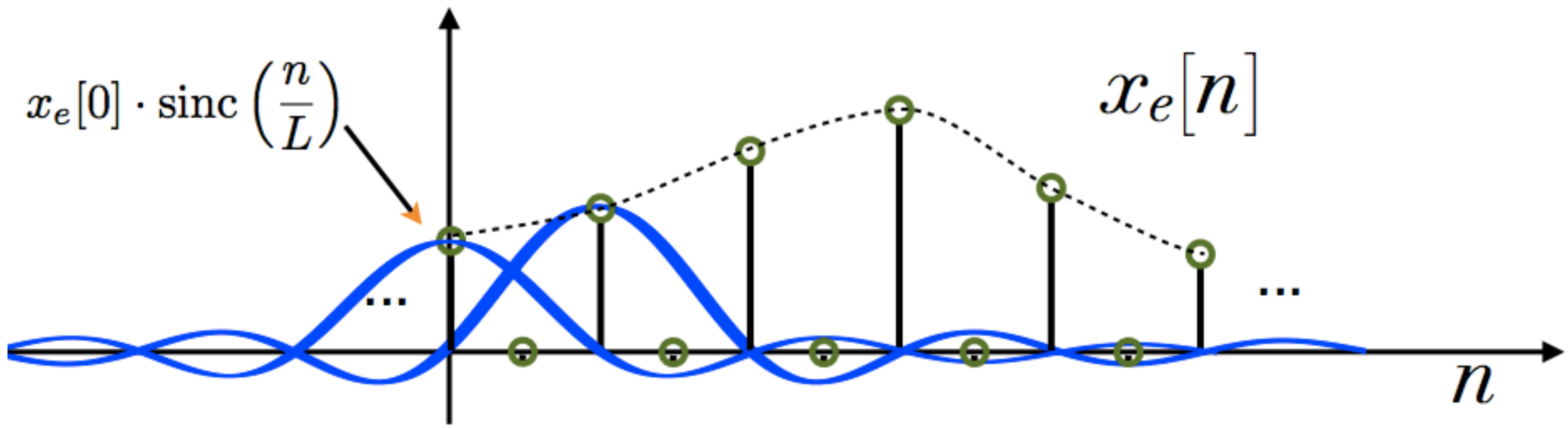
$$x_i[n] = x_c(nT') \quad \text{where } T' = \frac{T}{L} \qquad L \text{ integer}$$

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

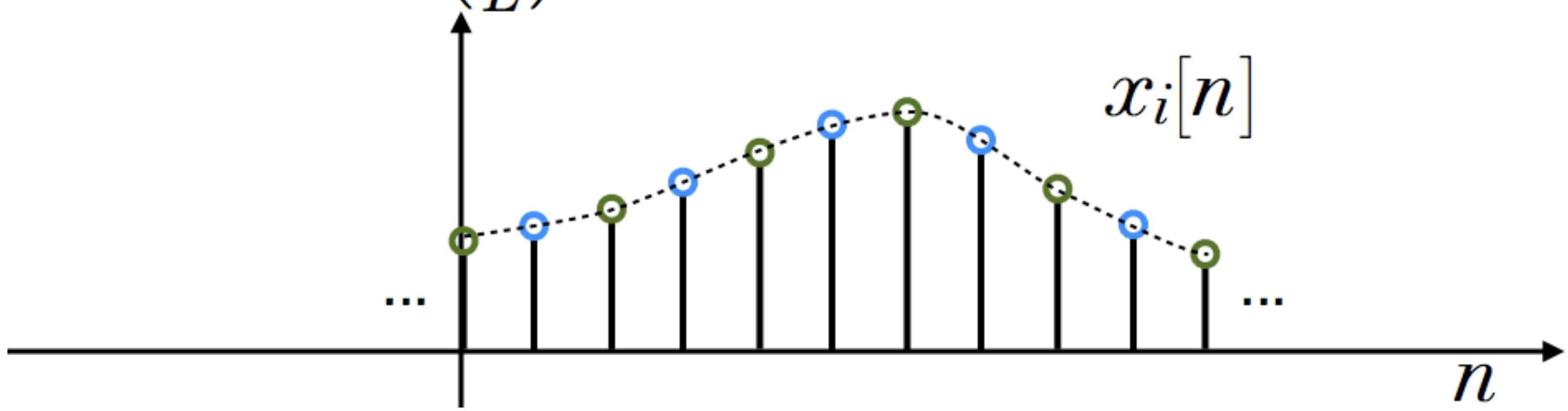
(1) Generate: $x_e[n] = \begin{cases} x[n/L] & n = 0, \ \pm L, \ \pm 2L, \cdots \\ 0 & \text{otherwise} \end{cases}$

Upsampling

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

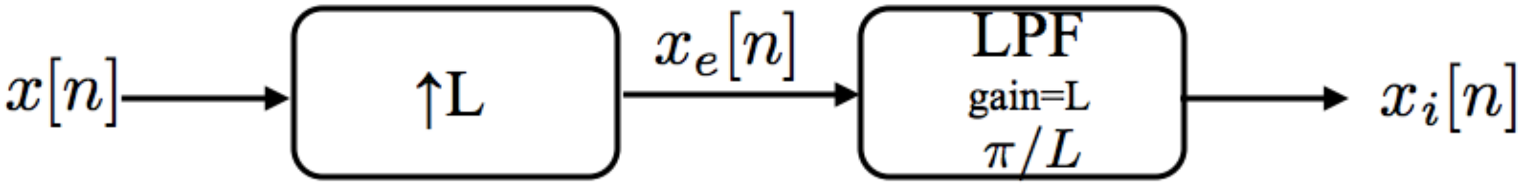


$$x_i[n] = x_e[n] * \text{sinc}\left(\frac{n}{L}\right)$$

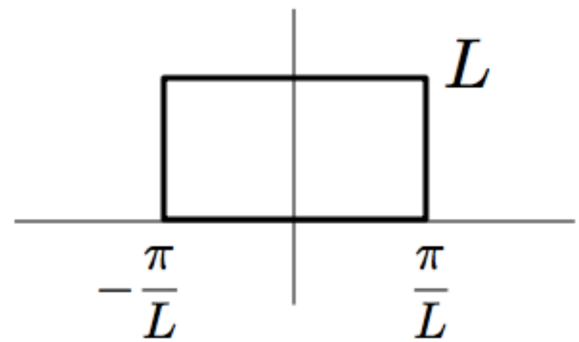


Frequency Domain Interpretation

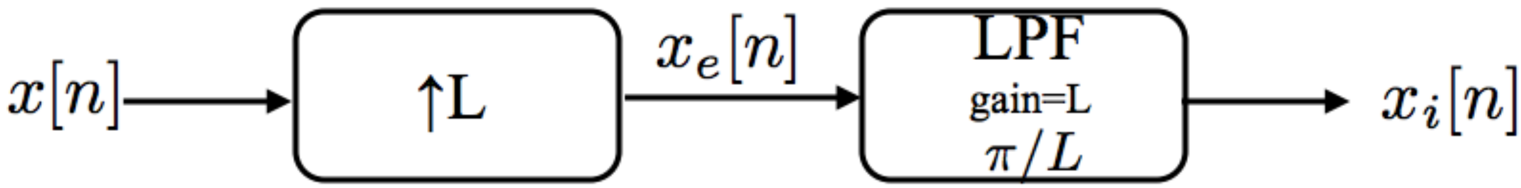
$$x_i[n] = x_e[n] * \text{sinc}(n/L)$$



$\text{sinc}(n/L)$ **DTFT ⇒**



Frequency Domain Interpretation

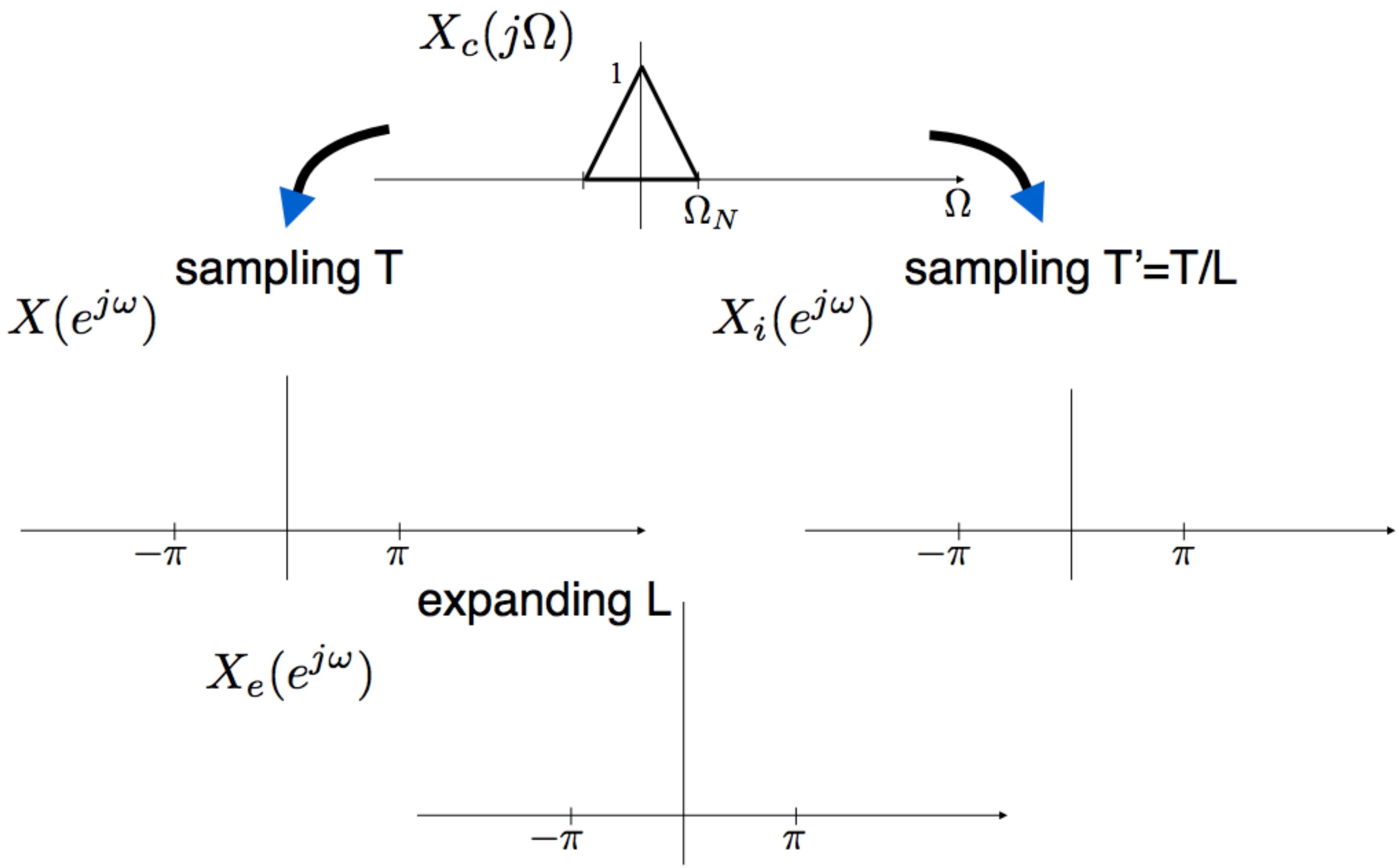


$$X_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \underbrace{x_e[n]}_{\neq 0 \text{ only for n=mL (integer m)}} e^{-j\omega n}$$

$$= \sum_{m=-\infty}^{\infty} \underbrace{x_e[mL]}_{=x[m]} e^{-j\omega mL} = \boxed{X(e^{j\omega L})}$$

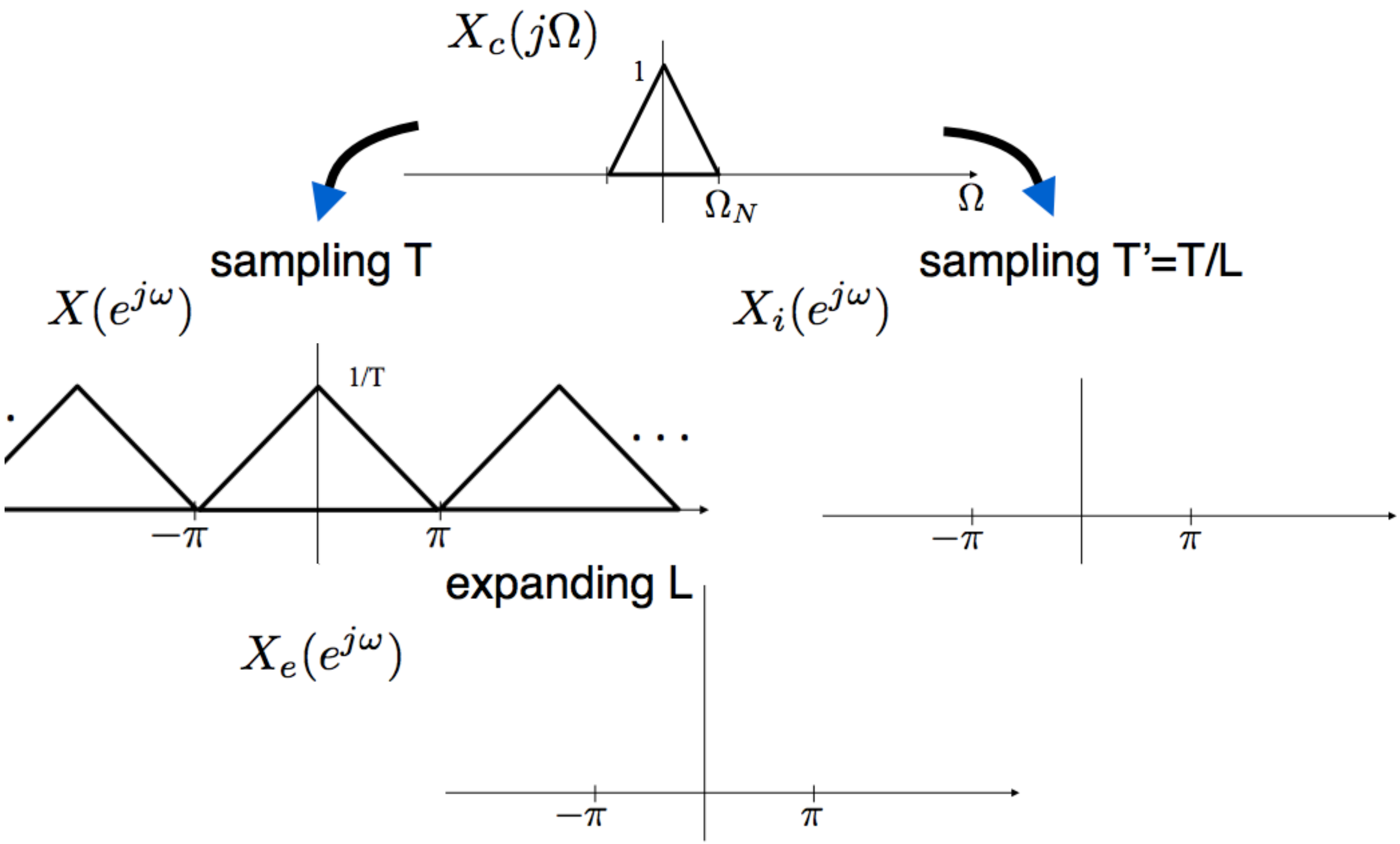
Compress DTFT by a factor of L!

Example

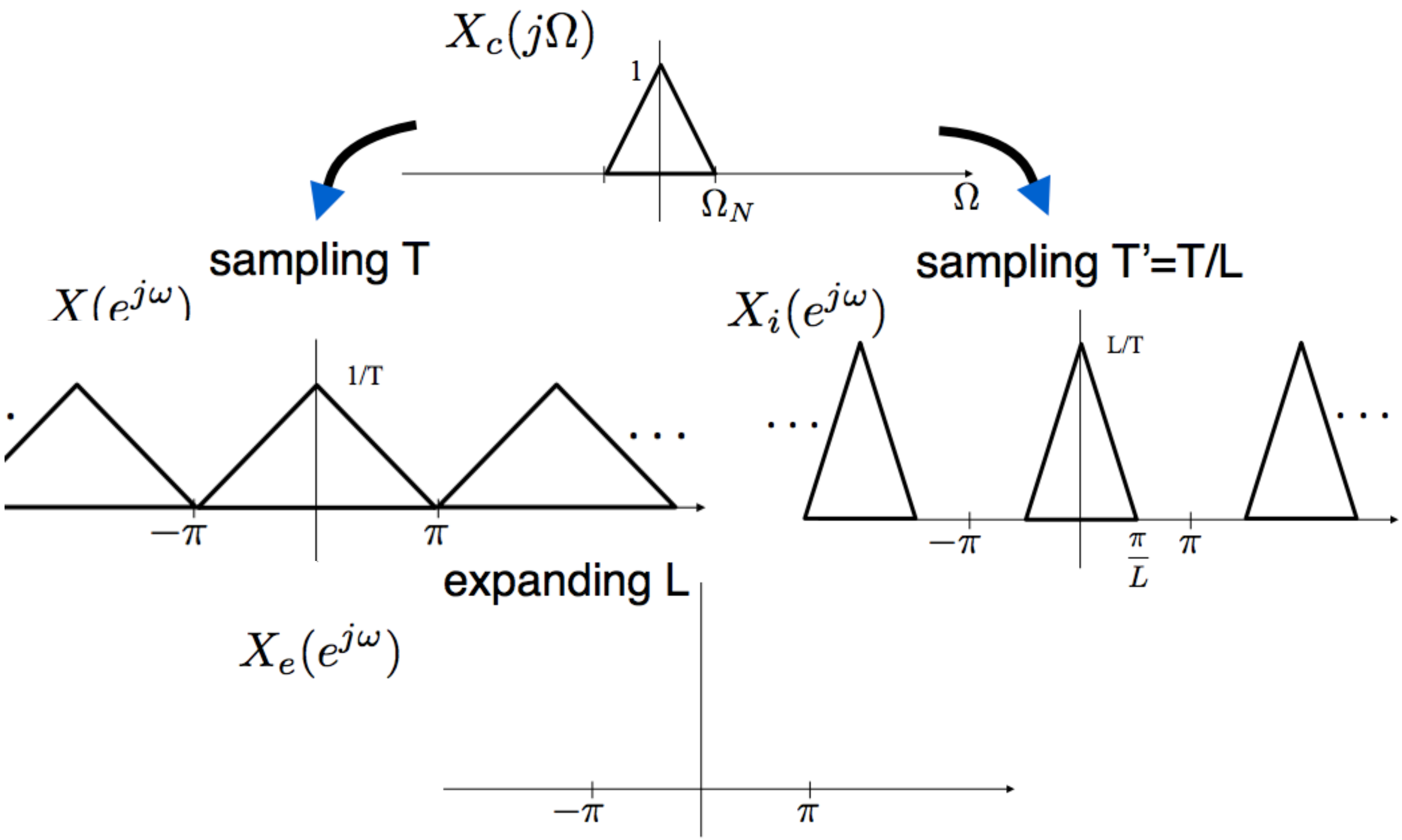


$X_c(j\Omega)$
1
Ω_N
Ω
sampling T
sampling T'=T/L
$X(e^{j\omega})$
$X_i(e^{j\omega})$
$-\pi$
π
$-\pi$
π
expanding L
$X_e(e^{j\omega})$
$-\pi$
π

