Digital Signal Processing

Module 6: Interpolation and Sampling

Module Overview:

- Module 6.1: Continuous-time signals
- Module 6.2: Interpolation
- Module 6.3: Sampling
- Module 6.4: Aliasing
- Module 6.5: Interpolation and sampling in practice
- Module 6.6: Discrete-time processing of continuous-time signals

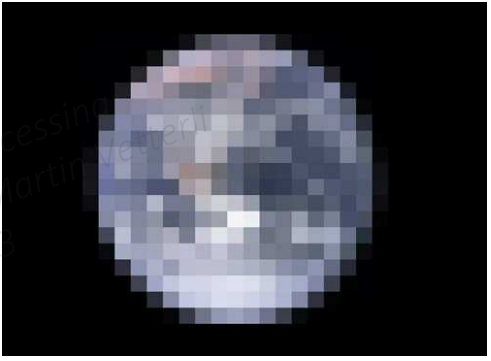
Digital Signal Processing

Module 6.1: The Continuous-Time Paradigm

Overview:

- Models of the world
- Continuous-time signals

Two views of the world



Analog/continuous versus discrete/digital

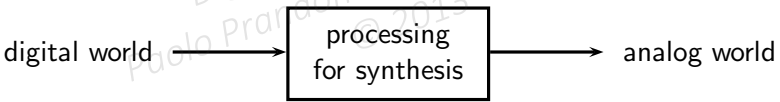
Digital processing of signals from/to the analog world

- input is continuous-time: $x(t)$
- output is continuous-time: $y(t)$
- processing is on sequences: $x[n], y[n]$

examples: MP3, digital photography

Digital processing of signals to the analog world

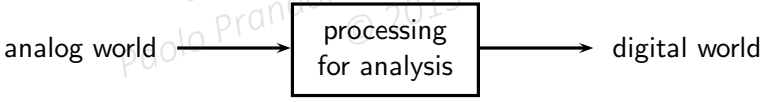
- input is discrete-time: $x[n]$
- output is continuous-time: $y(t)$
- processing is on sequences: $x[n], y[n]$



examples: computer graphics, video games

Digital processing of signals from the analog world

- input is continuous-time: $x(t)$
- output is discrete-time: $y[n]$
- processing is on sequences: $x[n], y[n]$



examples: control systems, monitoring

Two views of the world

digital worldview:

- arithmetic
- combinatorics
- computer science
- DSP

analog worldview:

- calculus
- distributions
- system theory
- electronics

Two views of the world

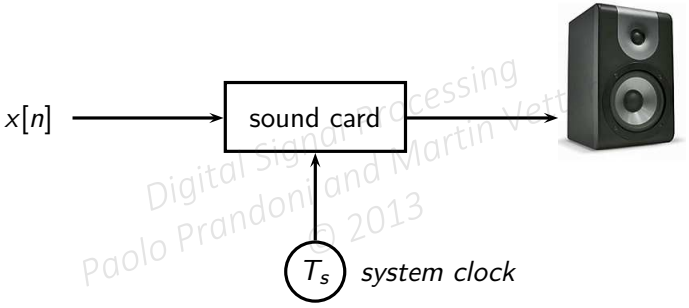
digital worldview:

- countable integer index n
- sequences $x[n] \in \ell_2(\mathbb{Z})$
- frequency $\omega \in [-\pi, \pi]$
- DTFT: $\ell_2(\mathbb{Z}) \mapsto L_2([-\pi, \pi])$

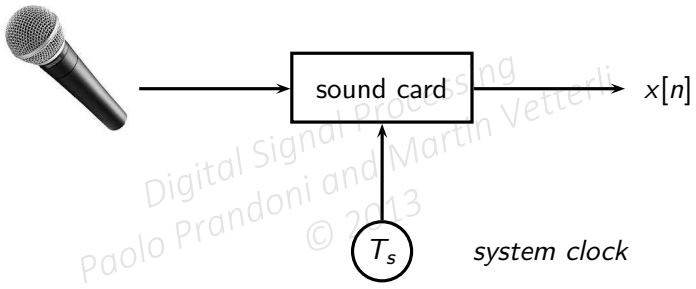
analog worldview:

- real-valued time t (sec)
- functions $x(t) \in L_2(\mathbb{R})$
- frequency $\Omega \in \mathbb{R}$ (rad/sec)
- FT: $L_2(\mathbb{R}) \mapsto L_2(\mathbb{R})$

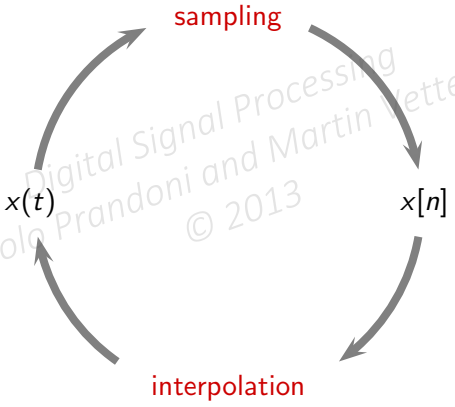
Bridging the gap



Bridging the gap



Bridging the gap



About continuous time

- time: real variable t
- signal $x(t)$: complex functions of a real variable
- finite energy: $x(t) \in L_2(\mathbb{R})$
- inner product in $L_2(\mathbb{R})$

$$\langle x(t), y(t) \rangle = \int_{-\infty}^{\infty} x^*(t) y(t) dt$$

- energy: $||x(t)||^2 = \langle x(t), x(t) \rangle$

Analog LTI filters

$$\begin{aligned} y(t) &= (x * h)(t) \\ &= \langle h^*(t-\tau), x(\tau) \rangle \\ &= \int_{-\infty}^{\infty} x(\tau) h(t-\tau) d\tau \end{aligned}$$

Fourier analysis

- in discrete time max angular frequency is $\pm\pi$
- in continuous time no max frequency: $\Omega \in \mathbb{R}$
- concept is the same:

$$X(j\Omega) = \int_{-\infty}^{\infty} x(t) e^{-j\Omega t} dt \qquad \leftarrow \textit{not periodic!}$$

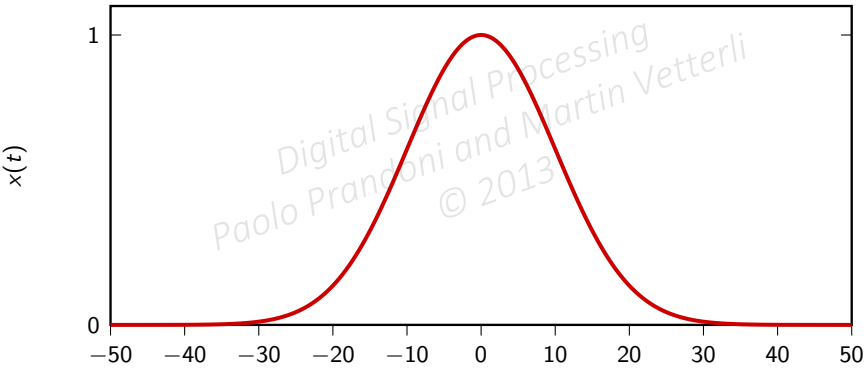
$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega) e^{j\Omega t} d\Omega$$

Real-world frequency

- Ω expressed in rad/s
- $F = \dfrac{\Omega}{2\pi}$, expressed in Hertz $(1/s)$
- period $T = \dfrac{1}{F} = \dfrac{2\pi}{\Omega}$

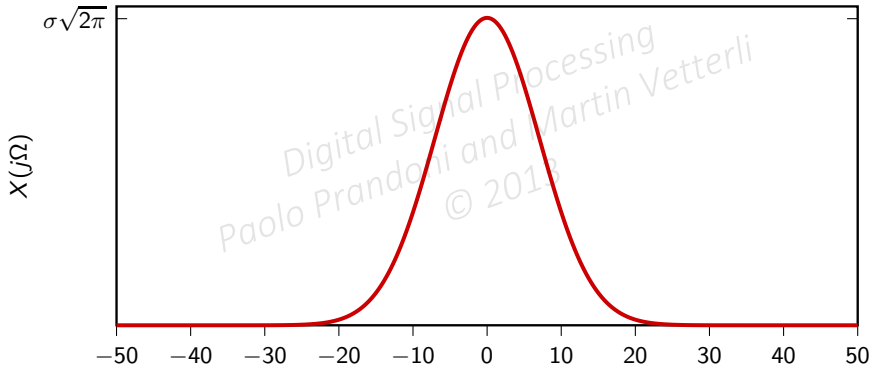
Example

$$x(t) = e^{-\frac{t^2}{2\sigma^2}}$$



Example

$$X(j\Omega) = \sigma\sqrt{2\pi}e^{-\frac{\sigma^2}{2}\Omega^2}$$



Convolution theorem

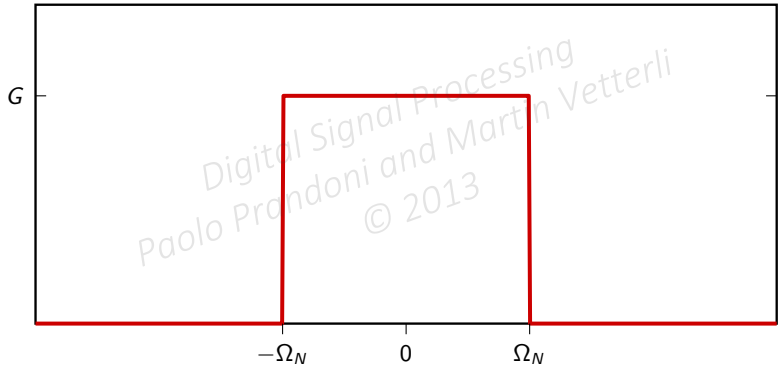
$$Y(j\Omega) = X(j\Omega)\, H(j\Omega)$$

A new concept: bandlimited functions

Ω_N-bandlimitedness:

$$X(j\Omega) = 0 \quad \text{for } |\Omega| > \Omega_N$$

Prototypical bandlimited function



The prototypical bandlimited function

$$\Phi(j\Omega) = G \text{ rect}\left(\frac{\Omega}{2\Omega_N}\right)$$

$$\begin{aligned}\varphi(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \Phi(j\Omega)e^{j\Omega t}d\Omega \\ &= \ldots \quad \text{see Module 5.5} \\ &= G\,\frac{\Omega_N}{\pi}\,\text{sinc}\left(\frac{\Omega_N}{\pi}t\right)\end{aligned}$$

The prototypical bandlimited function

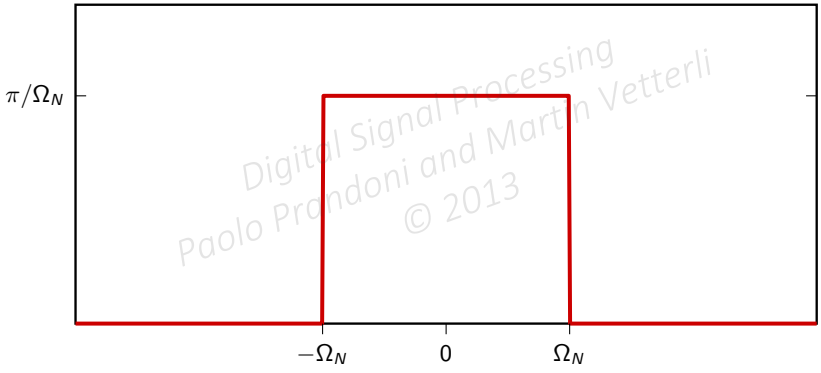
- normalization: $G = \dfrac{\pi}{\Omega_N}$
- total bandwidth: $\Omega_B = 2\Omega_N$
- define $T_s = \dfrac{2\pi}{\Omega_B} = \dfrac{\pi}{\Omega_N}$

The prototypical bandlimited function

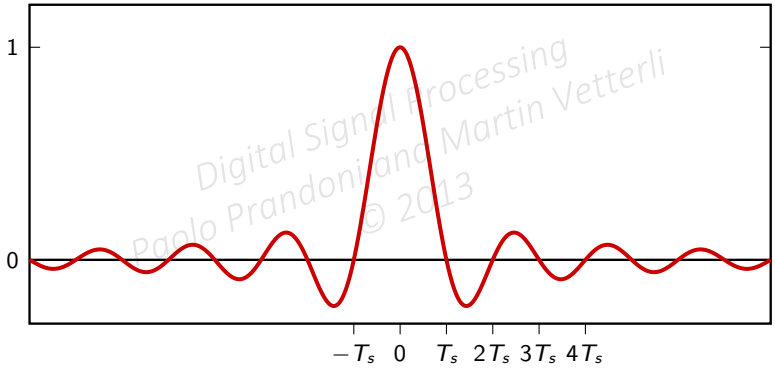
$$\Phi(j\Omega) = \frac{\pi}{\Omega_N}\text{rect}\left(\frac{\Omega}{2\Omega_N}\right)$$

$$\varphi(t) = \text{sinc}\left(\frac{t}{T_s}\right)$$

The prototypical bandlimited function



The prototypical bandlimited function



END OF MODULE 6.1

Digital Signal Processing

Module 6.2: Interpolation

Overview:

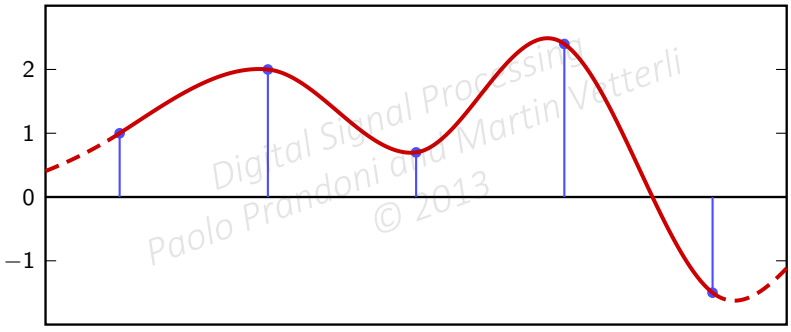
- Polynomial interpolation
- Local interpolation
- Sinc interpolation

Interpolation

$$x[n] \longrightarrow x(t)$$

"fill the gaps" between samples

Example



Interpolation requirements

- decide on T_s
- make sure $x(nT_s) = x[n]$
- make sure $x(t)$ is *smooth*

Why smoothness?

