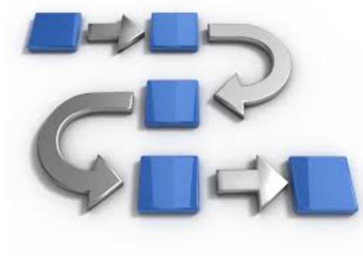
DSP: Part II

Lecture.1.2

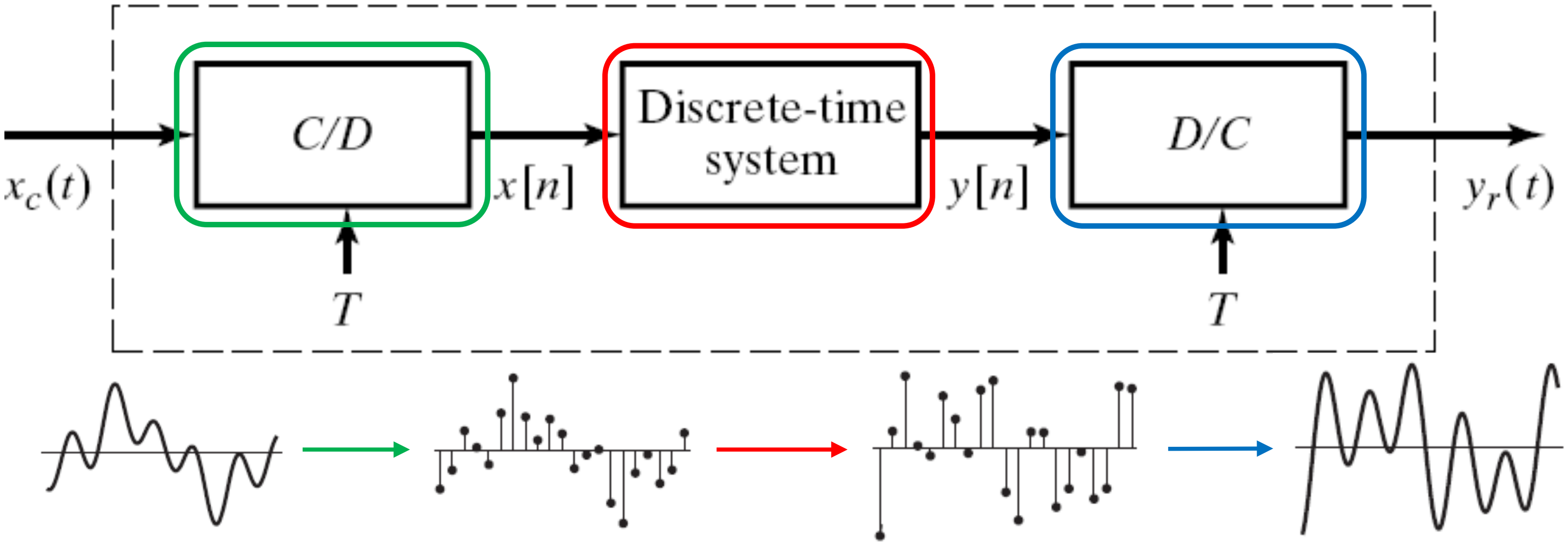
Sampling and Reconstruction (cont'd)

Methodology/Logic for Sampling and Reconstruction



A. Overview sampling – a conversion process
B. Sampling theorem & aliasing
C. A mathematical model of sampling in frequency domain
D. Reconstruction
E. Discrete time processing of continuous time signals
F. Up and down sampling
G. Quantization

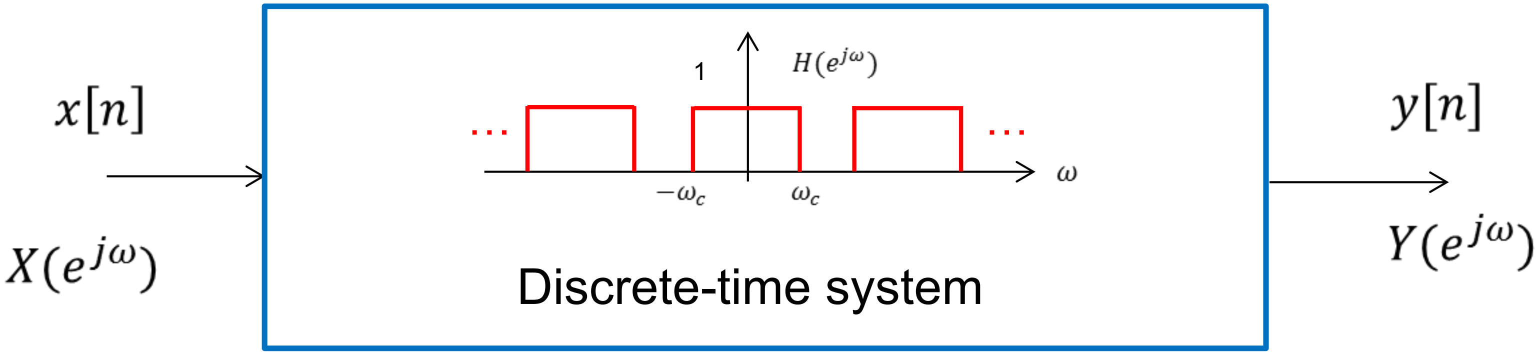
E. Discrete-time Processing of Continuous-time Signals



A general block diagram

E. Input-output Relationship for Discrete-time processing of continuous-time signals

- What is the relationship between input and output signals if the system is linear and time invariant ?



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

$$\boldsymbol{y[n] = h[n] * x[n]}$$

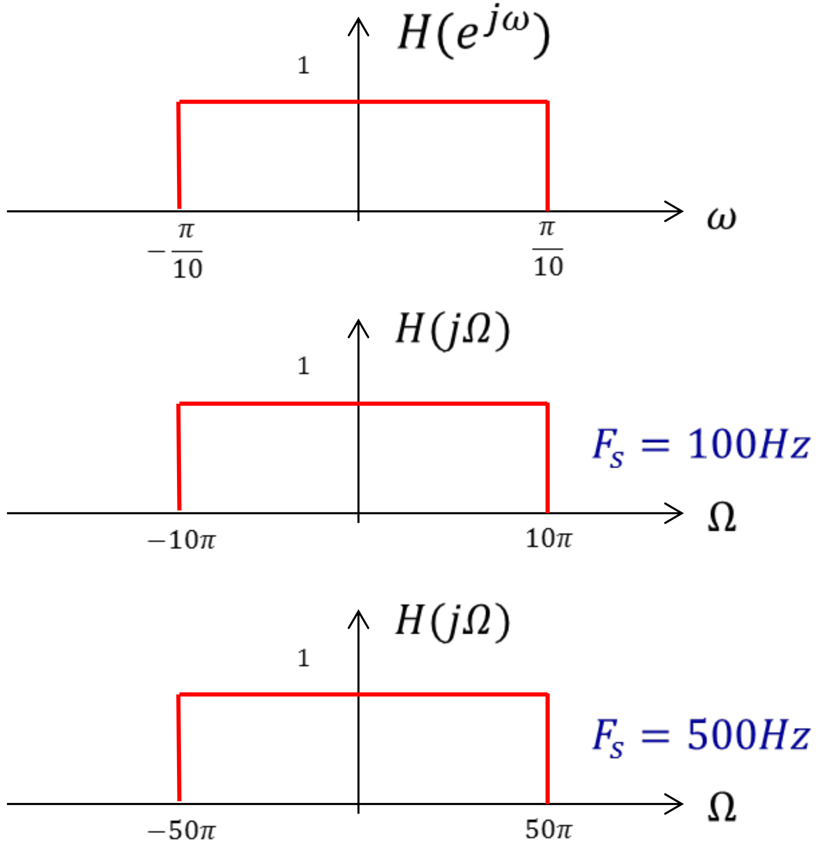
$$\omega = \frac{\Omega}{F_s} \text{ (radian/sample)}$$

E. Input-output Relationship for Discrete-time processing of continuous-time signals

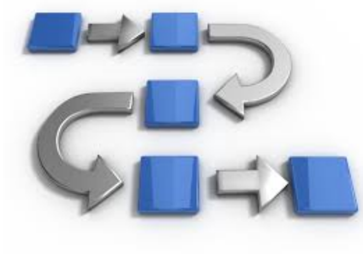
- Recap: Meaning and relationship between different types of frequencies?