Example

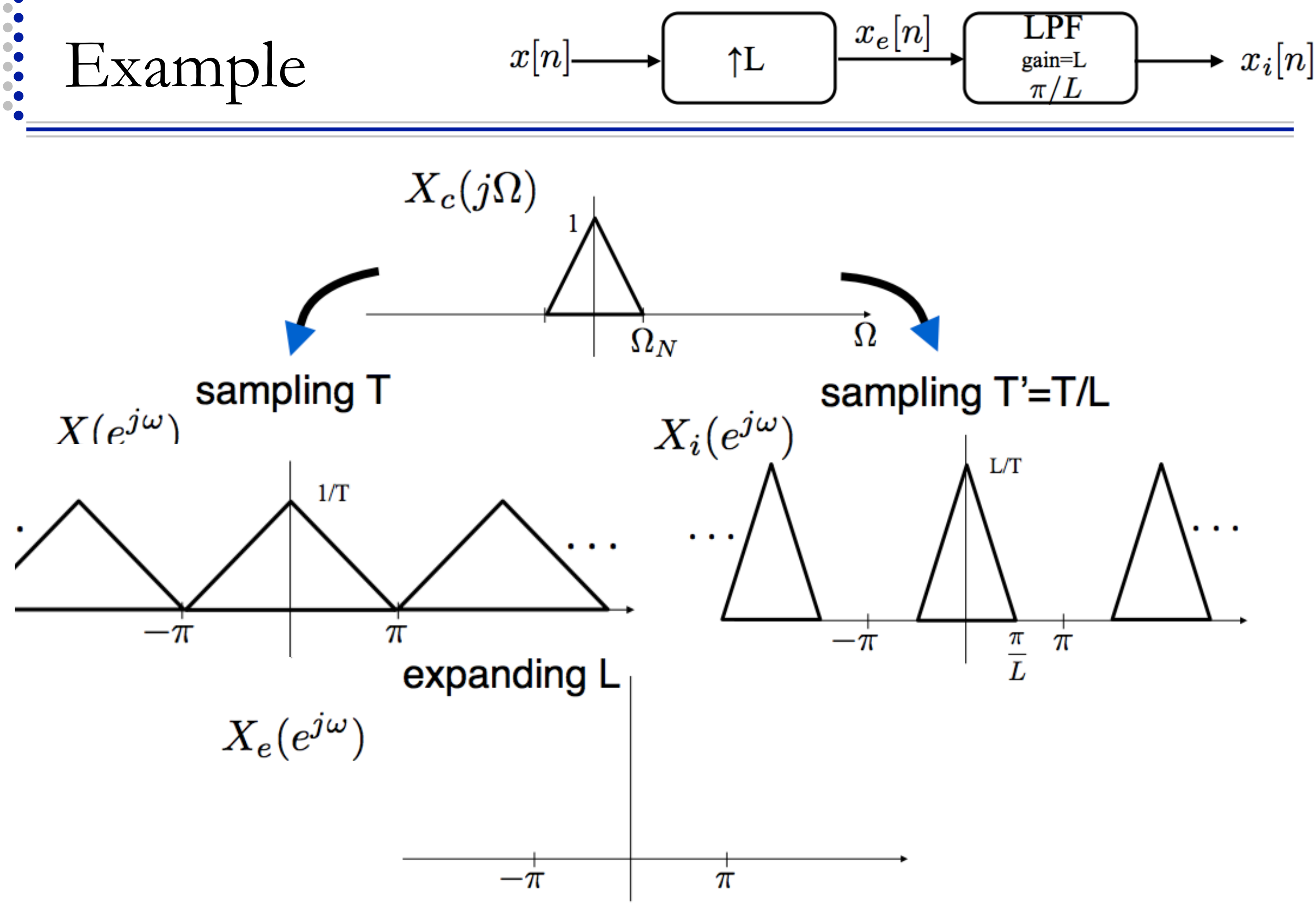

$X_c(j\Omega)$
1
Ω_N
Ω
sampling T
sampling T'=T/L
$X(e^{j\omega})$
1/T
$-\pi$
π
$X_i(e^{j\omega})$
$-\pi$
π
expanding L
$X_e(e^{j\omega})$
$-\pi$
π

Example



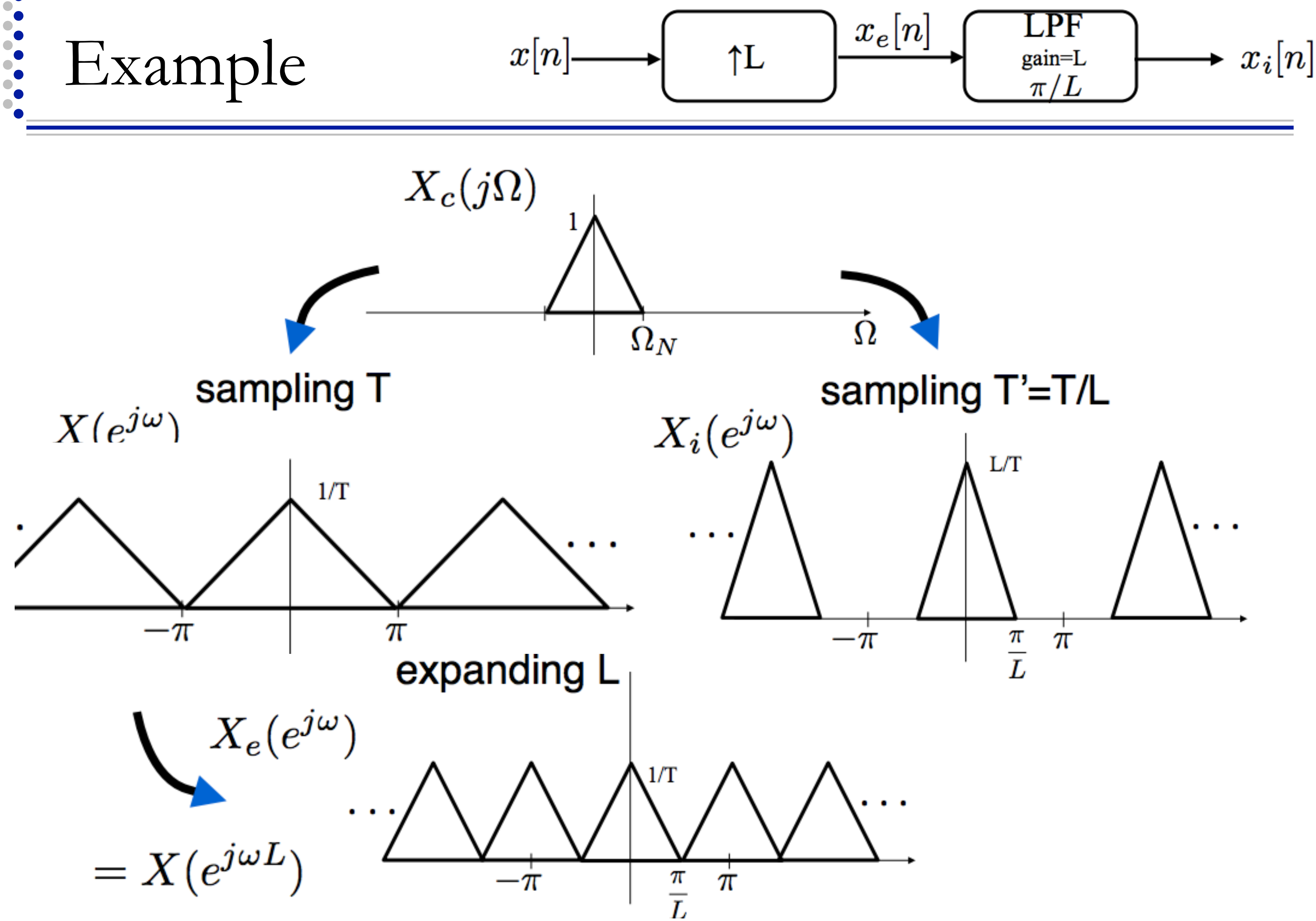
$X_c(j\Omega)$
1
Ω_N
Ω
sampling T
sampling T'=T/L
$X(e^{j\omega})$
1/T
$-\pi$
π
$X_i(e^{j\omega})$
L/T
$-\pi$
$\frac{\pi}{L}$
π
expanding L
$X_e(e^{j\omega})$
$-\pi$
π

Example



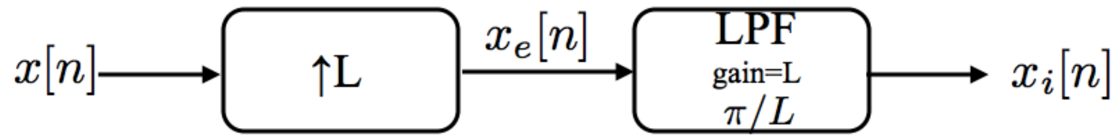
$x[n]$
↑L
$x_e[n]$
LPF
gain=L
π/L
$x_i[n]$
$X_c(j\Omega)$
1
Ω_N
Ω
sampling T
sampling T'=T/L
$X(e^{j\omega})$
1/T
$-\pi$
π
$X_i(e^{j\omega})$
L/T
$-\pi$
$\frac{\pi}{L}$
π
expanding L
$X_e(e^{j\omega})$
$-\pi$
π

Example

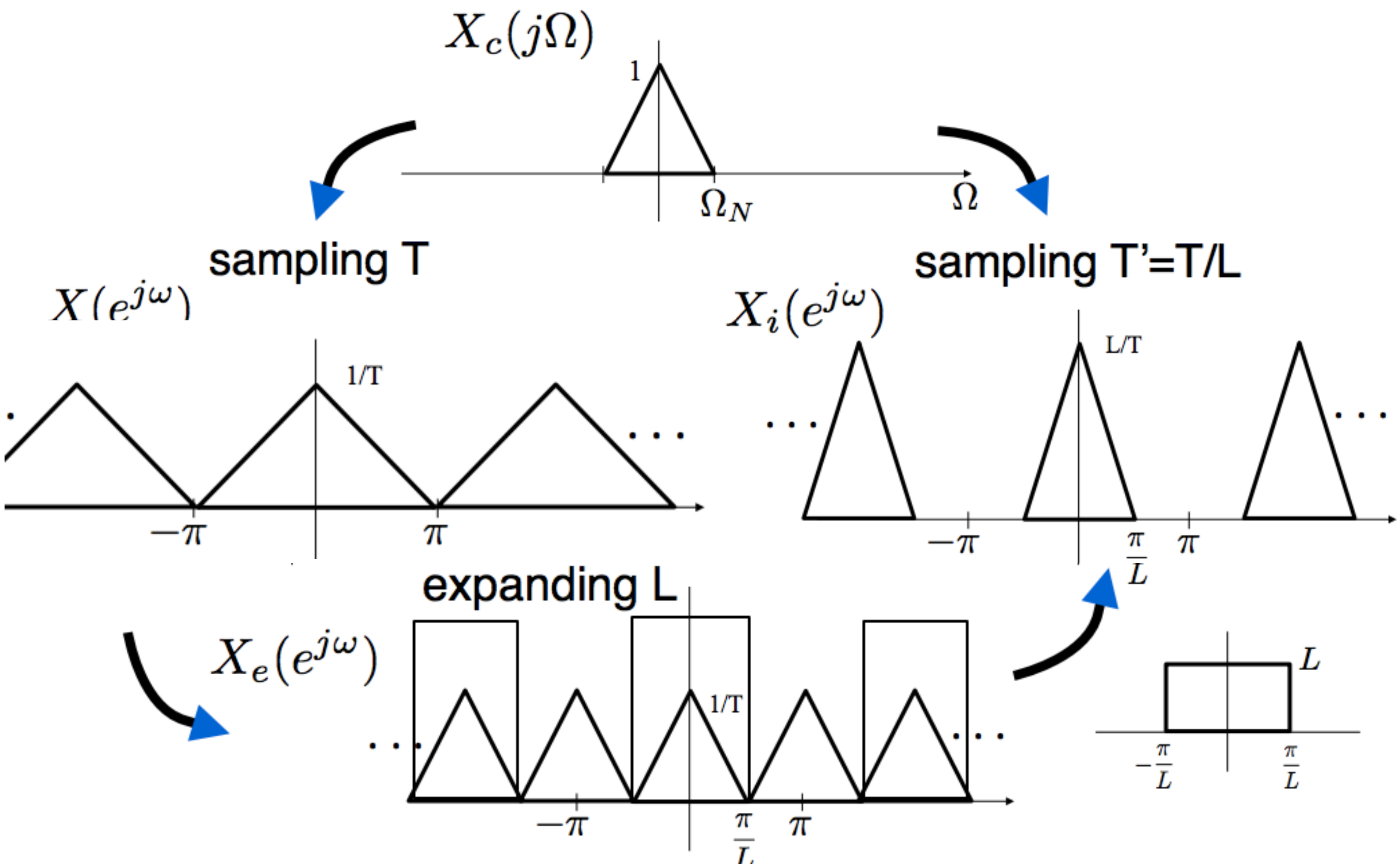


$x[n]$
↑L
$x_e[n]$
LPF
gain=L
π/L
$x_i[n]$
$X_c(j\Omega)$
1
Ω_N
Ω
sampling T
sampling T'=T/L
$X(e^{j\omega})$
1/T
$-\pi$
π
$X_i(e^{j\omega})$
L/T
$-\pi$
$\frac{\pi}{L}$
π
expanding L
$X_e(e^{j\omega})$
$= X(e^{j\omega L})$
1/T
$-\pi$
$\frac{\pi}{L}$
π

Example



x[n]
↑L
x_e[n]
LPF
gain=L
π/L
x_i[n]

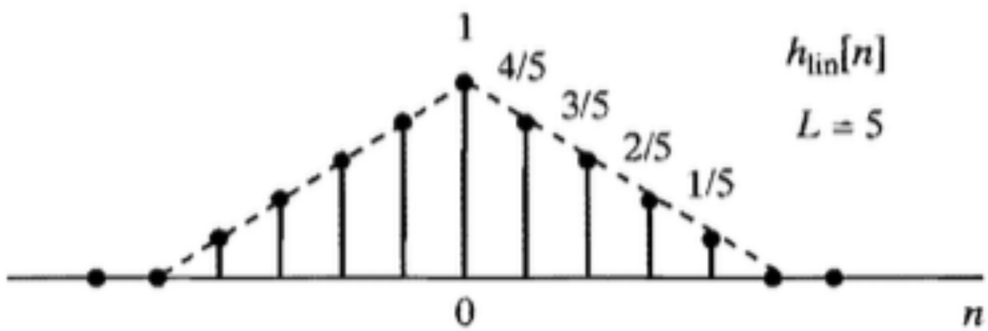


X_c(jΩ)
1
Ω_N
Ω
sampling T
X(e^{jω})
1/T
−π
π
sampling T'=T/L
X_i(e^{jω})
L/T
π/L
expanding L
X_e(e^{jω})
L
−π/L

Practical Interpolation

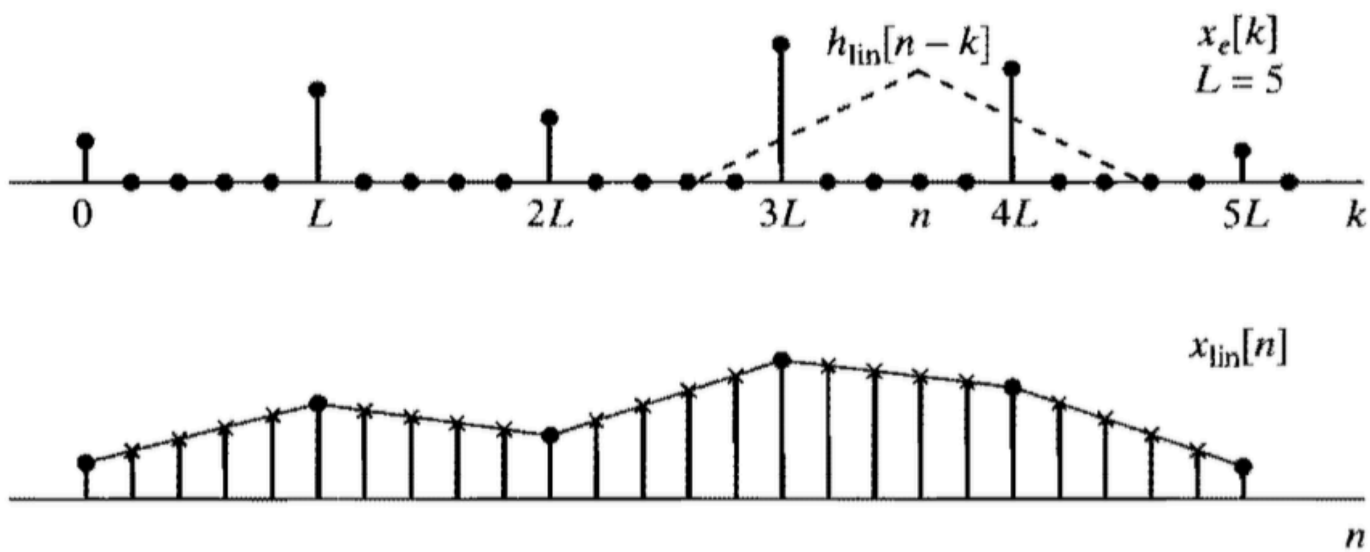
- Interpolate with simple, practical filters
  - Linear interpolation – samples between original samples fall on a straight line connecting the samples
    - Convolve with triangle instead of sinc

$$h_{\text{lin}}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise}, \end{cases}$$



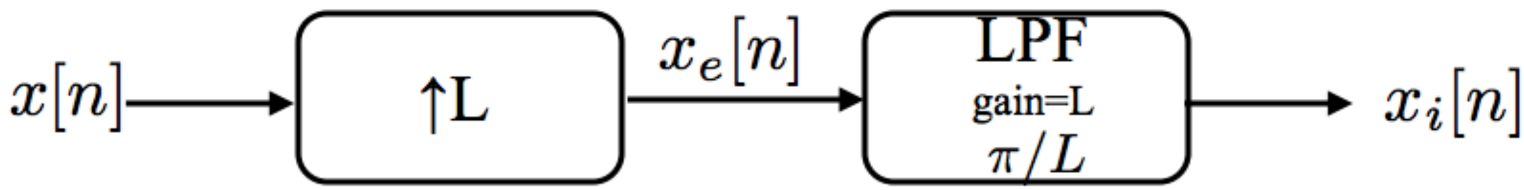
Practical Interpolation

- Interpolate with simple, practical filters
  - Linear interpolation – samples between original samples fall on a straight line connecting the samples
    - Convolve with triangle instead of sinc

