COM303: Digital Signal Processing

Lecture 17: Sampling and applications

overview

- raw sampling and aliasing
- DT processing of CT signals

Sinc Sampling

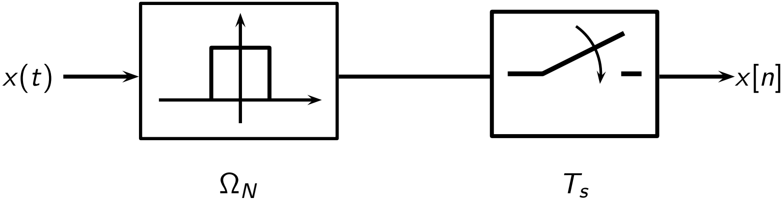
$$x[n] = \left\langle \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right), x(t) \right\rangle$$

Sinc Sampling

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

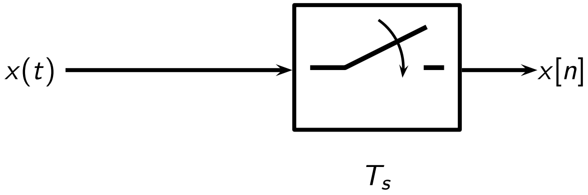
Sinc Sampling

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



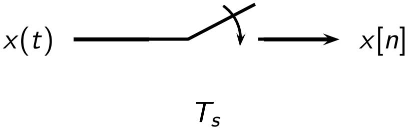
Sinc Sampling for Ω_N-BL signals

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



“Raw” sampling - can we always do that?

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



Remember the wagonwheel effect?

The continuous-time complex exponential

$$x(t) = e^{j\Omega_0 t} = e^{j2\pi F_0 t}$$

- always periodic, period $T_0 = 2\pi/\Omega_0 = 1/F_0$
- all angular speeds are allowed
- $\text{FT}\left\{e^{j\Omega_0 t}\right\} = 2\pi\delta(\Omega - \Omega_0)$
- bandlimited to Ω_0^+

The continuous-time complex exponential

$$x(t) = e^{j\Omega_0 t} = e^{j2\pi F_0 t}$$

- always periodic, period $T_0 = 2\pi/\Omega_0 = 1/F_0$
- 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

$$x(t) = e^{j\Omega_0 t} = e^{j2\pi F_0 t}$$

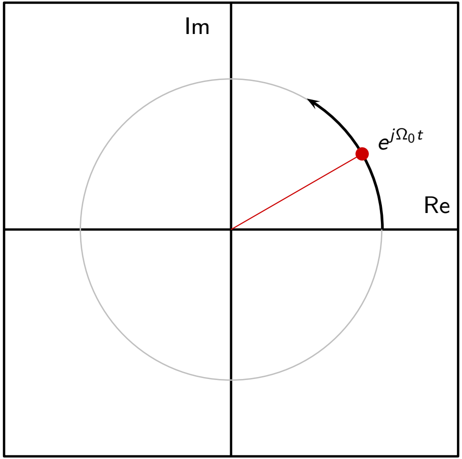
- always periodic, period $T_0 = 2\pi/\Omega_0 = 1/F_0$
- all angular speeds are allowed
- FT $\left\{e^{j\Omega_0 t}\right\} = 2\pi\delta(\Omega - \Omega_0)$
- bandlimited to Ω_0^+

The continuous-time complex exponential

$$x(t) = e^{j\Omega_0 t} = e^{j2\pi F_0 t}$$

- always periodic, period $T_0 = 2\pi/\Omega_0 = 1/F_0$
- 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

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

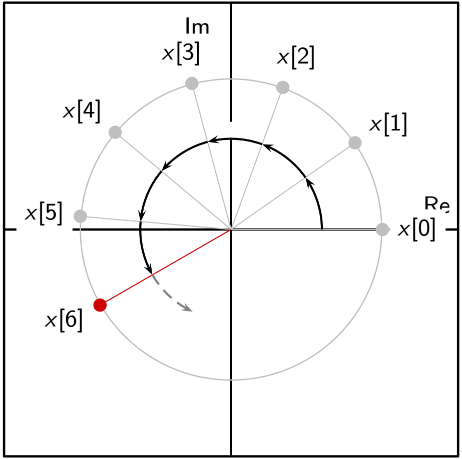
- raw samples are snapshots at regular intervals of the rotating point
- resulting digital frequency is $\omega_0 = \Omega_0 T_s = 2\pi(T_s/T_0)$

Raw samples of the continuous-time complex exponential

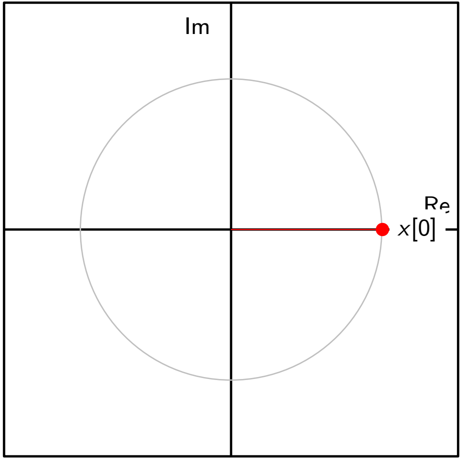
$$x[n] = e^{j\Omega_0 T_s n}$$

- raw samples are snapshots at regular intervals of the rotating point
- resulting digital frequency is $\omega_0 = \Omega_0 T_s = 2\pi(T_s/T_0)$

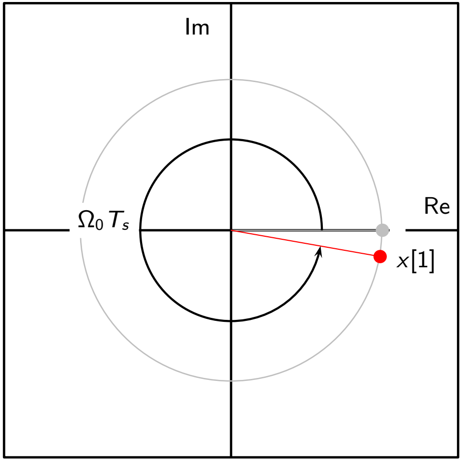
Easy: $T_s < T_0/2 \quad \Rightarrow \quad \omega_0 < \pi$



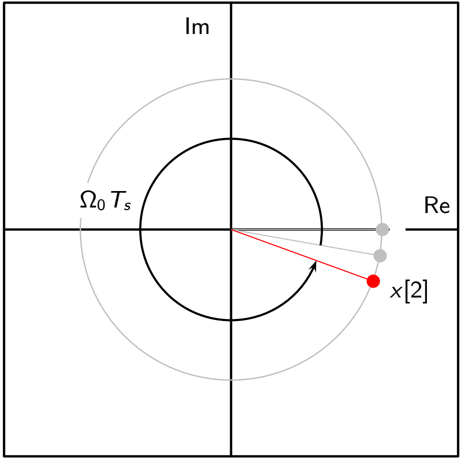
Tricky: $T_0/2 < T_s < T_0 \quad \Rightarrow \quad \pi < \omega_0 < 2\pi$



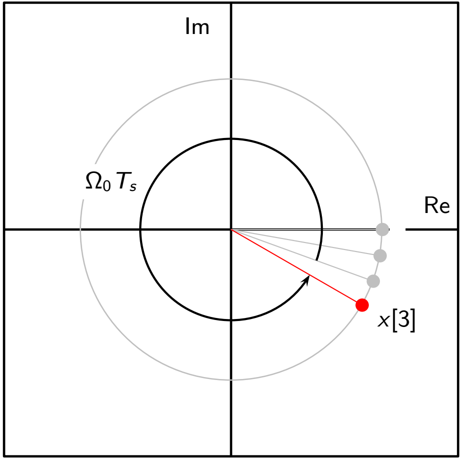
Tricky: $T_0/2 < T_s < T_0 \quad \Rightarrow \quad \pi < \omega_0 < 2\pi$



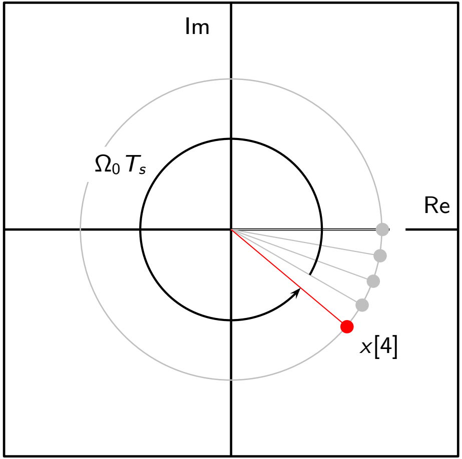
Tricky: $T_0/2 < T_s < T_0 \quad \Rightarrow \quad \pi < \omega_0 < 2\pi$



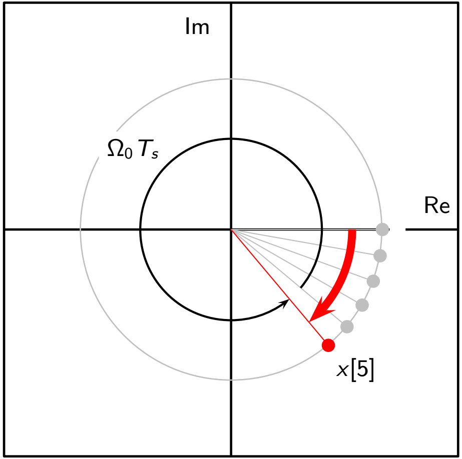
Tricky: $T_0/2 < T_s < T_0 \quad \Rightarrow \quad \pi < \omega_0 < 2\pi$



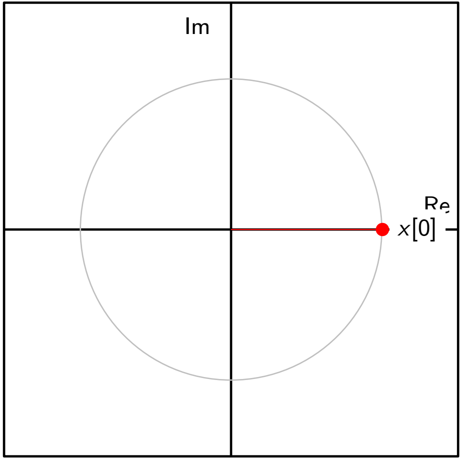
Tricky: $T_0/2 < T_s < T_0 \quad \Rightarrow \quad \pi < \omega_0 < 2\pi$



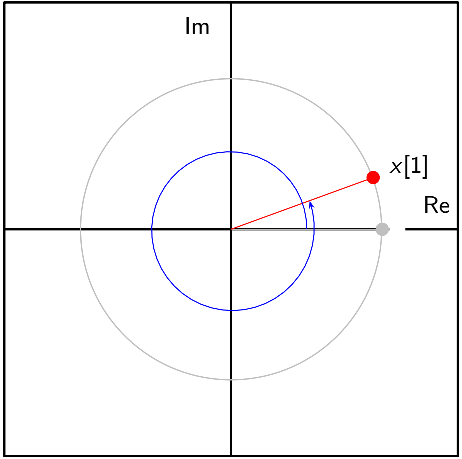
Tricky: $T_0/2 < T_s < T_0 \quad \Rightarrow \quad \pi < \omega_0 < 2\pi$



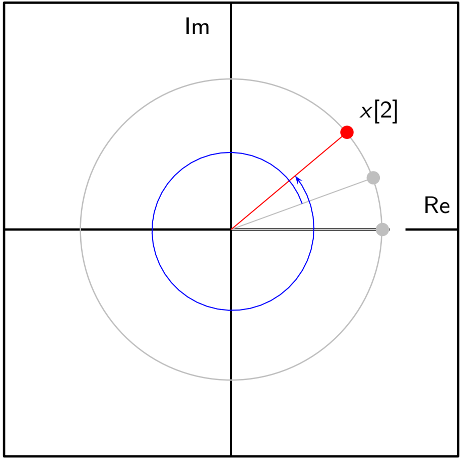
Trouble: $T_s > T_0 \quad \Rightarrow \quad \omega_0 > 2\pi$