Description	Notation	Unit
Analog frequency	F	Hz
Analog frequency	$\Omega = 2\pi F$	rad/sec
Digital Frequency	$\omega = {}^{2\pi F}/_{F_s}$	rad/sample

E.g, given $H(e^{j\omega})$ is a low pass filter with $\omega_c = 0.1\pi$,
What is its cut-off frequency in analog frequency $H(j\Omega)$?
Answer: Depending on F_s!
Using the relationship $\omega = \Omega T_s$,
therefore $\Omega_c = \omega_c F_s$



Methodology/Logic for Sampling and Reconstruction

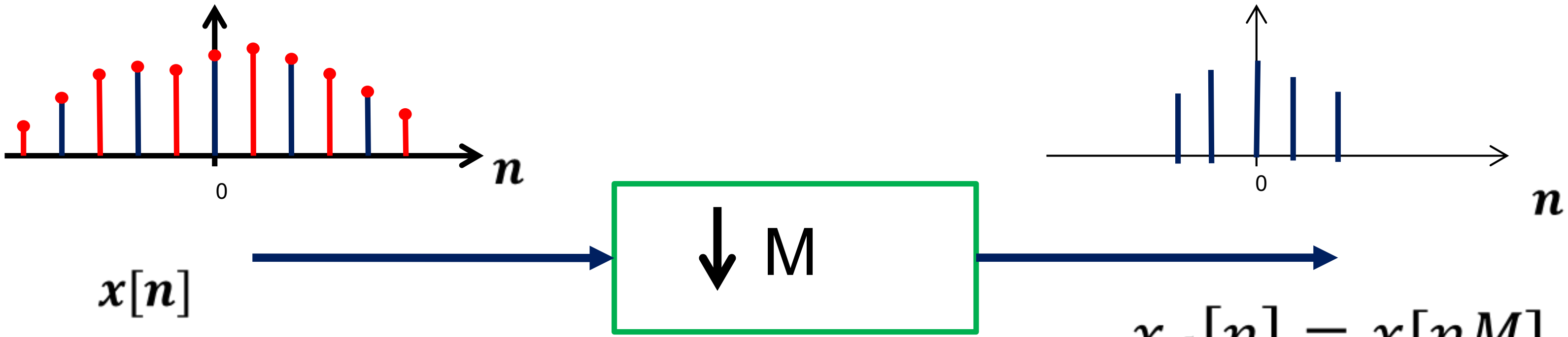


A. Overview sampling – a conversion process
B. Sampling theorem & aliasing
C. A mathematical model of sampling in frequency domain
D. Reconstruction
E. Discrete time processing of continuous time signals
F. Up and down sampling
G. Quantization

F. Sampling Rate Conversion: A Frequency Domain View

- Changing the Sampling rate using discrete-time processing

$x[n] = x_c(nT_s)$ $\qquad$ $x_d[n] = x[nM] = x_c(nMT_s)$



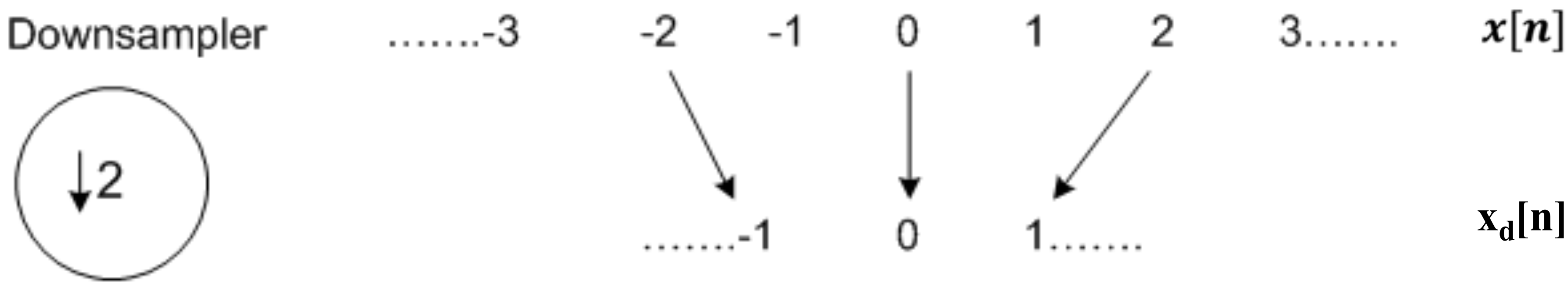
$x_d[n] = x[nM]$

Sampling period T_s $\qquad$ Sampling period $T'_s = MT_s$

- $x[n] = x_c(nT_s)$ then $x_d[n] = x[nM] = x_c(nMT_s)$

<u>Application</u>: Transferring signals from one medium to another, e.g. from Audio tape (F_s = 48 kHz) to CD (F_s = 44.1 kHz) ; Movies (24 frames per second) to TV (60 fields/sec)