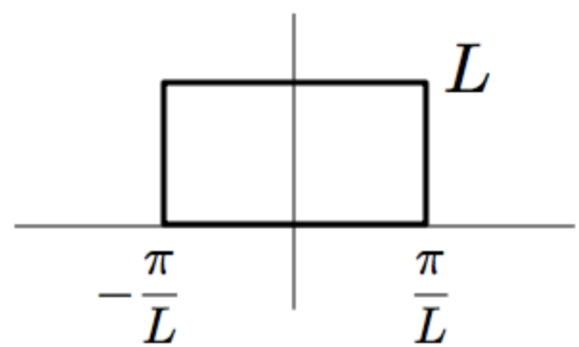


Frequency Domain Interpretation

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

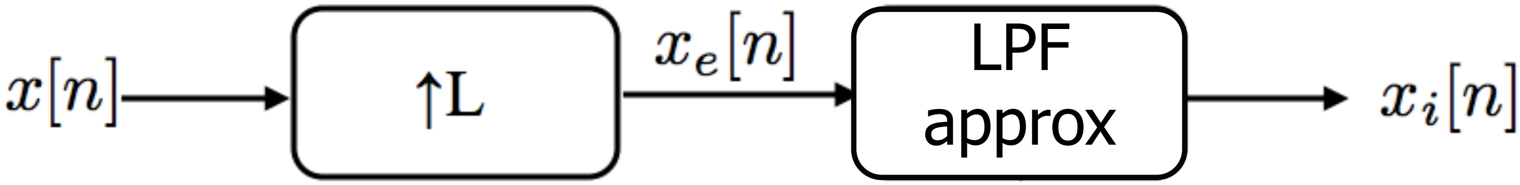


$\text{sinc}(n/L)$ **DTFT ⇒**

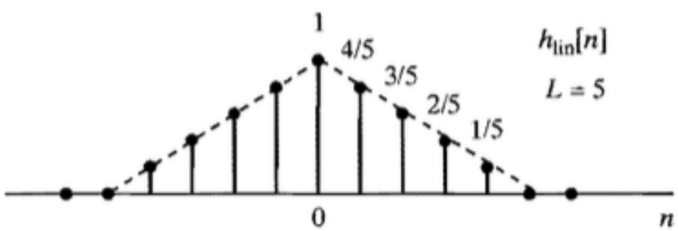


Linear Interpolation -- Frequency Domain

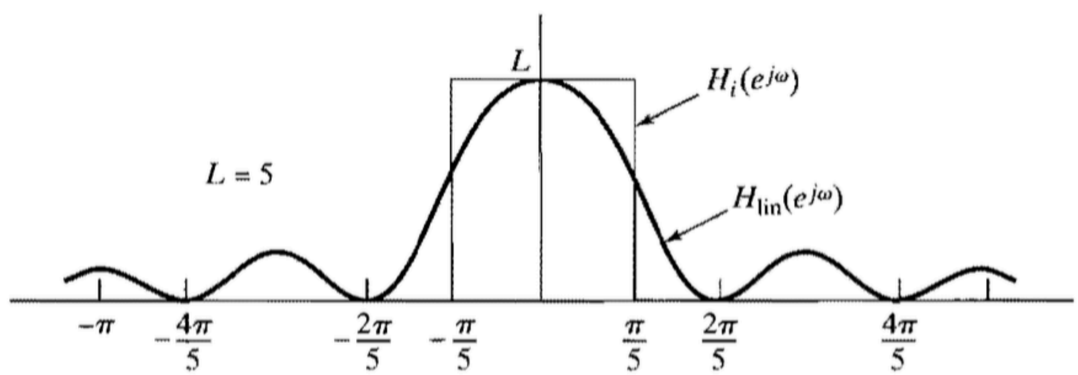
$$x_i[n] = x_e[n] * h_{lin}[n]$$



$$h_{\text{lin}}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise,} \end{cases}$$

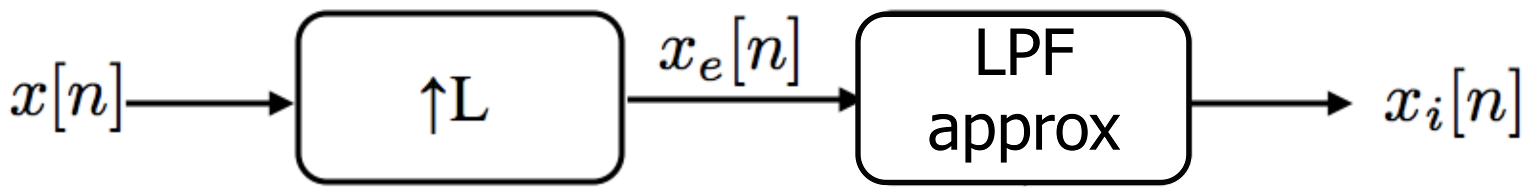


DTFT ⇒

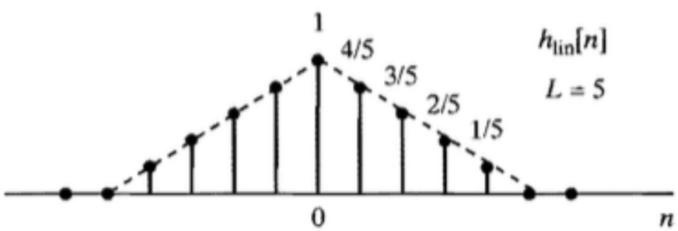


Linear Interpolation -- Frequency Domain

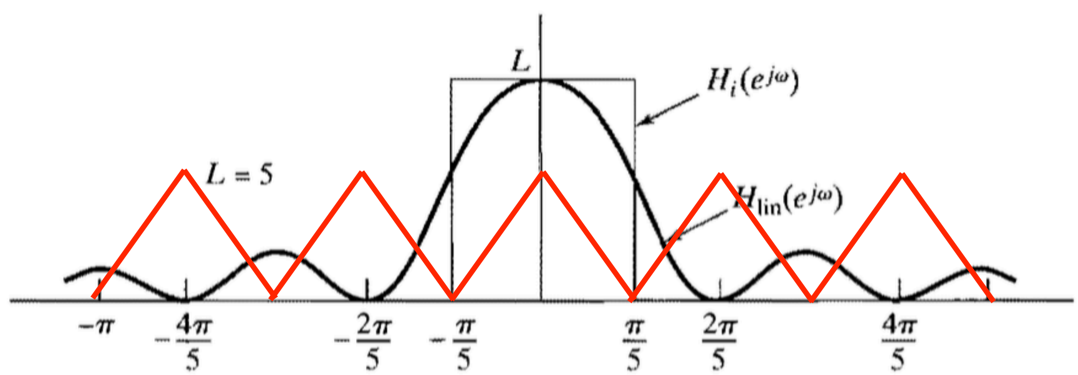
$$x_i[n] = x_e[n] * h_{lin}[n]$$



$$h_{\mathrm{lin}}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise,} \end{cases}$$

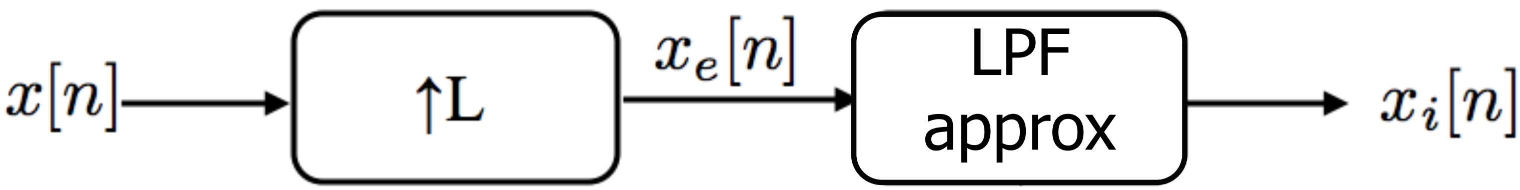


DTFT ⇒

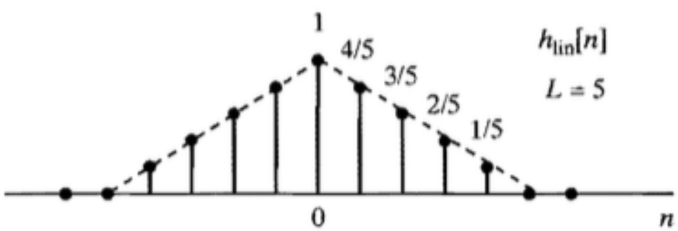


Linear Interpolation -- Frequency Domain

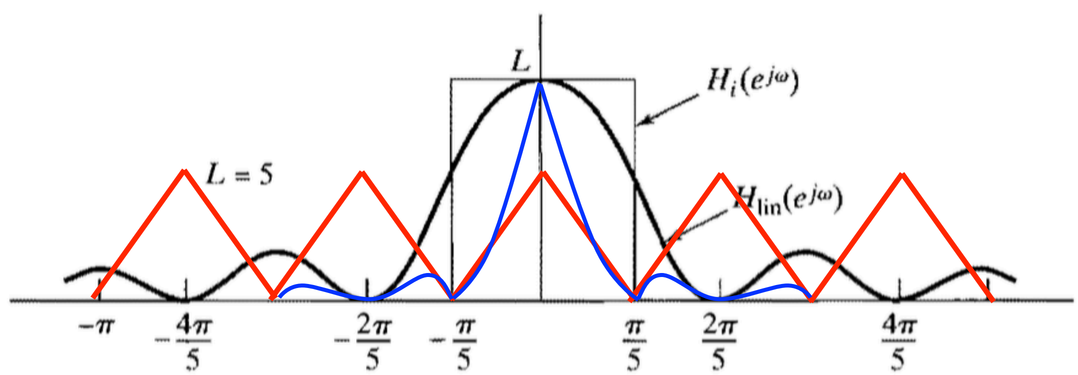
$$x_i[n] = x_e[n] * h_{lin}[n]$$



$$h_{\text{lin}}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise}, \end{cases}$$

