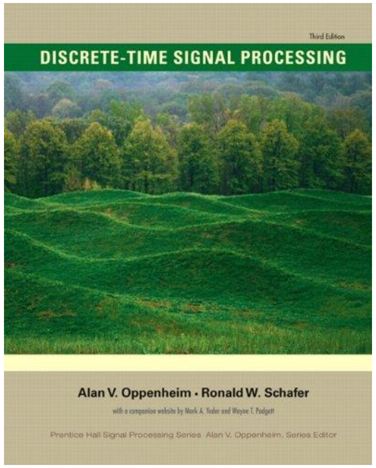


Lecture 07:

Changing the Sampling Rate

Changing the Sampling Rate

- **Sample-rate conversion** is the process of changing the sampling rate of a discrete signal to obtain a new discrete representation of the underlying continuous signal.

$$x[n] = x_c(nT) \longrightarrow x_d[n] = x_c(nT_d)$$

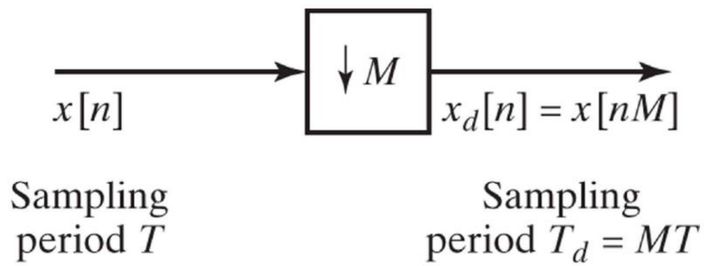
- One approach
  - Reconstruct $x_c(t)$ from $x[n]$
  - Resampling $x_c(t)$ to obtain $x_d[n]$

    Not desirable due to non-ideal analog reconstruction filter, and requiring additional D/A, A/D converters
- Other approach
  - involve only discrete-time operations

Outlines

1. Downsampling
2. Upsampling
3. General Rate Change
4. Some Practices

1. Downsampling by an Integer Factor



- If $\frac{2\pi}{T_d} = \frac{2\pi}{MT} > 2\Omega_N$

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

- What is DTFT of $x_d[n]$ in terms of $x[n]$?

DTFT of $x_d[n]$

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

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

- Then, DTFT of $x_d[n] = x_c(nT_d)$

- Since $T_d = MT$

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

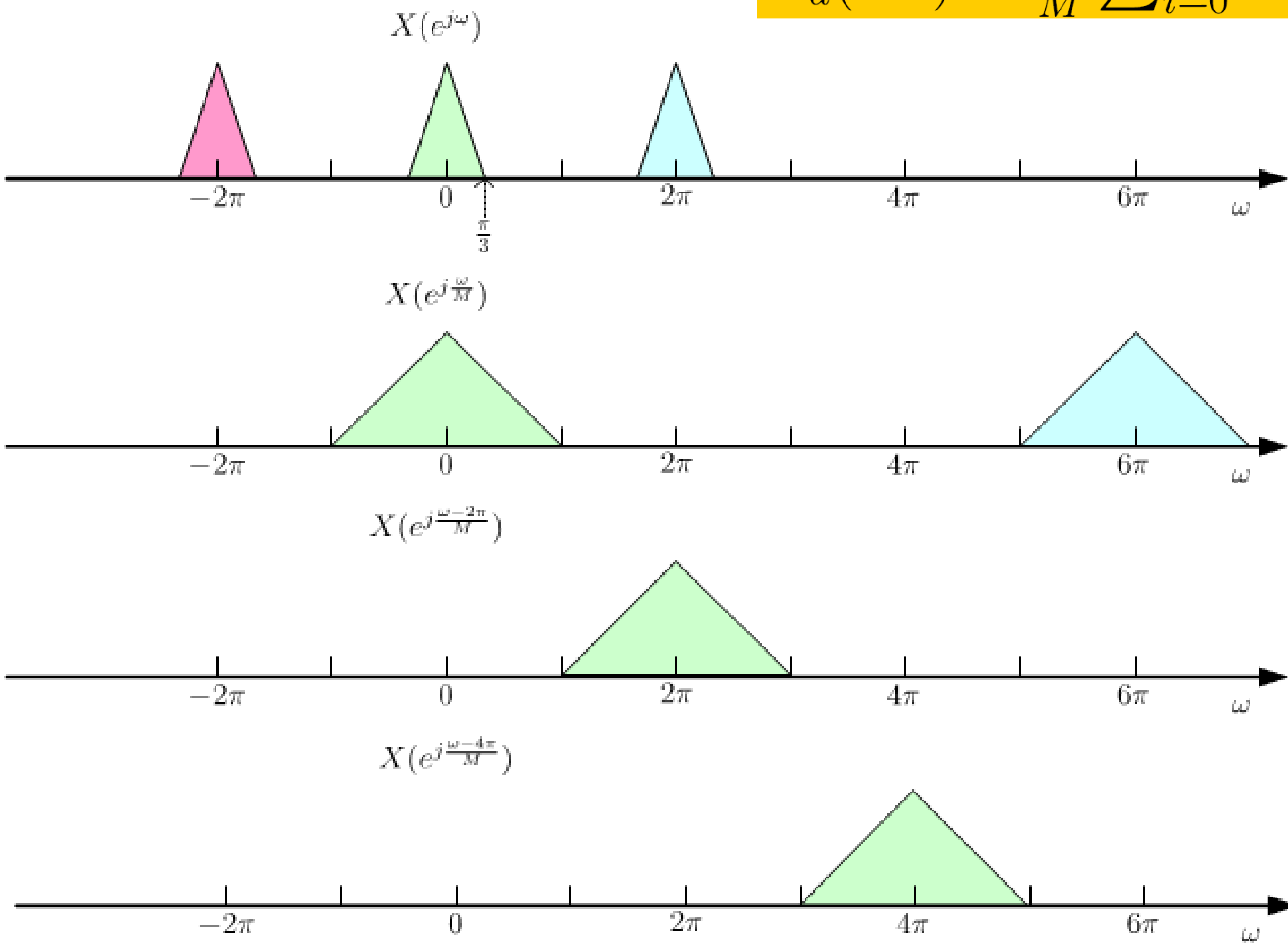
DTFT of $x_d[n]$

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

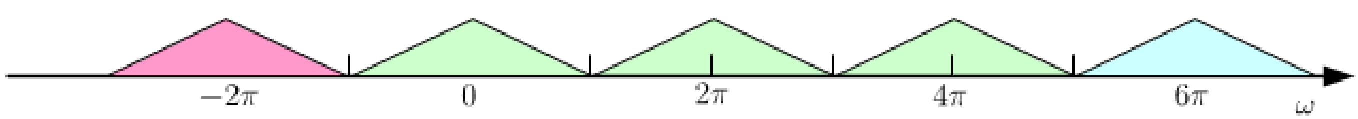
$\rightarrow r = i + kM$, where $-\infty \leq k \leq \infty$, $0 \leq i \leq M-1$