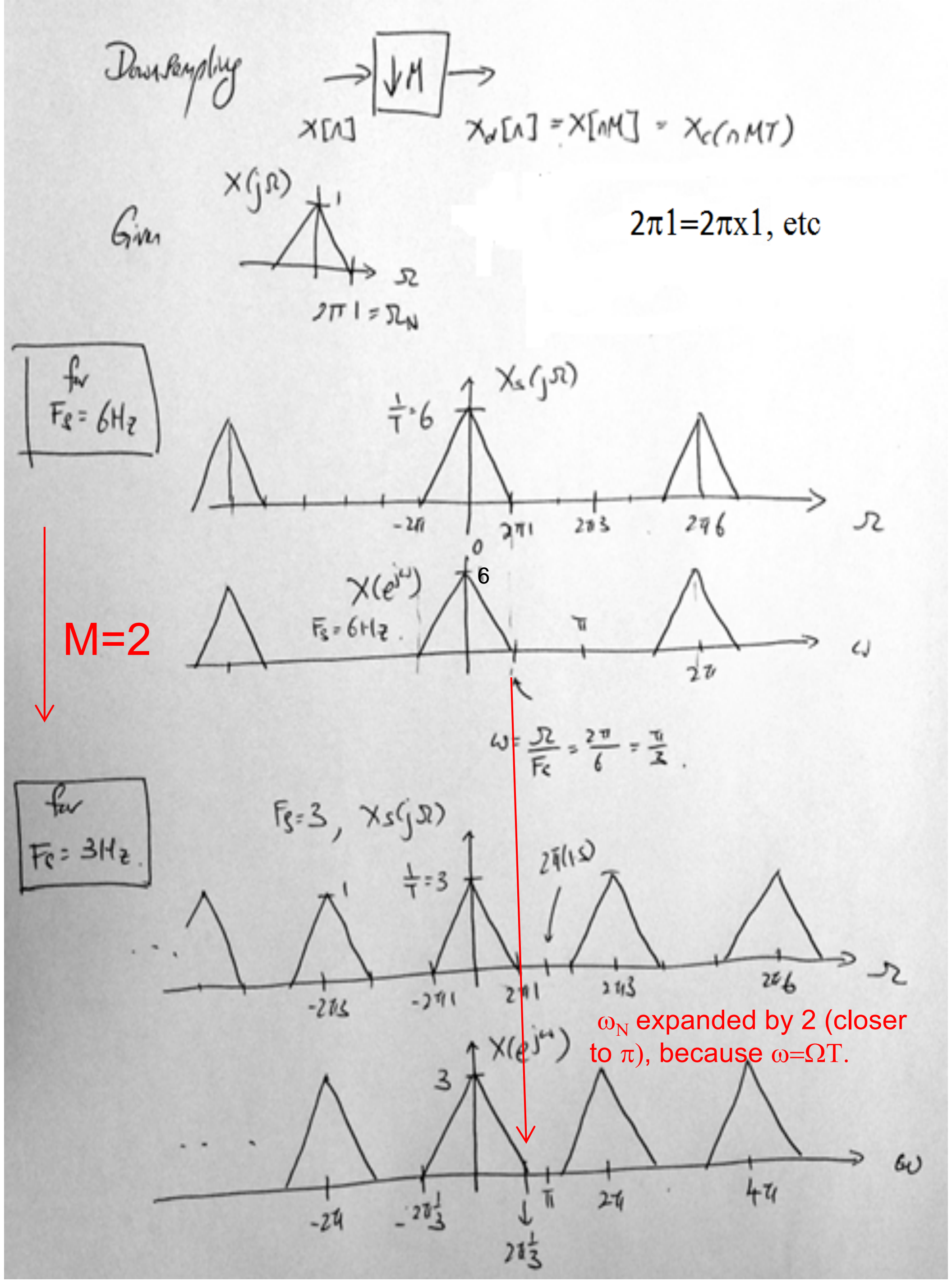
Manipulations in signal domain:



F. Down Sampling

An example (before presenting the general case):

Evaluate a signal of 1 Hz, sampled at 6 Hz and 3 Hz.



F. Down Sampling in Frequency domain

(to be further illustrated next with graphs)

Ref: Oppenheim (3rd edition) p. 209

- Discrete-time Fourier transform (DTFT) of $x[n] = x_c(nT_s)$

$$X(e^{j\omega}) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T_s} - \frac{2\pi k}{T_s}\right)\right) \quad \#$$

- Similarly, DTFT of $x_d[n] = x[nM] = x_c(nT'_s)$, $T'_s = MT_s$

$$X_d(e^{j\omega}) = \frac{1}{T'_s}\sum_{r=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T'_s} - \frac{2\pi r}{T'_s}\right)\right)$$

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

\# equivalent to $X(j\Omega) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s))$, $\Omega = \omega/T_s$, $\Omega_s = 2\pi/T_s$, as presented in Subpart C earlier.

\#\# equivalent to $X_d(j\Omega) = \frac{1}{T_s}\sum_{r=-\infty}^{\infty} X_c(j(\Omega - r\Omega_s/M))$, $\Omega = \omega/MT_s$.

F. Down Sampling in Frequency domain-a general case (graphic illustration)

Down sampling: further reduces the actual sampling frequency→ aliasing

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

Fourier Transform $x_c(t)$

Fourier Transform $x_s(t)$

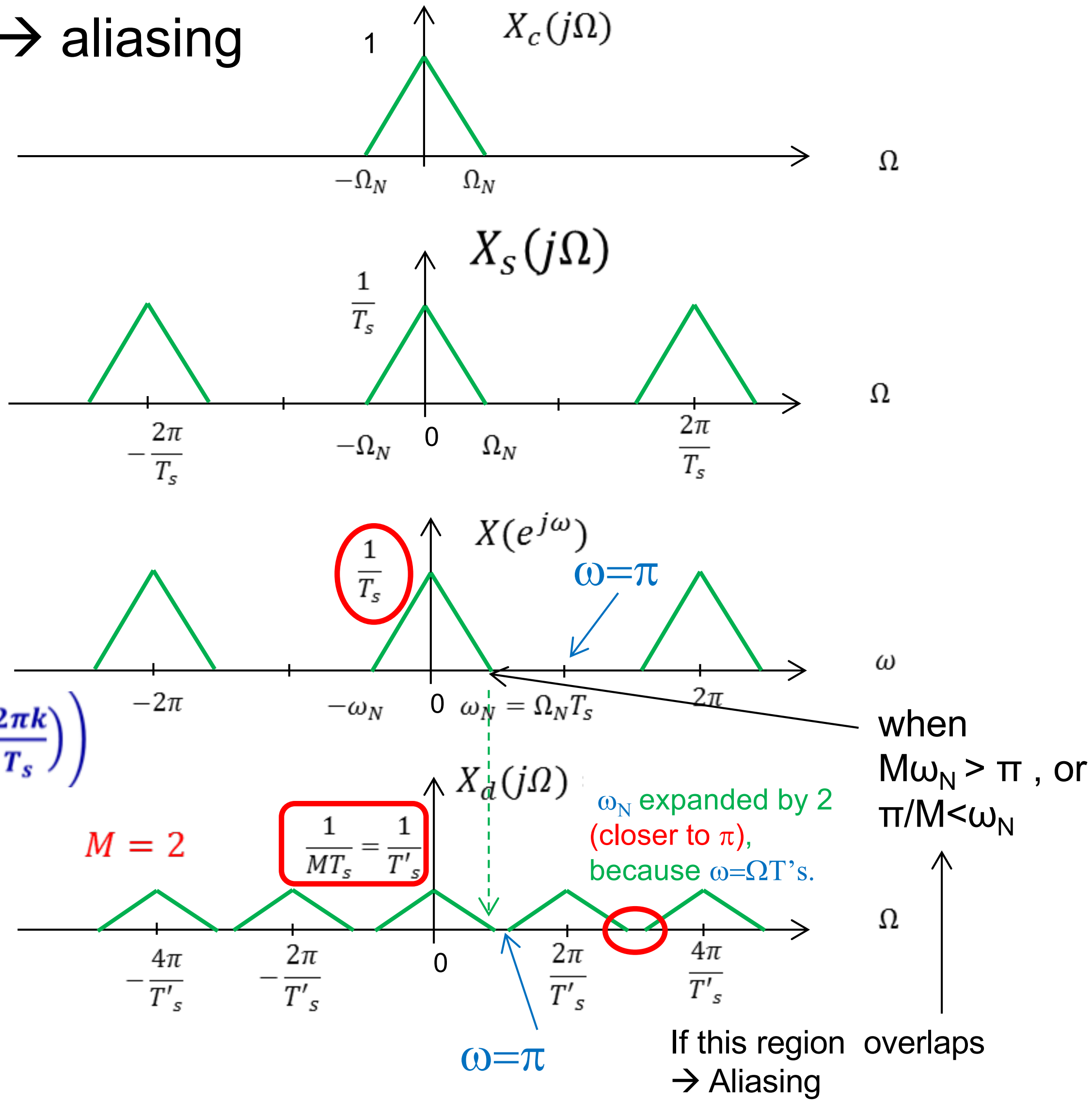
Before down sampling:

Fourier Transform $x[n]$

$$X(e^{j\omega}) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T_s} - \frac{2\pi k}{T_s}\right)\right)$$

After down sampling by 2:

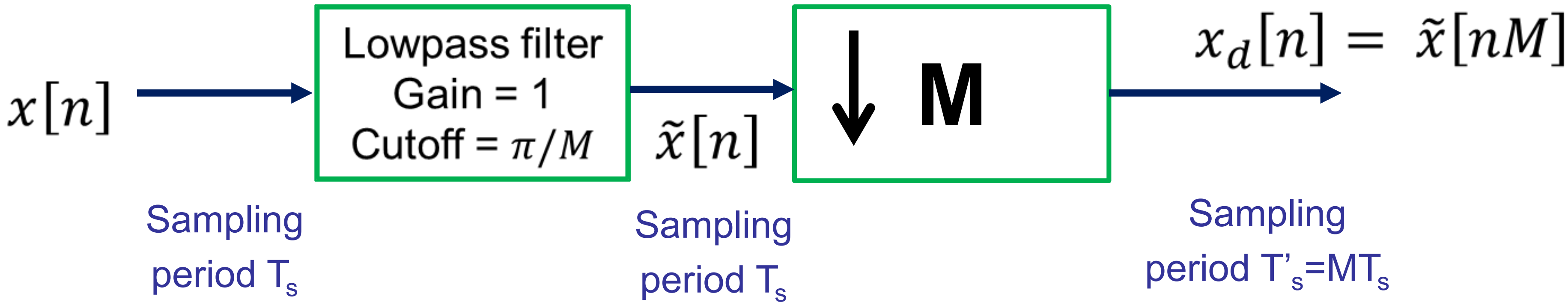
Fourier Transform $x_d[n]$ **with** $M = 2$



F. Downsampling with prefiltering to avoid aliasing

See previous graphs

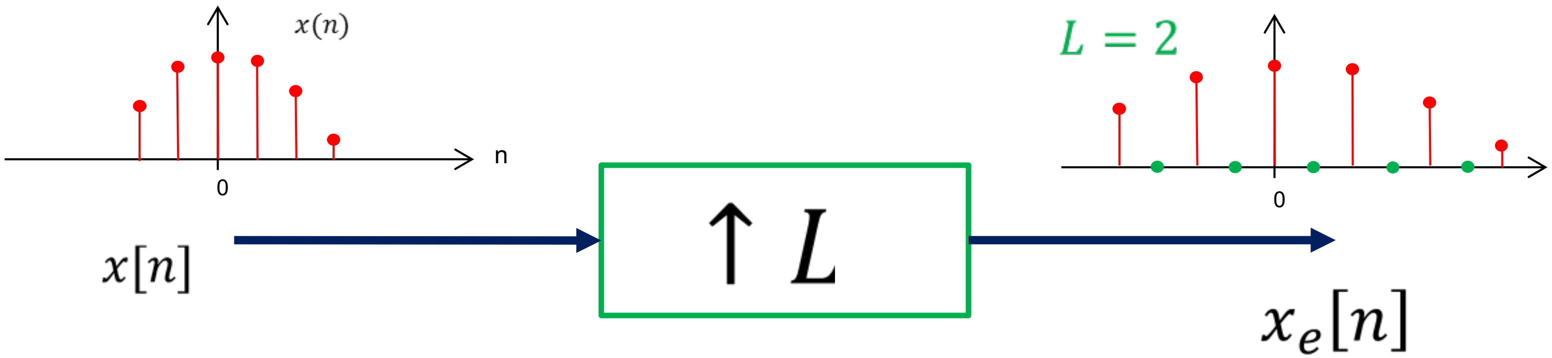
- To avoid aliasing, we need $\omega_N < \pi/M$ (why?), where ω_N is the highest frequency of the discrete-time signal $\boldsymbol{x}[\boldsymbol{n}]$.



- Hence, downsampling is usually accompanied with a pre-low-pass filtering, and a low-pass filter followed by down-sampling is usually called a decimator, and termed the process as decimation.

F. Up-Sampling

- Increase sampling rate by an integer factor, L

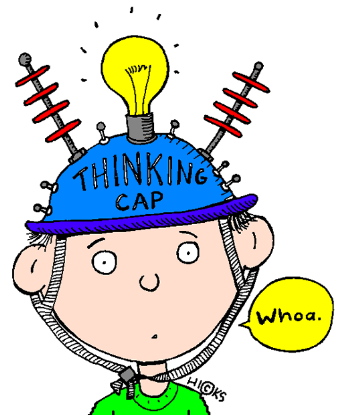


Sampling period T_s

Sampling period $T'_s = T_s/L$

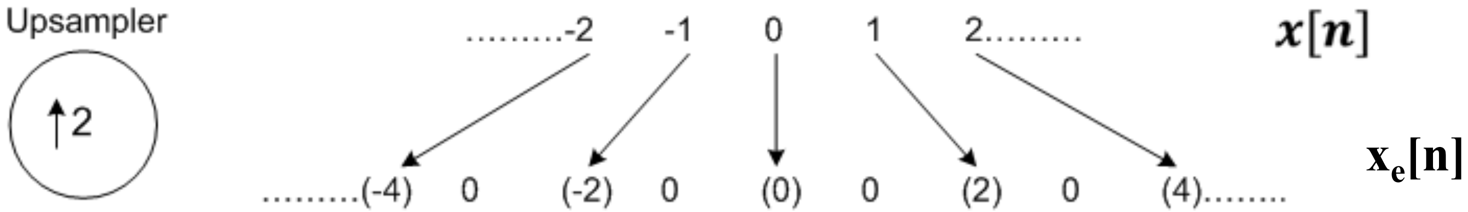
$$x_e[n] = \begin{cases} x[n/L], n = 0, \pm L, \pm 2L, \ldots \\ 0, \quad otherwise \end{cases}$$

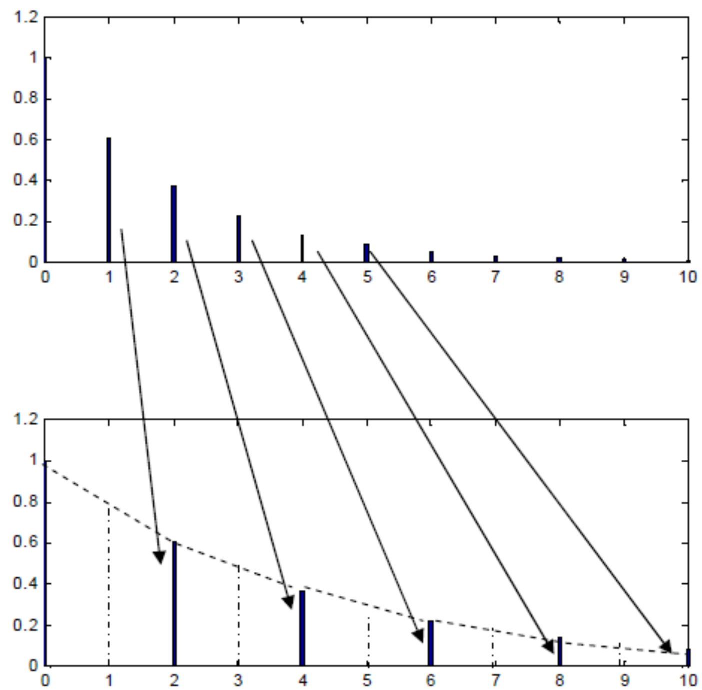
Equivalently $x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]$



- $x[n]=x_c[nT_s]$ then $x_e[n]=x[n/L]=x_c[nT_s/L]$

Manipulations in signal domain:





F. Up-sampling in Frequency domain

- Discrete-time Fourier transform (DTFT) of $\boldsymbol{x[n]}$

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

- Similarly, DTFT of $\boldsymbol{x_e[n]}$:

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

Actually, compresses (because L>1)

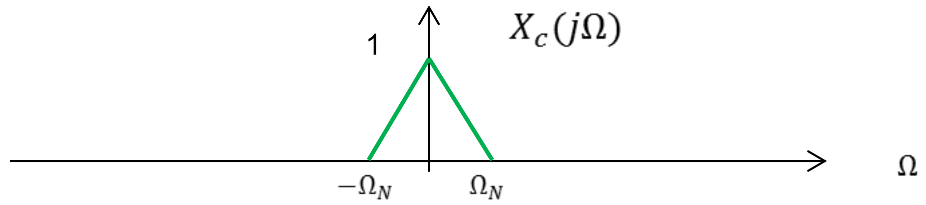
$$= \sum_{k=-\infty}^{\infty} x[k]e^{-j\omega kL} = X\left(e^{j\omega L}\right)$$

- This scales the frequency axis by a factor of L since $\omega' = \omega L$

F. Up-sampling in Frequency domain

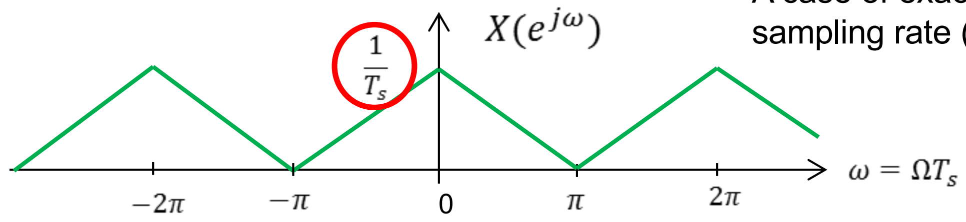
- Illustration

Fourier Transform $x_c(t)$



Before up sampling:

Fourier Transform $x[n]$

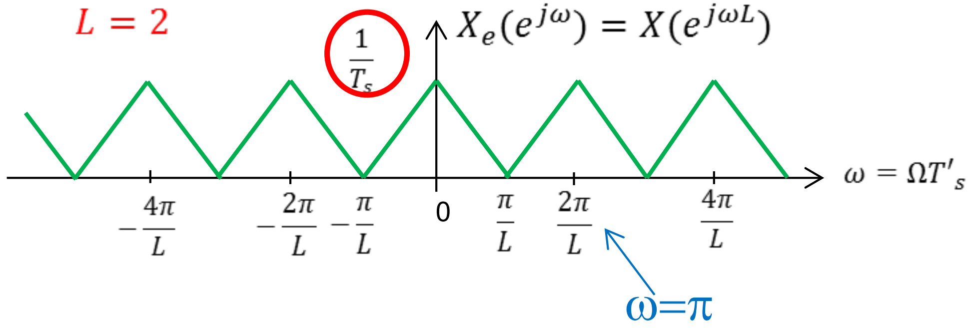


A case of exact Nyquist sampling rate ($\pi/T_s=\Omega_N$)

After up sampling by 2:

Fourier Transform $x_e[n]$ **with** $L = 2$

$L = 2$



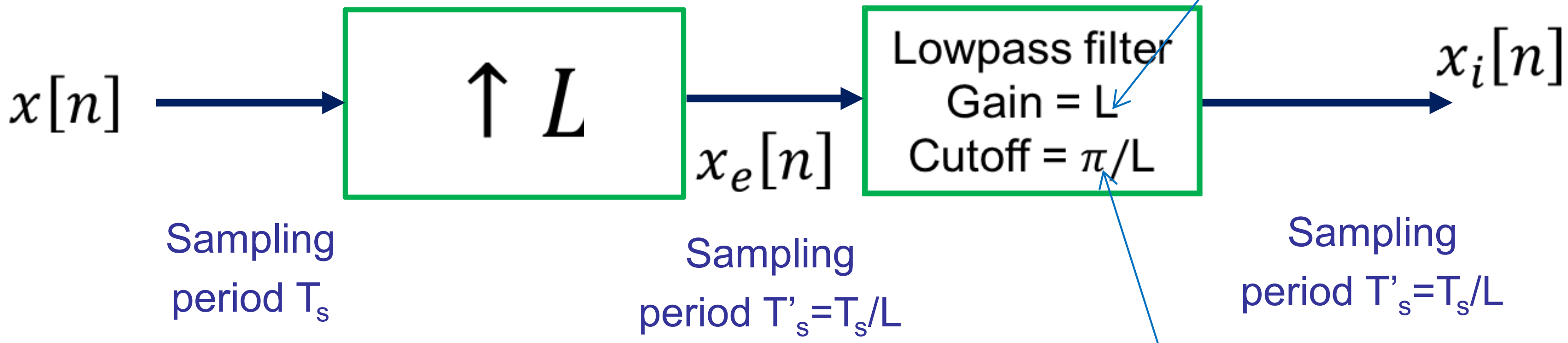
up sampling: compress L times within any interval of 2π.

F. Up-sampling with post lowpass filtering

- Similar to the case of D/C converter, upsampling is usually accompanied with a post low-pass filter with cutoff frequency π/L and gain L, to reconstruct the sequence.

Explain in the next slide

With important parameters indicated:



- A low-pass filter followed by up-sampling is called an interpolator, and the whole process is called interpolation.