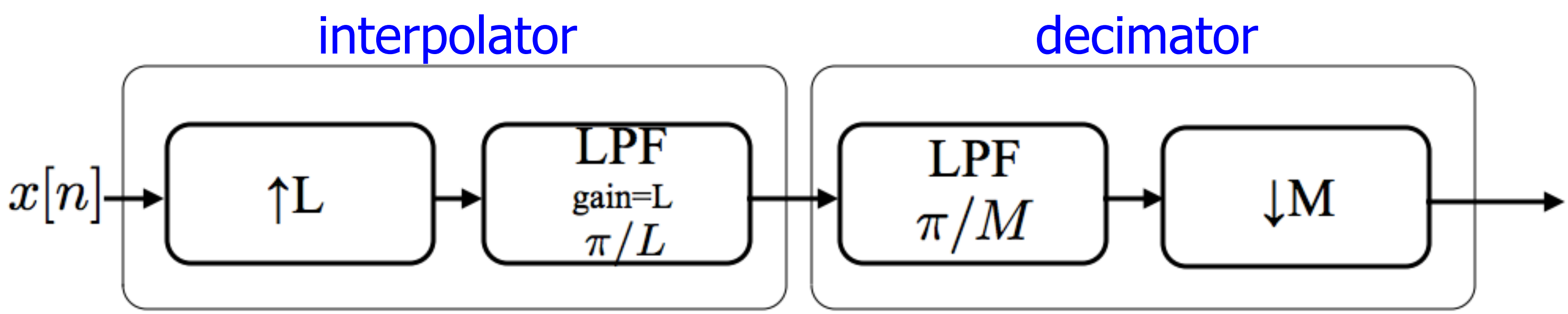


DTFT ⇒



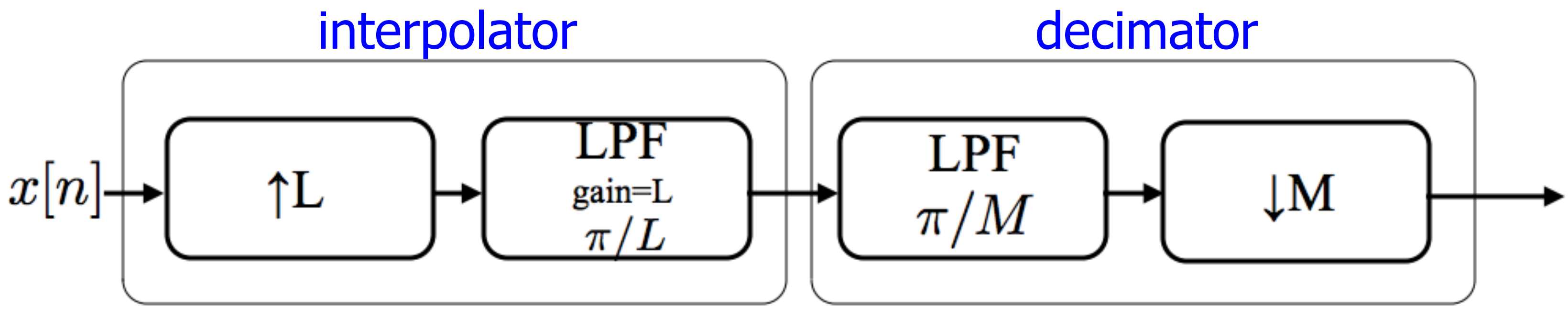
Non-integer Sampling

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

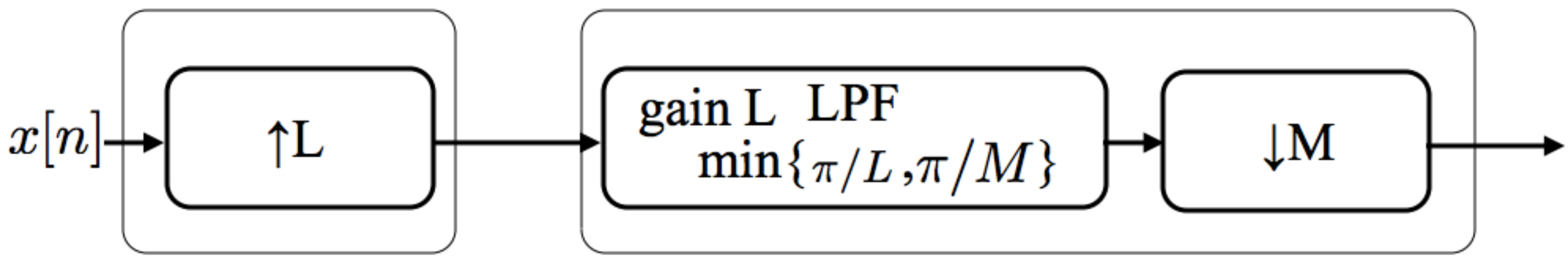


Non-integer Sampling

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

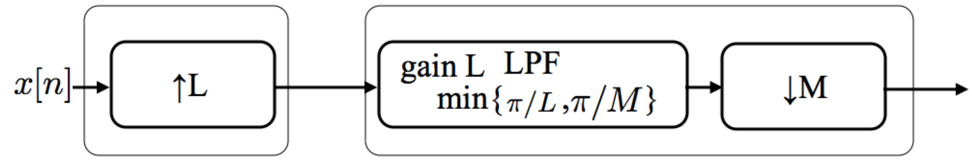


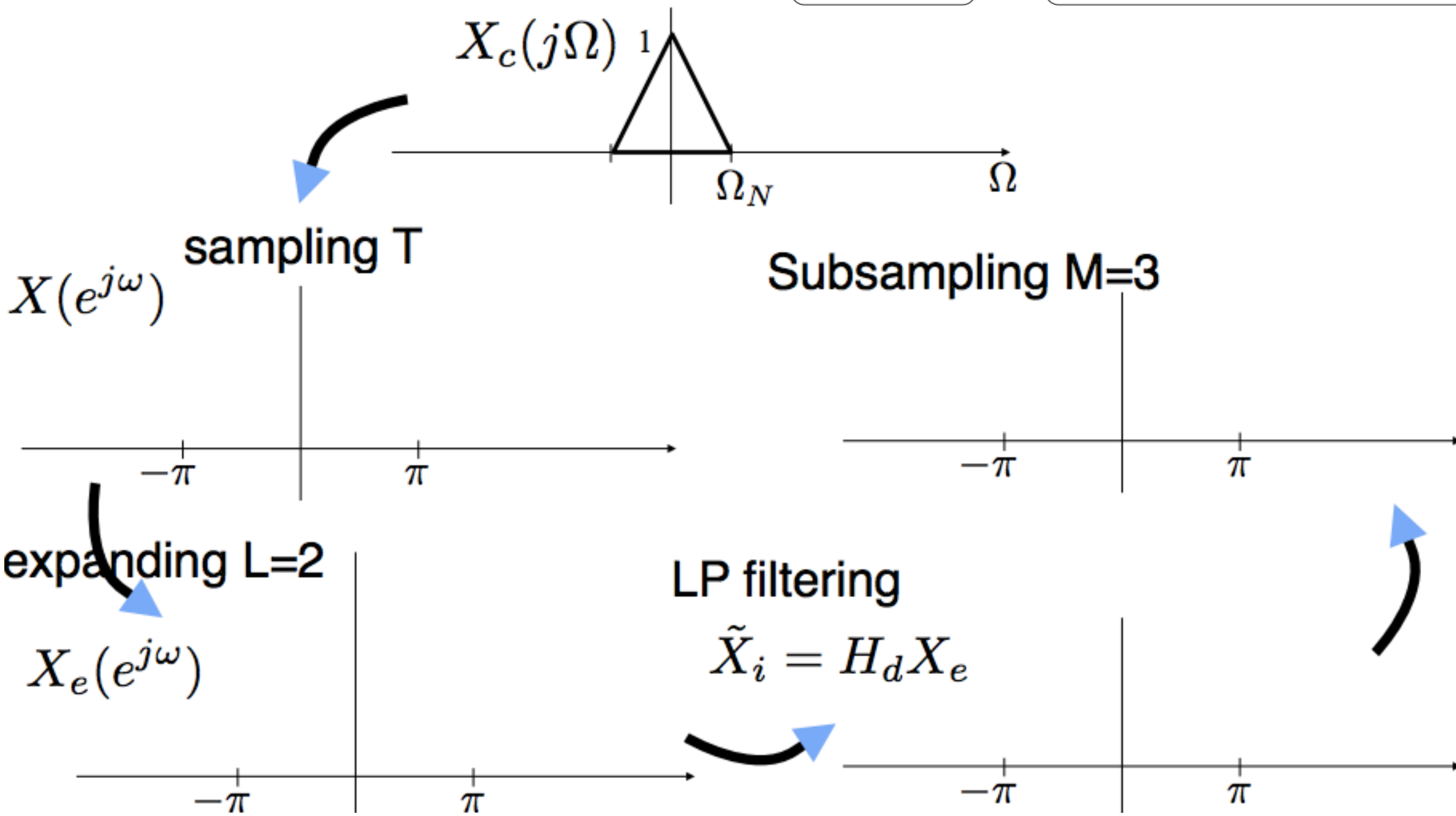
Or,



Example

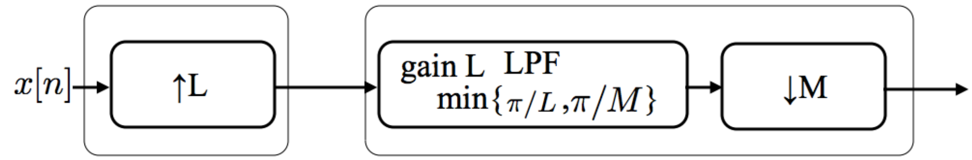
- T'=3/2T → L=2, M=3

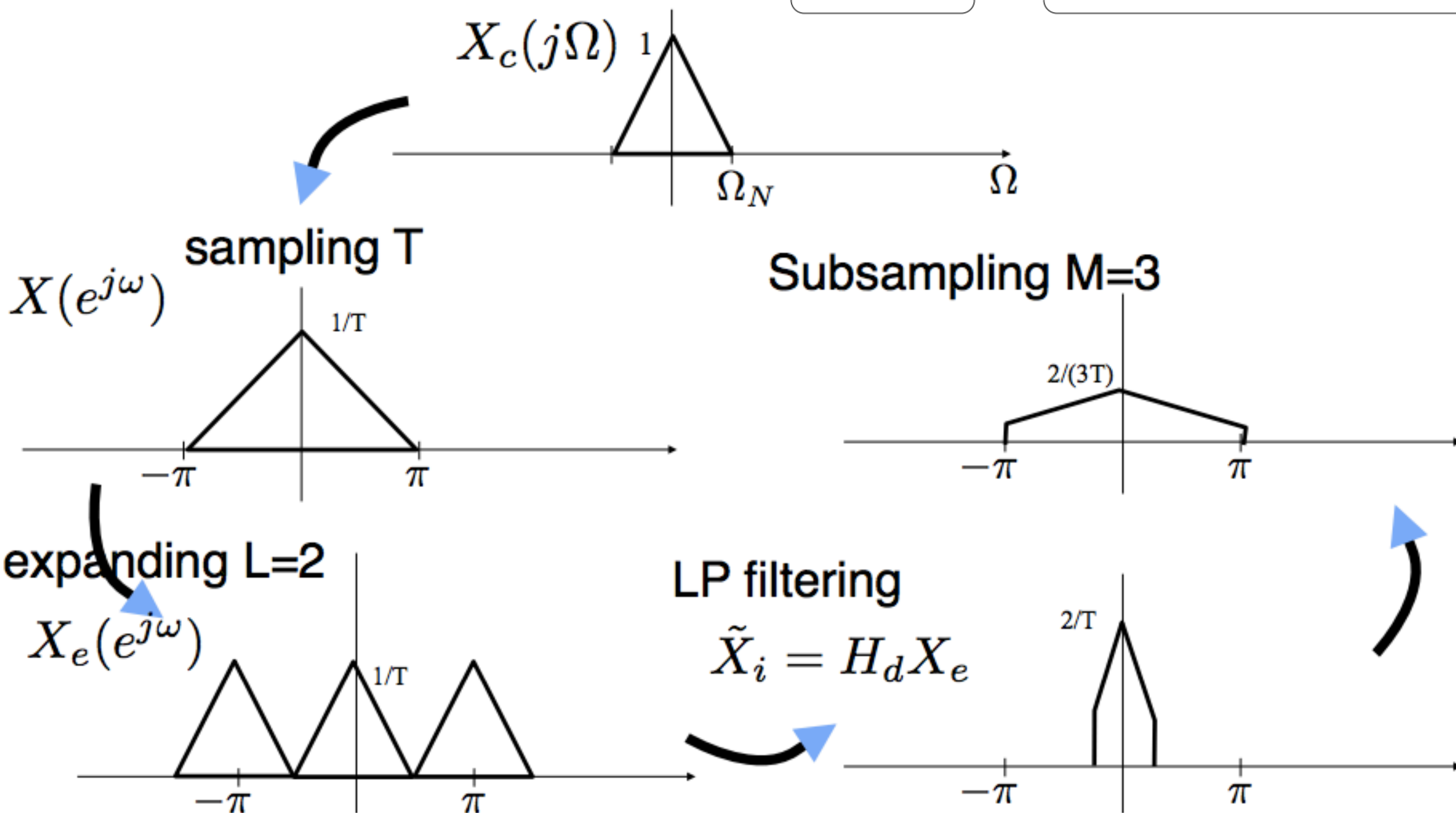




Example

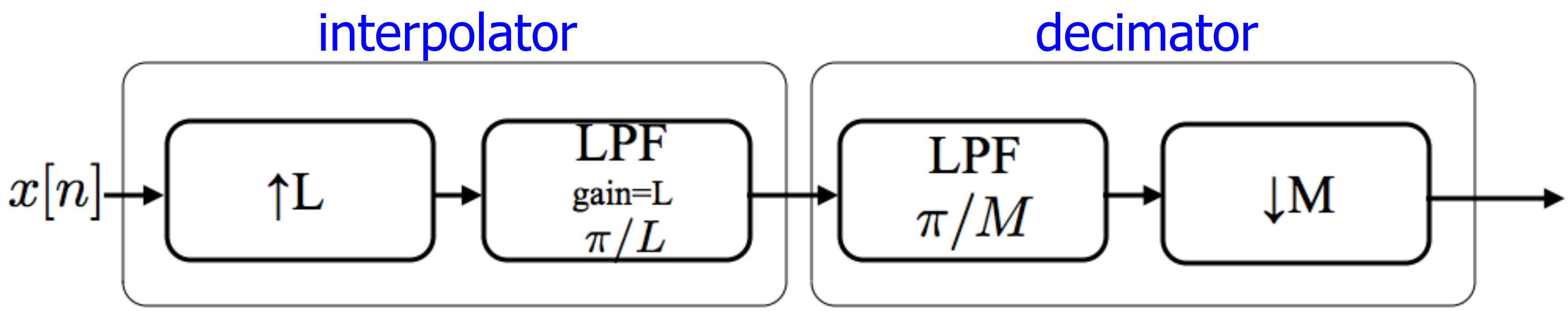
- T'=3/2T → L=2, M=3



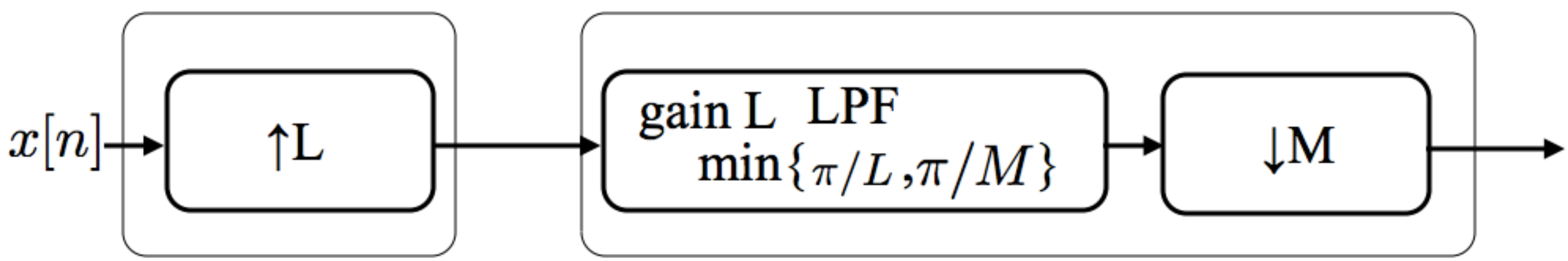


Non-integer Sampling

- T'=TM/L
  - Downsample by M, then upsample by L?



Or,

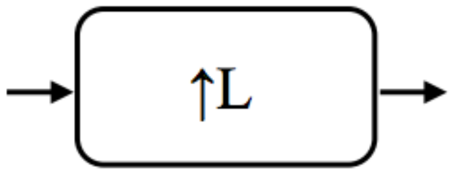


Multi-Rate Signal Processing

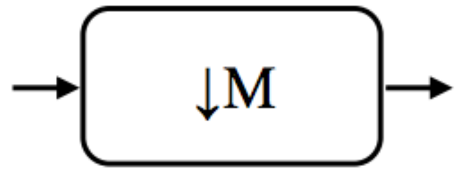
- What if we want to resample by 1.01T?
  - Expand by L=100
  - Filter $\pi/101$ ($$$$$)
  - Downsample by M=101

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

Interchanging Operations



"expander"



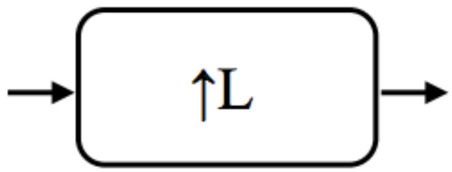
"compressor"

<u>not LTI!</u>

Upsampling
-expanding in time
-compressing in frequency

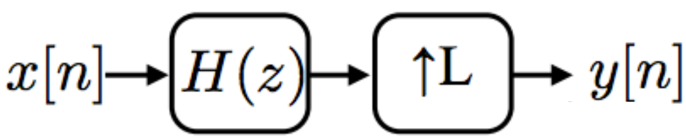
Downsampling
-compressing in time
-expanding in frequency

Interchanging Operations - Expander

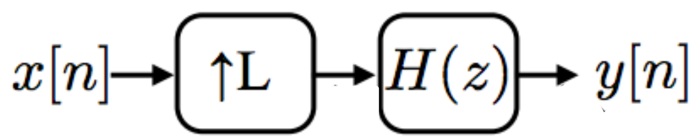


"expander"

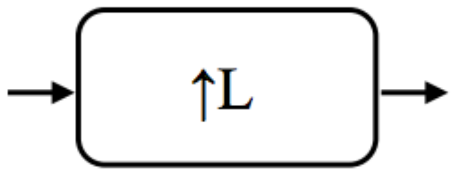
Upsampling
-expanding in time
-compressing in frequency



?

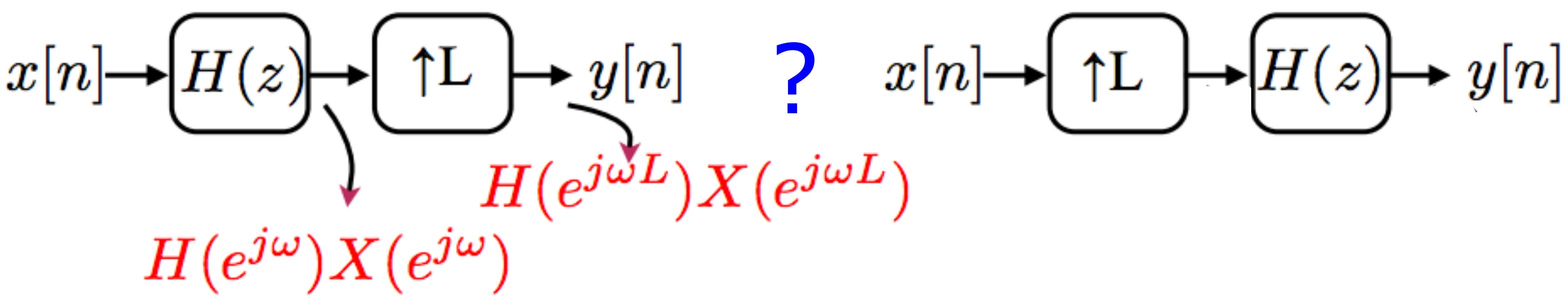


Interchanging Operations - Expander

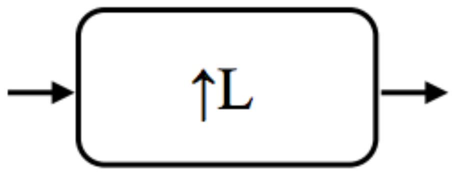


"expander"

Upsampling
-expanding in time
-compressing in frequency

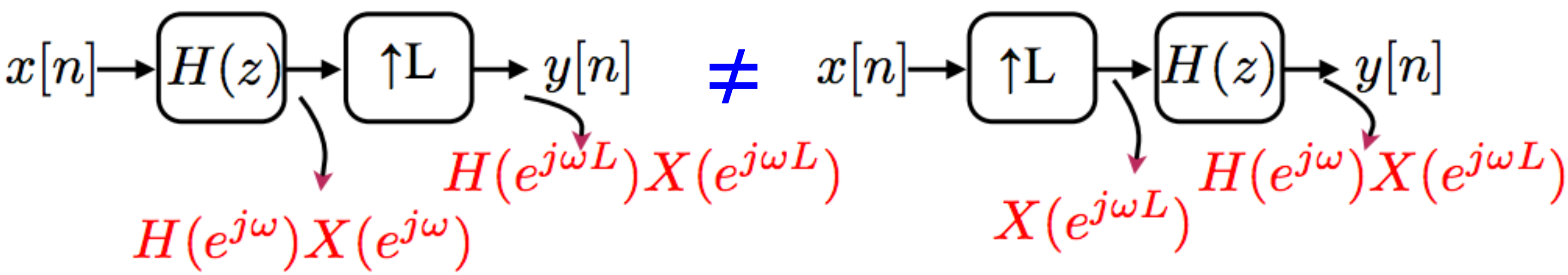


Interchanging Operations - Expander

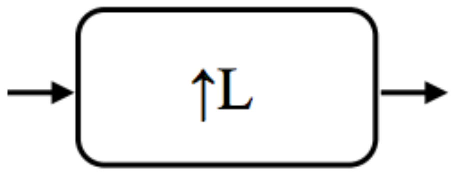


“expander”

Upsampling
-expanding in time
-compressing in frequency

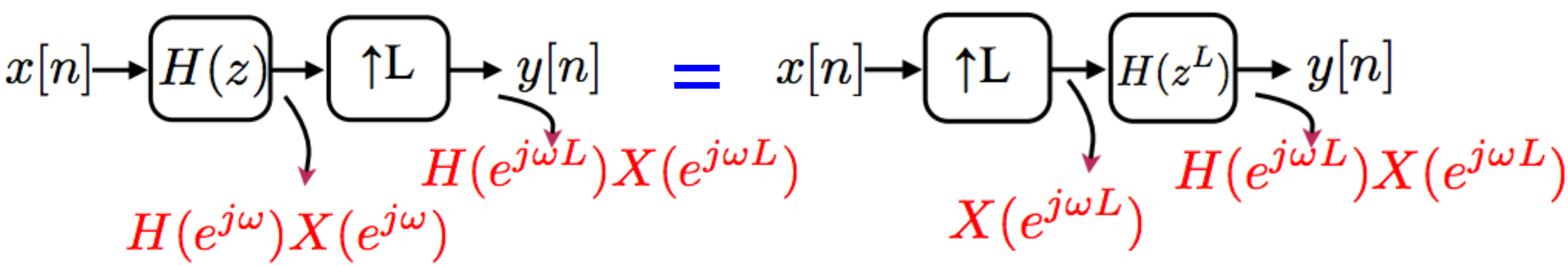


Interchanging Operations - Expander

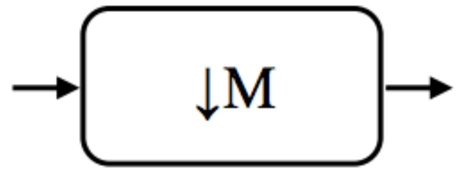


“expander”

Upsampling
-expanding in time
-compressing in frequency



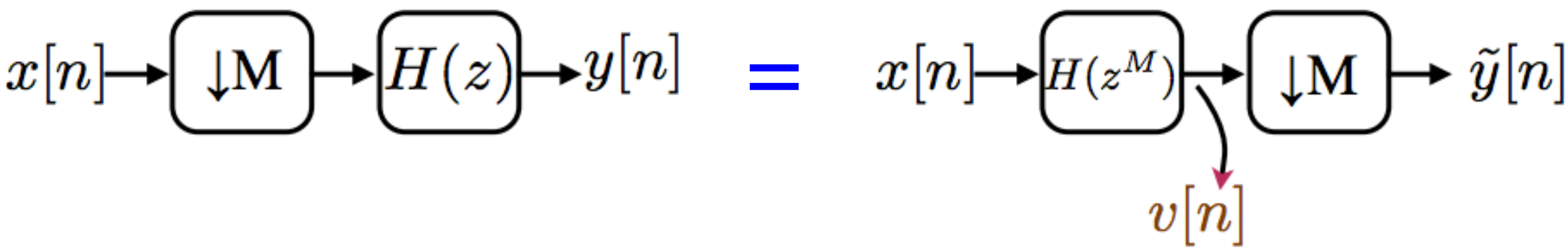
Interchanging Operations - Compressor



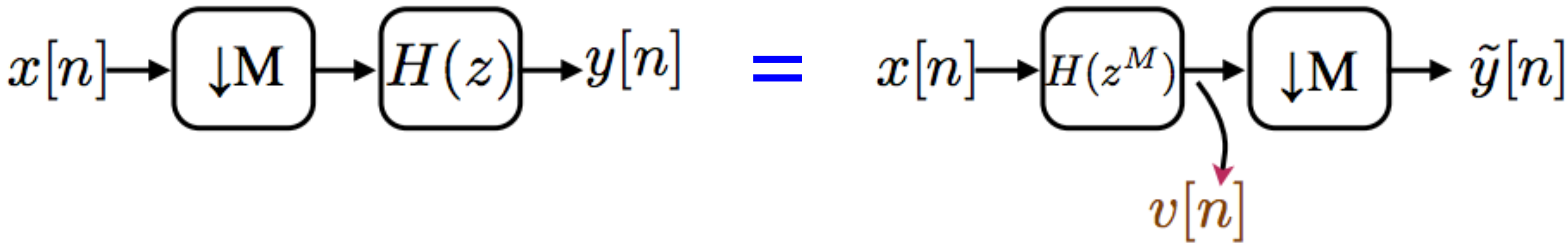
not LTI!

