EECE 491: Discrete-time Signal Processing

Mohammad M. Mansour
Dept. of Electrical and Compute Engineering
American University of Beirut

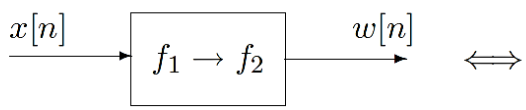
Lecture 10: Sample-Rate Conversion

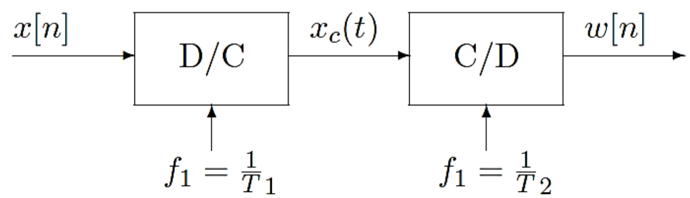
Announcements

- **Reading**
  - O&S
    - Chapter 4

Sample-Rate Conversion

- **It is often necessary to change the sampling rate of a DT signal to obtain a new DT representation of the underlying CT signal.**





$x[n] = x_c(nT)$ C.T. signal sampled at T

$x_1[n] = x_c(nT_1)$ C.T. signal sampled at $T_1 \neq T$

- **Operation is often called "resampling"**
- **One way to obtain $x_1[n]$ from $x[n]$ is as follows:**
  - Reconstruct $x_c(t)$ from $x[n]$ using ideal and band-limited interpolation

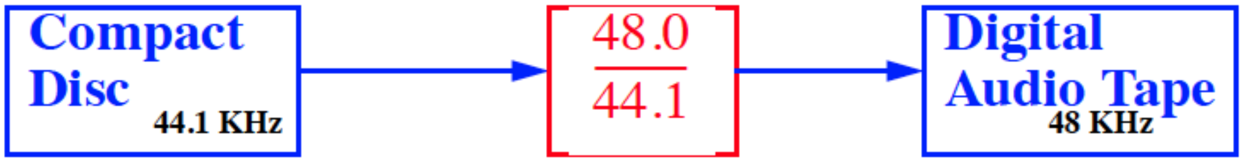
$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n] \frac{\sin\left[\pi(t-nT)/T\right]}{\pi(t-nT)/T}$$

  - Resample $x_c(t)$ with period T_1 to obtain $x_1[n]$
  - Approach is impractical due to non-ideal analog reconstruction filter, D/A converter and A/D converter
- **Objective: do sample rate conversion using only discrete-time operations**

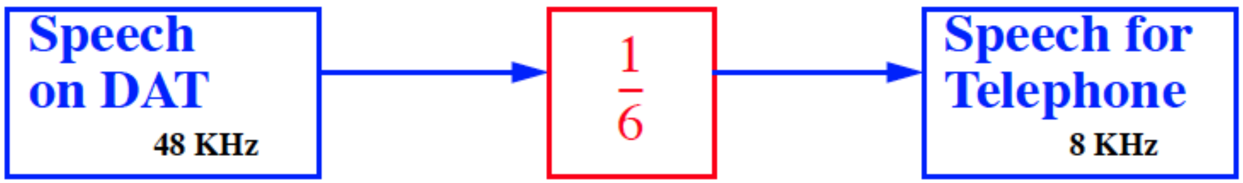
Resampling: Examples

Changing the Sampling Rate

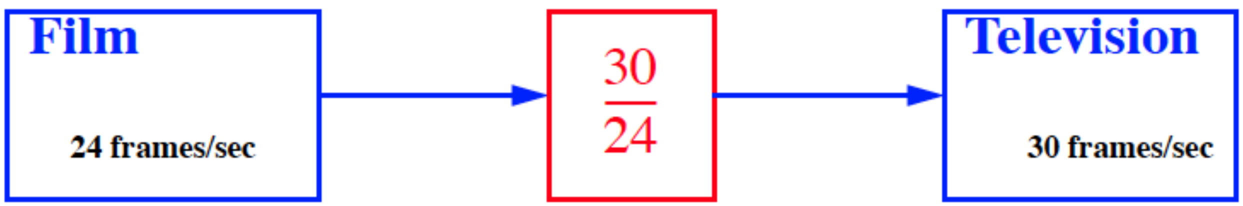
- **Conversion between audio formats**



- **Speech compression**



- **Video format conversion**



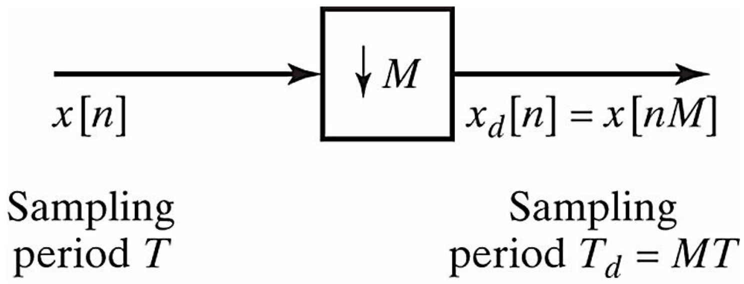
source: UT austin

Downsampling

Sample-Rate Reduction by an Integer Factor

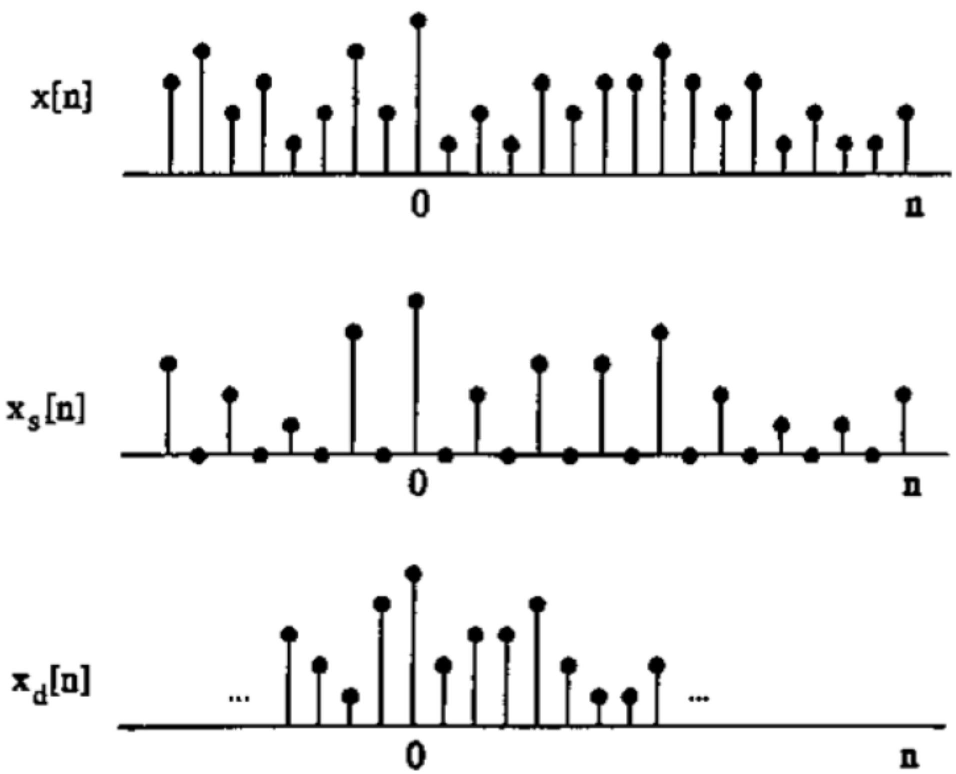
- **Sampling rate of a sequence can be reduced by "sampling" it:**
  - Called "down-sampling" or "compression"

