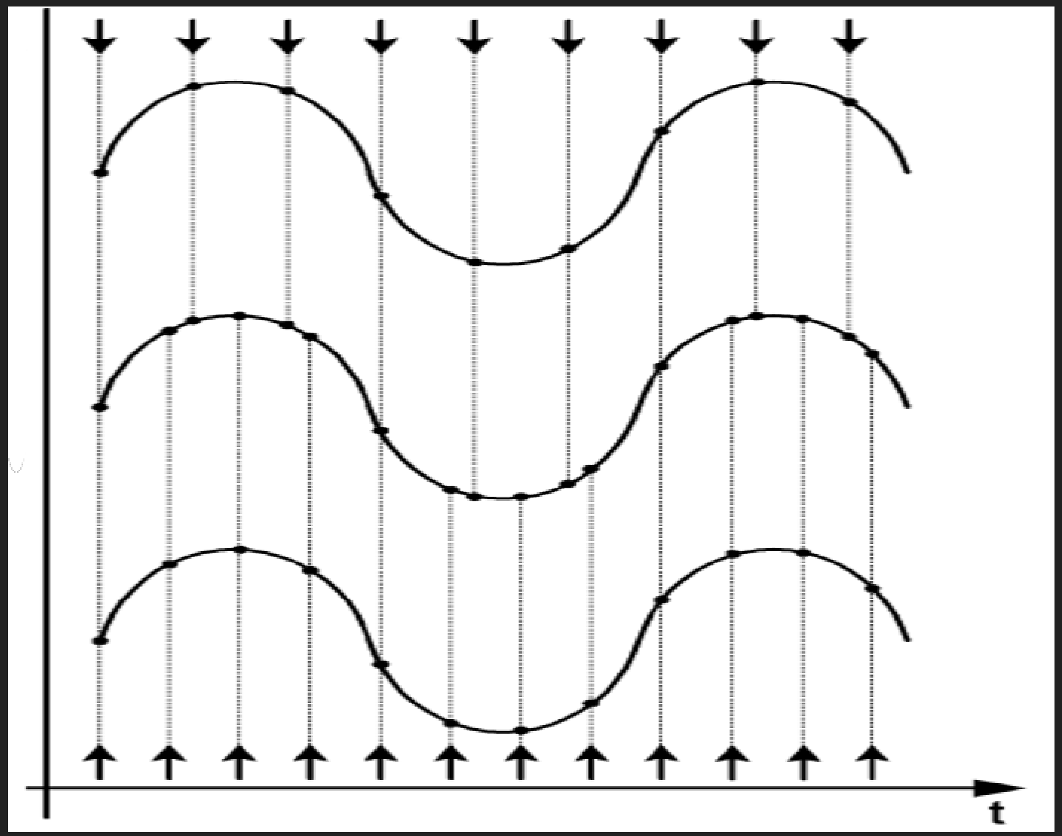
Digital Signal Processing for Music

Part 17: Sample Rate Conversion (SRC)

Andrew Beck

Sample Rate Conversion

From Wikipedia: "Changing the sampling rate of a discrete signal to obtain a new discrete representation of the underlying continusous signal"



Typical Applications

- Audio file/media conversion
- Word clock synchronization
- Oversampling
- DJing / Scratching