“compressor”

Downsampling
-compressing in time
-expanding in frequency

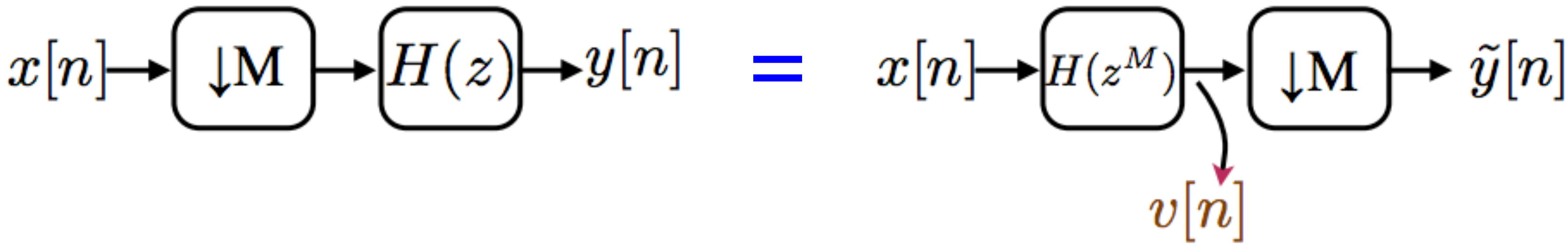


Interchanging Operations - Compressor



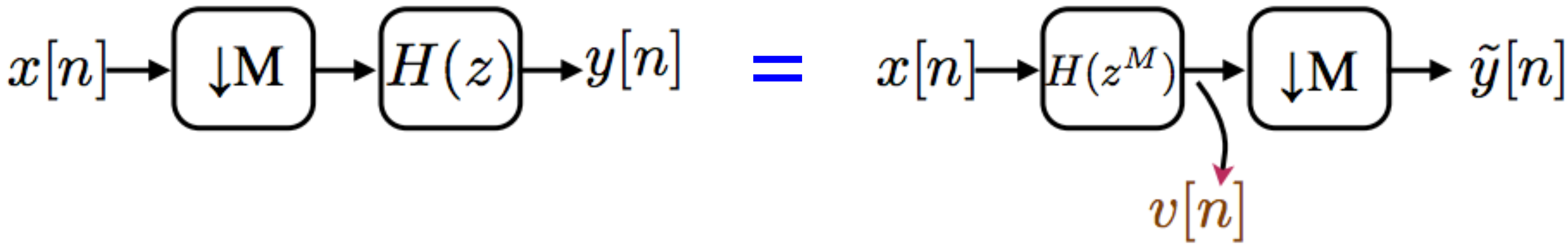
$$Y(e^{j\omega}) = H(e^{j\omega})\left(\frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j\left(\frac{\omega}{M}-\frac{2\pi i}{M}\right)}\right)\right)$$

Interchanging Operations - Compressor



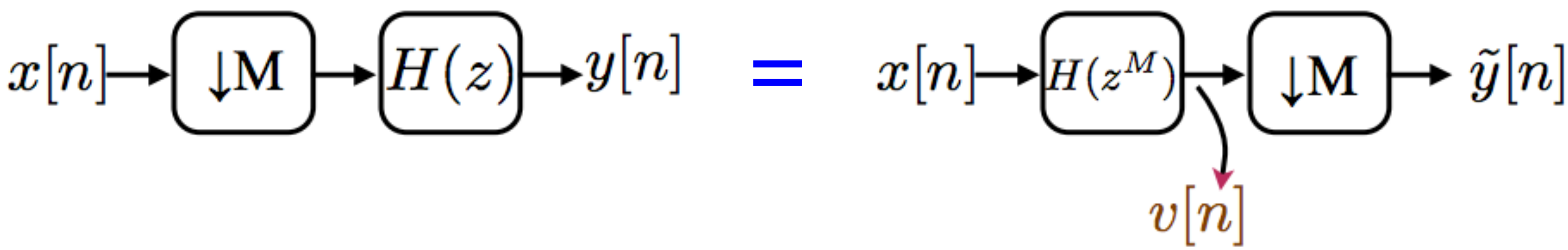
$$
\begin{aligned}
Y(e^{j\omega}) &= H(e^{j\omega}) \left(\frac{1}{M} \sum_{i=0}^{M-1} X\left(e^{j\left(\frac{\omega}{M} - \frac{2\pi i}{M}\right)}\right) \right) \\
&= \frac{1}{M} \sum_{i=0}^{M-1} \underbrace{H\left(e^{j(\omega - 2\pi i)}\right)}_{H(e^{j\omega})} X\left(e^{j\left(\frac{\omega}{M} -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

Interchanging Operations - Compressor



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

Interchanging Operations - Compressor



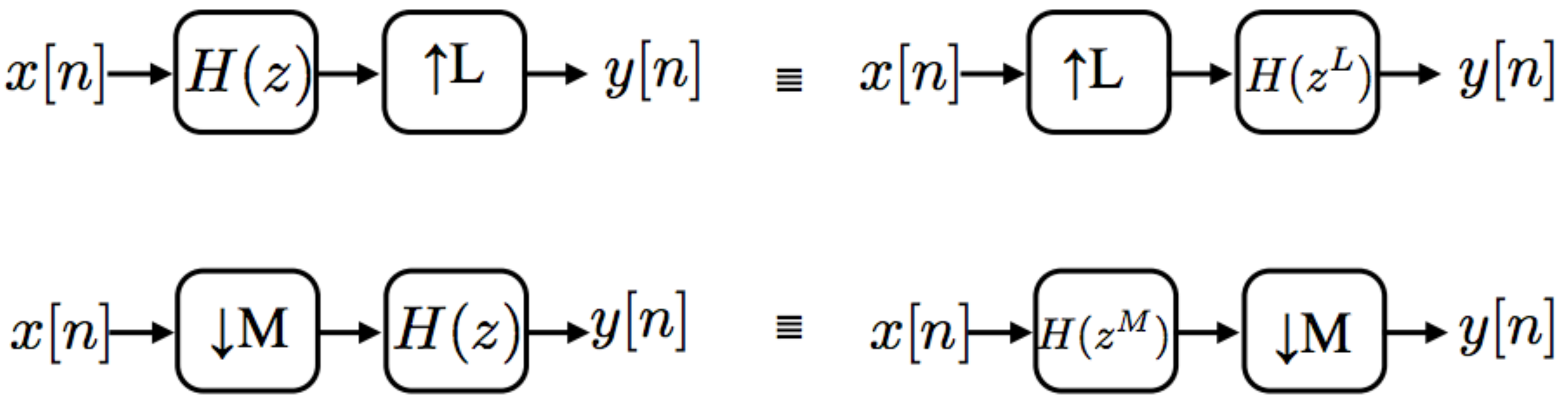
$$
\begin{aligned}
Y(e^{j\omega}) &= H(e^{j\omega})\left(\frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j\left(\frac{\omega}{M}-\frac{2\pi i}{M}\right)}\right)\right)\\
&= \frac{1}{M}\sum_{i=0}^{M-1} \underbrace{H\left(e^{j(\omega-2\pi i)}\right)}_{H(e^{j\omega})} X\left(e^{j\left(\frac{\omega}{M}-\frac{2\pi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

After compressing

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

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

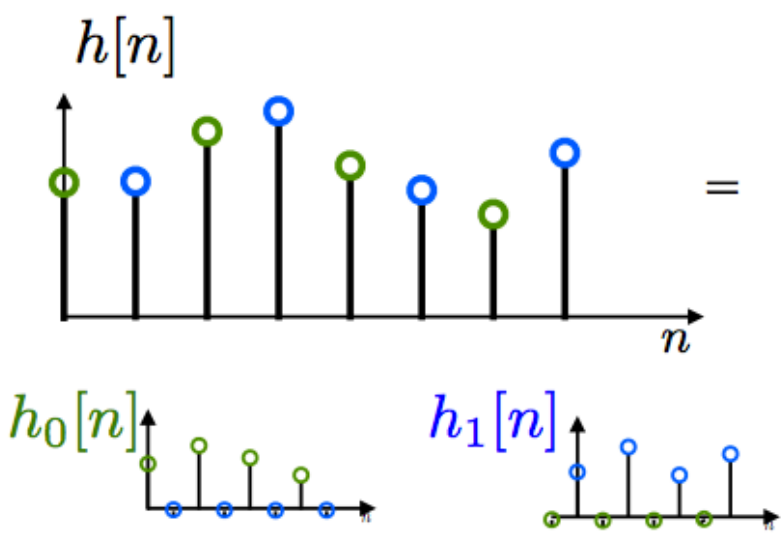
- What if we want to resample by 1.01T?
  - Expand by L=100
  - Filter $\pi/101$ ($$$$$)
  - Downsample by M=101

- 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:

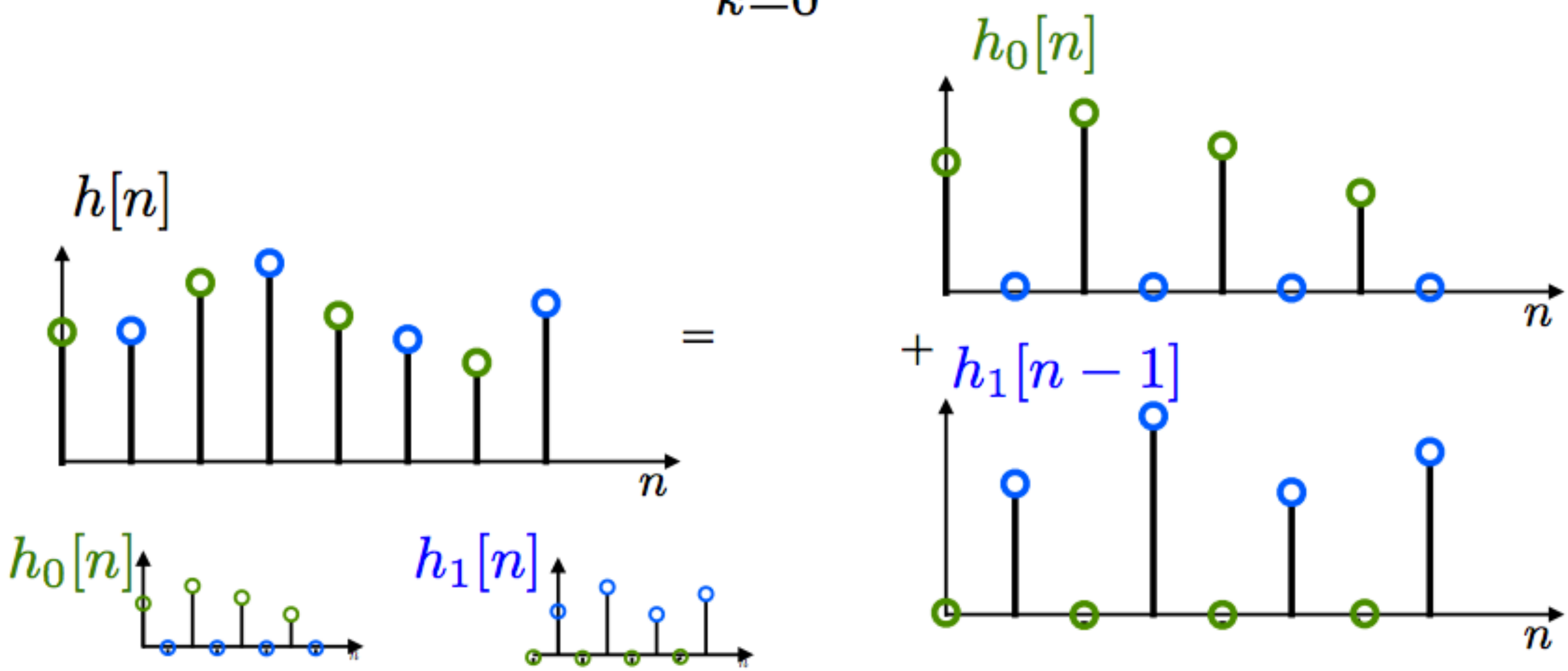
$$h[n] = \sum_{k=0}^{M-1} h_k[n-k]$$



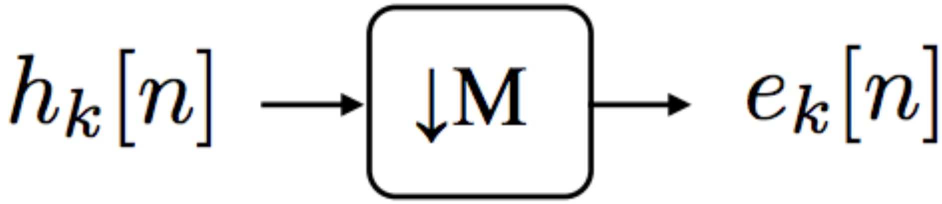
Polyphase Decomposition

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

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



Polyphase Decomposition

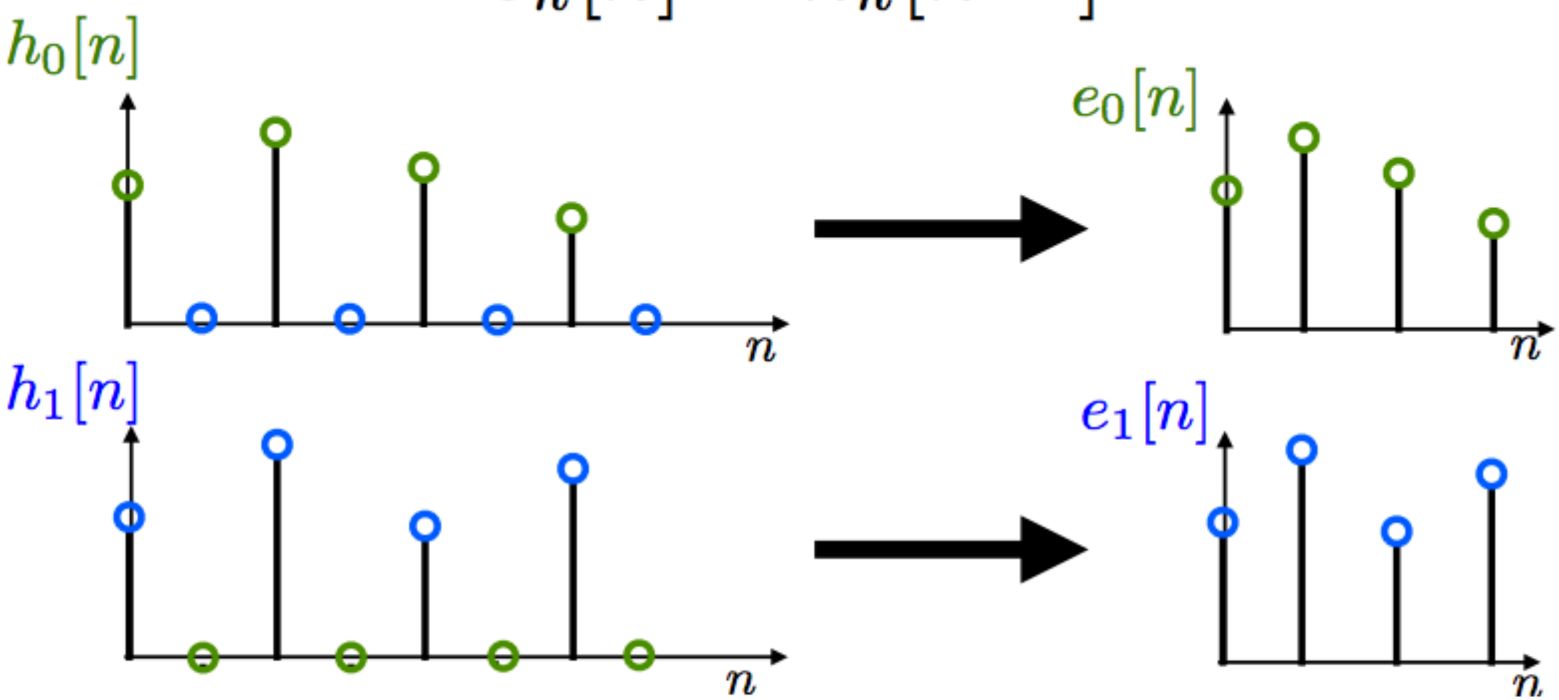


$$e_k[n] = h_k[nM]$$

Polyphase Decomposition

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

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



Polyphase Decomposition

$$e_k[n] \longrightarrow \boxed{\uparrow \mathrm{M}} \longrightarrow h_k[n]$$

recall upsampling ⇒ scaling

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

Polyphase Decomposition

$$e_k[n] \longrightarrow \boxed{\uparrow \mathrm{M}} \longrightarrow h_k[n]$$

recall upsampling ⇒ scaling

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

Also, recall:

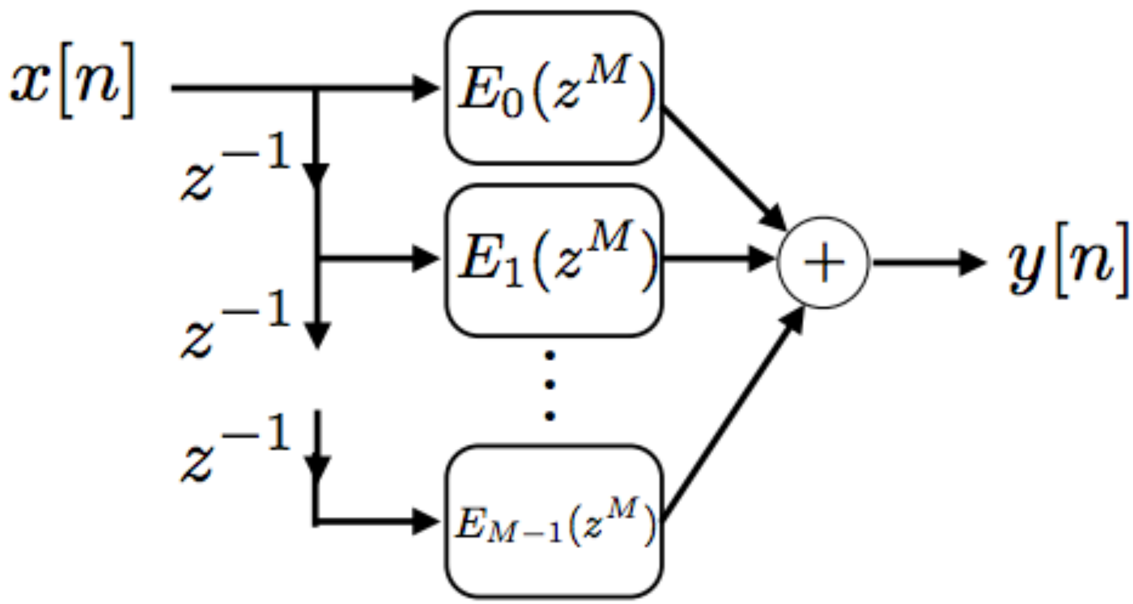
$$h[n] = \sum_{k=0}^{M-1} h_k[n-k]$$

So,

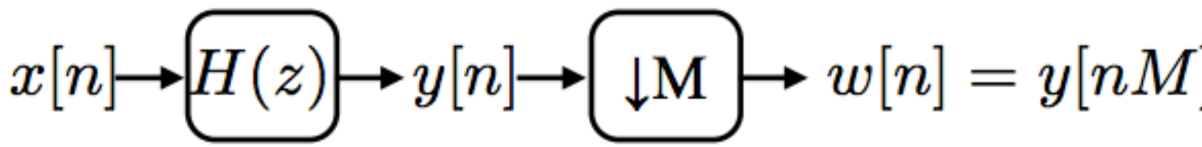
$$\boxed{H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k}}$$