- jumps (1st order discontinuities) would require the signal to move "faster than light"...
- 2nd order discontinuities would require infinite acceleration
- ...
- the interpolation should be infinitely differentiable
- "natural" solution: polynomial interpolation

Polynomial interpolation

- N points $\rightarrow$ polynomial of degree $(N-1)$
- $p(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_{N-1} t^{(N-1)}$
- "naive" approach:

$$\begin{cases} p(0) = x[0] \\ p(T_s) = x[1] \\ p(2T_s) = x[2] \\ \ldots \\ p((N-1)T_s) = x[N-1] \end{cases}$$

Polynomial interpolation

Without loss of generality:

- consider a symmetric interval $I_N = [-N, \ldots, N]$
- set $T_s = 1$

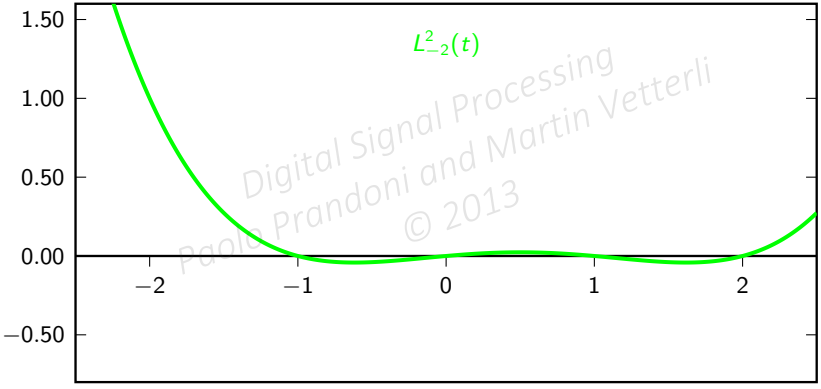
$$\begin{cases} p(-N) = x[-N] \\ p(-N+1) = x[-N+1] \\ \qquad \ldots \\ p(0) = x[0] \\ \qquad \ldots \\ p(N) = x[N] \end{cases}$$

Lagrange interpolation

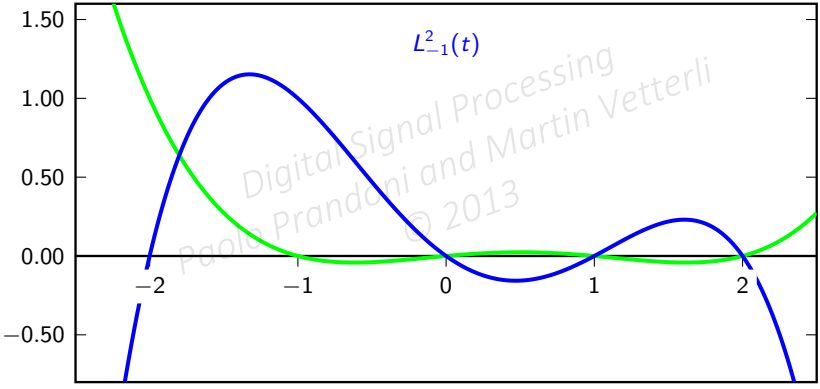
- P_N: space of degree-$2N$ polynomials over I_N
- a basis for P_N is the family of $2N+1$ Lagrange polynomials

$$L_n^{(N)}(t) = \prod_{\substack{k=-N \\ k \neq n}}^{N} \frac{t-k}{n-k} \qquad n = -N, \ldots, N$$

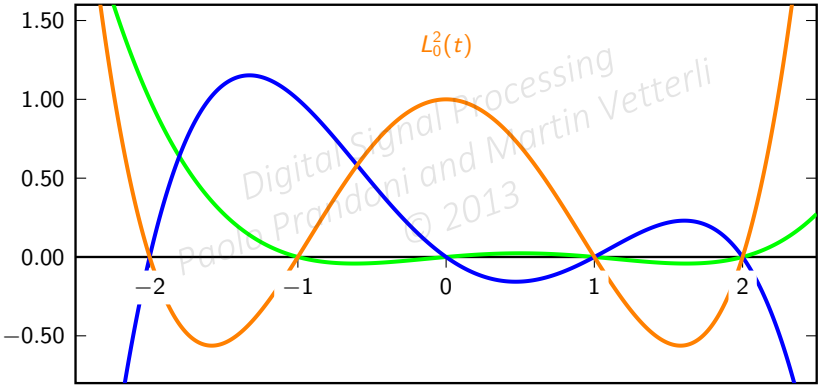
Lagrange interpolation polynomials



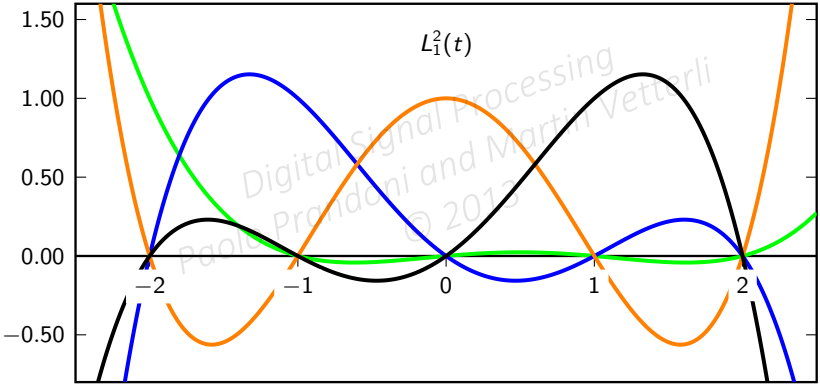
Lagrange interpolation polynomials



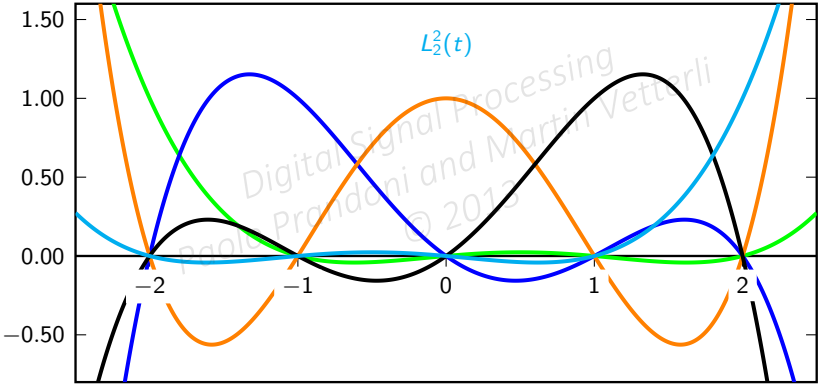
Lagrange interpolation polynomials



Lagrange interpolation polynomials



Lagrange interpolation polynomials



Lagrange interpolation

$$p(t) = \sum_{n=-N}^{N} x[n] L_n^{(N)}(t)$$

Lagrange interpolation

The Lagrange interpolation *is* the sought-after polynomial interpolation:

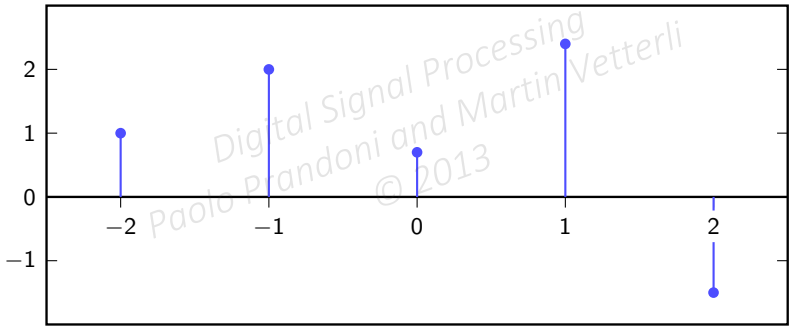
- polynomial of degree $2N$ through $2N+1$ points is unique
- the Lagrangian interpolator satisfies

$$p(n) = x[n] \qquad \text{for } -N \leq n \leq N$$

since

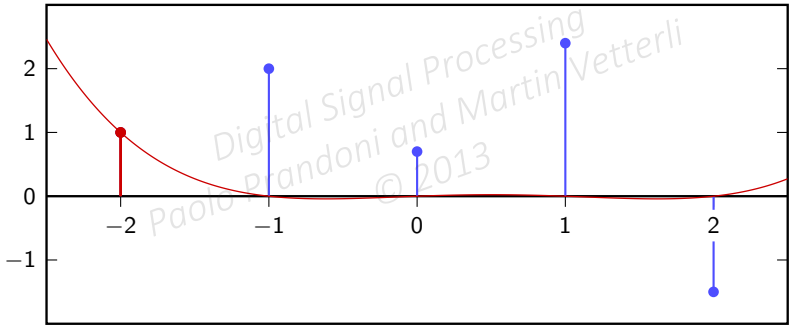
$$L_n^{(N)}(m) = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases} \qquad -N \leq n, m \leq N$$

Lagrange interpolation



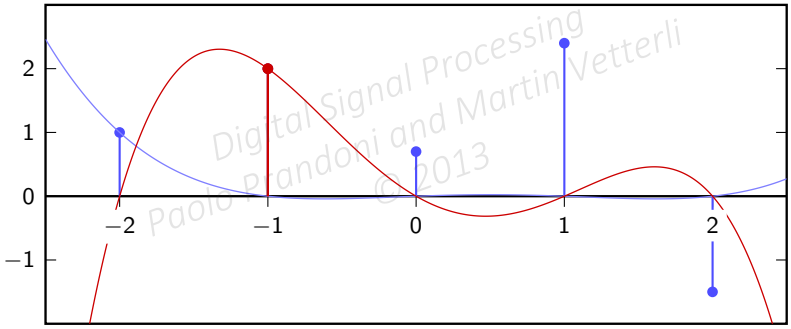
Lagrange interpolation

$x[-2]L_{-2}^{(2)}(t)$

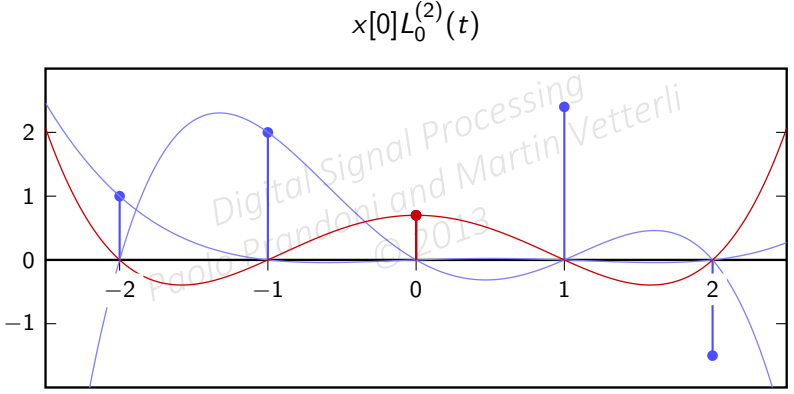


Lagrange interpolation

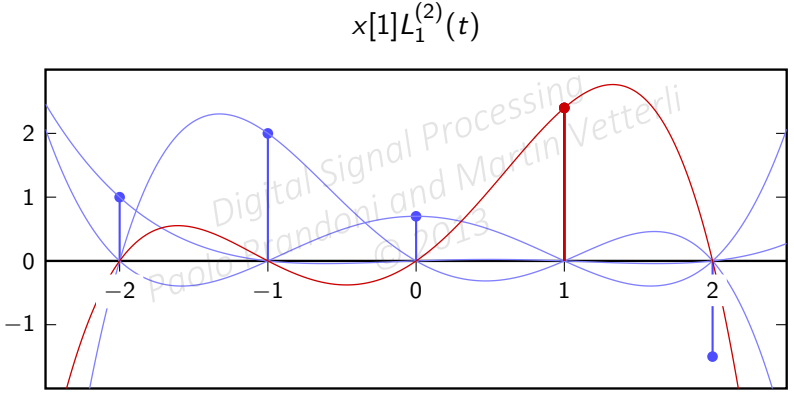
$x[-1]L_{-1}^{(2)}(t)$



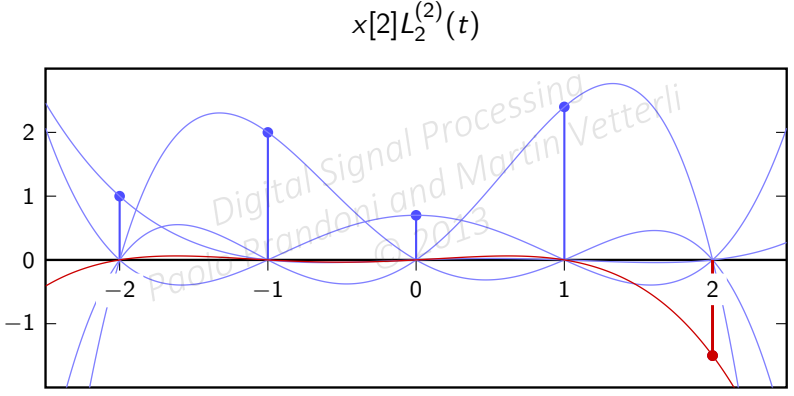
Lagrange interpolation



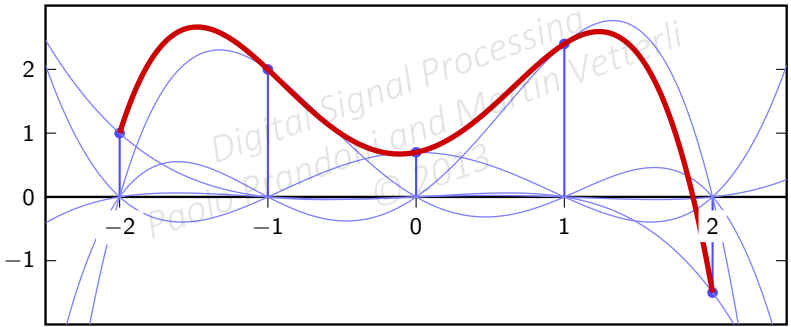
Lagrange interpolation



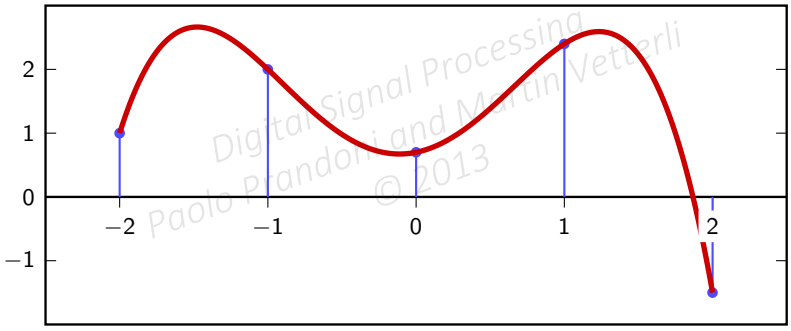
Lagrange interpolation



Lagrange interpolation



Lagrange interpolation



Polynomial interpolation

key property:

- maximally smooth (infinitely many continuous derivatives)

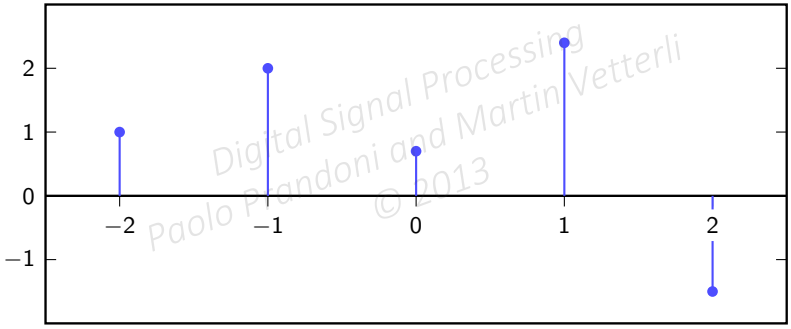
drawback:

- interpolation "bricks" depend on N

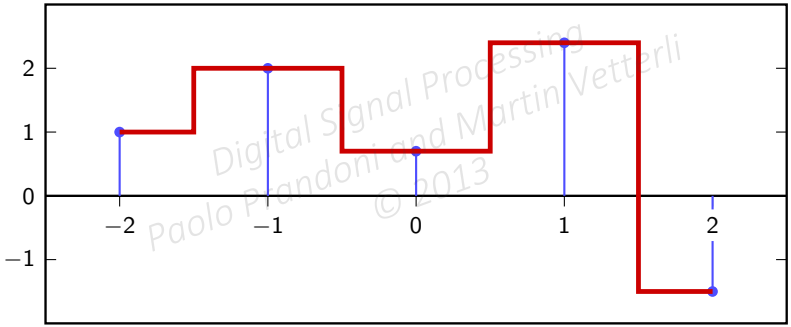
Relaxing the interpolation requirements

- decide on T_s
- make sure $x(nT_s) = x[n]$
- make sure $x(t)$ is *smooth*

Zero-order interpolation



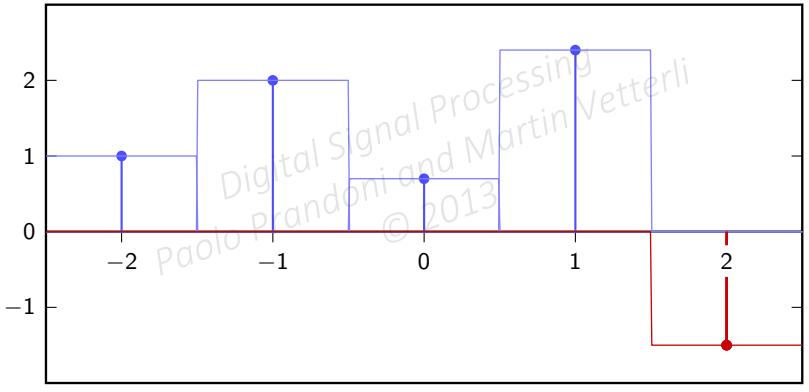
Zero-order interpolation



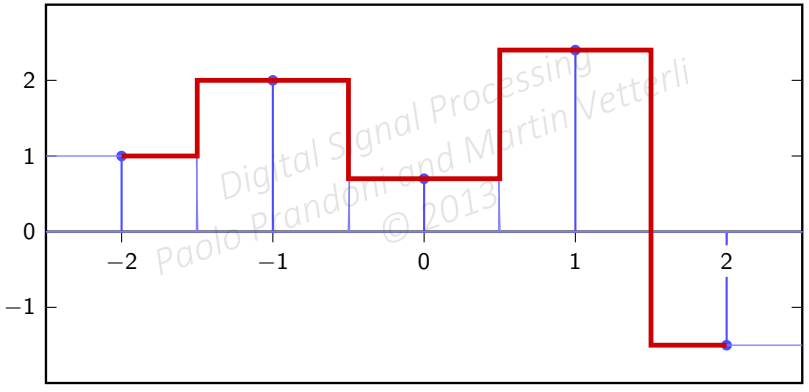
Zero-order interpolation

- $x(t) = x[\,\lfloor t + 0.5 \rfloor\,], \qquad -N \leq t \leq N$
- $x(t) = \sum_{n=-N}^{N} x[n]\,\mathrm{rect}(t-n)$
- interpolation kernel: $i_0(t) = \mathrm{rect}(t)$
- $i_0(t)$: "zero-order hold"
- interpolator's support is 1
- interpolation is not even continuous