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

$M = 3$



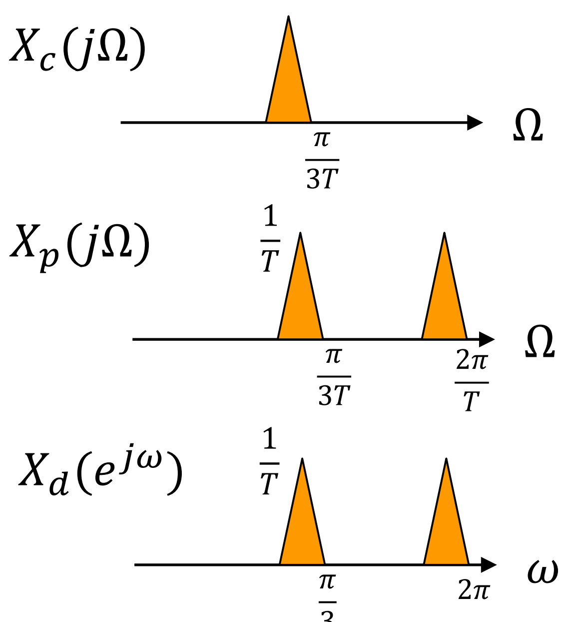
$$X_d(e^{j\omega}) = \frac{1}{3}\left(X(e^{j\frac{\omega}{M}}) + X(e^{j\frac{\omega-2\pi}{M}}) + X(e^{j\frac{\omega-4\pi}{M}})\right)$$

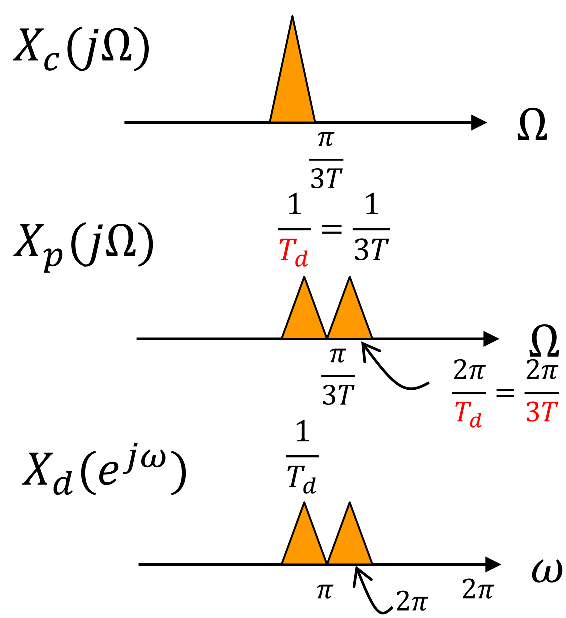


- DTFT of $x_d[n]$ in terms of DTFT of $x[n]$

→ infinite set of copies of $X_c(j\Omega)$, freq. scaled through $\omega = \Omega T_d$, shifted by integer multiple of $2\pi/T_d$

Recall



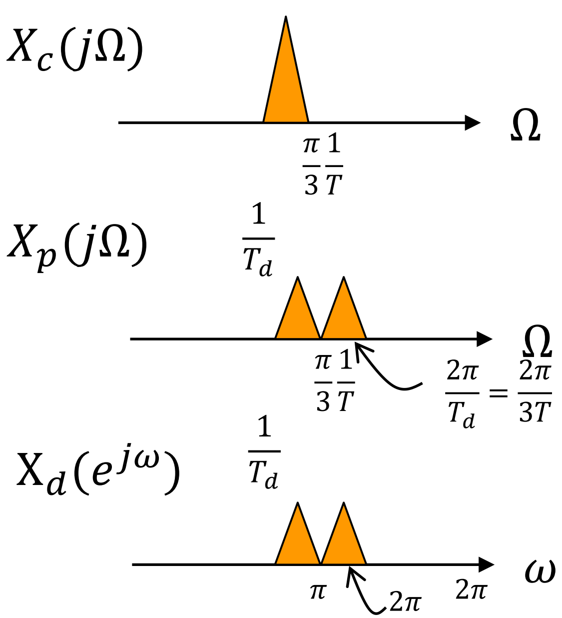


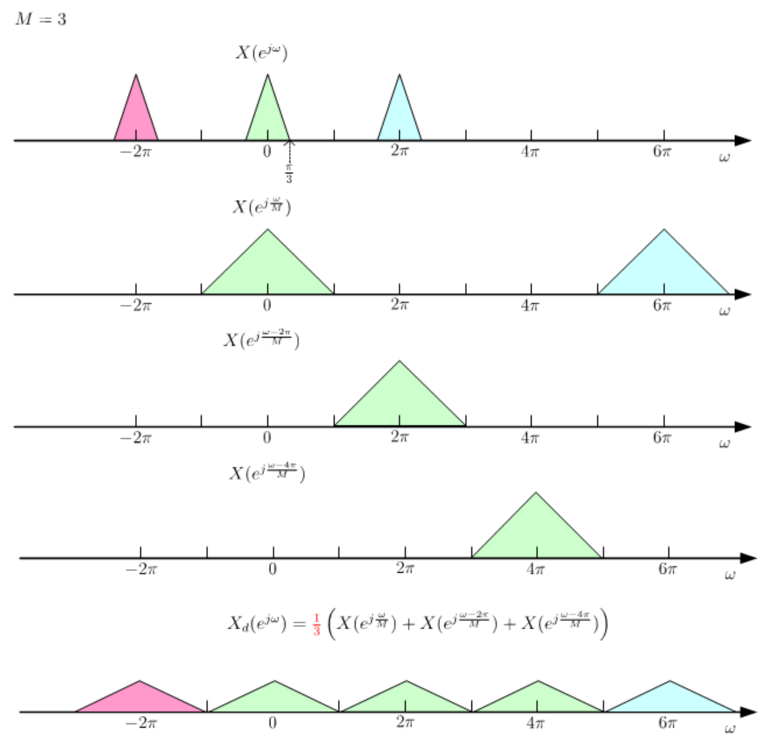
- DTFT of $x_d[n]$ in terms of DTFT of $x[n]$

→ infinite set of copies of $X_c(j\Omega)$, freq. scaled through $\omega = \Omega T_d$, shifted by integer multiple of $2\pi/T_d$

or

→ M copies of $X(e^{j\omega})$, freq. scaled by M, shifted by integer of $2\pi/M$