$$x_d[n] = x[nM] = x_c(nMT)$$



sample-rate compressor

***M* = 2**



Sample-Rate Reduction by an Integer Factor

$$\text{if } X_c(j\Omega) = 0 \text{ for } |\Omega| > \Omega_N \;\Rightarrow$$
$$x_d[n] \text{ is an exact representation of } x_c(t) \text{ if } \pi/\mathrm{T}_d = \pi/(MT) \geq \Omega_N$$

- **Therefore, the sampling rate can be reduced to π/M without aliasing if the original sampling rate is at least *M* times the Nyquist rate**

- **Or equivalently, if the bandwidth of the sequence is first reduced by a factor of *M* by discrete-time filtering.**

- **Note also that the principle of sampling in one domain results in replicated aliases in the dual domain applies here**
  - Since we are sampling the signal, we expect periodic replication in the frequency domain

Frequency-Domain Relationship of Downsampling

$$x[n] = x_c(nT) \quad \Rightarrow \quad X\left(e^{j\omega}\right) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right)$$

$$x_d[n] = x[nM] = x_c(nT_d) \quad \Rightarrow \quad X_d\left(e^{j\omega}\right) = \frac{1}{T_d}\sum_{r=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T_d} - \frac{2\pi r}{T_d}\right)\right)$$

$$= \frac{1}{MT}\sum_{r=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi r}{MT}\right)\right)$$

Set $r = i + kM; \quad -\infty < k < +\infty, \quad 0 \le i \le M-1$

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

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

DTFT of $x_d[n]$ — DTFT of $x[n]$

Interpretation of the Relationship

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

- **$X_d(e^{j\omega})$ can be thought of being composed of a superposition of infinite set of copies of $X_c(j\Omega)$, amplitude scaled by $1/MT$, frequency scaled through $\omega = \Omega T_d$, and shifted by integer multiples of 2π**

or using the relationship

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

- **$X_d(e^{j\omega})$ can be thought of being composed of M amplitude-scaled copies of the periodic DTFT $X(e^{j\omega})$, frequency scaled by M, shifted by integer multiples of 2π**

- **Either interpretation makes it clear that $X_d(e^{j\omega})$ is periodic and aliasing can be avoided if $X(e^{j\omega})$ is bandlimited:**

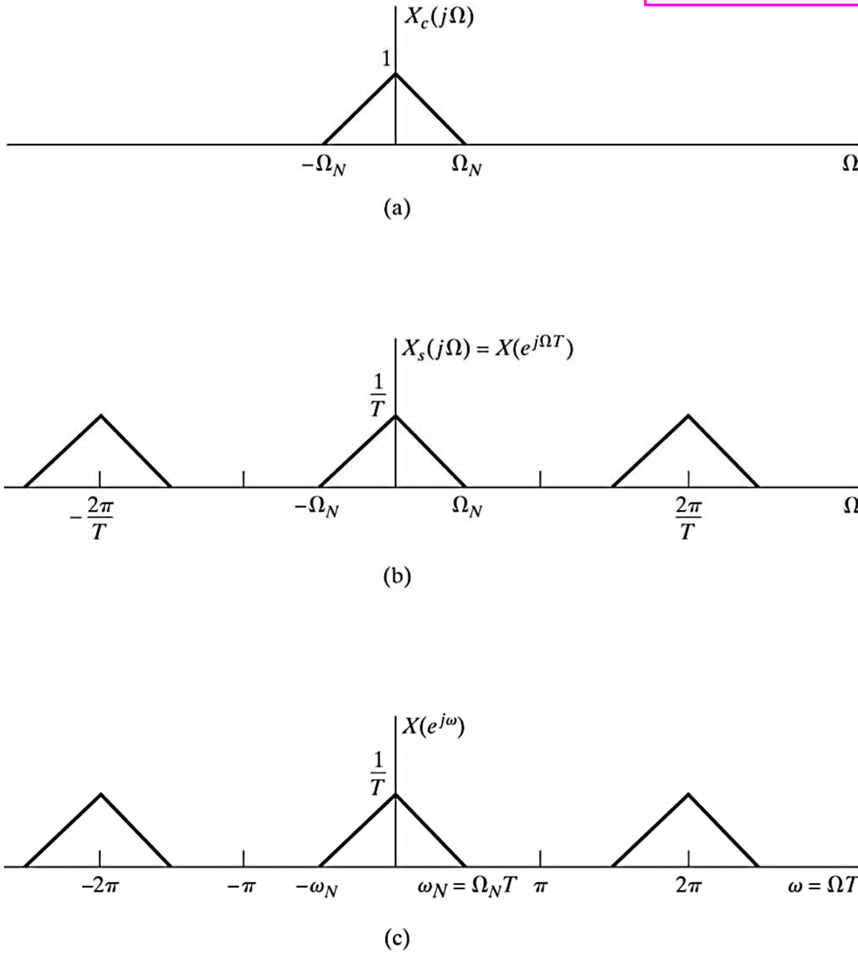
$$X\left(e^{j\omega}\right) = 0, \quad \omega_N \le |\omega| \le \pi$$

