Digital Signal Processing
Sampling-II

Lecture-8
12-April-16

Sample Rate Conversion

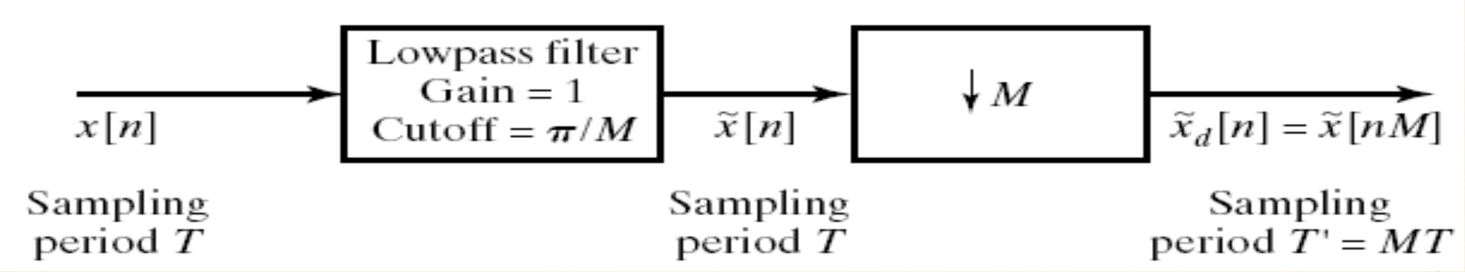
- The process of converting a signal form one rate to another is called sample rate conversion.
- Band limited interpolation can be used to go back to the continuous time signal from its samples.
- We need to change sampling rate for few applications and to obtain a new discrete time representation of the same continuous time signal of the form: $x'[n]=x_c(nT')$ where $T \neq T'$.
- The problem is to get $x'[n]$ given $x[n]$.
- One way is to:
  - Reconstruct the continuous time signal from $x[n]$.
  - Resample the continuous time signal using new rate to get $x'[n]$.
  - This requires analog processing which is often undesired.

Sample Rate Reduction by an Integer Factor

- We reduce the sampling rate of a sequence by "sampling" it:

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

- This is accomplished with sampling rate compressor:



- We obtain $x_d[n]$ that is identical to what we would get by reconstructing the signal and resampling it with T'=MT.

- There will be no aliasing if: $\frac{\pi}{T'} = \frac{\pi}{MT} > \Omega_N$

Sample Rate Reduction by an Integer Factor (cont.)

- The sampling rate can be reduced by a factor of M without aliasing if the original sampling rate was at least M times the Nyquist rate or if the bandwidth of the sequence is first reduced by a factor of M by discrete time filtering.
- Such an operation of reducing the sampling rate is known as Down sampling.

Frequency Domain Representation of Down Sampling

- Recall the DTFT of $x[n]=x_c(nT)$:

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

- The DTFT of the down sampled signal can similarly written as:

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

- Let's represent the summation index as:

