COM-202: Signal Processing

Chapter 7.b: Sampling and applications

from interpolation to sampling

Interpolation

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

- we want $x_c(n) = x[n]$ so, for all n:
  - $i_n(0) = 1$
  - $i_n(k) = 0$ for k a nonzero integer
- we would prefer $x_c(t)$ to be smooth (i.e. continuously differentiable)

Interpolation

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

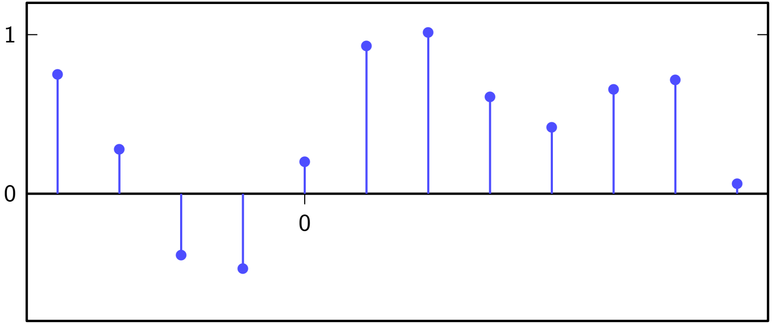
- global interpolation:
  - (good) $x_c(t)$ is a maximally smooth polynomial
  - (bad) must use $2N+1$ distinct interpolation kernels $i_n(t) = L_n^{(N)}(t)$
- local interpolation:
  - (good) just a single interpolation kernel $i_n(t) = i(t)$
  - (bad) discontinuities in $x_c(t)$ or its derivatives
- as $N \to \infty$ the two methods converge to sinc interpolation

Sinc interpolation

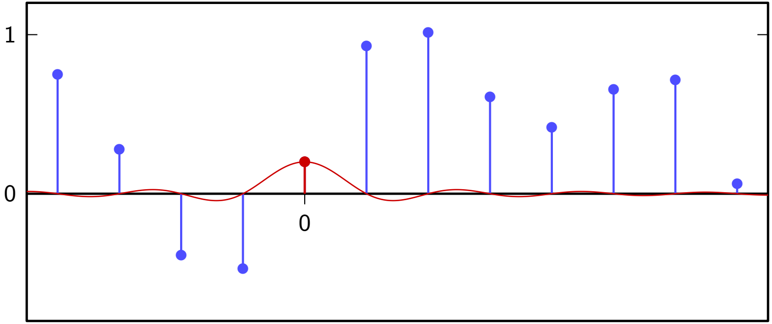
$$x_c(t) = \sum_{n=-\infty}^{\infty} x[n] \operatorname{sinc}(t-n)$$

$$X_c(f) = \begin{cases} X(2\pi f) & |f| \leq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

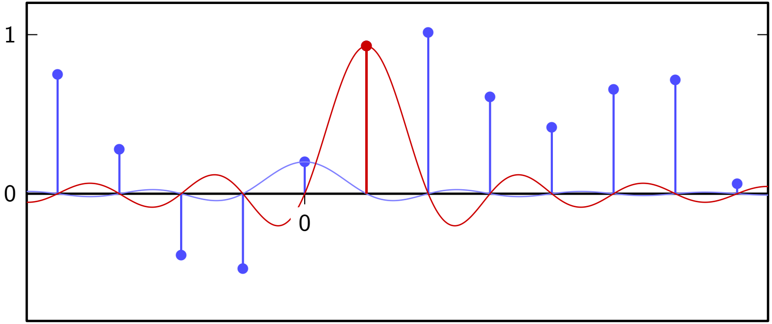
Sinc interpolation



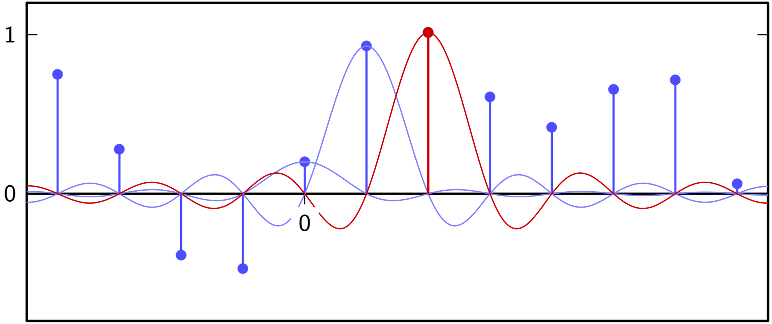
Sinc interpolation



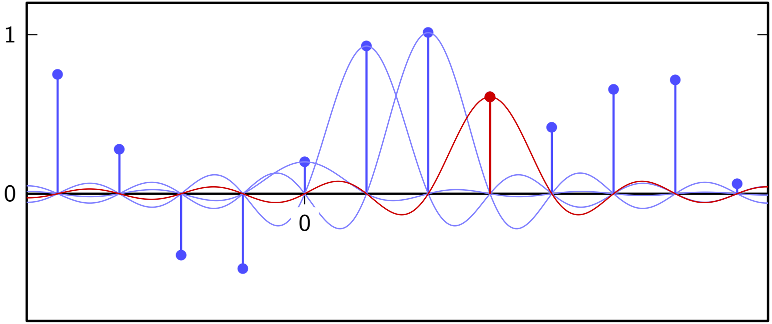
Sinc interpolation



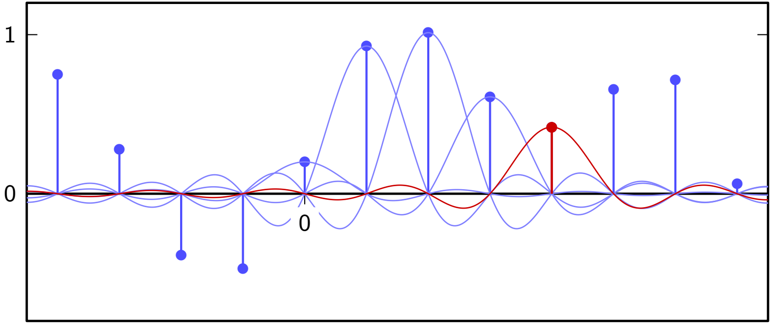
Sinc interpolation



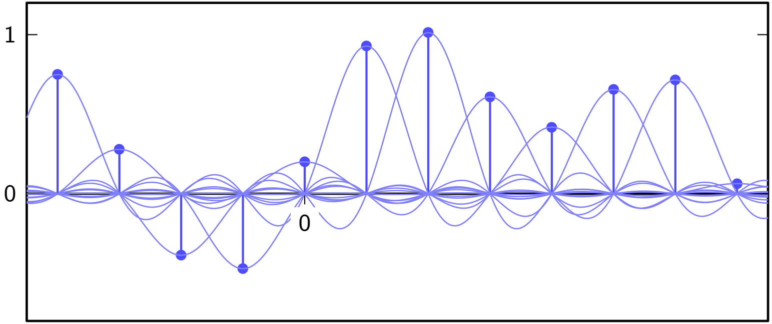
Sinc interpolation



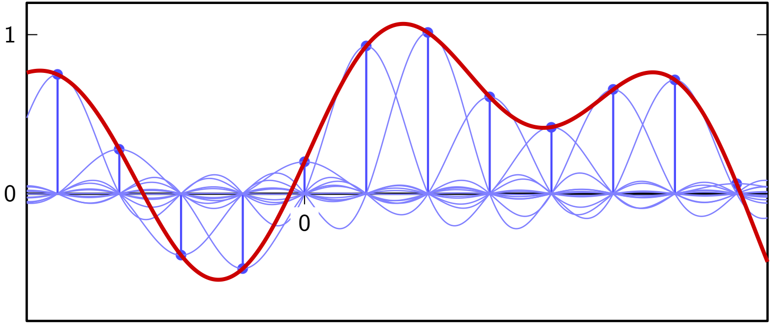
Sinc interpolation



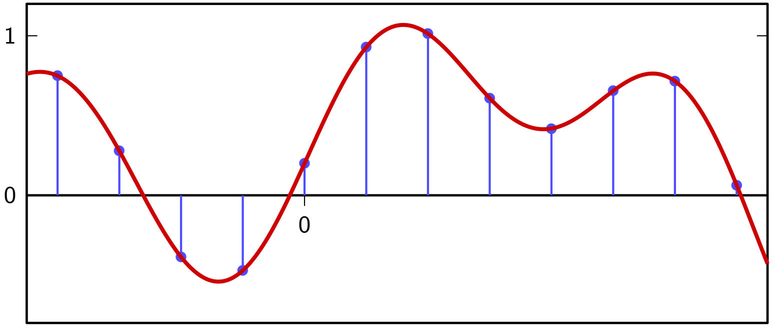
Sinc interpolation



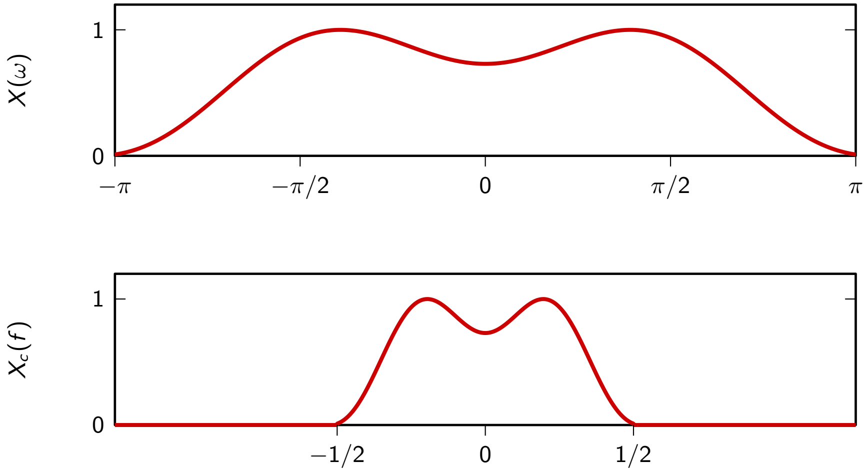
Sinc interpolation



Sinc interpolation



Spectrum of sinc-interpolated signal

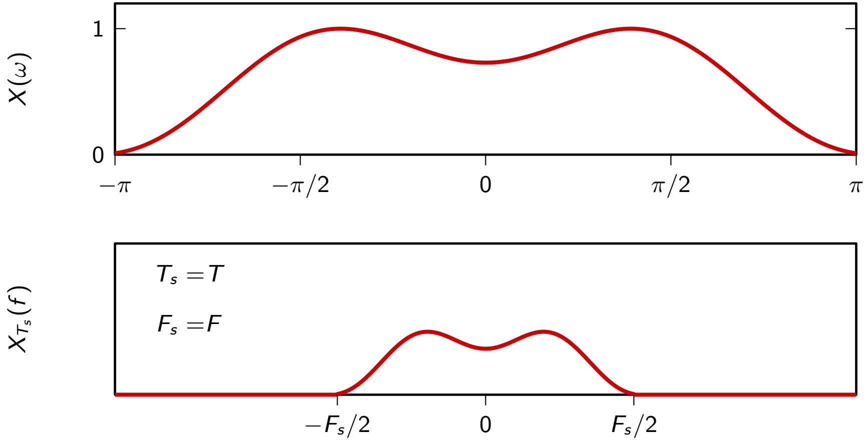


Sinc interpolation with timebase T_s

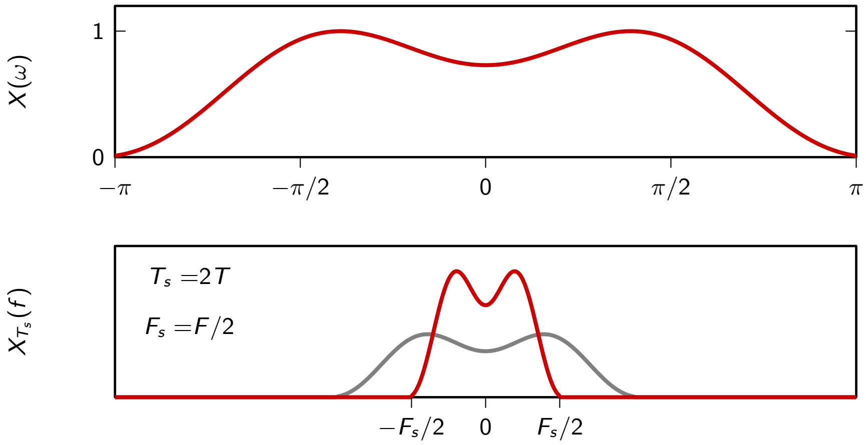
$$x_{T_s}(t) = x_c(t/T_s) = \sum_{n=-\infty}^{\infty} x[n]\,\mathrm{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

$$X_{T_s}(f) = \begin{cases} \dfrac{1}{F_s} X\left(2\pi \dfrac{f}{F_s}\right) & |f| \leq F_s/2 \\ 0 & \text{otherwise} \end{cases}$$

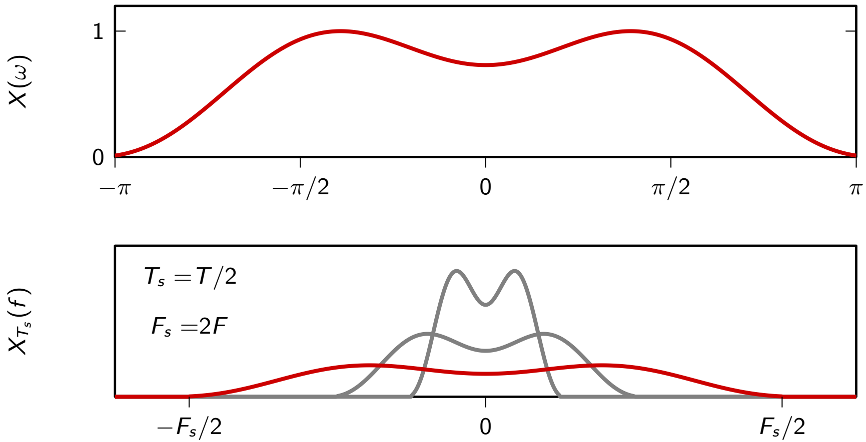
Spectrum of interpolated signals



Spectrum of interpolated signals



Spectrum of interpolated signals



the sampling theorem

slides from lecture 7.a

Space of bandlimited signals

every finite-energy sequence can be interpolated into a bandlimited signal

$$x[n] \in \ell_2(\mathbb{Z}) \xrightarrow{\quad T_s \quad} x_c(t) \in F_s\text{-BL} \subset L_2(\mathbb{R})$$

Space of bandlimited signals

every finite-energy sequence can be interpolated into a bandlimited signal

$$x[n] \in \ell_2(\mathbb{Z}) \quad \overset{T_s}{\underset{?}{\longleftrightarrow}} \quad x_c(t) \in F_s\text{-BL} \subset L_2(\mathbb{R})$$

is the reverse also true?

is every BL function the interpolation of a discrete-time sequence?

Let's simplify things

$$x(t) \stackrel{\text{CTFT}}{\longleftrightarrow} X(f) \quad \Longleftrightarrow \quad x(\alpha t) \stackrel{\text{CTFT}}{\longleftrightarrow} \frac{1}{\alpha} X\left(\frac{f}{\alpha}\right)$$

- if $x(t)$ is F_s-BL, then $x(F_s\, t) = x(t/T_s)$ is 1-BL
- let's focus on the set of 1-BL signals

The key points of the sampling theorem

- the space of 1-BL functions is a Hilbert space
- the set $\mathbf{S} = \{\varphi_n\}_{n \in \mathbb{Z}}$, where $\varphi_n(t) = \text{sinc}(t - n)$, is an orthonormal basis for it
- therefore any $\mathbf{x}_c \in$ 1-BL can be uniquely expressed as the linear combination

$$\mathbf{x}_c = \sum_n a_n \varphi_n$$

  where, because of orthonormality, $a_n = \langle \varphi_n, \mathbf{x}_c \rangle$
- we will show that $\langle \varphi_n, \mathbf{x}_c \rangle = x_c(n)$: the basis expansion coefficients are simply the samples of the continuous-time signal $\mathbf{x}_c$
- therefore the discrete-time sequence $x[n] = x_c(n)$ is an equivalent representation of the continuous-time signal $\mathbf{x}_c$

The space of 1-BL signals

- elements of the space are finite-energy (square-integrable) functions whose Fourier transform is zero outside of the $[-1/2, 1/2]$ interval
- closed under addition and scalar multiplication because linear combinations of 1-BL functions are still 1-BL functions
- inner product is the standard inner product in $L_2(\mathbb{R})$:

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

- we also should prove completeness... that is the tricky part but here we will simply accept that it's true

The sinc basis for the 1-BL space

let's show that $\mathbf{S} = \{\varphi_n\}_{n\in\mathbb{Z}}$ is an orthonormal basis

$$
\begin{aligned}
\langle \varphi_n, \varphi_m \rangle &= \int_{-\infty}^{\infty} \operatorname{sinc}(t-n)\operatorname{sinc}(t-m)\,dt \\
&= \int_{-\infty}^{\infty} \operatorname{sinc}(\tau)\operatorname{sinc}((m-n)-\tau)\,d\tau \\
&= (\varphi * \varphi)(m-n) \\
&= \int_{-\infty}^{\infty} \operatorname{rect}^2(f)\, e^{j2\pi f(m-n)} df \\
&= \int_{-1/2}^{1/2} e^{j2\pi f(m-n)} df = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\omega(m-n)} d\omega = \begin{cases} 1 & \text{for } m = n \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

Sampling as a basis expansion

for any $\mathbf{x}_c \in$ 1-BL:

$$\begin{aligned}
\langle \varphi_n, \mathbf{x}_c \rangle &= \int_{-\infty}^{\infty} \text{sinc}(t - n) x_c(t)\, dt \\
&= \int_{-\infty}^{\infty} \text{sinc}(n - t) x_c(t)\, dt \\
&= (\varphi * \mathbf{x}_c)(n) \\
&= \int_{-\infty}^{\infty} \text{rect}\,(f)\, X_c(f) e^{j2\pi f n} df \\
&= \int_{-\infty}^{\infty} X_c(f) e^{j2\pi f n} df \\
&= x_c(n)
\end{aligned}$$

Sampling as a basis expansion

for any $\mathbf{x}_c \in$ 1-BL:

analysis formula:

$$x[n] = \langle \varphi_n, \mathbf{x}_c \rangle$$

synthesis formula:

$$\mathbf{x}_c = \sum_{n=-\infty}^{\infty} x[n]\, \varphi_n$$

The sampling theorem, general case

- the space of F_s-bandlimited functions is a Hilbert space
- the functions $\left\{\text{sinc}\left(\frac{t-nT_s}{T_s}\right)\right\}_{n\in\mathbb{Z}}$ form an orthogonal basis for it ($T_s = 1/F_s$)
- basis vectors are not orthonormal, their norm is $\sqrt{T_s}$
- if $x(t) \in F_s$-BL then

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

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

- therefore the discrete-time sequence $x[n] = x(nT_s)$ is a complete representation of the continuous-time signal $x(t)$

Sampling as a basis expansion for arbitrary bandwidth

for any $\mathbf{x} \in F_s$-BL:

analysis formula:

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

synthesis formula:

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

The sampling theorem, lossless condition

- assume $x(t)$ is F_s-BL, that is, $X(f) = 0$ for $|f| > F_s/2$
- $x(t)$ is also F-BL for any choice of $F \geq F_s$
- therefore the sequence $x[n] = x(nT_s)$ is a complete representation of $x(t)$ as long as $T_s \leq 1/F_s$

an F_s-bandlimited continuous-time signal $x(t)$ can be sampled with no loss of information using any sampling frequency larger than F_s
(or, equivalently, using a sampling period $T_s \leq 1/F_s$)

The Nyquist frequency

- real-valued continuous-time signals have a symmetric magnitude spectrum
- the maximum frequency value F_N for which the spectrum is nonzero is called the Nyquist frequency
- the Nyquist frequency of an F_s-bandlimited real-valued signal is $F_N = F_s/2$

any real-valued signal can be sampled with no loss of information
as long as the sampling frequency is greater than $2F_N$

back to lecture 7.b

Space of bandlimited signals

every discrete-time signal can be interpolated into a **bandlimited** continuous-time signal

$$x[n] \in \ell_2(\mathbb{Z}) \quad \underset{F_s = 1/T_s}{\overset{T_s = 1/F_s}{\rightleftharpoons}} \quad x(t) \in F_s\text{-BL} \subset L_2(\mathbb{R})$$

every bandlimited signal can be represented **exactly** by a discrete-time sequence

Sinc sampling as an orthogonal basis decomposition

for any $\mathbf{x} \in F_s$-BL:

analysis formula:

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

synthesis formula:

