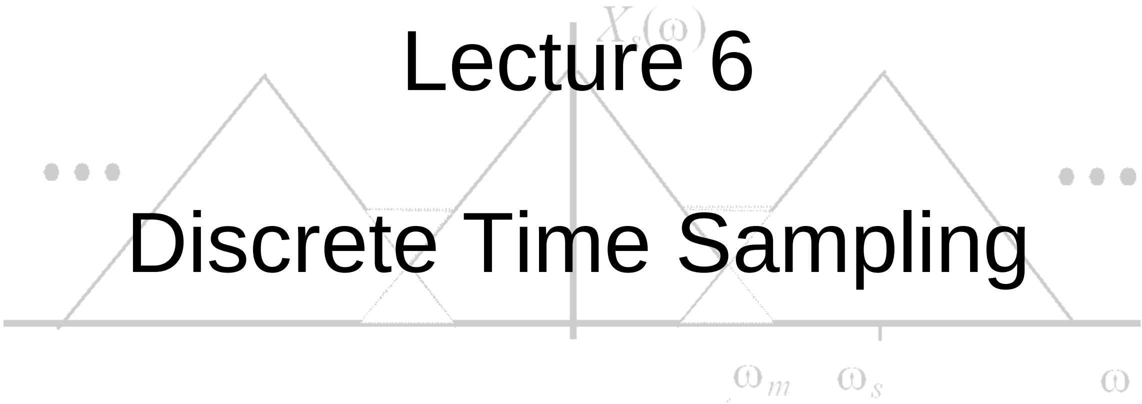


Lecture 6

Discrete Time Sampling

Changing the sampling rate : Downsampling & Upsampling

Dr. Dimitris Agrafiotis, Dept. Of Electrical & Electronic Engineering

Sampling Rate Changes

Downsampling & Upsampling

$x_u[n] = x[n/L]$

$\mathbf{x[n] = x_c(nT_s)}$

$x_d[n] = x[nM]$

Downsampling
Sampling rate reduction through decimation of existing samples (pixels): one pixel for every M pixels of the original

Upsampling
Sampling rate increase through insertion of L-1 zero pixels between existing pixels and interpolation of the in-between values

Discrete Time Sampling

Sampling Rate Changes

Downsampling & Upsampling

$x_u[n] = x[n/L]$

$x[n] = x_c(nT_s)$

$x_d[n] = x[nM]$

What happens to the spectrum when we reduce the sampling rate ?

What happens to the spectrum when we increase the sampling rate ?

Downsampling

Sampling rate reduction through decimation of existing samples (pixels): one pixel for every M pixels of the original

What affects the outcome of these operations and how can we get good results

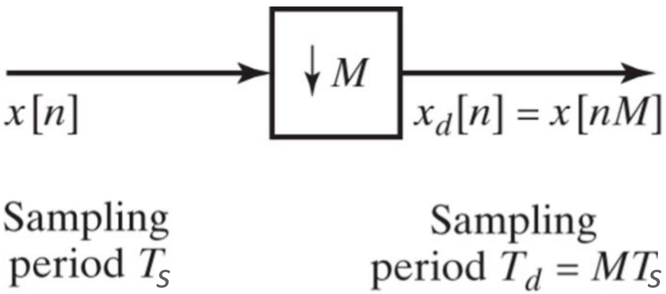
Upsampling

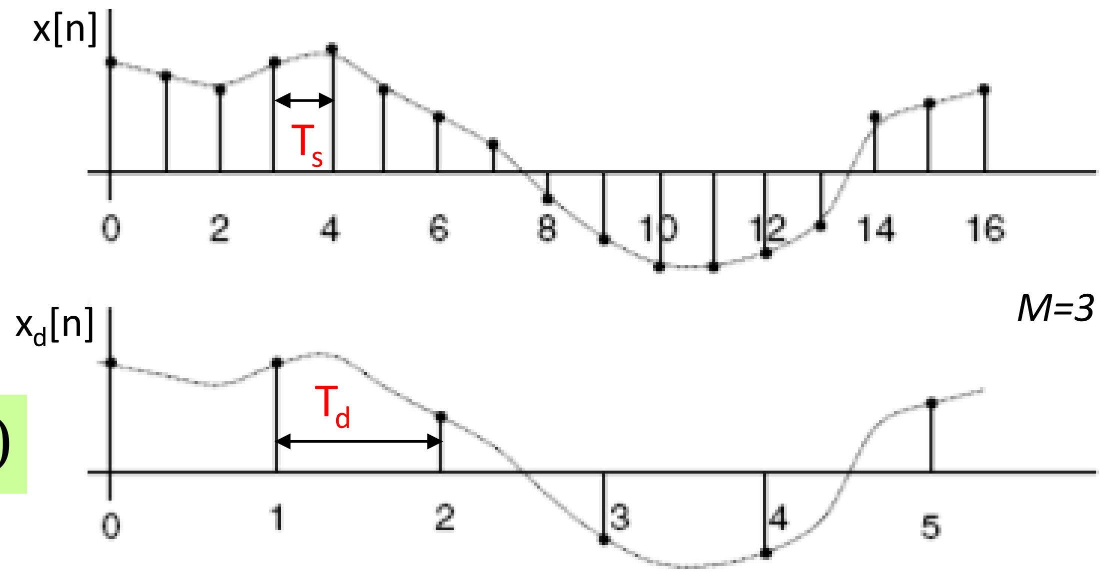
Sampling rate increase through insertion of L-1 zero pixels between existing pixels and interpolation of the in-between values

Discrete Time Sampling

Downsampling

Sampling rate reduction by an integer factor





$$x_d[n] = x[nM] = x_c(\underbrace{nMT_s})$$

Can we recover the original continuous time signal after we reduce the sampling rate by M ?

$$f_{sd} = \frac{1}{T_d} = \frac{1}{M\ T_s} = \frac{f_s}{M}$$

To avoid aliasing $f_{sd} \geq 2 f_N$

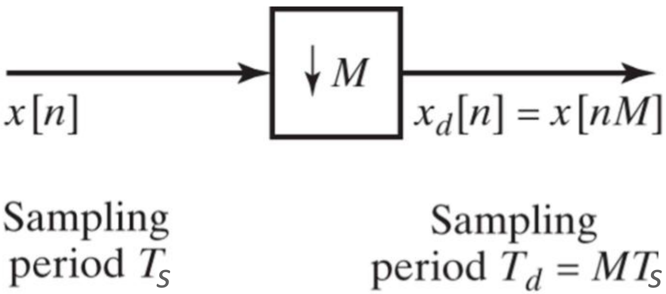
$$\Leftrightarrow \frac{f_s}{M} \geq 2 f_N \Leftrightarrow f_s \geq M 2 f_N \Leftrightarrow \omega_s \geq M 2 \omega_N$$

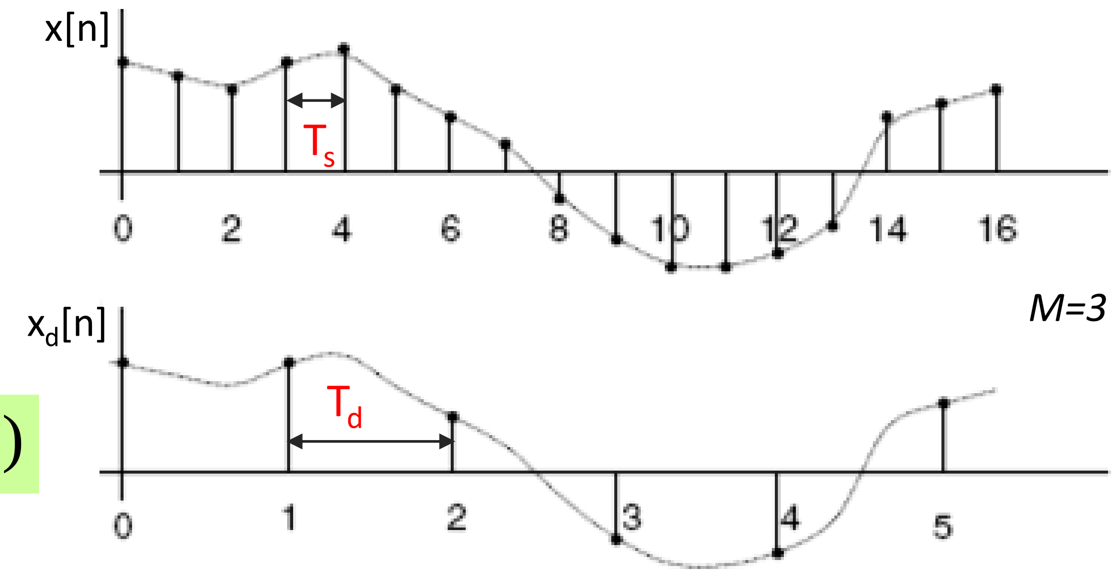
The original sampling rate applied to the CT signal must be at least M times the Nyquist rate

Discrete Time Sampling

Downsampling

Sampling rate reduction by an integer factor





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

How can we avoid aliasing when downsampling a discrete time signal by a factor of M?

$$\omega_s \geq M\,2\omega_N \Leftrightarrow \omega_N \leq \frac{\omega_s}{2M} \Leftrightarrow \omega_N T_s \leq \frac{\omega_s T_s}{2M} = \frac{2\pi}{T_s}\frac{T_s}{2M} \Leftrightarrow \Omega_N \leq \frac{\pi}{M} \Leftrightarrow F_N \leq \frac{1}{2M}$$

To avoid aliasing: $\Omega_N \leq \frac{\pi}{M} \Leftrightarrow F_N \leq \frac{1}{2M}$

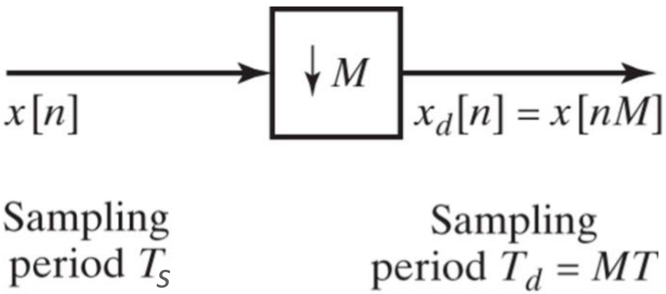
In other words we need : $X(\Omega) = 0,\ |\Omega| \geq \pi / M \Leftrightarrow X(F) = 0,\ |F| \geq 1/2M$

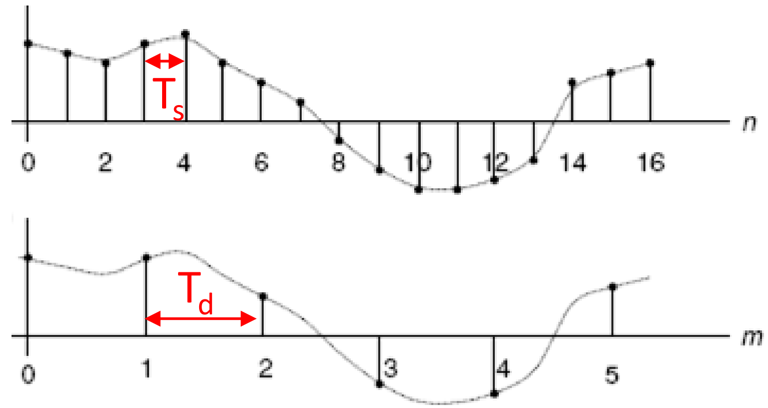
Filter the DT signal prior to downsampling with a digital low-pass filter with cut-off freq = π/M

Discrete Time Sampling

Downsampling – Frequency Domain

Effect of downsampling on the spectrum





DTFT of $x[n] = x_c[nT_s]$: $$X_s(\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X_c\left(\frac{\Omega}{T_s} - \frac{2\pi k}{T_s}\right) \quad (1)$$

lecture 5 expression of spectrum of sampled continuous time signals x_c

DTFT of $x_d[n] = x[nM] = x_c[nT_d]$: $$X_d(\Omega) = \frac{1}{T_d} \sum_{r=-\infty}^{\infty} X_c\left(\frac{\Omega}{T_d} - \frac{2\pi r}{T_d}\right)$$

$T_d = \mathrm{M}\, T_s$

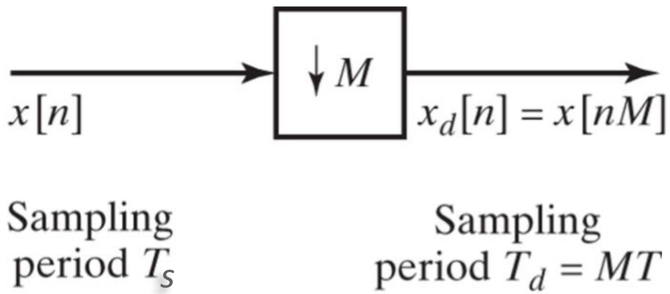
$\Rightarrow$

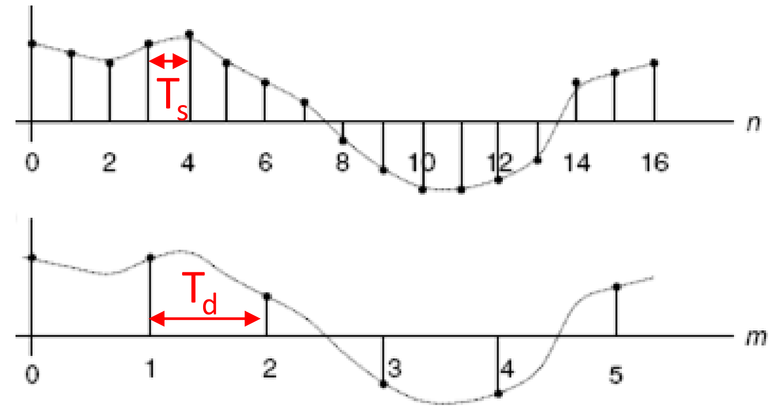
$$\Rightarrow X_d(\Omega) = \frac{1}{MT_s} \sum_{r=-\infty}^{\infty} X_c\left(\frac{\Omega}{MT_s} - \frac{2\pi r}{MT_s}\right)$$