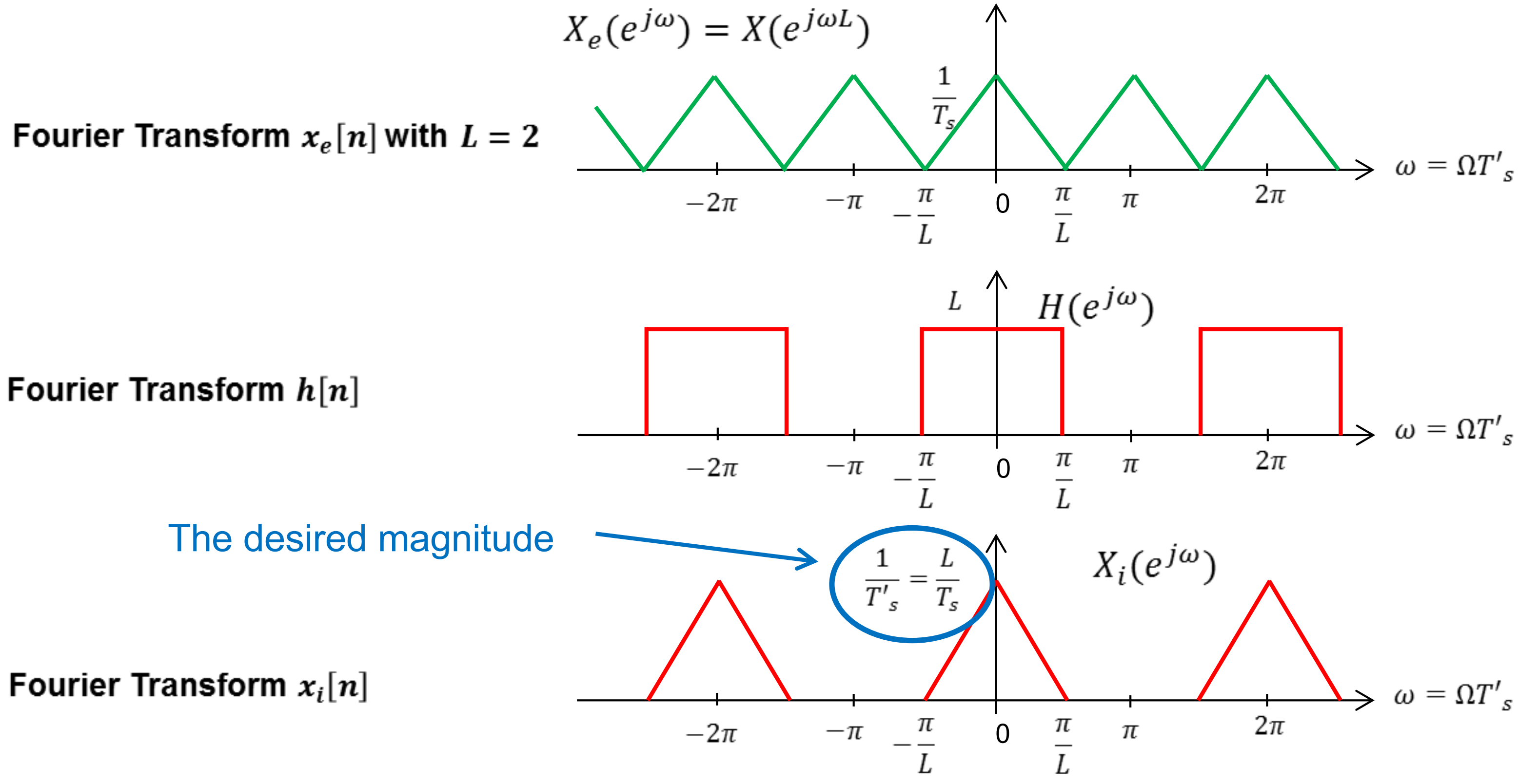
See previous graphs

F. Up-sampling with post lowpass filtering

- Example

$$x[n] = x_c(nT) \Rightarrow x_i[n] = x_c(nT_i), T_i = T/L$$

$$\Rightarrow x_i[n] = x_c(nT/L) = x[n/L]$$



The desired filter gain is L: $L(1/T_s)=L/T_s$

F. Interpolation

- Similar to the ideal D/C converter, #
  - If we choose an ideal lowpass filter with cutoff frequency π/L and gain L, its impulse response is

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

(Sinc filter: already learnt in Subpart D for the continuous case)

  - Then

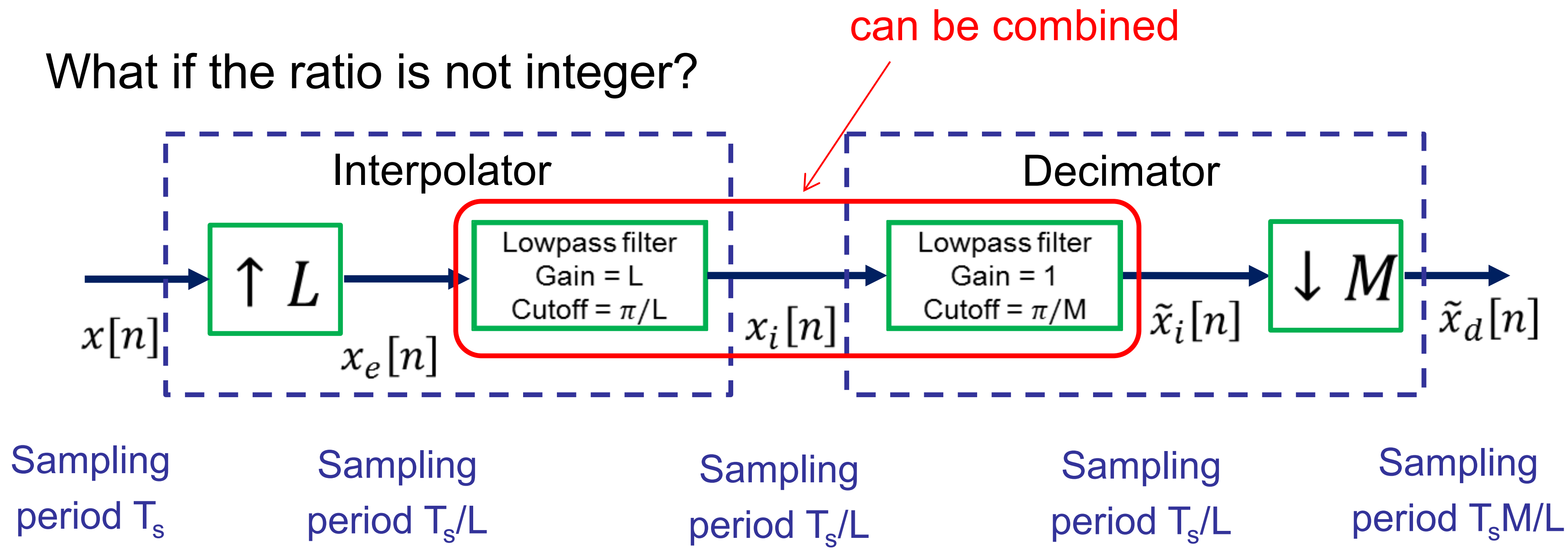
$$x_i[n] = x_e[n] * h_i[n] = \left(\sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]\right) * h_i[n]$$

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

up-sampling actually “approaches” to the continuous case.

F. Sample Rate Conversion (non-integer)

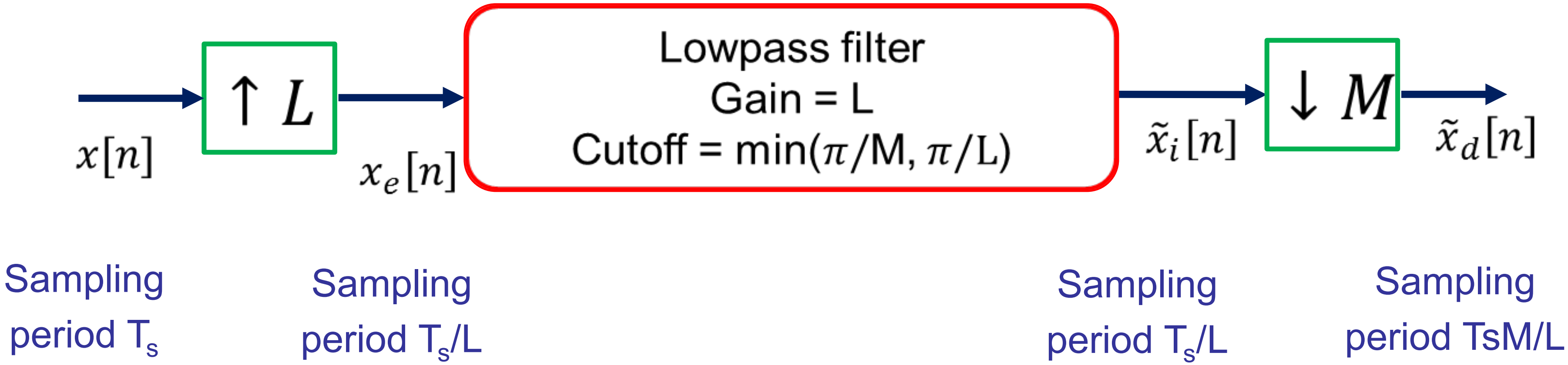
- What if the ratio is not integer?



- By combining the decimation and interpolation, we can change the sampling rate of a sequence.
  - Changing the sampling rate by a non-integer factor $T' = T_sM/L$.
  - Eg., L=100 and M=101, then T' = 1.01T.