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

Sinc sampling as an orthogonal subspace projection

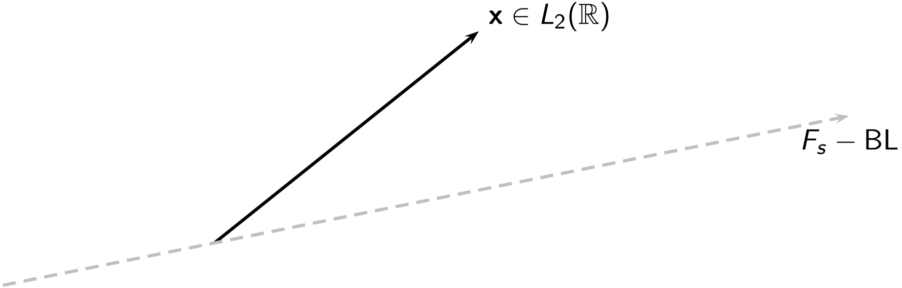
for any $\mathbf{x} \in L_2(\mathbb{R})$, the sequence

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

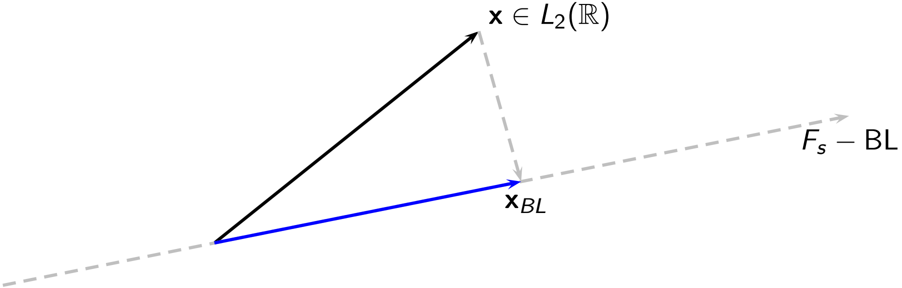
defines the orthogonal projection (i.e. the least squares approximation)
of $\mathbf{x}$ onto the subspace of F_s-BL functions

important: if $\mathbf{x} \notin F_s$-BL, then $x[n] \neq T_s x(nT_s)$

Sinc sampling as an orthogonal subspace projection



Sinc sampling as an orthogonal subspace projection



Sinc sampling: the internals

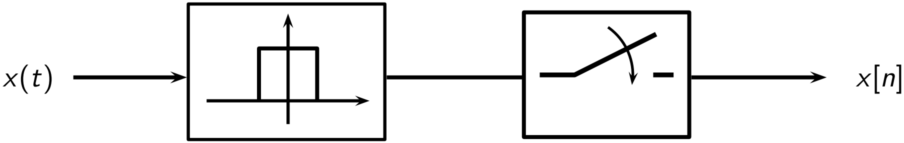
$$\begin{aligned} x[n] &= \left\langle \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right), x(t) \right\rangle \\ &= \int_{-\infty}^{\infty} \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right) x(t)dt \\ &= \int_{-\infty}^{\infty} \operatorname{sinc}\left(\frac{nT_s - t}{T_s}\right) x(t)dt \\ &= (\mathbf{h} * \mathbf{x})(nT_s) \qquad \text{where } h(t) = \operatorname{sinc}(t/T_s) \end{aligned}$$

h is the impulse response of a continuous-time ideal lowpass with cutoff $f_c = F_s/2$

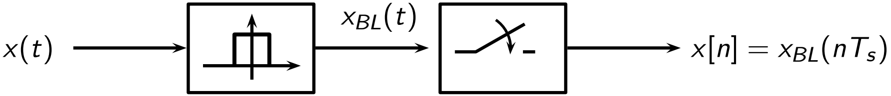
$$H(f) = \frac{1}{F_s} \operatorname{rect}\left(\frac{f}{F_s}\right)$$

Sinc sampling bandlimits the input!

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



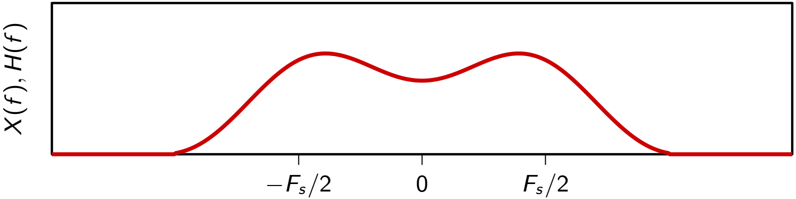
Sinc sampling bandlimits the input



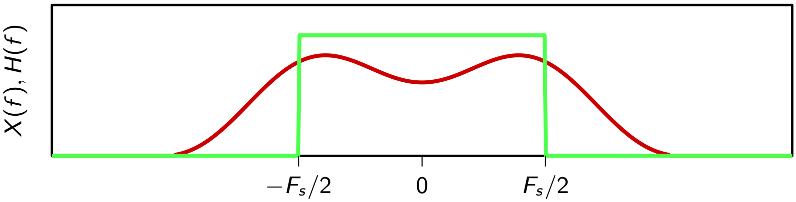
- implicit continuous-time lowpass: $h(t) = \text{sinc}(t/T_s)$, $\quad H(f) = \frac{1}{F_s}\,\text{rect}\left(\frac{f}{F_s}\right)$
- input to the raw sampler: $\mathbf{x}_{BL} = \mathbf{h} * \mathbf{x}$
- discrete-time samples: $x[n] = x_{BL}(nT_s)$

$\mathbf{x}_{BL}$ is the orthogonal projection of $\mathbf{x}$ onto the space of F_s-BL functions

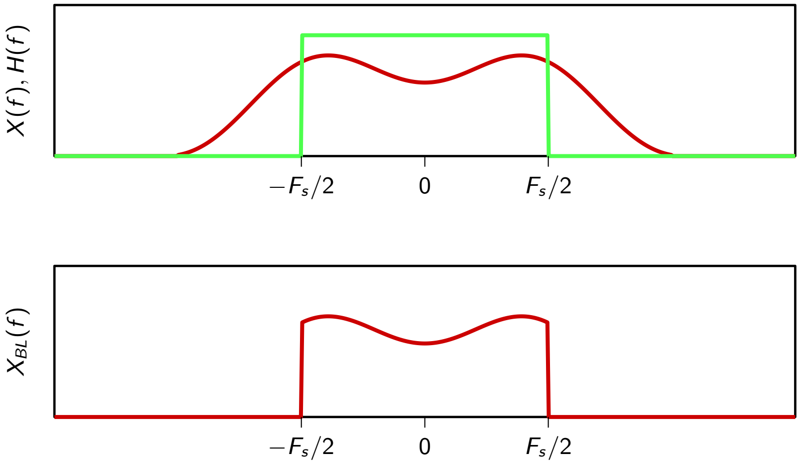
Projection onto a bandlimited subspace



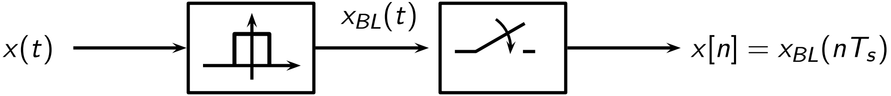
Projection onto a bandlimited subspace



Projection onto a bandlimited subspace



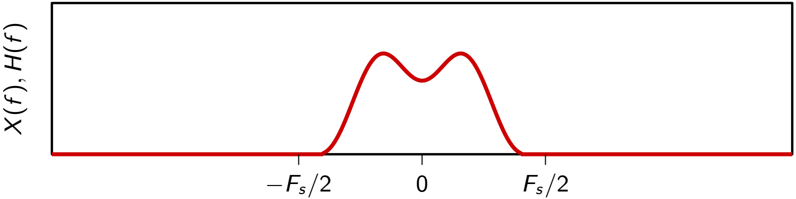
Sinc sampling of bandlimited signals



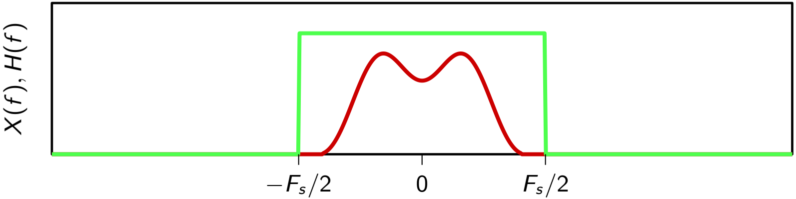
if $\mathbf{x} \in F_s$-BL:

- $\mathbf{x}_{BL} = \mathbf{x}$
- the filter doesn't do anything
- sinc sampling becomes raw sampling (which is easy to do)

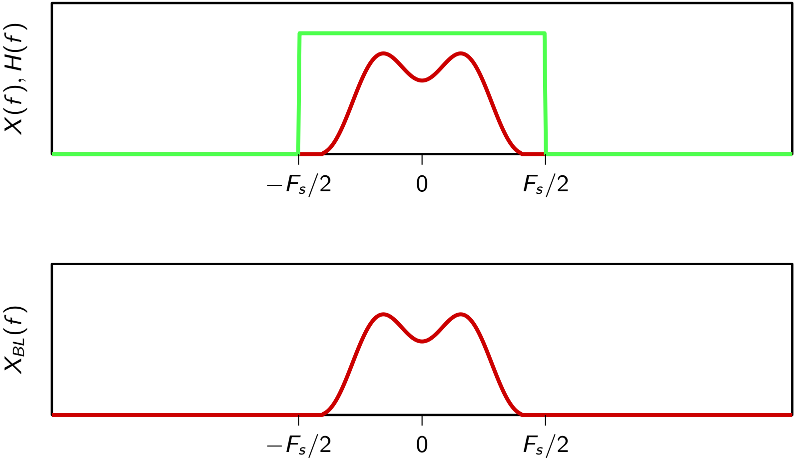
Projection onto a bandlimited subspace



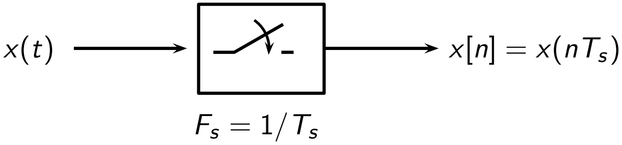
Projection onto a bandlimited subspace



Projection onto a bandlimited subspace



Raw sampling

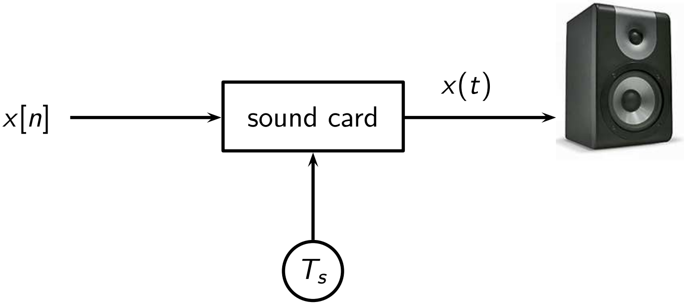


- if **x** is F_s-BL this is equivalent to sinc sampling (up to a scaling factor) and there is no loss of information
- but what happens if
  - **x** is not bandlimited?
  - **x** is bandlimited but the sampling frequency is too low?

we incur **aliasing!**

interpolation of sinusoidal signals

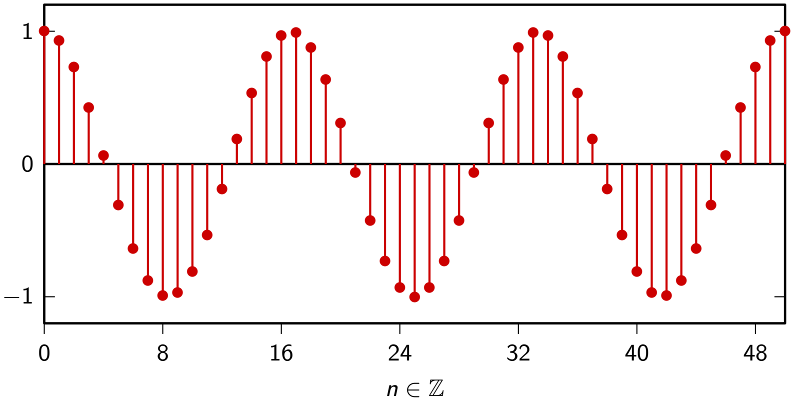
A soundcard is an interpolator



- interpolation interval T_s: interval in seconds between two consecutive samples
- interpolation rate $F_s = 1/T_s$: samples per second consumed by the soundcard

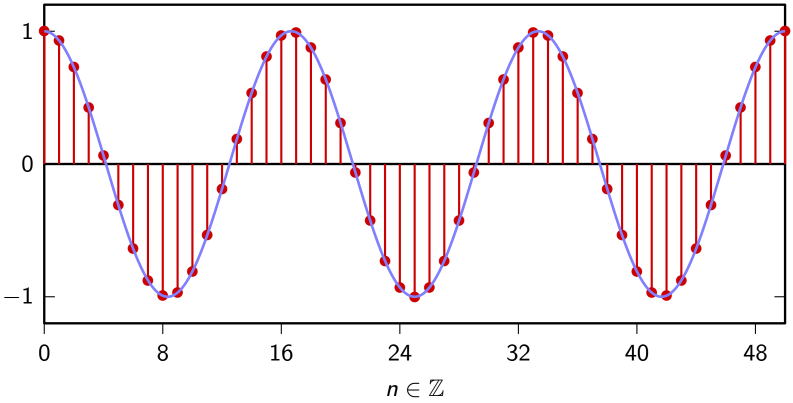
Playing a sinusoidal tone

$$x[n] = \cos(\omega_0 n) \qquad -\pi \leq \omega_0 \leq \pi$$



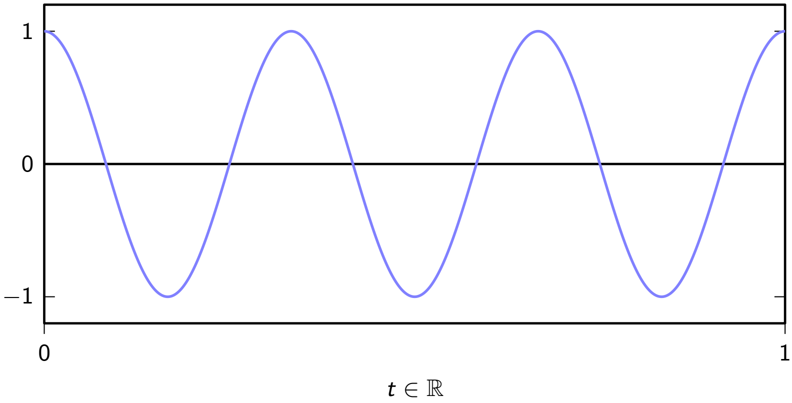
Playing a sinusoidal tone

$$x[n] = \cos(\omega_0 n) \qquad -\pi \leq \omega_0 \leq \pi$$

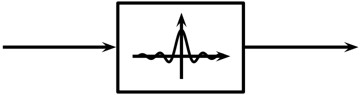


Playing a sinusoidal tone

$$x(t) = \cos(2\pi f_0 t) \qquad f_0 = (\omega_0/(2\pi))F_s$$



Sinc interpolation of a sinusoid

$x[n] = e^{j\omega_0 n}$  $s(t) = ?$

F_s

Sinc interpolation of a sinusoid

$$X(\omega) = \tilde{\delta}(\omega - \omega_0) = 2\pi \sum_{k=-\infty}^{\infty} \delta(\omega - \omega_0 - 2k\pi)$$

$$S(f) = \frac{1}{F_s} X\left(\frac{2\pi}{F_s} f\right) \operatorname{rect}\left(\frac{f}{F_s}\right)$$ spectrum of interpolation

$$= \frac{2\pi}{F_s} \delta\left(\frac{2\pi}{F_s} f - \omega_0\right)$$ rect selects only one Dirac

$$\equiv \delta\left(f - \frac{\omega_0}{2\pi} F_s\right)$$ $\delta(f/\alpha) \equiv \alpha\delta(f)$

$$= \text{CTFT}\{e^{j2\pi f_0 t}\}, \quad f_0 = (\omega_0/(2\pi))F_s$$

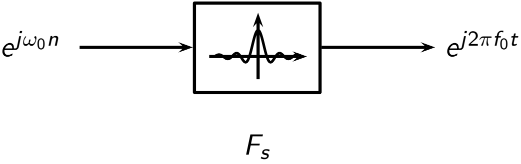
I don't like Dirac deltas...

$$\text{IDTFT}\left\{e^{j\omega\tau}\right\}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\omega\tau}e^{-j\omega n}d\omega = \ldots = \text{sinc}(n-\tau)$$

$$\text{DTFT}\left\{\text{sinc}(n-\tau)\right\}(\omega) = e^{j\omega\tau}$$

$$\begin{aligned}\sum_{n=-\infty}^{\infty} e^{j\omega_0 n}\,\text{sinc}\left(\frac{t-nT_s}{T_s}\right) &= \sum_{n=-\infty}^{\infty} \text{sinc}\left(n - t/T_s\right)e^{-j\omega_0 n} \\ &= \text{DTFT}\left\{\text{sinc}(n-t/T_s)\right\}(\omega_0) \\ &= e^{j\omega_0 t/T_s} = e^{j2\pi f_0 t}, \quad f_0 = (\omega_0/(2\pi))F_s\end{aligned}$$

Playing a sinusoidal tone



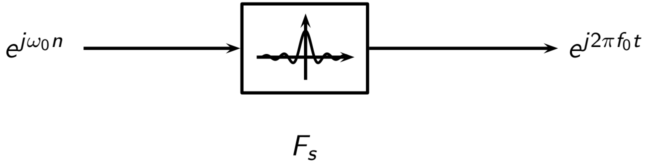
in discrete time:

- ω_0: phase increment per sample
- samples per period: $P_n = 2\pi/\omega_0$

after interpolation:

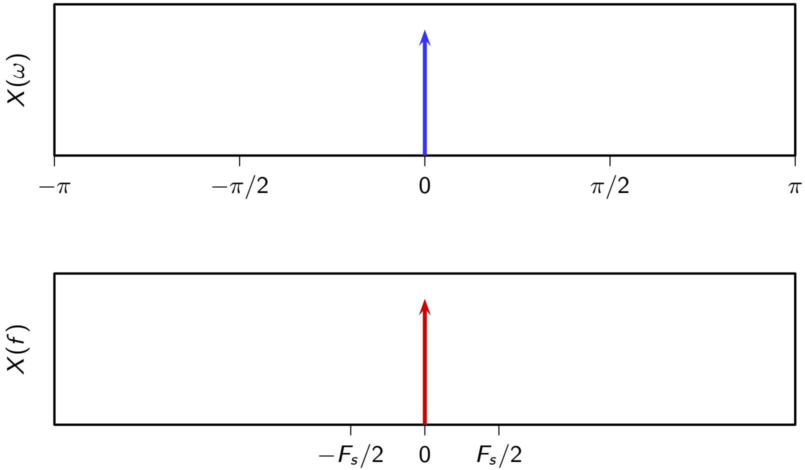
- one period lasts $P_t = P_n T_s = P_n/F_s$ seconds
- frequency is $f_0 = 1/P_t = F_s/P_n = (\omega_0/(2\pi))F_s$

Playing a sinusoidal tone: frequency range



$$-\pi \leq \omega_0 \leq \pi \qquad\qquad f_0 = \frac{\omega_0}{2\pi} F_s \qquad\qquad -F_s/2 \leq f_0 \leq F_s/2$$

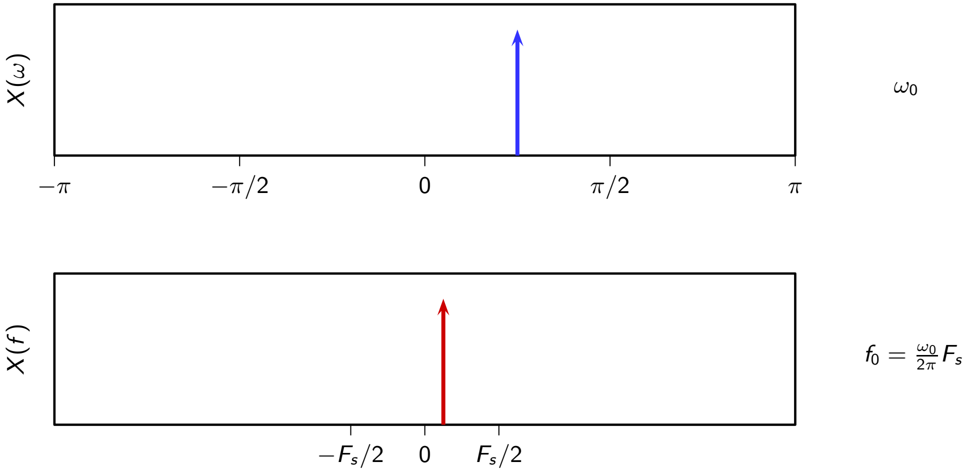
Frequency range of interpolated sinusoids