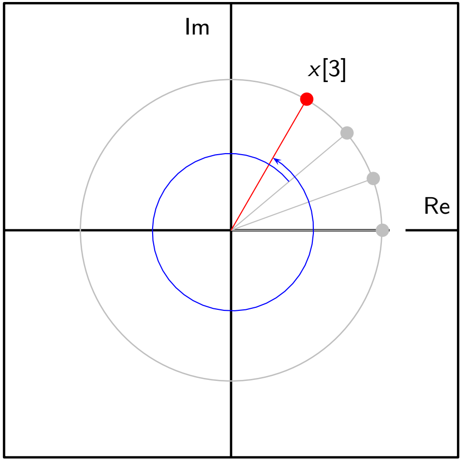
Trouble: $T_s > T_0 \quad \Rightarrow \quad \omega_0 > 2\pi$



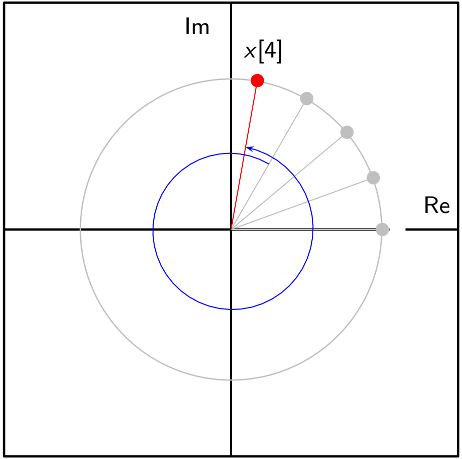
Trouble: $T_s > T_0 \quad \Rightarrow \quad \omega_0 > 2\pi$



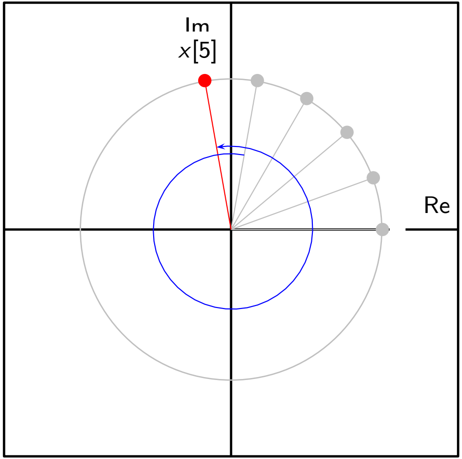
Trouble: $T_s > T_0 \quad \Rightarrow \quad \omega_0 > 2\pi$



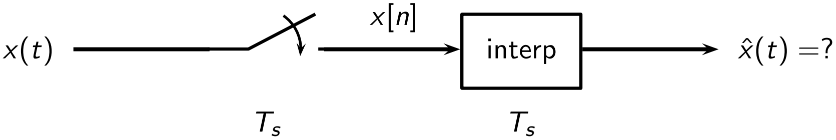
Trouble: $T_s > T_0 \quad \Rightarrow \quad \omega_0 > 2\pi$



Trouble: $T_s > T_0 \quad \Rightarrow \quad \omega_0 > 2\pi$



Aliasing



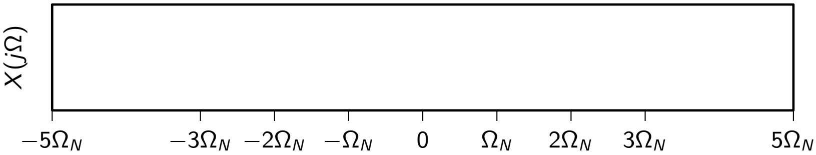
Aliasing

pick T_s: $\Omega_N = \pi / T_s$

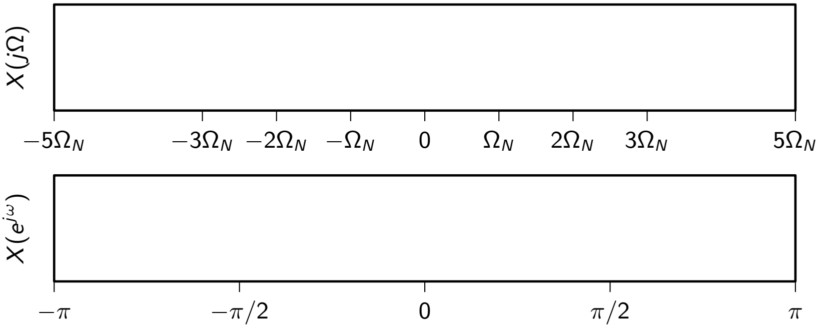
input: $x(t) = e^{j\Omega_0 t}$

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

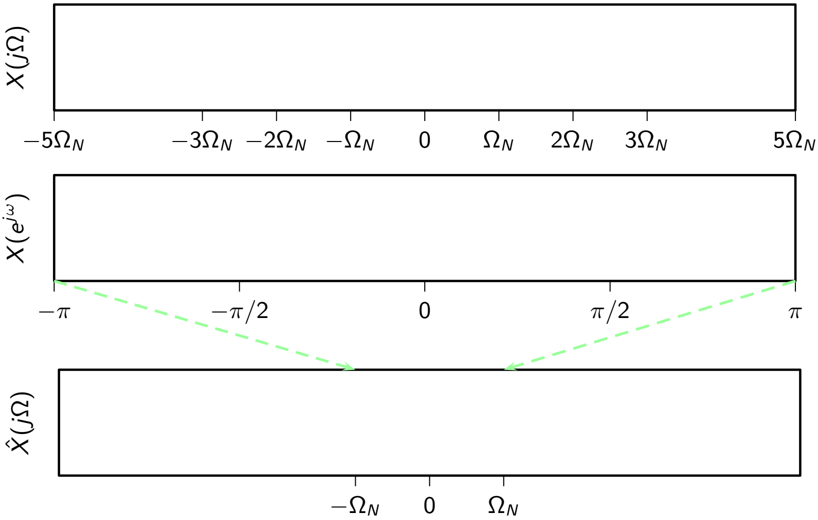
Aliasing of sinusoids: increasing the input frequency



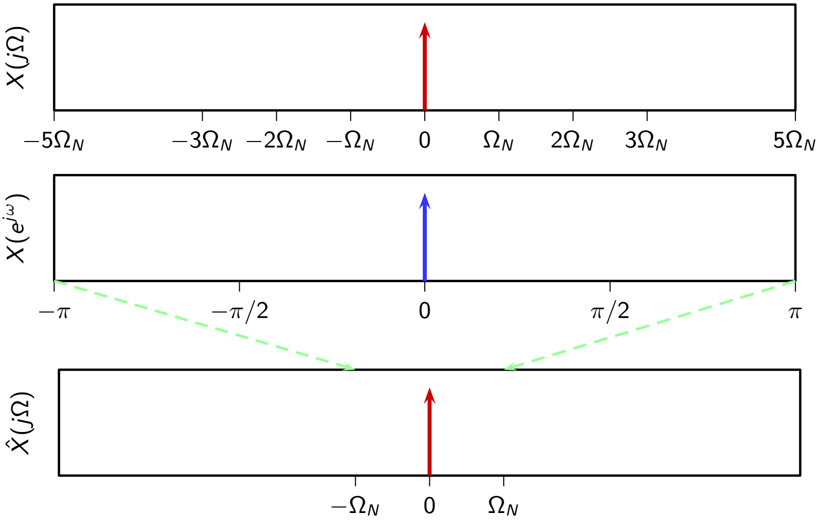
Aliasing of sinusoids: increasing the input frequency



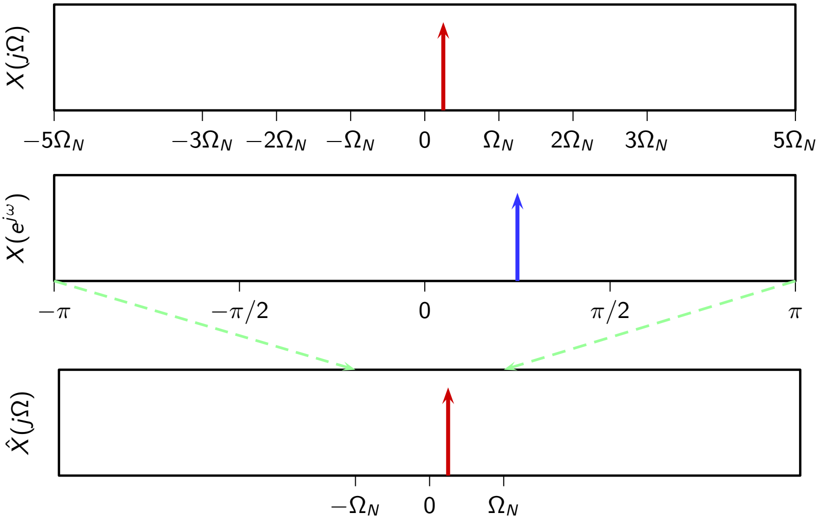
Aliasing of sinusoids: increasing the input frequency



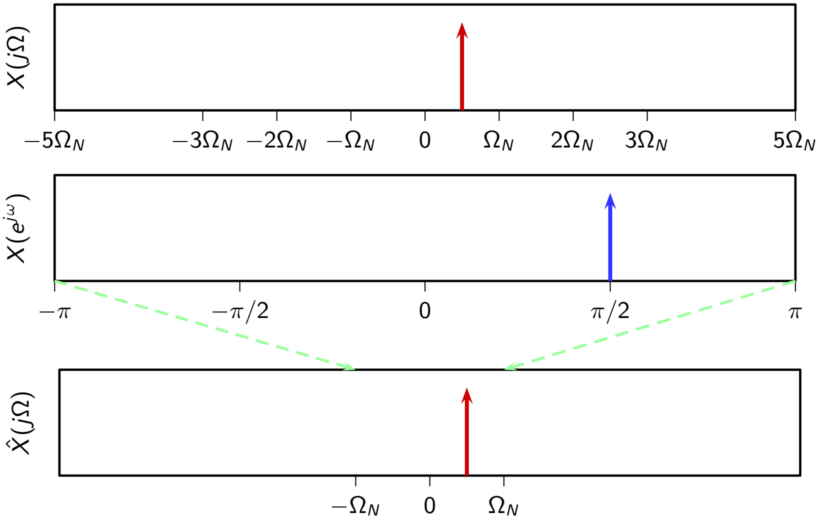
Aliasing of sinusoids: increasing the input frequency



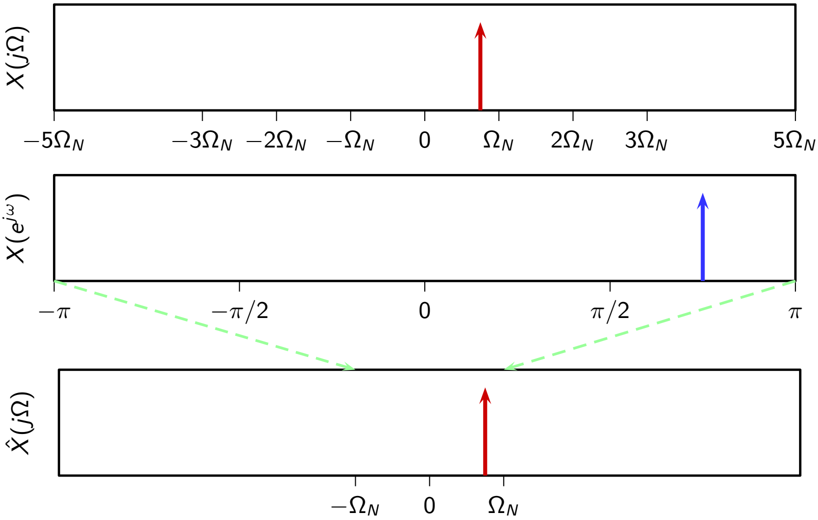
Aliasing of sinusoids: increasing the input frequency