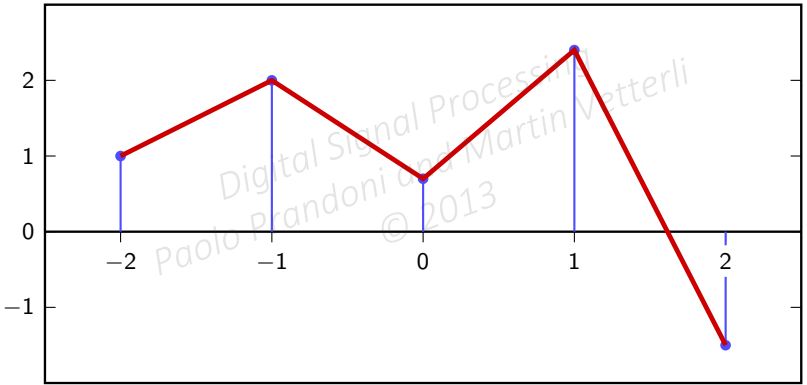
Zero-order interpolation



Zero-order interpolation



First-order interpolation



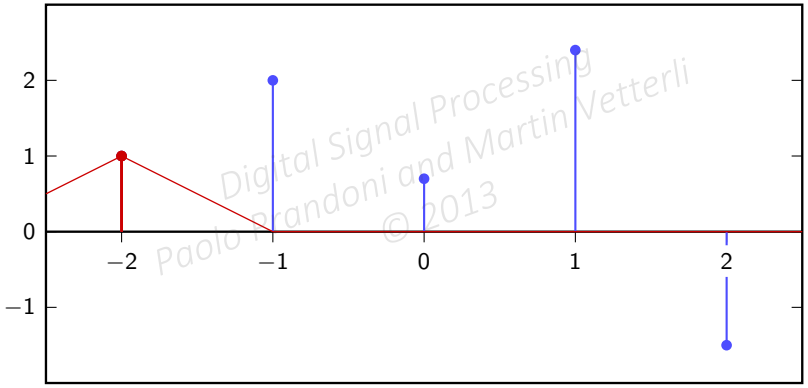
First-order interpolation

- "connect the dots" strategy
- $x(t) = \sum_{n=-N}^{N} x[n]\, i_1(t-n)$
- interpolation kernel:

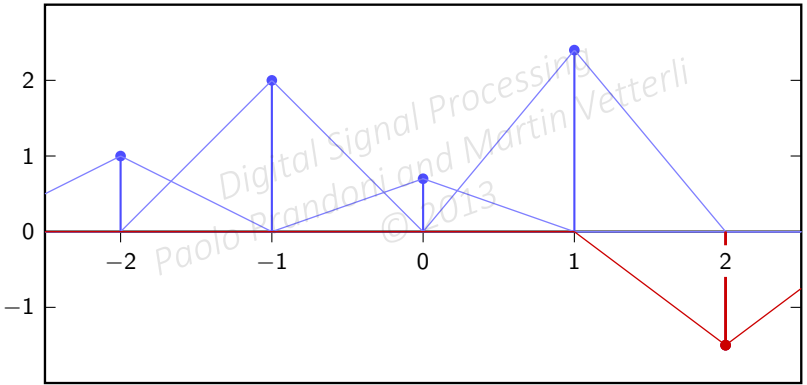
$$i_1(t) = \begin{cases} 1-|t| & |t| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

- interpolator's support is 2
- interpolation is continuous but derivative is not

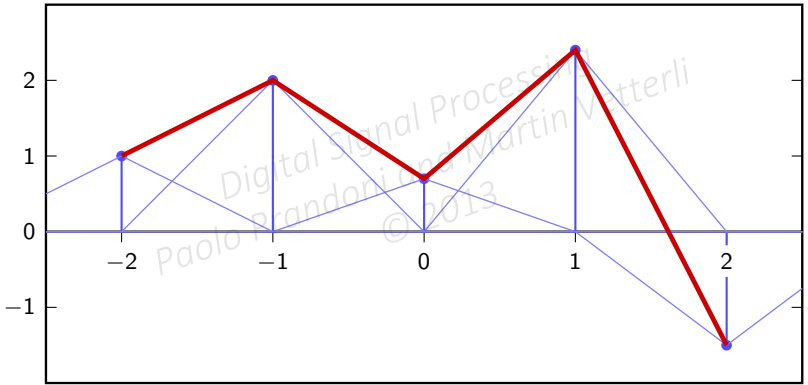
First-order interpolation



First-order interpolation



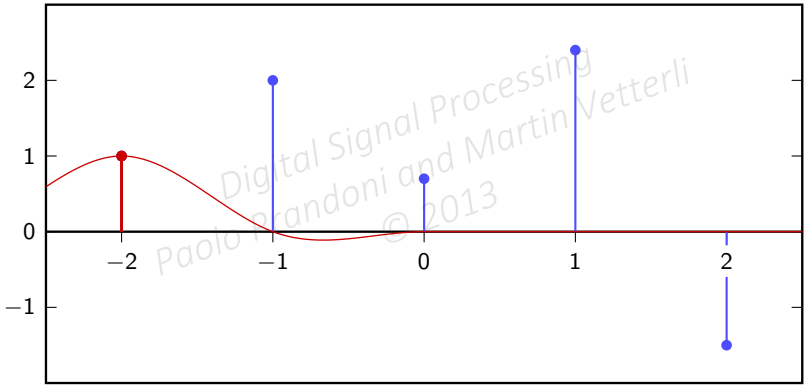
First-order interpolation



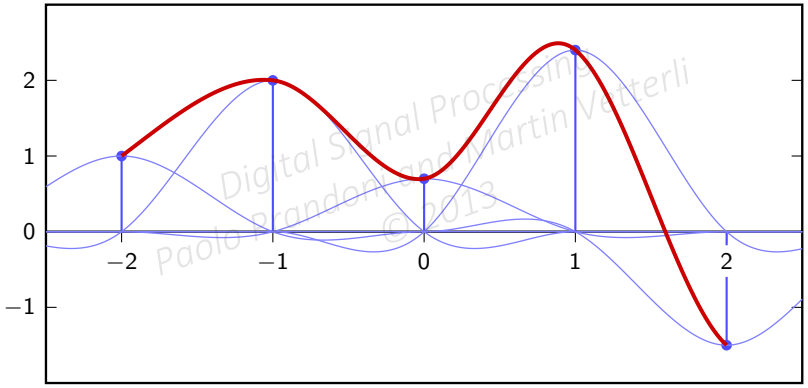
Third-order interpolation

- $x(t) = \sum_{n=-N}^{N} x[n]\, i_3(t-n)$
- interpolation kernel obtained by splicing two cubic polynomials
- interpolator's support is 4
- interpolation is continuous up to second derivative

Third-order interpolation



Third-order interpolation



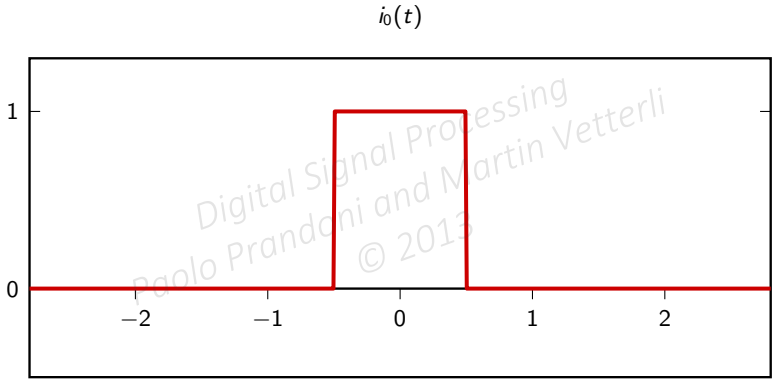
Local interpolation schemes

$$x(t) = \sum_{n=-N}^{N} x[n]\, i_c(t-n)$$

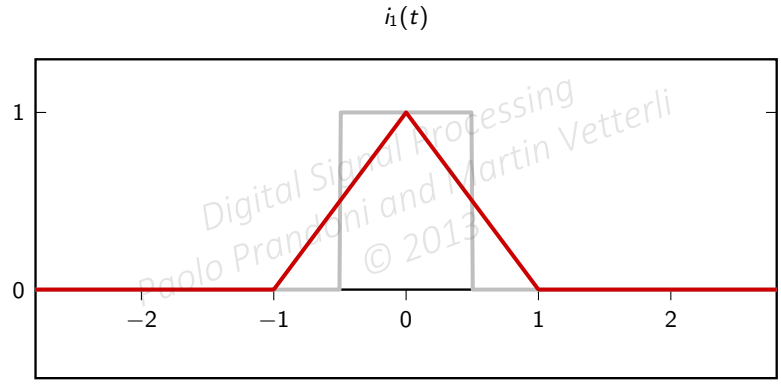
Interpolator's requirements:

- $i_c(0) = 1$
- $i_c(t) = 0$ for t a nonzero integer.

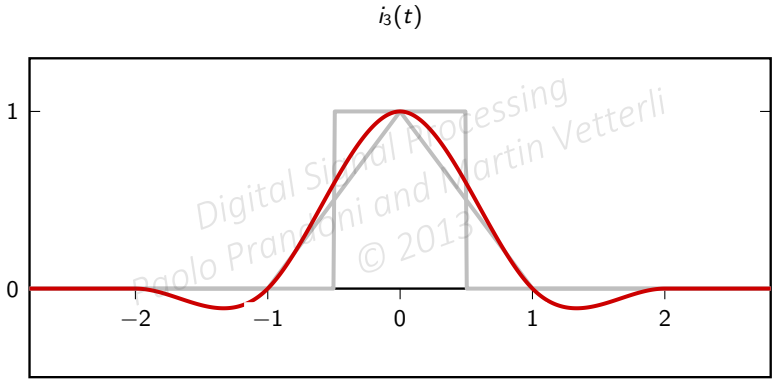
Local interpolators



Local interpolators



Local interpolators



Local interpolation

key property:

- same interpolating function independently of N

drawback:

- lack of smoothness

A remarkable result:

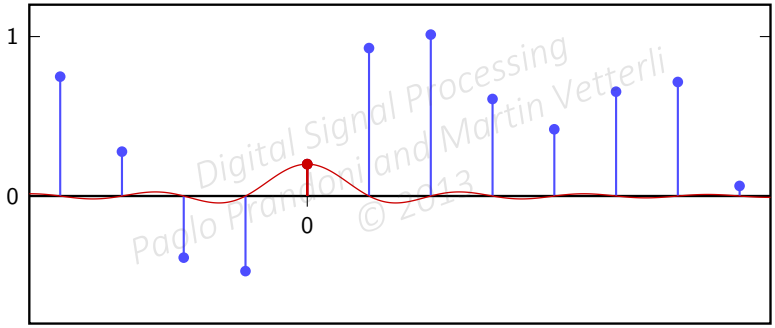
$$\lim_{N \to \infty} L_n^{(N)}(t) = \operatorname{sinc}(t - n)$$

in the limit, local and global interpolation are the same!

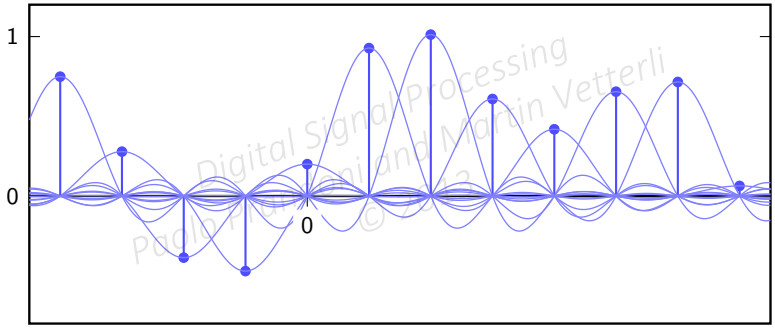
Sinc interpolation formula

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

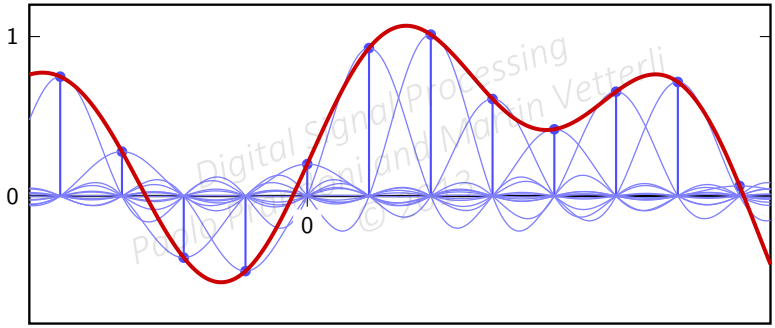
Sinc interpolation



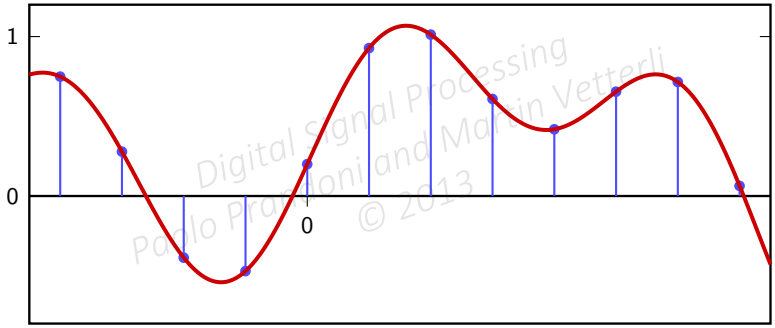
Sinc interpolation



Sinc interpolation



Sinc interpolation



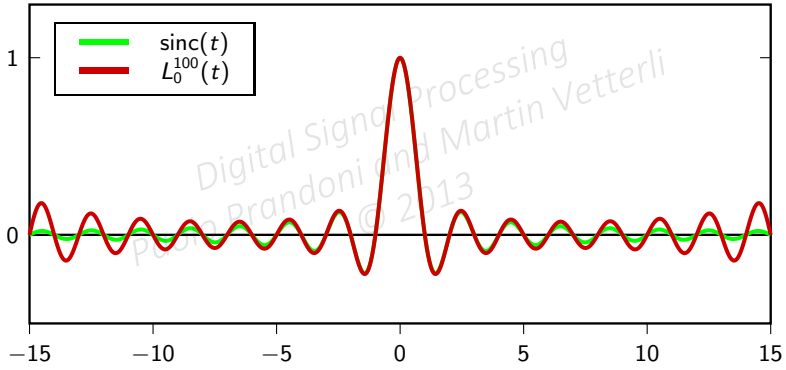
"Proof" that $L_n^{(N)}(t) \to \mathrm{sinc}\,(t-n)$

- real proof is rather technical (see the book)
- intuition: $\mathrm{sinc}(t-n)$ and $L_n^{(\infty)}(t)$ share an infinite number of zeros:

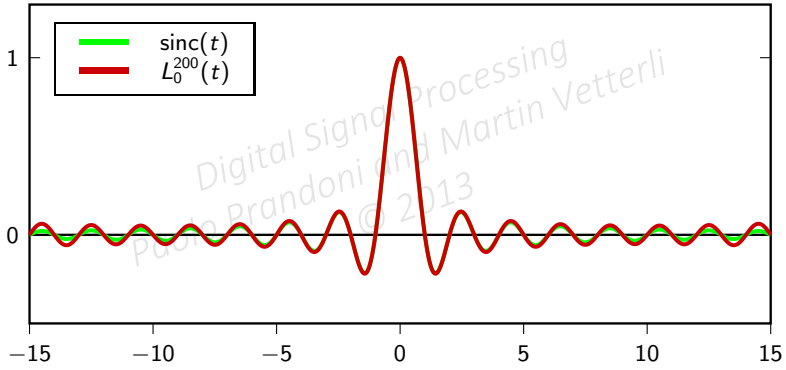
$$\mathrm{sinc}(m-n) = \delta[m-n] \qquad m, n \in \mathbb{Z}$$

$$L_n^{(N)}(m) = \delta[m-n] \qquad m, n \in \mathbb{Z}, \quad -N \leq n, m \leq N$$

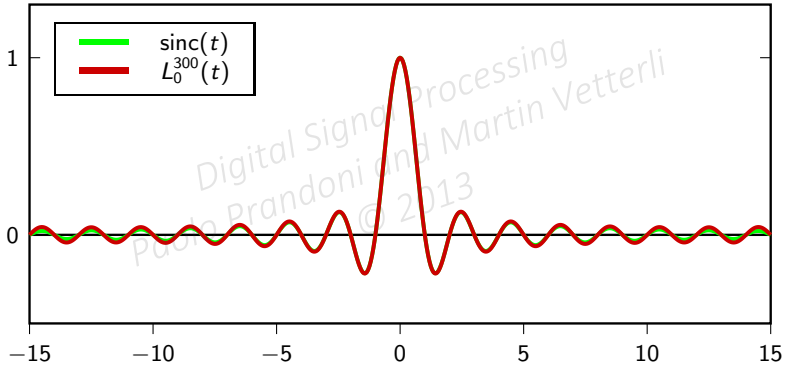
Graphically



Graphically



Graphically



END OF MODULE 6.2

Digital Signal Processing

Module 6.3: The space of bandlimited signals

Overview:

- Spectrum of interpolated signals
- Space of bandlimited functions
- Sinc sampling
- The sampling theorem

Sinc interpolation

the ingredients:

- discrete-time signal $x[n]$, $n \in \mathbb{Z}$ (with DTFT $X(e^{j\omega})$)
- interpolation interval T_s
- the sinc function

the result:

- a smooth, continuous-time signal $x(t)$, $t \in \mathbb{R}$

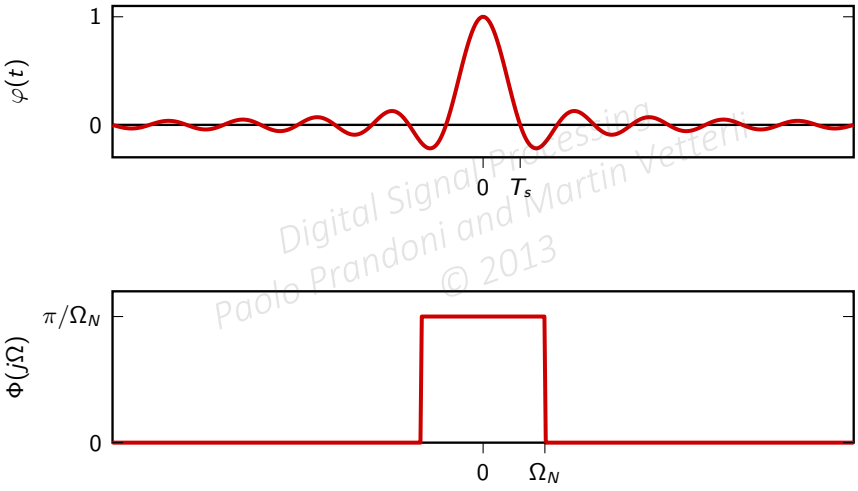
what does the spectrum of $x(t)$ look like?

Key facts about the sinc

$$\varphi(t) = \text{sinc}\left(\frac{t}{T_s}\right) \quad \longleftrightarrow \quad \Phi(j\Omega) = \frac{\pi}{\Omega_N}\,\text{rect}\left(\frac{\Omega}{2\Omega_N}\right)$$

$$T_s = \frac{\pi}{\Omega_N} \qquad \Omega_N = \frac{\pi}{T_s}$$

Key facts about the sinc



Sinc interpolation

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

Spectral representation (I)

$$
\begin{aligned}
X(j\Omega) &= \int_{-\infty}^{\infty} x(t)\, e^{-j\Omega t} dt \\
&= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n] \,\mathrm{sinc}\left(\frac{t - nT_s}{T_s}\right) e^{-j\Omega t} dt \\
&= \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} \mathrm{sinc}\left(\frac{t - nT_s}{T_s}\right) e^{-j\Omega t} dt \\
&= \sum_{n=-\infty}^{\infty} x[n] \left(\frac{\pi}{\Omega_N}\right) \mathrm{rect}\left(\frac{\Omega}{2\Omega_N}\right) e^{-jnT_s\Omega}
\end{aligned}
$$

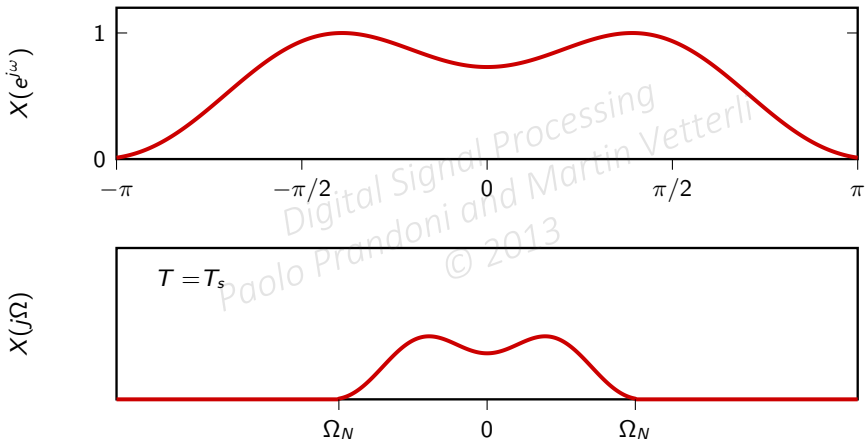
Spectral representation (II)

$$X(j\Omega) = \sum_{n=-\infty}^{\infty} x[n] \left(\frac{\pi}{\Omega_N}\right) \text{rect}\left(\frac{\Omega}{2\Omega_N}\right) e^{-jnT_s\Omega}$$

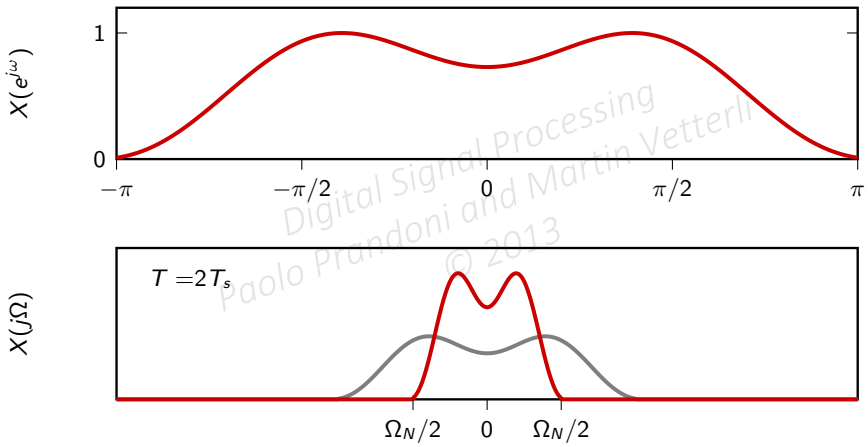
$$= \left(\frac{\pi}{\Omega_N}\right) \text{rect}\left(\frac{\Omega}{2\Omega_N}\right) \sum_{n=-\infty}^{\infty} x[n] e^{-j(\pi/\Omega_N)\Omega n}$$

$$= \begin{cases} (\pi/\Omega_N) X(e^{j\pi(\Omega/\Omega_N)}) & \text{for } |\Omega| \leq \Omega_N \\ 0 & \text{otherwise} \end{cases}$$

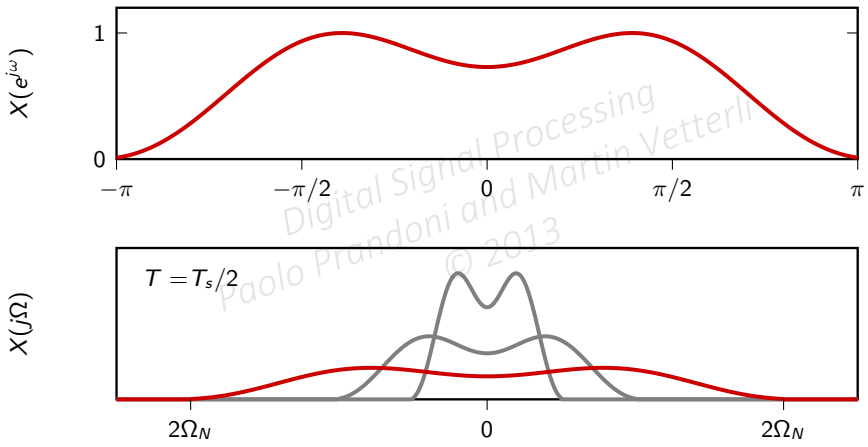
Spectrum of interpolated signals



Spectrum of interpolated signals



Spectrum of interpolated signals

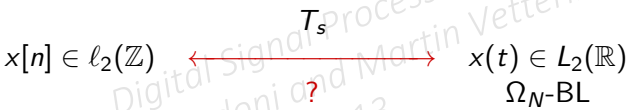


Spectrum of interpolated signals

pick interpolation period T_s:

- $X(j\Omega)$ is Ω_N-bandlimited, with $\Omega_N = \pi/T_s$
- fast interpolation (T_s small) $\rightarrow$ wider spectrum
- slow interpolation (T_s large) $\rightarrow$ narrower spectrum
- (for those who remember...) it's like changing the speed of a record player

Space of bandlimited functions



Let's lighten the notation

for a while we will proceed with

- $T_s = 1$
- $\Omega_N = \pi$

(derivations in the general case are in the book)

The road to the sampling theorem

claims:

- the space of π-bandlimited functions is a Hilbert space
- the functions $\varphi^{(n)}(t) = \mathrm{sinc}(t - n)$, with $n \in \mathbb{Z}$, form a basis for the space
- if $x(t)$ is π-BL, the sequence $x[n] = x(n)$, with $n \in \mathbb{Z}$, is a sufficient representation (i.e. we can reconstruct $x(t)$ from $x[n]$)

The space π-BL

- clearly a vector space because π-BL $\subset L_2(\mathbb{R})$ (and linear combinations of π-BL functions are π-BL functions)
- inner product is standard inner product in $L_2(\mathbb{R})$
- completeness... that's more delicate

The space of π-BL functions

recap:

- inner product:

$$\langle x(t), y(t) \rangle = \int_{-\infty}^{\infty} x^*(t) y(t) dt$$

- convolution:

$$(x * y)(t) = \langle x^*(\tau), y(t - \tau) \rangle$$

A basis for the π-BL space

$$\varphi^{(n)}(t) = \mathrm{sinc}(t-n), \qquad n \in \mathbb{Z}$$

$$\begin{aligned}
\langle \varphi^{(n)}(t), \varphi^{(m)}(t) \rangle &= \langle \varphi^{(0)}(t-n), \varphi^{(0)}(t-m) \rangle \\
&= \langle \varphi^{(0)}(t-n), \varphi^{(0)}(m-t) \rangle \\
&= \int_{-\infty}^{\infty} \mathrm{sinc}(t-n)\,\mathrm{sinc}(m-t)\,dt \\
&= \int_{-\infty}^{\infty} \mathrm{sinc}(\tau)\,\mathrm{sinc}((m-n)-\tau)\,d\tau \\
&= (\mathrm{sinc} * \mathrm{sinc})(m-n)
\end{aligned}$$

A basis for the π-BL space

now use the convolution theorem knowing that:

$$\text{FT}\{\text{sinc}(t)\} = \text{rect}\left(\frac{\Omega}{2\pi}\right)$$

$$(\text{sinc} * \text{sinc})(m-n) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\left[\text{rect}\left(\frac{\Omega}{2\pi}\right)\right]^2 e^{j\Omega(m-n)}d\Omega$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\Omega(m-n)}d\Omega$$

$$= \begin{cases} 1 & \text{for } m = n \\ 0 & \text{otherwise} \end{cases}$$

Sampling as a basis expansion

for any $x(t) \in \pi$-BL:

$$\begin{aligned}
\langle \varphi^{(n)}(t), x(t) \rangle &= \langle \mathrm{sinc}(t-n), x(t) \rangle = \langle \mathrm{sinc}(n-t), x(t) \rangle \\
&= (\mathrm{sinc} * x)(n) \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathrm{rect}\left(\frac{\Omega}{2\pi}\right) X(j\Omega) e^{j\Omega n} d\Omega \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega) e^{j\Omega n} d\Omega \\
&= x(n)
\end{aligned}$$

Sampling as a basis expansion, π-BL

Analysis formula:

$$x[n] = \langle \mathrm{sinc}(t-n), x(t) \rangle$$

Synthesis formula:

$$x(t) = \sum_{n=-\infty}^{\infty} x[n]\ \mathrm{sinc}(t-n)$$

Sampling as a basis expansion, Ω_N-BL

Analysis formula:

$$x[n] = \langle \text{sinc}\left(\frac{t - nT_s}{T_s}\right), x(t)\rangle = T_s\, x(nT_s)$$

Synthesis formula:

$$x(t) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} x[n]\ \text{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

The sampling theorem

- the space of Ω_N-bandlimited functions is a Hilbert space
- set $T_s = \pi/\Omega_N$
- the functions $\varphi^{(n)}(t) = \text{sinc}((t - nT_s)/T_s)$ form a basis for the space
- for any $x(t) \in \Omega_N$-BL the coefficients in the sinc basis are the (scaled) samples $T_s\, x(nT_s)$

for any $x(t) \in \Omega_N$-BL, a sufficient representation is the sequence $x[n] = x(nT_s)$

The sampling theorem, corollary

- Ω_N-BL $\subseteq$ Ω-BL for any $\Omega \geq \Omega_N$

for any $x(t) \in \Omega_N$-BL, a sufficient representation is the sequence
$x[n] = x(nT_s)$ for any $T_s \leq \pi/\Omega_N$

The sampling theorem, in hertz

any signal $x(t)$ bandlimited to F_N Hz can be sampled with no loss of information using a sampling frequency $F_s \geq 2F_N$ (i.e. a sampling period $T_s \leq 1/2F_N$)

END OF MODULE 6.3

Digital Signal Processing

Module 6.4: Sampling and Aliasing - Introduction

Overview:

- "Raw" sampling
- Sinusoidal aliasing

Sinc Sampling

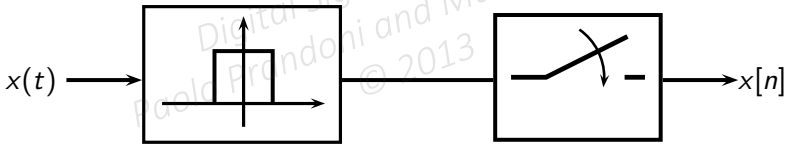
$$x[n] = \langle \text{sinc}\left(\frac{t - nT_s}{T_s}\right), x(t) \rangle$$