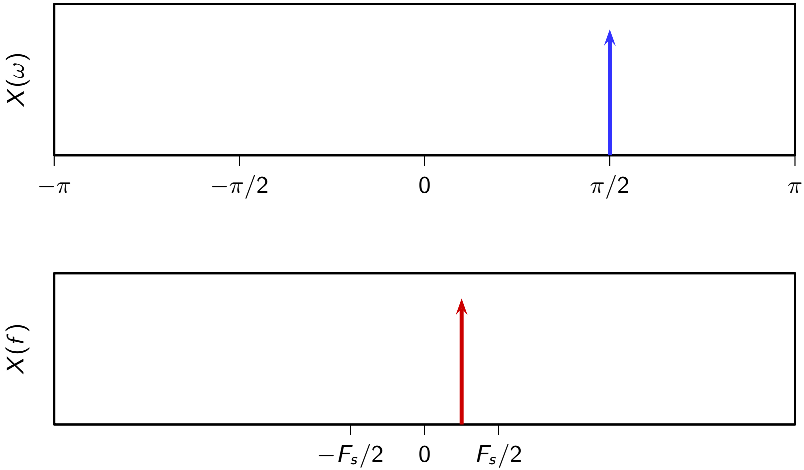
ω_0

$f_0 = \frac{\omega_0}{2\pi} F_s$

Frequency range of interpolated sinusoids



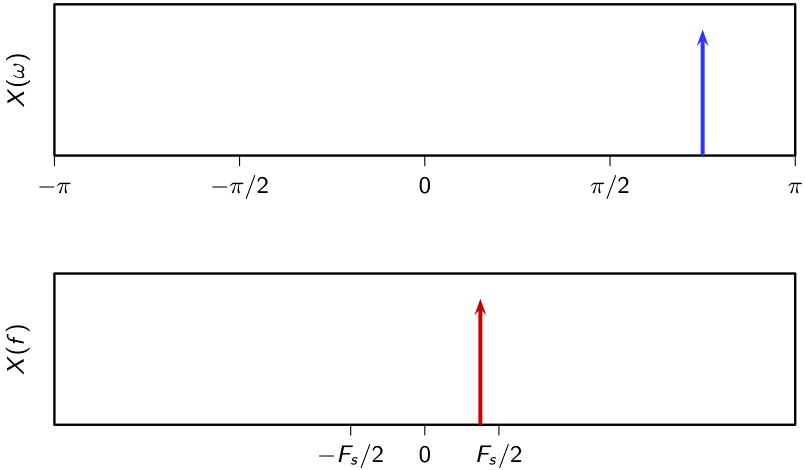
Frequency range of interpolated sinusoids



ω_0

$f_0 = \frac{\omega_0}{2\pi} F_s$

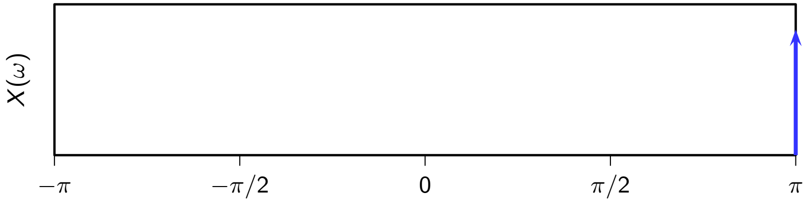
Frequency range of interpolated sinusoids



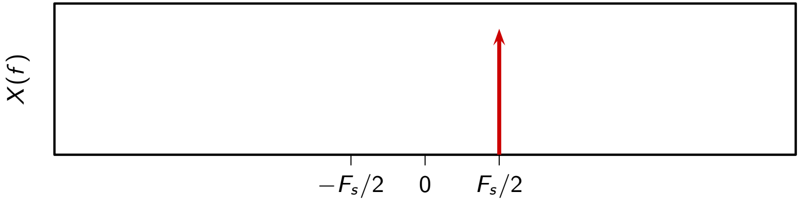
ω_0

$f_0 = \frac{\omega_0}{2\pi} F_s$

Frequency range of interpolated sinusoids

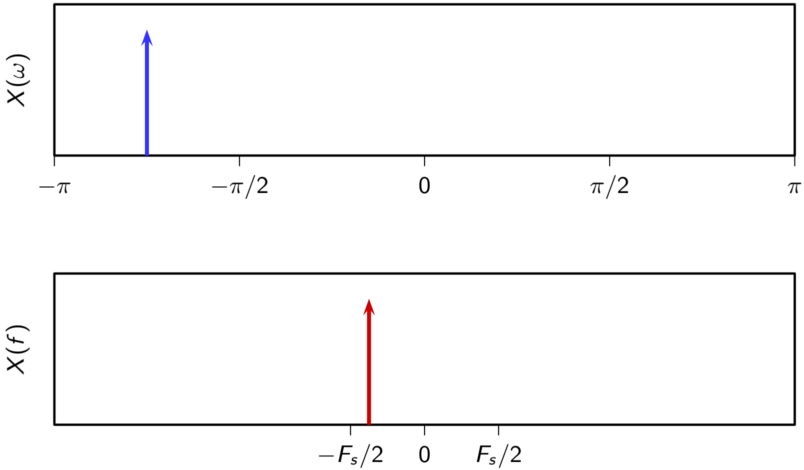


ω_0



$f_0 = \frac{\omega_0}{2\pi} F_s$

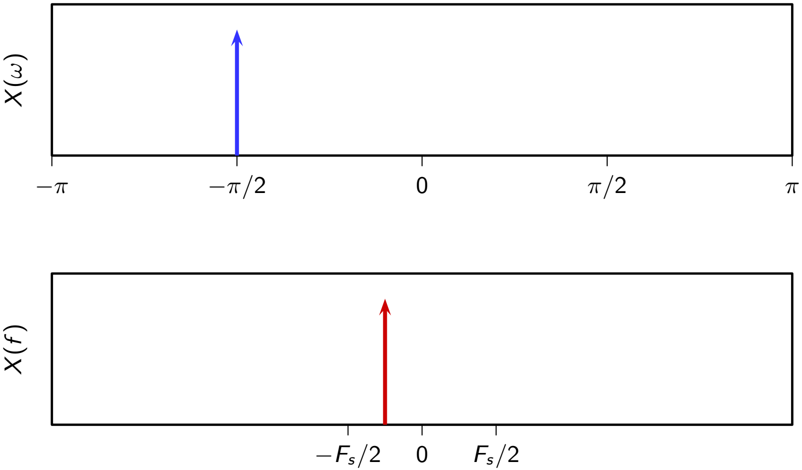
Frequency range of interpolated sinusoids



ω_0

$f_0 = \frac{\omega_0}{2\pi} F_s$

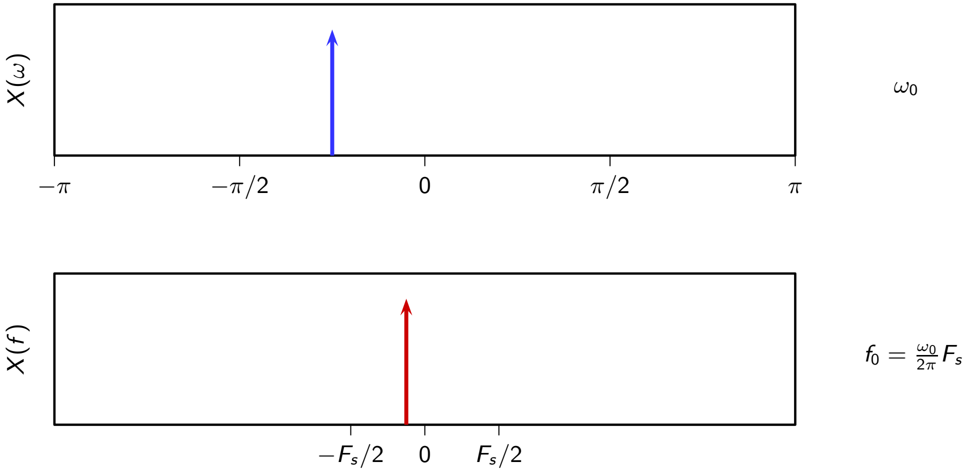
Frequency range of interpolated sinusoids



ω_0

$f_0 = \frac{\omega_0}{2\pi} F_s$

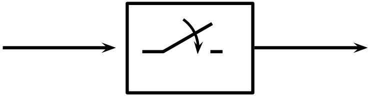
Frequency range of interpolated sinusoids



raw sampling of sinusoidal signals

Raw sampling of a sinusoid

$x(t) = e^{j2\pi f_0 t}$



$x[n] = e^{j\omega_0 n}$

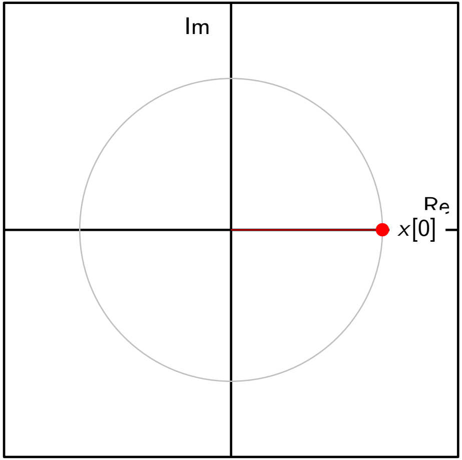
$F_s = 1/T_s$

$$x[n] = x(nT_s) = e^{j2\pi(f_0/F_s)n}$$

$$\omega_0 = 2\pi \frac{f_0}{F_s}$$

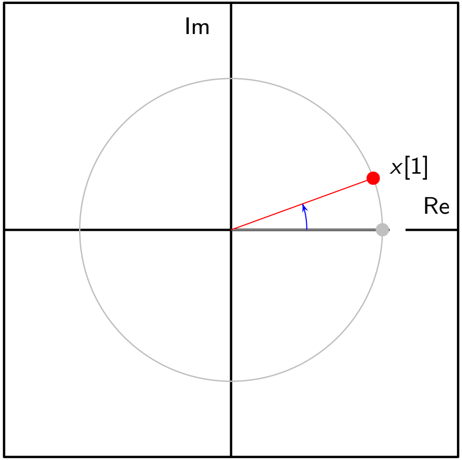
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j\omega_0 n}$$



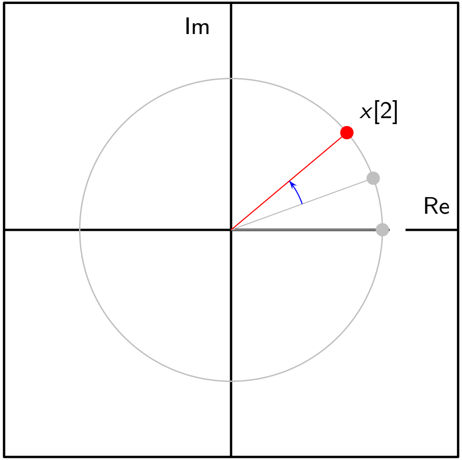
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j\omega_0 n}$$



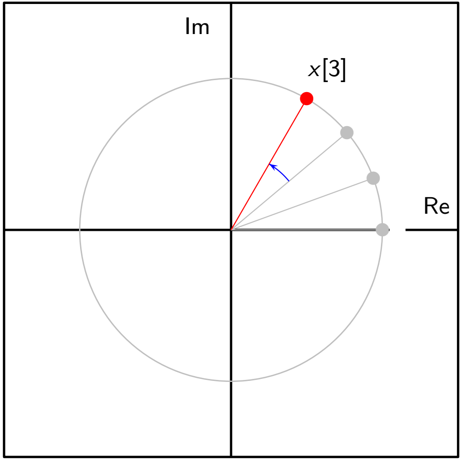
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j\omega_0 n}$$



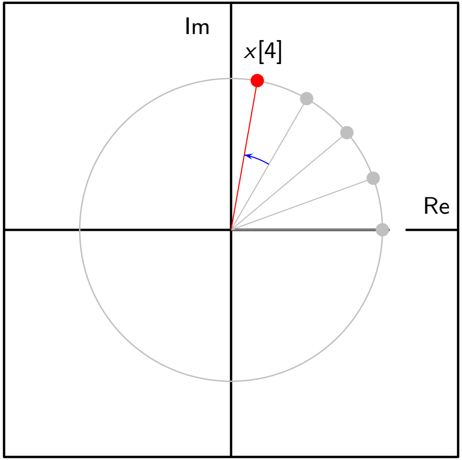
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j\omega_0 n}$$



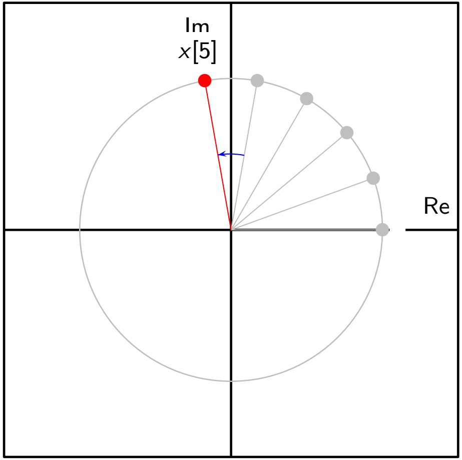
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j\omega_0 n}$$



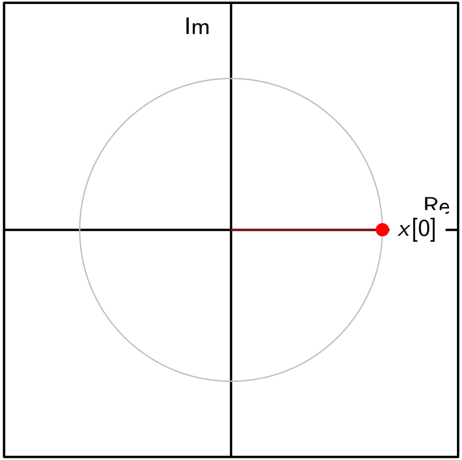
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j\omega_0 n}$$



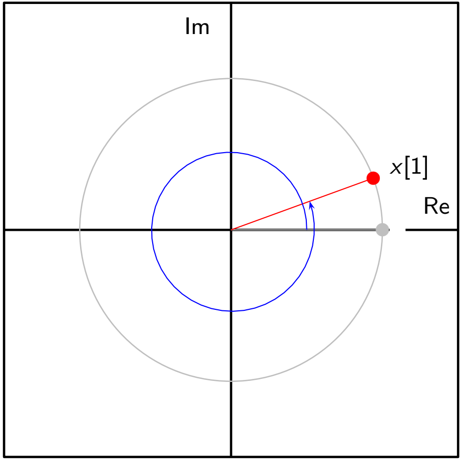
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j(\omega_0 + 2\pi)n}$$



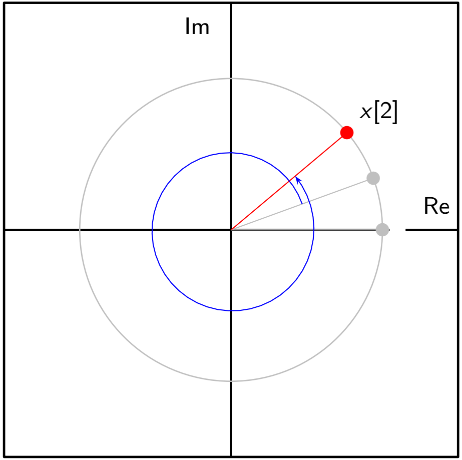
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j(\omega_0+2\pi)n}$$



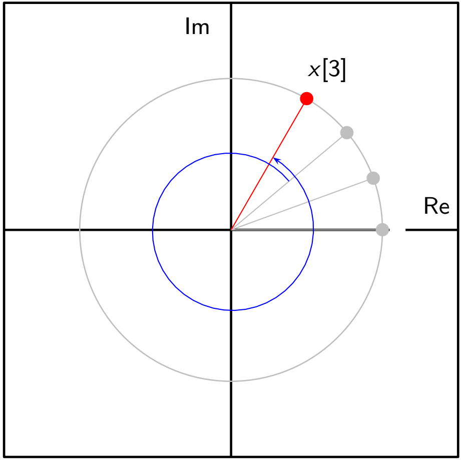
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j(\omega_0 + 2\pi)n}$$



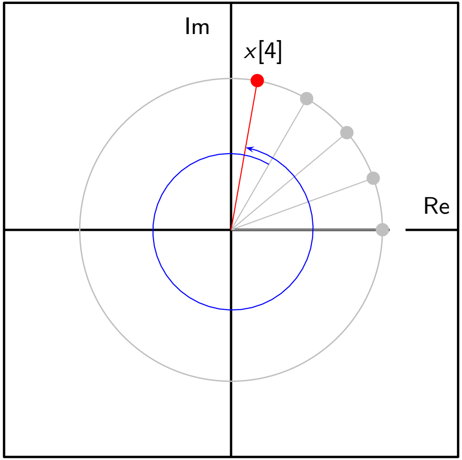
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j(\omega_0 + 2\pi)n}$$



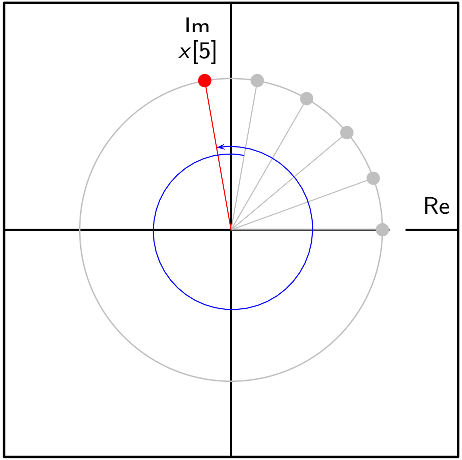
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j(\omega_0 + 2\pi)n}$$



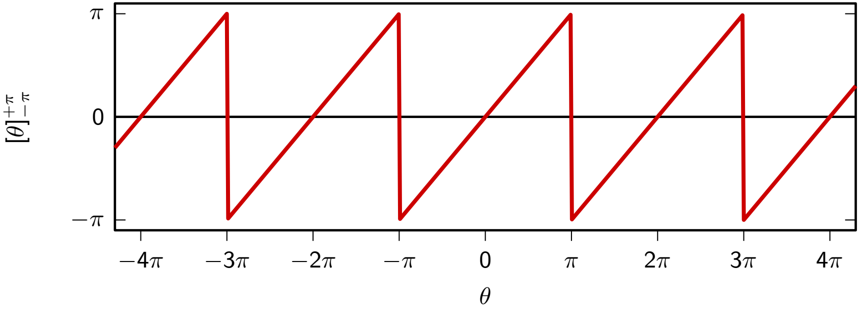
Reminder: discrete-time oscillations have a max speed

$$x[n] = e^{j(\omega_0 + 2\pi)n}$$



The phase can always be “wrapped”

$$e^{j\theta} = e^{j[\theta]_{-\pi}^{+\pi}}$$



The wrapping function

$$[\theta]_{-\pi}^{+\pi} = \theta - 2\pi \left\lfloor \frac{\theta}{2\pi} + \frac{1}{2} \right\rfloor$$

- $\lfloor x + 1/2 \rfloor$ is the integer closest to x;

  $2\pi\lfloor \theta/(2\pi) + 1/2 \rfloor$ is the multiple of 2π closest to θ

- to compute $[\theta]_{-\pi}^{+\pi}$ algorithmically:
  - if $\theta > \pi$, keep subtracting 2π from θ until the result is in $[-\pi, \pi]$
  - is $\theta < -\pi$, keep adding 2π to θ until the result is in $[-\pi, \pi]$

- example: $[18\pi/5]_{-\pi}^{+\pi} = -2\pi/5$
  1. $18\pi/5 - 2\pi = 8\pi/5 > \pi$
  2. $8\pi/5 - 2\pi = -2\pi/5 \in [-\pi, \pi]$

The wrapping function: properties

- general wrapping formula: $[x]_{-a}^{+a} = x - 2a\lfloor x/(2a) + 1/2 \rfloor$
- for any $k \in \mathbb{Z}$, $[x + 2ka]_{-a}^{+a} = [x]_{-a}^{+a}$
- for any $c \in \mathbb{R}^+$

$$\begin{aligned}
[cx]_{-a}^{+a} &= cx - 2a\left\lfloor \frac{cx}{2a} + \frac{1}{2} \right\rfloor \\
&= c\left(x - 2(a/c)\left\lfloor \frac{x}{2(a/c)} + \frac{1}{2} \right\rfloor\right) \\
&= c\,[x]_{-a/c}^{+a/c}
\end{aligned}$$