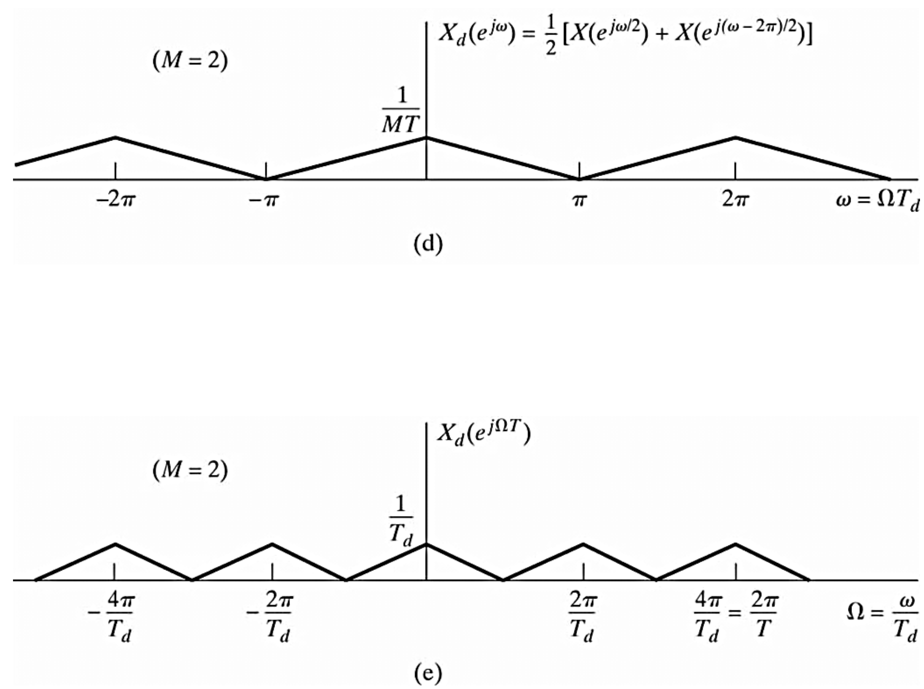
$$\frac{2\pi}{M} \ge 2\omega_N$$

Frequency-domain Illustration of Downsampling (*no aliasing*)

$$\text{Assume } \Omega_s = \frac{2\pi}{T} = 4\Omega_N$$

$$M = 2$$

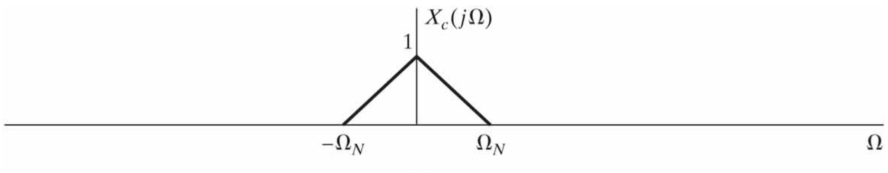




$$\Omega_s' = \frac{2\pi}{T_d} = \frac{2\pi}{2T} = 2\Omega_N$$

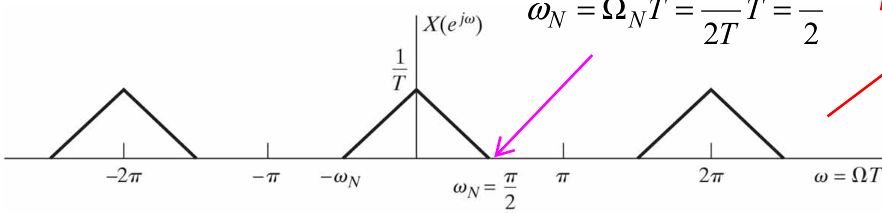
Frequency-domain Illustration of Downsampling (*with aliasing*)

$\text{Assume } \Omega_s = \frac{2\pi}{T} = 4\Omega_N$ $\Rightarrow \Omega_N = \frac{\pi}{2T}$ $M = 3$



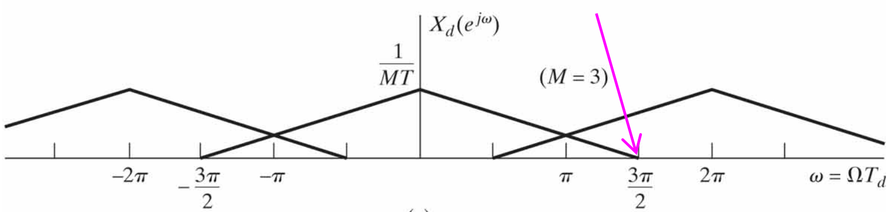
(a)

$\omega_N = \Omega_N T = \frac{\pi}{2T} T = \frac{\pi}{2}$

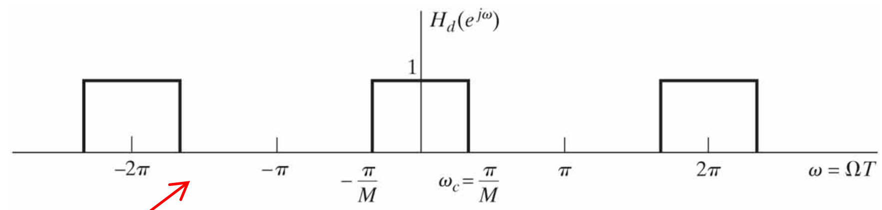


(b)

$\omega_N = \Omega_N T_d = \frac{\pi}{2T} 3T = \frac{3\pi}{2}$

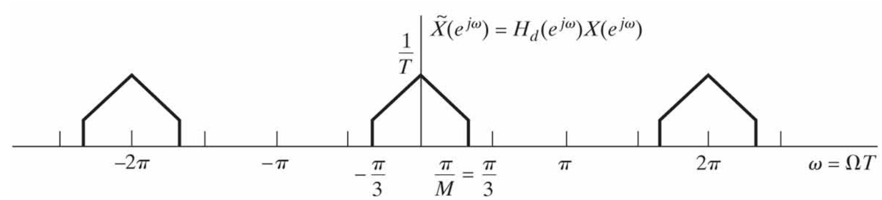


(c)

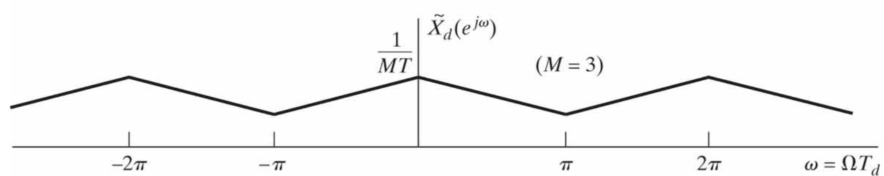


(d)

(to avoid aliasing, LPF $X(e^{j\Omega})$ with cutoff frequency $\omega_c = \pi/M$ before downsampling)



(e)



(f)

(*aliasing occurs due to downsampling*)