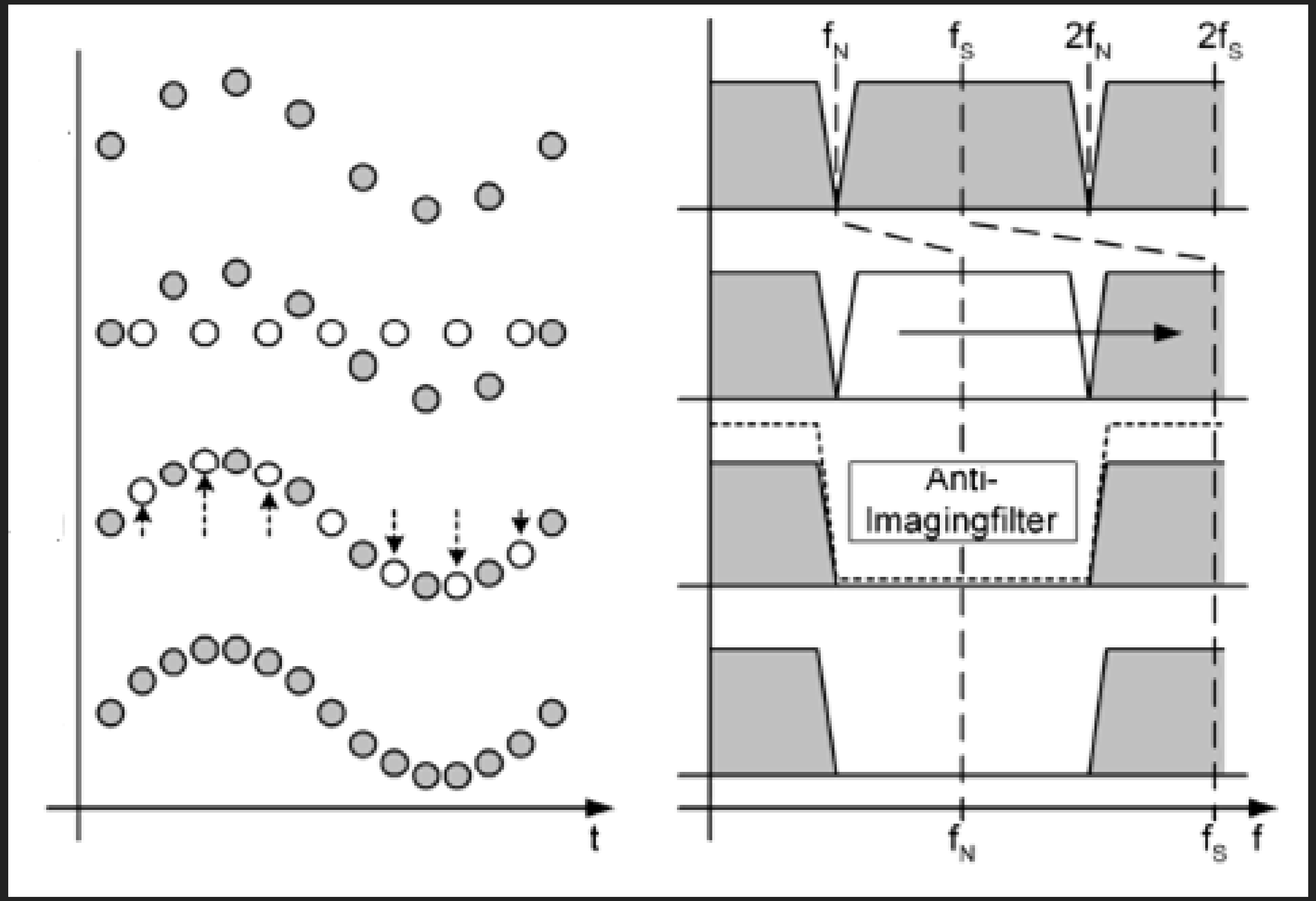
Introduction

- **Terminology**
  - *Synchronous*
    - Clock rates are coupled
    - Reasampling factor stays constant
  - *Asynchronous*
    - Clock rates are independent
    - Resampling factory may change
- **Ideal Result**
  - Spectrum in the used band unchanged
  - Spectral periodicity (determined by sample rate) changed

Upsampling by Inserting Zeros

Task: **Upsample by integer factor *L***

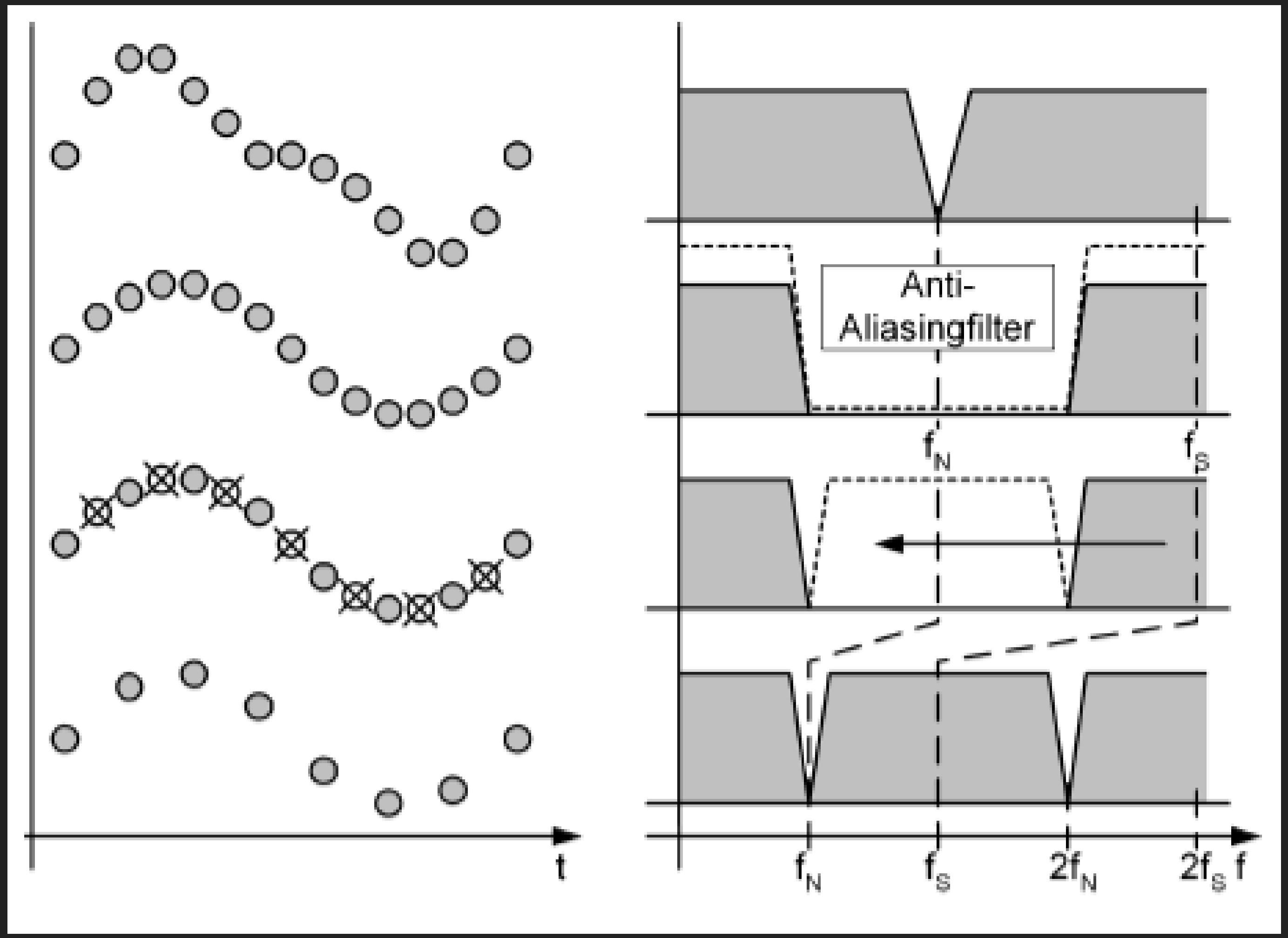
1. Insert *L* - 1 zeros between all samples
2. Apply Anti-Imaging filter