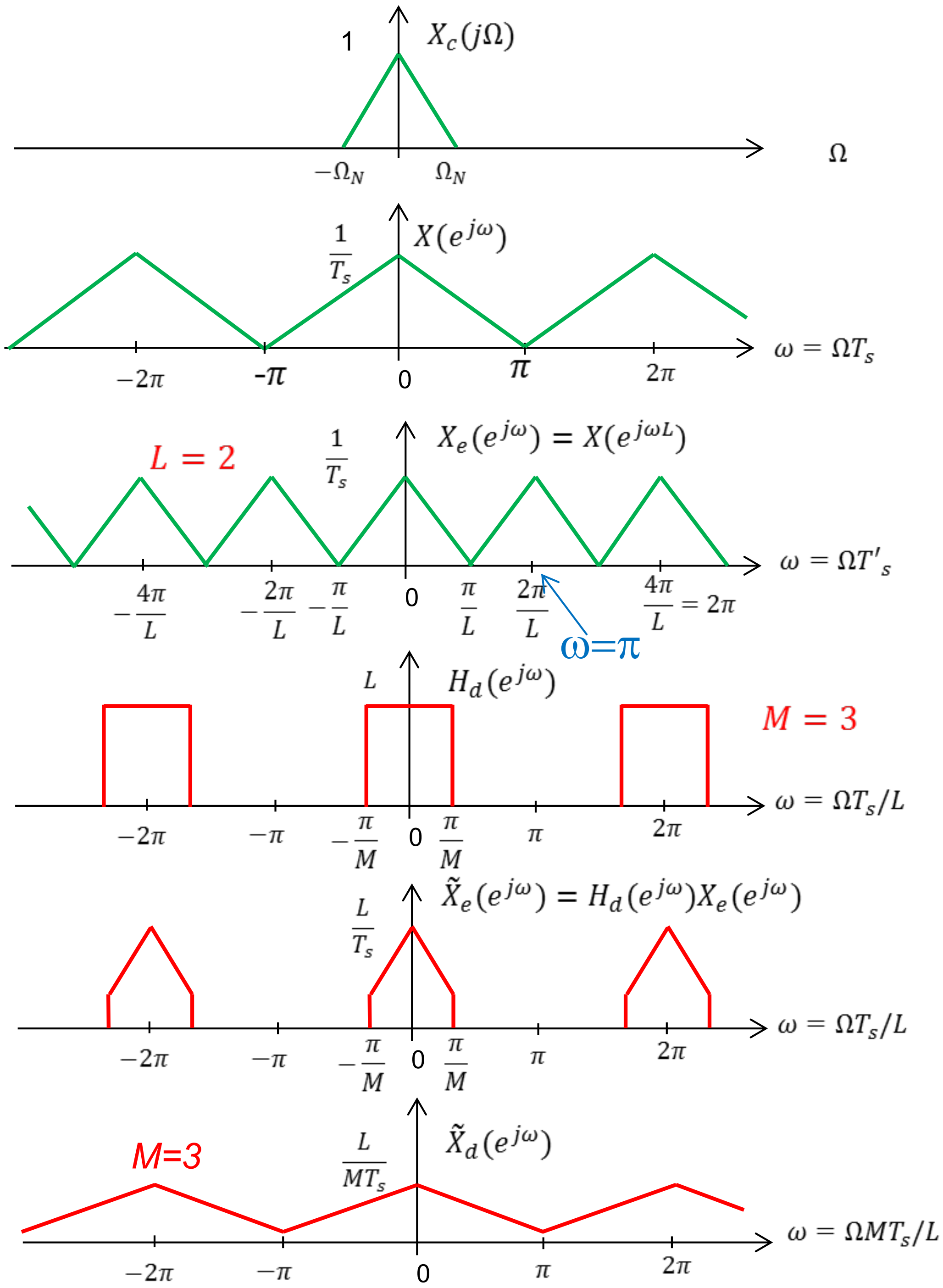
F. Sample Rate Conversion

- Since the interpolation and decimation filters are in cascade, they can be combined as:



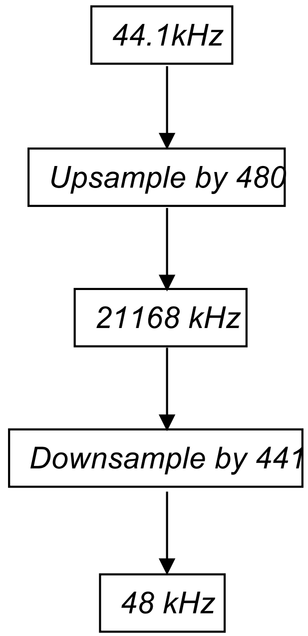
F. Example

In this example,
L=2, M=3,
Therefore, the overall
Up-down sampling
is to sample at
2/3 of the original
sampling frequency.

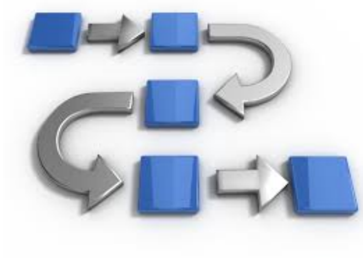


F. Sample Rate Conversion

- E.g., to go from 44.1kHz to 48kHz
- Upsample by 1.0884
- This is 480/441
- We can upsample by 480 and downsample by 441!



Methodology/Logic for Sampling and Reconstruction

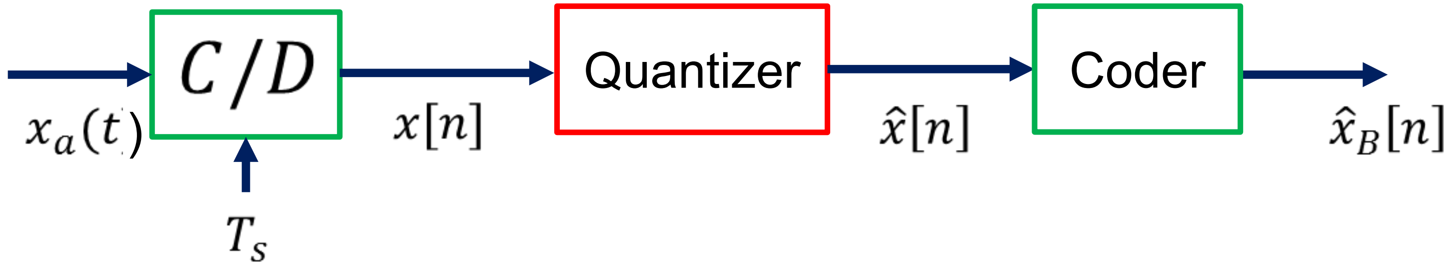


A. Overview sampling – a conversion process
B. Sampling theorem & aliasing
C. A mathematical model of sampling in frequency domain
D. Reconstruction
E. Discrete time processing of continuous time signals
F. Up and down sampling
G. Quantization

G. Quantization

- After C/D block, the amplitude of each sample is still continuous, the quantization process will convert this continuous amplitude into a discrete number to store as a code for digital processing.

$$\hat{x}[n] = Q(x[n])$$



- A quantizer is a nonlinear system: Q(ax+by) ≠ aQ(x)+bQ(y)