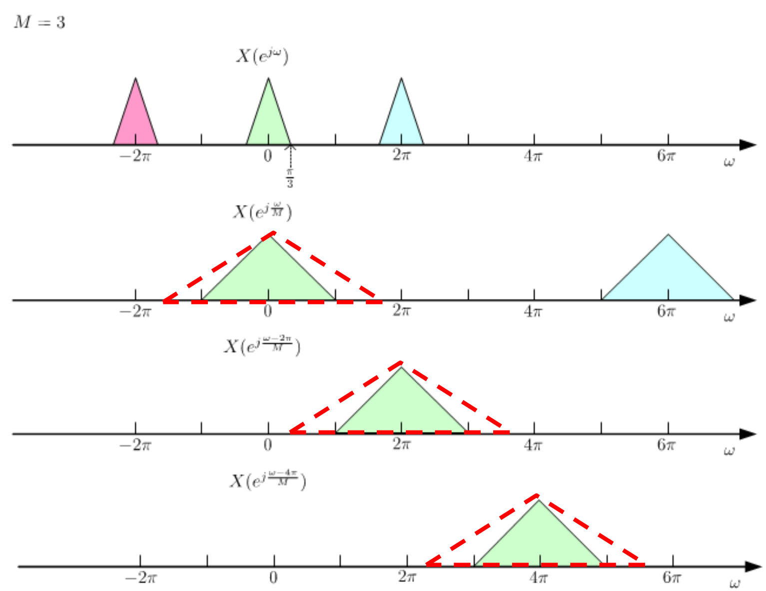




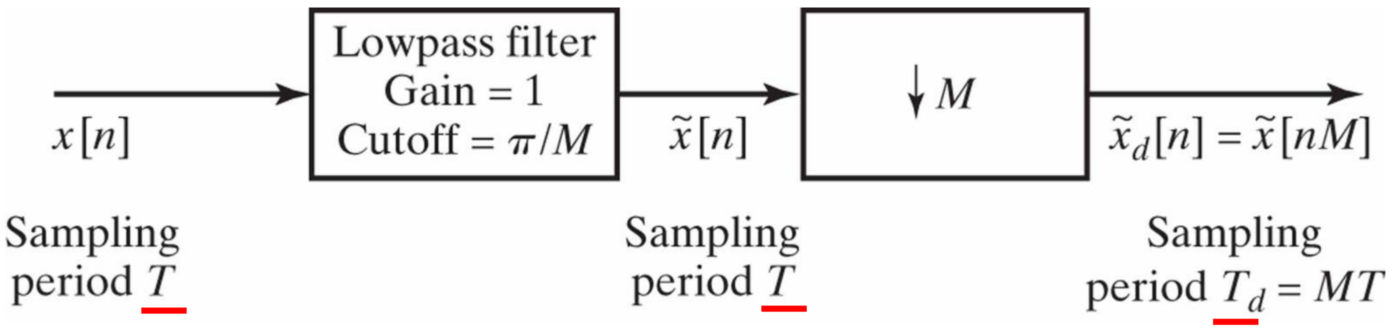
- Here, no aliasing since original sampled sequence is downsampled by $M = 3$
- If $M > 3 \longrightarrow$ aliasing occurs
- To avoid aliasing in downsampling

$$\omega_N M \le \pi$$

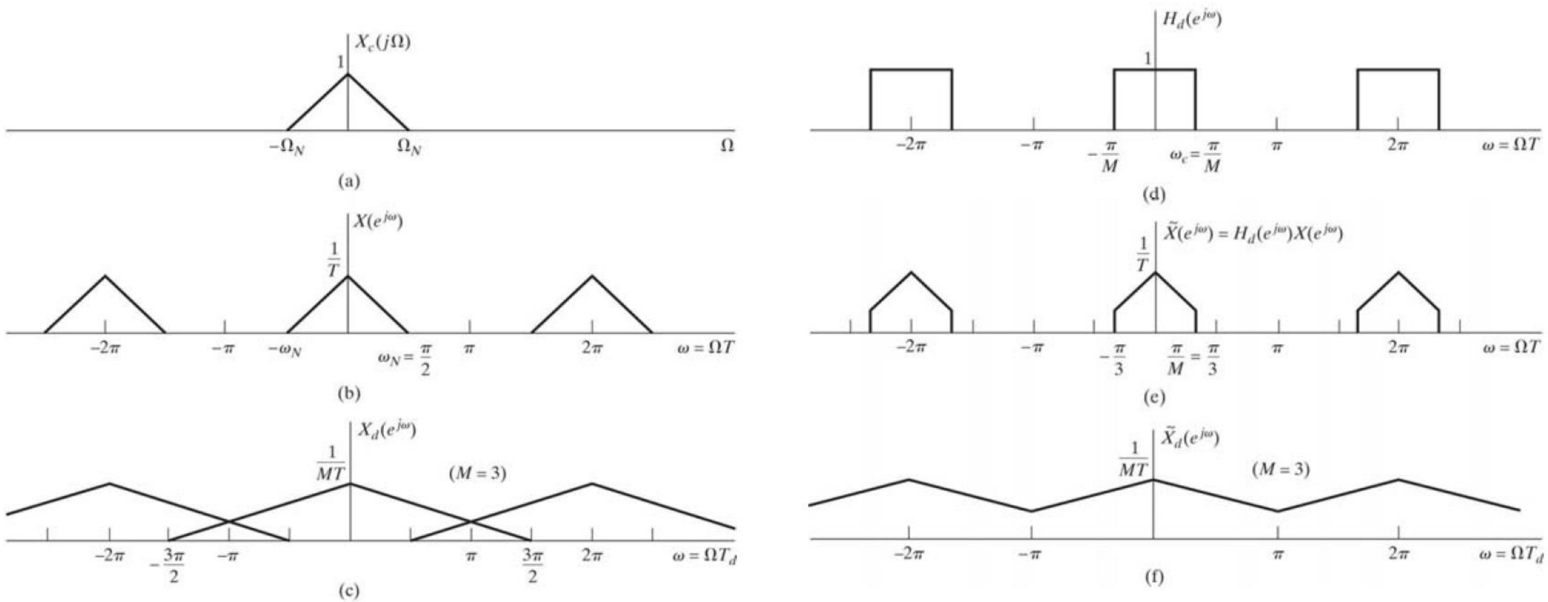
$$\longleftrightarrow \quad \omega_N \le \frac{\pi}{M}$$



In Case of Possible Aliasing, LPF



It's decimator! (downsampling by LPF followed by compression)