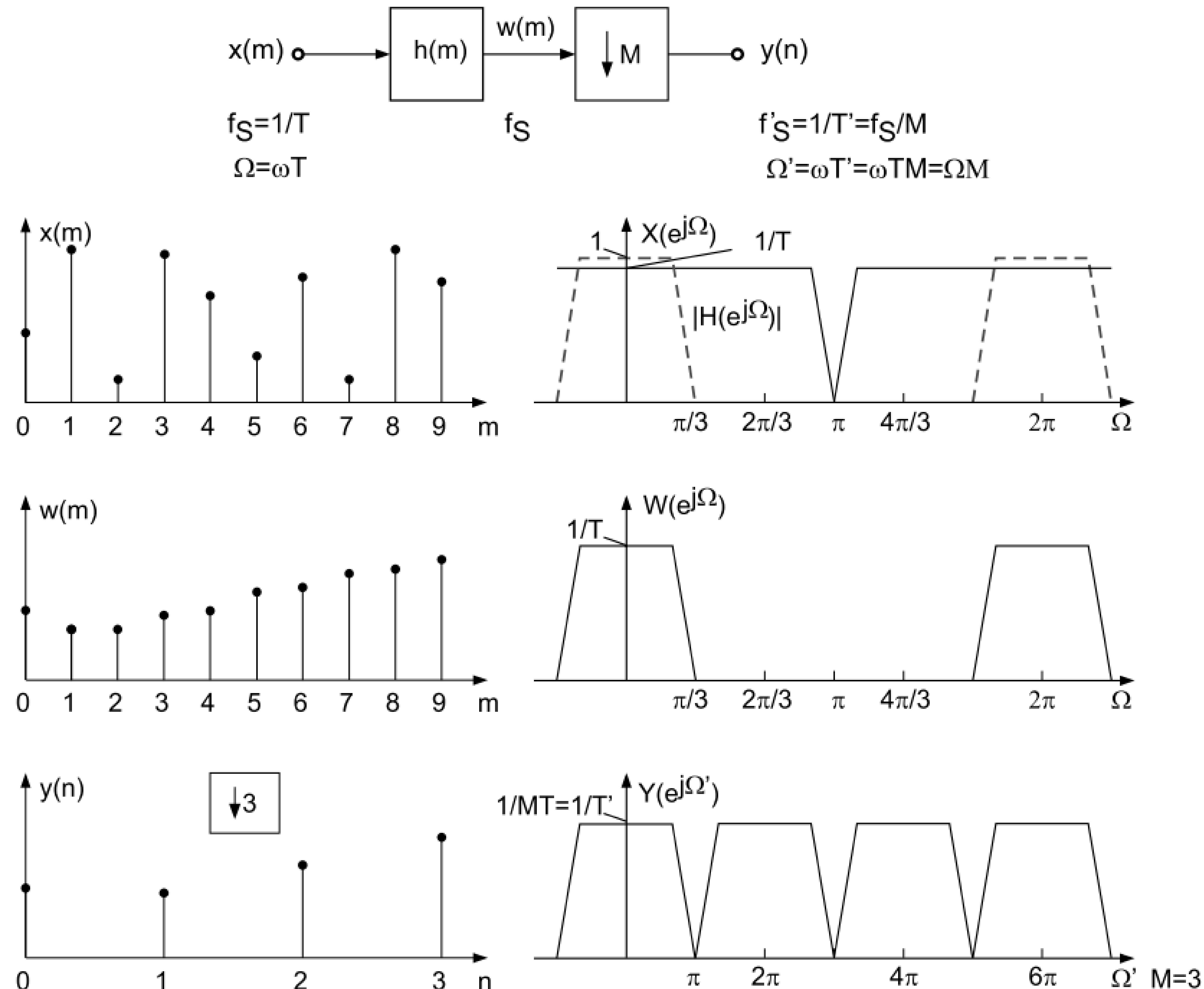


Downsampling by Removing Samples

Task: **Downsample by integer factor *M***

1. Apply anti-aliasing filter
2. Take every *M*th sample

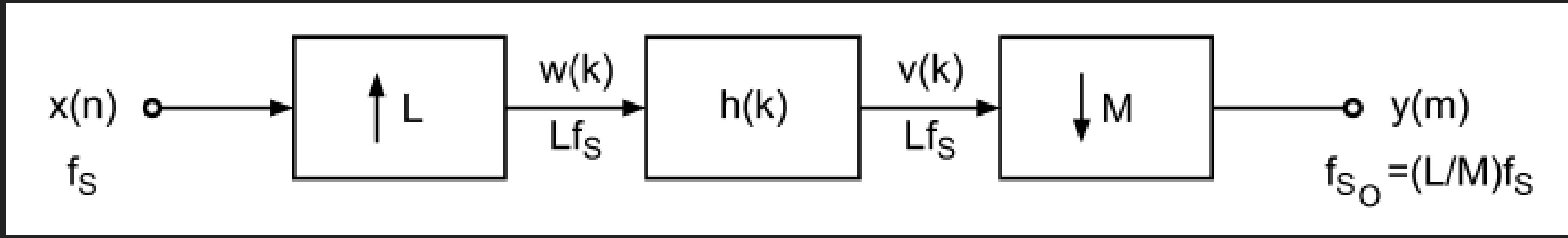



x(m)
h(m)
w(m)
M
y(n)
fS=1/T
Ω=ωT
fS