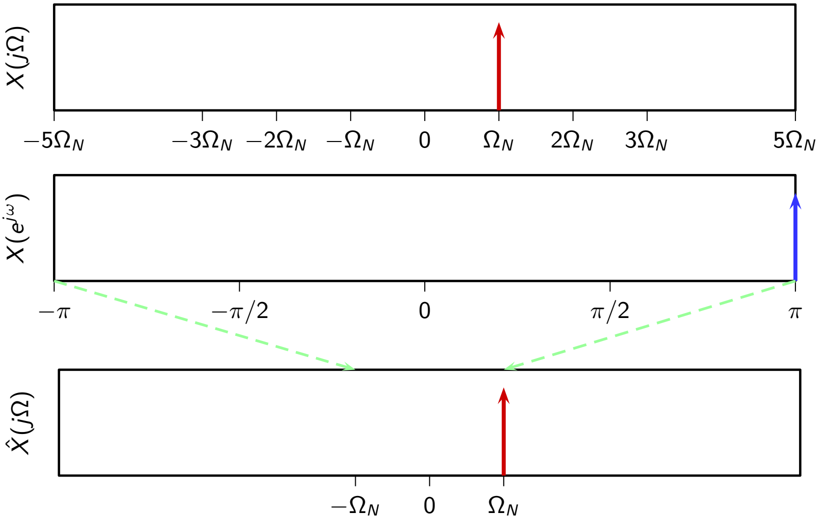
Aliasing of sinusoids: increasing the input frequency



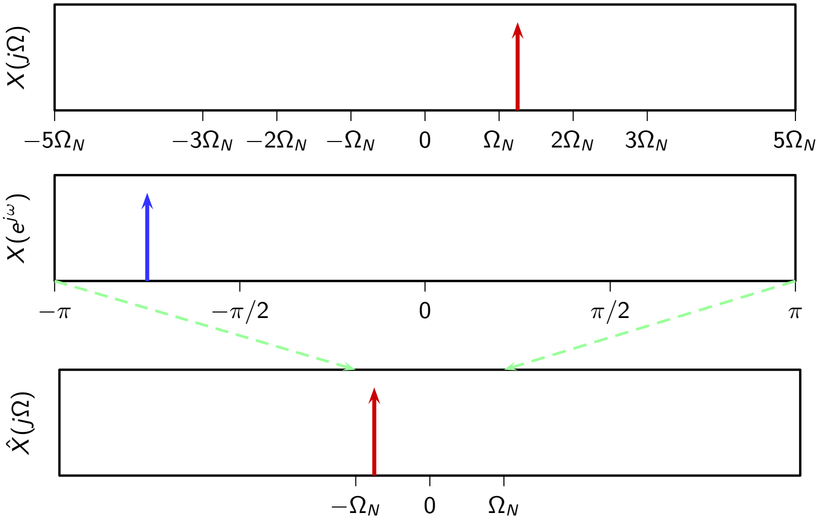
Aliasing of sinusoids: increasing the input frequency



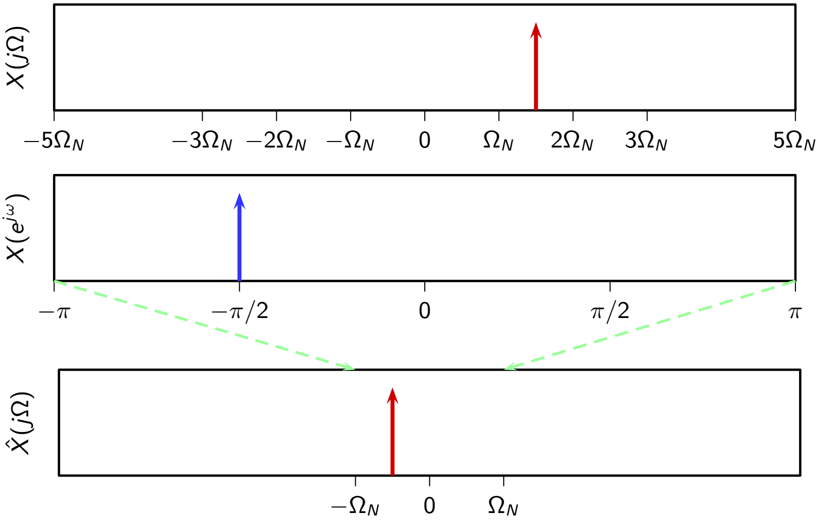
Aliasing of sinusoids: increasing the input frequency



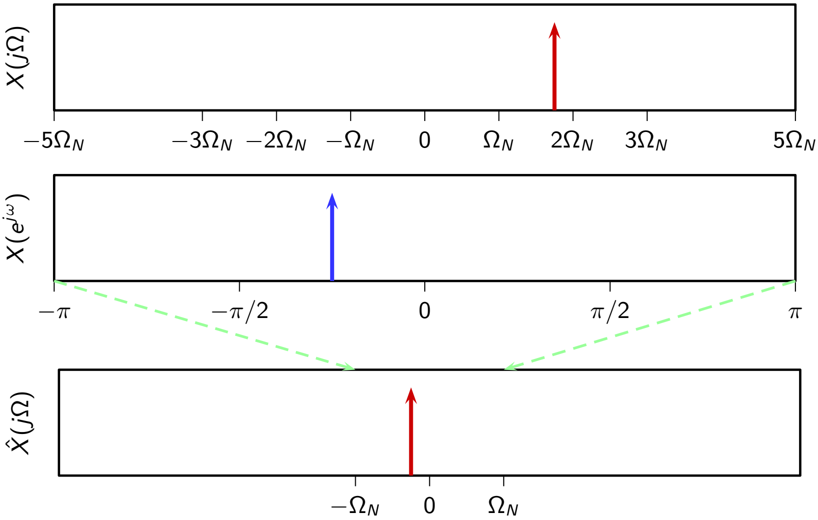
Aliasing of sinusoids: increasing the input frequency



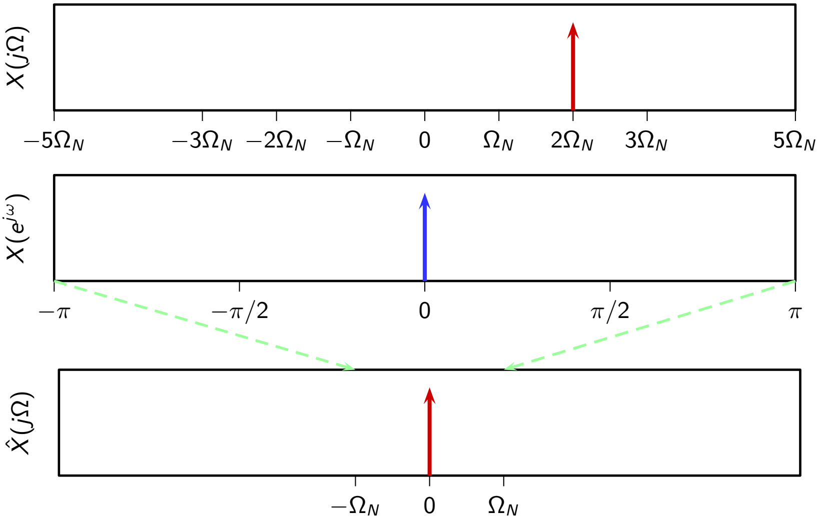
Aliasing of sinusoids: increasing the input frequency



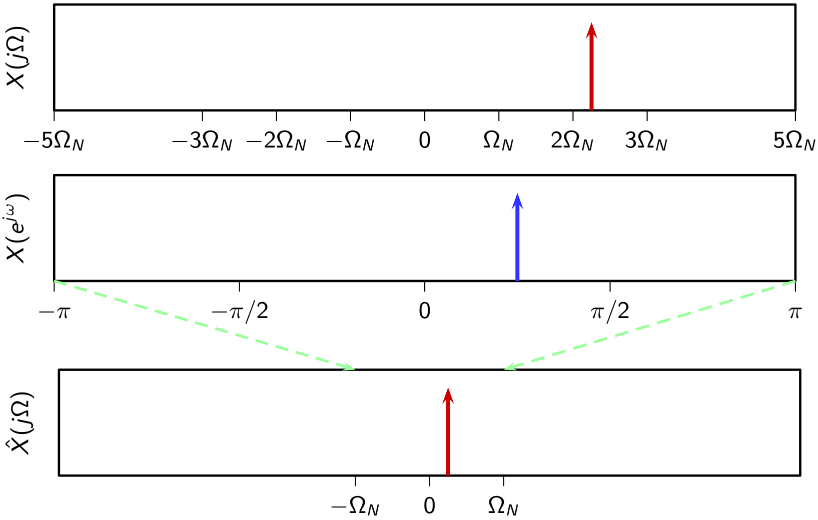
Aliasing of sinusoids: increasing the input frequency



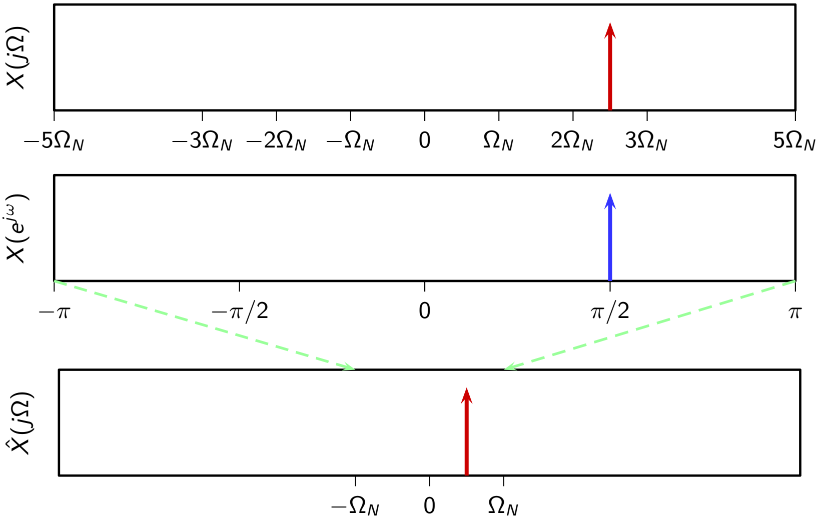
Aliasing of sinusoids: increasing the input frequency



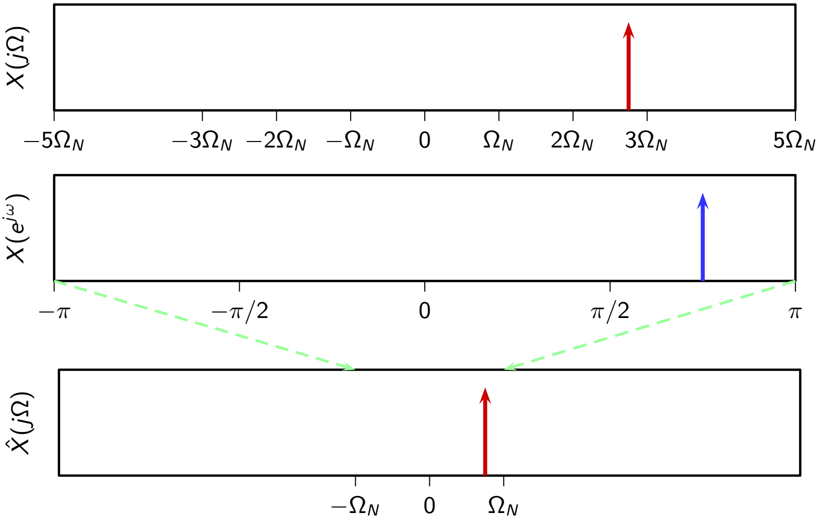
Aliasing of sinusoids: increasing the input frequency



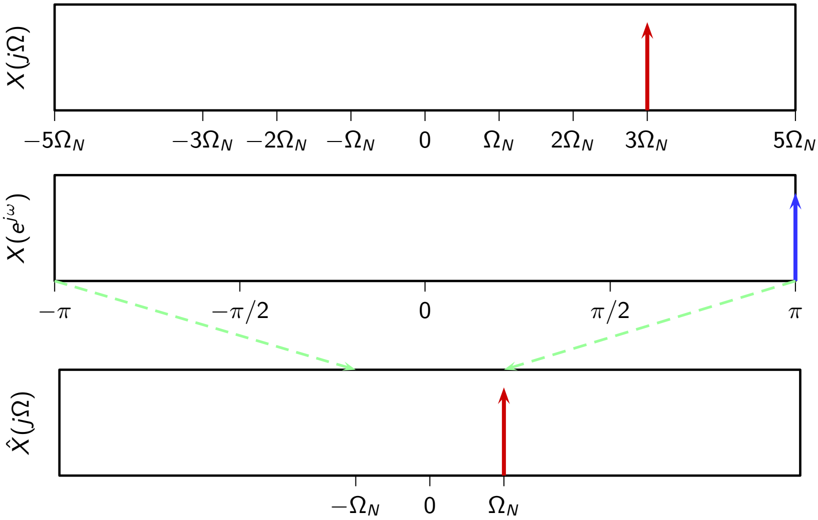
Aliasing of sinusoids: increasing the input frequency