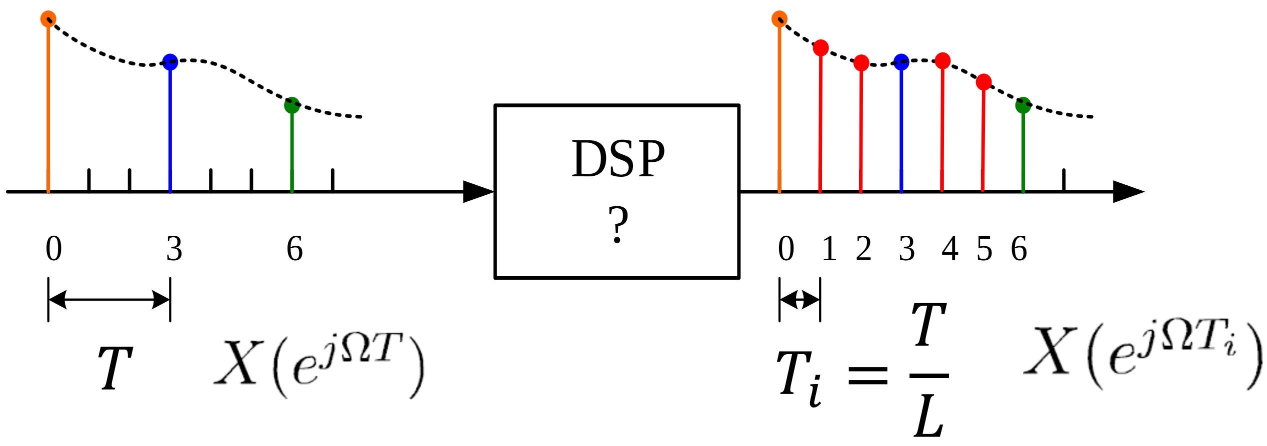


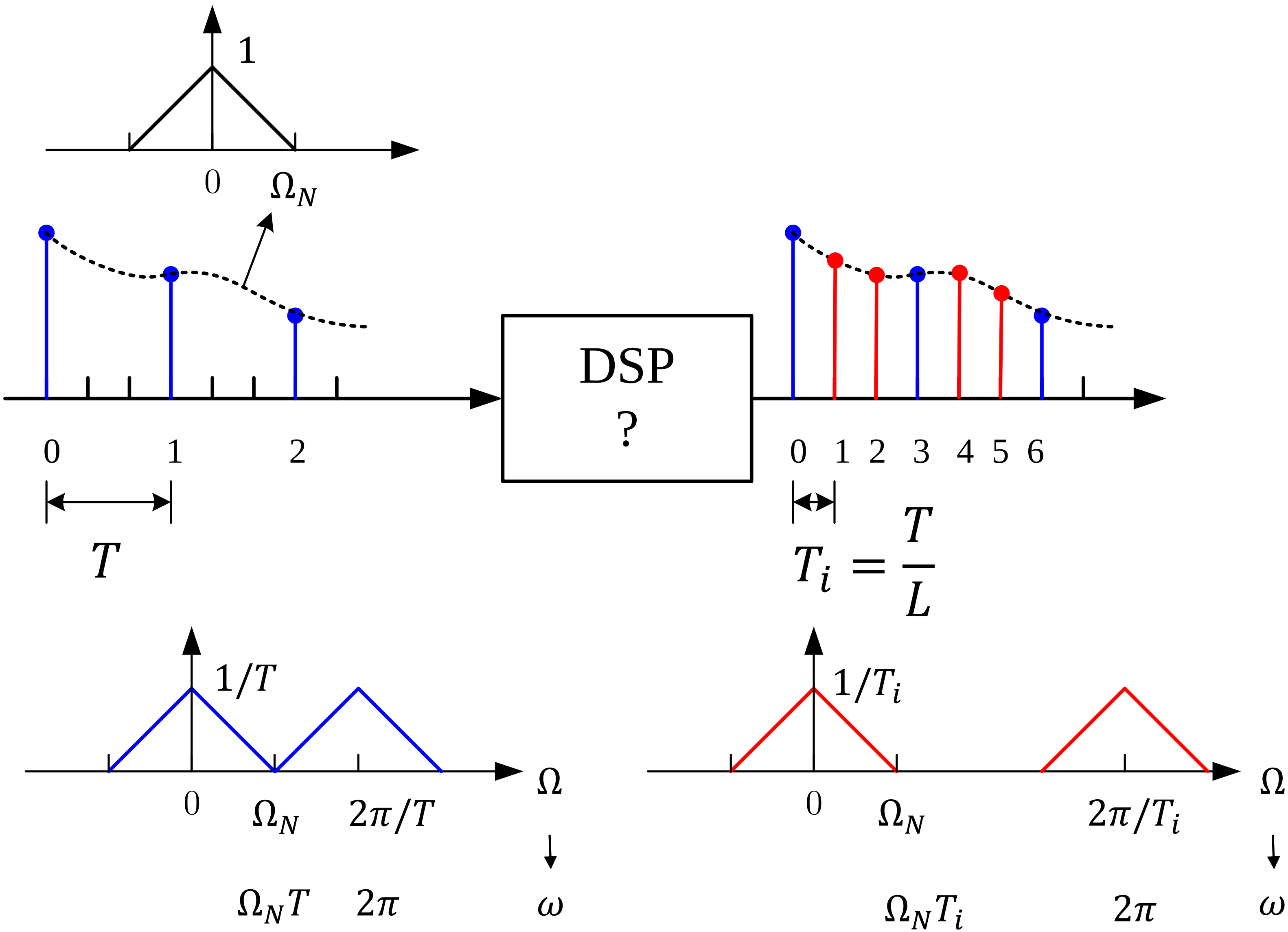
2. Increase the Sampling Rate using DSP

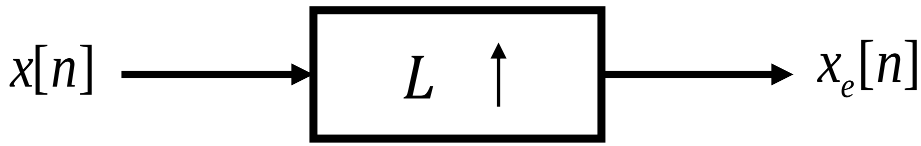
$$x[n] = x_c(nT)$$

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

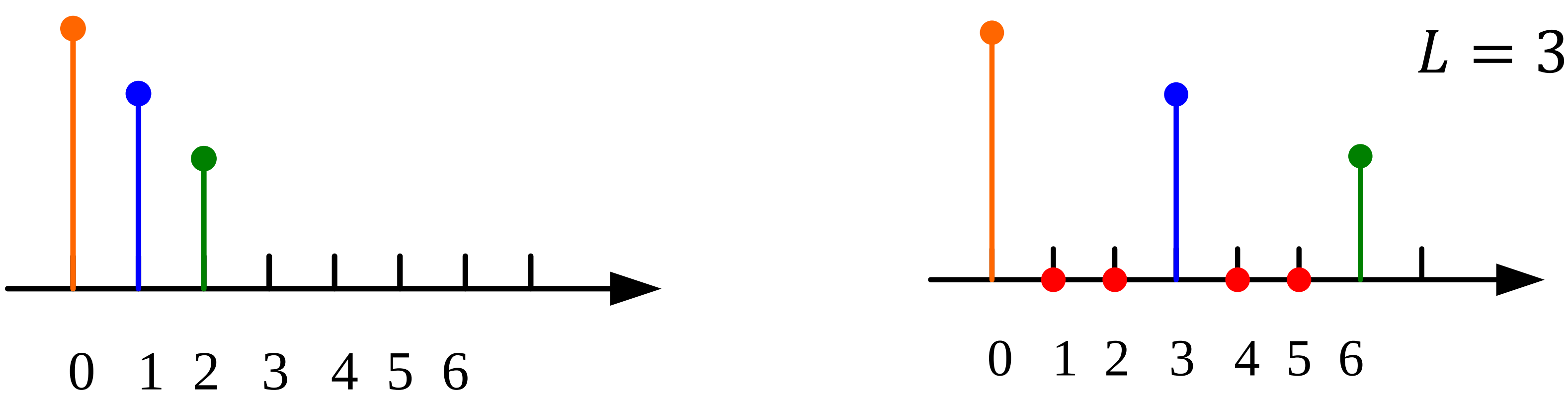
It's (sampling rate) expander !

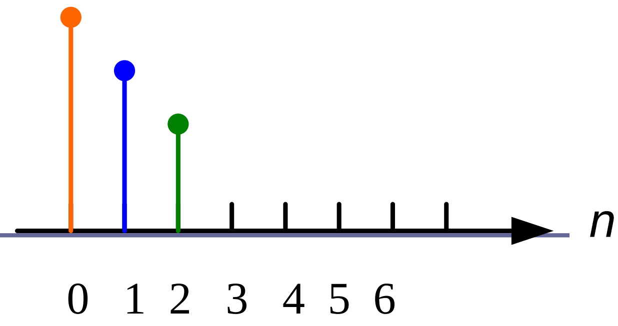



1
0
Ω_N
0
1
2
T
DSP
?
0 1 2 3 4 5 6
$T_i = \frac{T}{L}$
$1/T$
0
Ω_N
$2\pi/T$
Ω
ω