Polyphase Decomposition

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



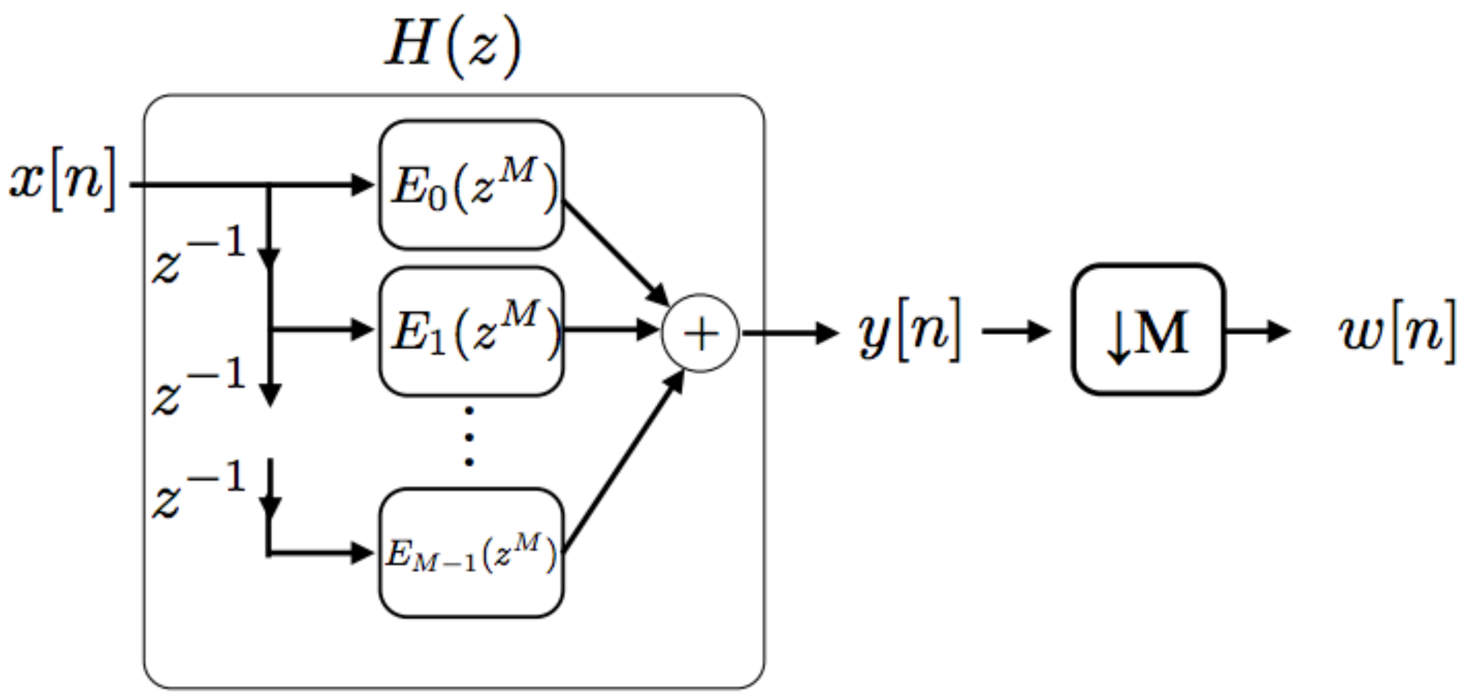
Polyphase Implementation of Decimation



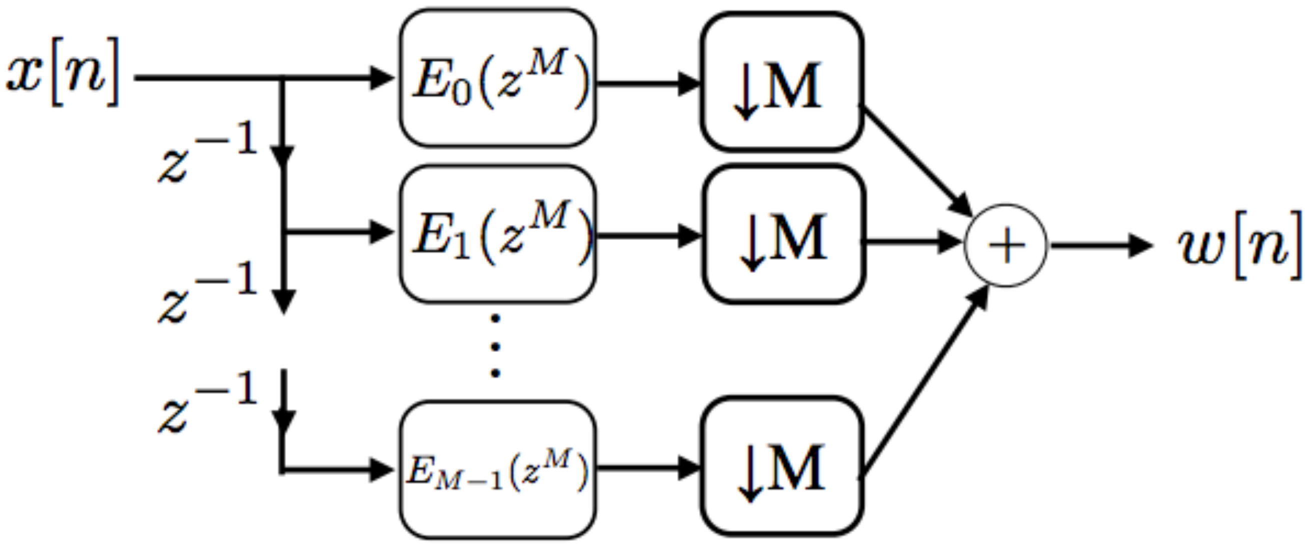
- Problem:
  - Compute all y[n] and then throw away -- wasted computation!
  - For FIR length N → N mults/unit time

Polyphase Implementation of Decimation

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



Polyphase Implementation of Decimation

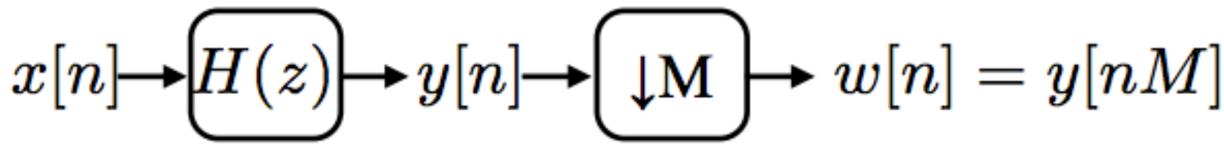


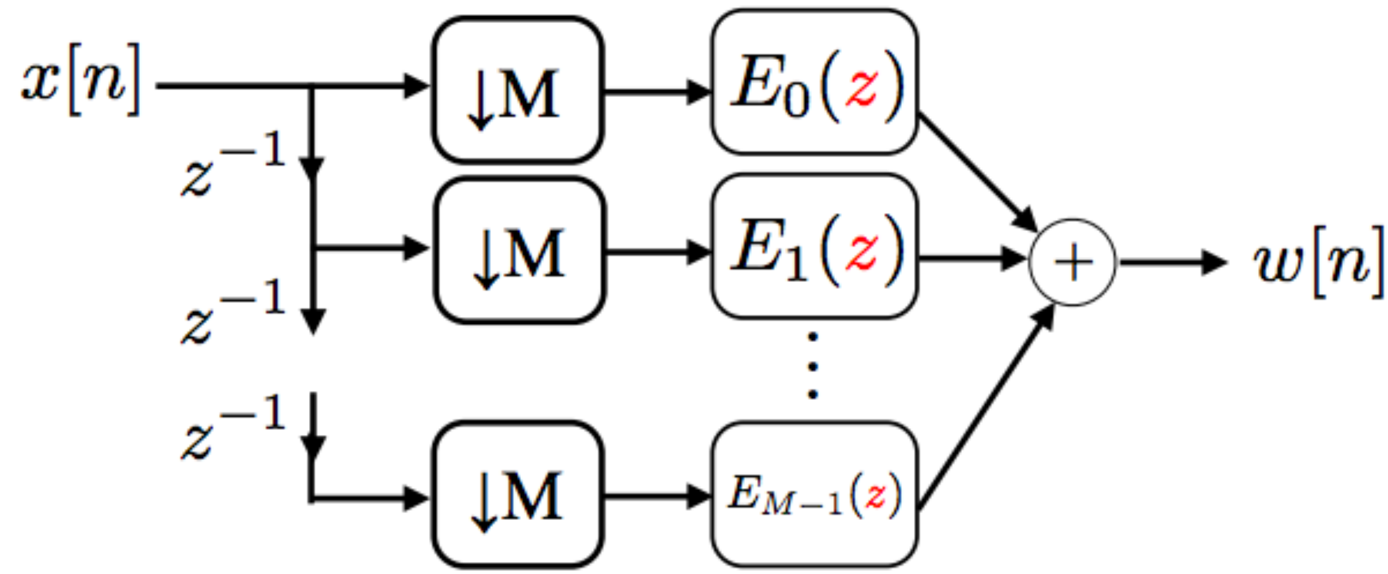
Interchanging Operations - Summary

Filter and expander　　　　Expander and expanded filter

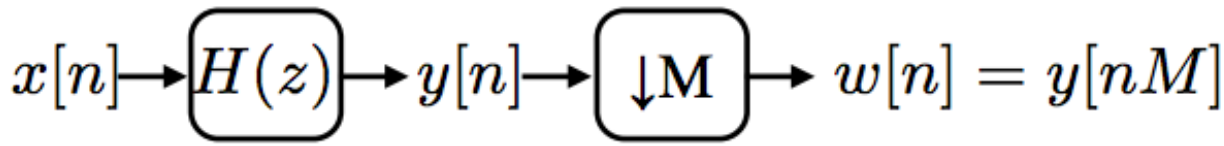
Compressor and filter　　　　Expanded filter and compressor

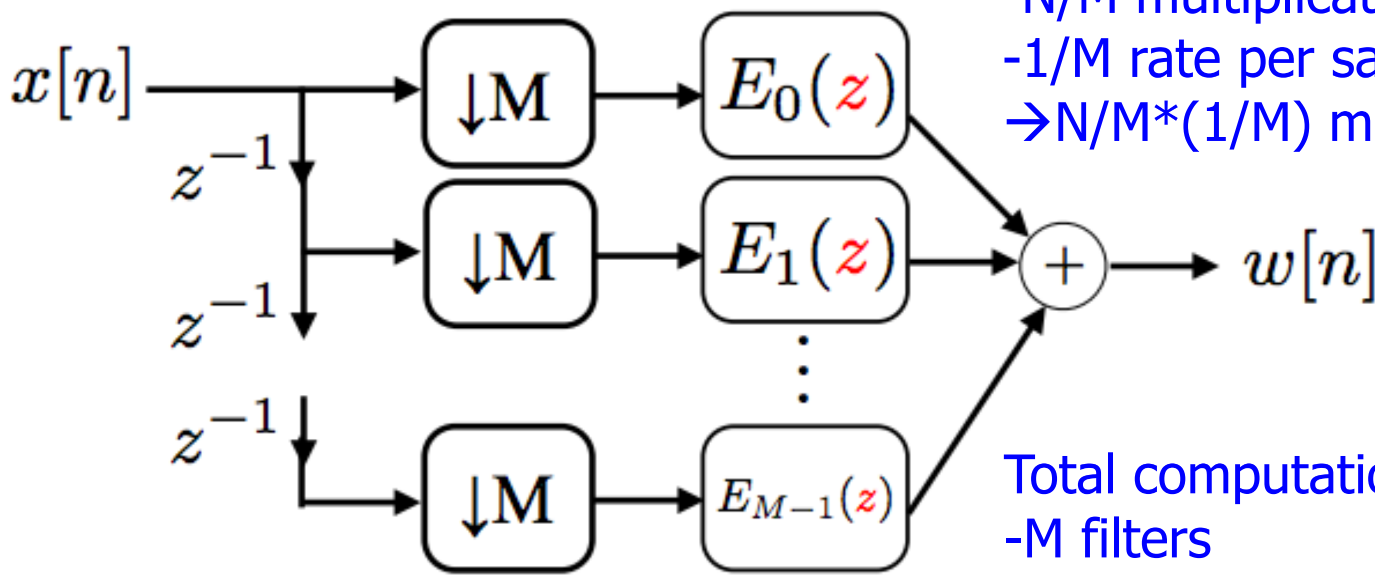
Polyphase Implementation of Decimation





Polyphase Implementation of Decimation





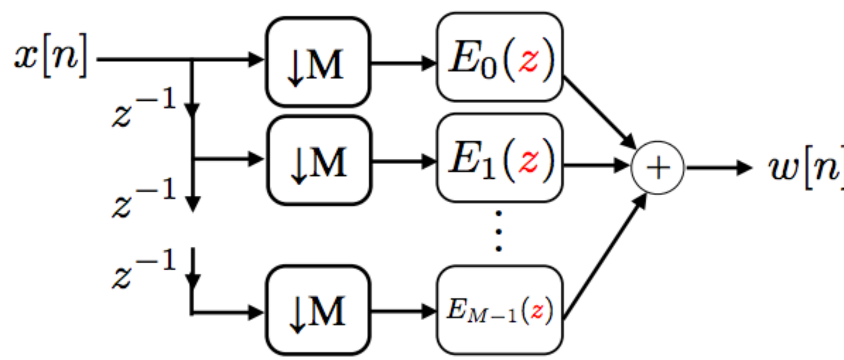
Each filter computation:
-N/M multiplications
-1/M rate per sample
→N/M*(1/M) mults/unit time

Total computation:
-M filters
→N/M mults/unit time

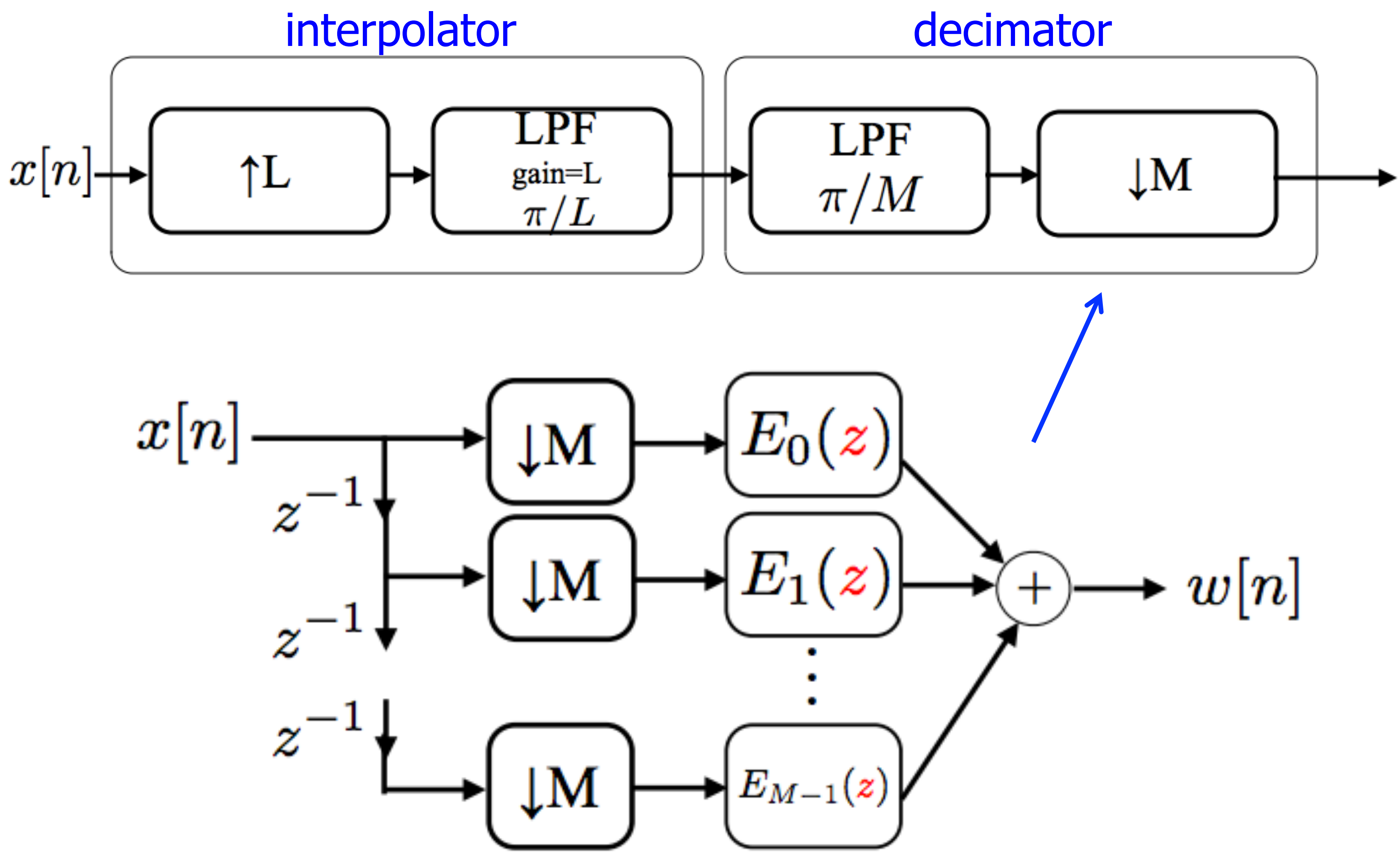
Multi-Rate Signal Processing

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

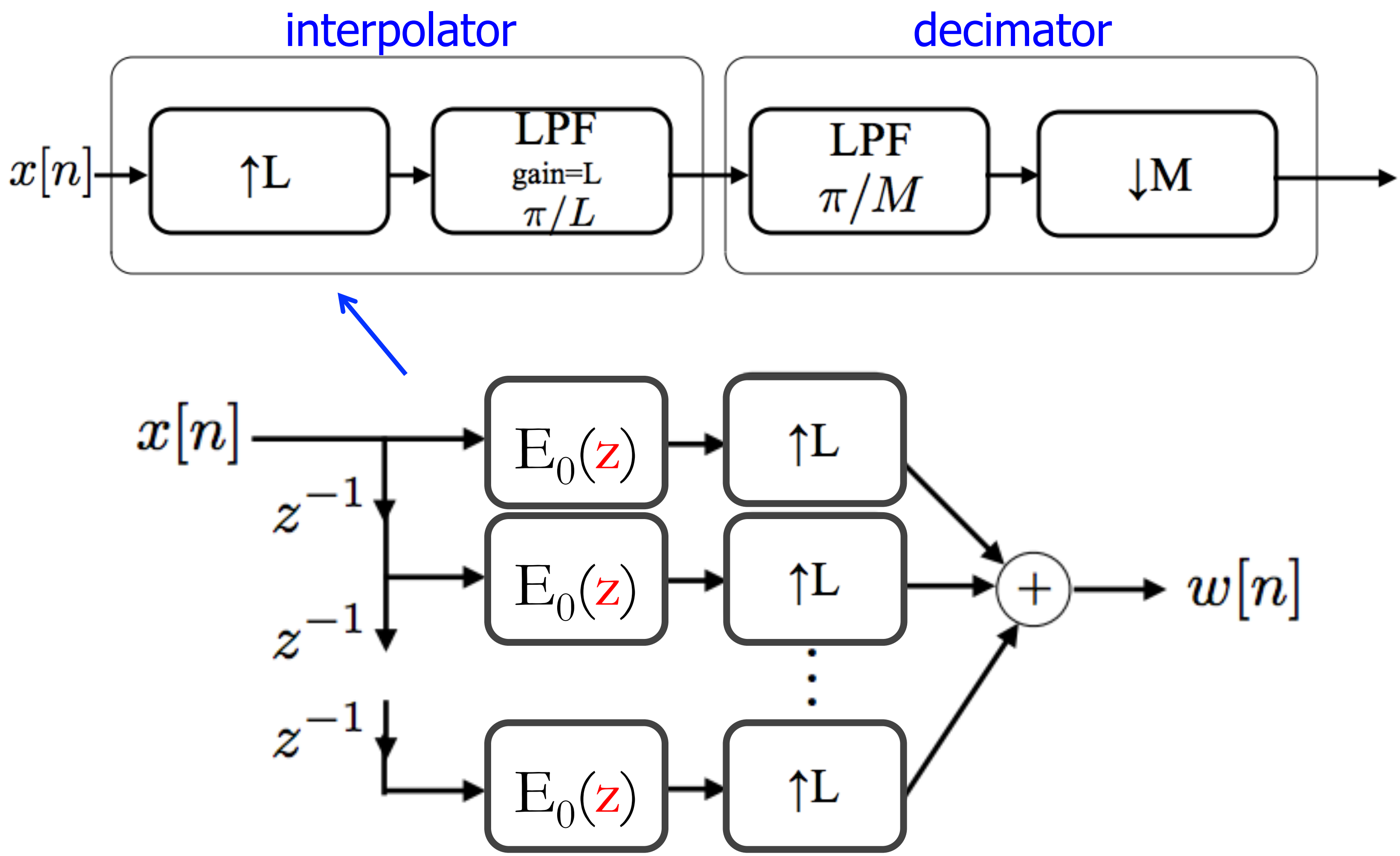
- Fortunately there are ways around it!
  - Called multi-rate
  - Uses compressors, expanders and filtering



