

Changing sampling rate

Lecture 5

Changing the sampling rate

Given the sequence

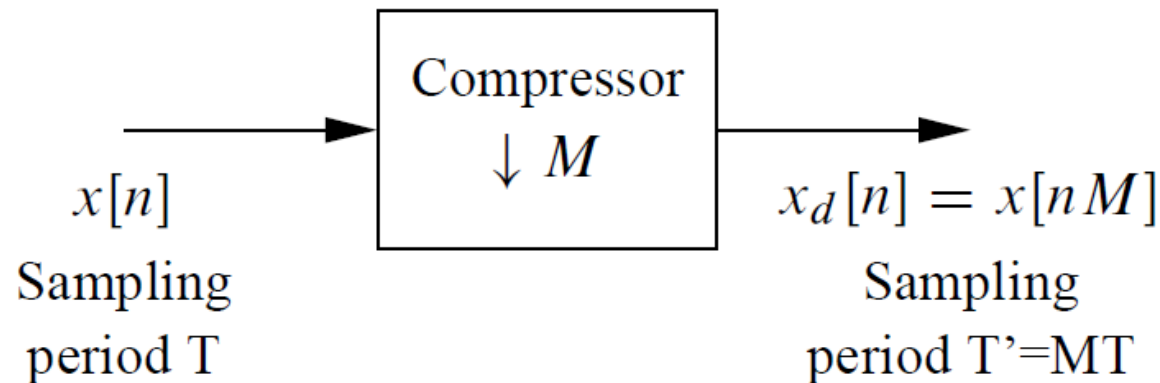
$$x[n] = x_c(nT)$$

obtained by sampling (with period T) the signal $x_c(t)$, we often want to change the sampling rate (to period T'):

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

One approach is to reconstruct $x_c(t)$ from $x[n]$, and then resample with new period T' . However, we want to do this using only discrete-time operations.

Sampling rate reduction by an integer factor



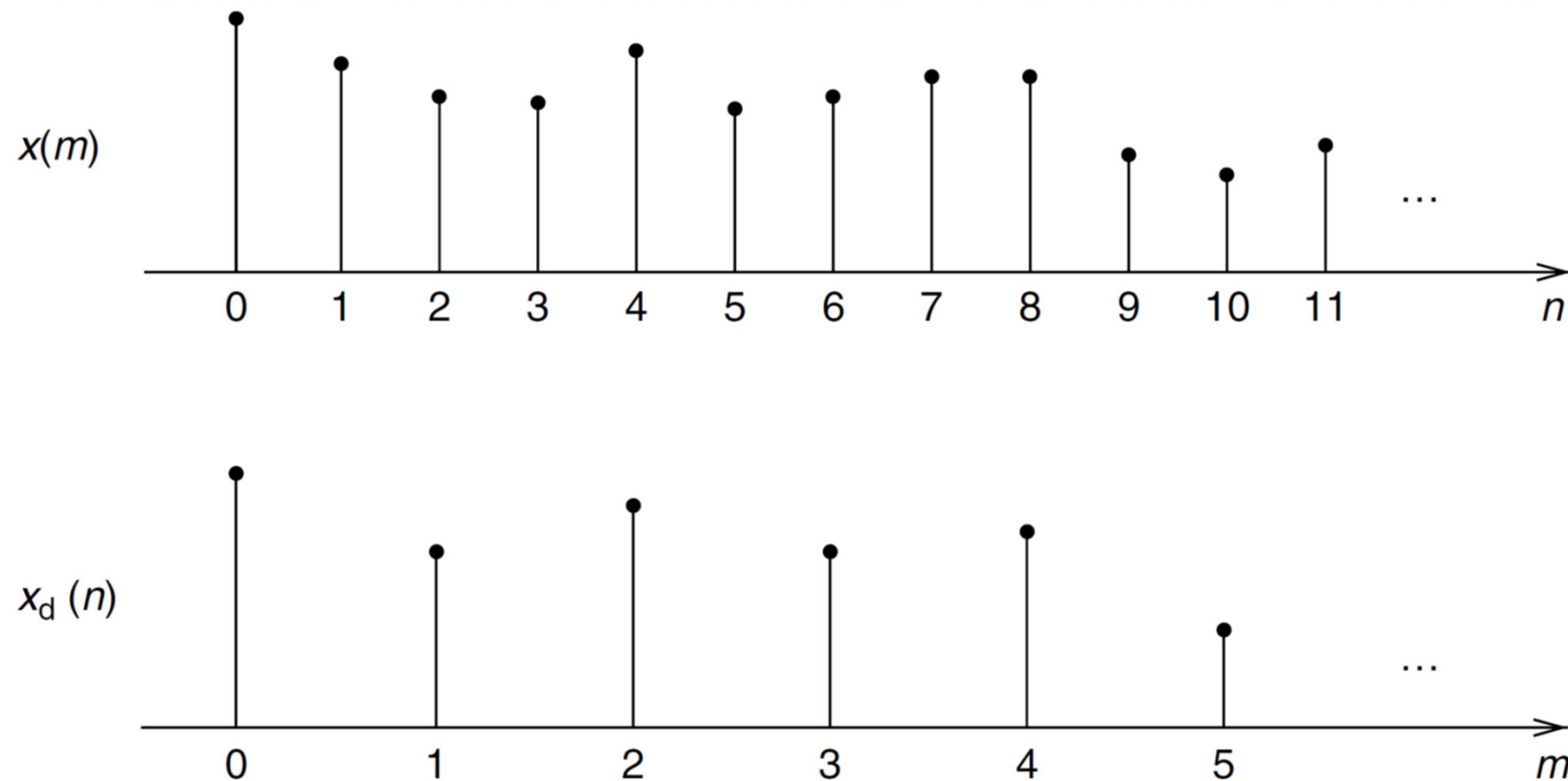
The sampling rate **compressor** implements the following function:

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

Here $x_d[n]$ is exactly the sequence that would be obtained by sampling $x_c(t)$ with period $T' = MT$.

If $X_c(j\Omega) = 0$ for $|\Omega| \geq \Omega_N$, then $x_d[n]$ is an exact (unaliased) representation of $x_c(t)$ if $\pi/(MT) \geq \Omega_N$.

Decimation (Downsampling)



Frequency domain

- Re-normalizing frequency to the new sample rate:

$$\omega = \frac{\Omega}{f_c} = \frac{\Omega}{\frac{1}{T}} = \Omega T$$

$$\omega' = \frac{\Omega}{f_c/M} = \frac{\Omega}{\frac{1}{MT}} = \Omega MT = M \omega$$

Frequency domain

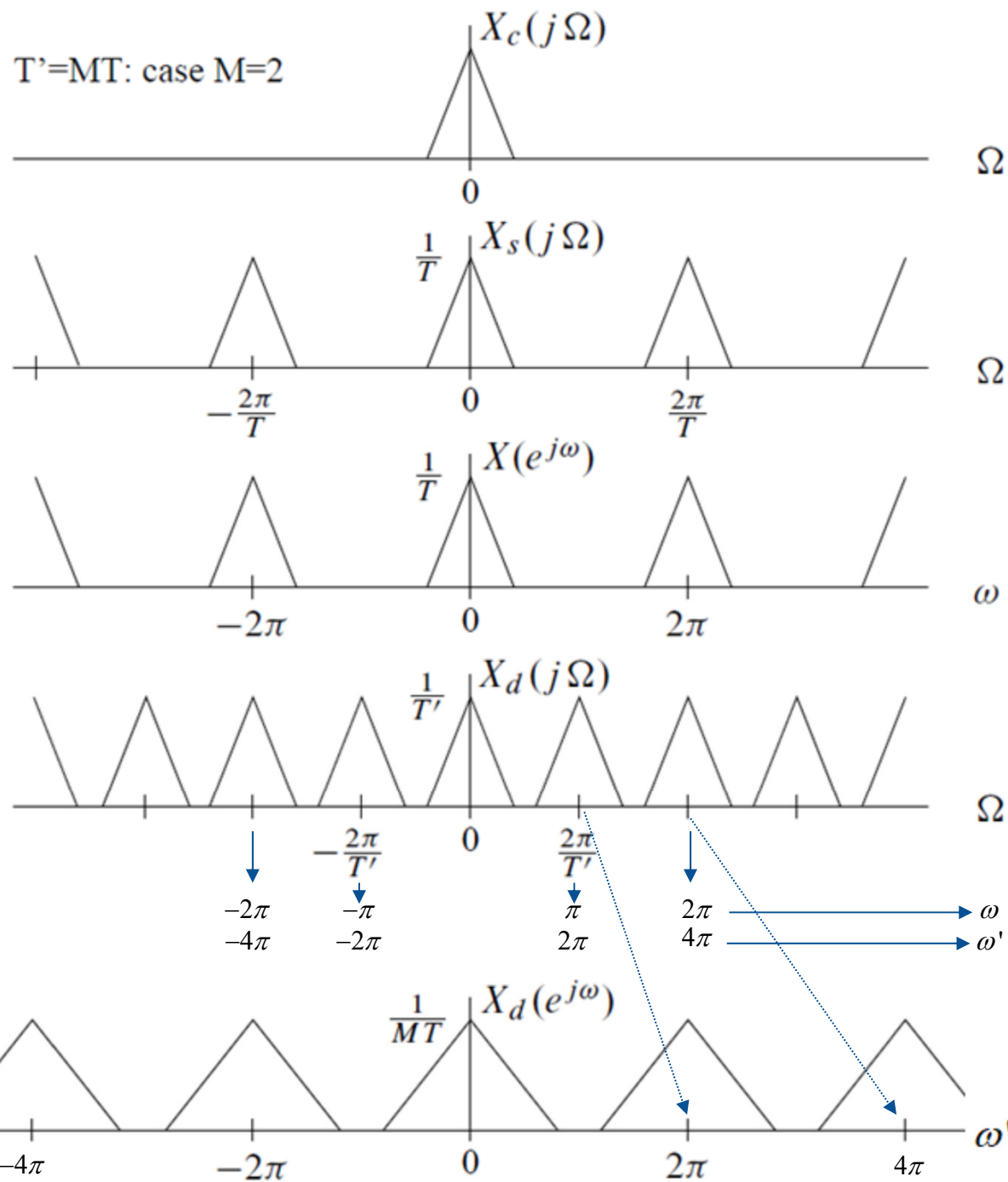
$$x_d(n) = x(nM).$$

- Since:
$$x'(m) = x(m) \sum_{n=-\infty}^{\infty} \delta(m - nM)$$

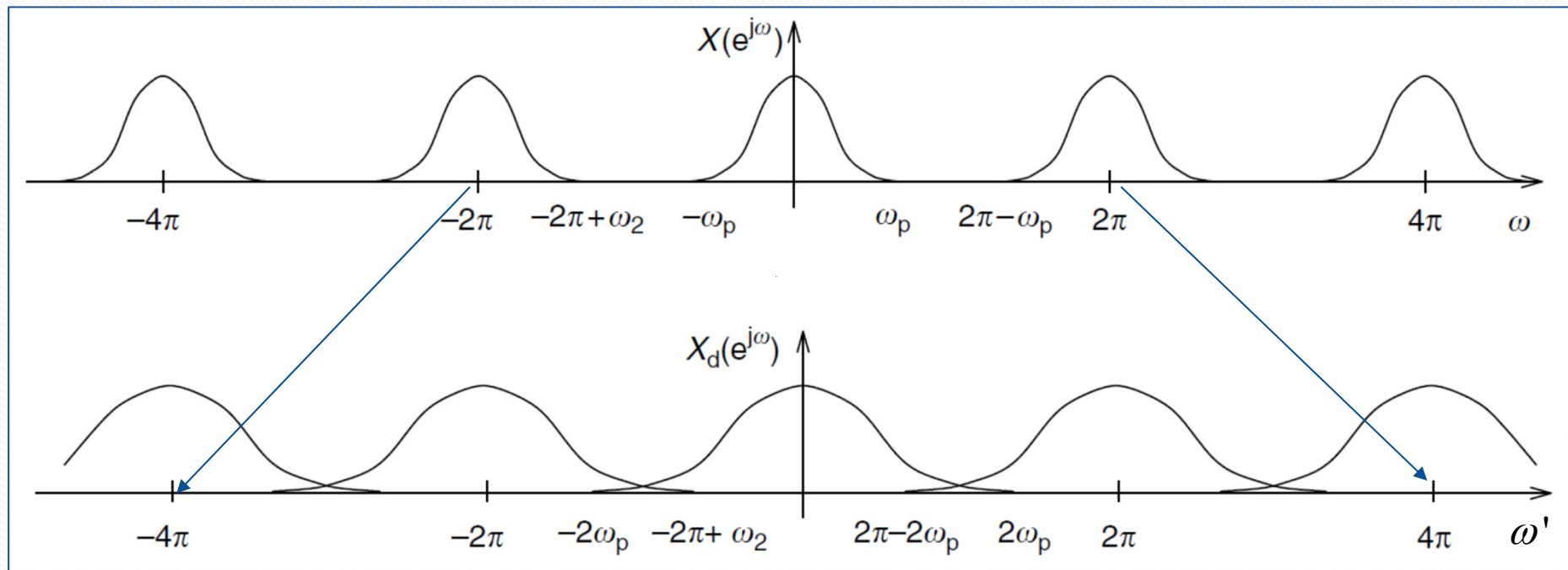
$$X'(e^{j\omega}) = X(e^{j\omega}) \circledast \mathcal{F} \left\{ \sum_{n=-\infty}^{\infty} \delta(m - nM) \right\}$$