G. Quantization

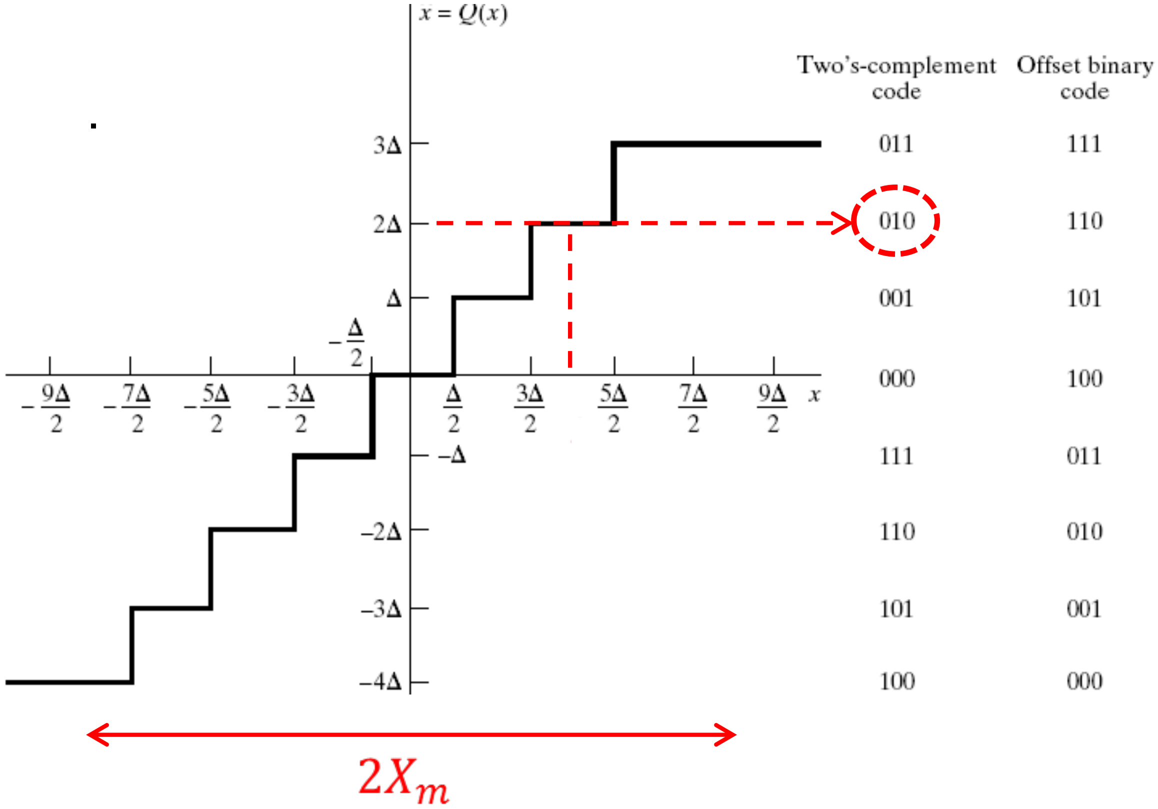
- We usually use two's complement code for representation

X_m is called the full- scale level of A/D. Example: 1Volt Then, step size is:

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

Binary symbol	Numeric value, $\hat{x}_B$
$0_\diamond 1\,1$	$3/4$
$0_\diamond 1\,0$	$1/2$
$0_\diamond 0\,1$	$1/4$
$0_\diamond 0\,0$	0
$1_\diamond 1\,1$	$-1/4$
$1_\diamond 1\,0$	$-1/2$
$1_\diamond 0\,1$	$-3/4$
$1_\diamond 0\,0$	-1

X_m=1, B=2, Δ=1/4.



G. Quantization

- Example

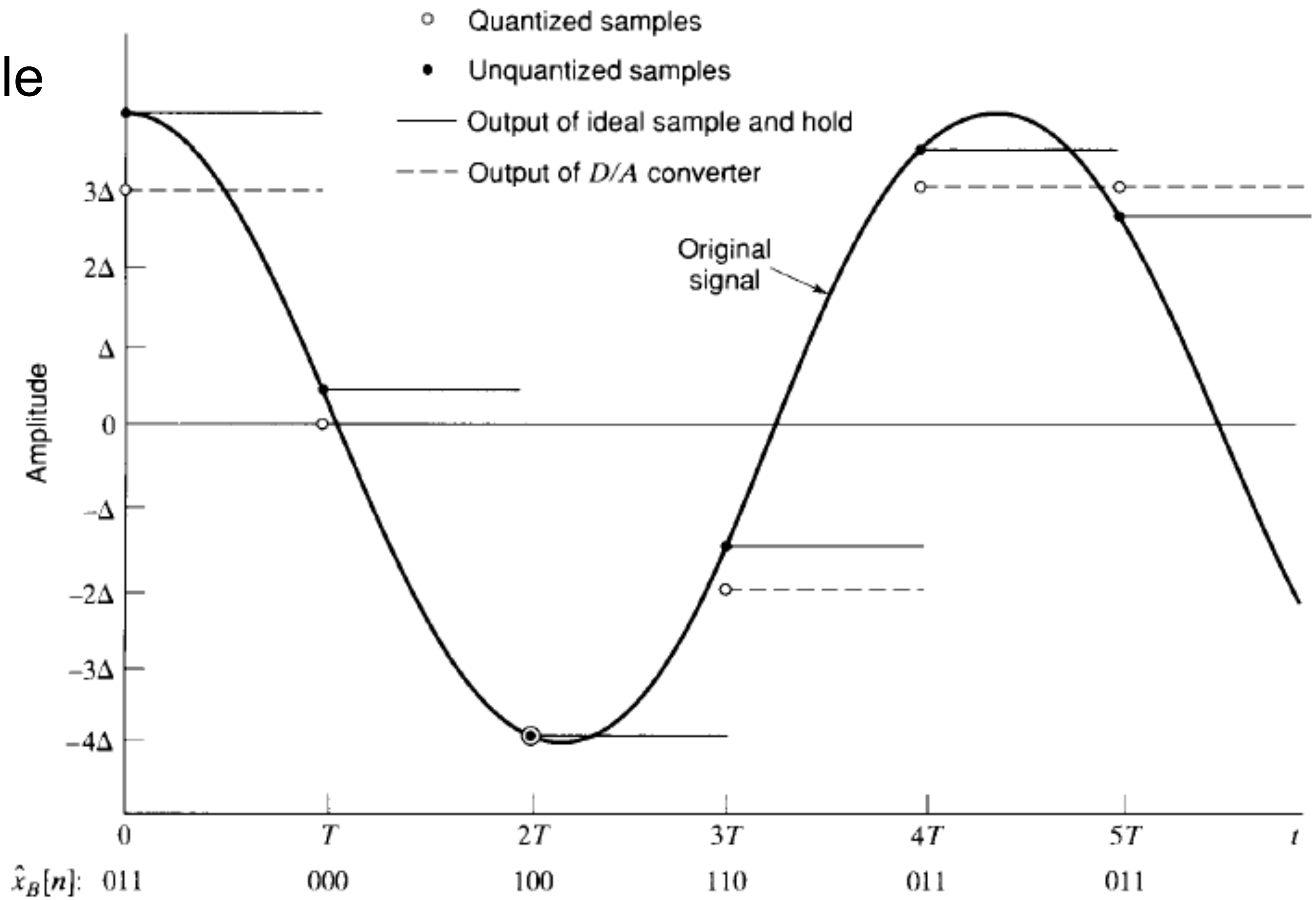


Figure Sampling, quantization, coding, and D/A conversion with a 3-bit quantizer.

G. Quantization error

- Quantization error is the actual difference between original amplitude with the quantized one

$$e[n] = \hat{x}[n] - x[n]$$

- In general, for a (B+1)-bit quantizer with step size Δ, the quantization error satisfies that

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

$$\text{when } \left(-X_m - \frac{\Delta}{2}\right) < x[n] < \left(X_m - \frac{\Delta}{2}\right)$$

- If x[n] is outside this range, then the quantization error is larger in magnitude than Δ/2, and such samples are said to be clipped.