Discrete Time Sampling

Downsampling – Frequency Domain

Effect of downsampling on the spectrum





$$X_d(\Omega)=\frac{1}{MT_s}\sum_{r=-\infty}^{\infty}X_c\left(\frac{\Omega}{MT_s}-\frac{2\pi r}{MT_s}\right)$$

Express r as $r=i+kM$, where $-\infty<k<\infty$, $0\le i\le M-1$

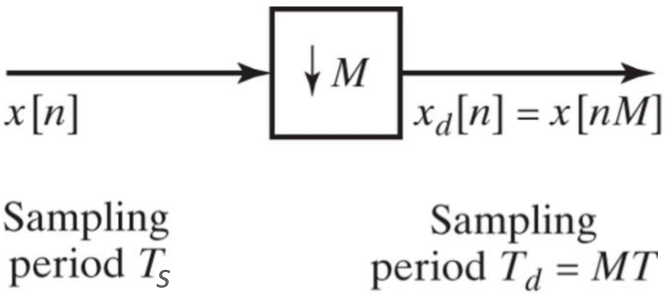
$$\Rightarrow X_d(\Omega)=\frac{1}{MT_s}\sum_{i=0}^{M-1}\sum_{k=-\infty}^{\infty}X_c\left(\frac{\Omega}{MT_s}-\frac{2\pi k}{T_s}-\frac{2\pi i}{MT_s}\right)$$

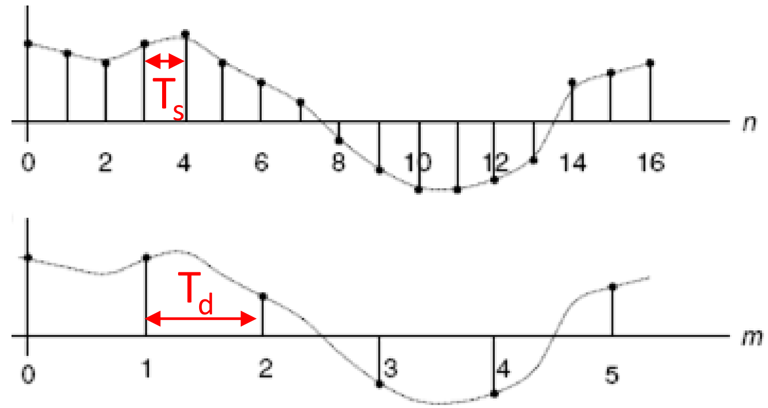
$$\Rightarrow X_d(\Omega)=\frac{1}{M}\sum_{i=0}^{M-1}\left\{\frac{1}{T_s}\sum_{k=-\infty}^{\infty}X_c\left(\frac{\Omega}{MT_s}-\frac{2\pi k}{T_s}-\frac{2\pi i}{MT_s}\right)\right\} \quad (2)$$

Discrete Time Sampling

Downsampling – Frequency Domain

Effect of downsampling on the spectrum





$$X_d(\Omega)=\frac{1}{M}\sum_{i=0}^{M-1}\left\{\frac{1}{T_s}\sum_{k=-\infty}^{\infty}X_c\left(\frac{\Omega}{MT_s}-\frac{2\pi k}{T_s}-\frac{2\pi i}{MT_s}\right)\right\} \quad (2)$$

$$X_s(\Omega)=\frac{1}{T_s}\sum_{k=-\infty}^{\infty}X_c\left(\frac{\Omega}{T_s}-\frac{2\pi k}{T_s}\right) \quad (1)$$

(1) (2)

$$\Rightarrow \quad X_s\left(\frac{\Omega-2\pi i}{M}\right)=\frac{1}{T_s}\sum_{k=-\infty}^{\infty}X_c\left(\frac{\Omega}{MT_s}-\frac{2\pi k}{T_s}-\frac{2\pi i}{MT_s}\right) \quad (3)$$

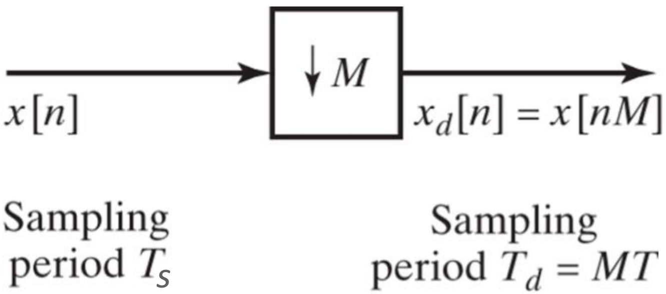
(2) (3)

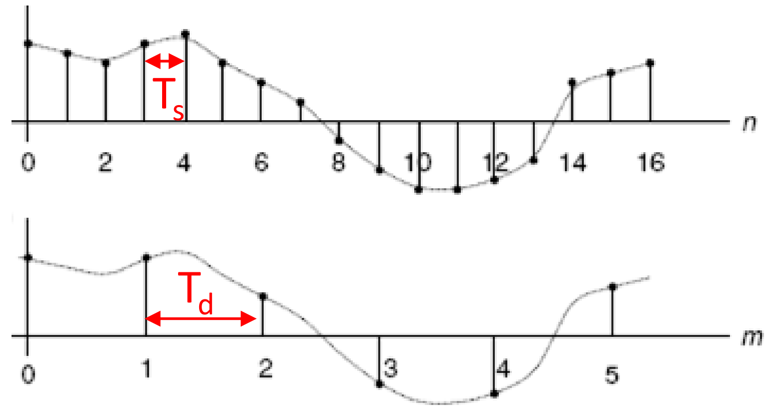
$$\Rightarrow \quad X_d(\Omega)=\frac{1}{M}\sum_{i=0}^{M-1}X_s\left(\frac{\Omega}{M}-\frac{2\pi i}{M}\right), \quad X_d(\Omega)=\frac{1}{M}\sum_{i=0}^{M-1}X\left(\frac{\Omega}{M}-\frac{2\pi i}{M}\right)$$

Discrete Time Sampling

Downsampling – Frequency Domain

Effect of downsampling on the spectrum





The spectrum of the downsampled signal is periodic with 2π
and is an amplitude and frequency scaled version of the original spectrum

Downsampling expands the spectrum and can introduce aliasing (overlapping spectral copies)

Amplitude Scaling

Frequency Scaling

Periodicity with 2π

$$X_d(\Omega)=\boxed{\frac{1}{M}}\sum_{i=0}^{M-1} X_s\left(\boxed{\frac{\Omega}{M}}-\frac{\boxed{2\pi i}}{M}\right)\quad , \quad X_d(\Omega)=\frac{1}{M}\sum_{i=0}^{M-1} X\left(\frac{\Omega}{M}-\frac{2\pi i}{M}\right)$$

Discrete Time Sampling

Downsampling – Analogue Frequency

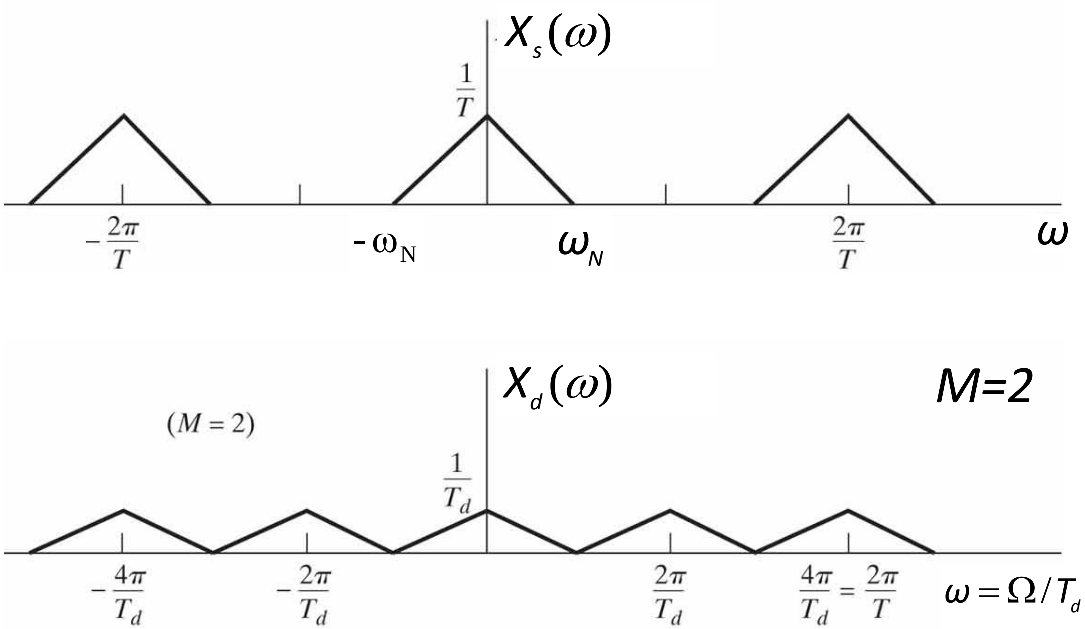
Sampling rate reduction by an integer factor

$X_s(\omega)$ – Sampled Signal

Spectrum of the sampled signal x_s consisting of periodic copies of $X_c(\omega)$ centred at $k\omega_s$, k=0,1,2,….

$X_d(\omega)$–Downsampled Signal (M=2)

Spectrum of the down-sampled signal $x_d[n]$. Downsampling brings the periodic copies closer to each other (centred at $k(\omega_s / M)$)



No aliasing in this case

$$\omega_s \geq M\,2\omega_N \Rightarrow \omega_{sd} = \frac{\omega_s}{M} \geq 2\omega_N$$

Downsampling reduces the sampling frequency - brings spectral copies closer to each other

Downsampling – Digital Frequency

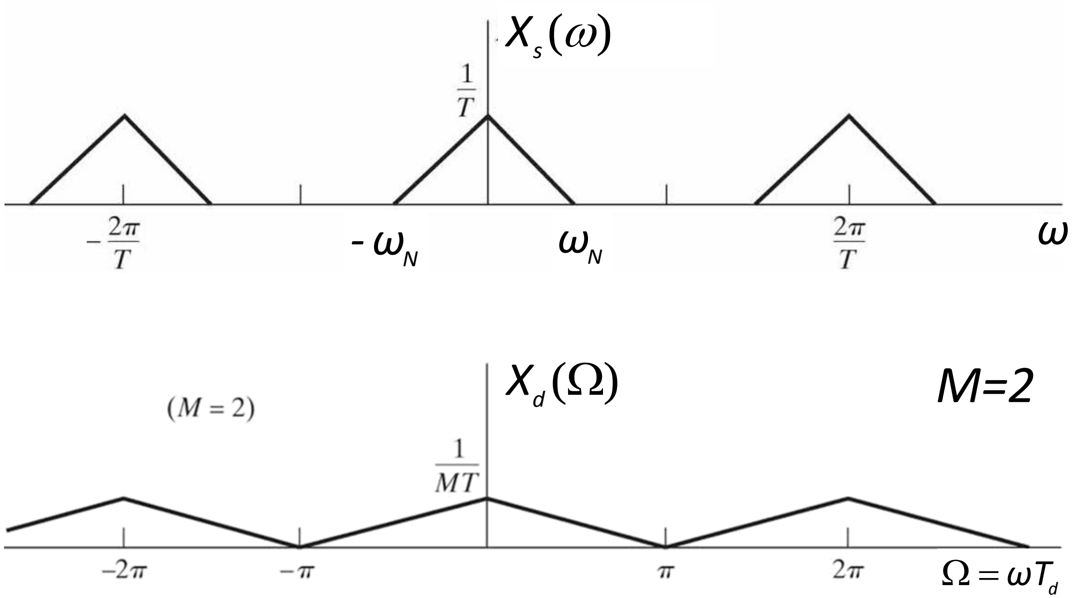
Sampling rate reduction by an integer factor

$X_s(\Omega)$ – Sampled Signal

Spectrum of the sampled signal *x[n]* consisting of periodic copies of $X_c(\Omega)$ centred at k2π, k=0,1,2…where $\Omega = \omega T_s$

$X_d(\Omega)$–Downsampled signal (M=2)