Sinc Sampling

$$x[n] = (\text{sinc}_{T_s} * x)(nT_s)$$

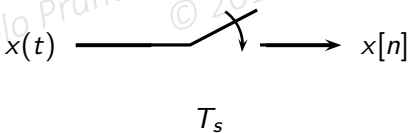
Sinc Sampling

$$x[n] = (\text{sinc}_{T_s} * x)(nT_s)$$



"Raw" Sampling

$$x[n] = x(nT_s)$$

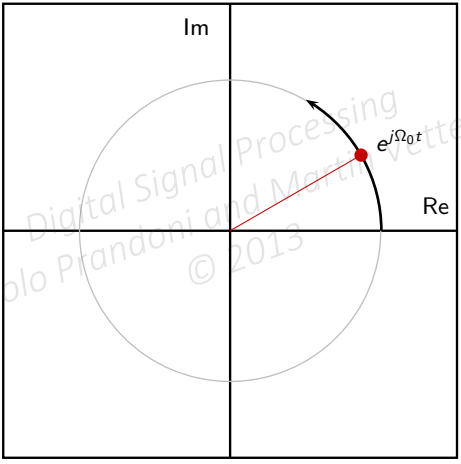


The continuous-time complex exponential

$$x(t) = e^{j\Omega_0 t}$$

- always periodic, period $T = 2\pi/\Omega_0$
- all angular speeds are allowed
- FT $\left\{e^{j\Omega_0 t}\right\} = 2\pi\delta(\Omega - \Omega_0)$
- bandlimited to Ω_0

The continuous-time complex exponential

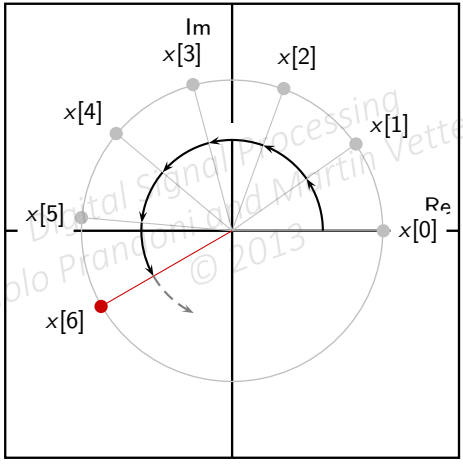


Raw samples of the continuous-time complex exponential

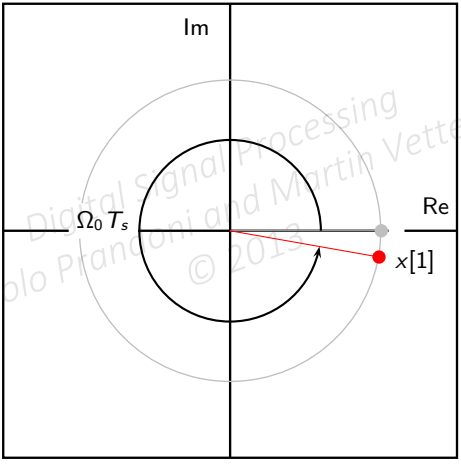
$$x[n] = e^{j\Omega_0 T_s n}$$

- raw samples are snapshots at regular intervals of the rotating point
- resulting digital frequency is $\omega_0 = \Omega_0 T_s$

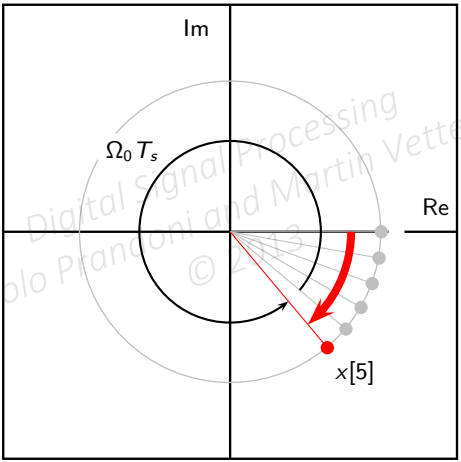
When $T_s < \pi/\Omega_0,\ \omega_0 < \pi \ldots$



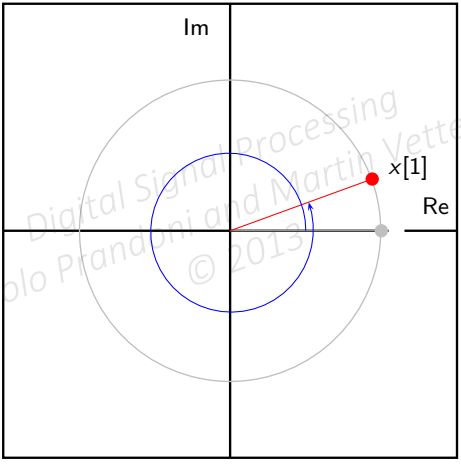
When $\pi/\Omega_0 < T_s < 2\pi/\Omega_0,\ \pi < \omega_0 < 2\pi \ldots$



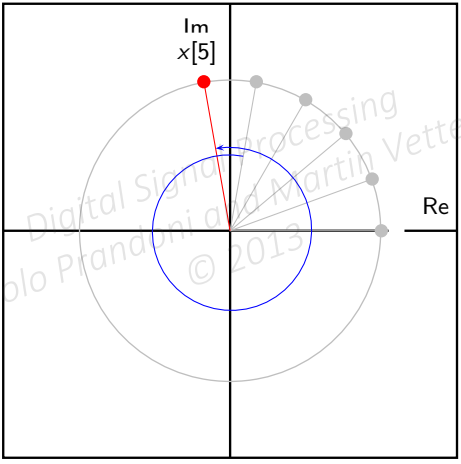
When $\pi/\Omega_0 < T_s < 2\pi/\Omega_0,\ \pi < \omega_0 < 2\pi \ldots$



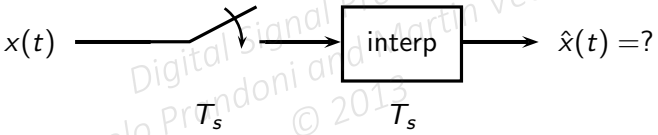
When $T_s > 2\pi/\Omega_0,\ \omega_0 > 2\pi \ldots$



When $T_s > 2\pi/\Omega_0,\ \omega_0 > 2\pi \ldots$



Aliasing



Aliasing

$$x(t) = e^{j\Omega_0 t}$$

sampling period	digital frequency	$\hat{x}(t)$	
$T_s < \pi/\Omega_0$	$0 < \omega_0 < \pi$	$e^{j\Omega_0 t}$	
$\pi/\Omega_0 < T_s < 2\pi/\Omega_0$	$\pi < \omega_0 < 2\pi$	$e^{j\Omega_1 t}$,	$\Omega_1 = \Omega_0 - 2\pi/T_s$
$T_s > 2\pi/\Omega_0$	$\omega_0 > 2\pi$	$e^{j\Omega_2 t}$,	$\Omega_2 = \Omega_0 \mod (2\pi/T_s)$

Again, with a simple sinusoid and using hertz

$$x(t) = \cos(2\pi F_0 t)$$

$$x[n] = x(nT_s) = \cos(\omega_0 n)$$

$$F_s = 1/T_s$$

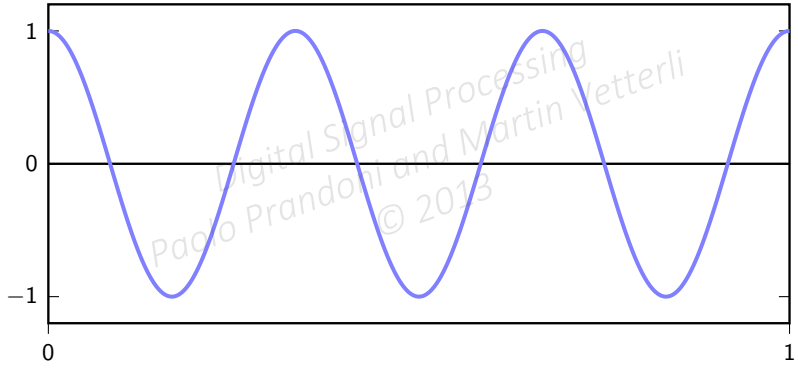
$$\omega_0 = 2\pi(F_0/F_s)$$

Sampling a Sinusoid

sampling frequency	digital frequency	result
$F_s > 2F_0$	$0 < \omega_0 < \pi$	OK
$F_s = 2F_0$	$\omega_0 = \pi$	max digital frequency: $x[n] = (-1)^n$
$F_0 < F_s < 2F_0$	$\pi < \omega_0 < 2\pi$	negative frequency $\omega_0 - 2\pi$
$F_s < F_0$	$\omega_0 > 2\pi$	full aliasing: $\omega_0 \mod 2\pi$

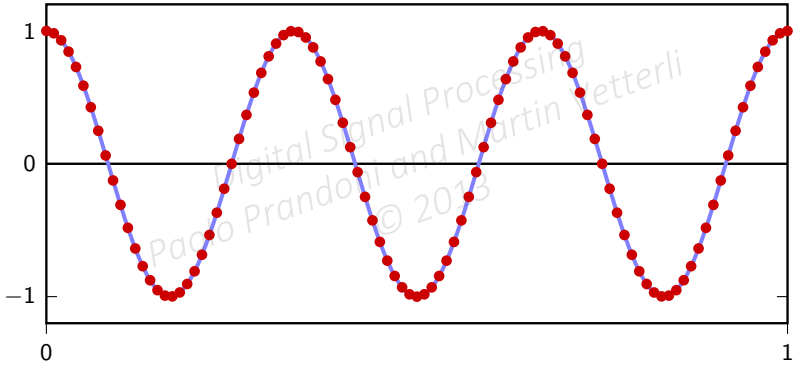
Aliasing: Sampling a Sinusoid

$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



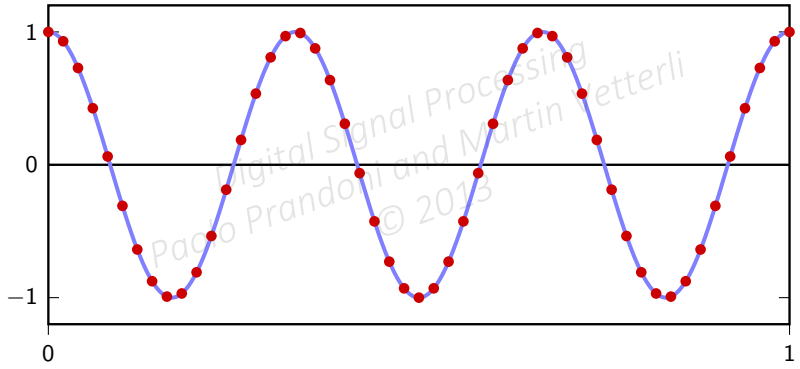
Aliasing: Sampling a Sinusoid

$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



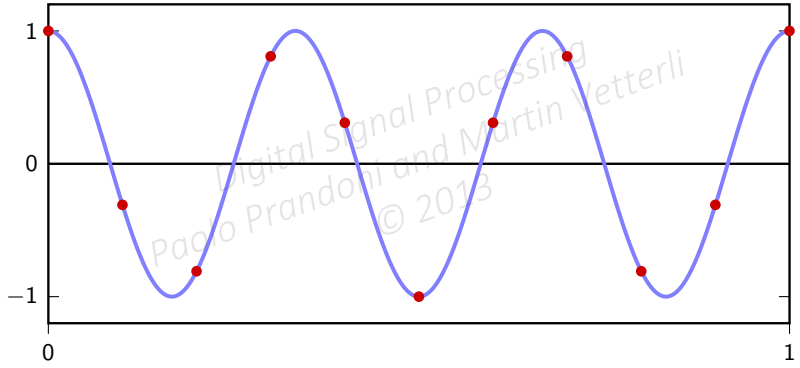
Aliasing: Sampling a Sinusoid

$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



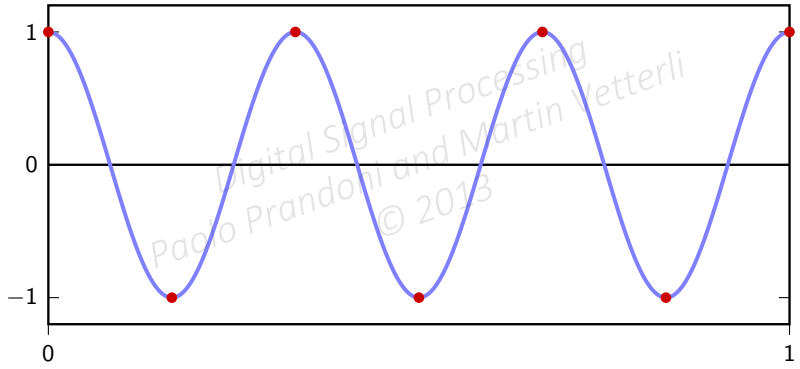
Aliasing: Sampling a Sinusoid

$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



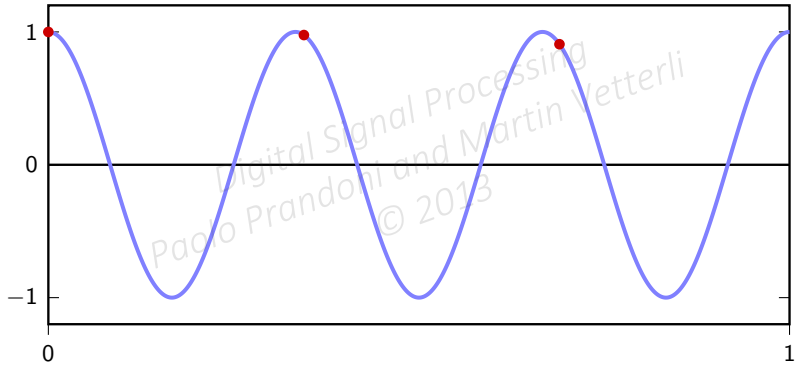
Aliasing: Sampling a Sinusoid

$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



Aliasing: Sampling a Sinusoid

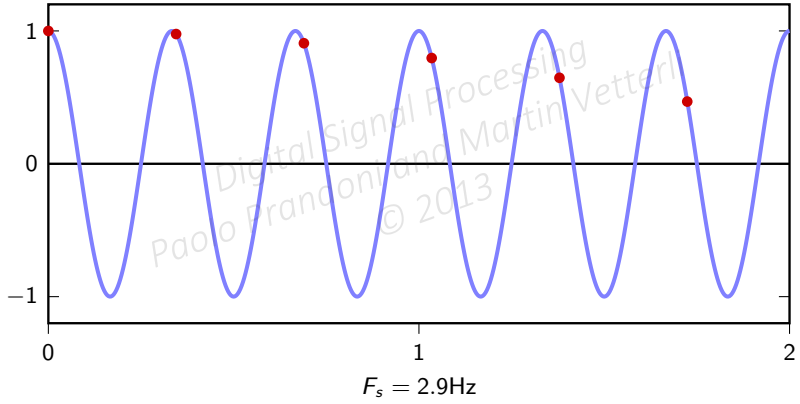
$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



$F_s = 2.9\text{Hz}$

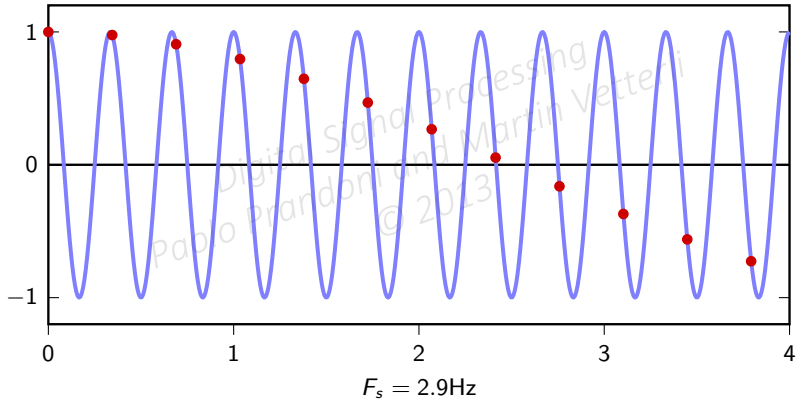
Aliasing: Sampling a Sinusoid

$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



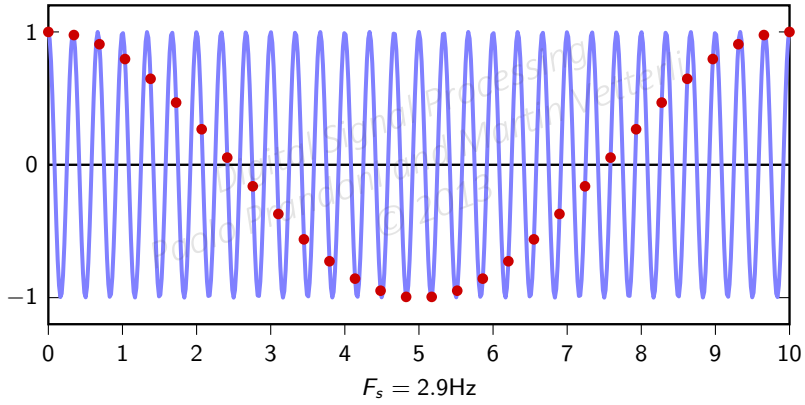
Aliasing: Sampling a Sinusoid

$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



Aliasing: Sampling a Sinusoid

$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



END OF MODULE 6.4

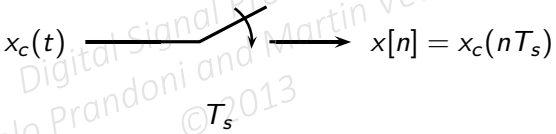
Digital Signal Processing

Module 6.5: Sampling and Aliasing

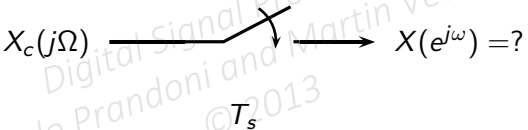
Overview:

- Aliasing for arbitrary spectra
- Examples

Raw-sampling an arbitrary signal

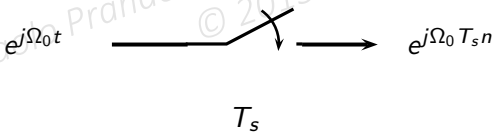


Raw-sampling an arbitrary signal



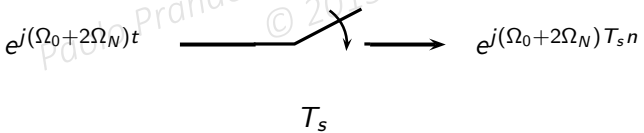
Key idea

- pick T_s (and set $\Omega_N = \pi / T_s$)
- pick $\Omega_0 < \Omega_N$



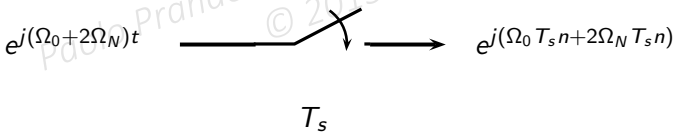
Key idea

- pick T_s (and set $\Omega_N = \pi / T_s$)
- pick $\Omega_0 < \Omega_N$



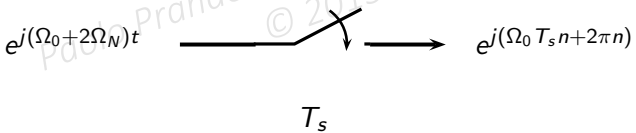
Key idea

- pick T_s (and set $\Omega_N = \pi / T_s$)
- pick $\Omega_0 < \Omega_N$



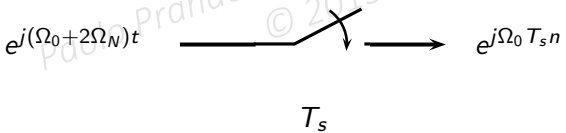
Key idea

- pick T_s (and set $\Omega_N = \pi / T_s$)
- pick $\Omega_0 < \Omega_N$



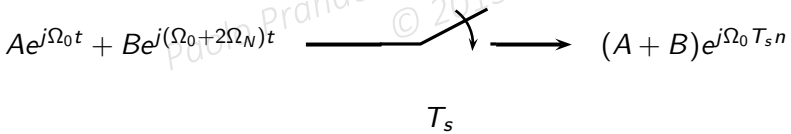
Key idea

- pick T_s (and set $\Omega_N = \pi / T_s$)
- pick $\Omega_0 < \Omega_N$



Key idea

- pick T_s (and set $\Omega_N = \pi / T_s$)
- pick $\Omega_0 < \Omega_N$



Spectrum of raw-sampled signals

start with the inverse Fourier Transform

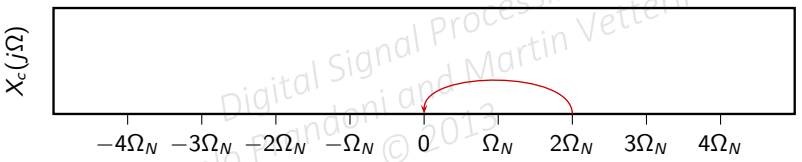
$$x[n] = x_c(nT_s) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_c(j\Omega)e^{j\Omega\, nT_s}d\Omega$$

Spectrum of raw-sampled signals

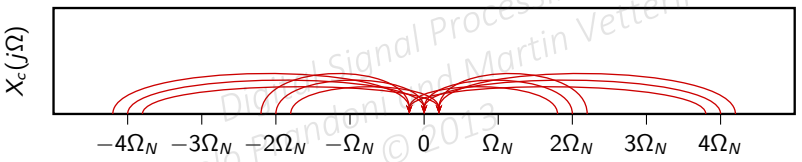
frequencies $2\Omega_N$ apart will be aliased, so split the integration interval

$$x[n] = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \int_{(2k-1)\Omega_N}^{(2k+1)\Omega_N} X_c(j\Omega) e^{j\Omega n T_s} d\Omega$$

Spectrum of raw-sampled signals



Spectrum of raw-sampled signals



Spectrum of raw-sampled signals

with a change of variable and using $e^{j(\Omega+2k\Omega_N)T_s n} = e^{j\Omega T_s n}$:

$$x[n] = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \int_{-\Omega_N}^{\Omega_N} X_c(j(\Omega - 2k\Omega_N)) e^{j\Omega nT_s} d\Omega$$

$$= \frac{1}{2\pi} \int_{-\Omega_N}^{\Omega_N} \left[\sum_{k=-\infty}^{\infty} X_c(j(\Omega - 2k\Omega_N)) \right] e^{j\Omega nT_s} d\Omega$$

Spectrum of raw-sampled signals

periodization of the spectrum; define:

$$\tilde{X}_c(j\Omega) = \sum_{k=-\infty}^{\infty} X_c(j(\Omega - 2k\Omega_N))$$

so that:

$$x[n] = \frac{1}{2\pi} \int_{-\Omega_N}^{\Omega_N} \tilde{X}_c(j\Omega) e^{j\Omega\, nT_s} d\Omega$$

Spectrum of raw-sampled signals

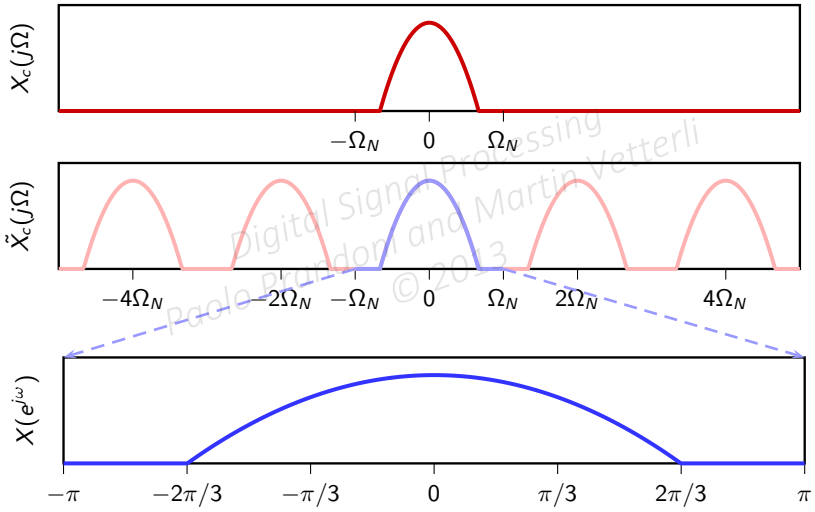
set $\omega = \Omega T_s$:

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{1}{T_s} \tilde{X}_c \left(j \frac{\omega}{T_s} \right) e^{j\omega n} d\omega$$

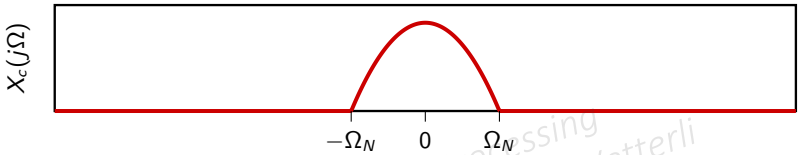
$$= \text{IDTFT} \left\{ \frac{1}{T_s} \tilde{X}_c \left(j \frac{\omega}{T_s} \right) \right\}$$

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

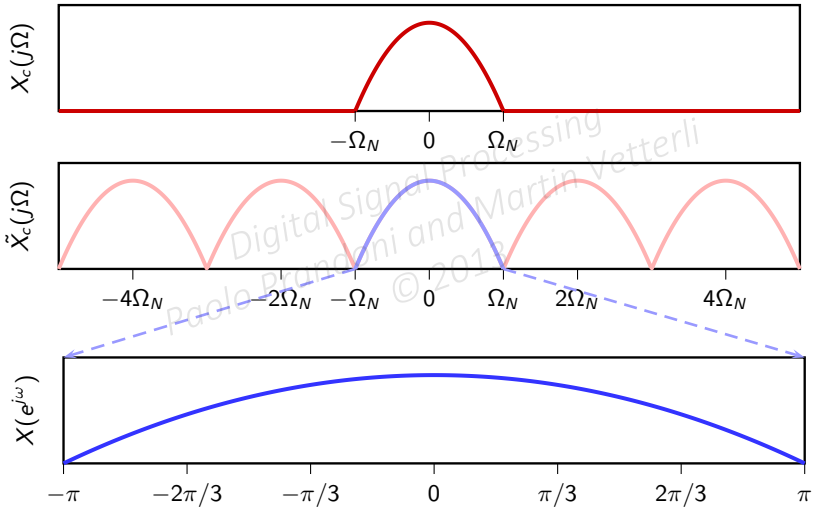
Example: signal bandlimited to Ω_0 and $\Omega_N > \Omega_0$



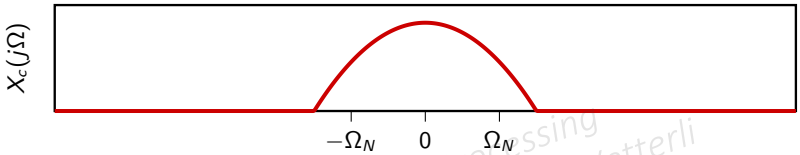
Example: signal bandlimited to Ω_0 and $\Omega_N = \Omega_0$



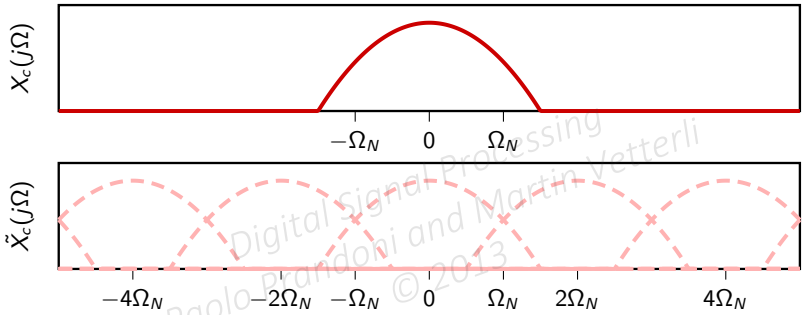
Example: signal bandlimited to Ω_0 and $\Omega_N = \Omega_0$



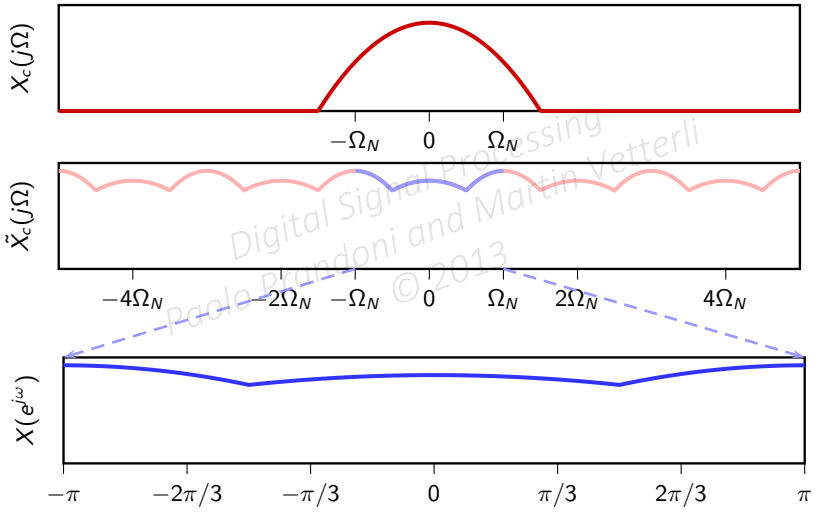
Example: signal bandlimited to Ω_0 and $\Omega_N < \Omega_0$



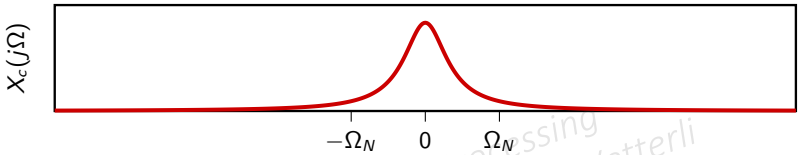
Example: signal bandlimited to Ω_0 and $\Omega_N < \Omega_0$



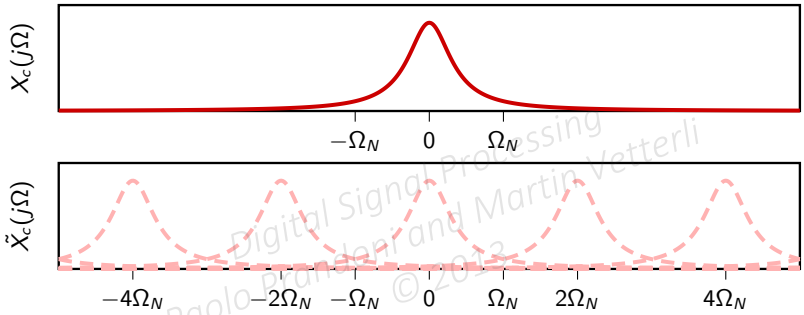
Example: signal bandlimited to Ω_0 and $\Omega_N < \Omega_0$



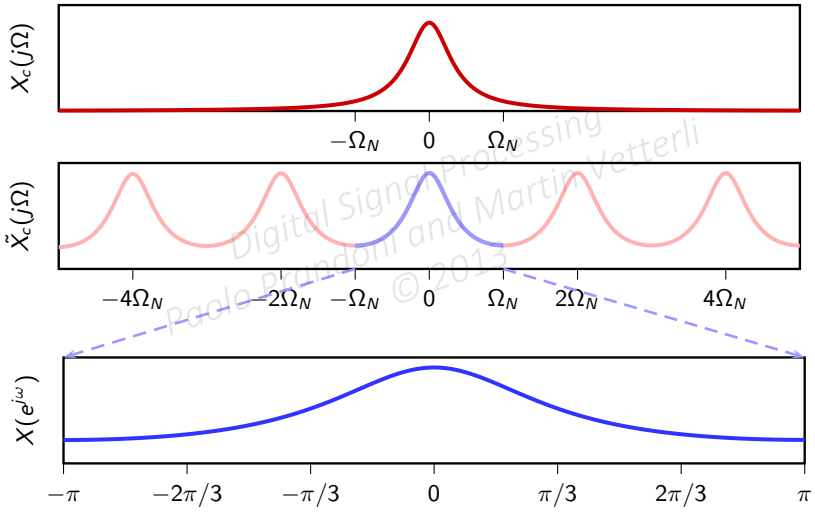
Example: non-bandlimited signal



Example: non-bandlimited signal



Example: non-bandlimited signal



Sampling strategies

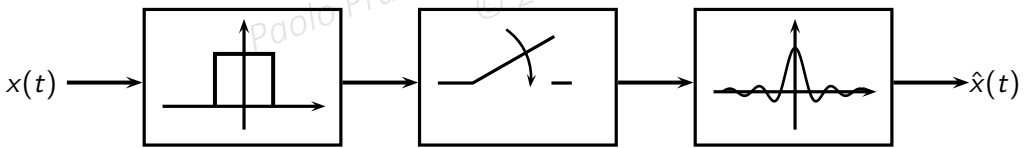
given a sampling period T_s

- if the signal is bandlimited to π/T_s or less, raw sampling is fine (i.e. equivalent to sinc sampling up to a scaling factor T_s)
- if the signal is not bandlimited, two choices:
  - bandlimit via a lowpass filter *in the continuous-time domain* before sampling (i.e. sinc sampling)
  - or, raw sample the signal and incur aliasing
- aliasing sounds horrible, so usually we choose to bandlimit in continuous time