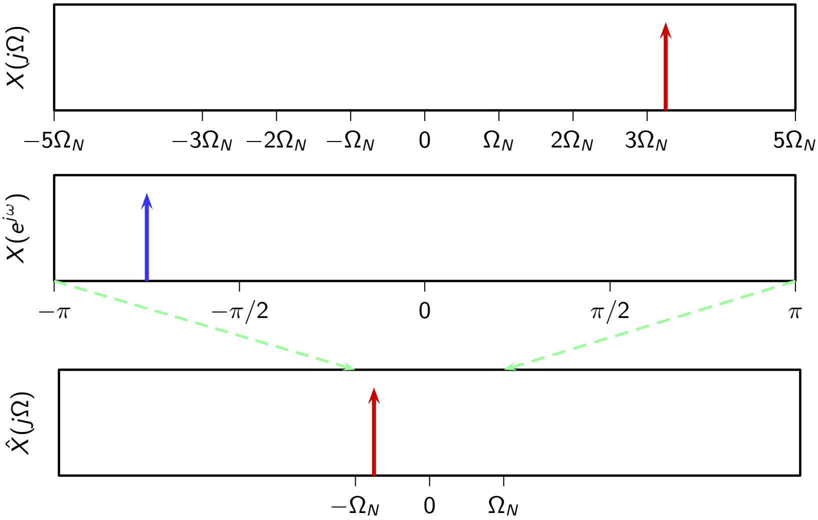
Aliasing of sinusoids: increasing the input frequency



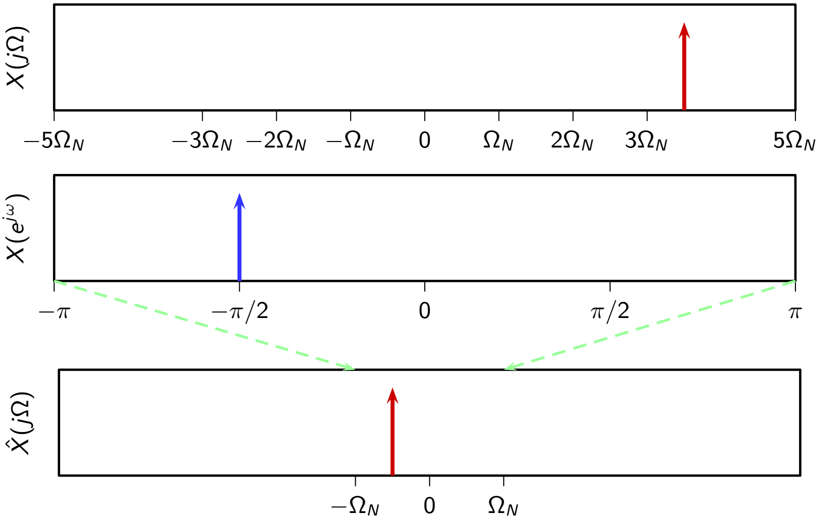
Aliasing of sinusoids: increasing the input frequency



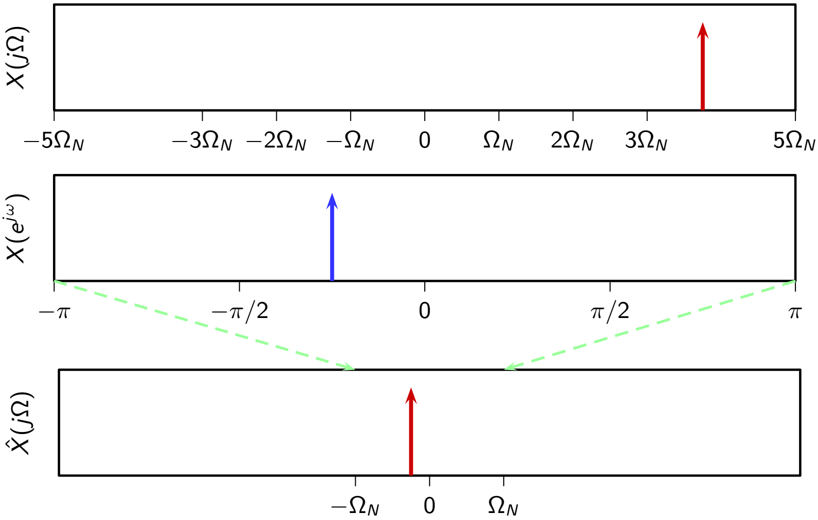
Aliasing of sinusoids: increasing the input frequency



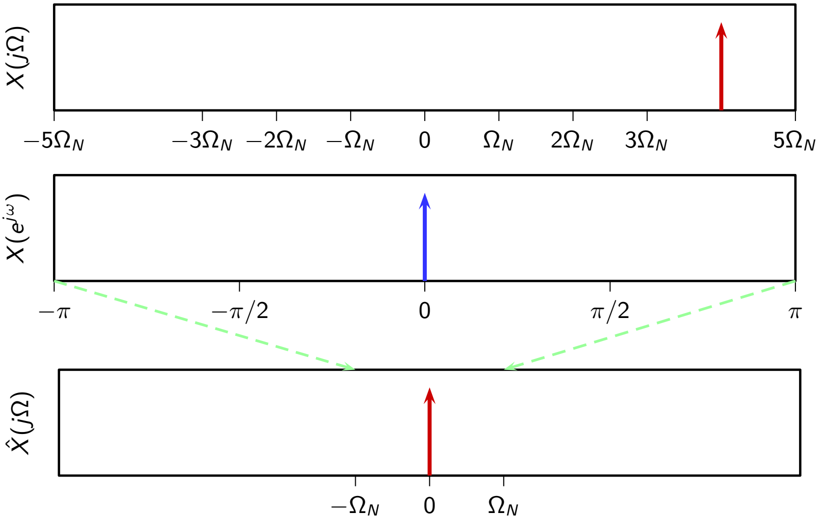
Aliasing of sinusoids: increasing the input frequency



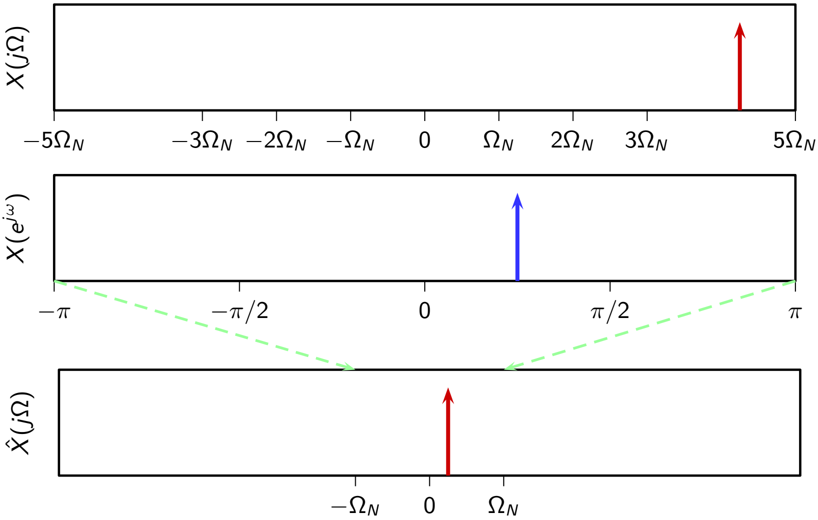
Aliasing of sinusoids: increasing the input frequency



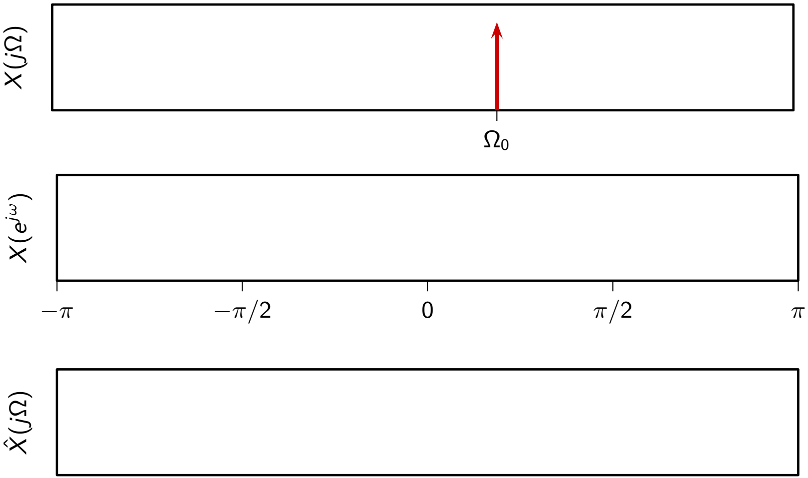
Aliasing of sinusoids: increasing the input frequency



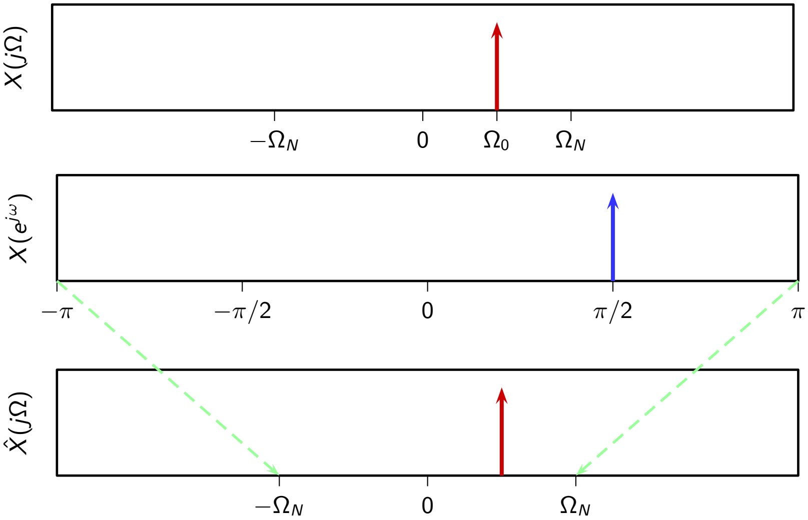
Aliasing of sinusoids: increasing the input frequency



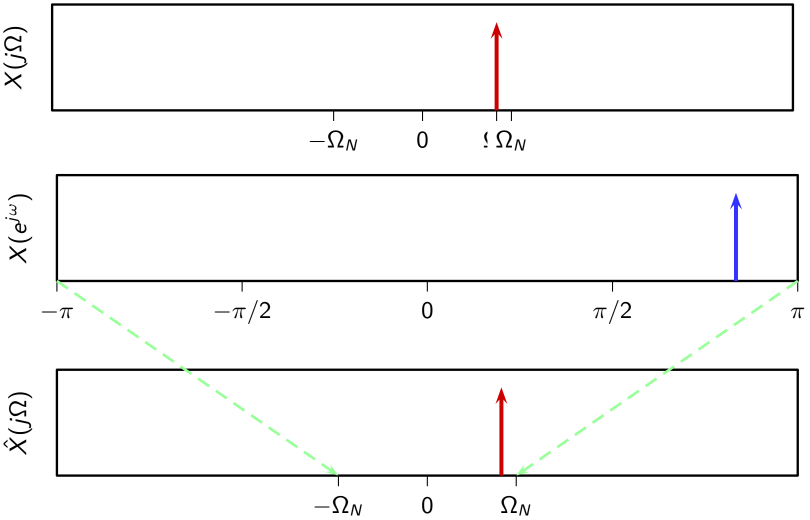
Aliasing of sinusoids: decreasing the sampling frequency