$$r=i+kM \quad where \quad -\infty<k<\infty \quad and \quad 0\le i<M$$

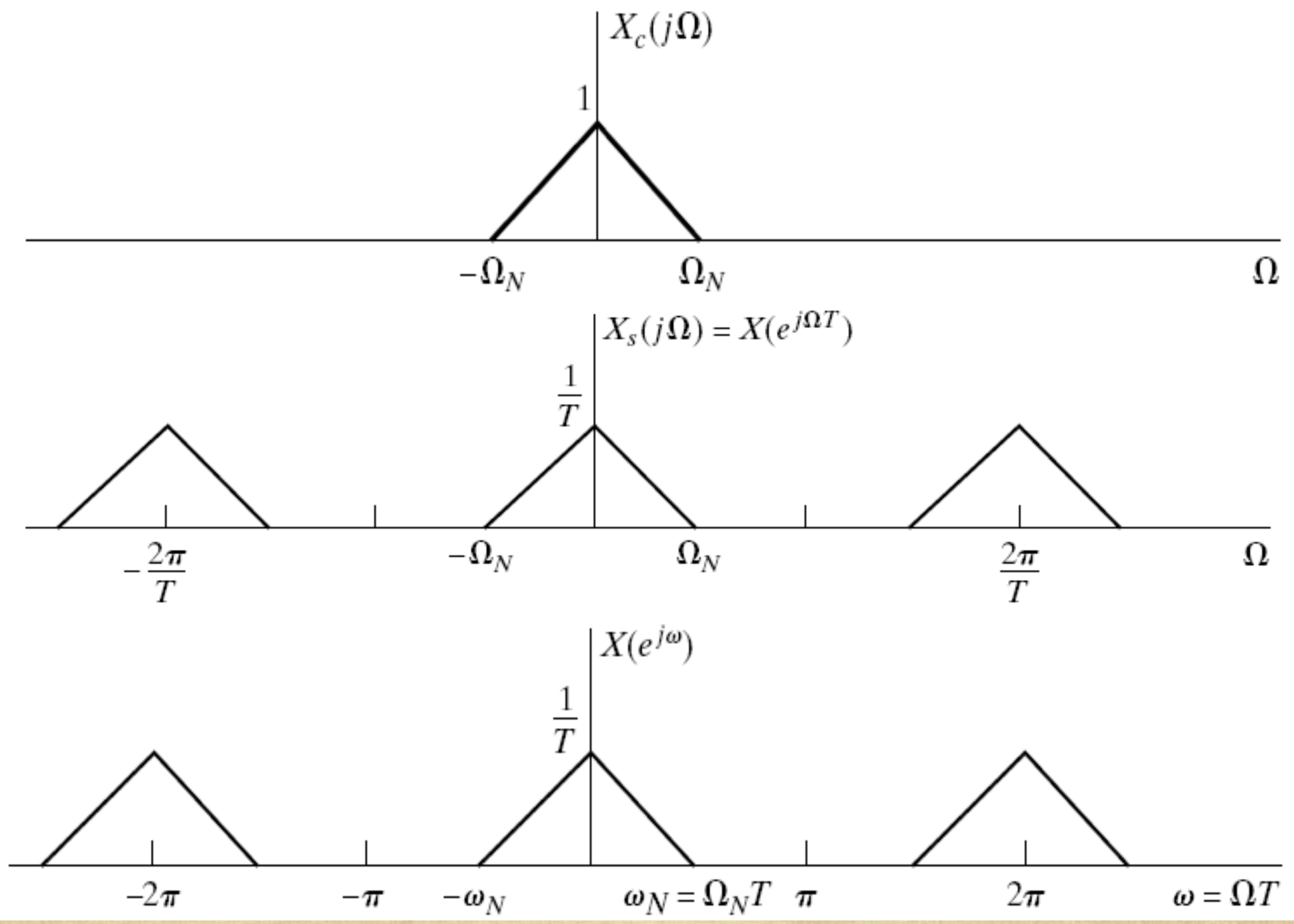
$$X_d\left(e^{j\omega}\right)=\frac{1}{M}\sum_{i=0}^{M-1}\left[\frac{1}{T}\sum_{r=-\infty}^{\infty}X_c\left(j\left(\frac{\omega}{MT}-\frac{2\pi k}{T}-\frac{2\pi i}{MT}\right)\right)\right]$$

Frequency Domain Representation of Down Sampling (cont.)