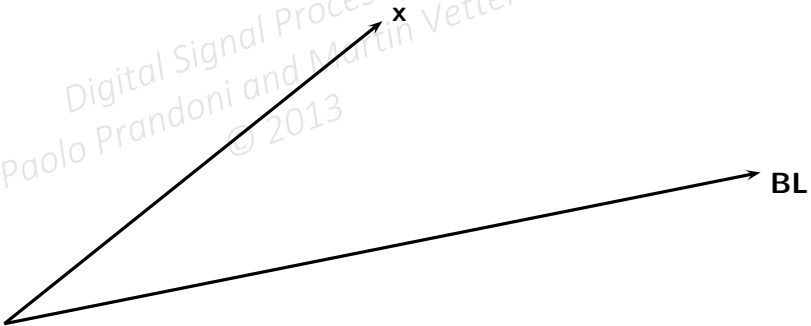
Sinc Sampling and Interpolation

$$\hat{x}[n] = \langle \text{sinc}\left(\frac{t - nT_s}{T_s}\right), x(t) \rangle \;=\; (\text{sinc}_{T_s} * x)(nT_s)$$

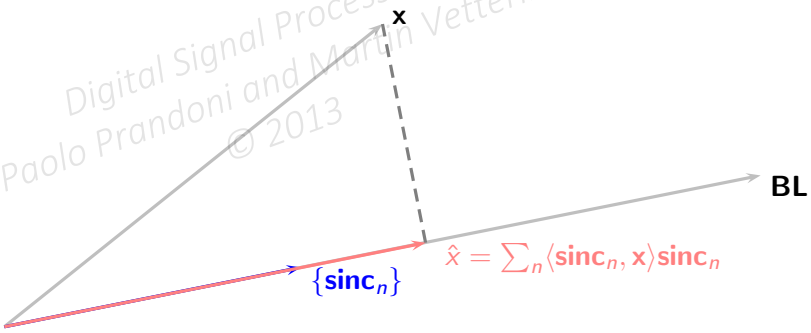
$$\hat{x}(t) = \sum_n x[n]\,\text{sinc}\left(\frac{t - nT_s}{T_s}\right)$$



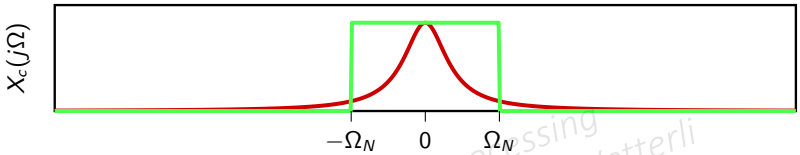
Least squares approximation with sinc sampling and interpolation



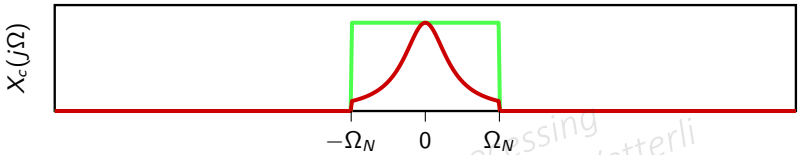
Least squares approximation with sinc sampling and interpolation



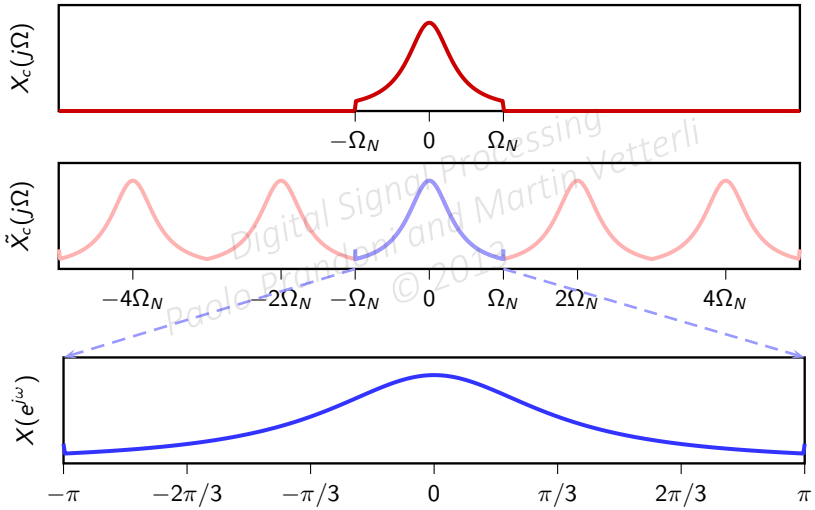
Least squares approximation with sinc sampling and interpolation



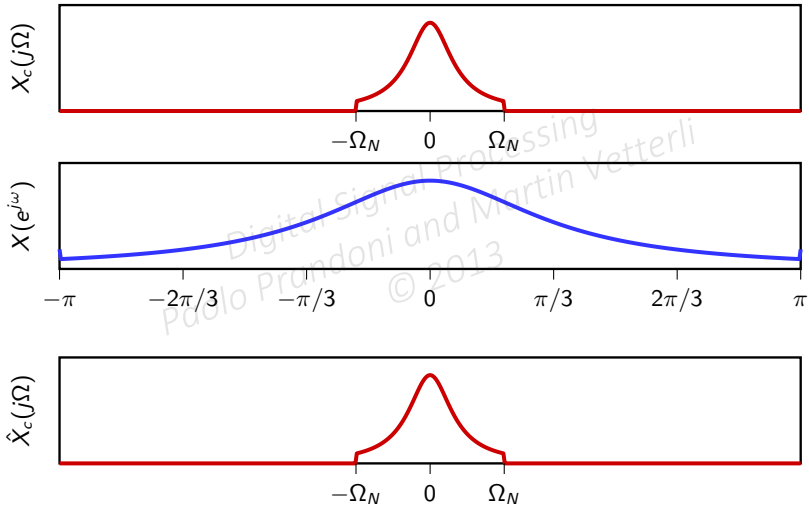
Least squares approximation with sinc sampling and interpolation



Least squares approximation with sinc sampling and interpolation



Least squares approximation with sinc sampling and interpolation



END OF MODULE 6.5

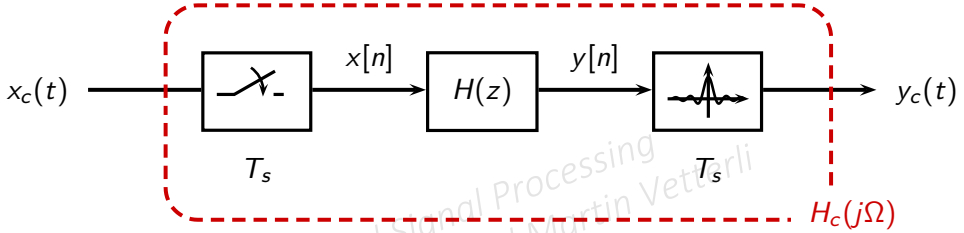
Digital Signal Processing

Module 6.6: Discrete-time Processing and Continuous-time Signals

Overview:

- Impulse invariance
- Duality
- Examples

Basic setup



assume $x_c(t)$ is Ω_N-bandlimited:

- $X(e^{j\omega}) = \frac{1}{T_s} X_c\left(j\frac{\omega}{T_s}\right)$
- $Y(e^{j\omega}) = X(e^{j\omega})\, H(e^{j\omega})$
- $Y_c(j\Omega) = T_s\, Y(e^{j\Omega T_s})$

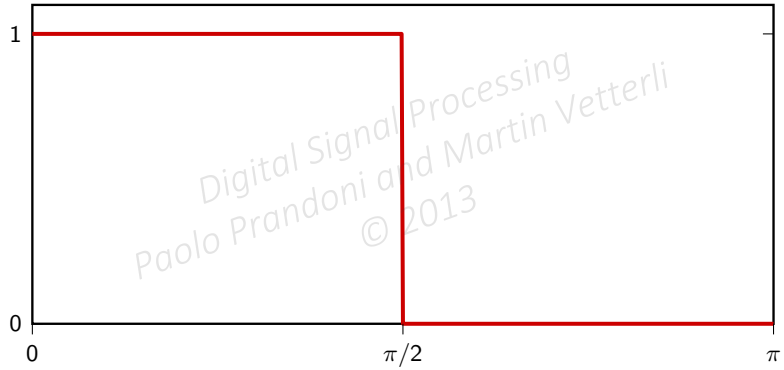
Impulse invariance

$$Y_c(j\Omega) = X_c(j\Omega)\, H(e^{j\pi\Omega/\Omega_N})$$

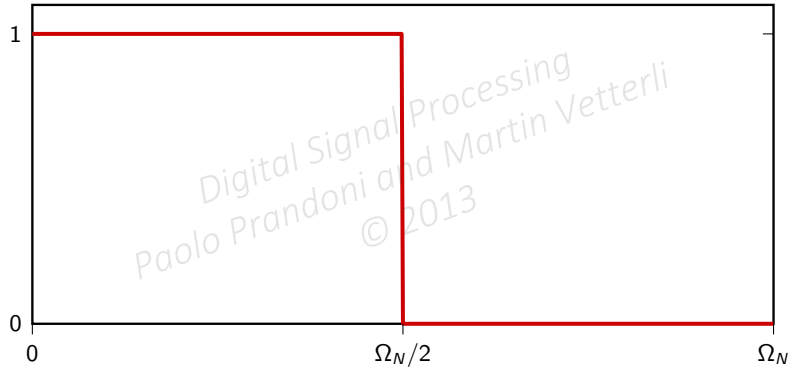
Impulse invariance

$$H_c(j\Omega) = H(e^{j\pi\Omega/\Omega_N})$$

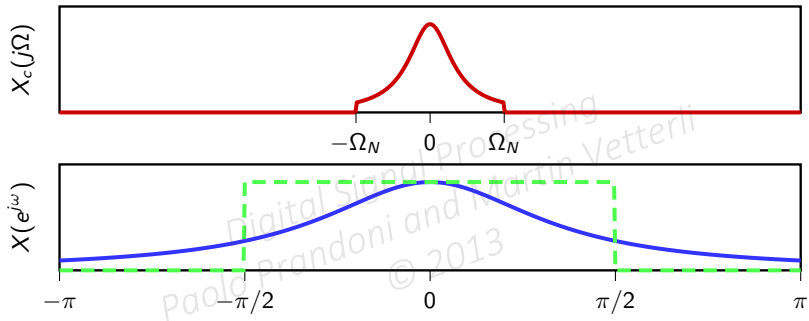
Impulse invariance



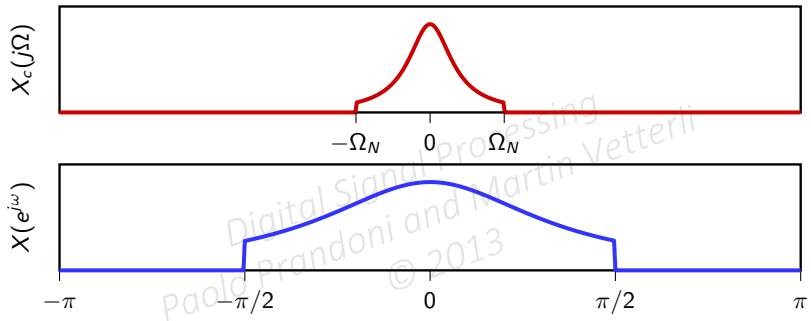
Impulse invariance



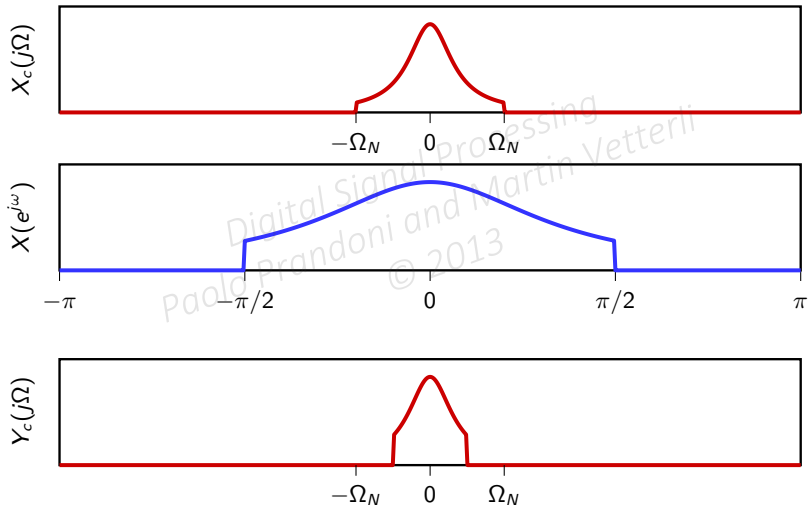
DT processing of CT signals



DT processing of CT signals



DT processing of CT signals



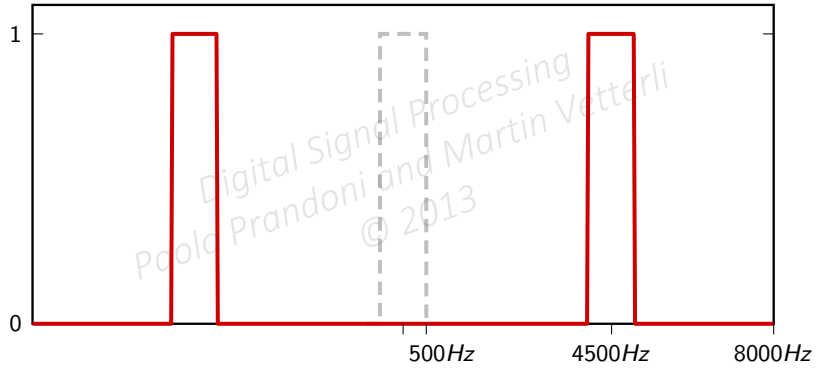
Example

design a discrete-time filter to isolate a band of frequencies between 4000 and 5000Hz; input signals are bandlimited to 7KHz.