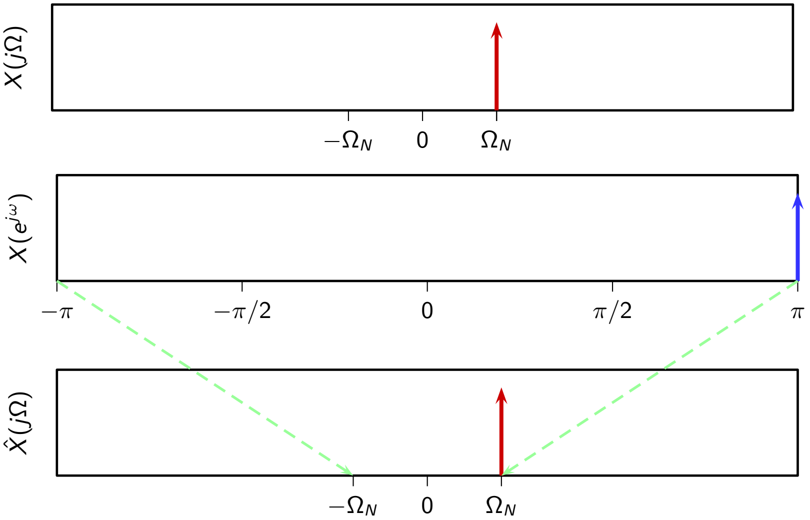
Aliasing of sinusoids: decreasing the sampling frequency



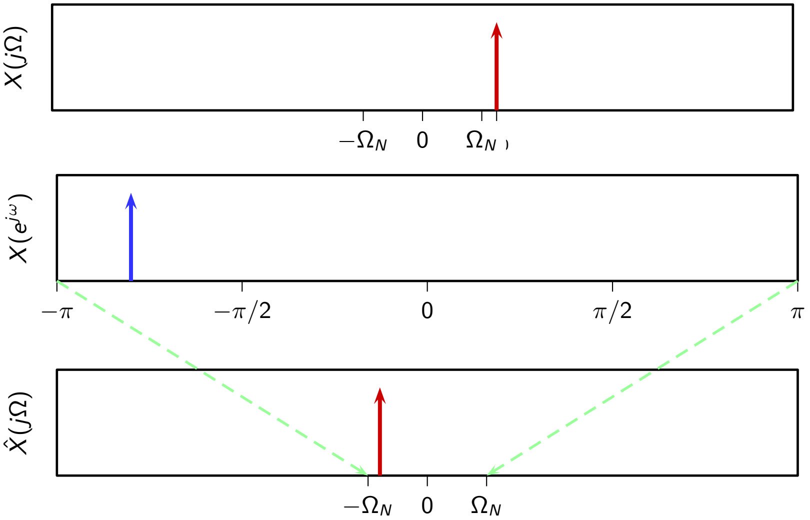
Aliasing of sinusoids: decreasing the sampling frequency



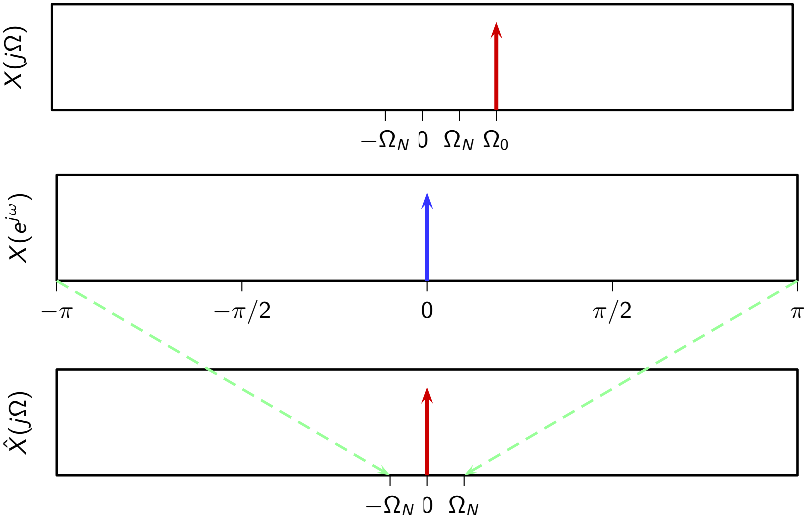
Aliasing of sinusoids: decreasing the sampling frequency



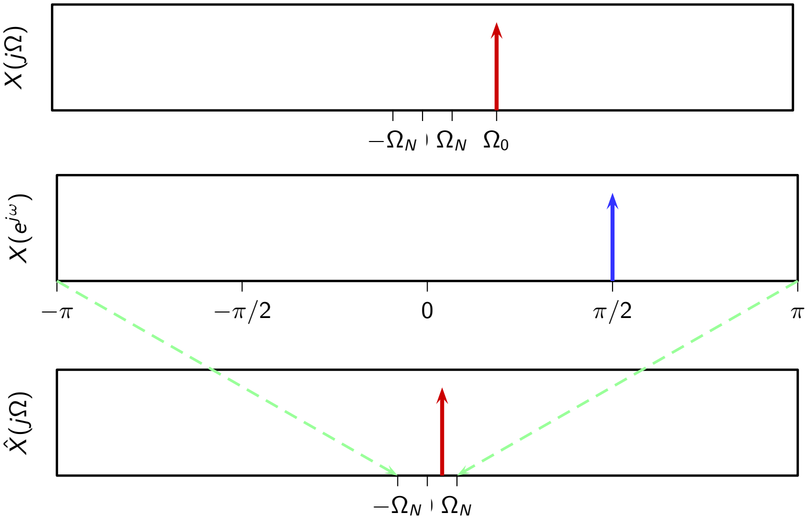
Aliasing of sinusoids: decreasing the sampling frequency



Aliasing of sinusoids: decreasing the sampling frequency



Aliasing of sinusoids: decreasing the sampling frequency

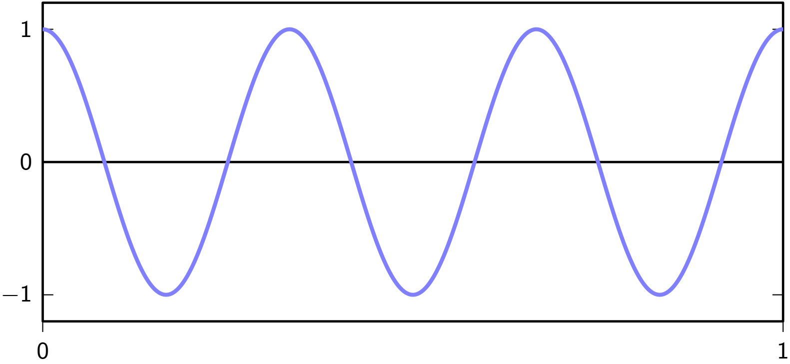


Sampling a Sinusoid

sampling frequency	digital frequency	interpolation
$F_s > 2F_0$	$0 < \omega_0 < \pi$	OK: $\hat{F}_0 = F_0$
$F_s = 2F_0$	$\omega_0 = \pi$	OK (max frequency $\hat{F}_0 = F_s$)
$F_0 < F_s < 2F_0$	$\pi < \omega_0 < 2\pi$	negative frequency: $\hat{F}_0 = F_0 - F_s$
$F_s < F_0$	$\omega_0 > 2\pi$	full aliasing: $\hat{F}_0 = F_0 \mod F_s$

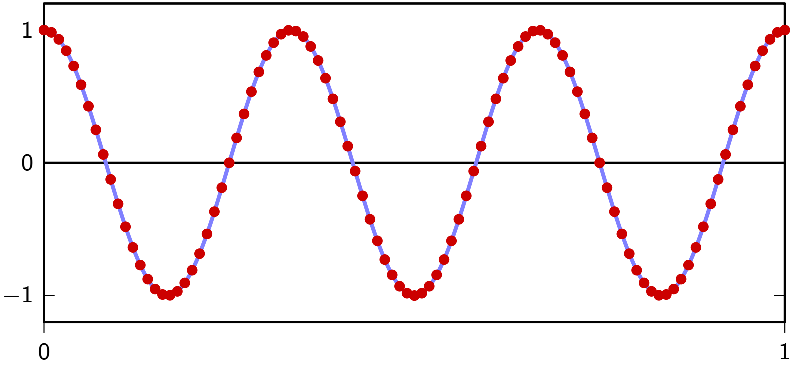
Aliasing: Sampling a Sinusoid

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



Aliasing: Sampling a Sinusoid

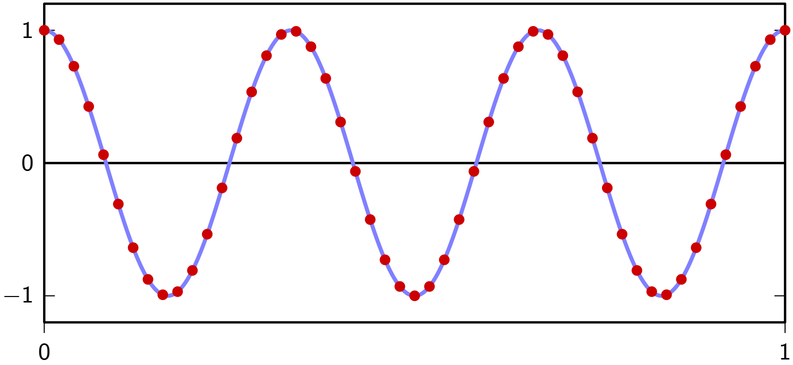
$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



$F_s = 100\text{Hz}$

Aliasing: Sampling a Sinusoid

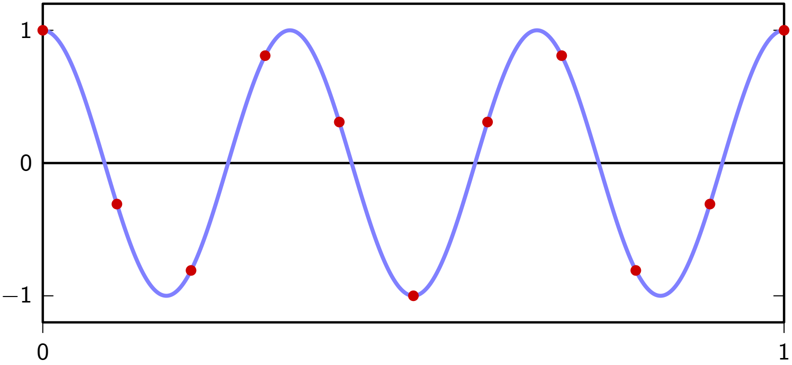
$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



$F_s = 50\text{Hz}$

Aliasing: Sampling a Sinusoid

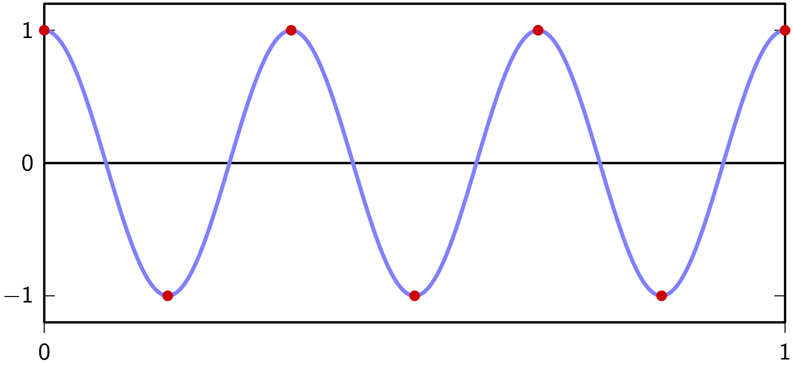
$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



$F_s = 10\text{Hz}$

Aliasing: Sampling a Sinusoid

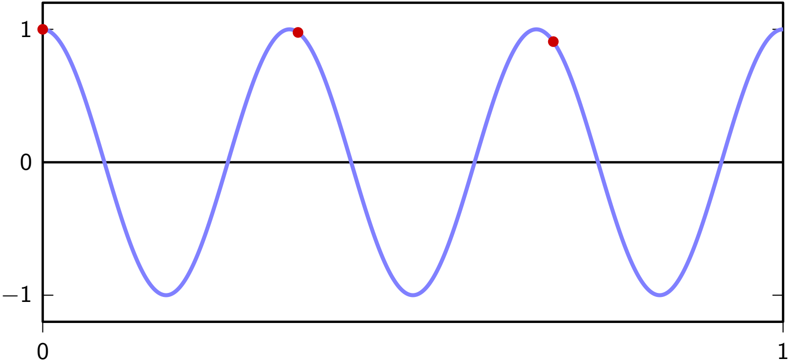
$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