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

$$= \frac{1}{M} \sum_{k=0}^{M-1} X \left\{ e^{j[\omega - (2\pi k/M)]} \right\}.$$

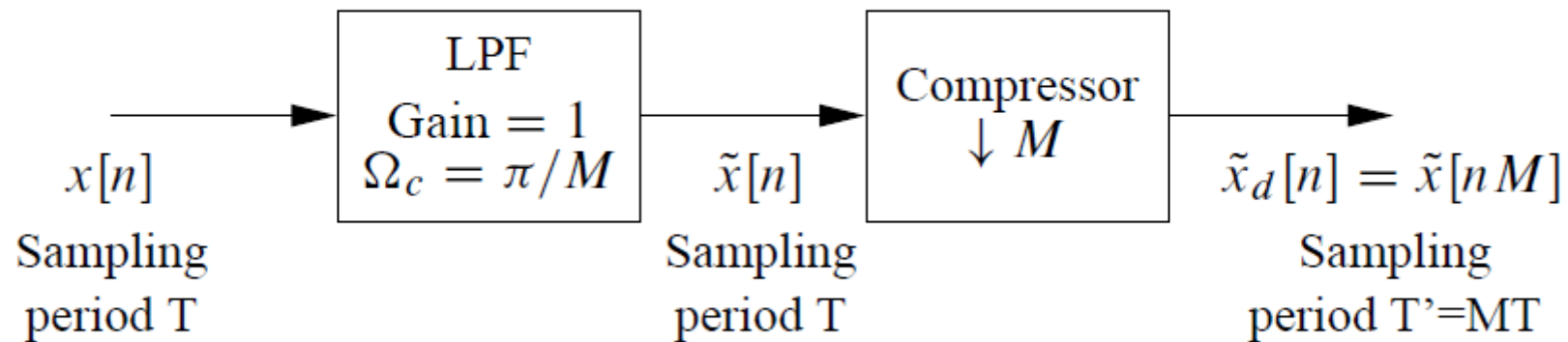


Aliasing from downsampling



Avoid aliasing

Applying a compressor to a signal can result in aliasing. This can be avoided (at the cost of some information) by prefiltering with a lowpass filter, and then compressing the sampling rate:



This is referred to as **downsampling (or decimation) by a factor M** .