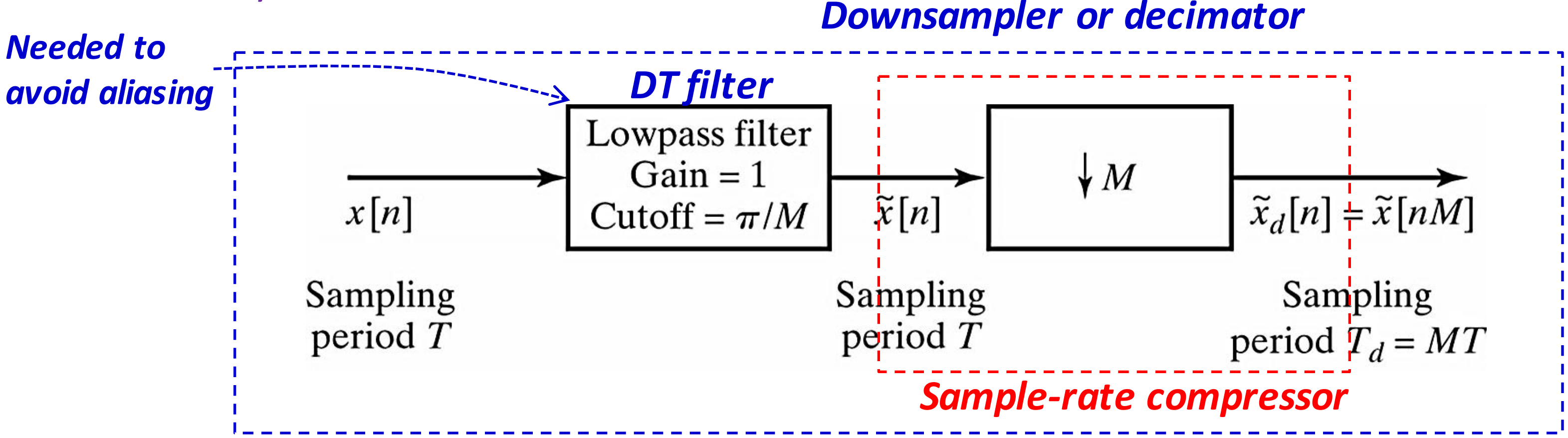
In general aliasing occurs if $\omega_N M \geq \pi$ or $\omega_N \geq \pi/M$

Decimator

- **We saw that aliasing occurs due to downsampling if**

$$\omega_N M \geq \pi$$

- **To avoid aliasing, need to low-pass filter *x*[*n*] with an ideal LPF with cutoff frequency $\omega_c = \pi/M$ before downsampling**
  - The output $\tilde{x}[n]$ can then be downsampled by a factor of *M* without aliasing
- **Downsampling by lowpass filtering followed by *compression* is called *decimation***
  - System is called ***decimator***



- **Note: $\tilde{x}_d[n] = \tilde{x}[nM]$ no longer represents the original underlying CT signal $x_c(t)$**
  - **Rather, $\tilde{x}_d[n] = \tilde{x}_c(nT_d)$ where $T_d = MT$ and $\tilde{x}_c(t)$ is obtained from** $x_c(t)$ by low-pass filtering with cutoff frequency

$$\Omega_c = \pi / T_d = \pi(MT)$$

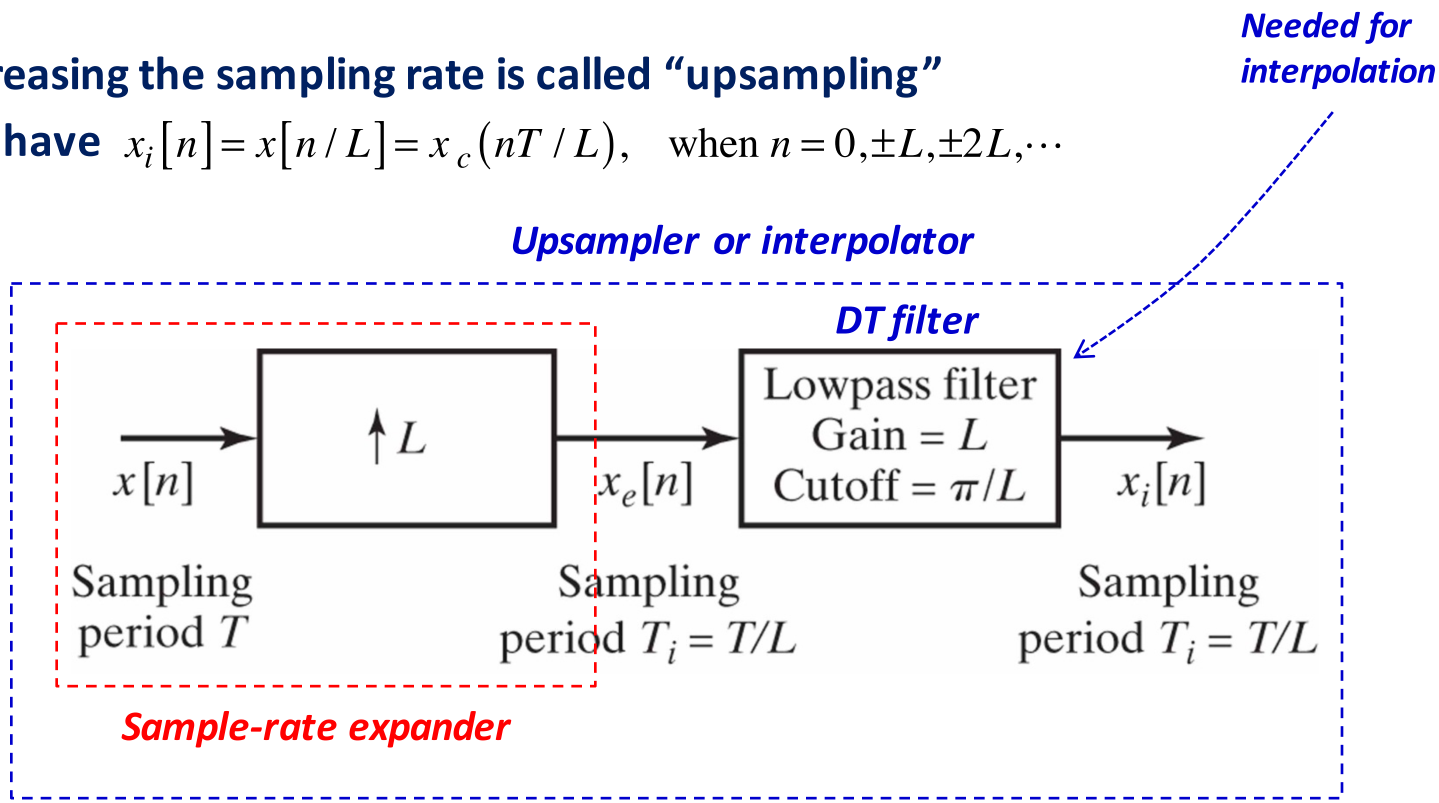
Upsampling

Increasing the Sample-Rate by an Integer Factor

- **Consider DT signal $x[n]$ whose sample rate we wish to increase by a factor of L.**
  - Underlying CT signal is $x_c(t)$
- **Obtain**

$$x_i[n] = x_c(nT_i), \quad \text{where } T_i = T/L,$$
$$\text{from } x[n] = x_c(nT)$$

- **Increasing the sampling rate is called "upsampling"**
- **We have** $x_i[n] = x[n/L] = x_c(nT/L), \quad \text{when } n = 0, \pm L, \pm 2L, \cdots$