Spectrum of the downsampled signal $x_d[n] = x[2n]$. $\Omega = \omega T_d$. Downsampling scaled the amplitude & doubled the bandwidth of $X_s(\Omega)$



$$\Omega_d = \omega T_d = \omega M T_s = M\Omega$$

$$F_d = \frac{f}{f_{sd}} = \frac{f}{f_s / M} = M\frac{f}{f_s} = MF$$

No aliasing in this case

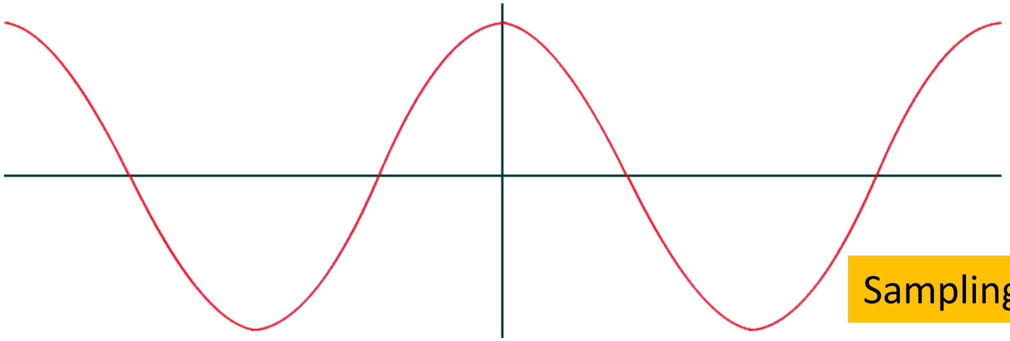
$$X(\Omega) = 0,\ |\Omega| \geq \frac{\pi}{M} \Rightarrow X_d(\Omega) = 0,\ |\Omega| \geq \pi$$

Downsampling stretches the digital frequency spectrum - introduces higher <u>digital</u> frequencies

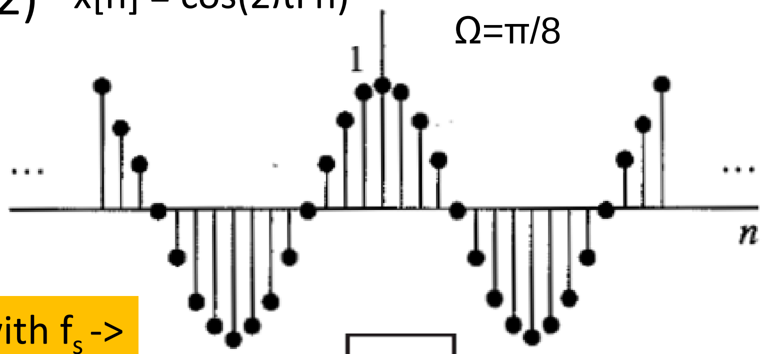
Downsampling – Digital & Analogue Frequency

Sampling rate reduction by an integer factor

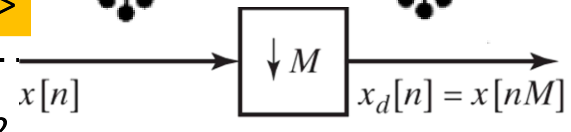
1) $x_1(t) = \cos(2\pi f_1 t)$



2) $x[n] = \cos(2\pi F n)$ $F=1/16$ $\Omega=\pi/8$

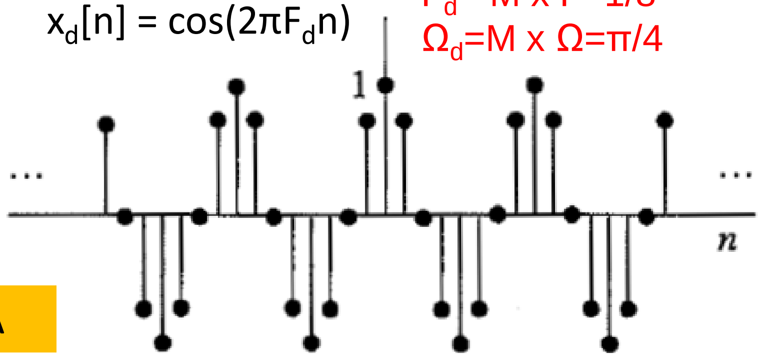


Sampling with f_s ->



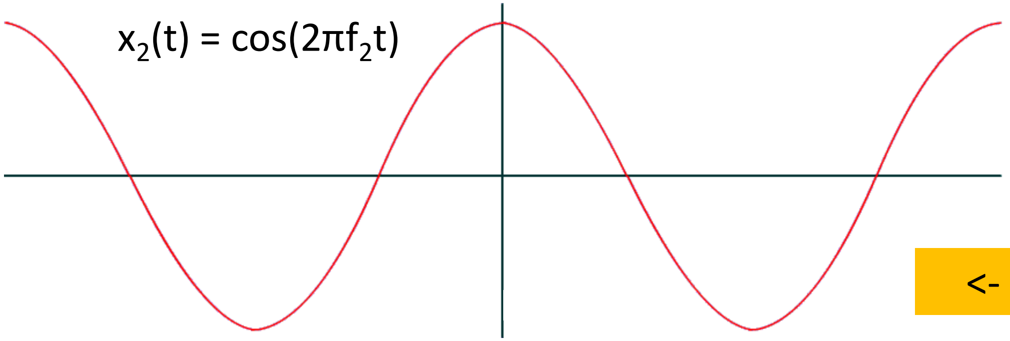
3) *M=2*

$x_d[n] = \cos(2\pi F_d n)$ $F_d = M \times F = 1/8$ $\Omega_d = M \times \Omega = \pi/4$



4) $x_1(t) = x_2(t)$ if $f_s \geq M2f_N$ or $F_N \leq 1/2M$ (no aliasing)

$x_2(t) = \cos(2\pi f_2 t)$



<- D/A

Downsampling stretches the digital frequency spectrum - introduces higher <u>digital</u> frequencies

Discrete Time Sampling

Downsampling – Digital Frequency

a) $X_s(\Omega)$ – Normalised Frequency

$\Omega = \omega T$, $\Omega_N = \pi/2$

b) $X_d(\Omega)$ – Downsampling (M=3)

Down-sampling with M=3 triples the bandwidth of $X_s(\Omega)$ resulting in aliasing

c) $H_d(\Omega)$ – Pre-filtering

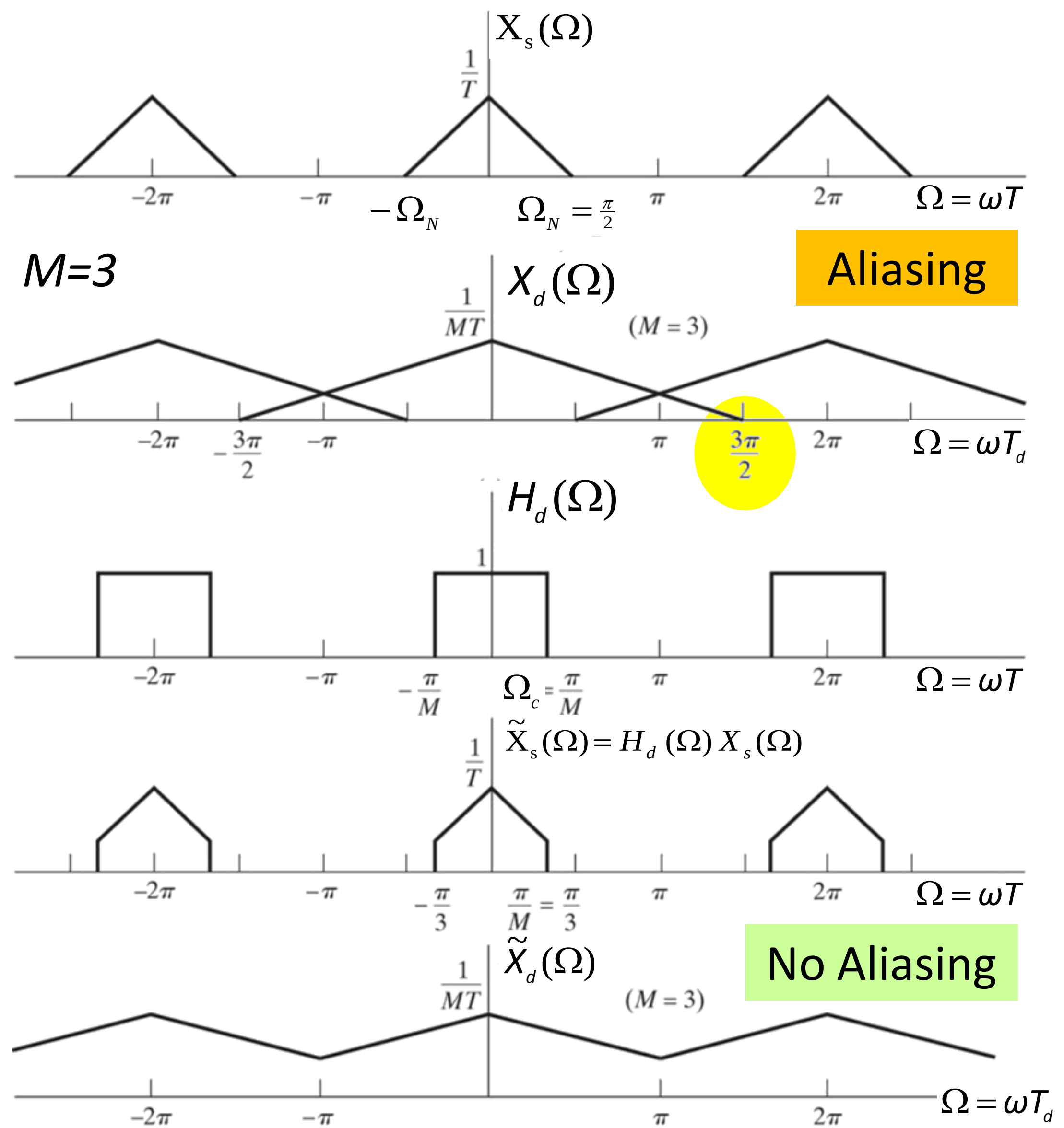
Frequency response of ideal discrete time low pass filter used for removing frequencies above π/M in $X_s(\Omega)$

d) $\tilde{X}_s(\Omega)$ – Spectrum of filtered signal

Spectrum of filtered signal *x[n]*

b) $\tilde{X}_d(\Omega)$ –Downsampling (M=3)

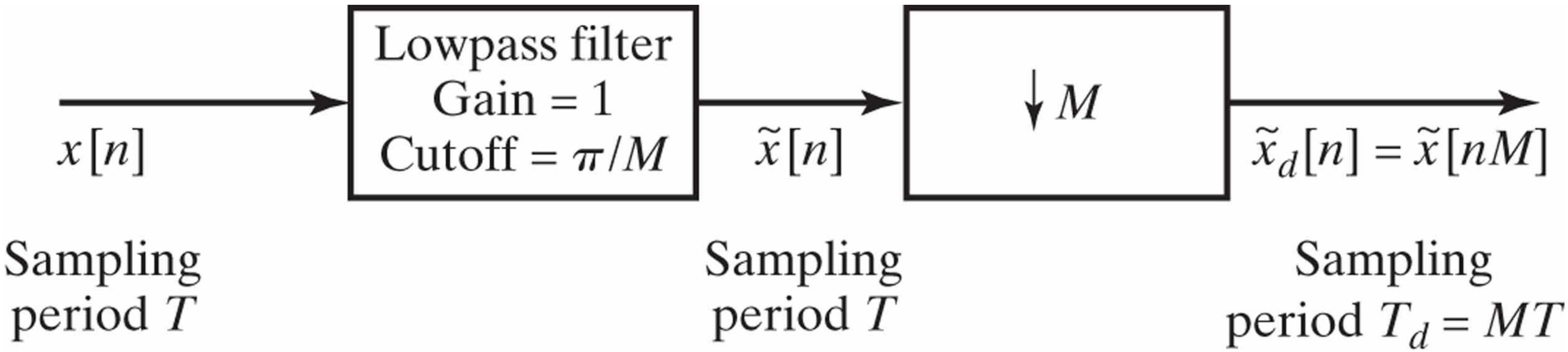
Down-sampling of filtered DT signal. No aliasing but we have lost some of the original high frequencies



Discrete Time Sampling

Downsampling

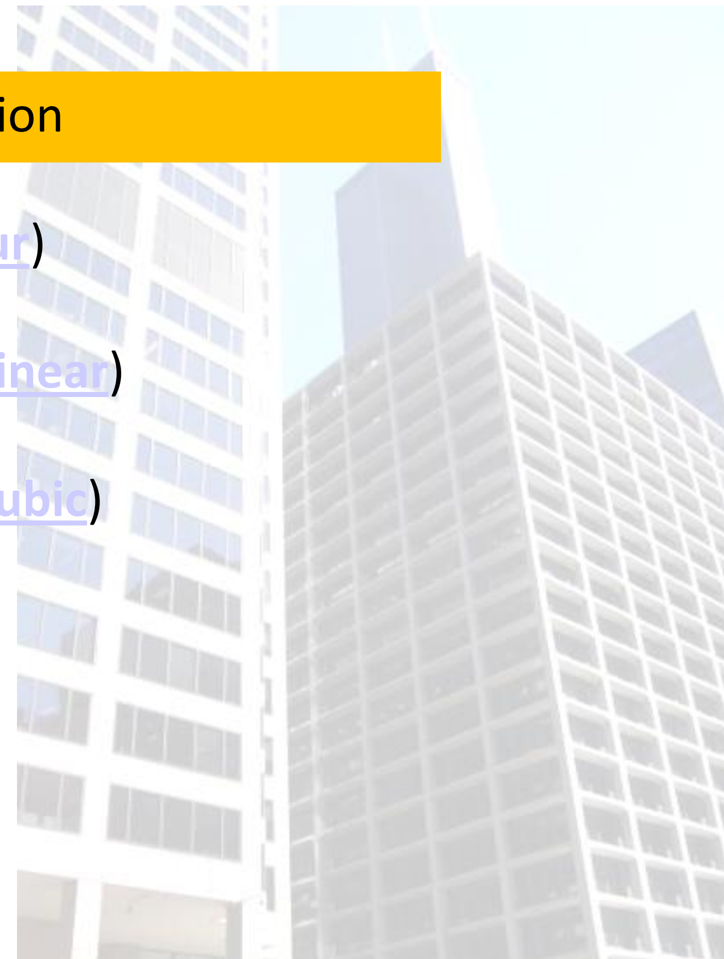
General system for sampling rate reduction by *M* (Decimation)