Increasing sampling rate

With underlying continuous-time signal $x_c(t)$, we want to obtain samples

$$x_i[n] = x_c(nT')$$

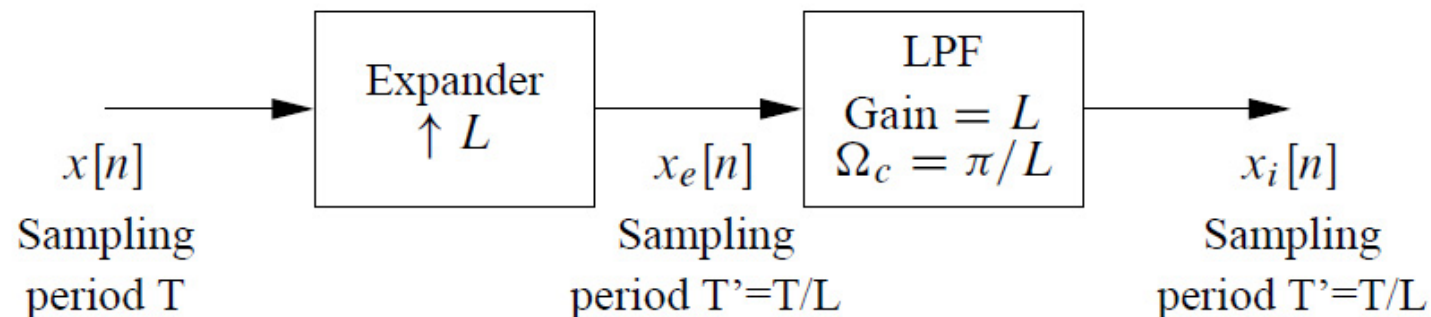
from

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

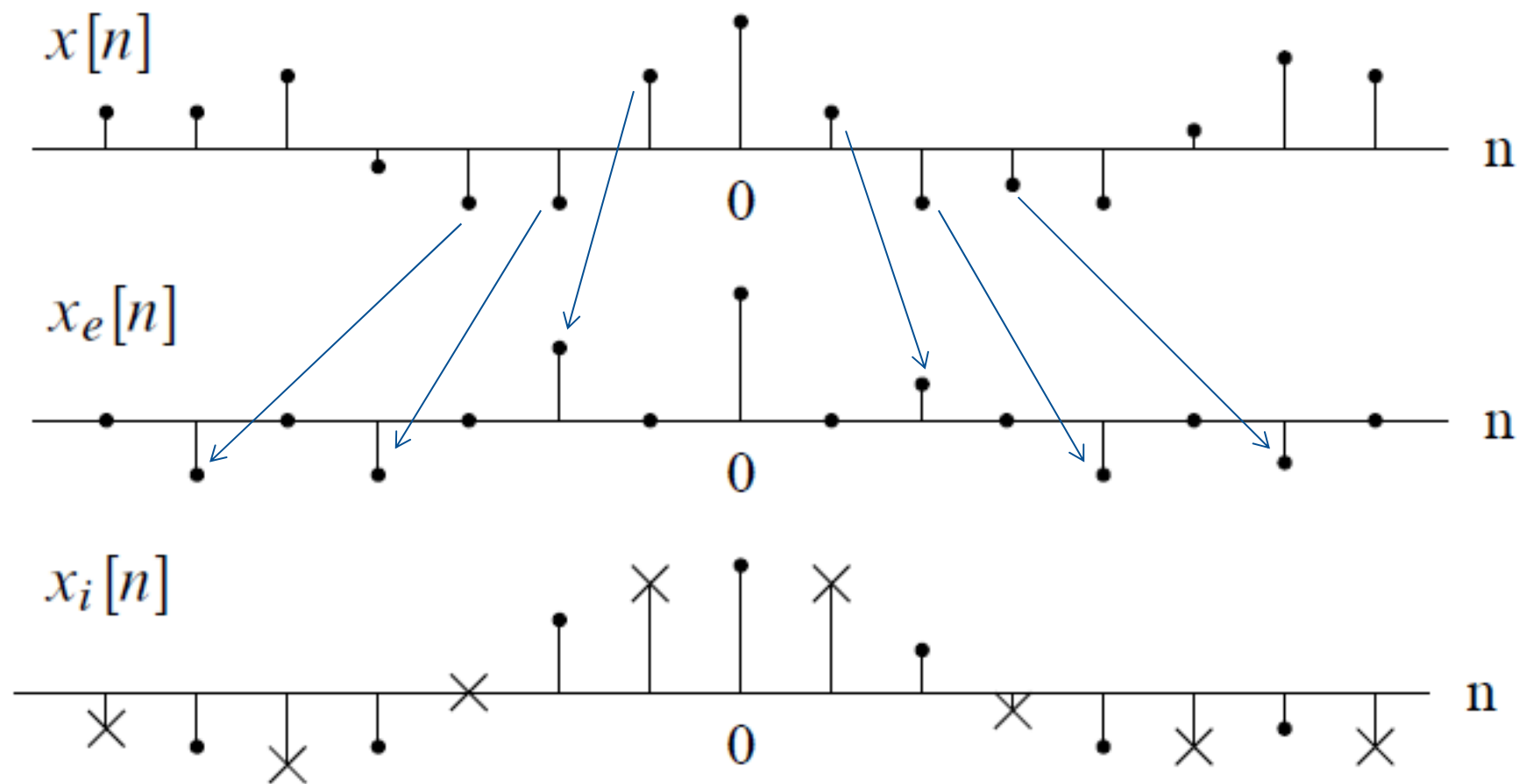
where $T' = T/L$. Therefore

$$x_i[n] = x[n/L] = x_c(nT/L), \quad n = 0, \pm L, \pm 2L, \dots$$

This is referred to as **upsampling (or interpolating) by a factor L** , and is performed by **expanding** the sampling rate, and then lowpass filtering:



Upsampling example



Expanded signal

The expanded signal is

$$x_e[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \dots, \\ 0, & \text{otherwise,} \end{cases}$$
$$= \sum_{k=-\infty}^{\infty} x[k] \delta[n - kL].$$

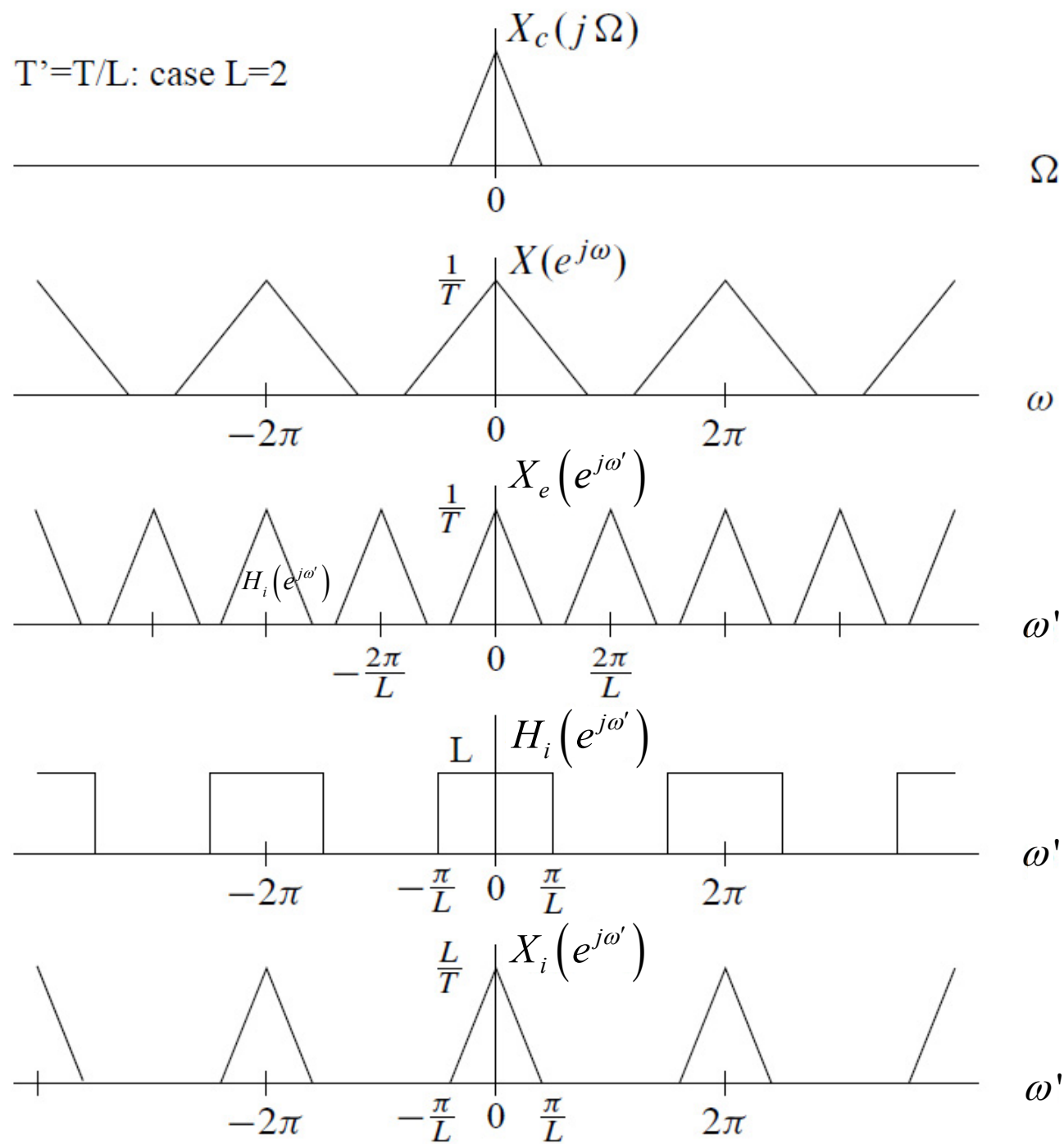
$$\omega = \frac{\Omega}{f_c} = \frac{\Omega}{\frac{1}{T}} = \Omega T$$
$$\omega' = \frac{\Omega}{Lf_c} = \frac{\Omega}{\frac{L}{T}} = \frac{\Omega}{L} T = \frac{\omega}{L}$$