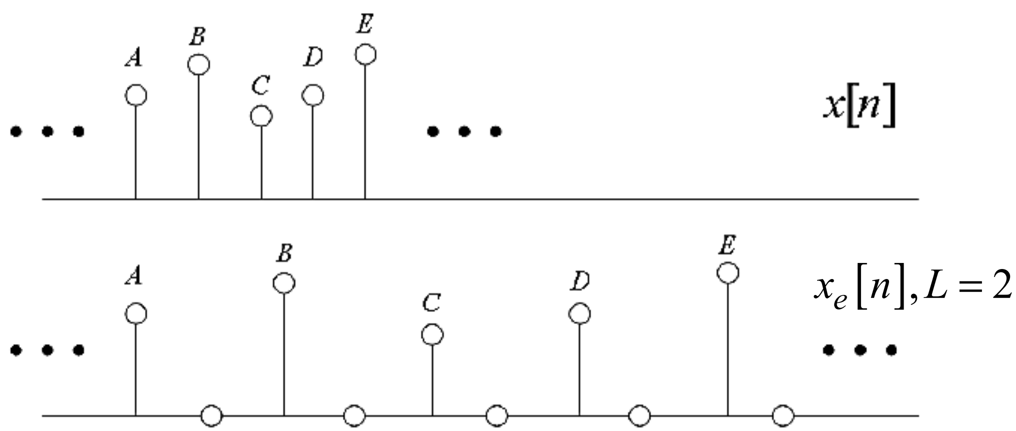


(1) Expander: Time-Domain Relations

$$x_e[n] = \begin{cases} x[n/L] & n = 0, \pm L, \pm 2L, \ldots \\ 0 & \text{otherwise} \end{cases}$$

$$\text{or equivalently } x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\,\delta[n-kL]$$

Time domain illustration of expander with L = 2



(2) Expander: Frequency-Domain Relations

$$\text{Assume } \Omega_s = \frac{2\pi}{T} = 2\Omega_N \Rightarrow \Omega_N = \frac{\pi}{T}$$

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]$$

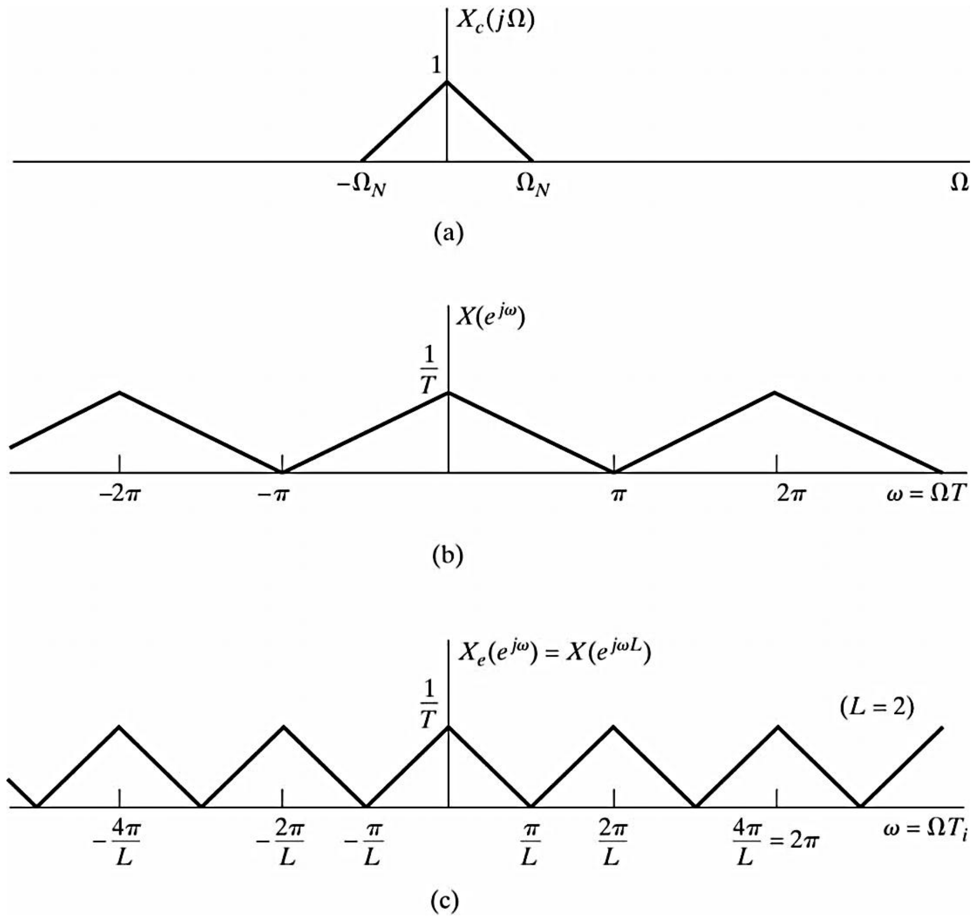
DTFT

$$X_e\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]\right)e^{-j\omega n}$$