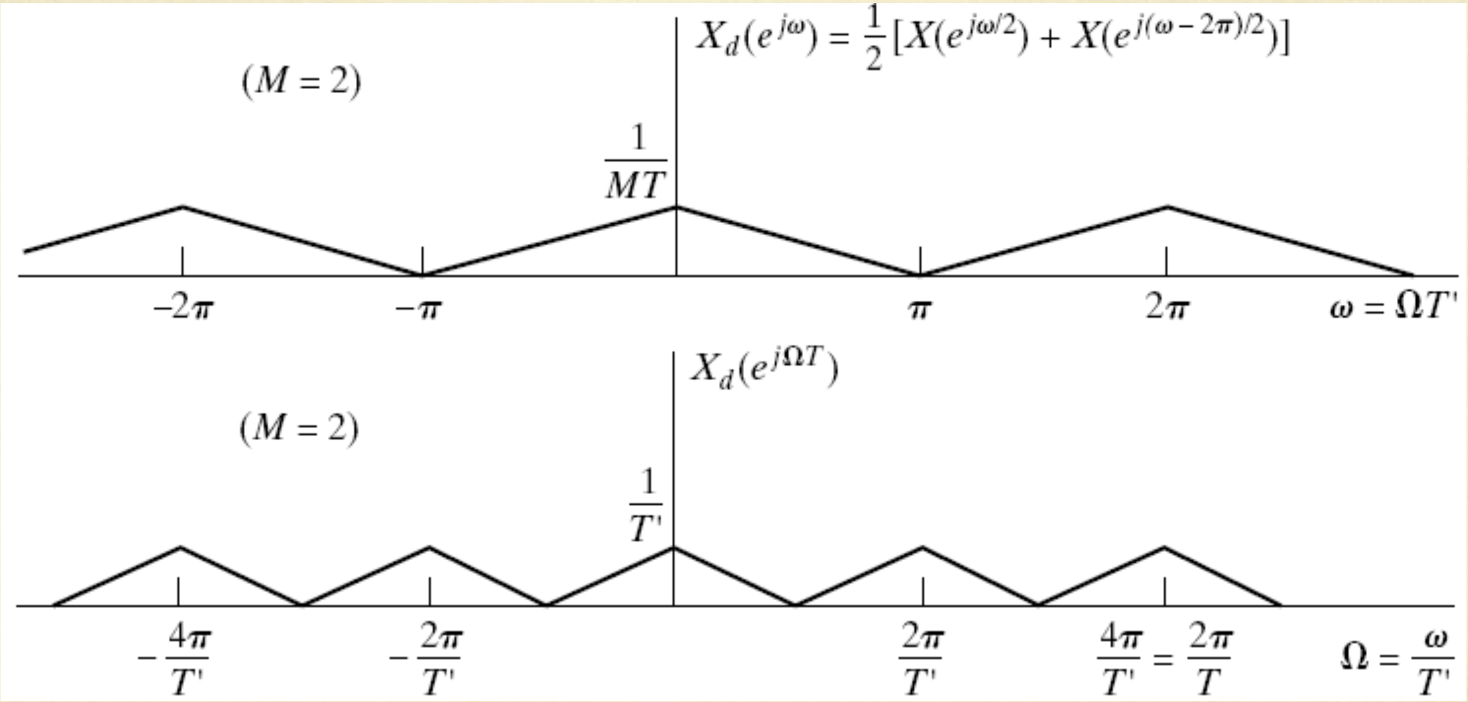
- And finally:

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

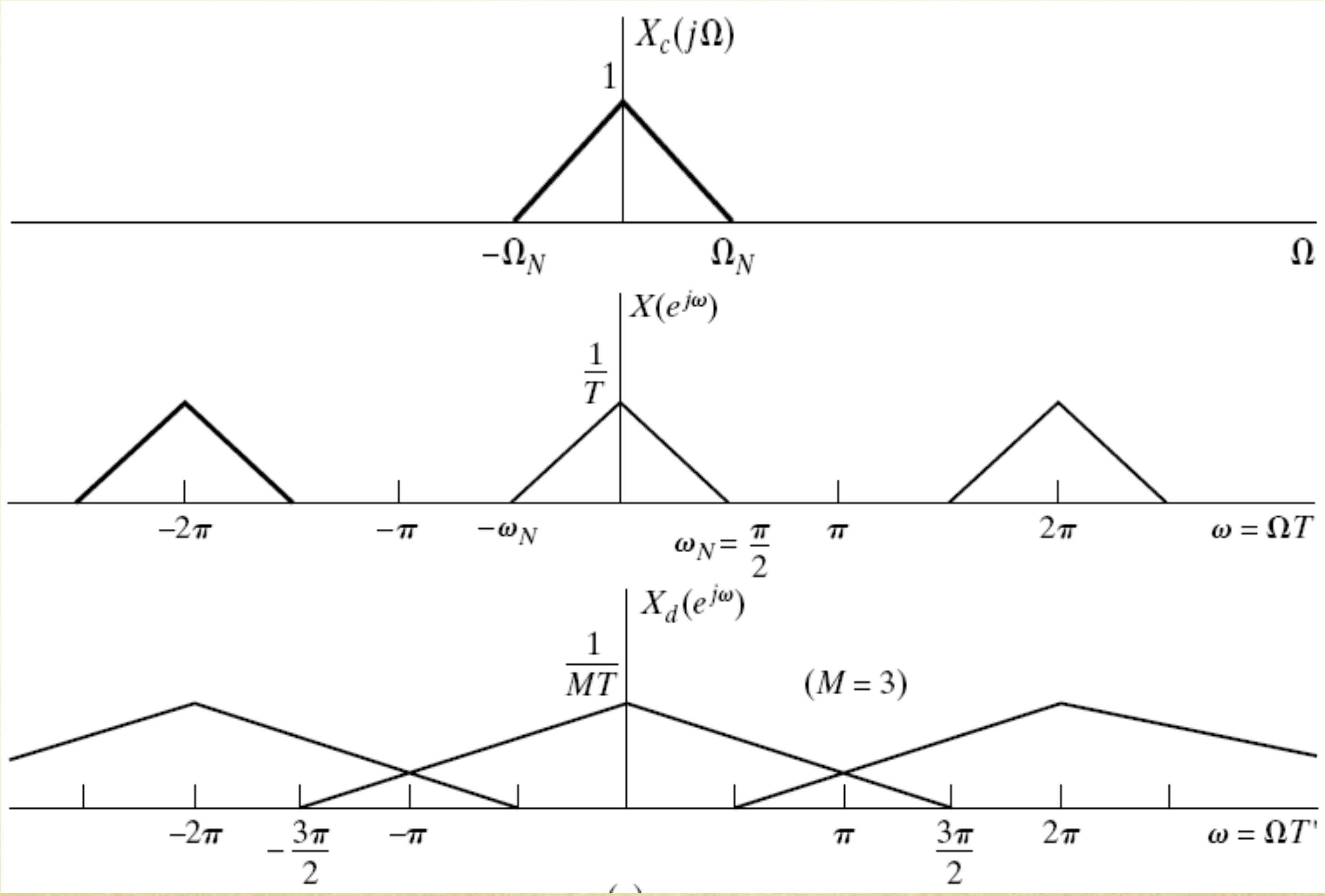
Frequency Domain Representation of Down Sampling: No Aliasing