- corollary: $c\,[x]_{-a}^{+a} = [cx]_{-ac}^{+ac}$

Wrapping frequencies

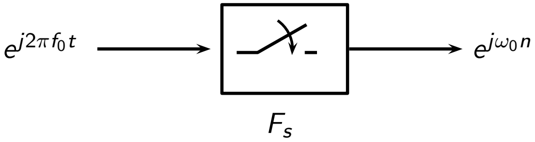
- for any $n \in \mathbb{Z}$:

$$[x]_{-a}^{+a} = \hat{x} \implies [nx]_{-a}^{+a} = [n\hat{x}]_{-a}^{+a}$$

- all discrete-time frequencies can (and should) be wrapped

$$e^{j\omega_0 n} = e^{j[\omega_0 n]_{-\pi}^{+\pi}} = e^{j[\omega_0]_{-\pi}^{+\pi} n}$$

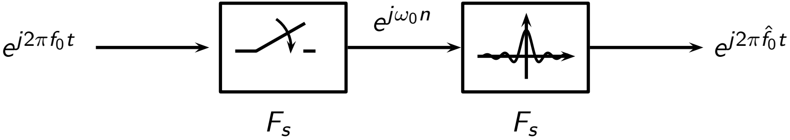
Sinusoidal raw sampling



$$\omega_0 = \left[2\pi \frac{f_0}{F_s}\right]_{-\pi}^{+\pi}$$

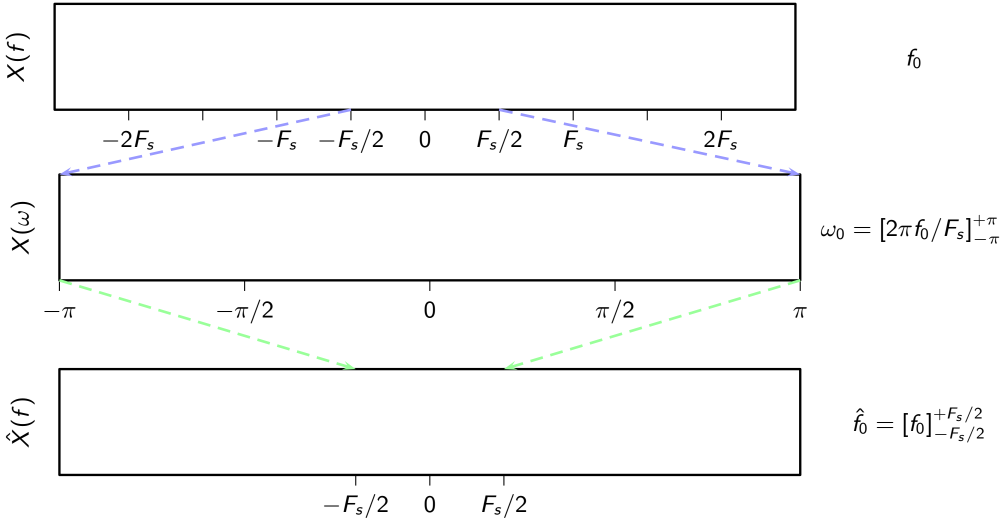
$$= 2\pi \left[\frac{f_0}{F_s}\right]_{-1/2}^{+1/2}$$

Sinusoidal raw sampling and sinc interpolation

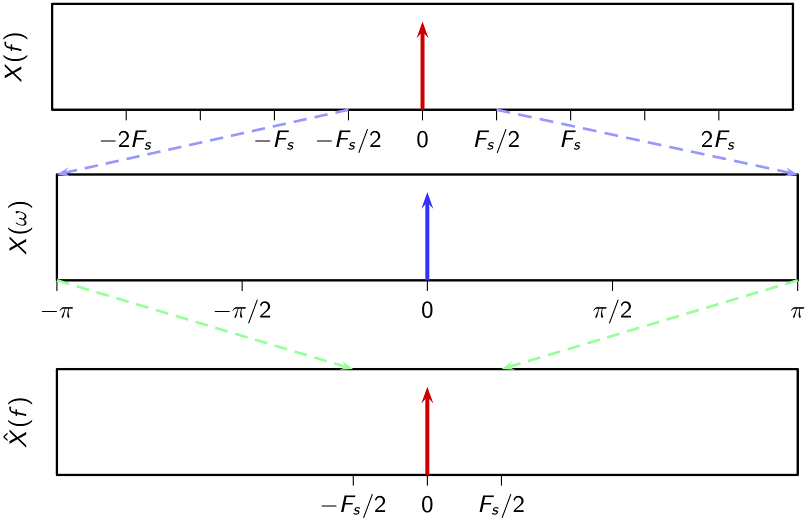


$$
\begin{aligned}
\hat{f}_0 &= \frac{\omega_0}{2\pi} F_s \\
&= F_s \left[\frac{f_0}{F_s} \right]_{-1/2}^{+1/2} \\
&= [f_0]_{-F_s/2}^{+F_s/2}
\end{aligned}
$$

Sinusoidal aliasing: increasing the frequency



Sinusoidal aliasing: increasing the frequency

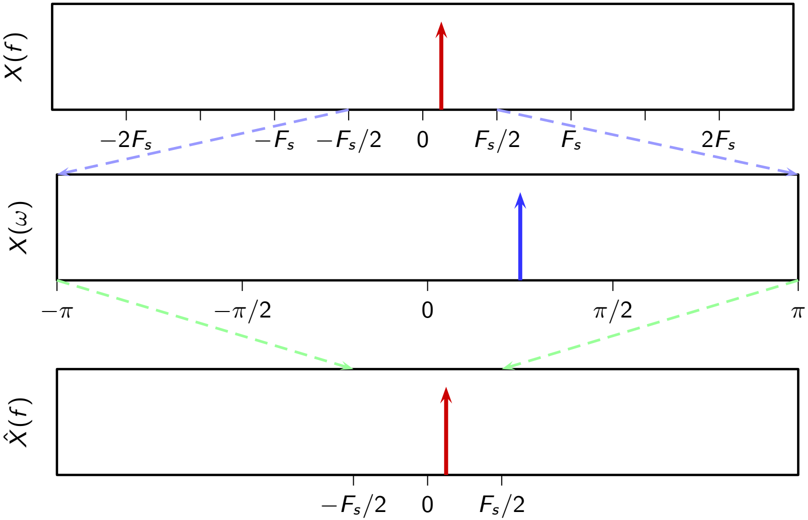


f_0

$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

Sinusoidal aliasing: increasing the frequency

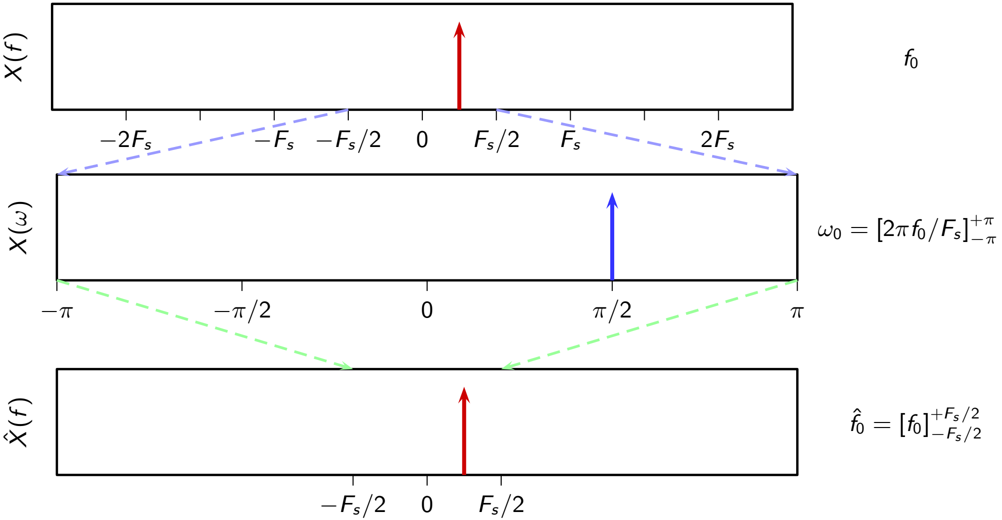


f_0

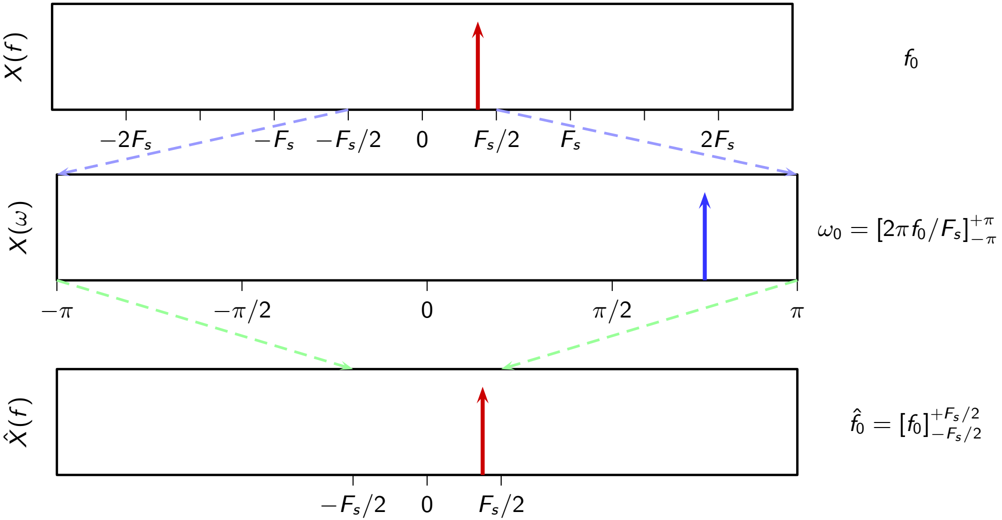
$\omega_0 = [2\pi f_0 / F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

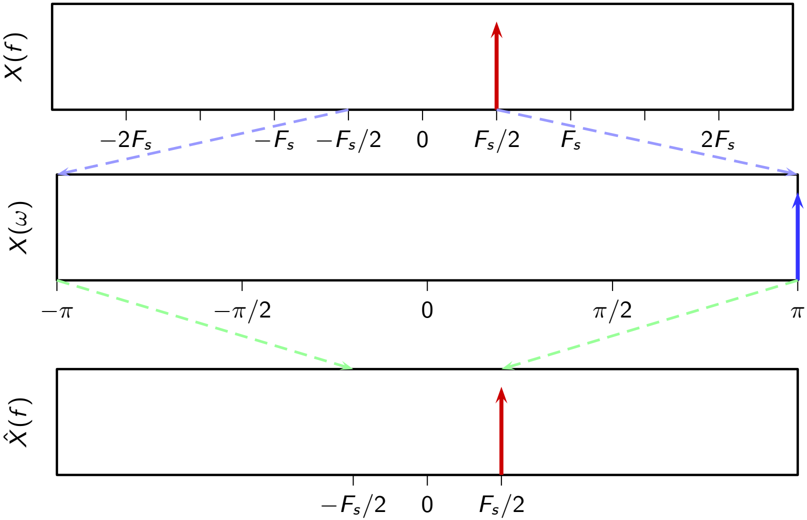
Sinusoidal aliasing: increasing the frequency



Sinusoidal aliasing: increasing the frequency



Sinusoidal aliasing: increasing the frequency

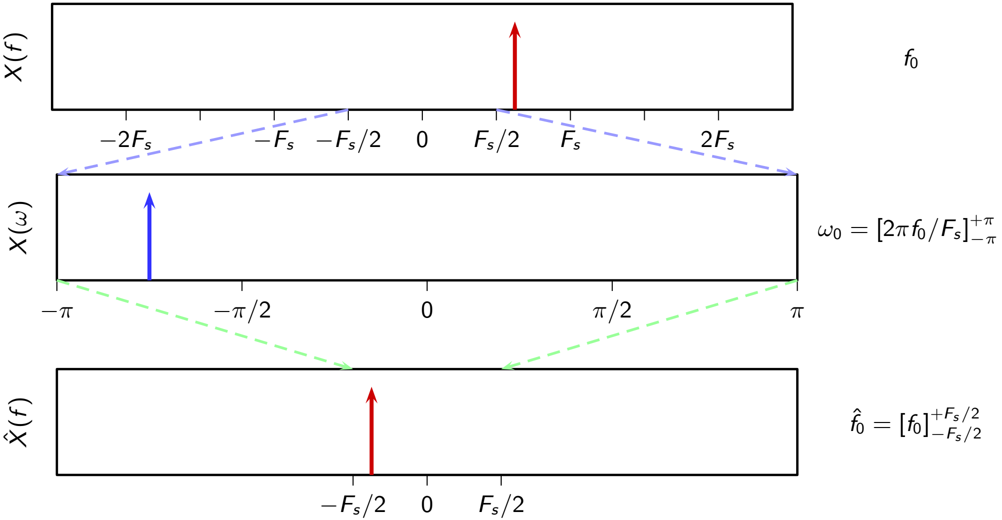


f_0

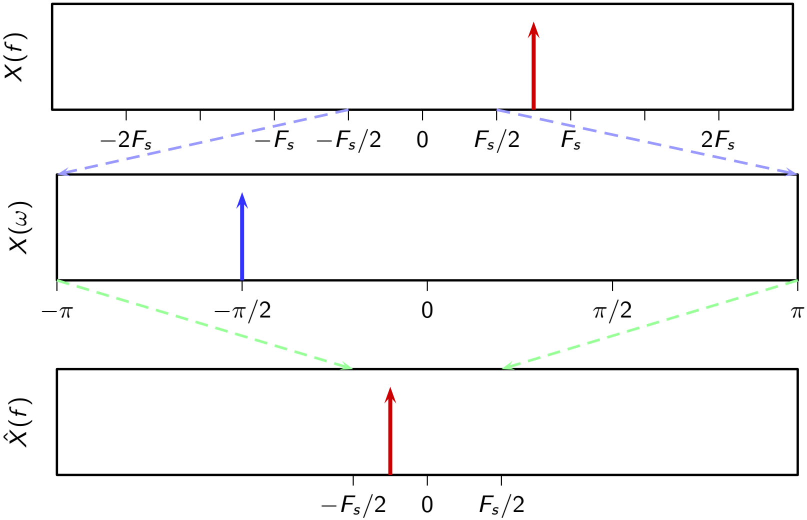
$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

Sinusoidal aliasing: increasing the frequency



Sinusoidal aliasing: increasing the frequency

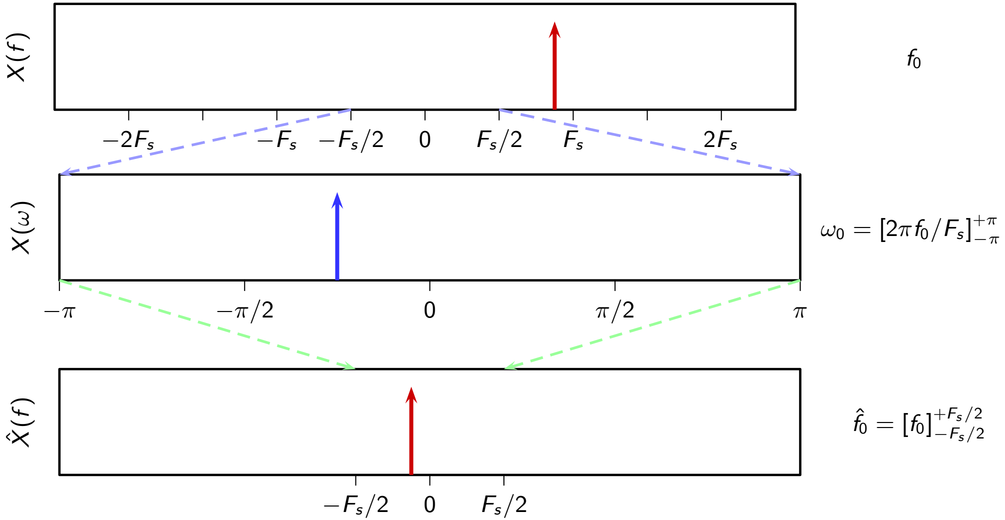


f_0

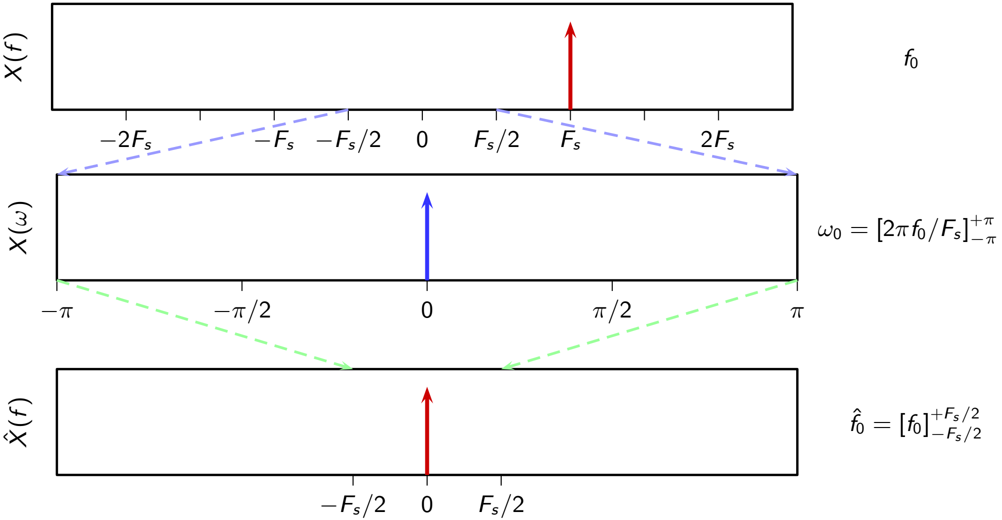
$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

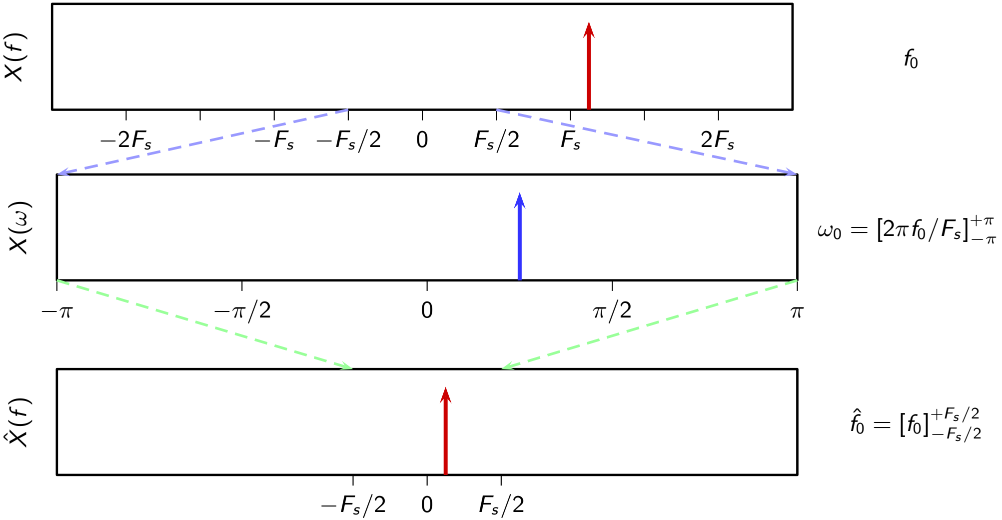
Sinusoidal aliasing: increasing the frequency



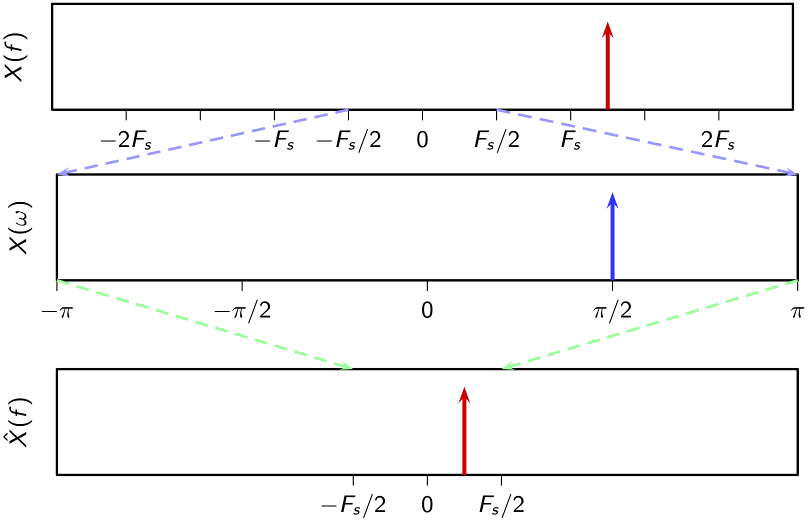
Sinusoidal aliasing: increasing the frequency