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

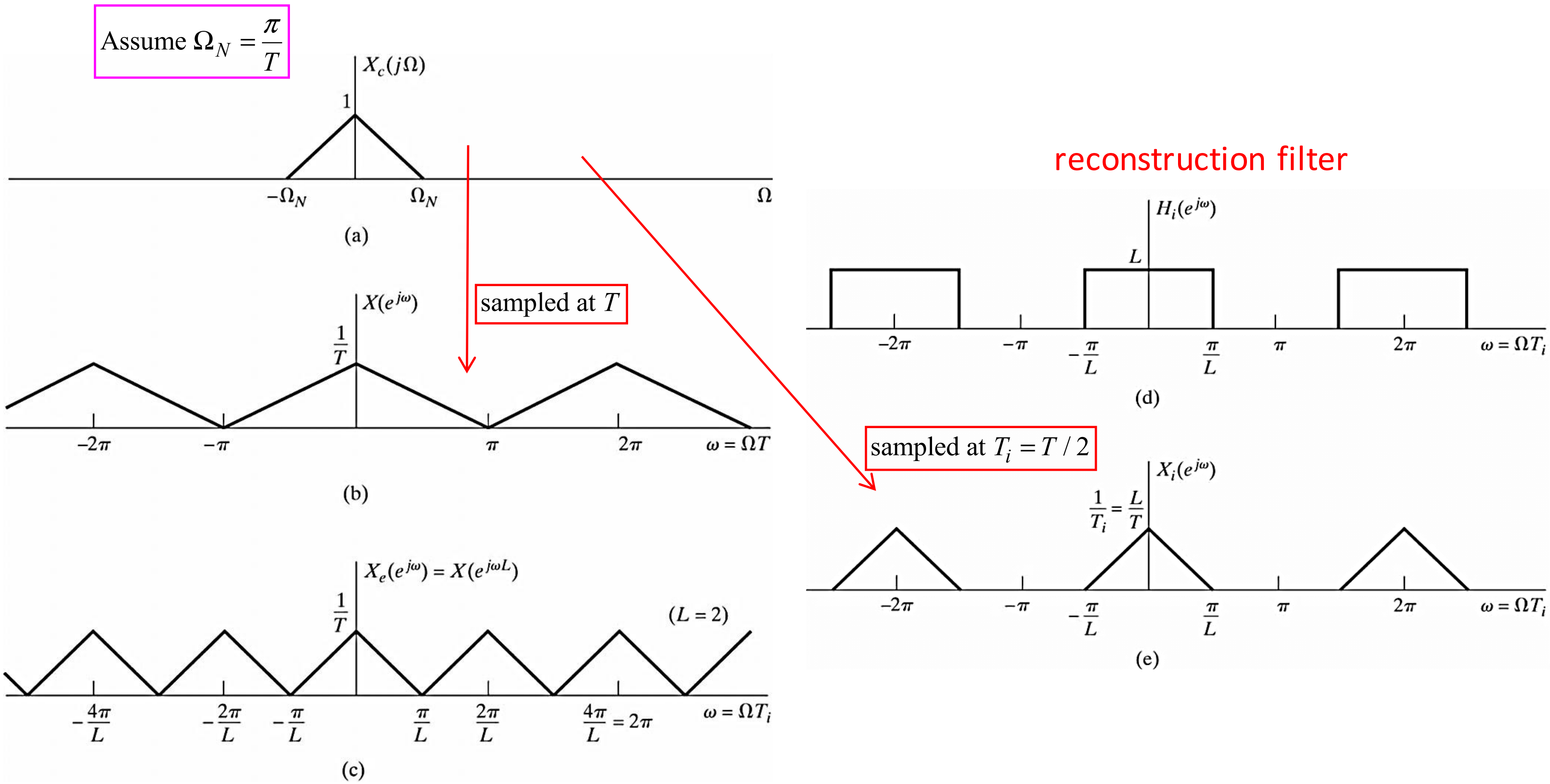
$$= X\left(e^{j\omega L}\right)$$

- **Output DTFT is frequency-scaled version of DTFT of input**
- ***ω* is replaced by *ωL***



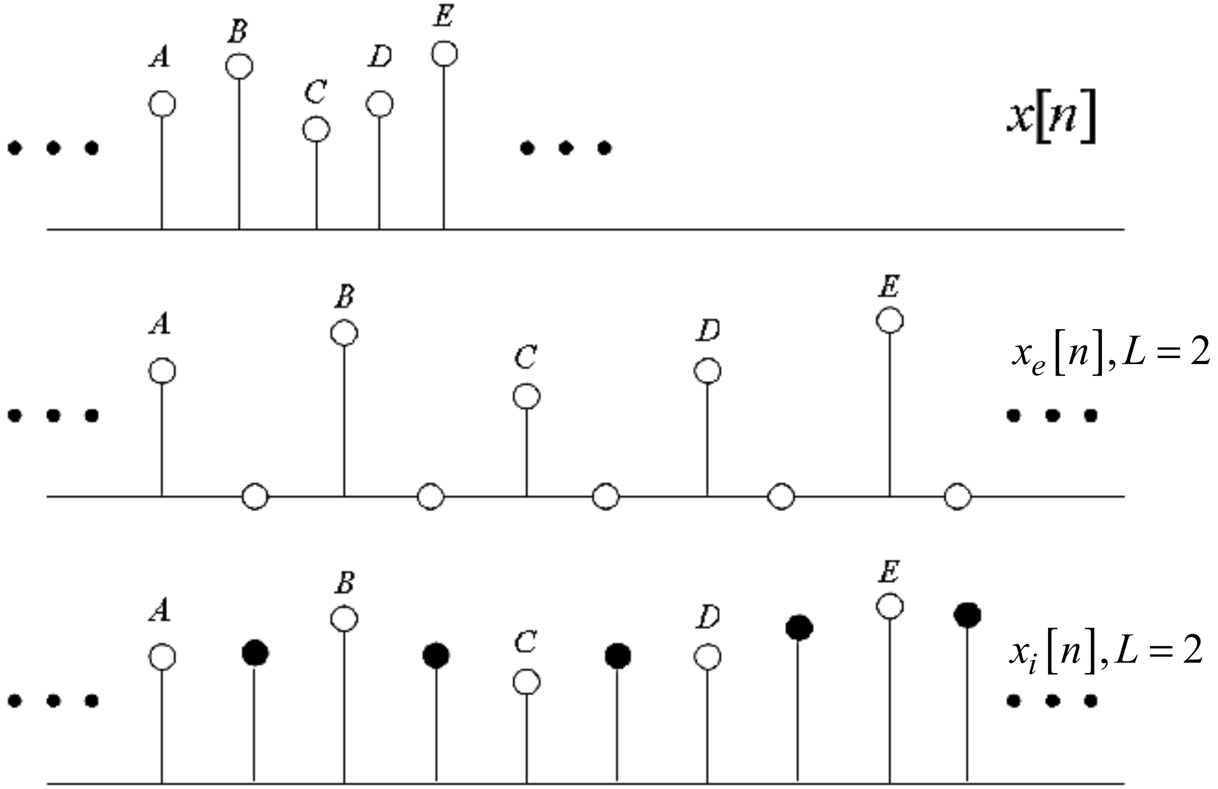
(3) Lowpass Filter

- **The LPF with gain L and cutoff frequency $\omega_c = \pi/L$ plays the role of the reconstruction filter to obtain $x_i[n]$ from $x_e[n]$**



(3) Lowpass Filter

- **The reconstruction LPF, fills in the intermediate values to obtain $x_i[n]$ from $x_e[n]$**
  - Hence the reconstruction filter does interpolation



Interpolation

- **Obtain a time-domain relationship between $x_i[n]$ and $x[n]$**
- **Impulse response of the LPF is**

$$h_i[n] = \frac{\sin(\pi n / L)}{\pi n / L}; \ \text{cutoff } \omega_c = \pi / L$$

- **We have**

$$x_i[n] = x_e[n] * h[n]$$

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n - kL]$$

$$\Rightarrow \quad x_i[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{\sin[\pi(n-kL)/L]}{\pi(n-kL)/L}$$