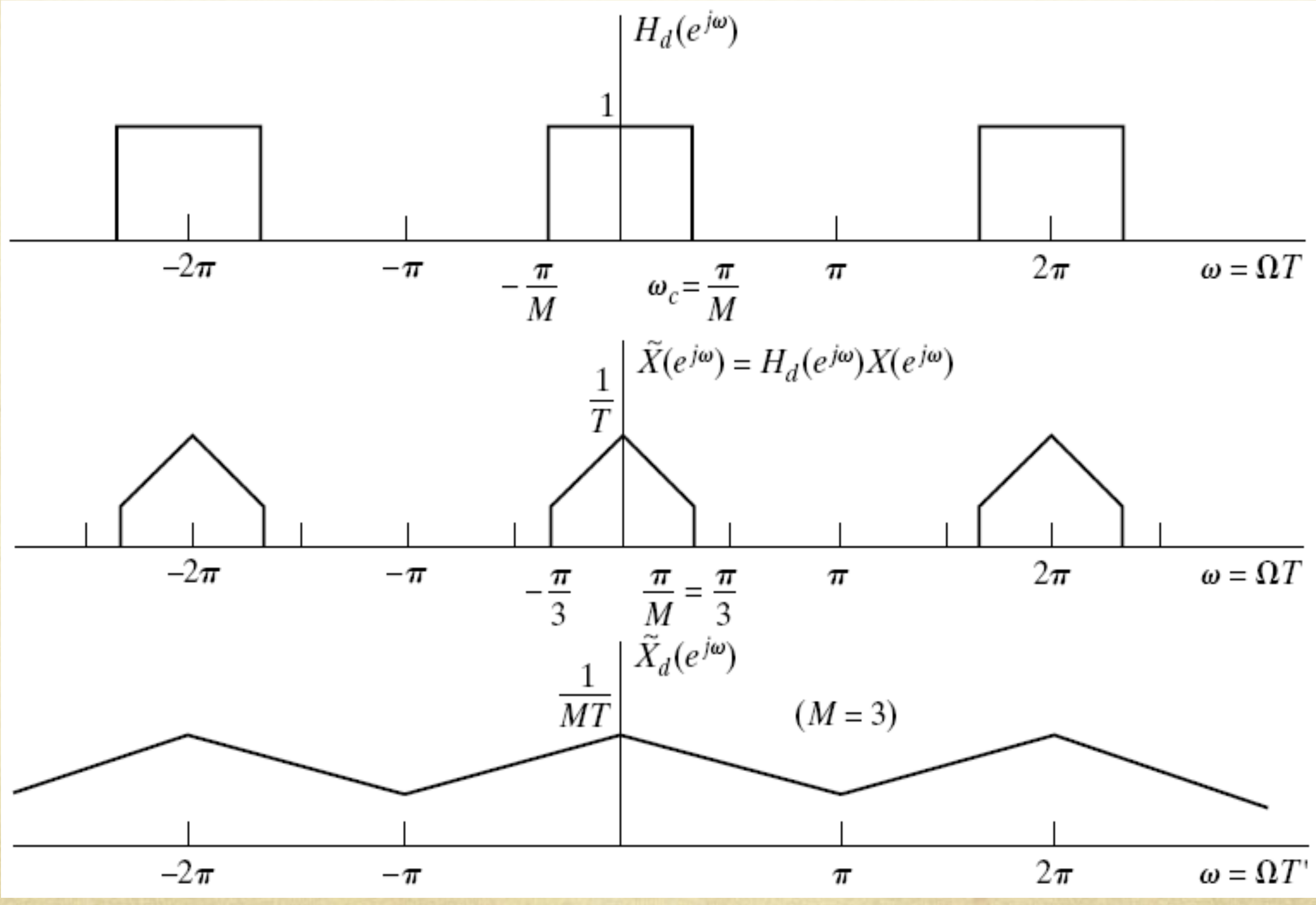
Frequency Domain Representation of Down Sampling: No Aliasing



Frequency Domain Representation of Down Sampling w/Prefilter



Frequency Domain Representation of Down Sampling w/Prefilter

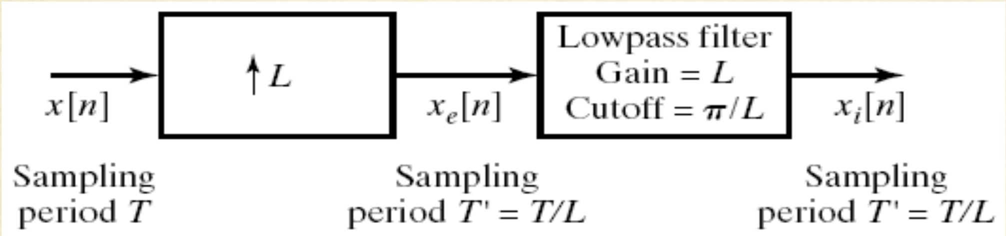


Increasing the Sampling Rate by an Integer Factor: Up Sampling

- We increase the sampling rate of a sequence interpolating it:

$$x_i[n] = x\left[\frac{n}{L}\right] = x_c\left(\frac{nT}{L}\right)$$

- This is accomplished with a sampling rate expander:



- We obtain $x_i[n]$ that is identical to what we would get by reconstructing the signal and resampling it with T'=T/L.