Discrete Time Sampling

Downsampling

Example: Image Downsampling



Resize image to 25% (M=4) of the original resolution

1. **without** pre-filtering (Nearest Neighbour)

2. **with** pre-filtering using a linear filter (Bilinear)

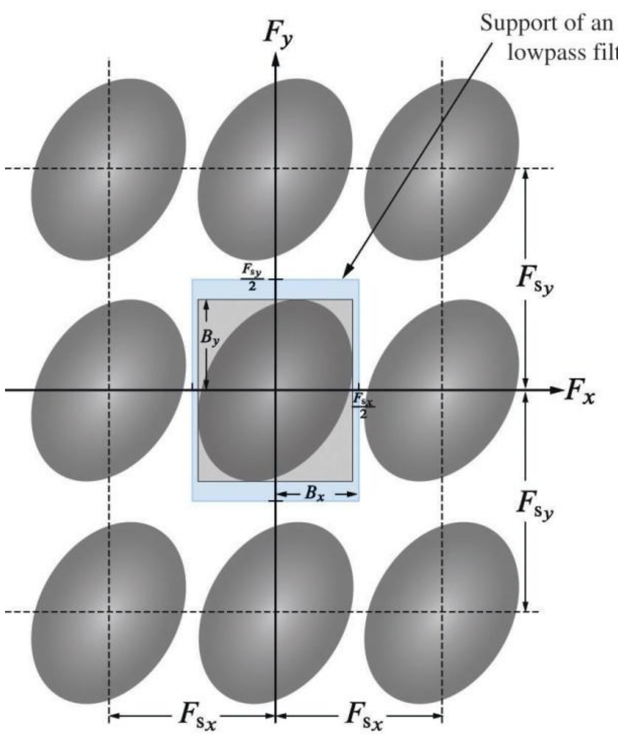
3. **with** pre-filtering using a cubic filter (Bicubic)

Always display the image at 1:1 – zooming in or making the image smaller will create additional artifacts due to upsampling and downsampling respectively

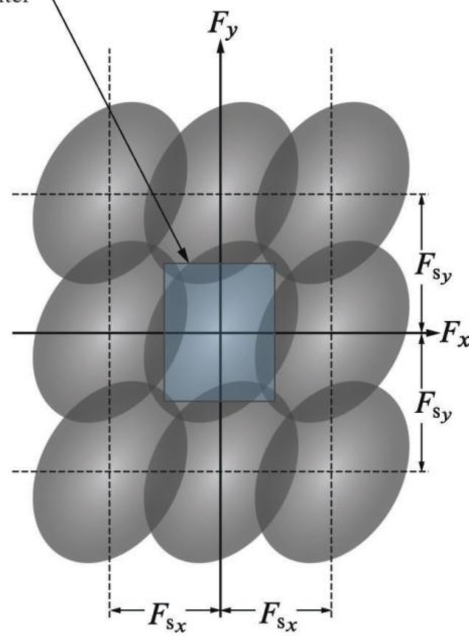
Image (2D Signal) Downsampling

Aliasing and Artifacts

2D Spectrum of image with and without aliasing

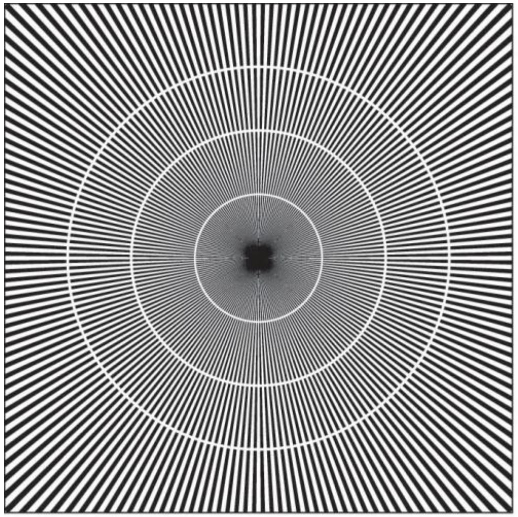


(a) No Aliasing

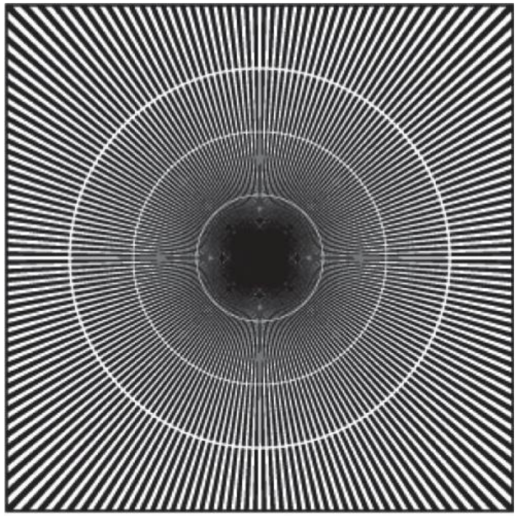


(b) With Aliasing

(a)



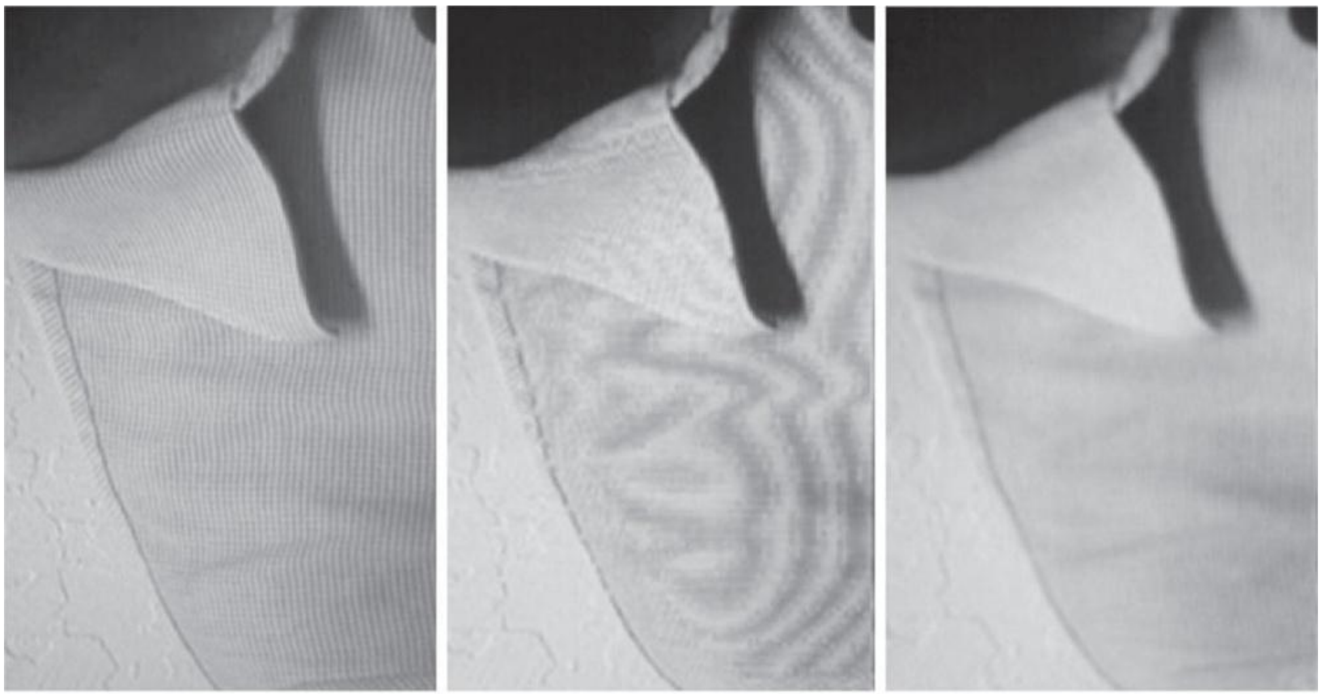
(b)



Discrete Time Sampling

Image (2D Signal) Downsampling

Aliasing and Artifacts



(a) Original image

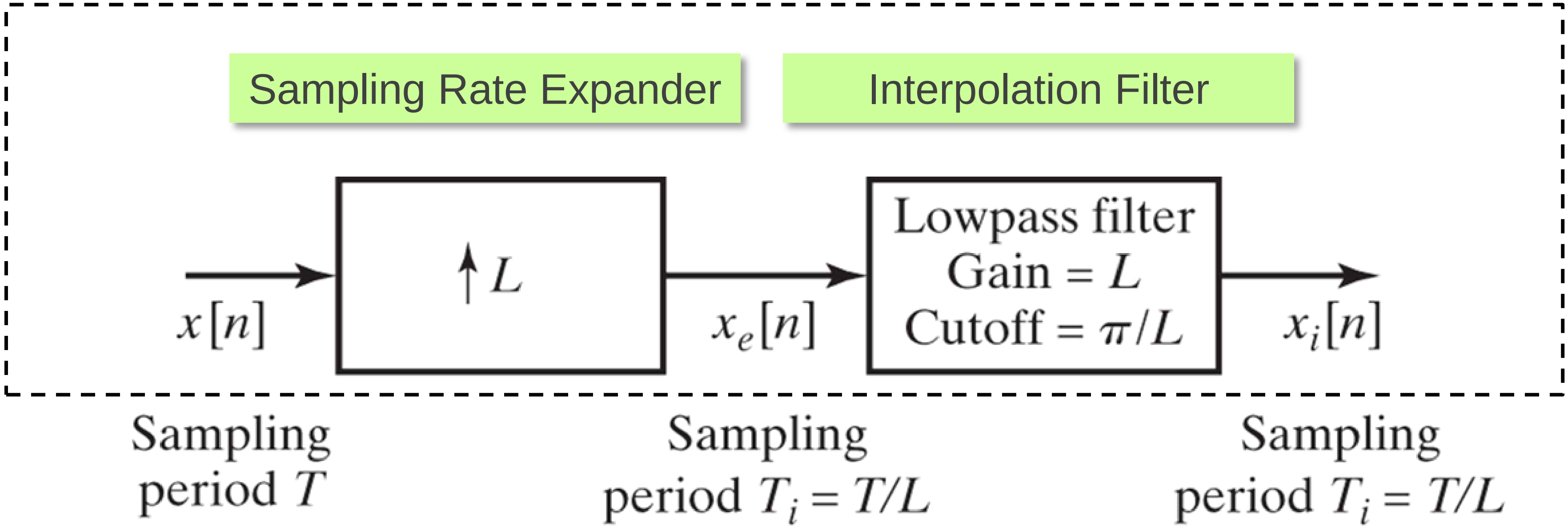
(b) Downsampled (50%) without pre-filtering

(c) Downsampled (50%) with pre-filtering

Upsampling

General system for increasing the sampling rate by *L* (Interpolation)

Interpolator



Sampling Rate f_s

Sampling Rate Lf_s

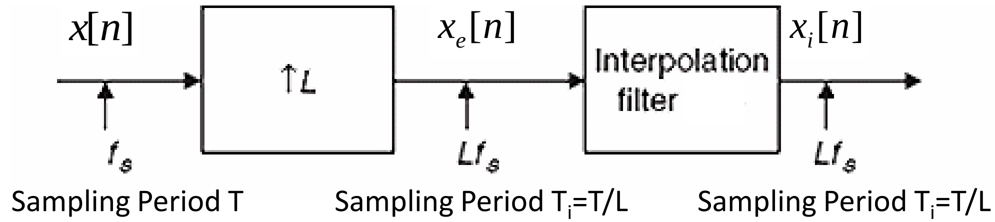
Sampling Rate Lf_s

Discrete Time Sampling

Upsampling – Time Domain

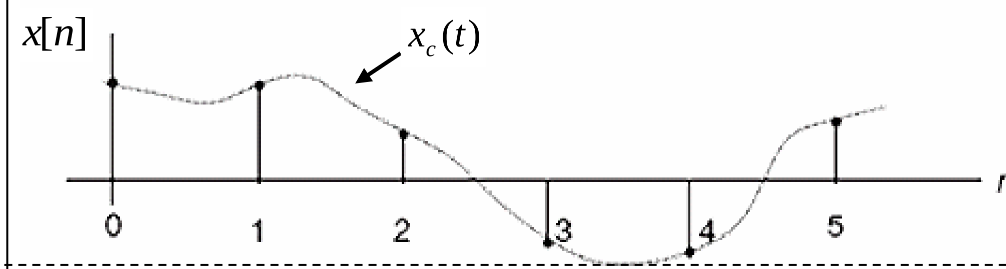
Time domain description

$$\left.\begin{aligned} x_i[n] &= x_c(nT_i) \\ x[n] &= x_c(nT) \\ T_i &= T/L \end{aligned}\right\} \begin{aligned} & x_i[n] = x[n/L] = x_c(nT/L) \\ & n = 0, \pm L, \pm 2L, \ldots \end{aligned}$$