f'S=1/T'=fS/M
Ω'=ωT'=ωTM=ΩM
X(ejΩ)
1
1/T
|H(ejΩ)|
π/3
2π/3
π
4π/3
2π
Ω
m
W(ejΩ)
y(n)
3
1/MT=1/T'
Y(ejΩ')
4π
6π
Ω'
M=3
n

Resampling by Rational Factor

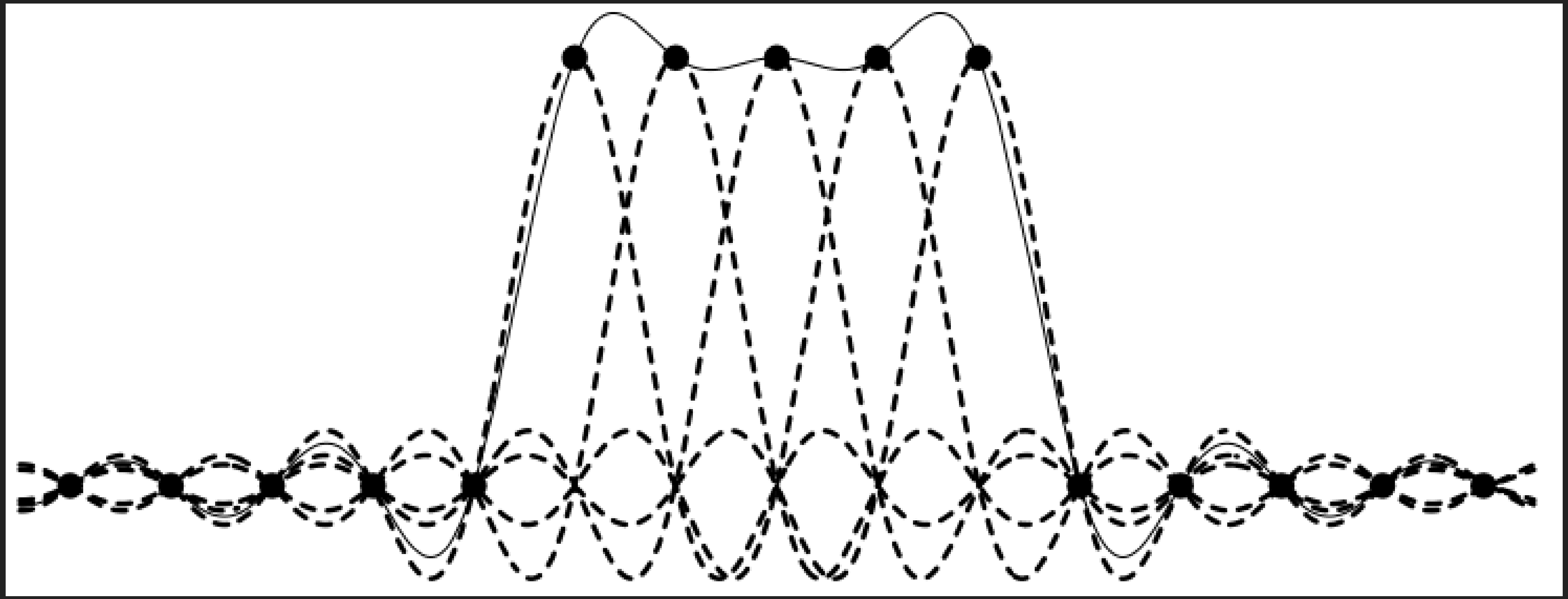
Task: **Convert sample rate** to any other (coupled) sample rate)