$\Omega_N T$
2π
$1/T_i$
0
Ω_N
$2\pi/T_i$
Ω
ω
$\Omega_N T_i$
2π



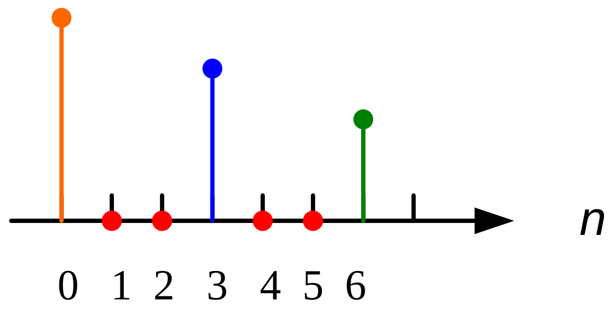
$$x_e[n] = \begin{cases} x[n/L], & n/L : \text{an integer} \\ 0, & \text{otherwise} \end{cases}$$





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

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

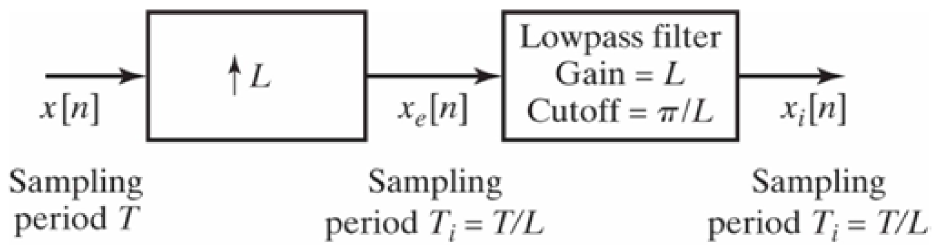


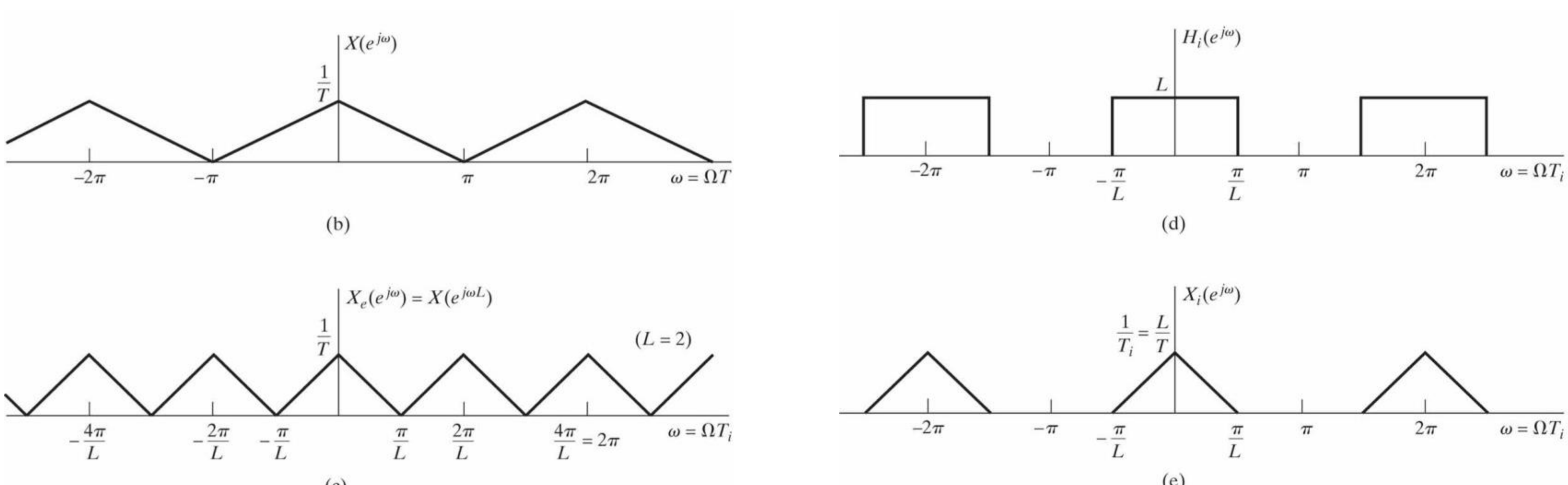
In frequency domain, DTFT of $x_e[n]$ is