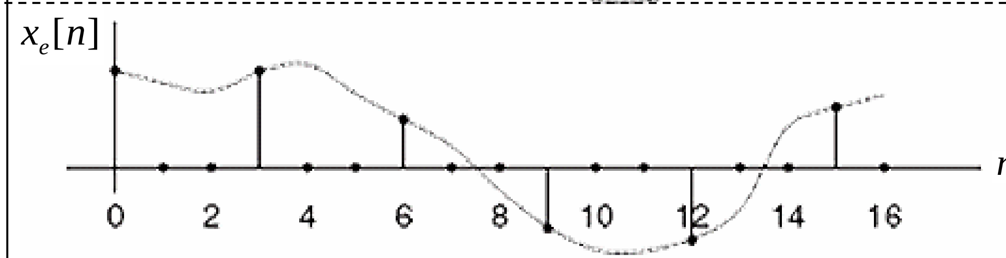
Sampling

Sampling of CT signal $x_c(t)$ with sampling period T resulting in DT signal $x[n]=x_c(nT)$



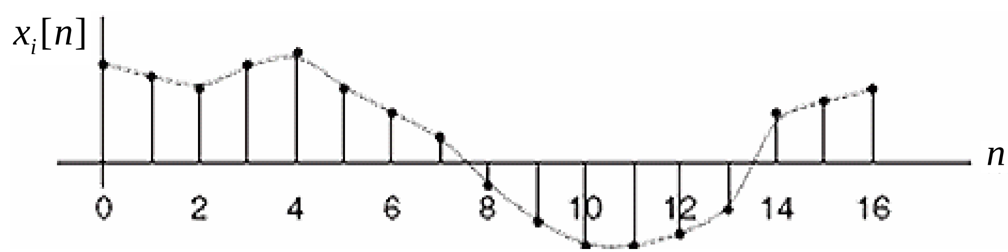
Signal Expansion

Insertion of L-1 zeros between every two samples of $x[n]$ resulting in signal $x_e[n]$



Interpolation

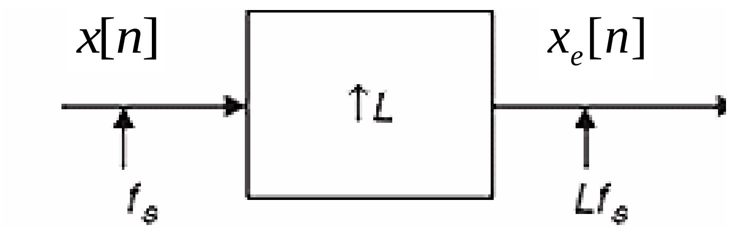
Fill in the missing samples of $x_e[n]$ by connecting the dots to form signal $x_i[n]= x_c(nT/L)$



Discrete Time Sampling

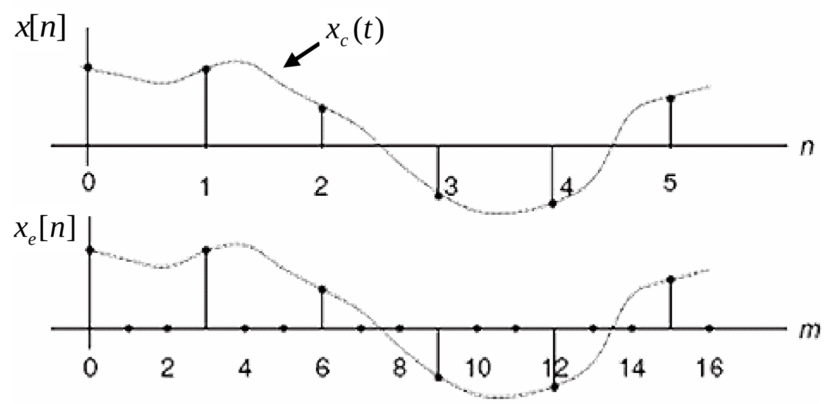
Upsampling – Frequency Domain

Effect of signal expander on the spectrum



Sampling Period T

Sampling Period T_i=T/L



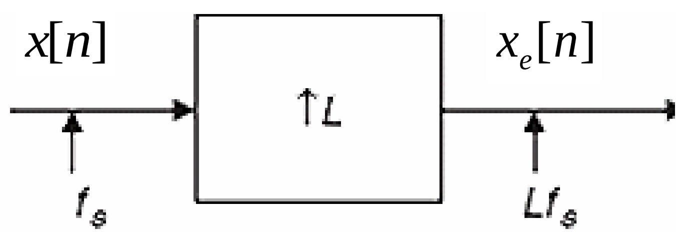
Definition of $x_e[n]$: $x_e[n]=\begin{cases} x[n/L], & n=0,\pm L,\pm 2L,\ldots \\ 0, & \text{otherwise} \end{cases} \Leftrightarrow x_e[n]=\sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]$

DTFT of $x_e[n]$: $X_e[\Omega]=\sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]\right)e^{-j\Omega n}=\sum_{k=-\infty}^{\infty} x[k]\, e^{-j\Omega k\, L}$

$$X_e[\Omega]=X[\Omega L],\quad \Omega=\omega T \xrightarrow{Expander} \Omega=\omega(T/L)\Leftrightarrow \Omega=\omega T_i$$

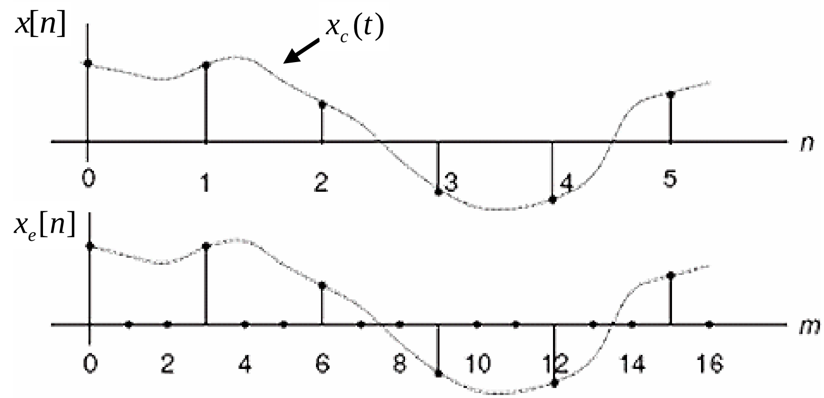
Upsampling – Frequency Domain

Effect of signal expander on the spectrum



Sampling Period T

Sampling Period T_i=T/L



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

Upsampling shrinks the digital frequency spectrum and introduces spectral copies in the fundamental frequency range

Ω Scaling factor

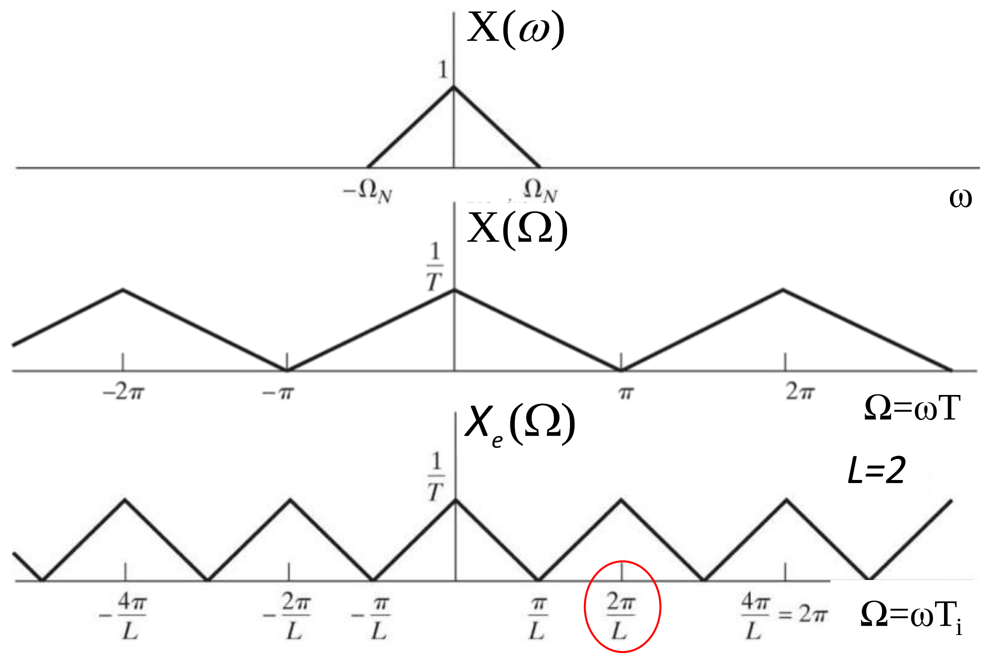
$X[\Omega]$

$$X_e[\Omega]=X[\Omega L],\quad \Omega=\omega T \xrightarrow{Expander} \Omega=\omega(T/L)\Leftrightarrow \Omega=\omega T_i$$

Upsampling – Frequency Domain

Spectrum of CT signal

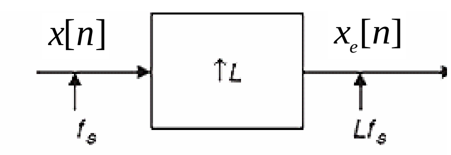
$$x(t) \overset{FT}{\longleftrightarrow} X(\omega)$$



Spectrum of sampled signal

$$x[n] \overset{DTFT}{\longleftrightarrow} X(\Omega) \quad , \Omega = \omega T$$

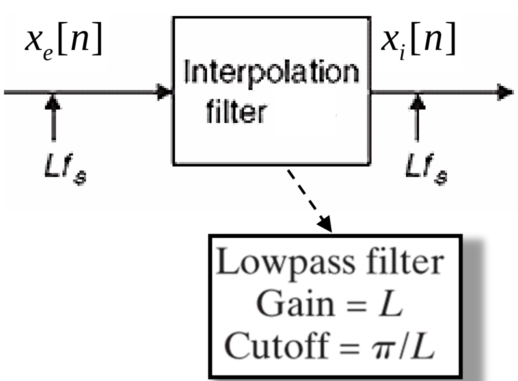
Spectrum after sampling rate expansion

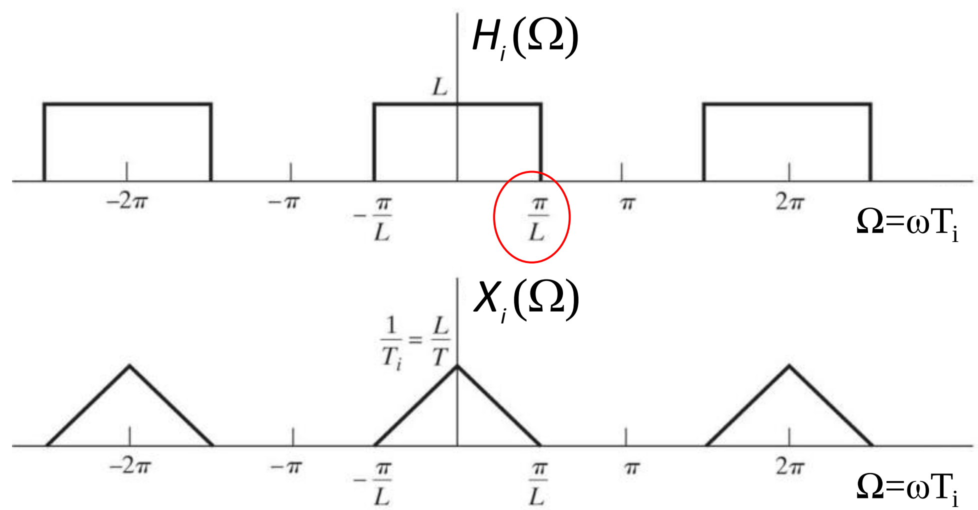


$$x_e[n] \overset{DTFT}{\longleftrightarrow} X_e(\Omega)$$

$$\Omega = \omega T_i$$

Keep only spectral copies located at *k* x *2π* (clear fundamental range of unwanted copies)





Interpolation Filters

Linear Interpolation

Impulse Response (Time Domain)

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

Filtering process (convolution)