1. Convert Sample rate ratio to integer factors
   e.g.: $\frac{48}{44.1} \mapsto L = 160, M = 147$
2. Insert zeros
3. Apply anti-imaging filter
4. Apply anti-aliasing filter
5. Remove Samples



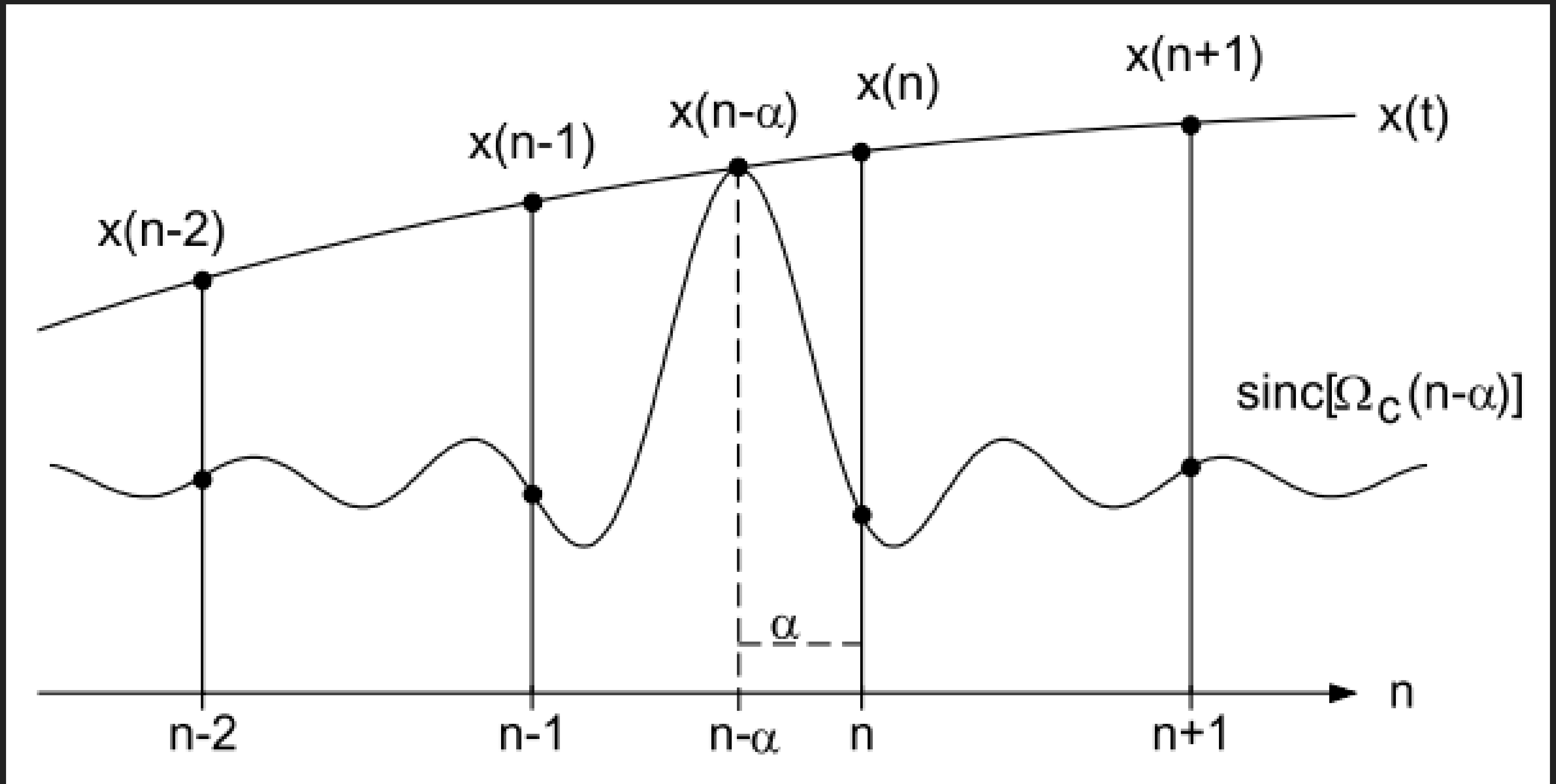
Sinc Interpolation

- Perfect reconstruction of the sample spectrum is possible with ideal filter
- Resampling should be possible by time domain convolution with sinc



$$x(i-\alpha) = \sum_{m=-\infty}^{\infty} x(m) \frac{\Omega_C}{\pi} \frac{sin(\Omega_C(i-\alpha-m))}{\Omega_C(i-\alpha-m)}$$