Polyphase Implementation of Decimator



Polyphase Implementation of Interpolation



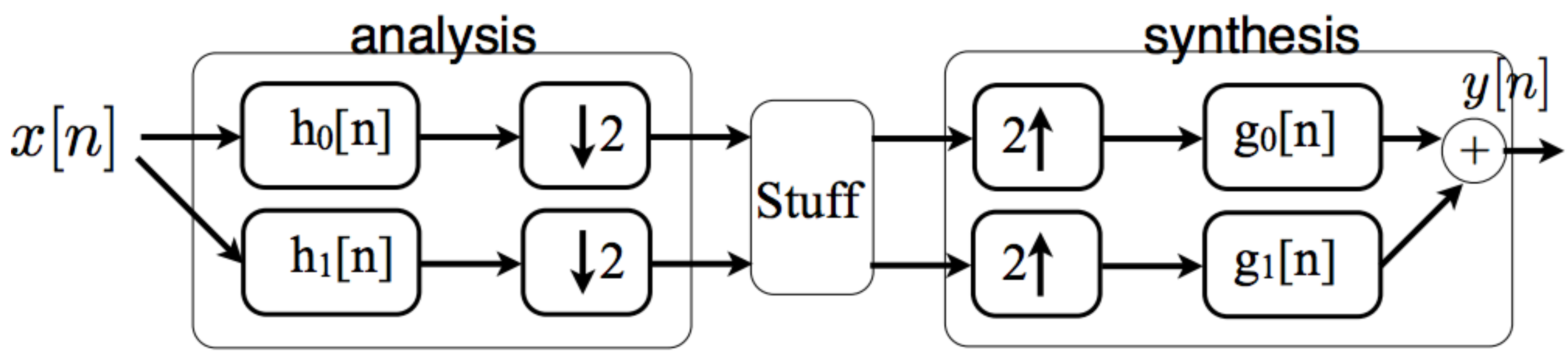
Multi-Rate Filter Banks

- 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

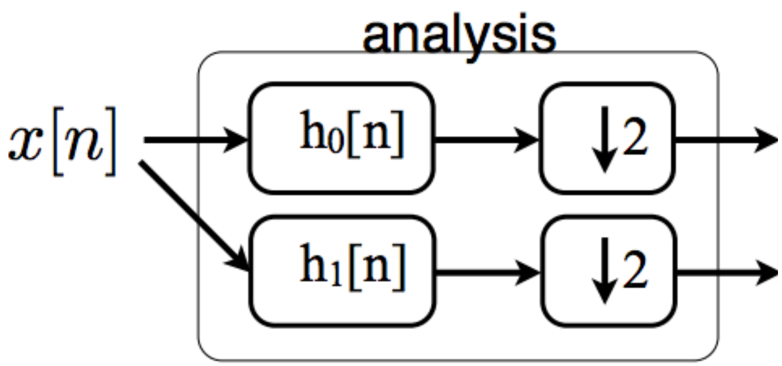
- Use filter banks to operate on a signal differently in different frequency bands
  - To save computation, reduce the rate after filtering

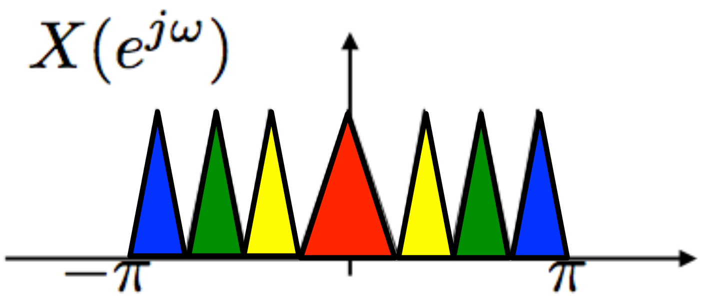
- $h_0[n]$ is low-pass, $h_1[n]$ is high-pass
  - Often $h_1[n]=e^{j\pi n}h_0[n]$ ← shift freq resp by π



Multi-Rate Filter Banks

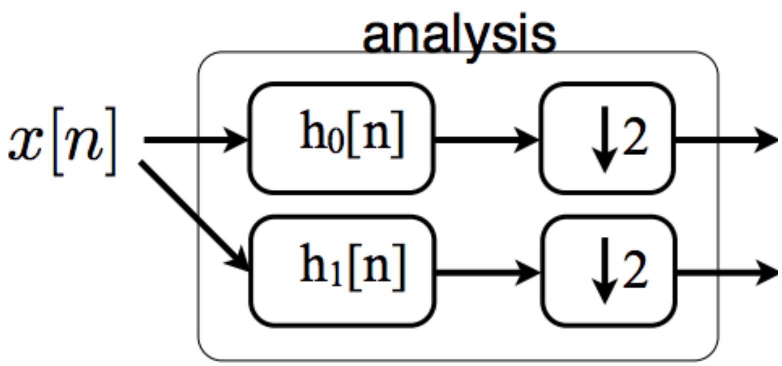
- Assume h_0, h_1 are ideal low/high pass

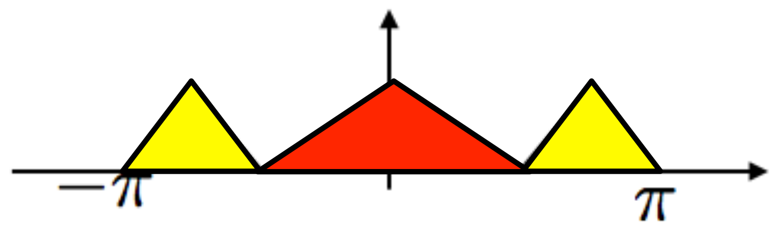


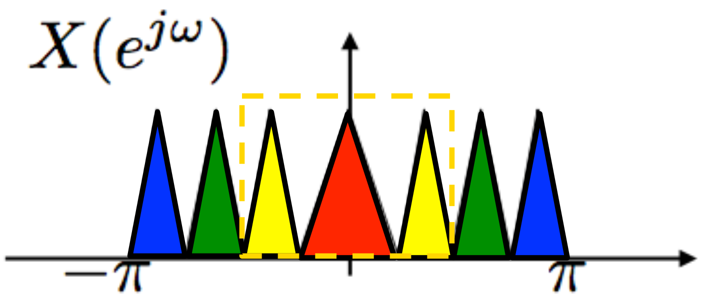


Multi-Rate Filter Banks

- Assume h_0, h_1 are ideal low/high pass

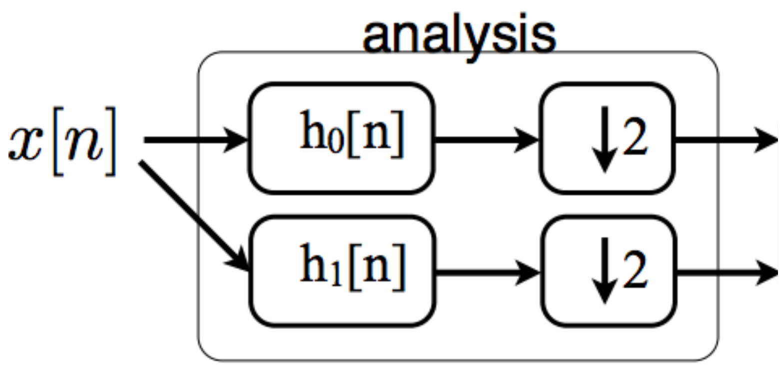


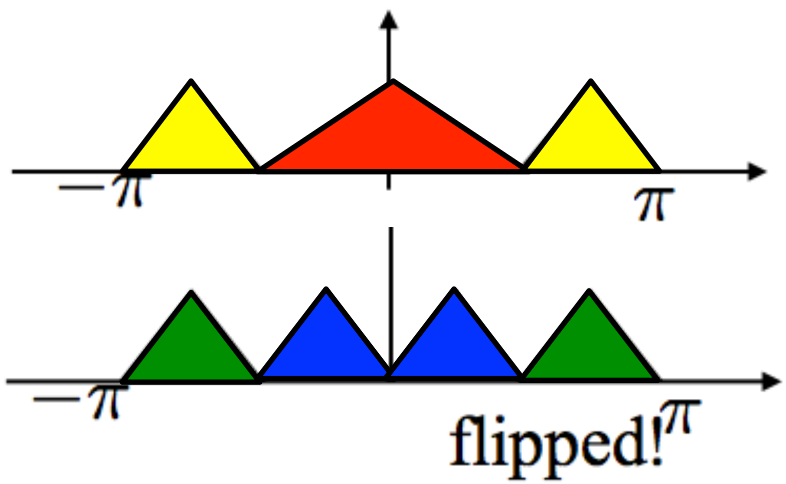


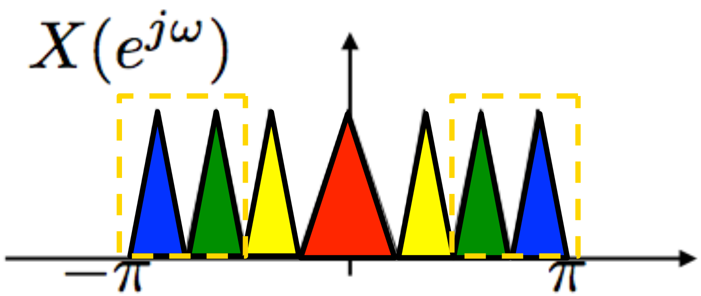


Multi-Rate Filter Banks

- Assume h_0, h_1 are ideal low/high pass

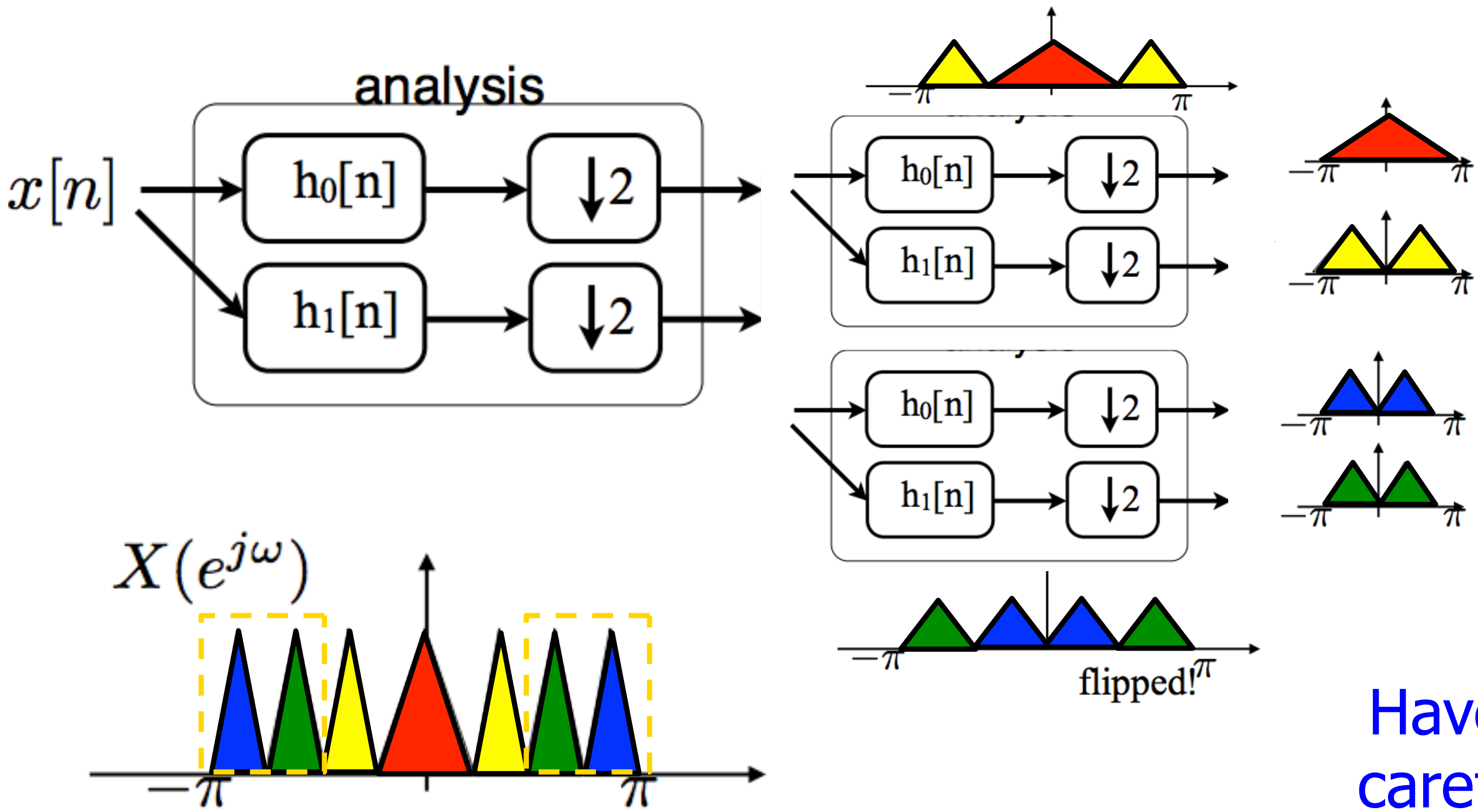






Multi-Rate Filter Banks

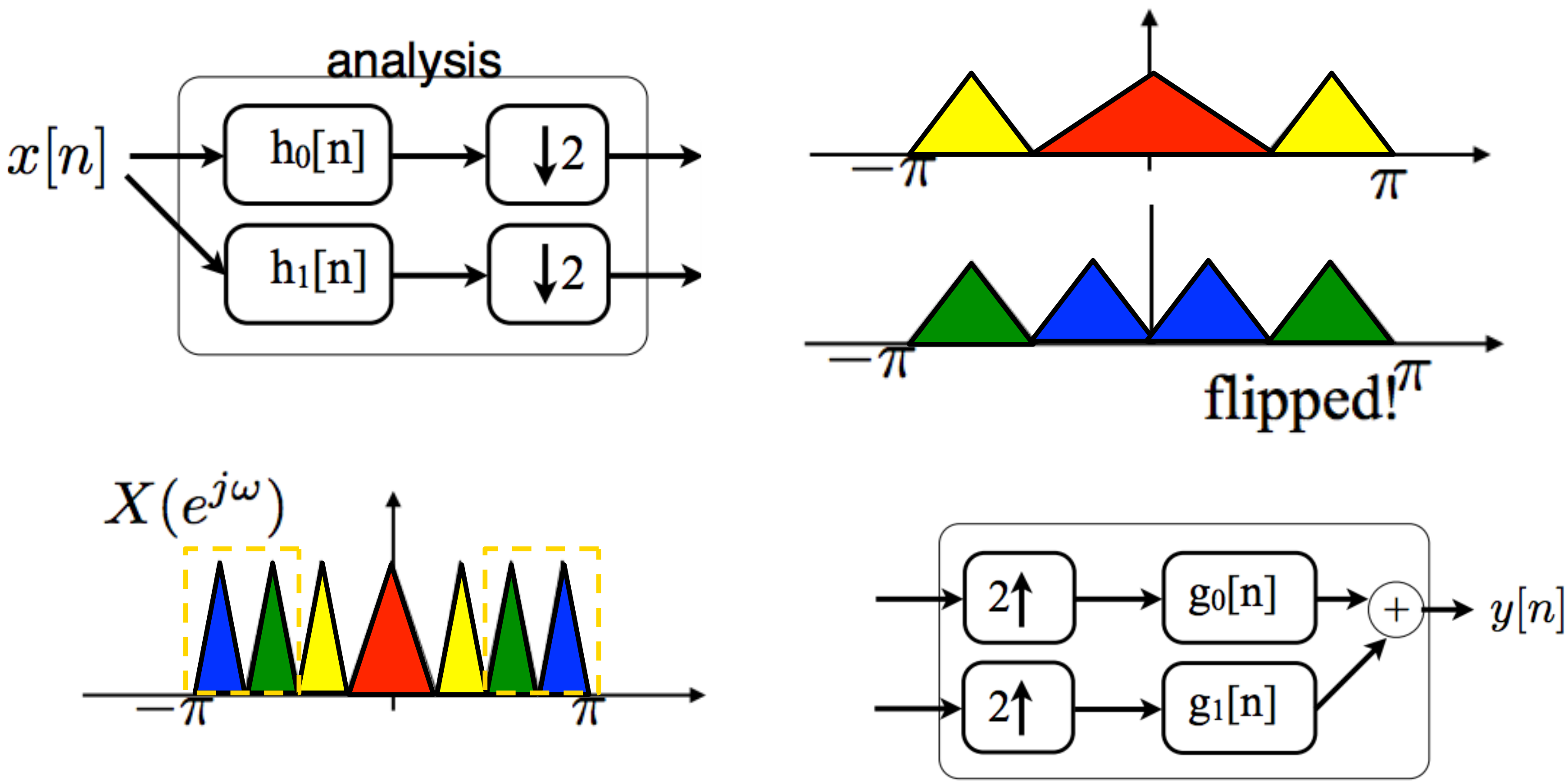
- Assume h_0, h_1 are ideal low/high pass



Have to be careful with order!