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

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

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

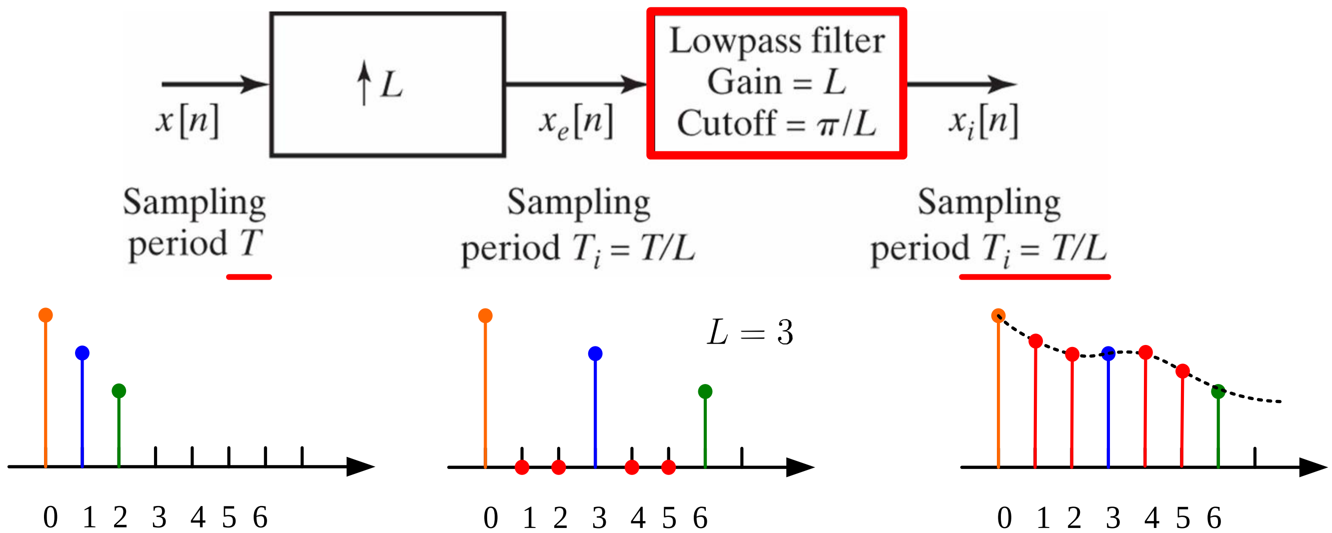




freq. scaled version of DTFT of the input

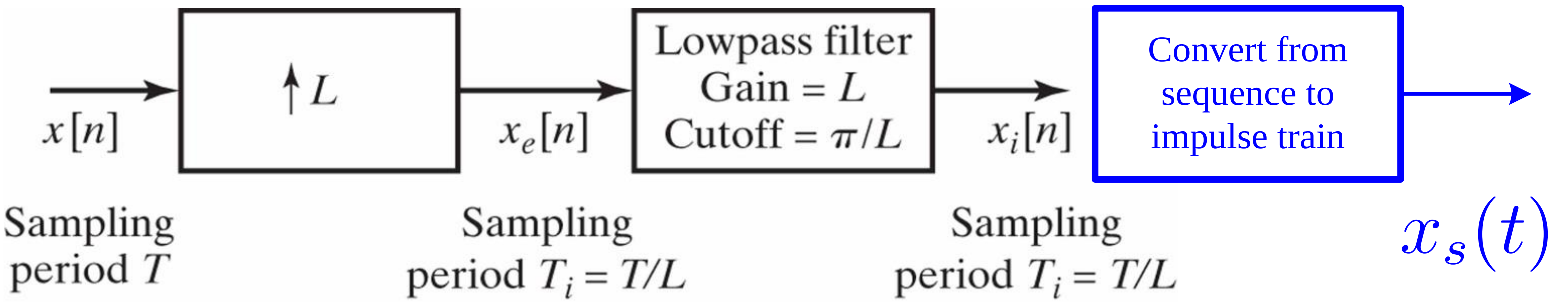
Low-pass Filter

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



$$H(e^{j\omega}) = \begin{cases} L, & |\omega| \le \pi/L, \\ 0, & \text{otherwise} \end{cases}$$

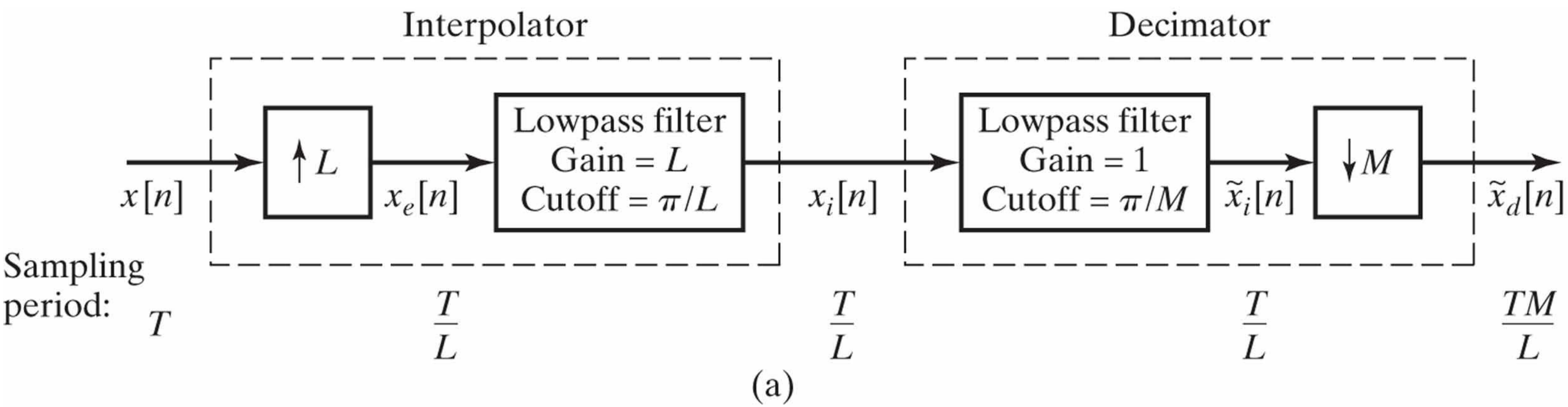
$$h[n] = \frac{\sin\left(\frac{\pi}{L}n\right)}{\frac{\pi}{L}n}, \quad -\infty < n < \infty$$