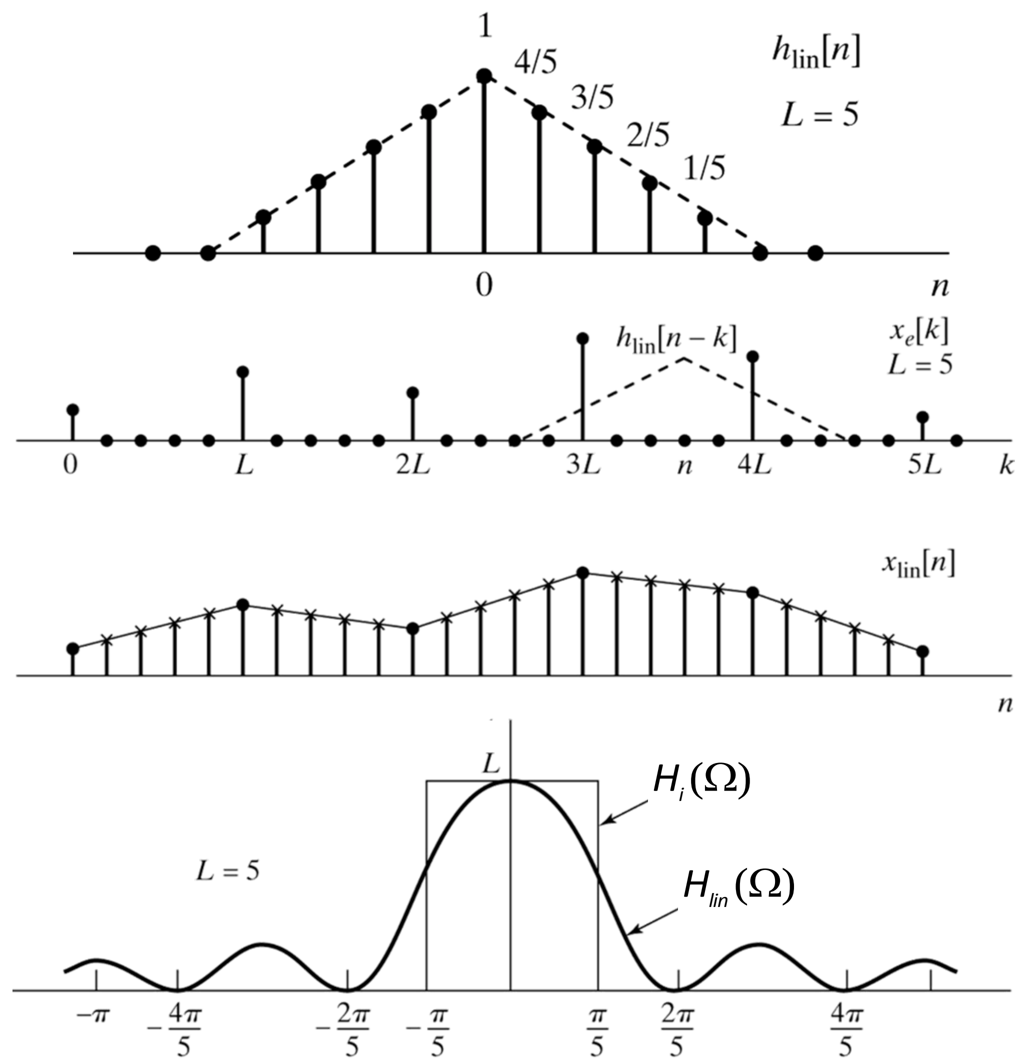
$$x_{lin}[n] = \sum_{k=n-L+1}^{n+L-1} x_e[k]\, h_{lin}[n-k]$$

Interpolated samples assumed to lie on a straight line connecting the 2 original sample values.

Frequency Response

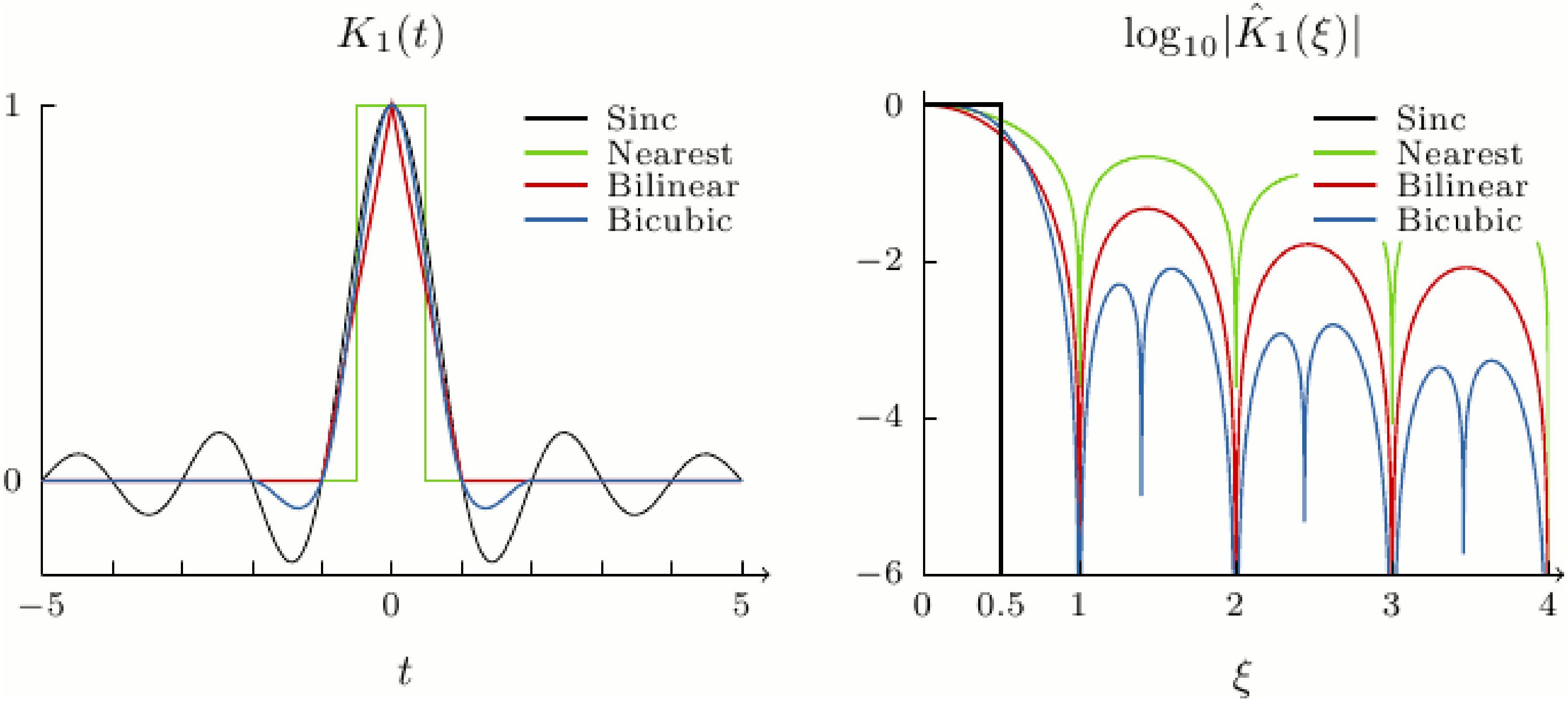
$$H_{lin}[\Omega] = \frac{1}{L}\left[\frac{\sin(\Omega L/2)}{\sin(\Omega/2)}\right]^2$$

Compare with ideal frequency response



Interpolation Filters

Comparison

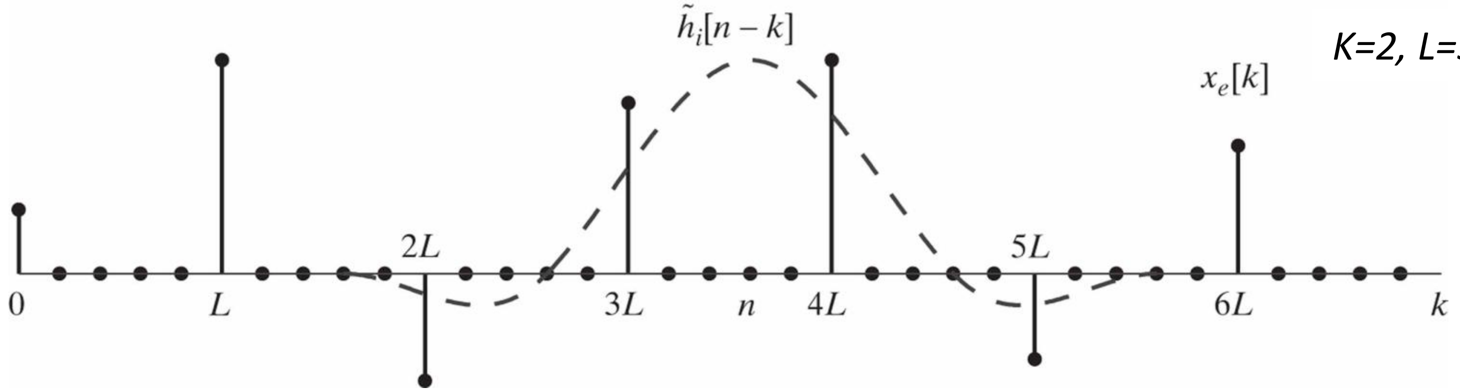


Comparison of the nearest neighbour, bilinear, and bicubic kernels and the sinc.

Filters with longer impulse response are necessary in order to approximate the ideal bandlimited response of the Sinc function

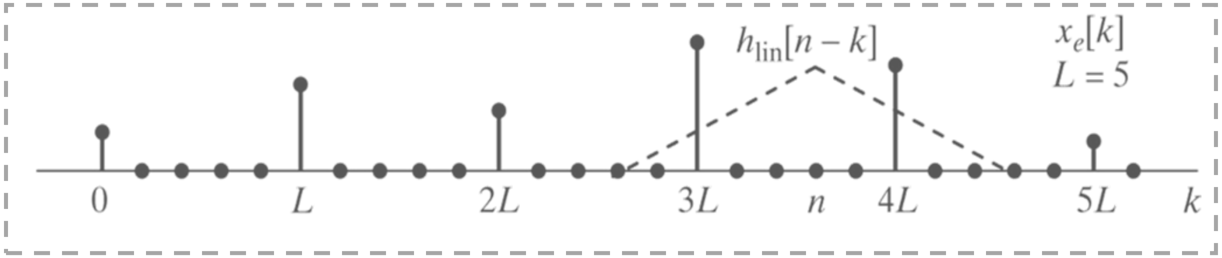
Interpolation Filters

Cubic filter



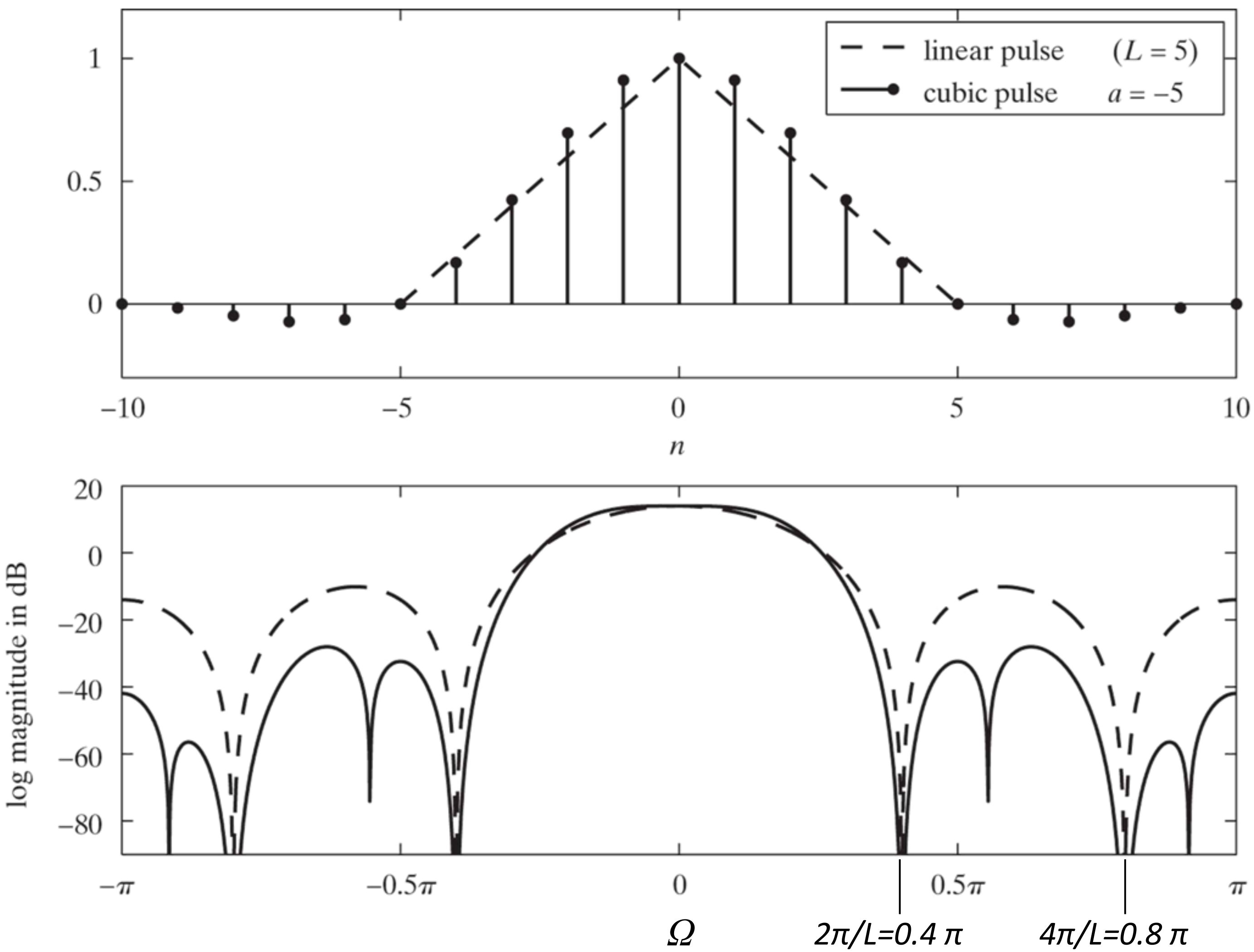
$$\tilde{h}_i[n] = \begin{cases} (a+2)|n/L|^3 - (a+3)|n/L|^2 + 1, & 0 \le n \le L \\ a|n/L|^3 - 5|n/L|^2 + 8a|n/L| - 4a, & L \le n \le 2L \\ 0, & \text{otherwise} \end{cases}$$