Ω_C is the cutoff frequency of the ideal lowpass



» Practical implementation: **Windowed Sinc**

Polynomial Interpolation

Interpolation Methods

- Can be interpreted as filters with time-variant filter coefficients
- Not based on traditional filter design methods

Polynomial interpolation

$$f(t) = \sum_{k=0}^{\mathcal{O}} x_k p_k(t)$$

$$p_k(t) = \prod_{j=0}^{\mathcal{O}} \frac{t - t_j}{t_k - t_j}$$

$$x(t) = \frac{1}{t}$$

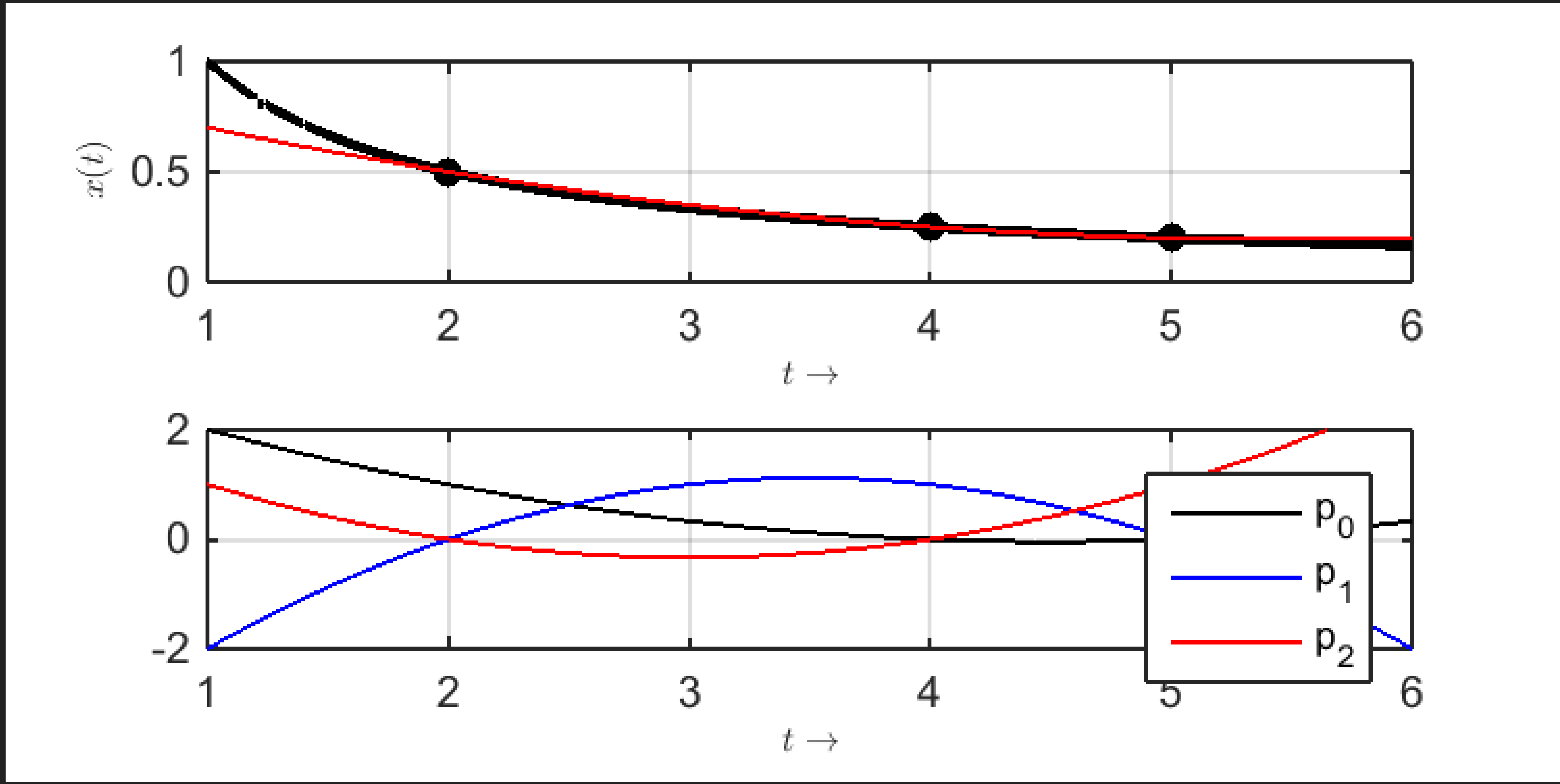
$$\text{nodes: } t = [2, 4, 5]$$

$$p_0(t) = \frac{(t-4)(t-5)}{(2-4)(2-5)} = \frac{(t-4)(t-5)}{6}$$

$$p_1(t) = \frac{(t-2)(t-5)}{(4-2)(4-5)} = -\frac{(t-2)(t-5)}{2}$$

$$p_2(t) = \frac{(t-2)(t-4)}{(5-2)(5-4)} = \frac{(t-2)(t-4)}{3}$$

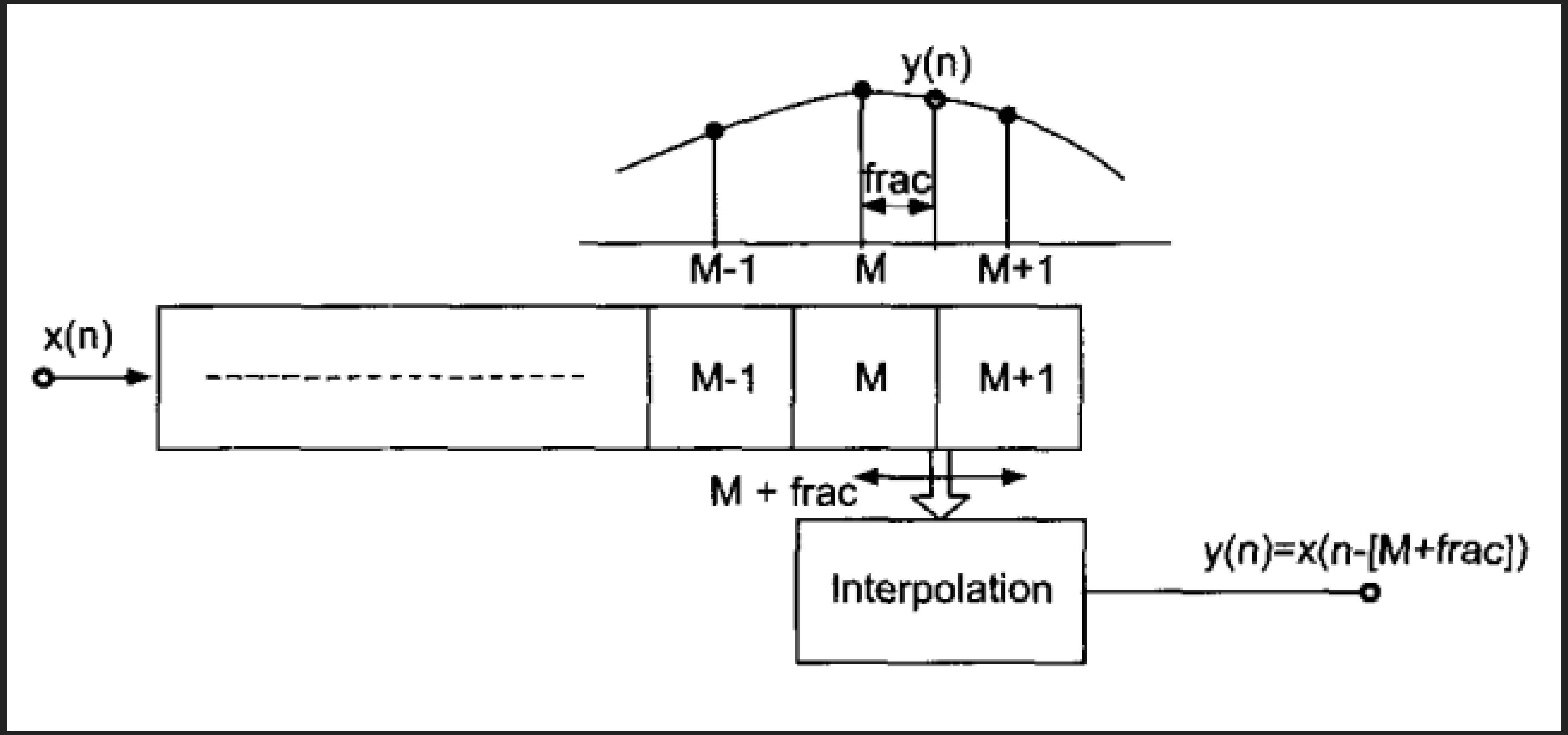
$$f(t) = \sum_{k=0}^{\mathcal{O}} x_k p_k(t)$$