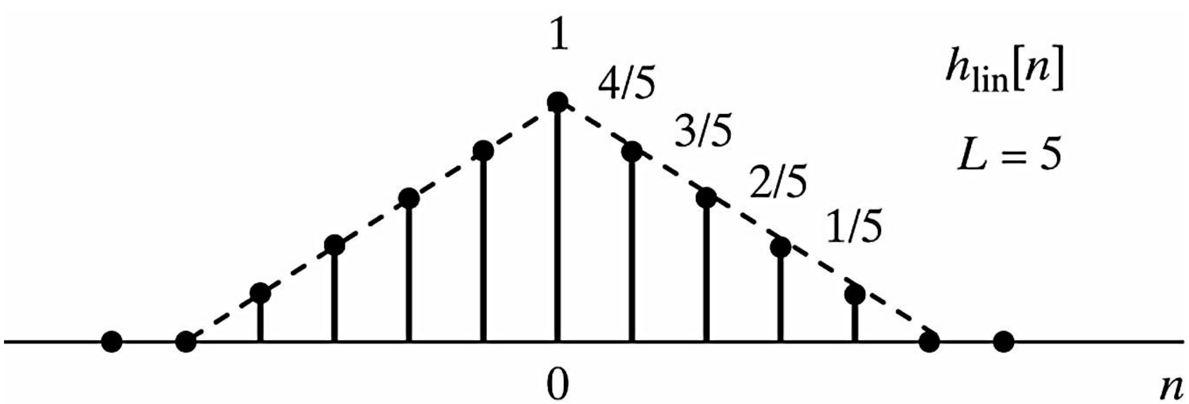
- **$h_i[n]$ has the properties:**
  - $h_i[0] = 1$
  - $h_i[n] = 0$, for $n =$ L, $2L$,…
- **Therefore,**

$$x_i[n] = x[n/L] = x_c(nT/L) = x_c(nT_i), \quad \text{for all } n$$

Practical Interpolators

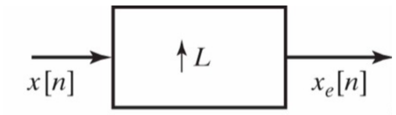
- **Ideal LPFs for interpolation cannot be implemented exactly**
  - PM algorithm gives good FIR approximations
- **Examine other forms of interpolation**
  - Ex: linear interpolators

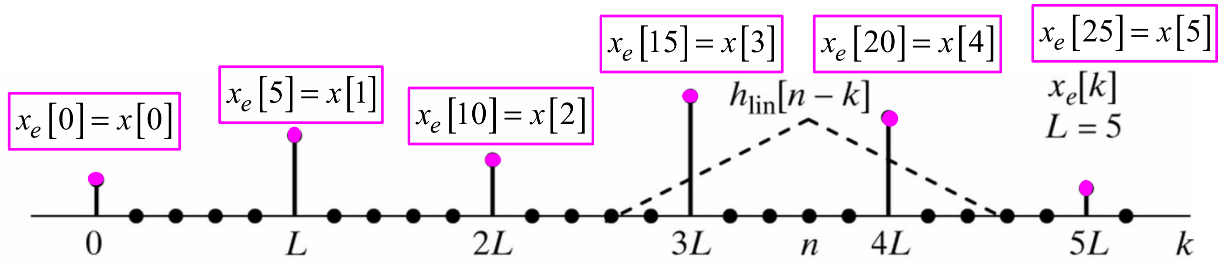
$$h_{\text{lin}}[n] = \begin{cases} 1 - |n|/L & |n| \le L \\ 0 & \text{otherwise} \end{cases}$$



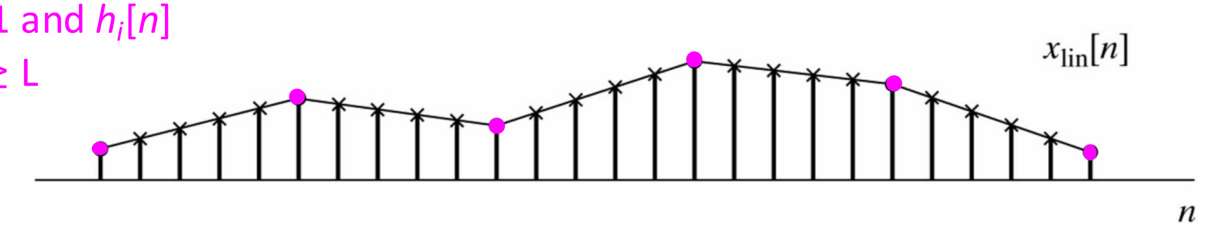
Impulse response for linear interpolation.

Example: Linear Interpolation by Filtering





Original samples in $x_{\text{lin}}[n]$ are preserved since $h_i[0] = 1$ and $h_i[n] = 0$, for $|n| \geq L$



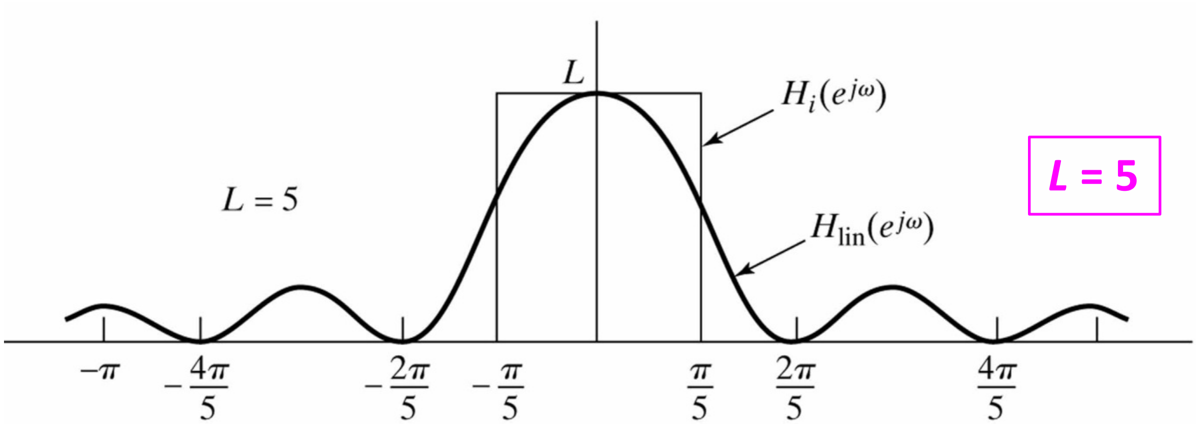
$$x_{\text{lin}}[n] = \sum_{k=n-L+1}^{n+L-1} x_e[k] h_{\text{lin}}[n-k]$$

Example: Linear Interpolation by Filtering (cont'd)

Nature of distortion in the intervening samples is better understood by comparing frequency responses of ideal and linear interpolators

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

$$H_{lin}\left(e^{j\omega}\right) = \frac{1}{L}\left[\frac{\sin(\omega L/2)}{\sin(\omega/2)}\right]^2$$



***L* = 5**

If original sampling rate greatly exceeds Nyquist rate, signal will not vary significantly between samples, and hence linear interpolation will be more accurate for oversampled signals