- Up sampling consists of two steps:
  - Expanding:

$$x_e[n] = \begin{cases} x\left[\frac{n}{L}\right] & n = 0, \mp L, \mp 2L, ... \\ 0 & else \end{cases} = \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]$$

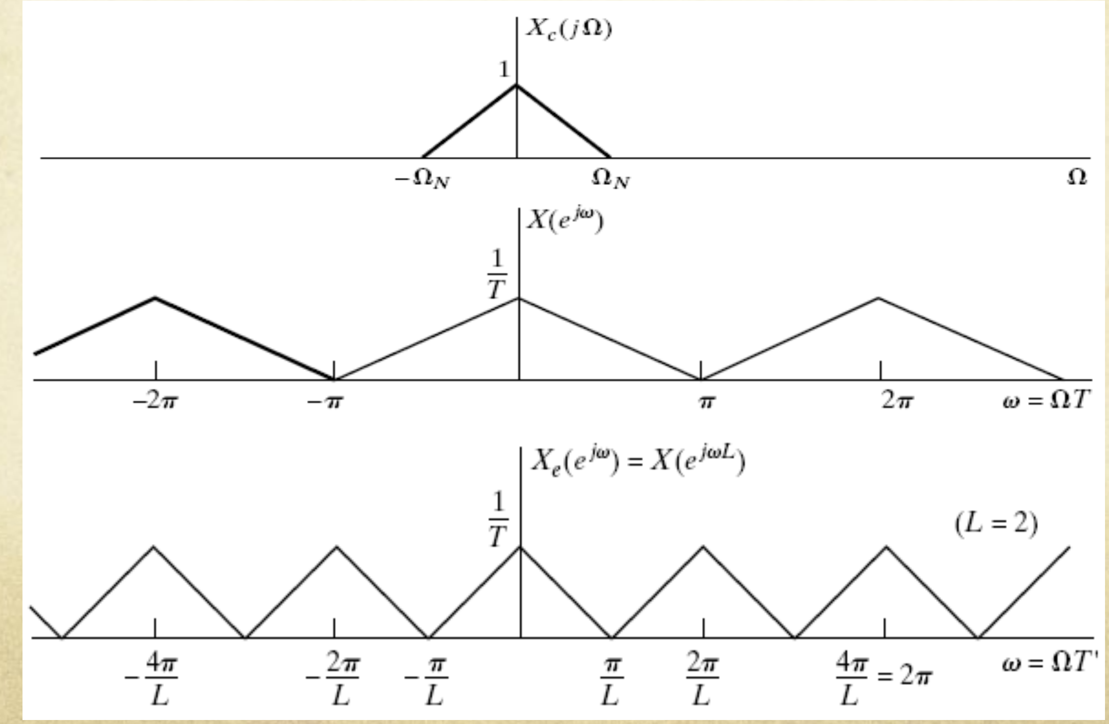
  - Interpolating.

Frequency Domain Representation of Expander

- The DTFT of $x_e[n]$ can be written as:

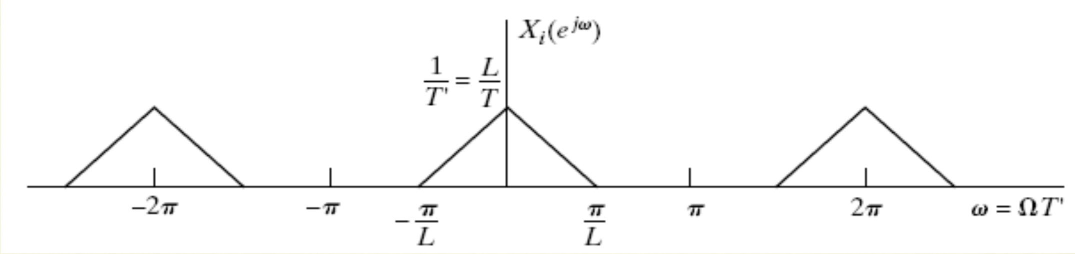
$$X_e\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]\right)e^{-j\omega n} = \sum_{k=-\infty}^{\infty} x[k]e^{-j\omega LK} = X\left(e^{j\omega L}\right)$$

- The output of the expander is frequency scaled:

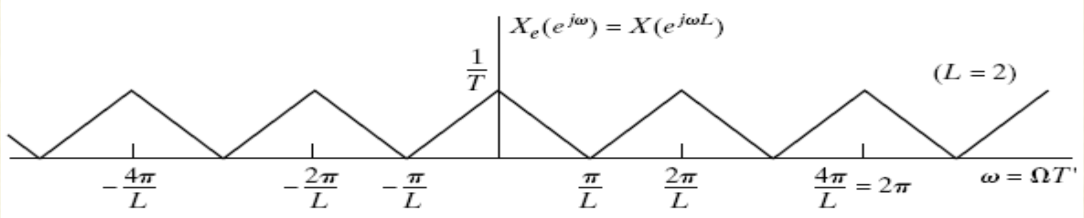


Frequency Domain Representation of Interpolator

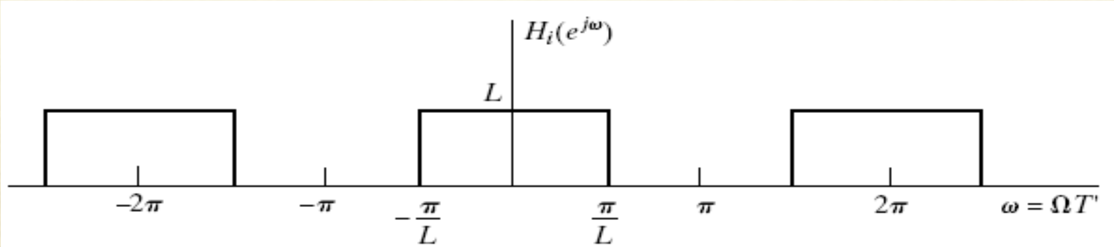
- The DTFT of the desired interpolated signals is:



- The extrapolator output is given as:



- To get interpolated signal we apply the following LPF:



Interpolator in Time Domain

- $X_i[n]$ in a low-pass filtered version of x[n].
- The low-pass filter impulse response is:

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

- Hence the interpolated signal is written as:

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

- Note that:

$$h_i\left[0\right]=1$$

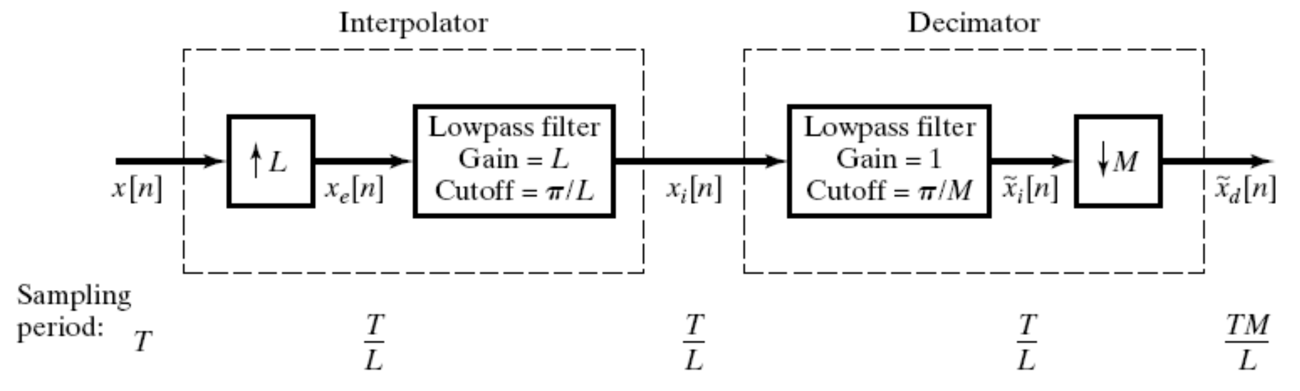
$$h_i\left[n\right]=0 \qquad n=\mp L,\mp 2L,\ldots$$

- Therefore the filter output can be written as:

$$x_i\left[n\right]=x\left[n/L\right]=x_c\left(nT/L\right)=x_c\left(nT'\right) \qquad \text{for } n=0,\mp L,\mp 2L,\ldots$$

Changing the Sampling Rate by Non-Integer Factor

- Combine decimation and interpolation for non-integer factors



- The two low-pass filters can be combined into a single one