Sinusoidal aliasing: increasing the frequency



Sinusoidal aliasing: increasing the frequency

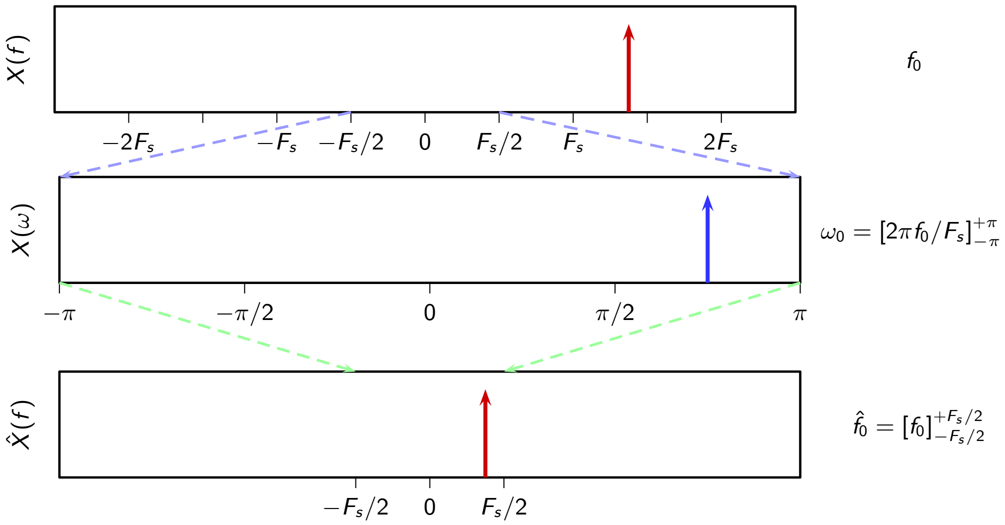


f_0

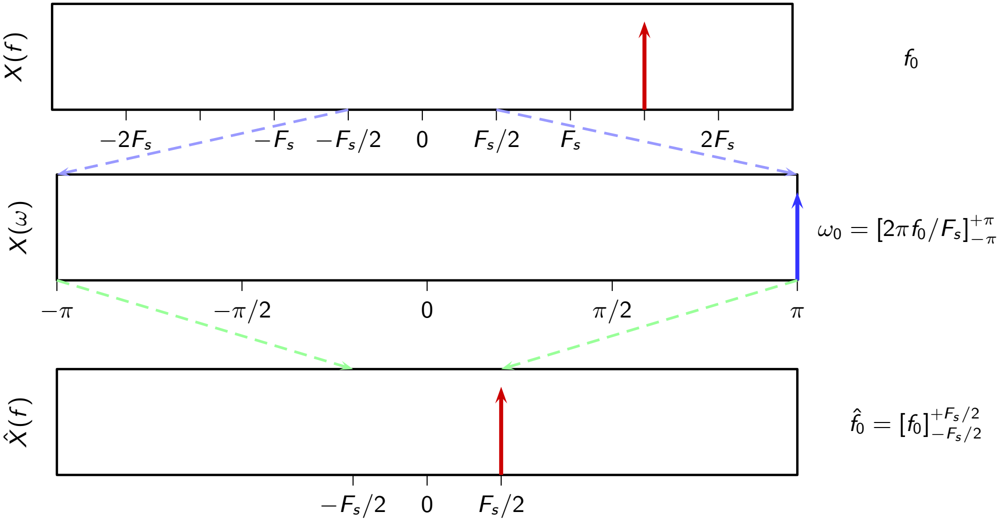
$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

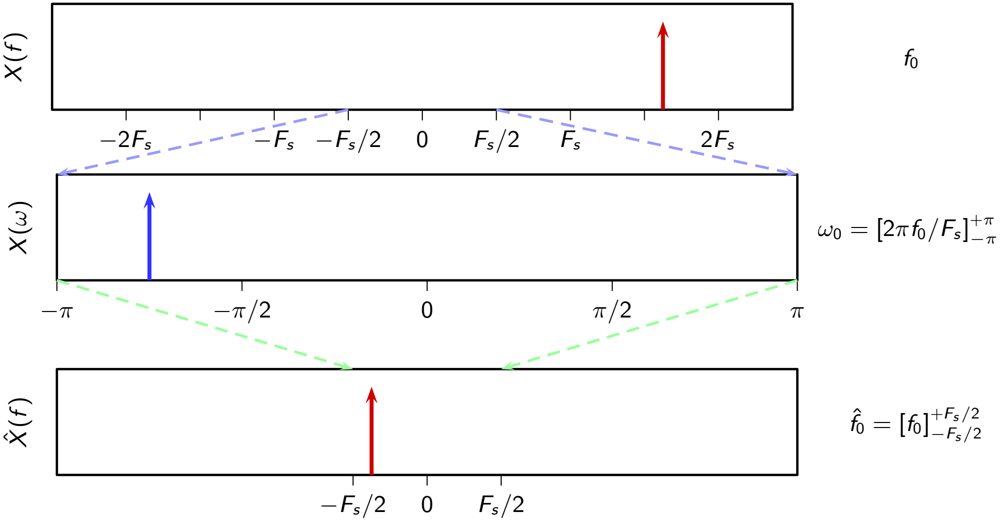
Sinusoidal aliasing: increasing the frequency



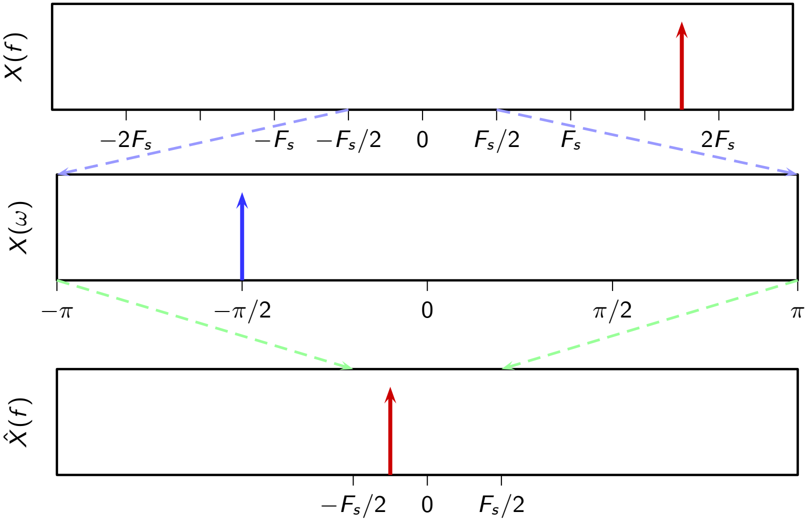
Sinusoidal aliasing: increasing the frequency



Sinusoidal aliasing: increasing the frequency



Sinusoidal aliasing: increasing the frequency

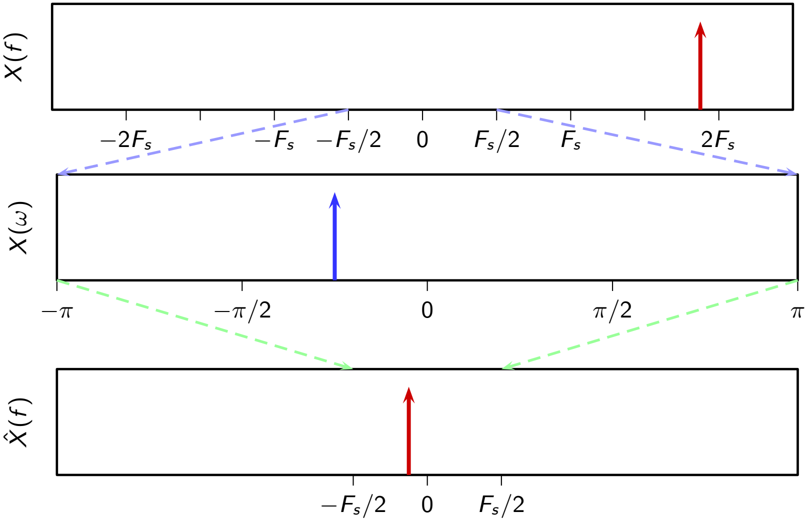


f_0

$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

Sinusoidal aliasing: increasing the frequency

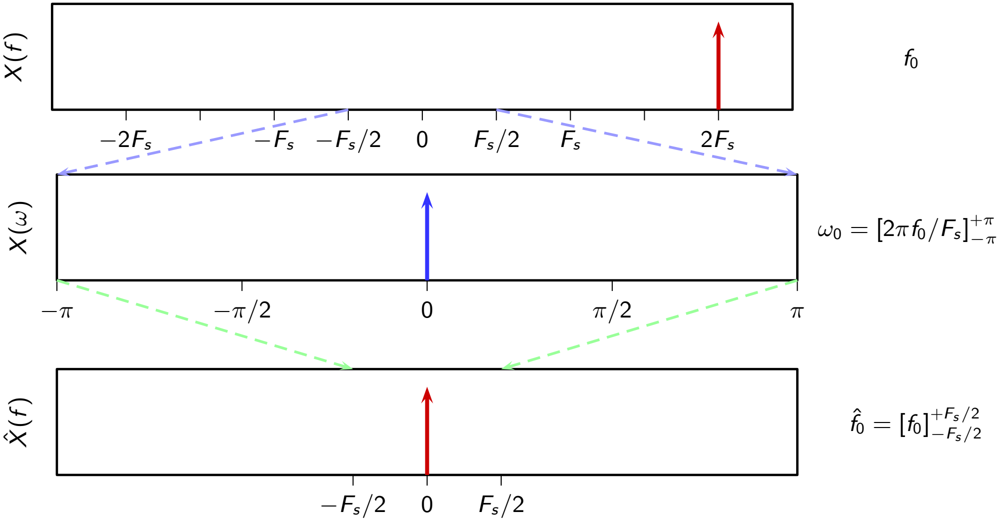


f_0

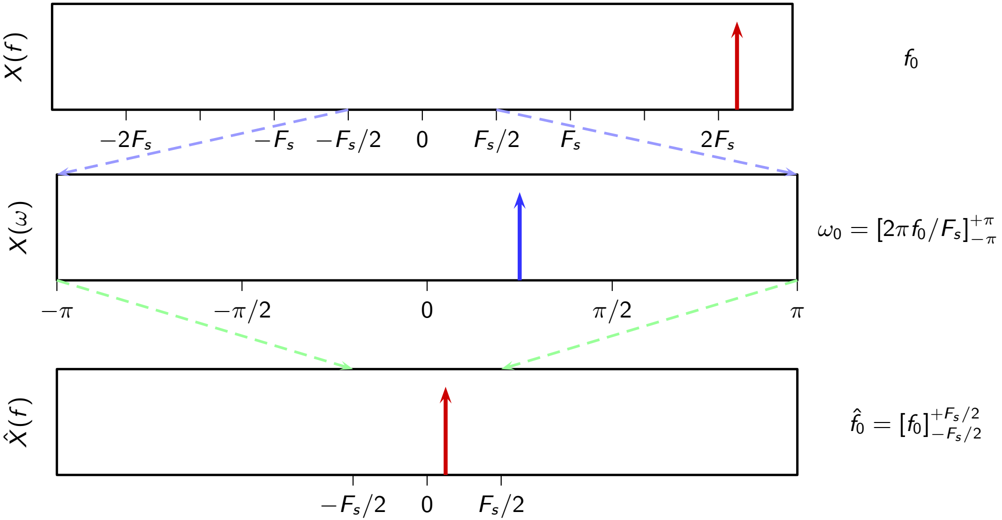
$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

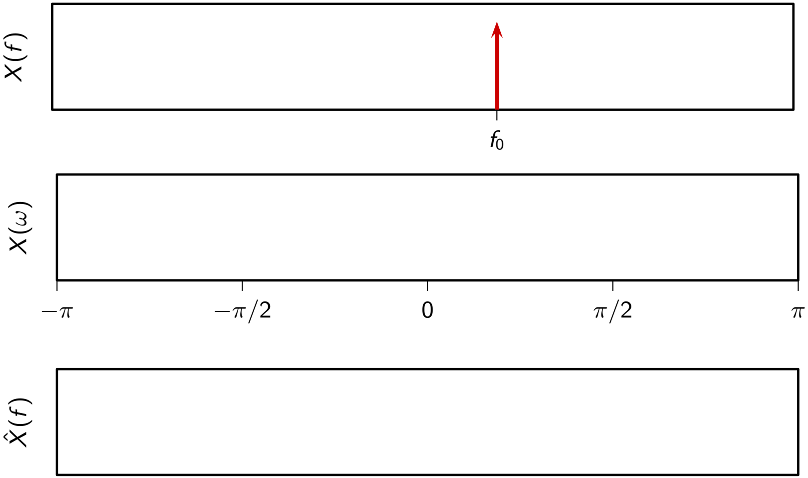
Sinusoidal aliasing: increasing the frequency