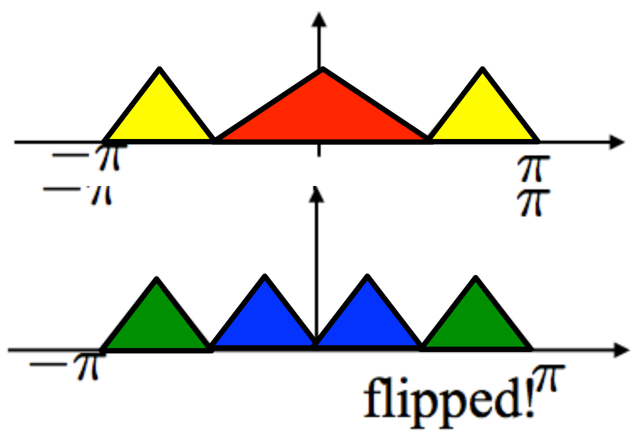
Multi-Rate Filter Banks

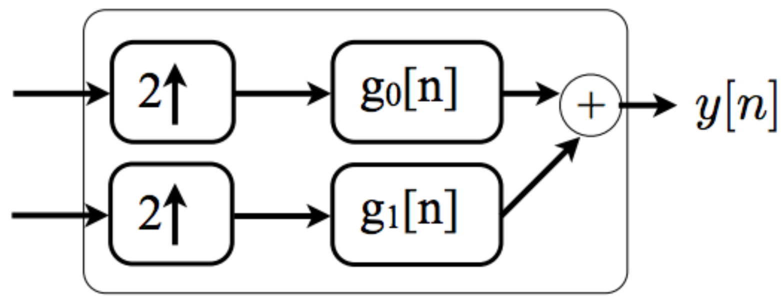
- Assume h_0, h_1 are ideal low/high pass

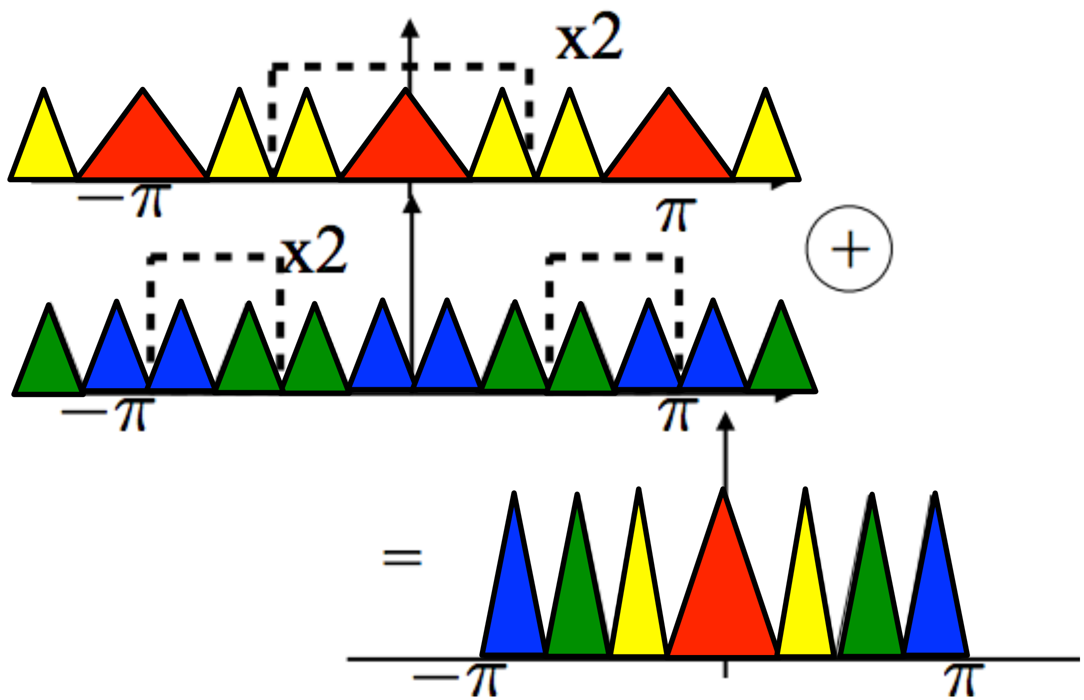


Multi-Rate Filter Banks

- Assume h_0, h_1 are ideal low/high pass

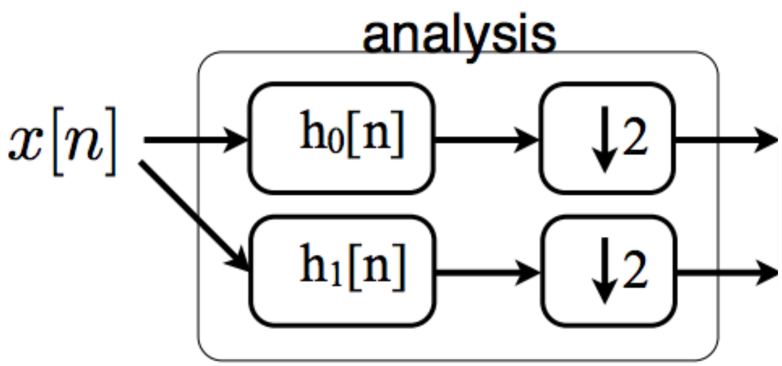


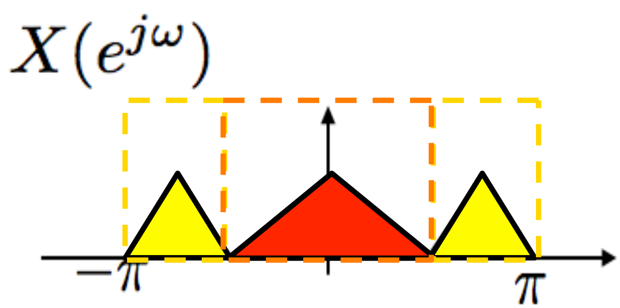


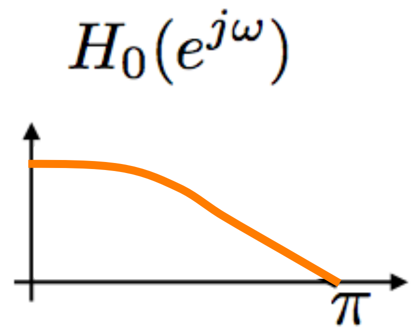


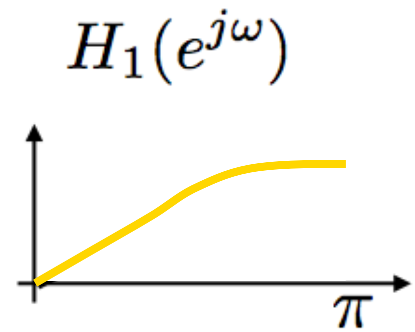
Multi-Rate Filter Banks

- h_0, h_1 are NOT ideal low/high pass



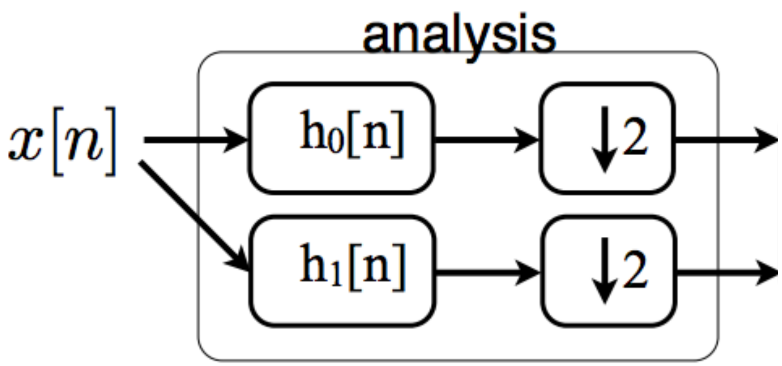


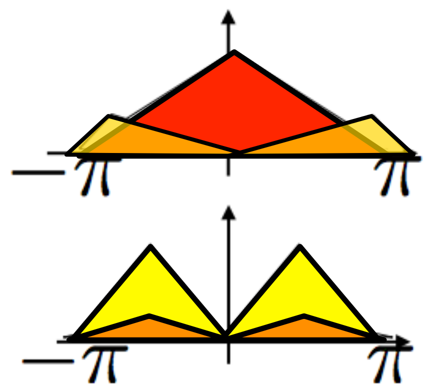


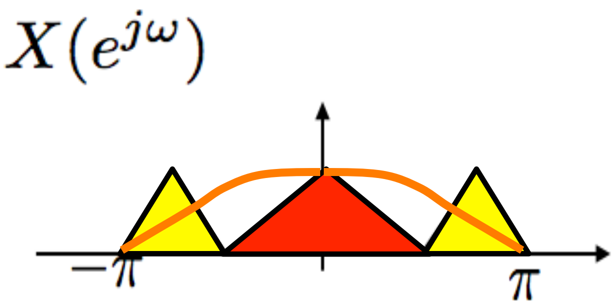


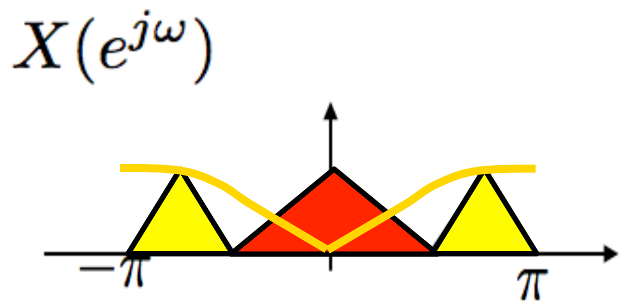
Non Ideal Filters

- h_0, h_1 are NOT ideal low/high pass

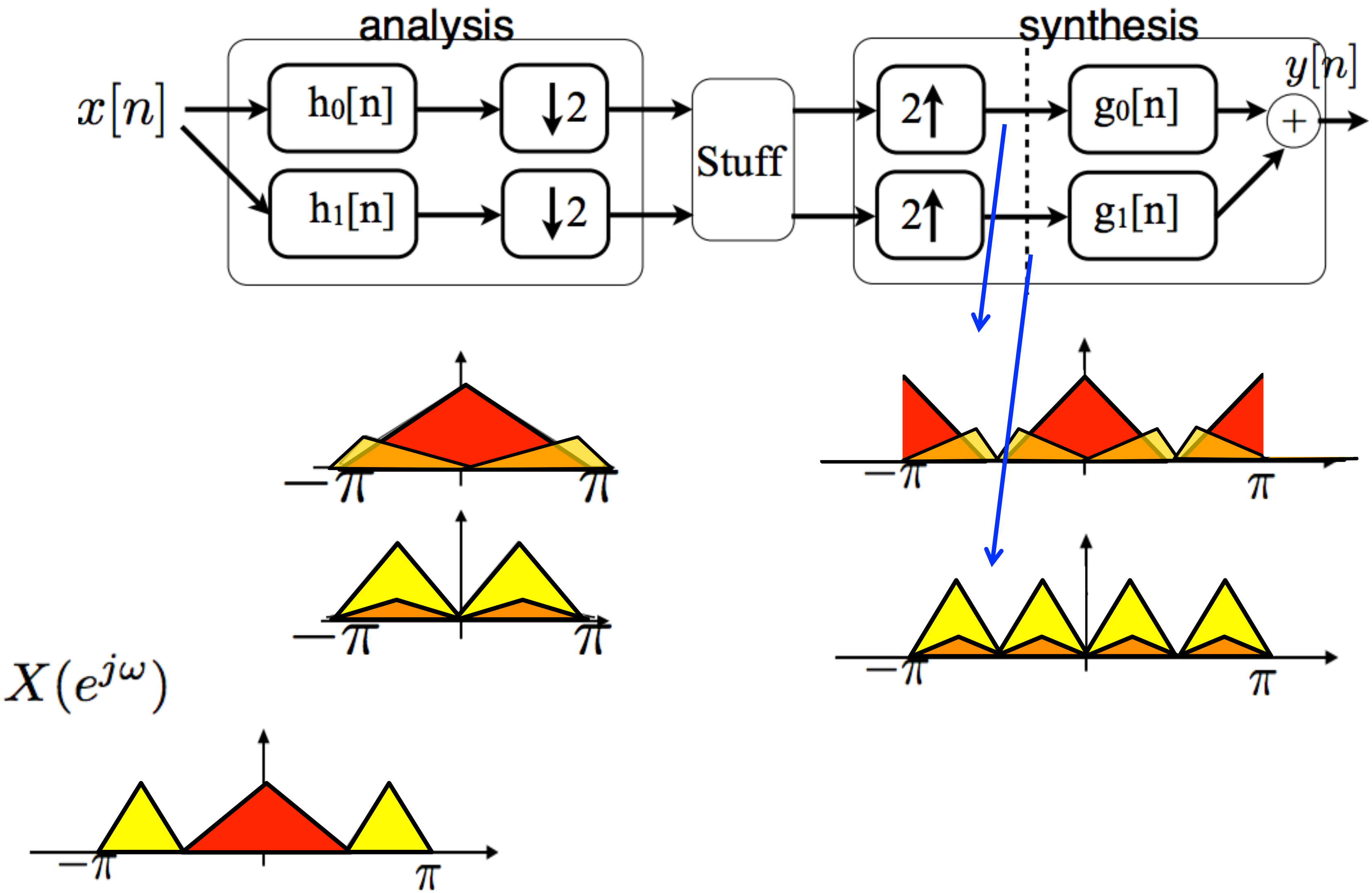




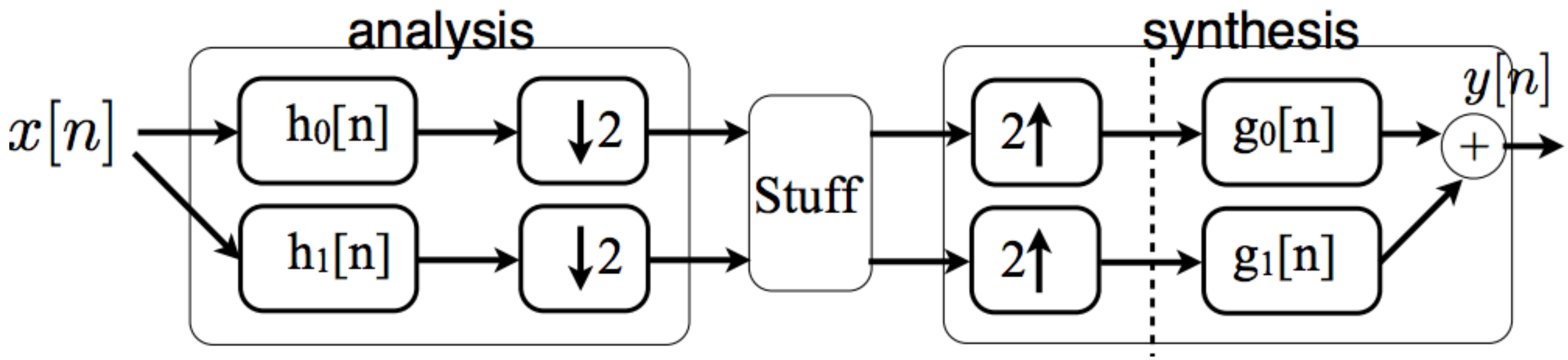




Non Ideal Filters



Perfect Reconstruction non-Ideal Filters

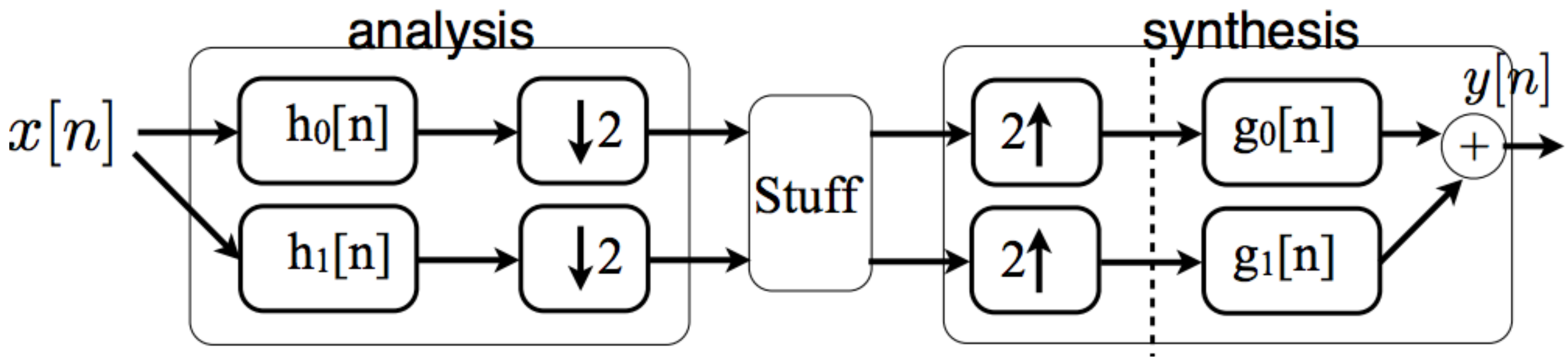


$$
\begin{aligned}
Y(e^{j\omega}) &= \frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j\omega}) + G_1(e^{j\omega})H_1(e^{j\omega})\right] X(e^{j\omega}) \\
&+ \frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j(\omega-\pi)}) + G_1(e^{j\omega})H_1(e^{j(\omega-\pi)})\right] X(e^{j(\omega-\pi)})
\end{aligned}
$$

need to cancel!

aliasing

Quadrature Mirror Filters

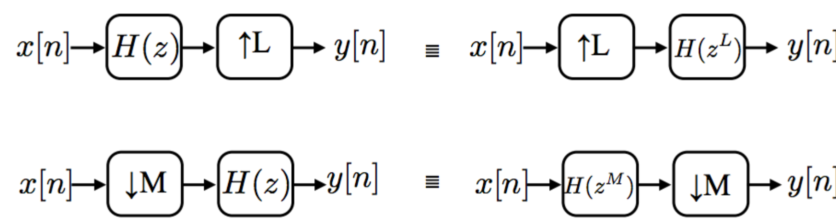


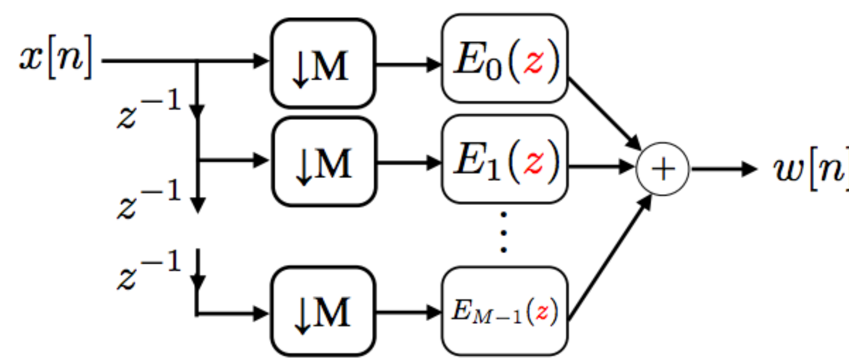
Quadrature mirror filters

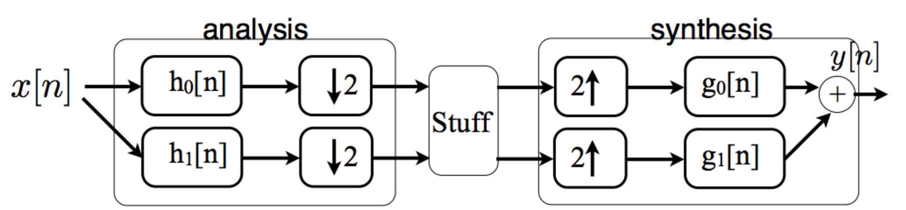
$$
\begin{aligned}
H_1(e^{j\omega}) &= H_0(e^{j(\omega-\pi)}) \\
G_0(e^{j\omega}) &= 2H_0(e^{j\omega}) \\
G_1(e^{j\omega}) &= -2H_1(e^{j\omega})
\end{aligned}
$$

Big Ideas

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







Admin

- HW 4 due Friday
  - Typo in code in MATLAB problem, corrected handout
  - See Piazza for more information