$F_s = 6\text{Hz}$

Aliasing: Sampling a Sinusoid

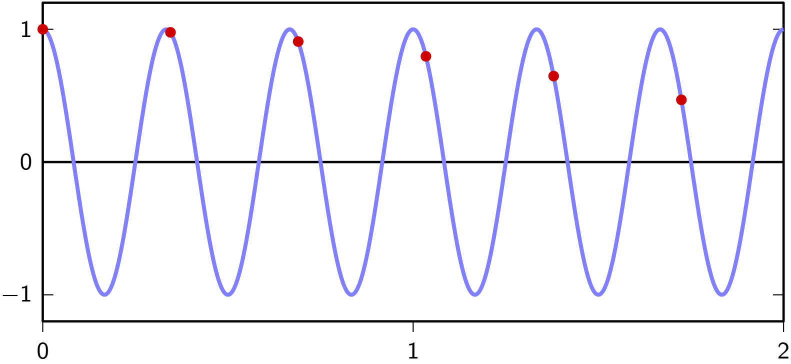
$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



$F_s = 2.9\text{Hz}$

Aliasing: Sampling a Sinusoid

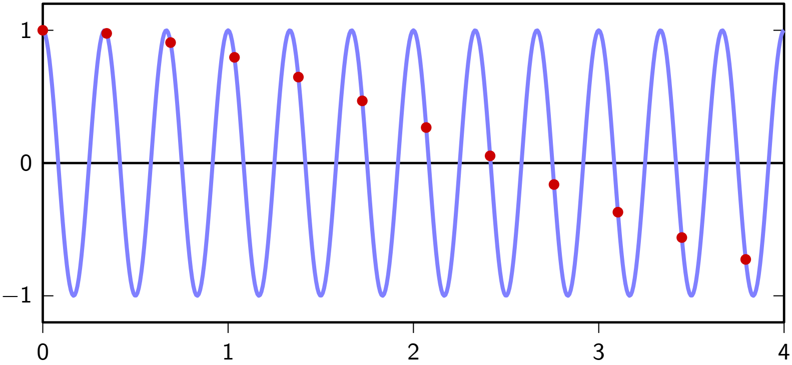
$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



$F_s = 2.9\text{Hz}$

Aliasing: Sampling a Sinusoid

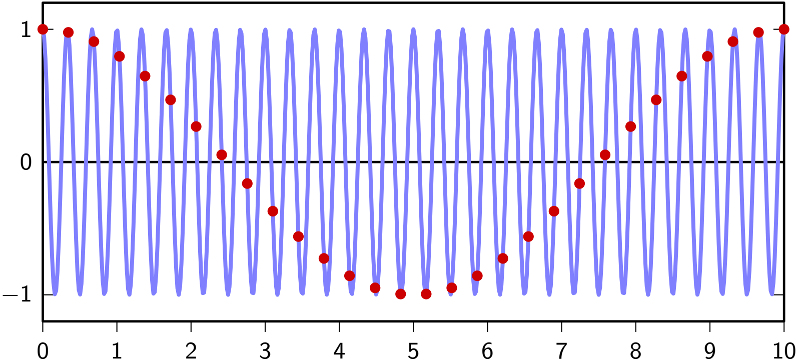
$$x(t) = \cos(6\pi t) \qquad (F_0 = 3\text{Hz})$$



$F_s = 2.9\text{Hz}$

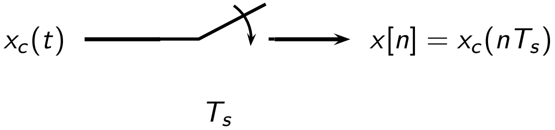
Aliasing: Sampling a Sinusoid

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

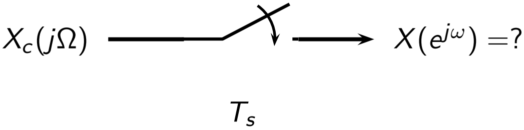


$F_s = 2.9\text{Hz}$

Raw-sampling an arbitrary signal

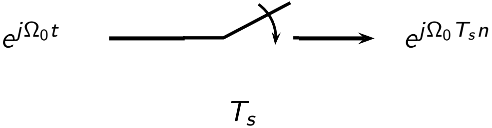


Raw-sampling an arbitrary signal



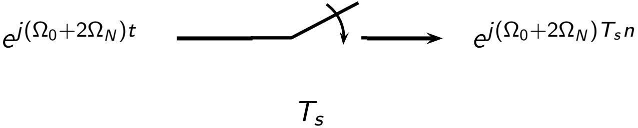
Key idea

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



Key idea

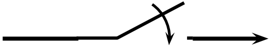
- pick T_s (and set $\Omega_N = \pi/T_s$)
- pick $\Omega_0 < \Omega_N$



Key idea

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

$e^{j(\Omega_0 + 2\Omega_N)t}$



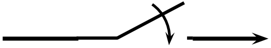
$e^{j(\Omega_0 T_s n + 2\Omega_N T_s n)}$

T_s

Key idea

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

$$e^{j(\Omega_0 + 2\Omega_N)t}$$

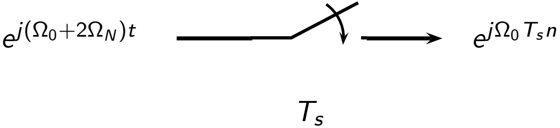


$$e^{j(\Omega_0 T_s n + 2\pi n)}$$

T_s

Key idea

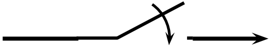
- pick T_s (and set $\Omega_N = \pi / T_s$)
- pick $\Omega_0 < \Omega_N$



Key idea

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

$$Ae^{j\Omega_0 t} + Be^{j(\Omega_0 + 2\Omega_N)t}$$



T_s

$$(A + B)e^{j\Omega_0 T_s n}$$

Spectrum of raw-sampled signals

outline: start with the inverse Fourier Transform

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

and manipulate the integral until it looks like

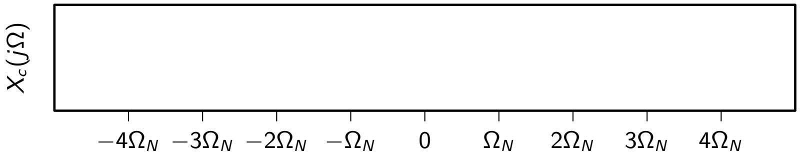
$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\omega)e^{j\omega n}d\omega$$

Spectrum of raw-sampled signals

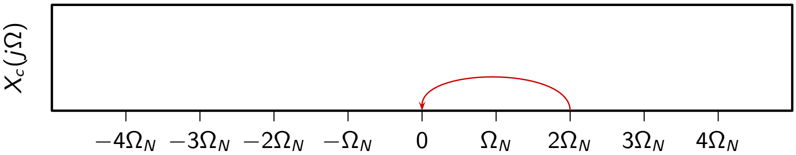
frequencies $2\Omega_N$ apart will be aliased, so split the integration interval

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

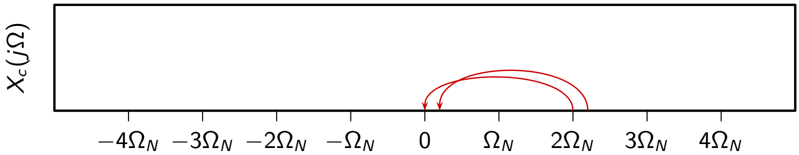
Spectrum of raw-sampled signals



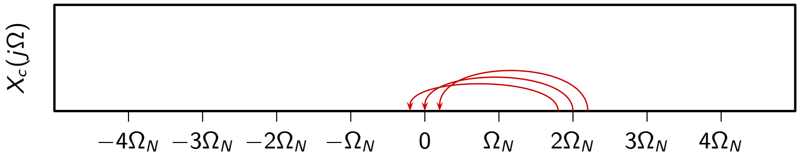
Spectrum of raw-sampled signals



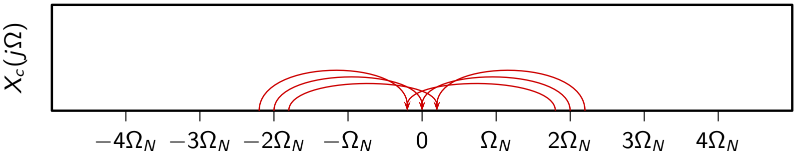
Spectrum of raw-sampled signals



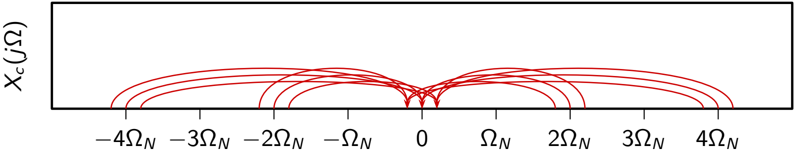
Spectrum of raw-sampled signals



Spectrum of raw-sampled signals



Spectrum of raw-sampled signals



Spectrum of raw-sampled signals

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

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:

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

Spectrum of raw-sampled signals

set $\omega = \Omega T_s$:

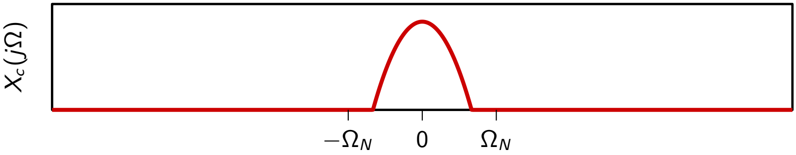
$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{1}{T_s}\tilde{X}_c\left(j\frac{\omega}{T_s}\right) e^{j\omega n} d\omega$$

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

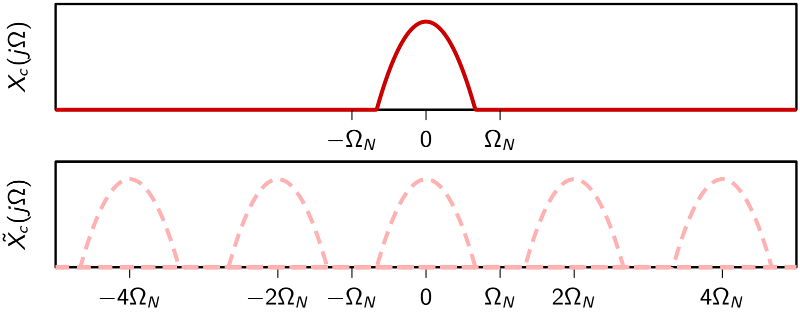
$$X(e^{j\omega}) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_c\left(j\frac{\omega}{T_s} - j\frac{2\pi k}{T_s}\right)$$

$$= \frac{\pi}{\Omega_N}\sum_{k=-\infty}^{\infty} X_c\left(j\frac{\omega\Omega_N}{\pi} - 2jk\Omega_N\right)$$

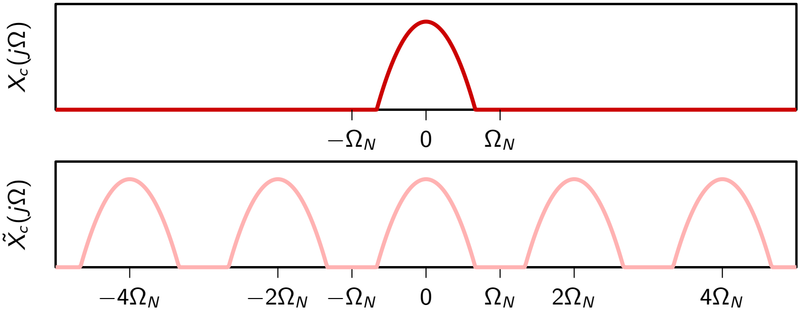
Example: signal bandlimited to Ω_0 and $\Omega_N > \Omega_0$



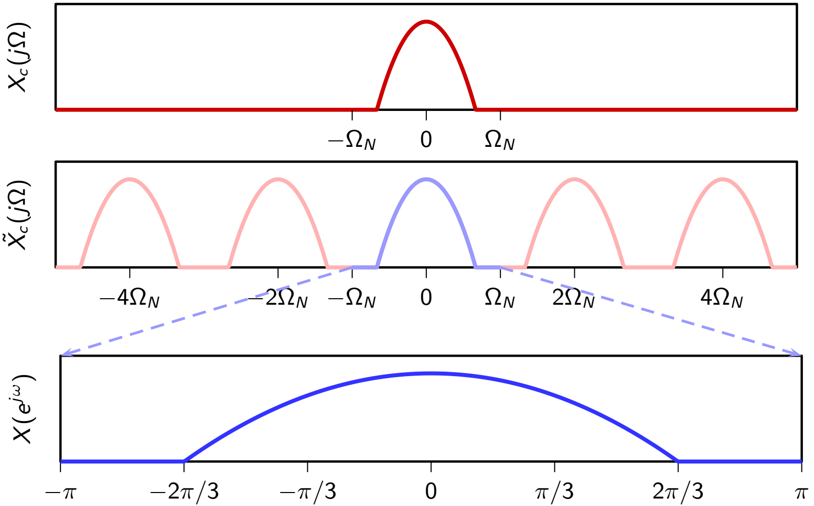
Example: signal bandlimited to Ω_0 and $\Omega_N > \Omega_0$