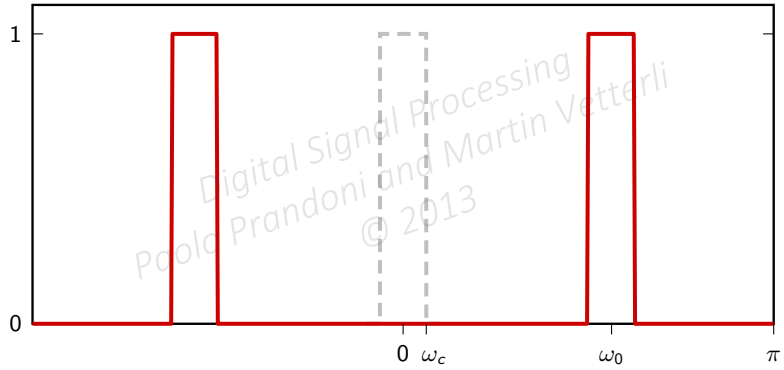
Example

- 7KHz band limit $\Rightarrow$ we can use any sampling frequency above 14KHz
- pick $F_s = 16KHz$ so that $\Omega_N = 2\pi \cdot 8000$ rad/s
- we need a bandpass with a 1000Hz bandwidth
- start with a lowpass with cutoff 500Hz
- modulate it to center it around 4500Hz

Impulse invariance



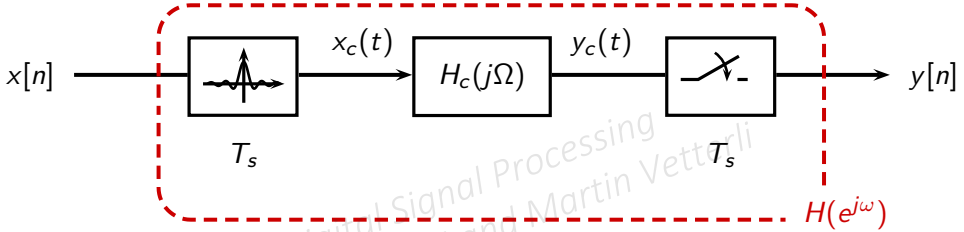
Impulse invariance



Example

- $\omega_c = \pi \frac{\Omega_c}{\Omega_N} = \pi \frac{500}{8000} = 0.0625\pi$
- $\omega_0 = \pi \frac{4500}{8000} = 0.5625\pi$
- design an FIR lowpass with cutoff ω_c using your favorite method
- multiply the impulse response by $2 \cos \omega_0 n$

Duality



we can pick any T_s so pick $T_s = 1$:

- $X_c(j\Omega) = X(e^{j\Omega})$
- $Y_c(j\Omega) = X_c(j\Omega)H_c(j\Omega)$
- LTI systems cannot change the bandwidth $\Rightarrow Y(e^{j\omega}) = Y_c(j\omega)$

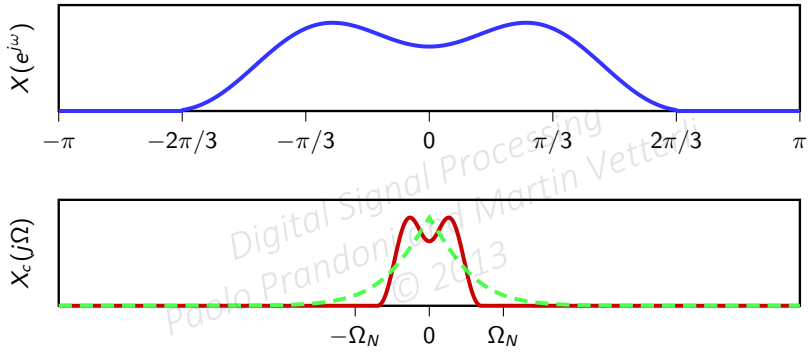
Duality

$$Y(e^{j\omega}) = X(e^{j\omega})\, H_c(j\omega)$$

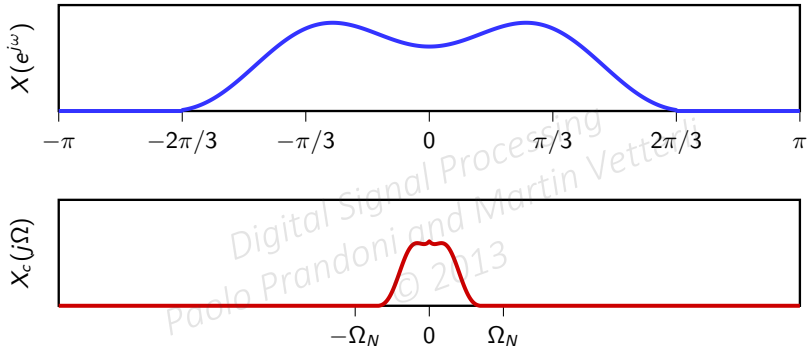
Duality

$$H(e^{j\omega}) = H_c(j\omega)$$

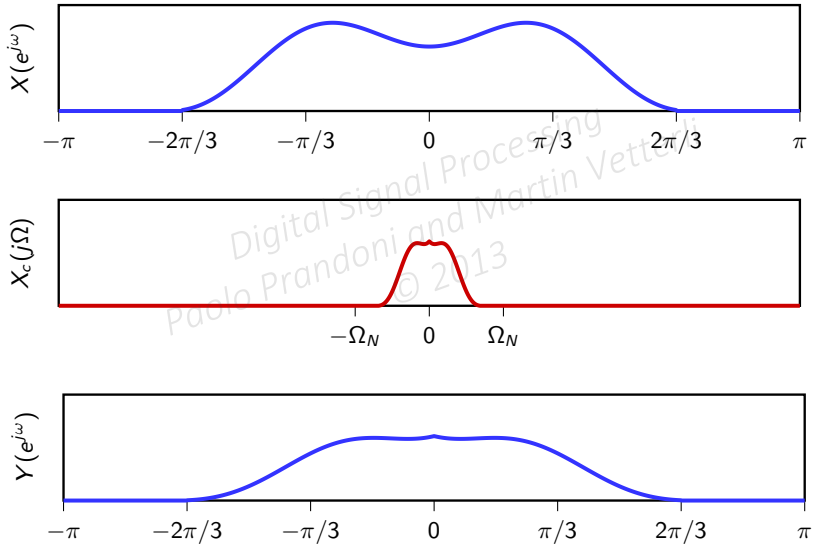
CT processing of DT signals



CT processing of DT signals



CT processing of DT signals

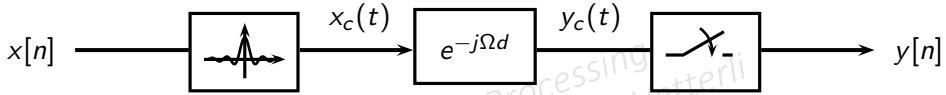


Example: fractional delay

$$H(e^{j\omega}) = e^{-j\omega d}$$

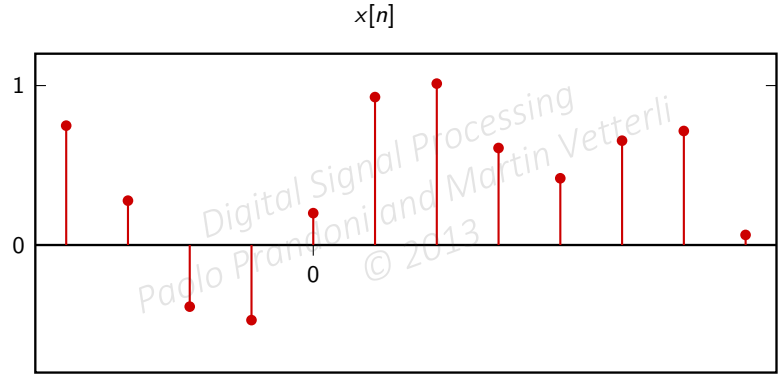
- if $d \in \mathbb{Z}$, simple delay
- if $d \notin \mathbb{Z}$, $h[n] = \text{sinc}(n - d)$...

By duality

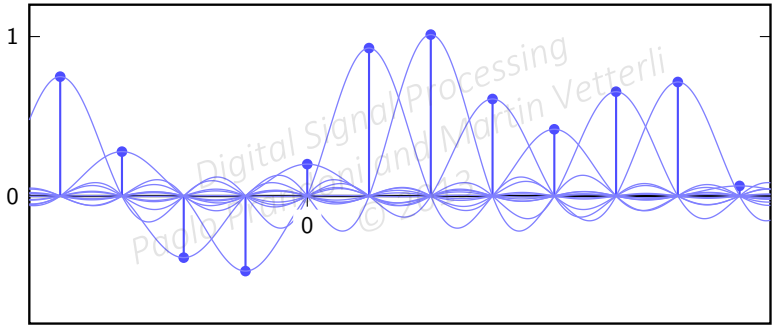


- $Y_c(j\Omega) = e^{-j\Omega d} X_c(j\Omega)$
- $y_c(t) = x_c(t - d)$
- $y[n]$ is the sampled interpolation of $x[n]$ delayed by d

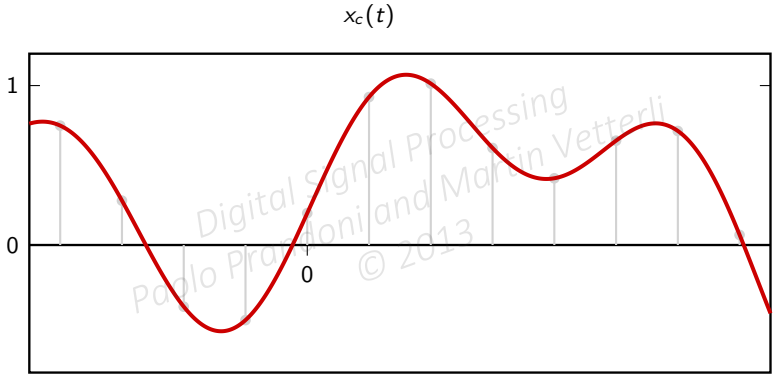
Example: fractional delay



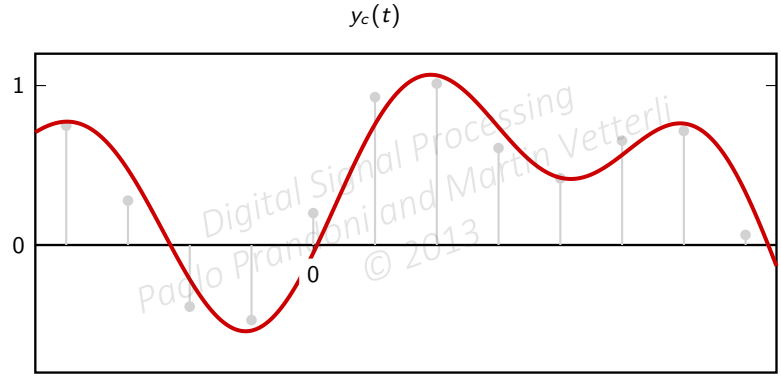
Example: fractional delay



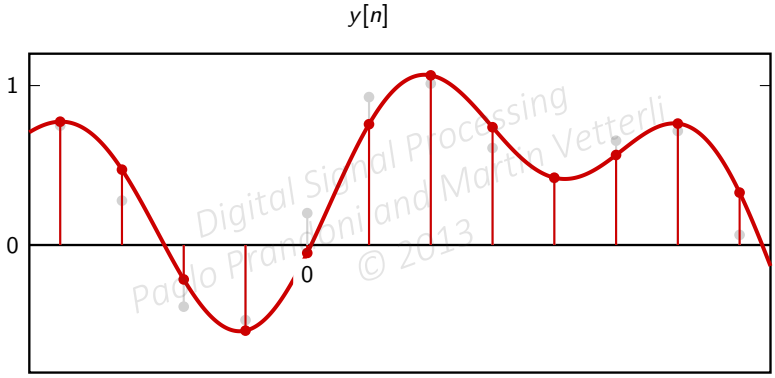
Example: fractional delay



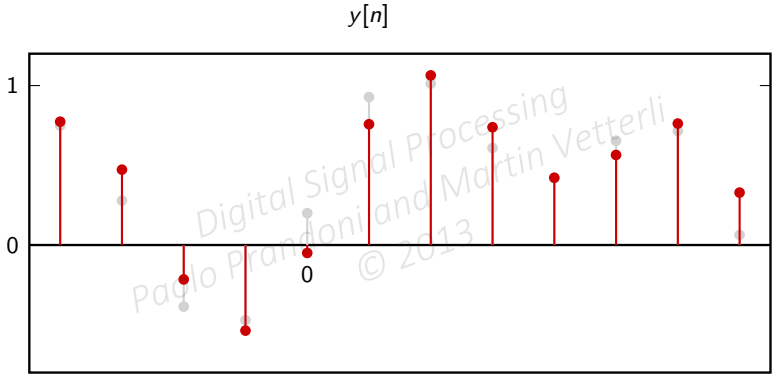
Example: fractional delay



Example: fractional delay



Example: fractional delay



Example: fractional delay

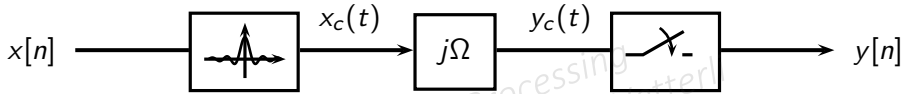
- to delay a discrete-time signal by a fraction of a sample we need an ideal filter!
- efficient time-variant approximations exist (see Module 11)

Example: differentiator

$$H(e^{j\omega}) = j\omega$$

- in continuous time we know that $\mathrm{FT}\{x_c'(t)\} = j\Omega\, X_c(j\Omega)$
- in discrete time...

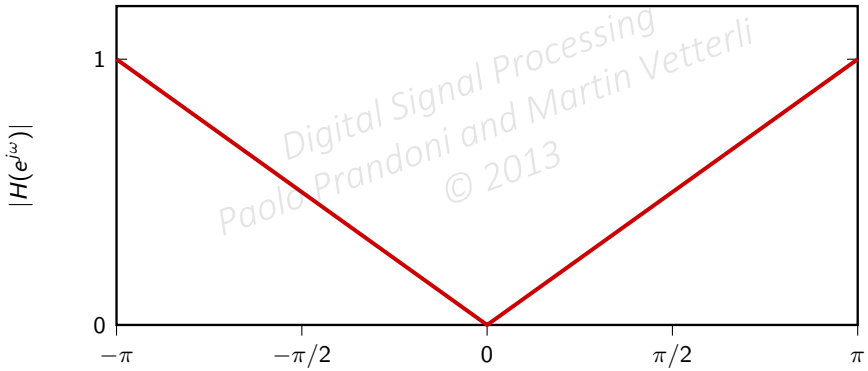
By duality



- $Y_c(j\Omega) = j\Omega X_c(j\Omega)$
- $y_c(t) = x_c'(t)$
- $y[n]$ is the sampled interpolation of $x[n]$, differentiated

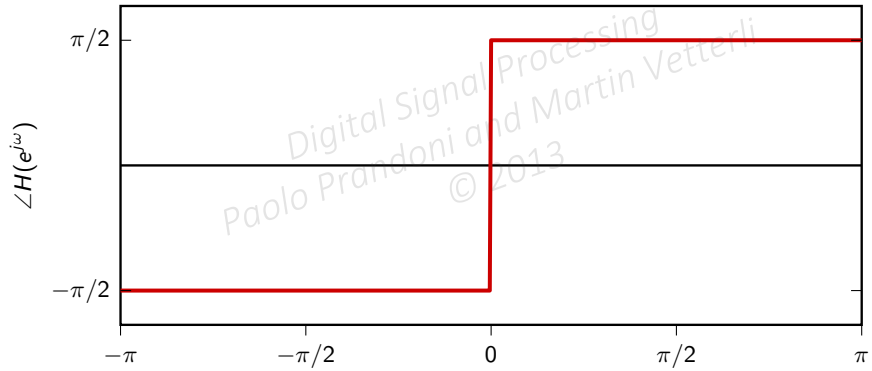
Digital differentiator, magnitude response

$$H(e^{j\omega}) = j\omega$$



Digital differentiator, phase response

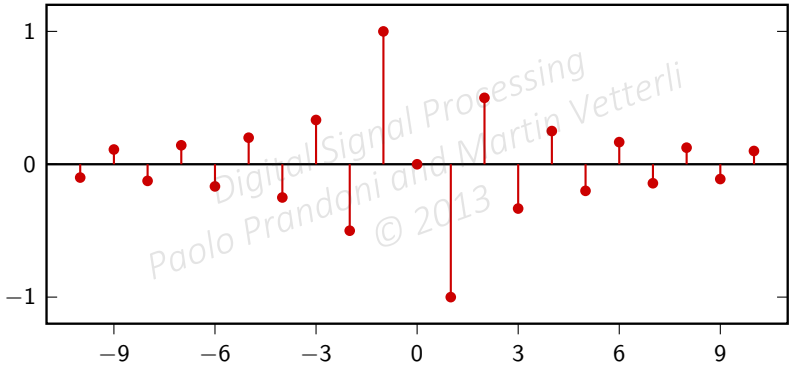
$$H(e^{j\omega}) = j\omega$$



Digital differentiator, impulse response

$$
\begin{aligned}
h[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} j\omega e^{j\omega n} d\omega \\
&= \ldots \textit{(integration by parts)} \ldots \\
&= \begin{cases} 0 & n = 0 \\ \dfrac{(-1)^n}{n} & n \neq 0 \end{cases}
\end{aligned}
$$

Digital differentiator, impulse response



Digital differentiator

- the digital differentiator is again an ideal filter!
- many approximations exist, with different properties

Wrap up

- Continuous-time processing of discrete-time sequences
- Discrete-time processing of continuous-time signals
- Jumping back and forth using sampling and interpolation
- In practice: Many applications of processing continuous-time signals in discrete time!

END OF MODULE 6.6

END OF MODULE 6