4 original samples participate in cubic interpolation (2 with linear interpolation)



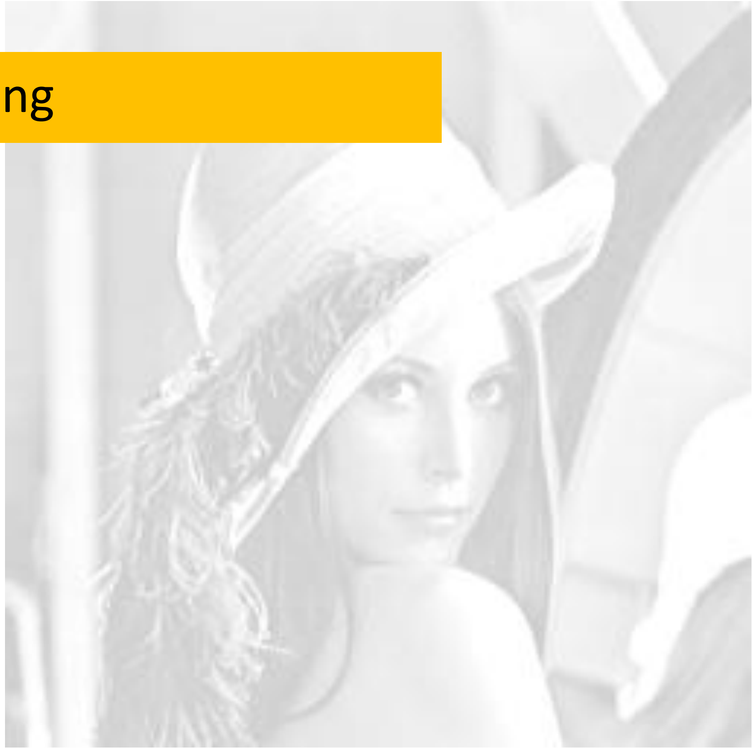
Interpolation Filters

Linear vs. Cubic



Upsampling

Example: Image Upsampling



Resize image to 200% (L=2) of the original size using

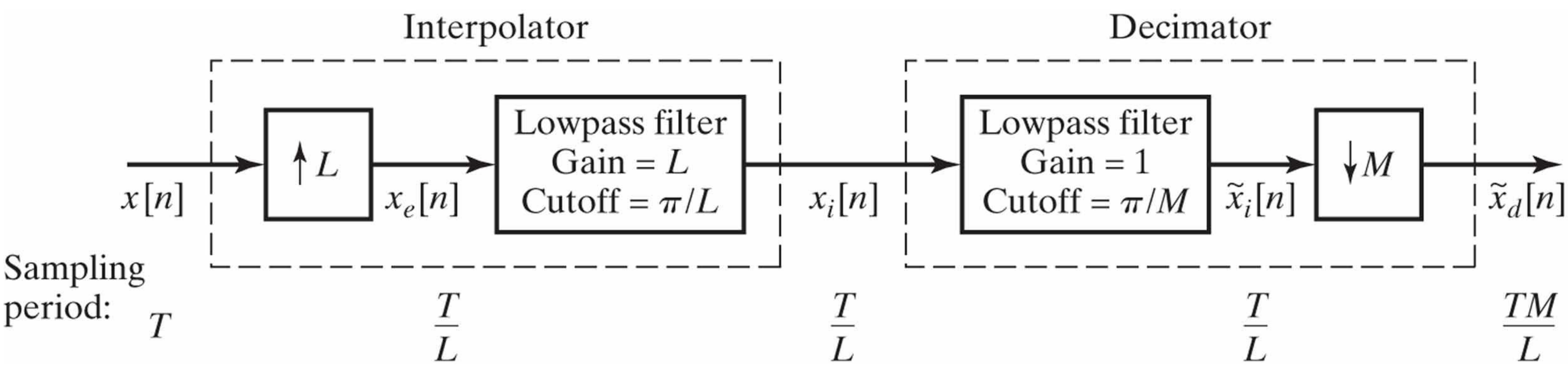
1. Nearest Neighbour interpolation
2. Bilinear interpolation
3. Bicubic interpolation

Always display the image at 1:1 – zooming in or making the image smaller will create additional artifacts due to upsampling and downsampling respectively

Discrete Time Sampling

Sampling rate change by an non-integer factor

General system for changing the sampling rate by a non-integer factor



E.g. If L=100 and M=101 then the effective sampling period is 1.01*T*

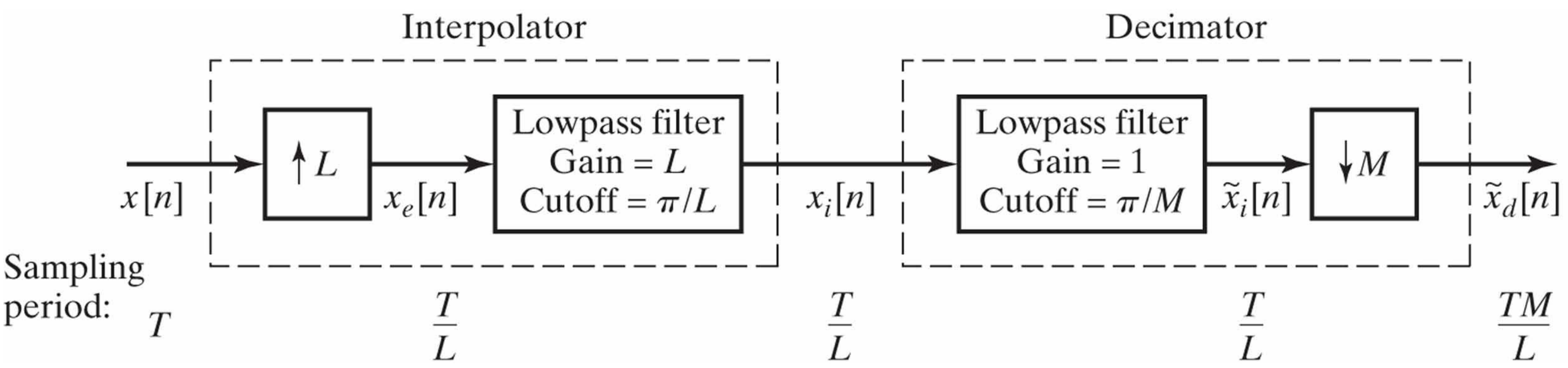
Depending on the ratio M/L we have an increase or decrease in the sampling rate:

- **M>L** (M/L >1) - Net increase in sampling period => Decrease in sampling rate
- **M<L** (M/L <1) - Net decrease in sampling period => Increase in sampling rate

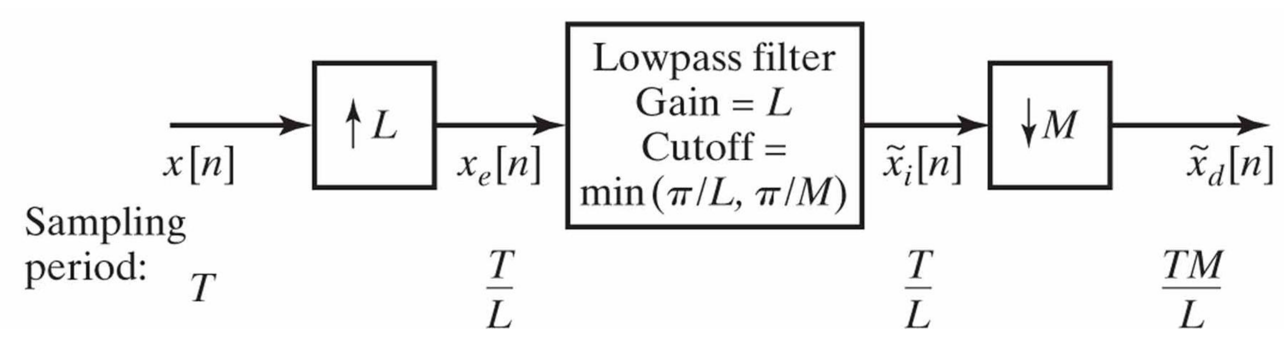
Discrete Time Sampling

Sampling rate change by an non-integer factor

General system for changing the sampling rate by a non-integer factor



Simplified system in which the decimation and interpolation filters are combined.



Cut-off frequency of filter depends on the ratio M/L

- **M>L** => *$\pi/M < \pi/L$ =>* *$Min(\pi/M,\pi/L) = \pi/M$*

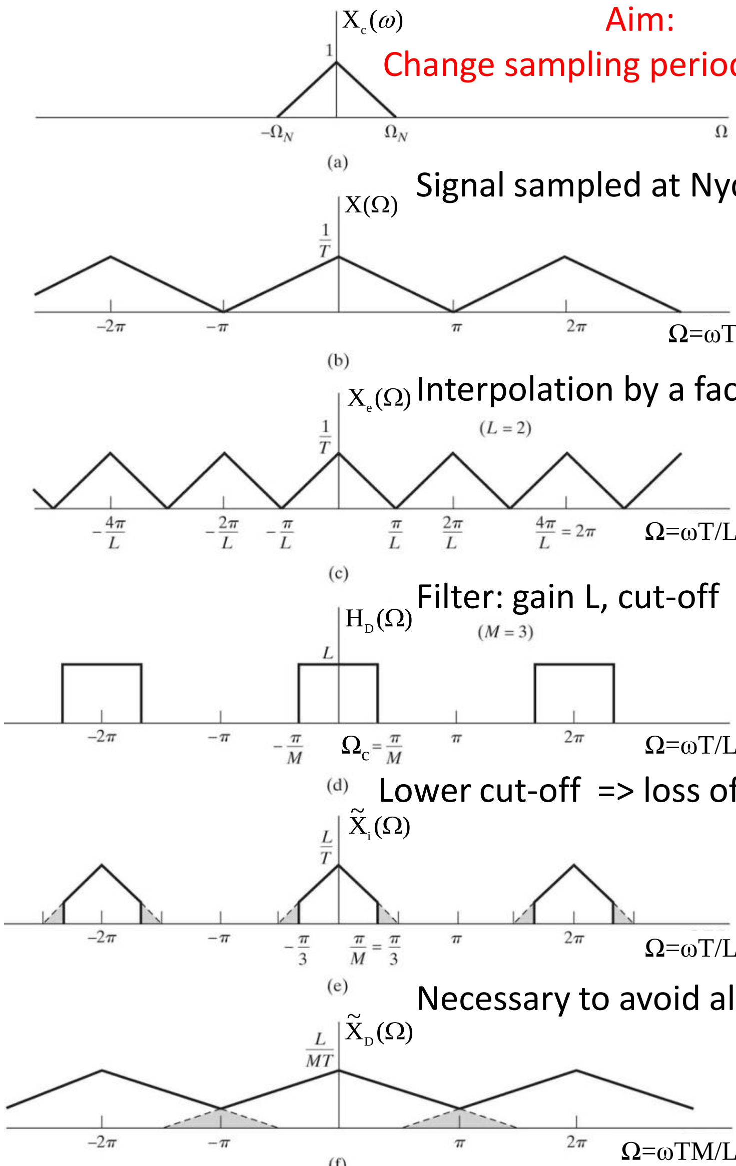
  Cut-off freq. = π /M

- **M<L** => *$\pi /M > \pi/L$ =>* *$Min(\pi/M,\pi/L) = \pi/L$*

  Cut-off freq. = π /L

Discrete Time Sampling

Sampling rate change by an non-integer factor



Aim:
Change sampling period to (3/2)T

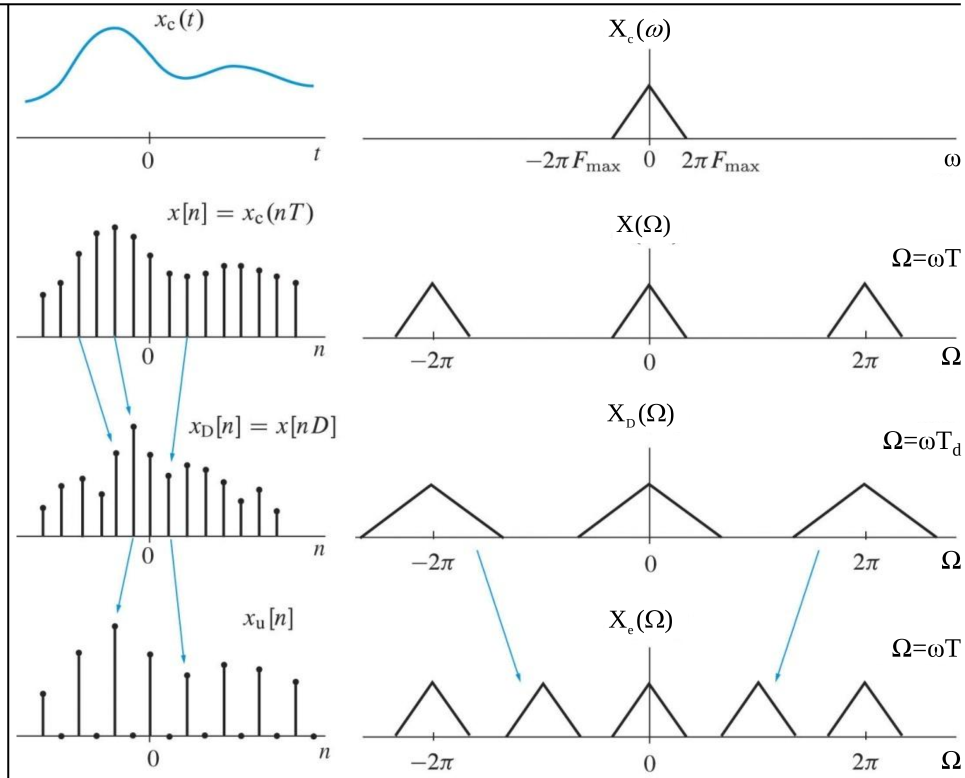
Signal sampled at Nyquist Rate

Interpolation by a factor L=2

Filter: gain L, cut-off π/M (M=3)