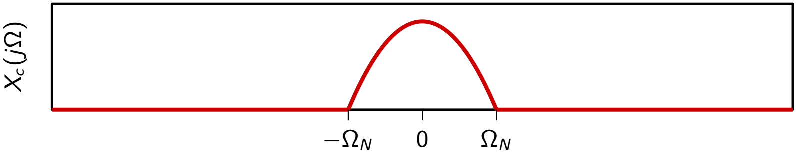
Example: signal bandlimited to Ω_0 and $\Omega_N > \Omega_0$



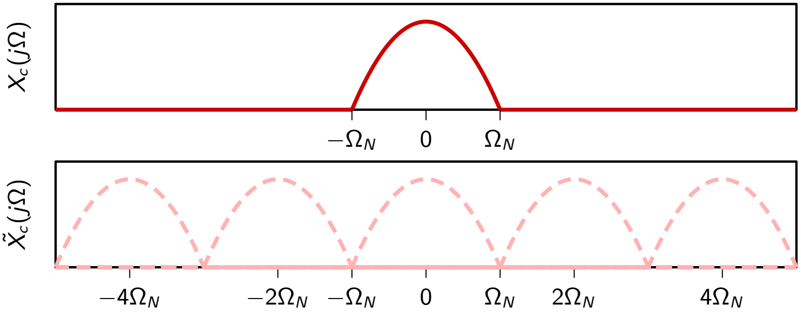
Example: signal bandlimited to Ω_0 and $\Omega_N > \Omega_0$



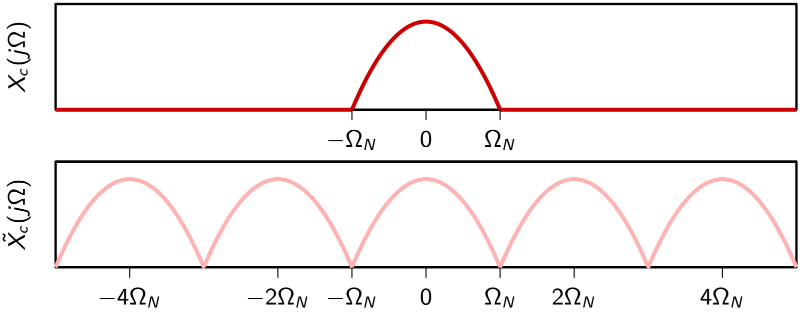
Example: signal bandlimited to Ω_0 and $\Omega_N = \Omega_0$



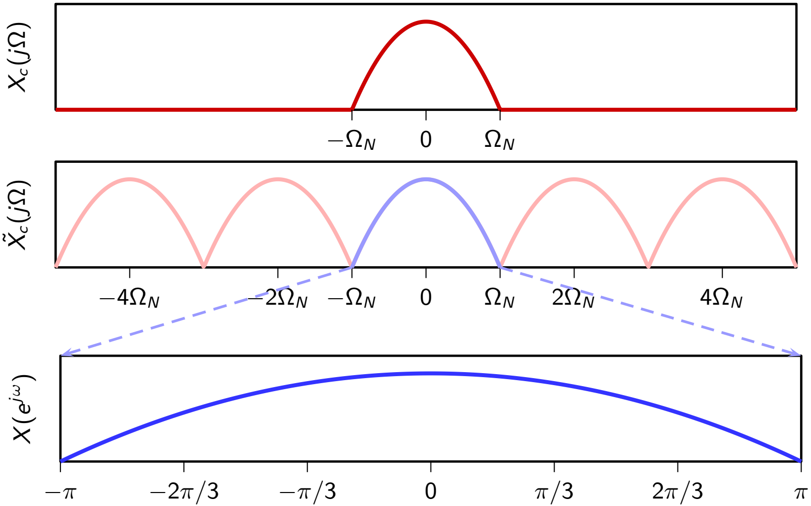
Example: signal bandlimited to Ω_0 and $\Omega_N = \Omega_0$



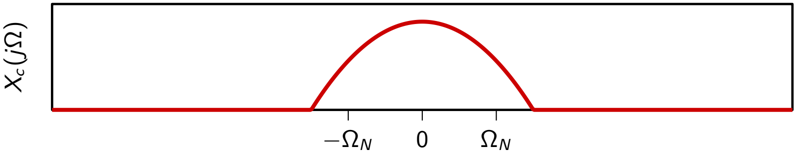
Example: signal bandlimited to Ω_0 and $\Omega_N = \Omega_0$



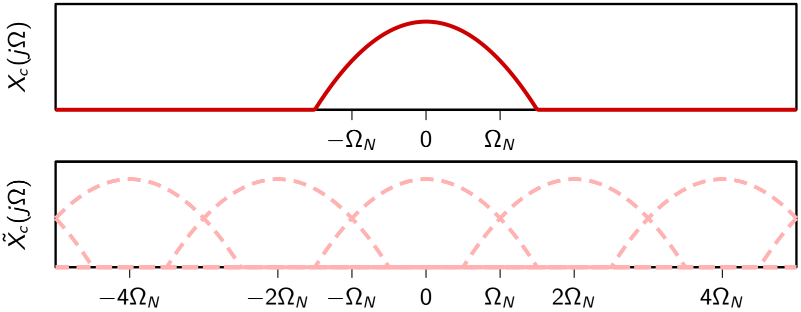
Example: signal bandlimited to Ω_0 and $\Omega_N = \Omega_0$



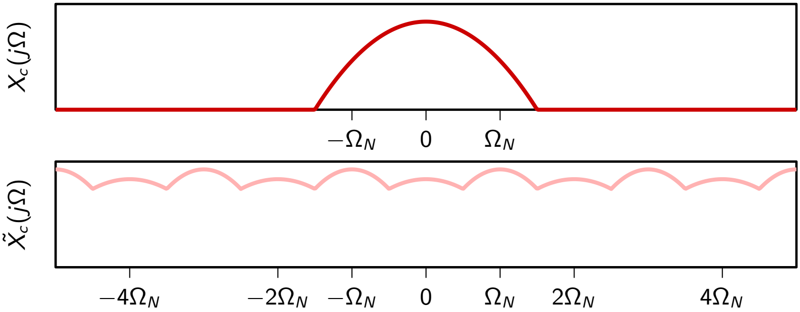
Example: signal bandlimited to Ω_0 and $\Omega_N < \Omega_0$



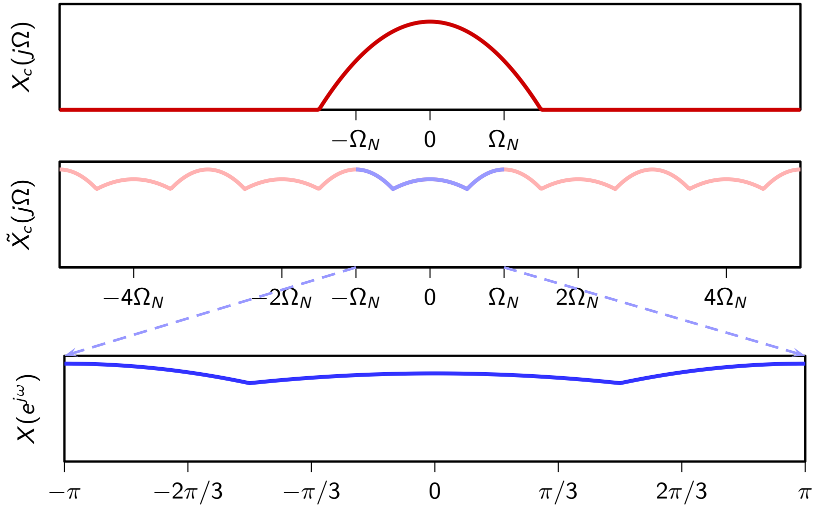
Example: signal bandlimited to Ω_0 and $\Omega_N < \Omega_0$



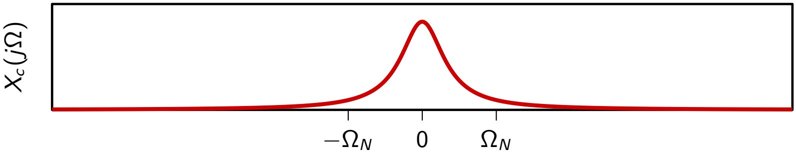
Example: signal bandlimited to Ω_0 and $\Omega_N < \Omega_0$



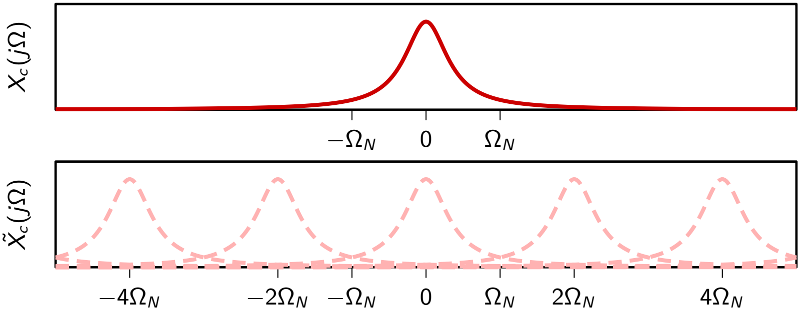
Example: signal bandlimited to Ω_0 and $\Omega_N < \Omega_0$



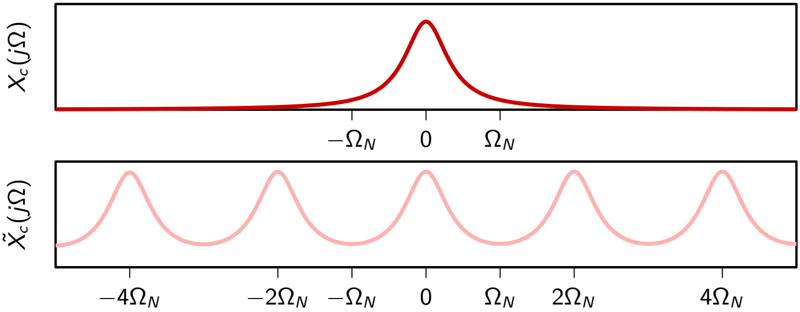
Example: non-bandlimited signal



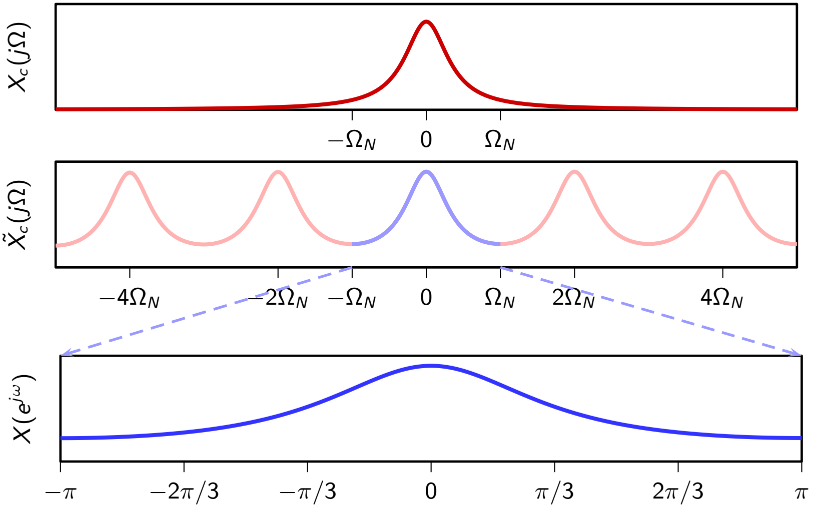
Example: non-bandlimited signal



Example: non-bandlimited signal



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 introduces errors we cannot control, so the sensible choice is to bandlimit in continuous time
- bandlimiting is also optimal wrt least squares approximation!