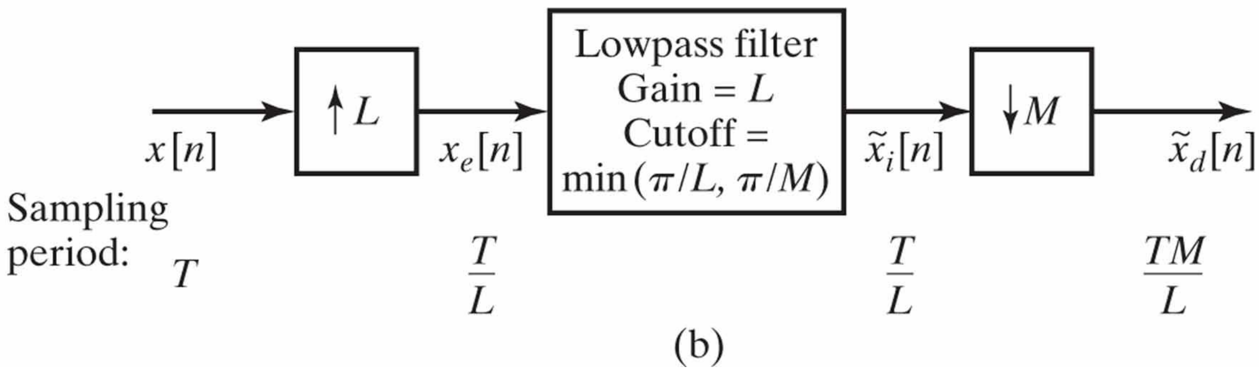


$\omega \to \omega L,$ $\quad\quad$ $\omega \leftrightarrow \Omega T_i$

Down sampling	Up sampling
1. sampling rate reduction	1. sampling rate increase
2. increase sample period	2. decrease sample period
3. decimation	3. interpolation

3. General Rate Change

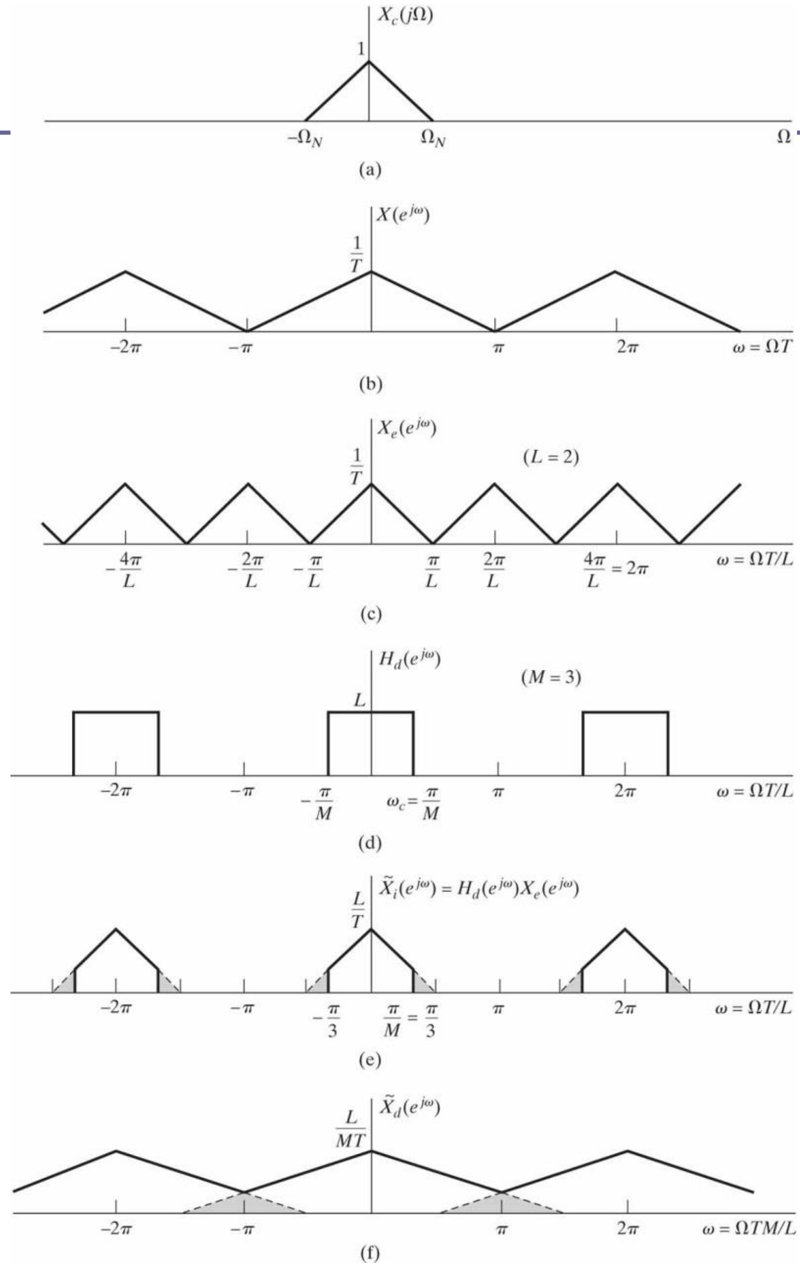




(a) System for changing the sampling rate by a non-integer factor

(b) Simplified system in which the decimation and interpolation filters are combined