FIR Filters as Interpolators

- **Ideal bandlimited interpolators involve all original samples in the convolution of each interpolated sample**
- **In contrast, linear interpolation involves only two**
- **To get better approximation, use longer impulse responses**
- **FIR filters $\tilde{h}_i[n]$ are advantageous in this case. To interpolate by a factor *L*, they are usually designed with the following properties:**

$$\tilde{h}_i[n] = 0, \qquad |n| \geq KL$$

$$\tilde{h}_i[n] = \tilde{h}_i[-n], \qquad |n| \leq KL$$

$$\tilde{h}_i[0] = 1, \qquad n = 0$$

$$\tilde{h}_i[n] = 0, \qquad n = \pm L, \pm 2L, \cdots, \pm KL$$

Interpolated output

$$\tilde{x}_i[n] = \sum_{k=n-KL+1}^{n+KL-1} x_e[k]\tilde{h}_i[n-k]$$

→ *Guarantee original signal samples are preserved in output* (from $\tilde{h}_i[0] = 1$ and $\tilde{h}_i[n] = 0$ at $n = \pm L, \pm 2L, \cdots, \pm KL$)

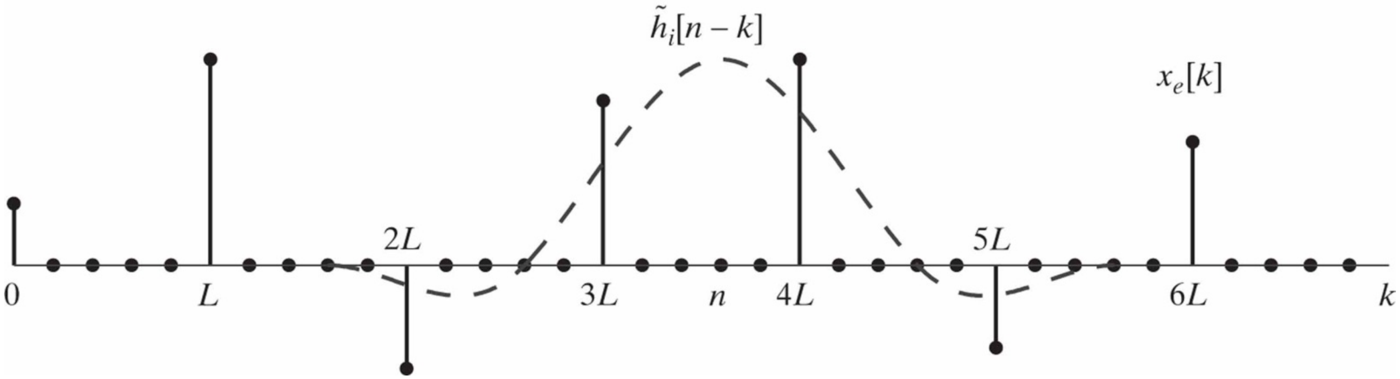
$$\tilde{x}_i[n] = x[n/L], \quad \text{at } n = 0, \pm L, \pm 2L, \cdots$$

→ *Filter will not introduce any phase shift into the interpolated samples* (from $\tilde{h}_i[n] = \tilde{h}_i[-n]$)

→ *Only 2K non-zero samples within the region of support of $\tilde{h}_i[n-k]$ are involved in the interpolation* (from $\tilde{h}_i[n] = 0$, $|n| \geq KL$)

FIR Filters as Interpolators

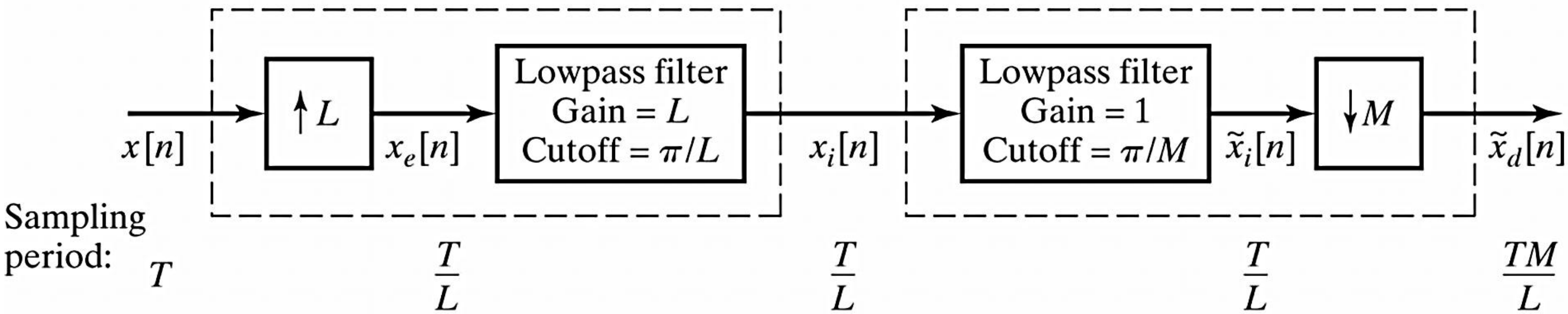
- **Illustration of interpolation involving 2K = 4 samples when L = 5.**
  - Each interpolated value depends on $2K = 4$ samples of the original input
  - Computation requires only $2K$ multiplications and $2K - 1$ additions since there are always $L - 1$ zero samples in $x_e[k]$ between each of the original samples



- **Higher-order Lagrange interpolation formulas are possible from theory of numerical analysis**

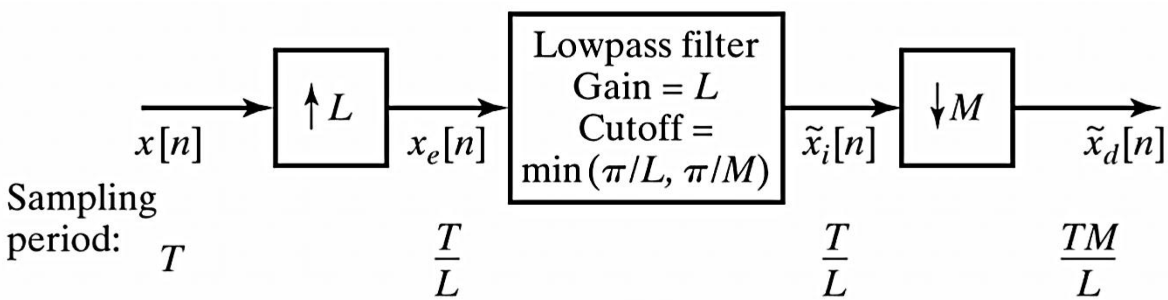
Changing the Sample Rate by an Noninteger Factor

- **By combining decimation and interpolation, it is possible to change the sample rate by a noninteger factor**



(a)

Choosing L, M we get arbitrarily close to any desired ratio of sampling periods



(b)

Example

- Assume $x_c(t)$ is sampled at Nyquist rate $\Omega_N=\pi/T$.
- Want to change sampling Period to 3T/2.
- Can choose L=2, M=3.
- LPF has gain 2 and cutoff $\omega_c = \min(\pi/2, \pi/3) = \pi/3$