Sinusoidal aliasing: increasing the frequency



Sinusoidal aliasing: decreasing the sampling rate

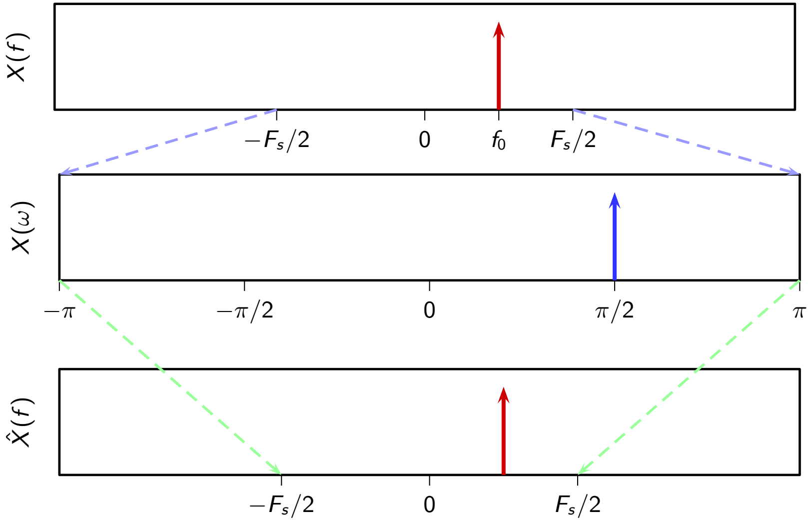


f_0

$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

Sinusoidal aliasing: decreasing the sampling rate

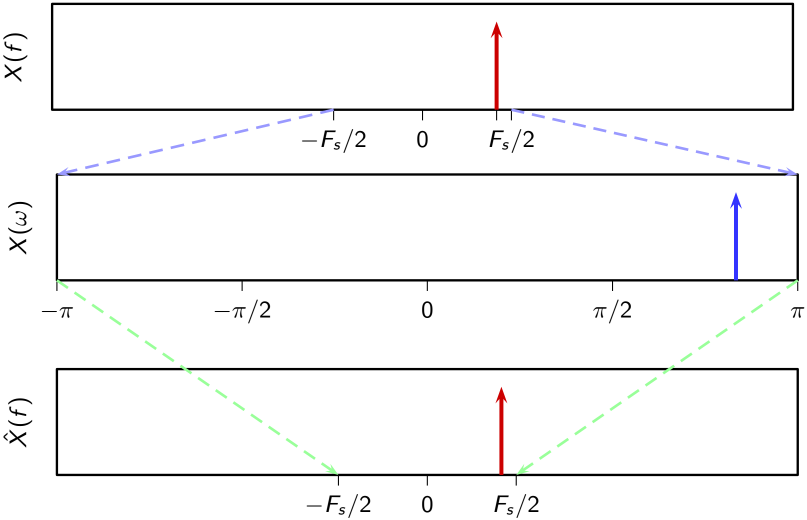


f_0

$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

Sinusoidal aliasing: decreasing the sampling rate

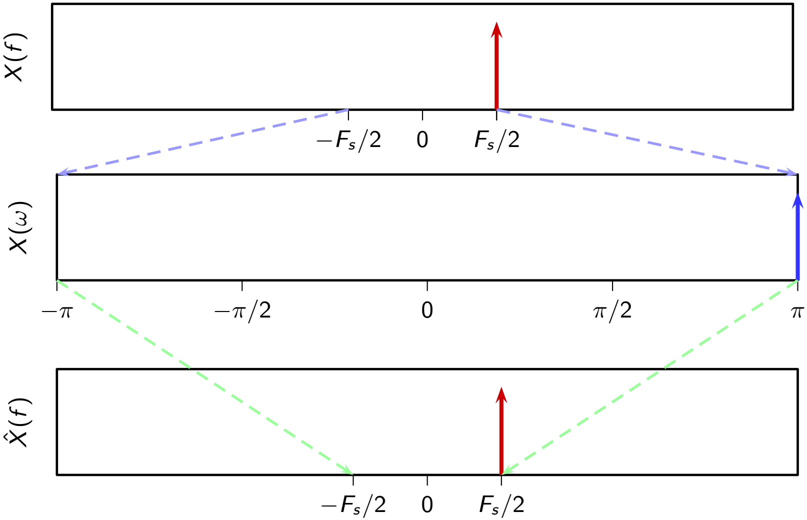


f_0

$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

Sinusoidal aliasing: decreasing the sampling rate

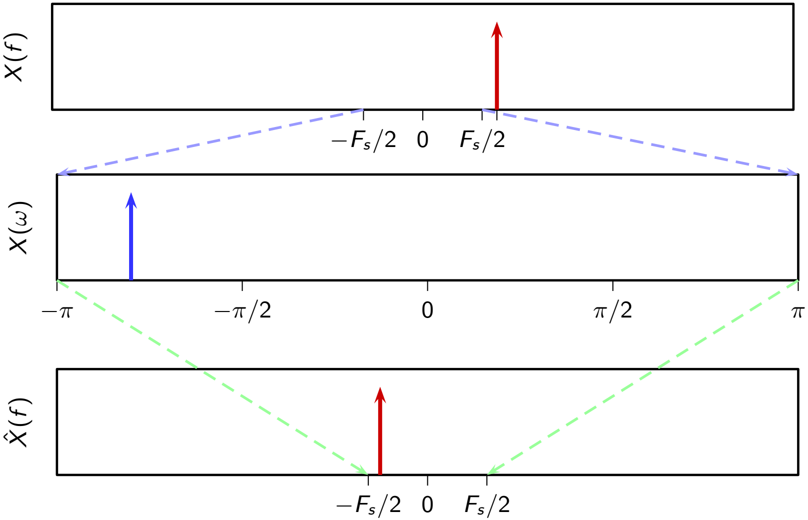


f_0

$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

Sinusoidal aliasing: decreasing the sampling rate

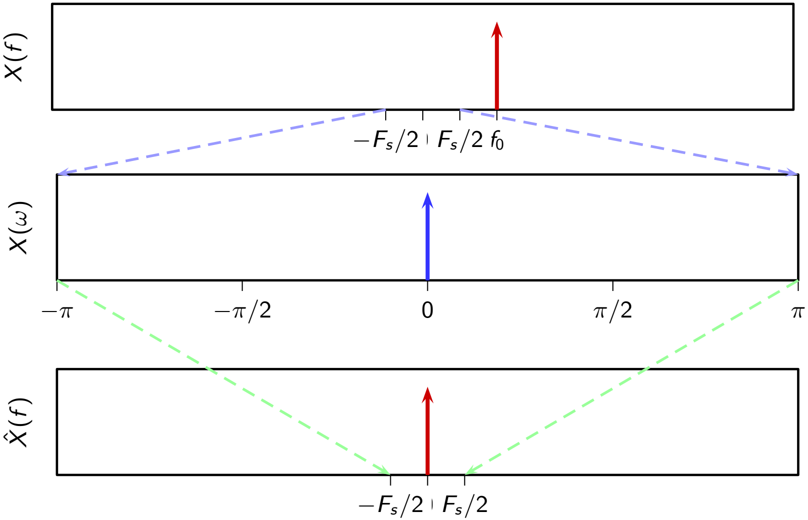


f_0

$\omega_0 = [2\pi f_0 / F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

Sinusoidal aliasing: decreasing the sampling rate

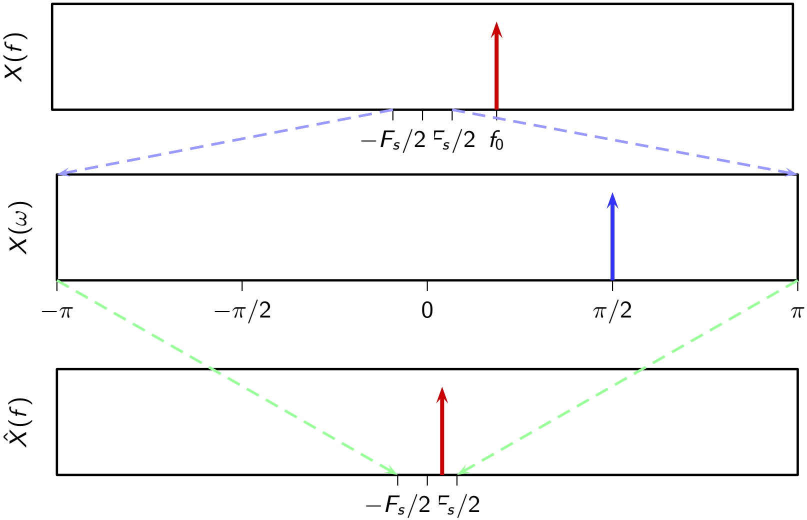


f_0

$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

Sinusoidal aliasing: decreasing the sampling rate



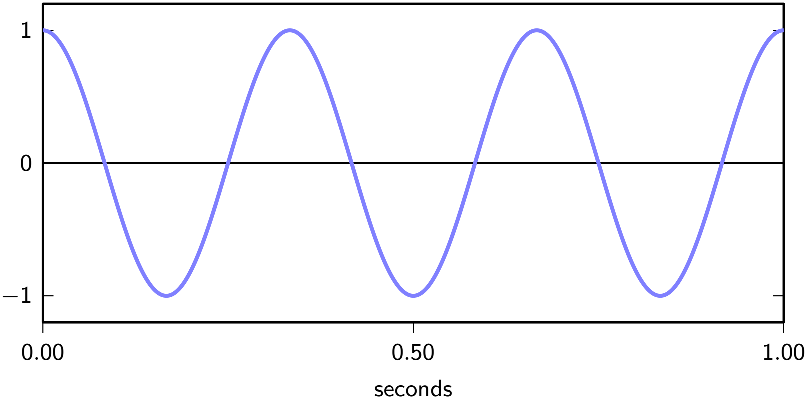
f_0

$\omega_0 = [2\pi f_0/F_s]_{-\pi}^{+\pi}$

$\hat{f}_0 = [f_0]_{-F_s/2}^{+F_s/2}$

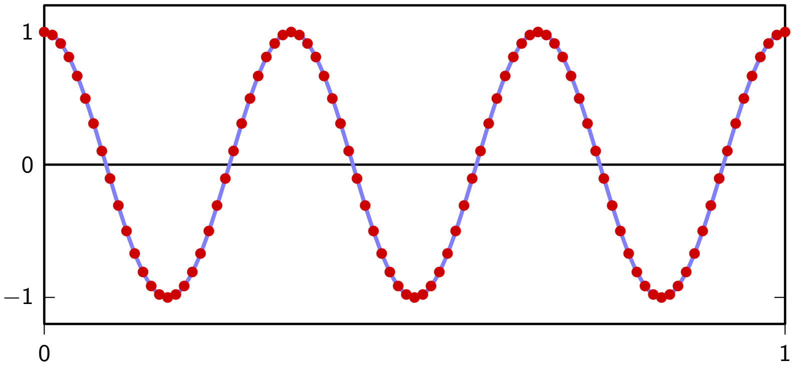
Sinusoidal aliasing in the time domain

$$x(t) = \cos(2\pi f_0 t), \qquad f_0 = 3 \text{ Hz}$$



Sinusoidal aliasing in the time domain

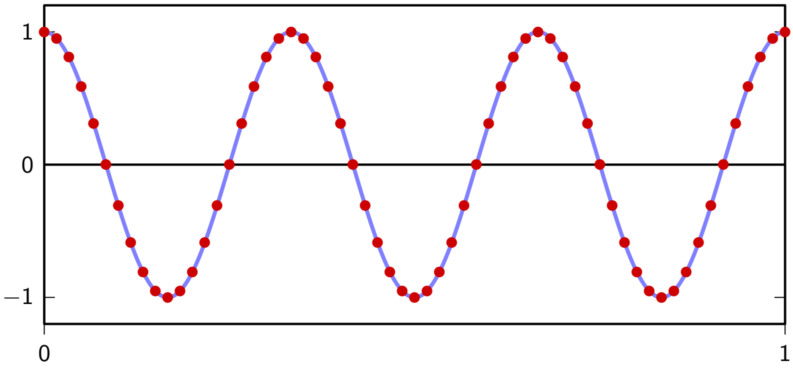
$$x(t) = \cos(2\pi f_0 t), \qquad f_0 = 3 \text{ Hz}$$



$$F_s = 90 \text{ Hz}, \quad \hat{f}_0 = [3]_{-45}^{+45} = 3 \text{ Hz}$$

Sinusoidal aliasing in the time domain

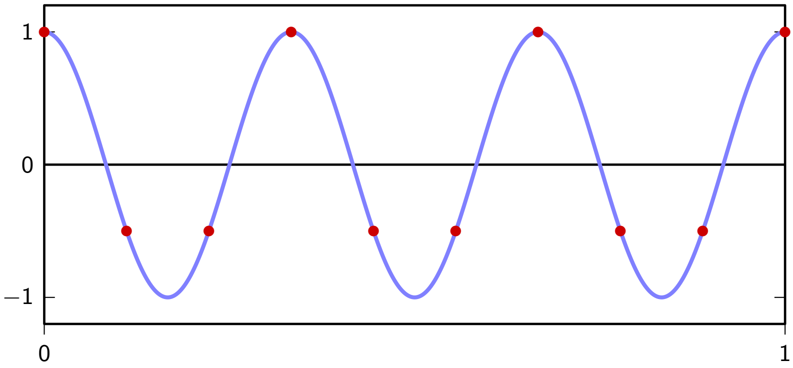
$$x(t) = \cos(2\pi f_0 t), \qquad f_0 = 3 \text{ Hz}$$



$$F_s = 60 \text{ Hz}, \quad \hat{f}_0 = [3]_{-30}^{+30} = 3 \text{ Hz}$$

Sinusoidal aliasing in the time domain

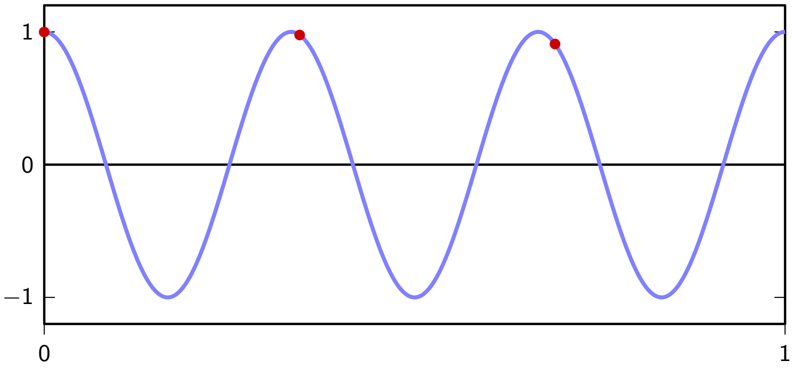
$$x(t) = \cos(2\pi f_0 t), \qquad f_0 = 3 \text{ Hz}$$



$$F_s = 9 \text{ Hz}, \quad \hat{f}_0 = [3]_{-4.5}^{+4.5} = 3 \text{ Hz}$$

Sinusoidal aliasing in the time domain

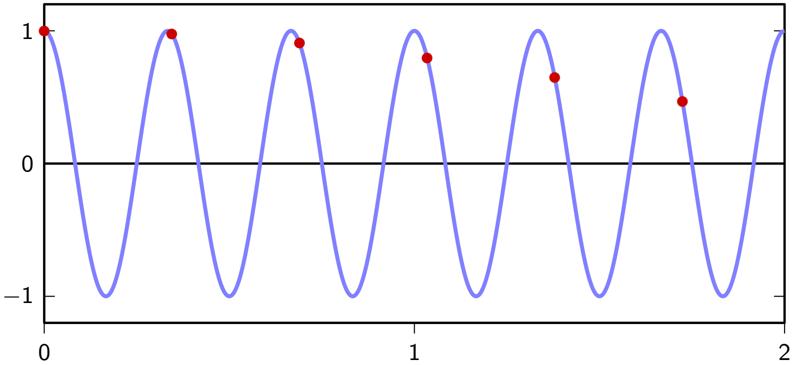
$$x(t) = \cos(2\pi f_0 t), \qquad f_0 = 3 \text{ Hz}$$



$$F_s = 2.9 \text{ Hz}, \quad \hat{f}_0 = [3]_{-1.45}^{+1.45} = 0.1 \text{ Hz}$$

Sinusoidal aliasing in the time domain

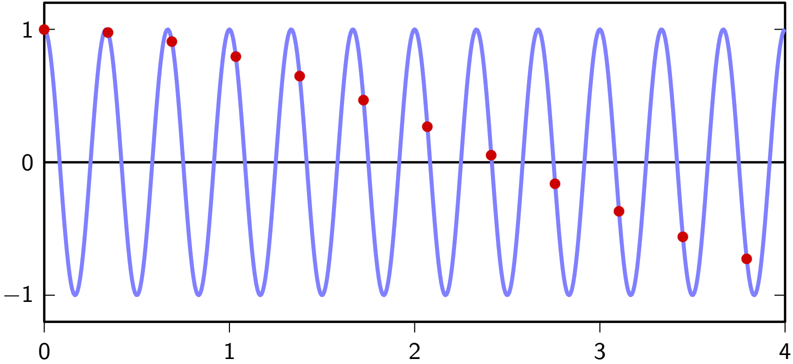
$$x(t) = \cos(2\pi f_0 t), \qquad f_0 = 3 \text{ Hz}$$



$$F_s = 2.9 \text{ Hz}, \quad \hat{f}_0 = 0.1 \text{ Hz}$$

Sinusoidal aliasing in the time domain

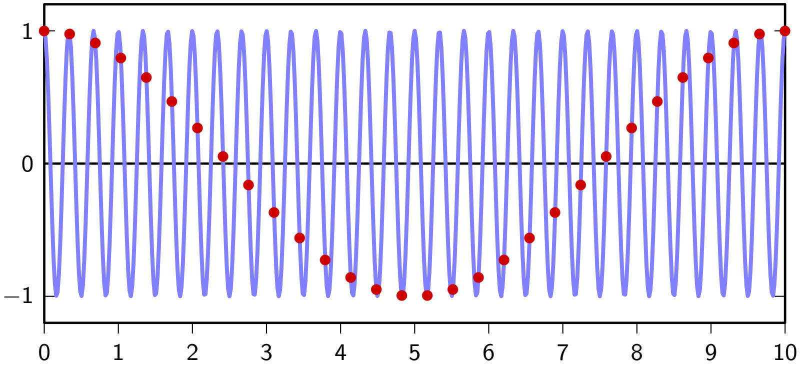
$$x(t) = \cos(2\pi f_0 t), \qquad f_0 = 3 \text{ Hz}$$



$$F_s = 2.9 \text{ Hz}, \quad \hat{f}_0 = 0.1 \text{ Hz}$$

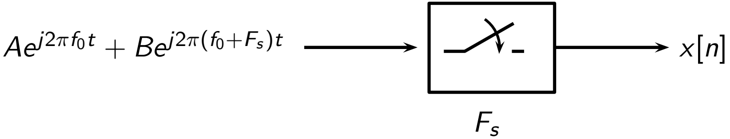
Sinusoidal aliasing in the time domain

$$x(t) = \cos(2\pi f_0 t), \qquad f_0 = 3 \text{ Hz}$$



$$F_s = 2.9 \text{ Hz}, \quad \hat{f}_0 = 0.1 \text{ Hz}$$

The key concept for general aliasing



$$x[n] = Ae^{j\omega_0 n} + Be^{j\omega_1 n}$$

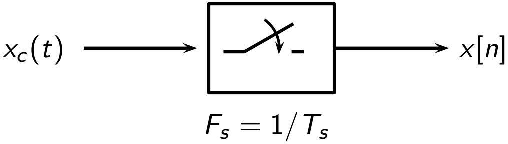
$$\omega_0 = \left[2\pi \frac{f_0}{F_s}\right]_{-\pi}^{+\pi}$$

$$\omega_1 = \left[2\pi \frac{f_0 + F_s}{F_s}\right]_{-\pi}^{+\pi} = \left[2\pi \frac{f_0}{F_s} + 2\pi\right]_{-\pi}^{+\pi} = \left[2\pi \frac{f_0}{F_s}\right]_{-\pi}^{+\pi} = \omega_0$$

$$x[n] = (A + B)e^{j\omega_0 n}$$

aliasing

Raw sampling

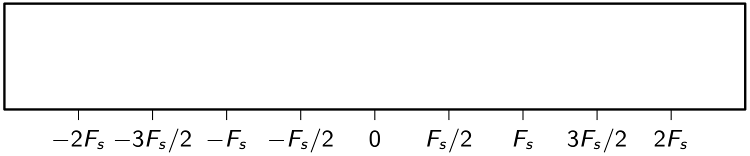


- what is the spectrum of the sampled signal?
- the input signal is composed of sinusoids at all frequencies

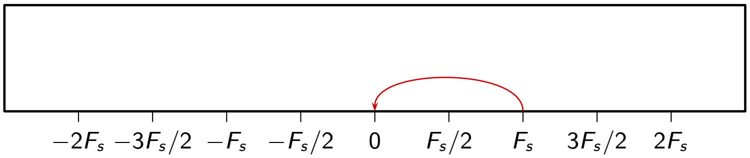
$$x_c(t) = \text{ICTFT}\{X_c(f)\}(t) = \int_{-\infty}^{\infty} X_c(f) e^{j2\pi ft} df$$

- after sampling, the spectral components at frequencies $f + kF_s$ for $k \in \mathbb{Z}$ will be lumped together

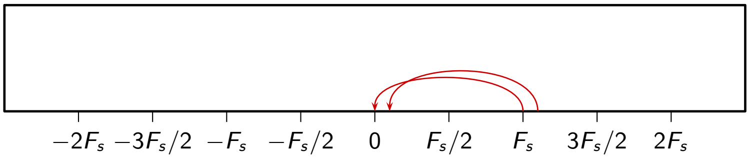
Spectrum of raw-sampled signals



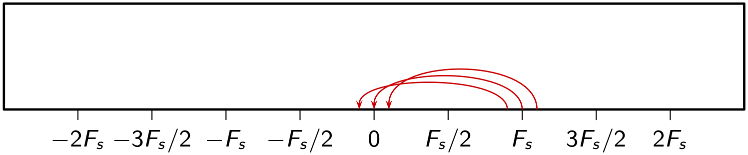
Spectrum of raw-sampled signals



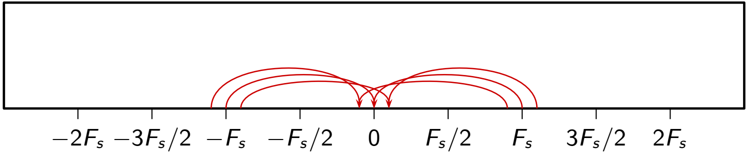
Spectrum of raw-sampled signals



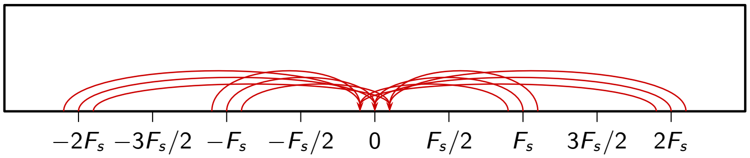
Spectrum of raw-sampled signals



Spectrum of raw-sampled signals



Spectrum of raw-sampled signals



Spectrum of raw-sampled signals (I)

start by expressing $x[n]$ as the inverse CTFT computed in $t = nT_s$

$$x[n] = x_c(nT_s) = \int_{-\infty}^{\infty} X_c(f) e^{j2\pi f\, nT_s} df$$

components F_s Hz apart will be aliased, so split the integration interval

$$= \sum_{k=-\infty}^{\infty} \int_{(k-1/2)F_s}^{(k+1/2)F_s} X_c(f) e^{j2\pi f\, nT_s} df$$

change of variable: $f = \varphi + kF_s$

$$= \sum_{k=-\infty}^{\infty} \int_{-F_s/2}^{F_s/2} X_c(\varphi + kF_s) e^{j(2\pi/F_s)(\varphi + kF_s)n} d\varphi$$

Spectrum of raw-sampled signals (II)

$$x[n] = \int_{-F_s/2}^{F_s/2} \sum_{k=-\infty}^{\infty} X_c(\varphi + kF_s) e^{j(2\pi/F_s)\varphi n} d\varphi$$

define the F_s-periodization of the CT spectrum $\tilde{X}_c(f) = \sum_{k=-\infty}^{\infty} X_c(f + kF_s)$

$$= \int_{-F_s/2}^{F_s/2} \tilde{X}_c(\varphi) e^{j(2\pi/F_s)\varphi n} d\varphi$$

change of variable: $\varphi = \frac{F_s}{2\pi}\omega$

$$= \frac{F_s}{2\pi} \int_{-\pi}^{\pi} \tilde{X}_c\left(\frac{F_s}{2\pi}\omega\right) e^{j\omega n} d\omega$$

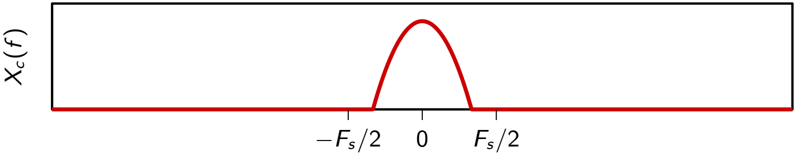
$$= \text{IDTFT}\left\{F_s \tilde{X}_c\left(\frac{F_s}{2\pi}\omega\right)\right\}$$

Spectrum of raw-sampled signals (III)

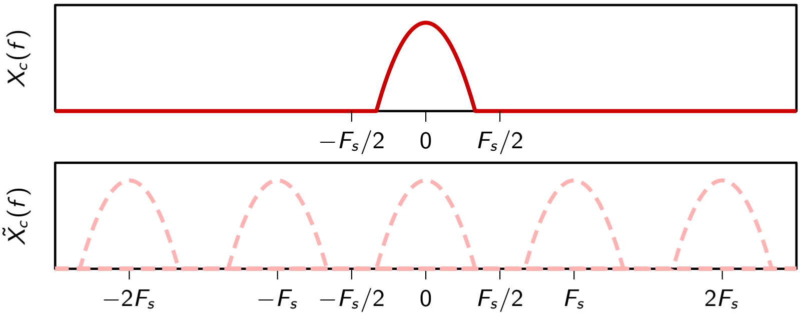
- periodize $X(f)$ with period F_s
- rescale frequency axis so $[-F_s/2, F_s/2] \to [-\pi, \pi]$

$$X(\omega) = F_s \tilde{X}_c\left(\frac{\omega}{2\pi} F_s\right) = F_s \sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega}{2\pi} F_s - kF_s\right)$$

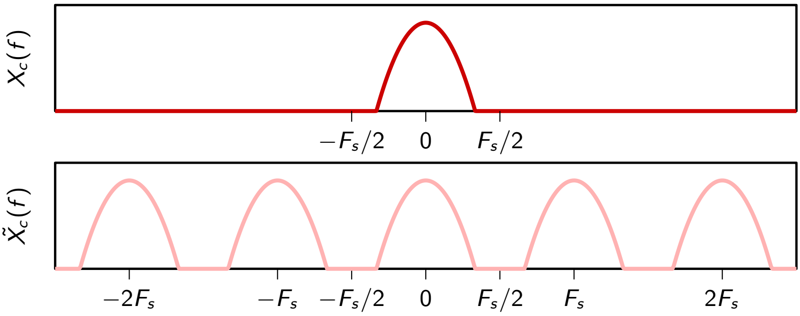
Example: signal bandlimited to f_0 and $F_s > 2f_0$