Illustration of changing the sampling rate by a non-integer factor

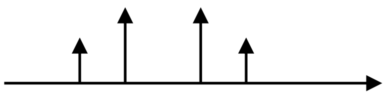


$$T \rightarrow 1.5T = \frac{3}{2}T$$

SOME PRACTICES

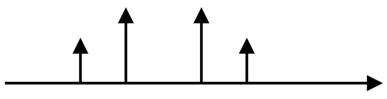
test_aliasing.m

$2\pi \cdot 30/1000$

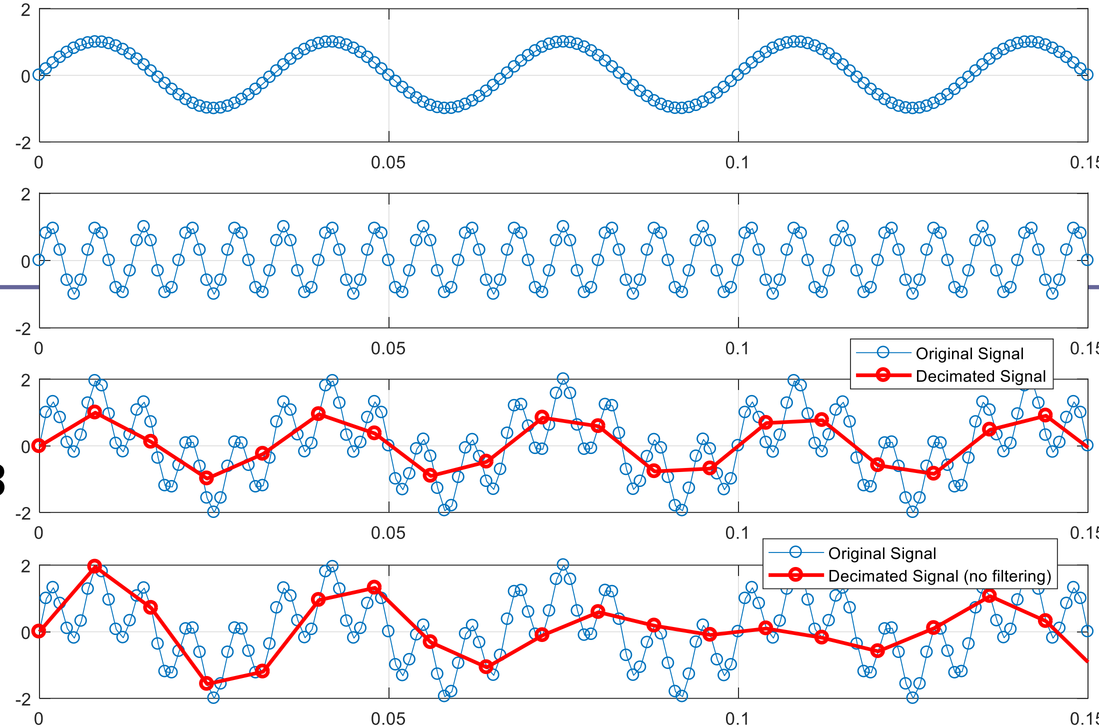


$2\pi \cdot 150/1000$

$2\pi \cdot 30/1000 \cdot 8$



$2\pi \cdot 150/1000 \cdot 8$



30 Hz

150 Hz

Original Sample Frequency = 1000Hz

- Decimate/downsample to 1000/8=125Hz
- Pay attention to the aliasing of high-frequency components when resampling!! (High-frequency components are aliased into the low frequency and mixed with other low-frequency components)

Difference：

- deci_signal = decimate(signal, 8);
- down_signal = downsample(signal, 8);

- decimate will pass the low-pass filter before downsampling

[MATLAB FUNCTION] RESAMPLE

- help resample
- Y = resample(X, P, Q)

  resamples the sequence in vector X at P/Q times the original sample rate using a polyphase implementation. Y is P/Q times the length of X (or the ceiling of this if P/Q is not an integer). P and Q must be positive integers.

  Resample applies an anti-aliasing (lowpass) FIR filter to X during the resampling process, and compensates for the filter's delay.

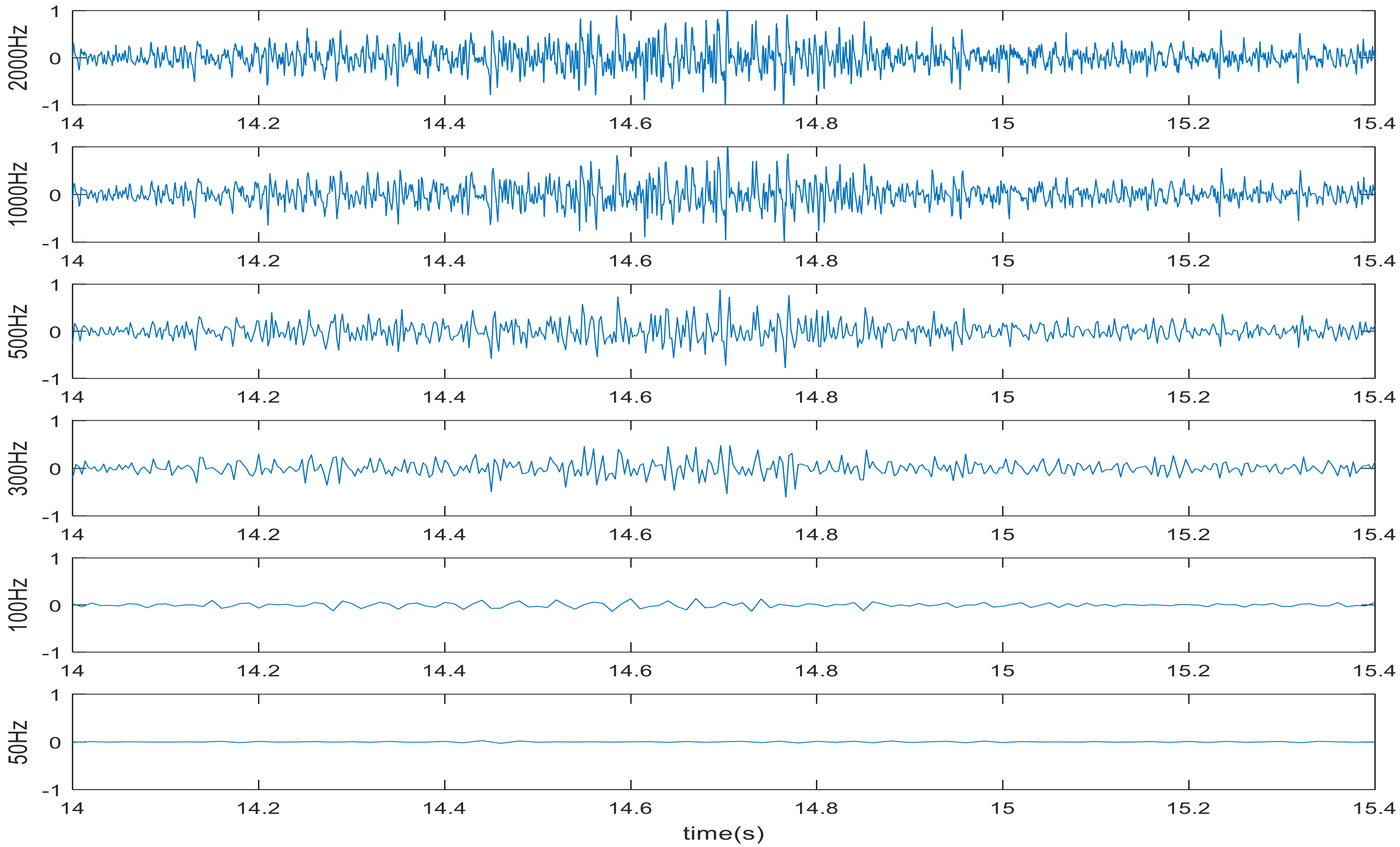
[MATLAB FUNCTION] RAT

- help rat
- [P,Q] = rat(X, tol)

  returns two integer matrices so that P./Q is close to X in the sense that abs(P./Q - X) <= tol.

  The rational approximations are generated by truncating continued fraction expansions.

  tol = 1.e-6*norm(X(:),1) is the default.



- org_SR=2000; % in Hz
- new_SR3=300; % in Hz

```
[p,q]=rat(new_SR/org_SR);
new_signal3=resample(org_signal,p,q);
new_taxis3=[1:length(new_signal3)]'/new_SR3;
```

Don't forget to redefine the timeline after re-sampling!

Commonly used functions

- interp % Interpolate the signal to increase the sampling frequency
- decimate % Downsampling after low-pass filtering
- resample % Perform P times interp to increase the sampling frequency, and then perform Q times decimate to decrease the sampling frequency
- rat % Find the fraction form that is closest to the input value (P/Q)

Example

https://www.mathworks.com/help/signal/ug/changing-signal-sample-rate.html