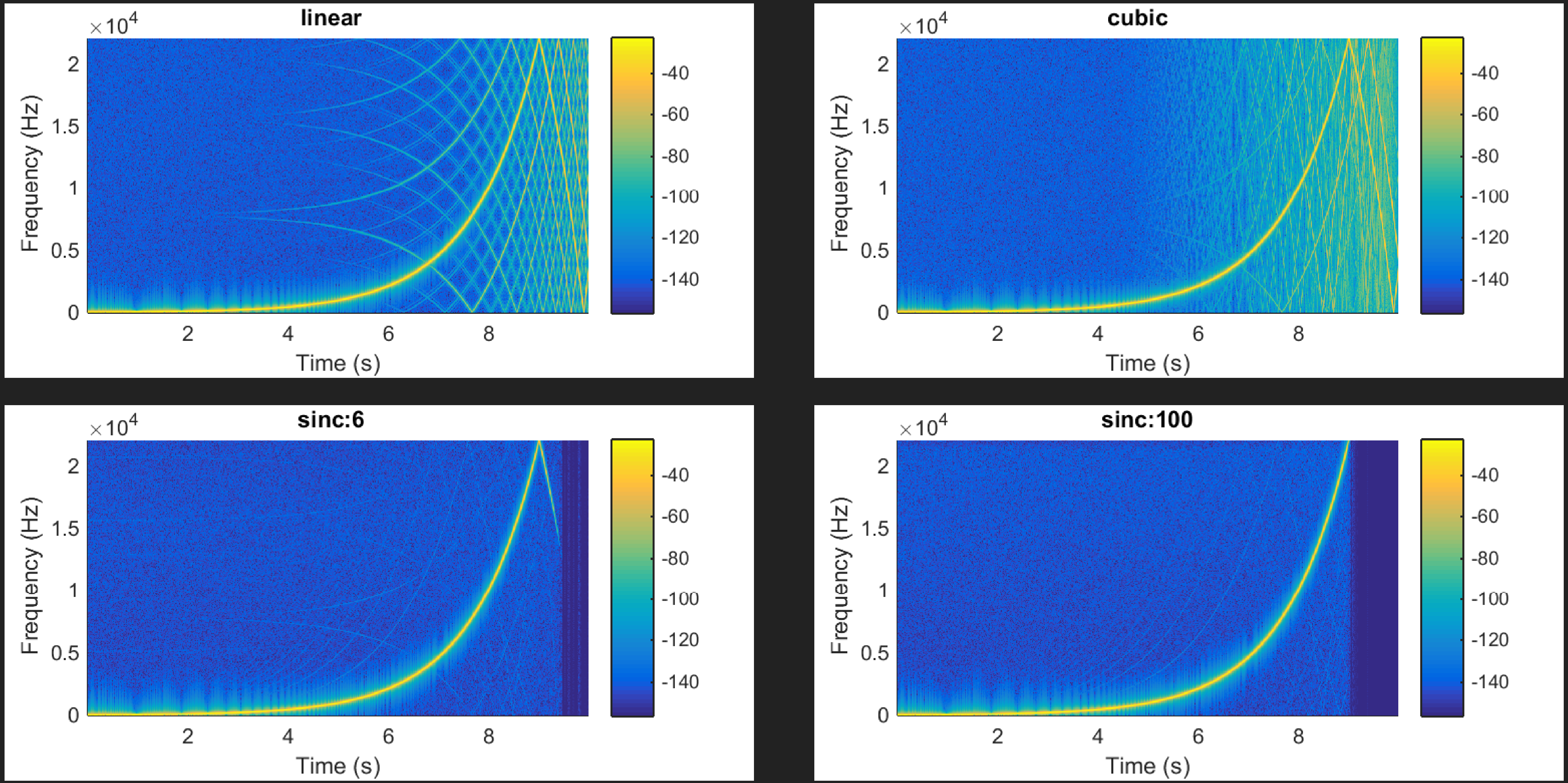
1
0.5
0
$x(t)$
1
2
3
4
5
6
$t \rightarrow$
2
0
-2
1
2
3
4
5
6
$t \rightarrow$
p_0
p_1
p_2

Linear Interpolation (1st order → 2 points)

$$
\begin{aligned}
x(t) &= \frac{1}{t} \\
\text{nodes: } t &= [2, 4] \\
p_0(t) &= \frac{(t-4)}{(2-4)} = \frac{(4-t)}{2} \\
p_1(t) &= \frac{(t-2)}{(4-2)} = \frac{(t-2)}{2} \\
f(t) &= \sum_{k=0}^{\mathcal{O}} x_k p_k(t) \\
\Rightarrow f(t) &= p_0 \frac{1}{2} + p_1 \frac{1}{4} \\
&= -\frac{1}{8} t + \frac{3}{4}.
\end{aligned}
$$



$$\hat{x} = x_l \cdot (1 - frac) + x_r \cdot frac$$


linear
cubic
sinc:6
sinc:100
Frequency (Hz)
×10^4
Time (s)
0
0.5
1
1.5
2
4
6
8
-40
-60
-80
-100
-120
-140

	Orig (48kHz)	ds (6kHz, w/o filt)	ds (6kHz, w/ filt)
Sax			
Big Band			

Summary

- resampling: estimate different sample points of underlying continuous signal
- as with sampling, proper filtering has to take place
- some interpolation approaches have filter “built-in”
- perfect reconstruction impossible (infinite sinc), however, perceptually artifact-free resampling is possible
  - main issue: filter cut-off and steepness vs. aliasing