Lower cut-off => loss of spectrum

Necessary to avoid aliasing



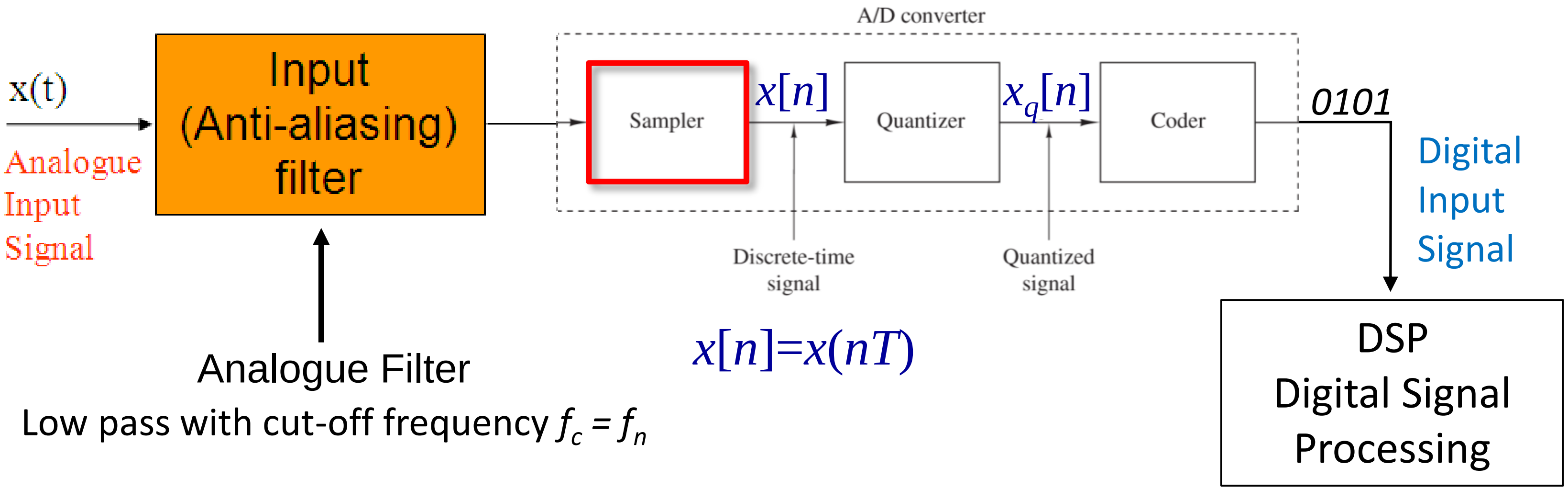
Downsampling first followed by upsampling would make things worse even in the case shown above where the sampling frequency is a lot higher than the Nyquist rate

Why?

Discrete Time Sampling

Avoiding Aliasing in Analogue to Digital Conversion

Anti-aliasing Filter



Analogue Filter

Low pass with cut-off frequency $f_c = f_n$

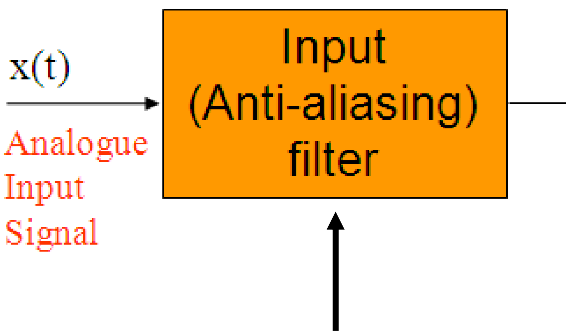
$x[n]=x(nT)$

Apply low-pass filtering prior to sampling in order to ensure that the input CT signal is band-limited below the Nyquist frequency i.e. $X(f)=0,\ \forall |f| > f_n, \quad f_n = f_s/2$

Discrete Time Sampling

Oversampling Analogue to Digital Conversion

Analogue Anti-aliasing Filter

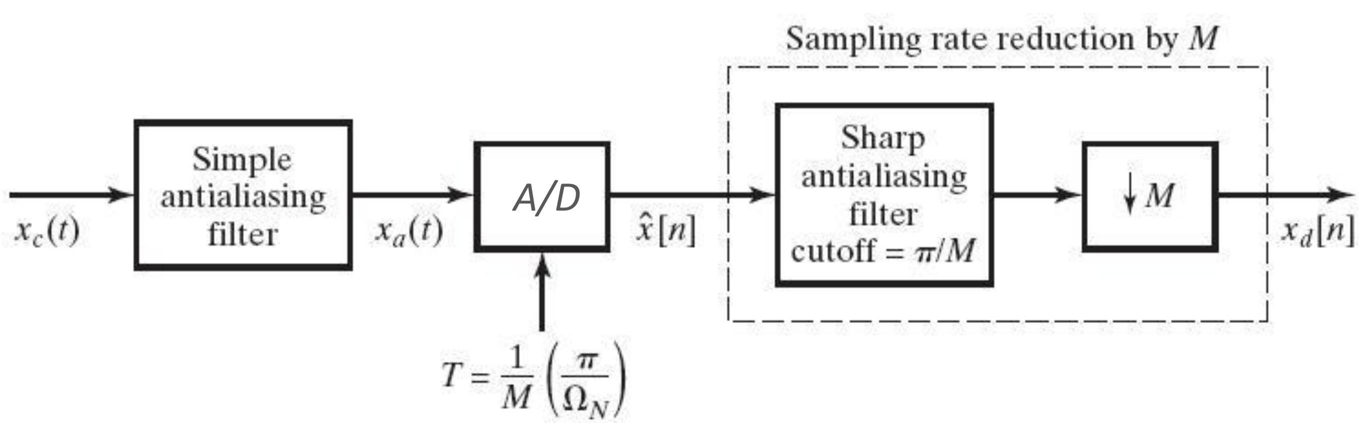


Analogue Filter

Low pass with cut-off frequency $f_c = f_n$

- Analogue filter with sharp cut-off required
- Possible but expensive - Analogue filter may account for large part of cost of system using inexpensive but powerful digital processors
- Sharp cut-off analogue filters generally have highly nonlinear phase response
- If sampling rate is variable, adjustable filters would have to be used

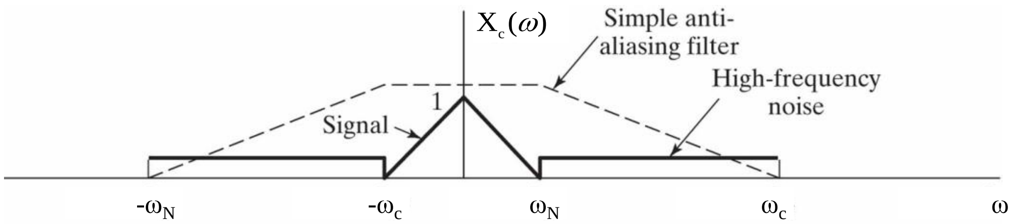
Use oversampled A/D conversion to simplify a continuous-time anti-aliasing filter



Oversampling Analogue to Digital Conversion

a) $X_c(\omega)$ – Analogue Input Signal

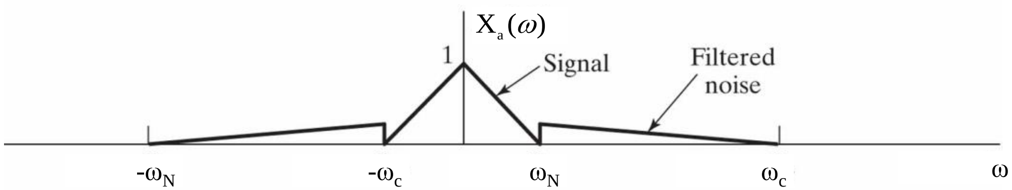
Fourier transform of input signal that occupies band $|\omega| < \omega_N$ plus the FT of high frequency "noise". Also shown frequency response of analogue AA filter



(a)

b) $X_a(\omega)$ – FT of Filter Output

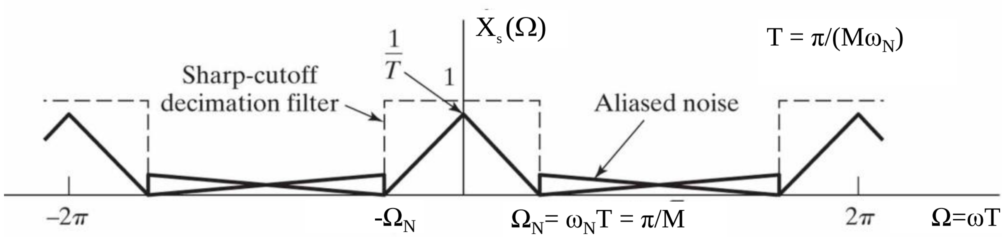
Spectrum of filtered analogue input signal



(b)

c) $X_s(\Omega)$ – Sampling

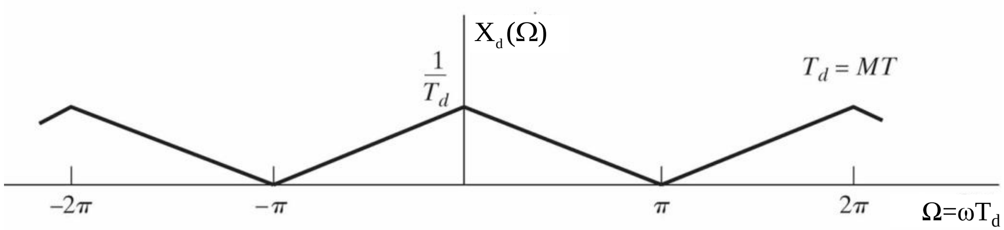
DTFT of the sampled signal with a sampling rate $\Omega \geq (\Omega_N + \Omega_c)$. Also shown the frequency response of the sharp cut-off digital filter. Aliasing affects the noise but not the signal.



(c)

d) $X_d(\Omega)$ – Downsampling

New sampling period is M times the original sampling period T (or equivalent the new sampling rate is the original sampling rate divided by M)



(d)

Oversampling Analogue to Digital Conversion

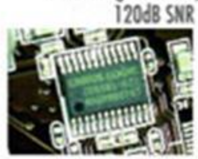
Xonar HDAV1.3
Enjoy 100% Blu-Ray Quality Audio with HDMI 1.3 Compliance and Video Enhancement
- Full-HD Audio/Video digital output
- HDMI v1.3a compliance
- Non-downsampled 192kHz Blu-Ray audio
- Splendid Technology Video enhancement
- Complete Dolby True HD & DTS Master Audio support

Xonar HDAV1.3 invents its fantastic audio quality with proprietary ASUS AV200, high quality oversampling digital to analog converters (Burr-Brown PCM1796, 123dB SNR), National Semiconductor LM4562 OP AMP, and analog-to-digital converter (Cirrus Logic CS5381, 120dB SNR).

Burr-Brown PCM1796, 123dB SNR

ASUS AV200

Cirrus Logic CS5381, 120dB SNR



CIRRUS LOGIC® **CS5381**

120 dB, 192 kHz, Multi-Bit Audio A/D Converter

Features
- Advanced Multi-bit Delta-Sigma Architecture
- Supports Logic Levels Between 5 and 2.5 V
- Differential Analog Architecture

General Description

The CS5381 is a complete analog-to-digital converter for digital audio systems. It performs sampling, analog-to-digital conversion and anti-alias filtering, generating 24-bit values for both left and right inputs in serial form at sample rates up to 200 kHz per channel.

The CS5381 uses a 5th-order, multi-bit delta-sigma modulator followed by digital filtering and decimation, which removes the need for an external anti-alias filter. The ADC uses a differential architecture which provides excellent noise rejection.

The CS5381 is ideal for audio systems requiring wide dy-

CDB5381 Evaluation Board

The CS5381 uses a 5th-order, multi-bit delta-sigma modulator followed by digital filtering and decimation, which removes the need for an external anti-alias filter.

4.4 Analog Connections

The analog modulator samples the input at 6.144 MHz. The digital filter will reject signals within the stop-band of the filter. However, there is no rejection for input signals which are (n × 6.144 MHz) the digital