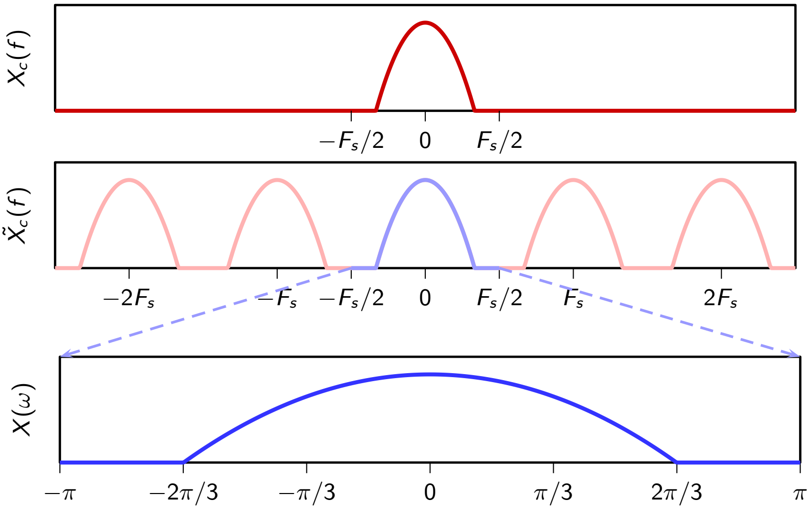
Example: signal bandlimited to f_0 and $F_s > 2f_0$



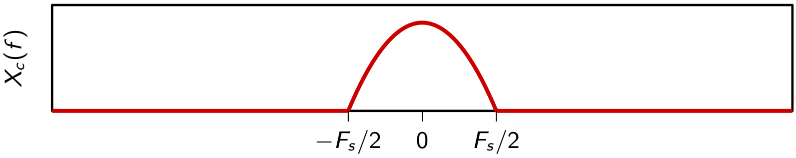
Example: signal bandlimited to f_0 and $F_s > 2f_0$



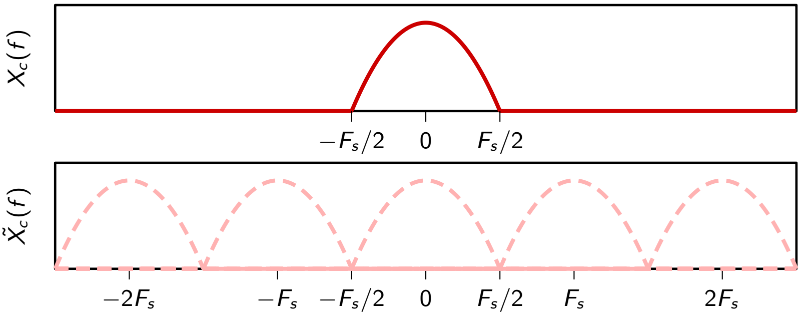
Example: signal bandlimited to f_0 and $F_s > 2f_0$



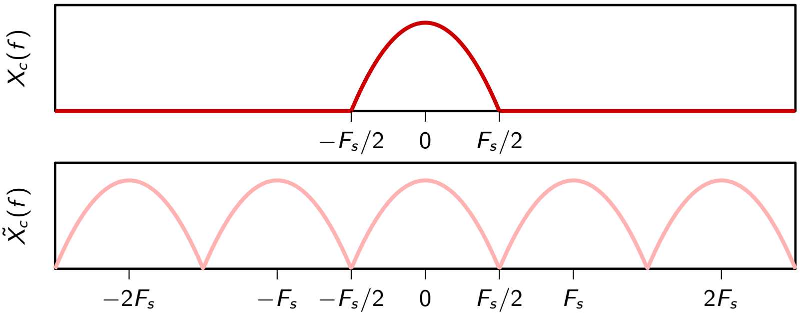
Example: signal bandlimited to f_0 and $F_s = 2f_0$



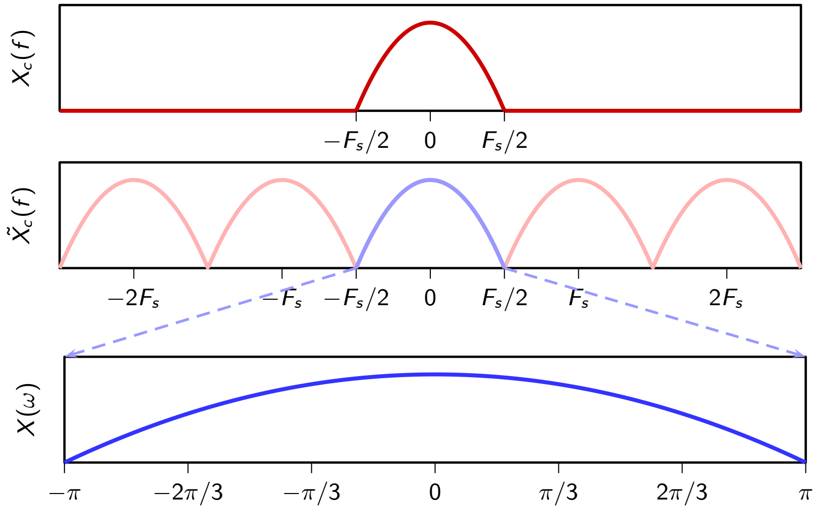
Example: signal bandlimited to f_0 and $F_s = 2f_0$



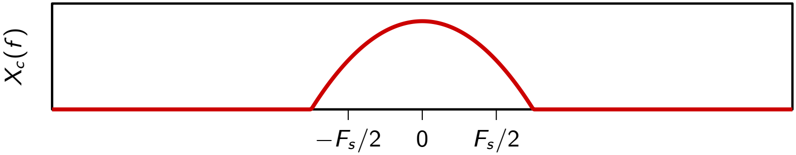
Example: signal bandlimited to f_0 and $F_s = 2f_0$



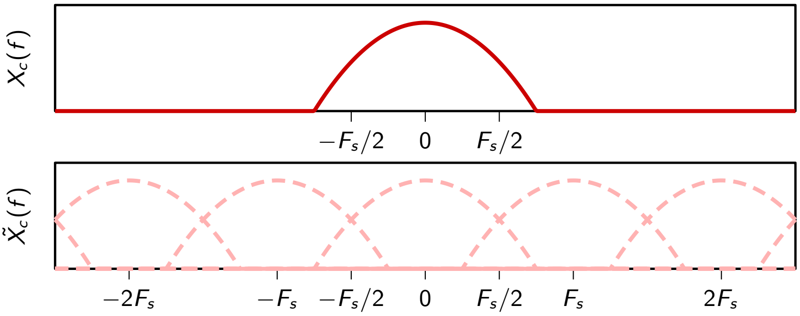
Example: signal bandlimited to f_0 and $F_s = 2f_0$



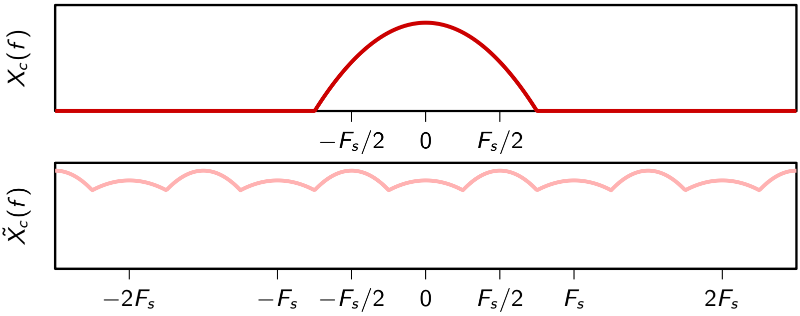
Example: signal bandlimited to f_0 and $F_s < 2f_0$



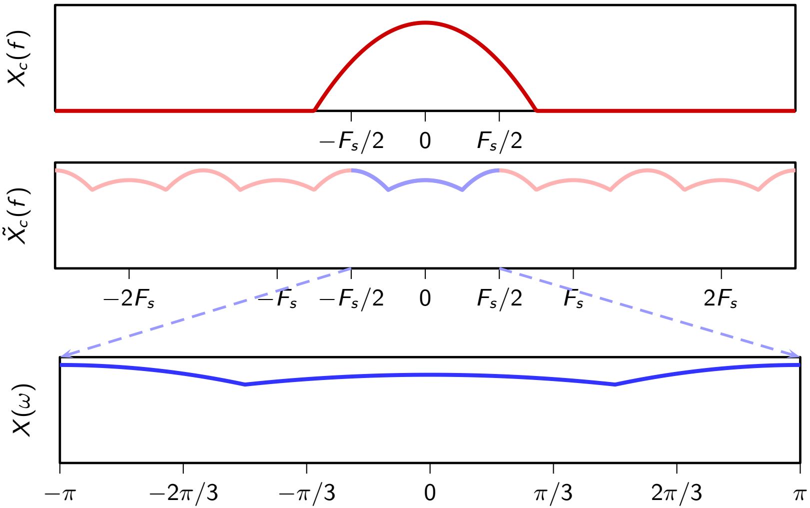
Example: signal bandlimited to f_0 and $F_s < 2f_0$



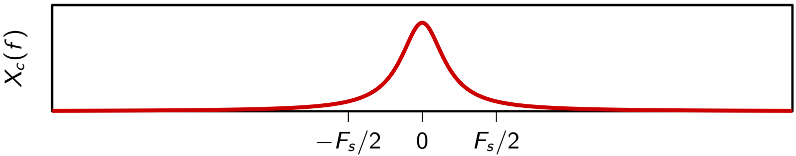
Example: signal bandlimited to f_0 and $F_s < 2f_0$



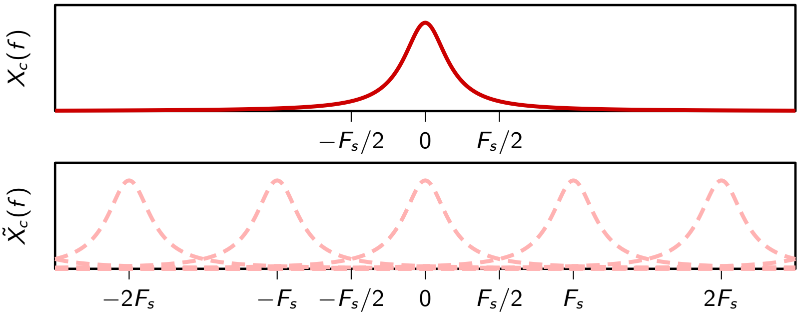
Example: signal bandlimited to f_0 and $F_s < 2f_0$



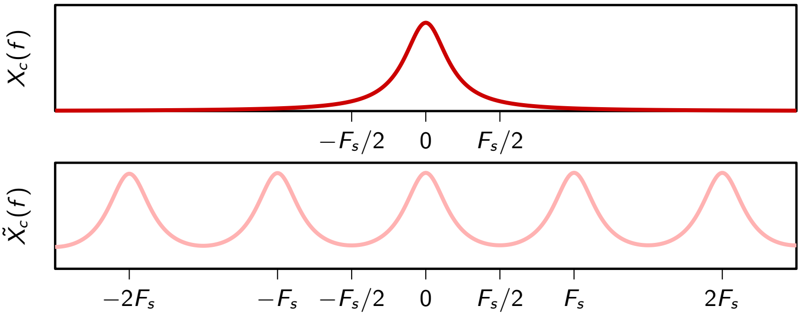
Example: non-bandlimited signal



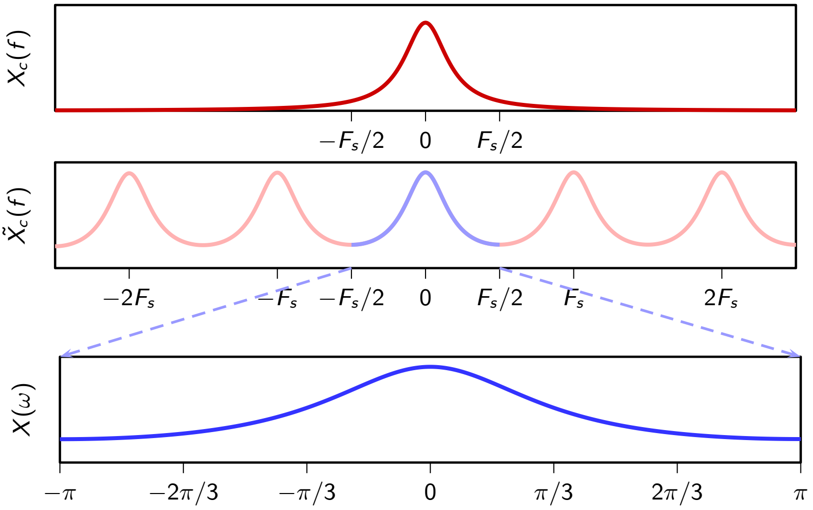
Example: non-bandlimited signal



Example: non-bandlimited signal



Example: non-bandlimited signal

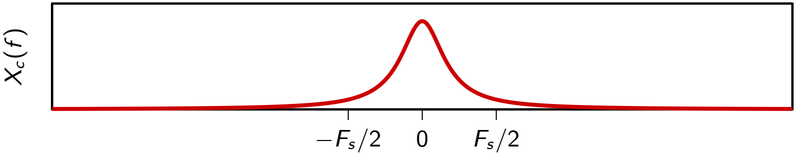


Sampling strategies

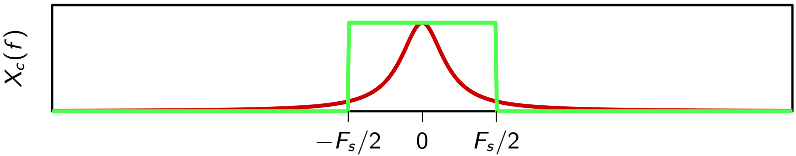
given a raw sampler at frequency F_s

- if the signal is F_s-bandlimited, no problem
- if the signal is not F_s-bandlimited, two choices:
  - apply a continuous-time (analog) lowpass filter with cutoff $F_s/$ before raw sampling, that is, implement an approximation of sinc sampling
  - accept the distortion due to aliasing
- aliasing errors are unpredictable and very disrupting, so always use an analog lowpass
- antialias bandlimiting minimizes the energy of the error

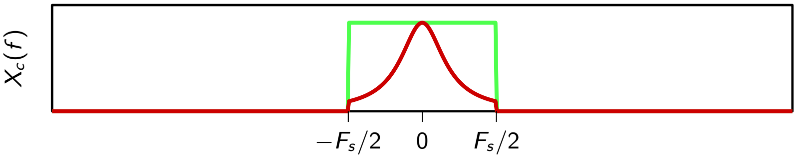
Sampling with antialiasing filter



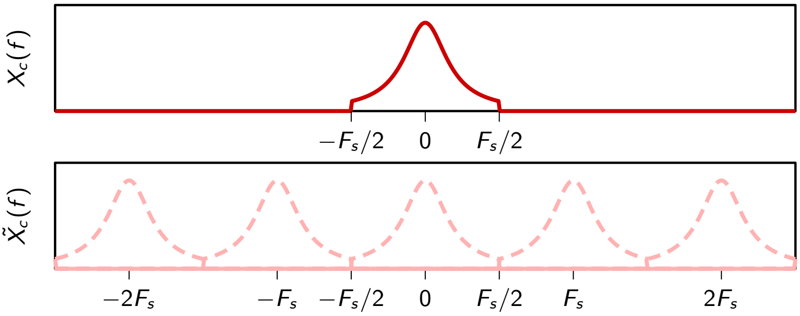
Sampling with antialiasing filter



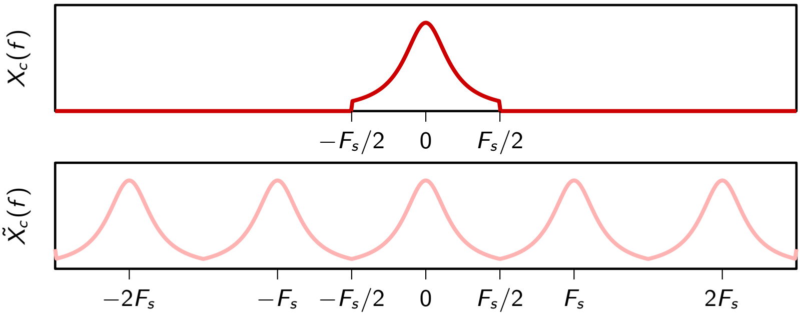
Sampling with antialiasing filter



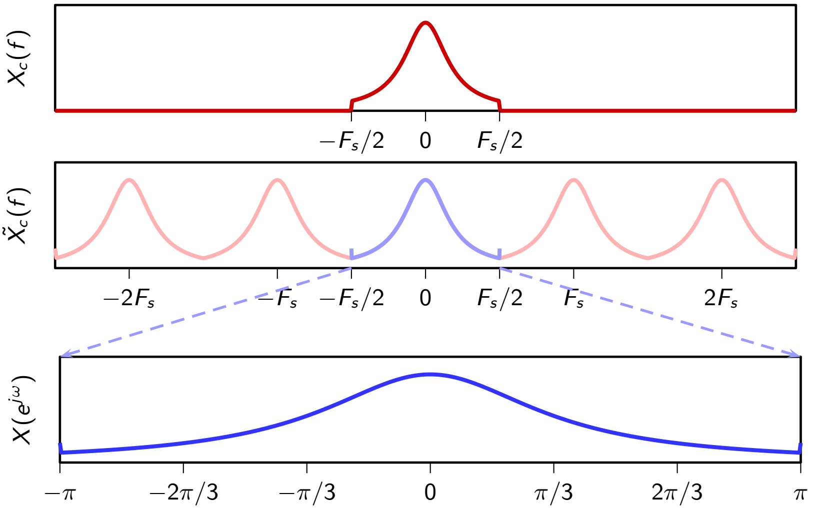
Sampling with antialiasing filter



Sampling with antialiasing filter

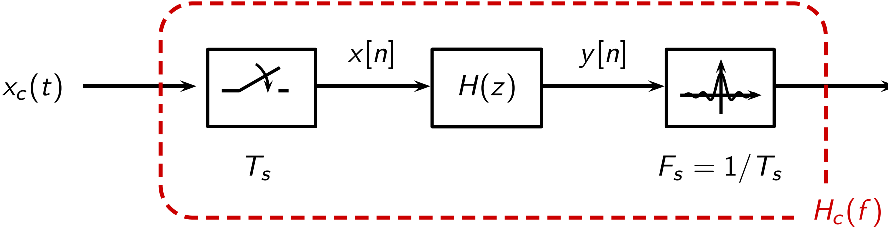


Sampling with antialiasing filter



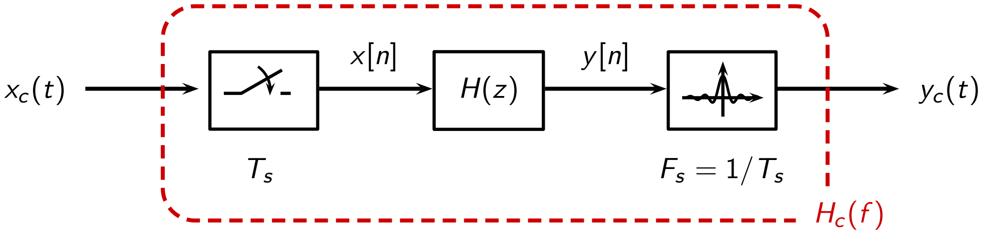
discrete-time processing of analog signals

Equivalent analog response: basic setup



what is the equivalent analog frequency response $H_c(f)$?