Fourier transform of the expanded signal

The Fourier transform of the expanded signal is

$$\begin{aligned} X_e(e^{j\omega'}) &= \sum_{n=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} x[k] \delta[n - kL] \right) e^{-j\omega'n} = \\ &= \sum_{k=-\infty}^{\infty} x[k] e^{-j\omega'Ln} = X(e^{j\omega'L}). \end{aligned}$$

Final upsampling is obtained by lowpass filtering the expanded signal.



Interpolation formula

We can obtain an interpolation formula for $x_i[n]$ in terms of $x[n]$: since the LPF has impulse response

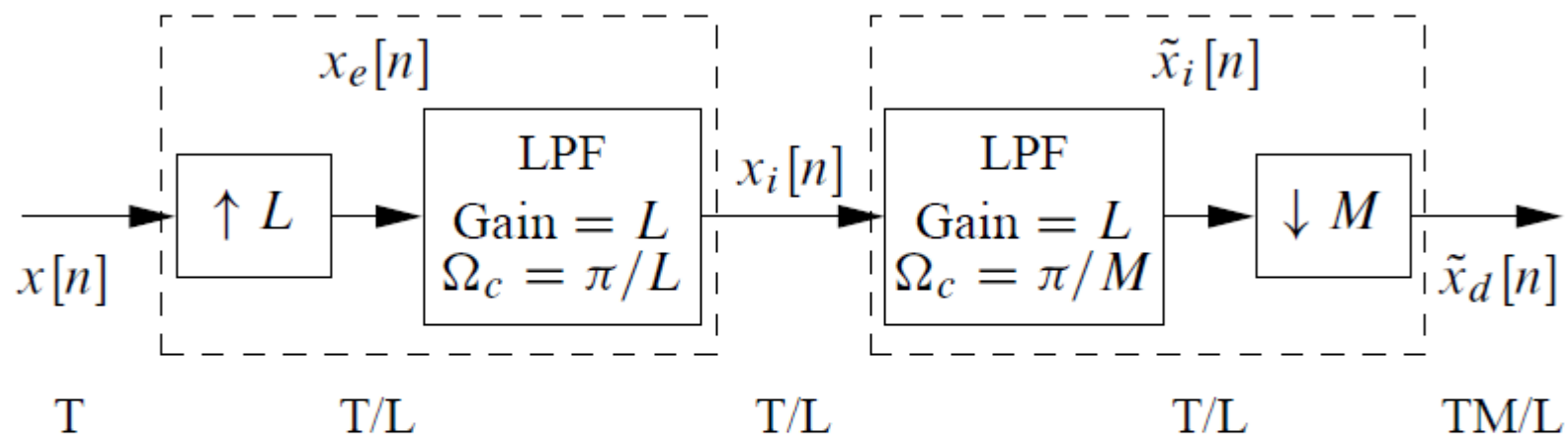
$$h_i[n] = \frac{\sin(\pi n/L)}{\pi n/L},$$

we have

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

Changing sample rate by a noninteger factor

By cascading upsampling (by factor L) and downsampling (by factor M), the sampling rate can be changed by a noninteger factor.



This forms the basis of **multirate signal processing**, where highly efficient structures are developed for implementing complicated signal processing operations. The discrete wavelet transform (DWT) can be developed in this framework.