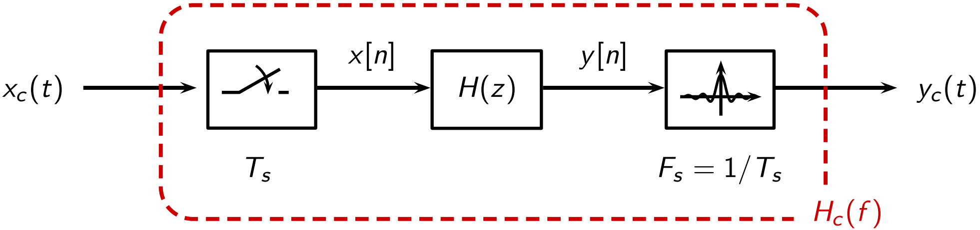
Equivalent analog response: basic setup



assume $x_c(t)$ is Fs-BL and $T_s = 1/F_s$

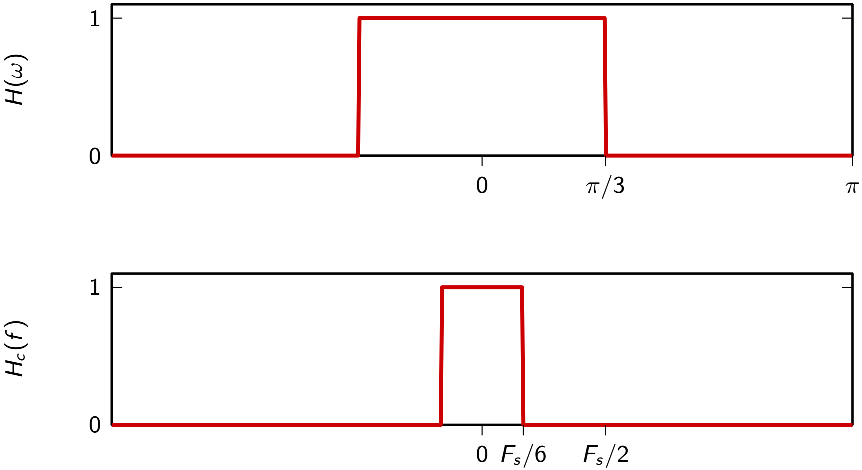
- $X(\omega) = F_s X_c\left(F_s \frac{\omega}{2\pi}\right)$
- $Y(\omega) = H(\omega)X(\omega)$
- $Y_c(f) = \frac{1}{F_s} Y(2\pi\frac{f}{F_s}) = H\left(2\pi\frac{f}{F_s}\right) X_c(f)$

Equivalent analog response: basic setup

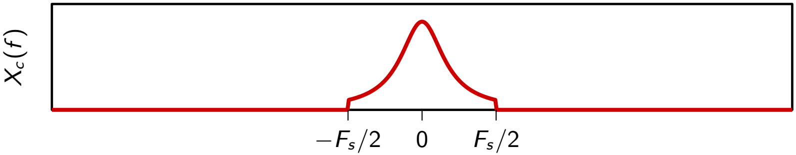


$$H_c(f) = H\left(2\pi \frac{f}{F_s}\right)$$

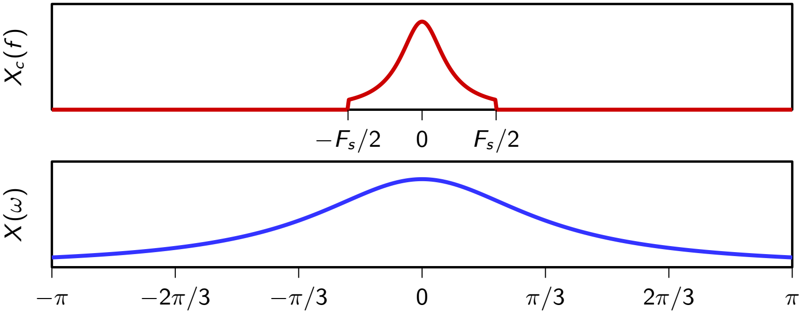
Equivalent analog response



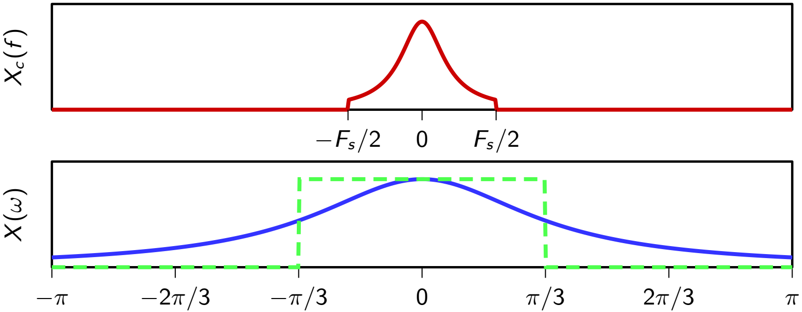
DT processing of CT signals



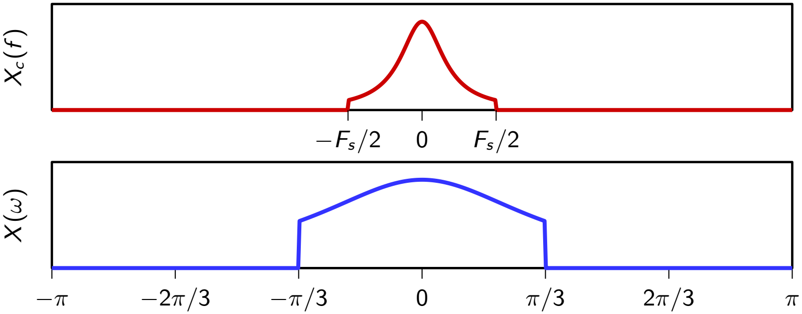
DT processing of CT signals



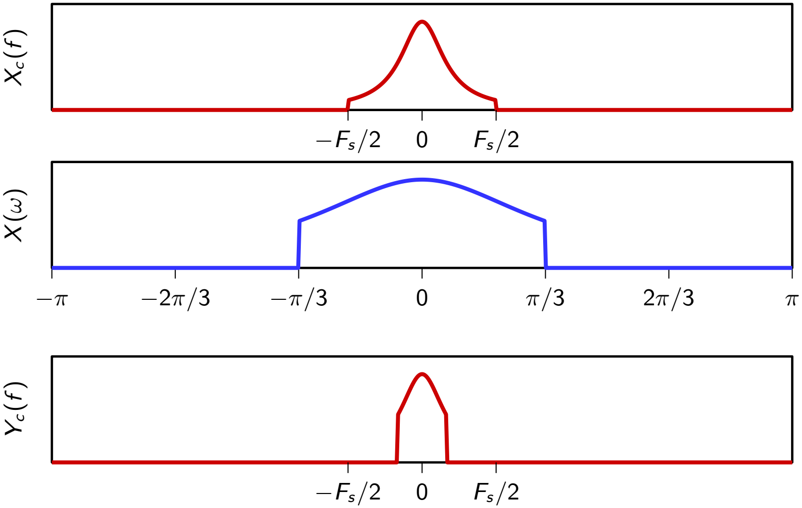
DT processing of CT signals



DT processing of CT signals



DT processing of CT signals



Example: analog bandpass with digital processing

- we want to implement a bandpass filter to select frequencies from 1 kHz to 2 kHz
- input signals are bandlimited with max positive frequency $F_N = 4\ kHz$
- we want to use digital processing

Example: analog bandpass with digital processing

analog bandpass filter:

- filter passband is $2f_c = 1$ kHz ($f_c = 500$ Hz)
- filter center frequency is $f_0 = 1500$ Hz

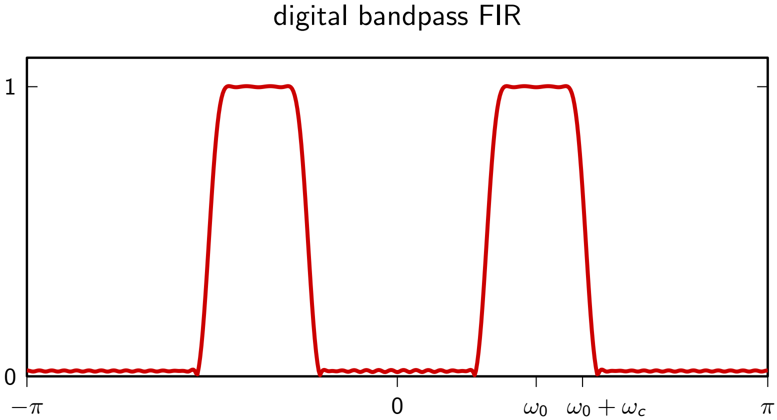
discrete-time processing chain

- input is 8 kHz-BL so we can use a sampling frequency $F_s = 8$ kHz
- design a FIR lowpass with cutoff $\omega_c = 2\pi(f_c/F_s)$
- modulate the impulse respose with $\omega_0 = 2\pi(f_0/F_s)$

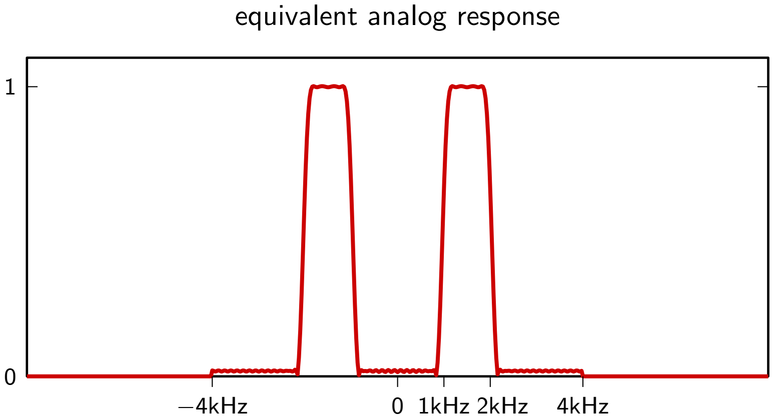
Example: analog bandpass with digital processing

```
import scipy.signal as sp

fc, f0, Fs = 500, 1500, 8000
wc, w0 = fc / Fs, f0 / Fs

N = 61
tbp = 0.2 # 20% transition band

h = sp.signal.remez(N, [0, wc*(1-tbp), wc*(1+tbp), 0.5], [1, 0], weight=[10, 1])
h *= 2 * np.cos(2 * np.pi * w0 * np.arange(len(h)))
```

Example: analog bandpass with digital processing

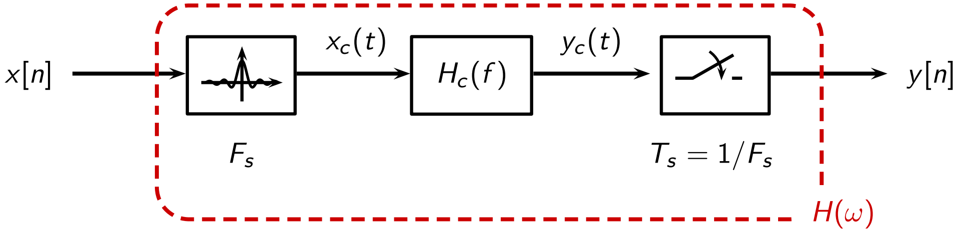


Example: analog bandpass with digital processing



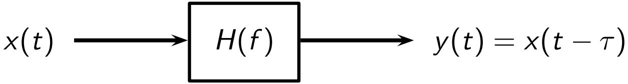
two more ideal filters

Dual setup



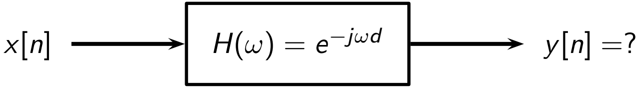
- $X_c(f) = (1/F_s)X(2\pi f/F_s)$
- $Y_c(f) = H_c(f)X_c(f)$
- $Y(\omega) = F_s Y_c(\frac{\omega}{2\pi}F_s) = H_c(\frac{\omega}{2\pi}F_s)X(\omega)$
- $H(\omega) = H_c(\frac{\omega}{2\pi}F_s)$

Delays in continuous time



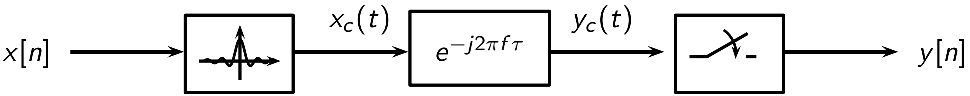
- in continuous time, delays are well defined for all $\tau \in \mathbb{R}$
- $H(f) = e^{-j2\pi f \tau}$

Delays in discrete time



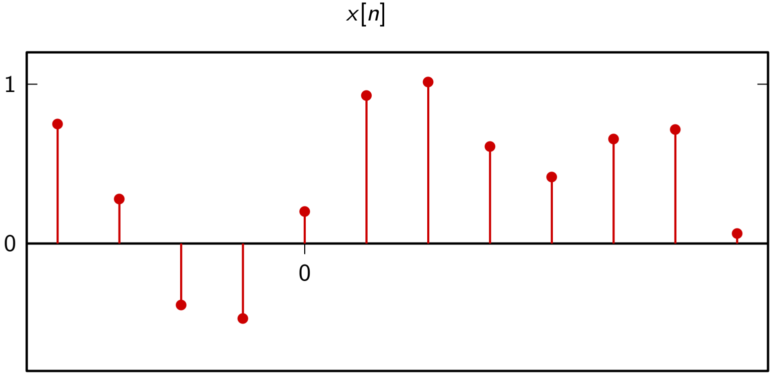
- when $d \in \mathbb{Z}$, then $y[n] = x[n-d]$
- what happens when d is not an integer?

Interpretation by duality

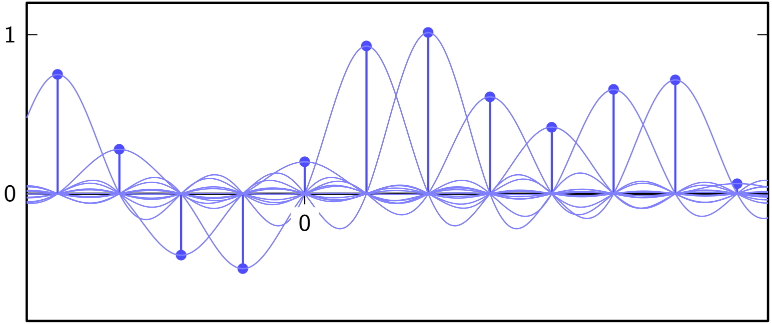


- a discrete-time delay can be implemented with interpolation, delay, and resampling
- equivalent filter: $H(\omega) = H_c(\omega/(2\pi)F_s) = e^{-j\omega d}$ with $d = \tau/T_s \in \mathbb{R}$
- impulse response: $h[n] = \text{sinc}(n - d)$
- if $d \in \mathbb{Z}$ then $h[n] = \delta[n - d]$ (normal delay) otherwise we have an ideal filter!

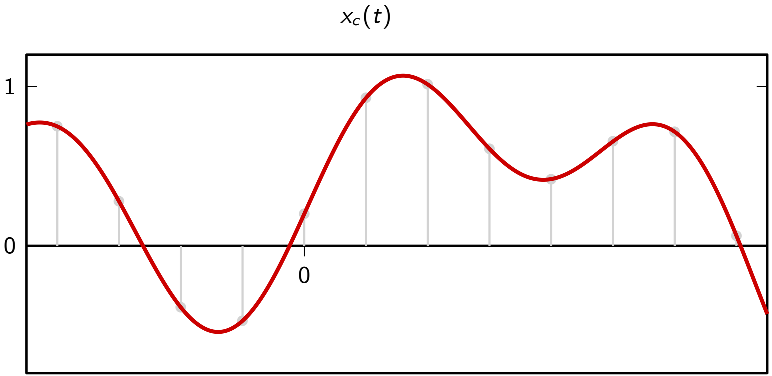
Fractional delay



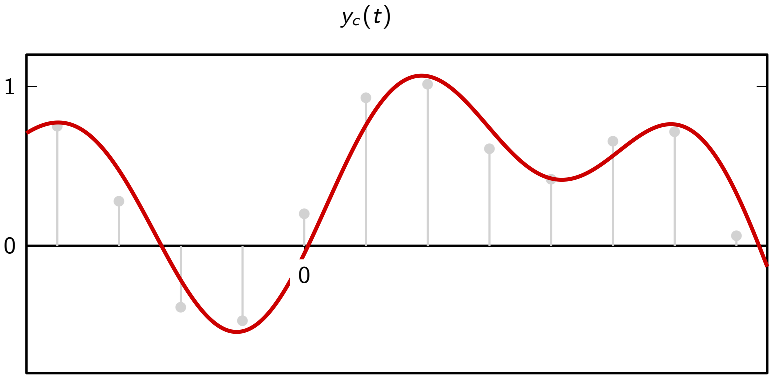
Fractional delay



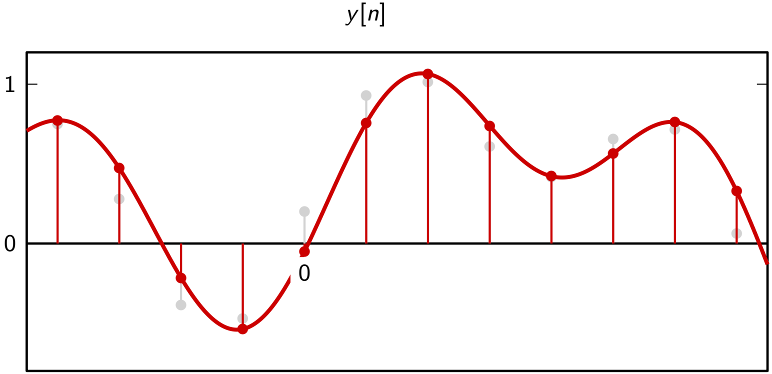
Fractional delay



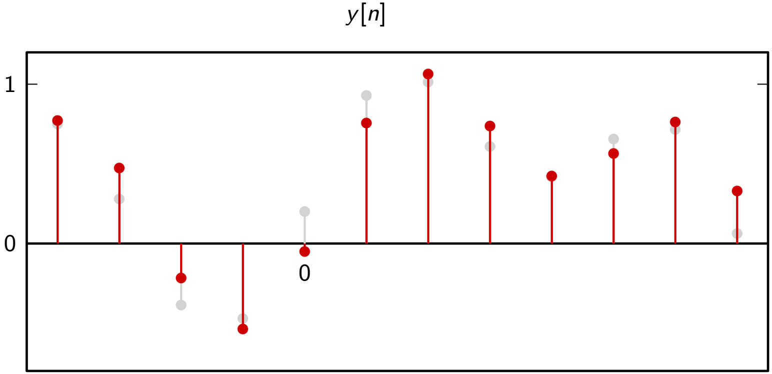
Fractional delay



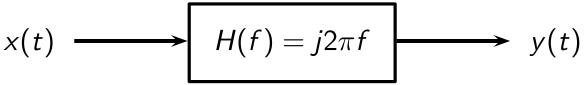
Fractional delay



Fractional delay

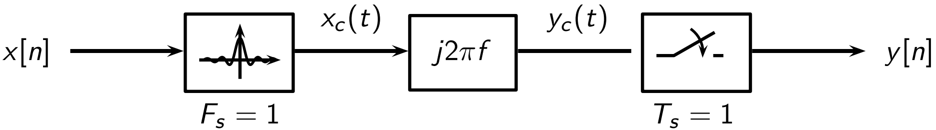


Differentiation in continuous time



- easy to show that $y(t) = x'(t) = \frac{\partial}{\partial t}x(t)$
- first derivative can be computed exactly via filtering

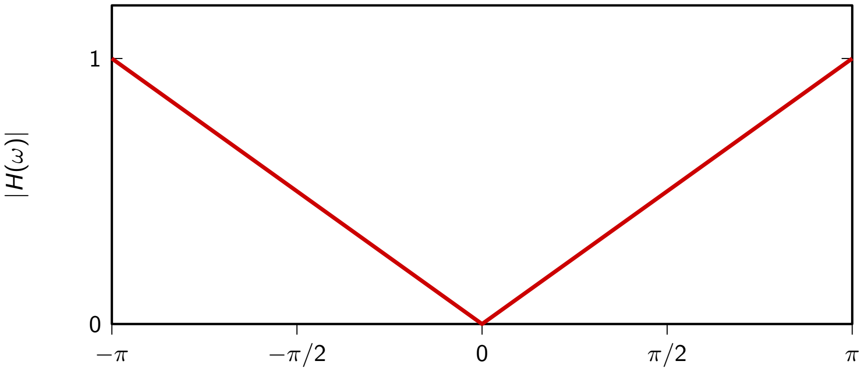
By duality



- chain interpolates the discrete-time input, differentiates the interpolation and resamples it
- equivalent filter $H(\omega) = H_c(\omega/(2\pi)) = j\omega$
- $H(\omega)$ is a "digital differentiator"

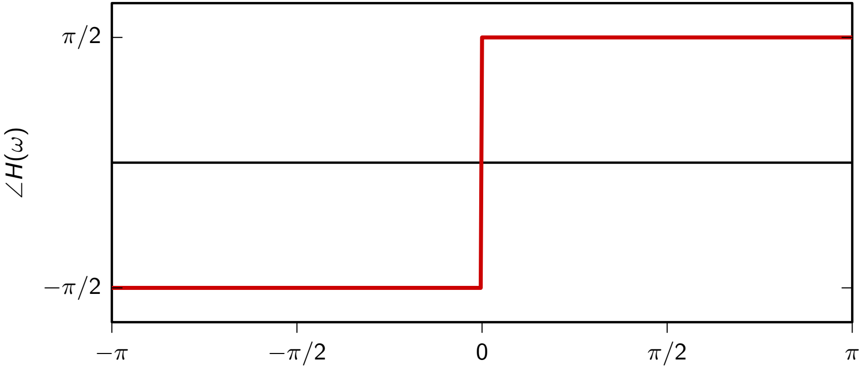
Digital differentiator, magnitude response

$|H(\omega)| = |\omega|$, highpass filter



Digital differentiator, phase response

$$\angle H(\omega) = (\pi/2)\,\mathrm{sign}(\omega)$$



Digital differentiator, impulse response

$$h[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} j\omega e^{j\omega n} d\omega$$

$$= \ldots \textit{(integration by parts)} \ldots$$

$$= \begin{cases} 0 & n = 0 \\ \dfrac{(-1)^n}{n} & n \neq 0 \end{cases}$$

the differentiator is an ideal filter

Digital differentiator, impulse response