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 introduces errors we cannot control, so the sensible choice is to bandlimit in continuous time
- bandlimiting is also optimal wrt least squares approximation!

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 introduces errors we cannot control, so the sensible choice is to bandlimit in continuous time
- bandlimiting is also optimal wrt least squares approximation!

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 introduces errors we cannot control, so the sensible choice is to bandlimit in continuous time
- bandlimiting is also optimal wrt least squares approximation!

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 introduces errors we cannot control, so the sensible choice is to bandlimit in continuous time
- bandlimiting is also optimal wrt least squares approximation!

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 introduces errors we cannot control, so the sensible choice is to bandlimit in continuous time
- bandlimiting is also optimal wrt least squares approximation!

Sinc Sampling and Interpolation

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

Sinc Sampling and Interpolation

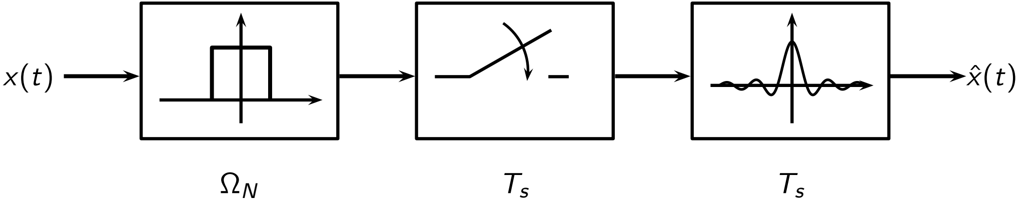
$$\hat{x}[n] = \left\langle \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right), x(t) \right\rangle \;=\; (\operatorname{sinc}_{T_s} * x)(nT_s)$$

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

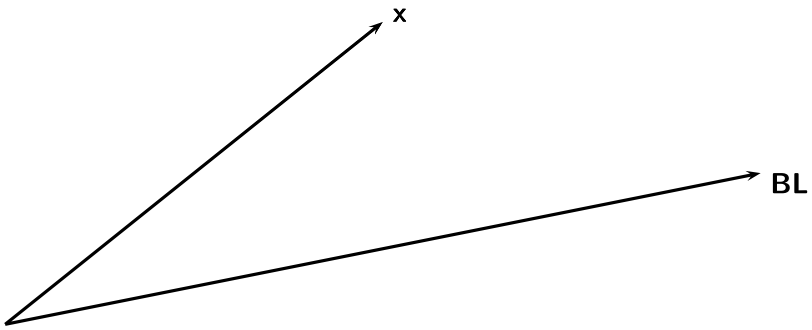
Sinc Sampling and Interpolation

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

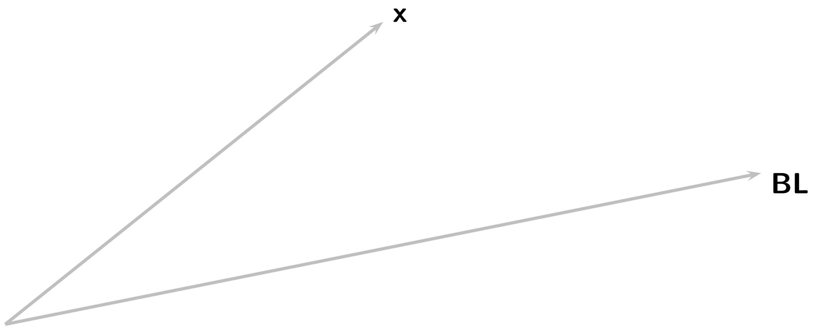
$$\hat{x}(t) = \sum_n x[n] \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)$$



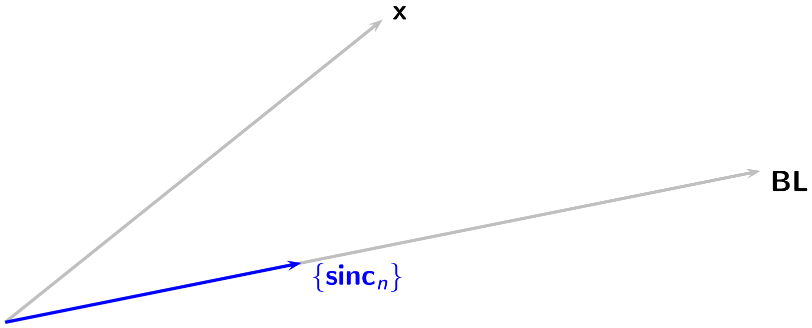
Least squares approximation with sinc sampling and interpolation



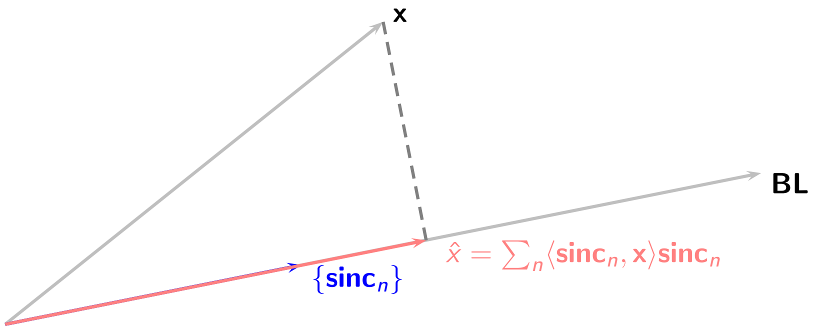
Least squares approximation with sinc sampling and interpolation



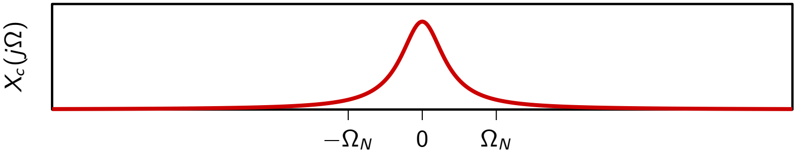
Least squares approximation with sinc sampling and interpolation



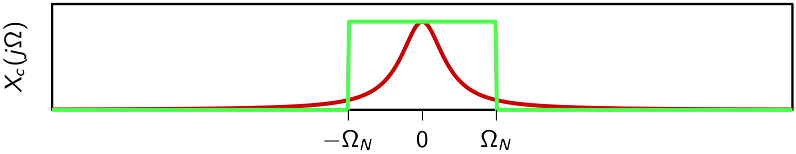
Least squares approximation with sinc sampling and interpolation



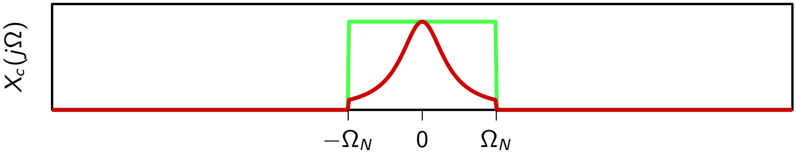
Least squares approximation with sinc sampling and interpolation



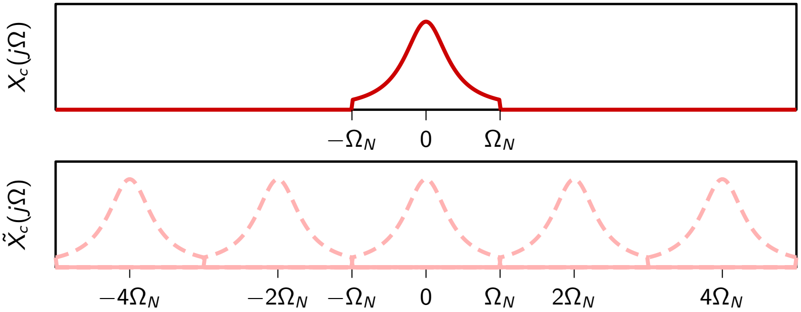
Least squares approximation with sinc sampling and interpolation



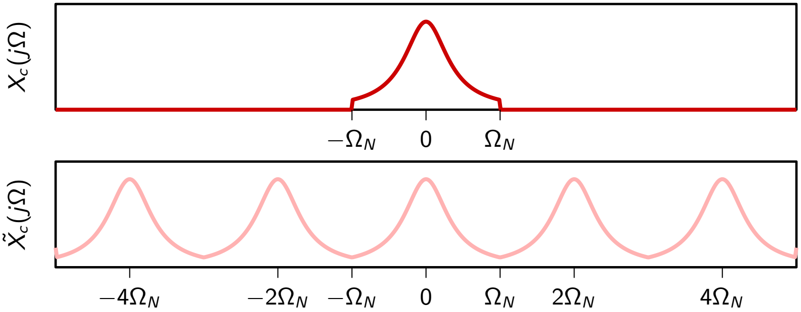
Least squares approximation with sinc sampling and interpolation



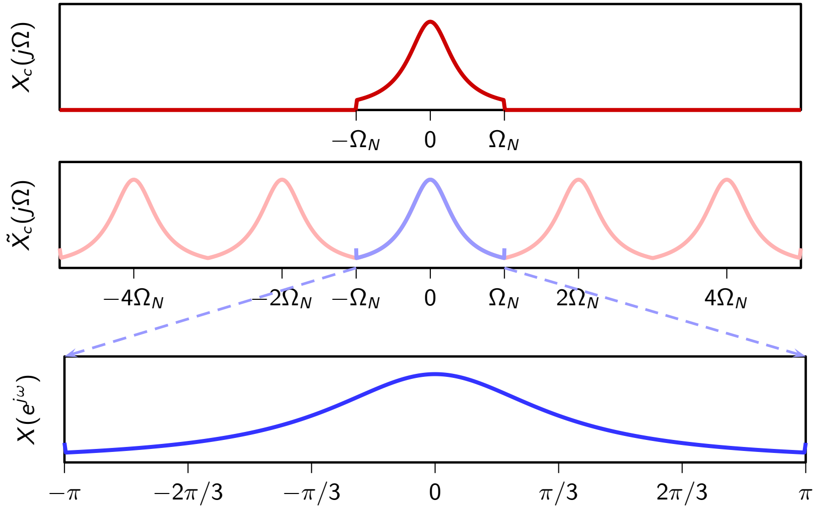
Least squares approximation with sinc sampling and interpolation



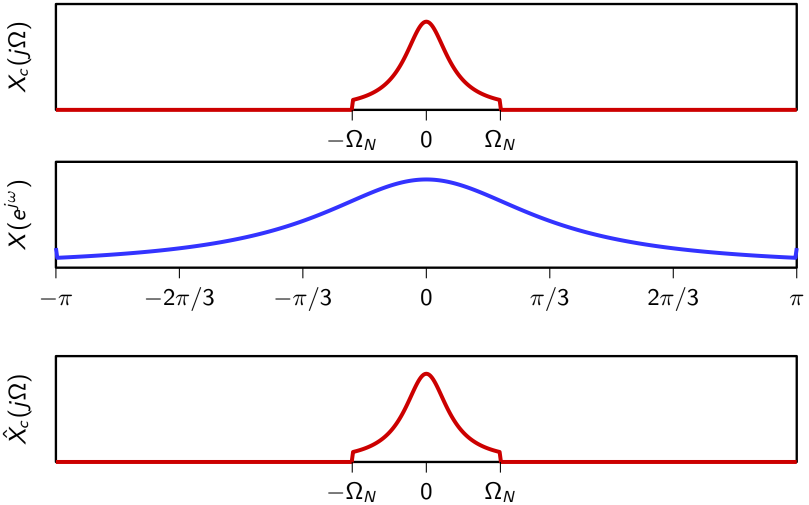
Least squares approximation with sinc sampling and interpolation



Least squares approximation with sinc sampling and interpolation



Least squares approximation with sinc sampling and interpolation



processing of analog signals

Overview:

- Impulse invariance
- Duality
- Examples

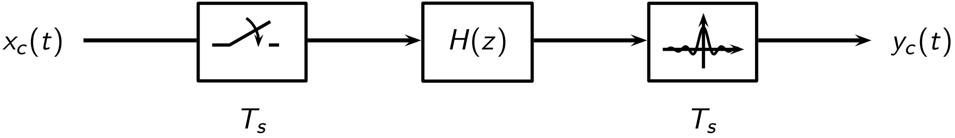
Overview:

- Impulse invariance
- Duality
- Examples

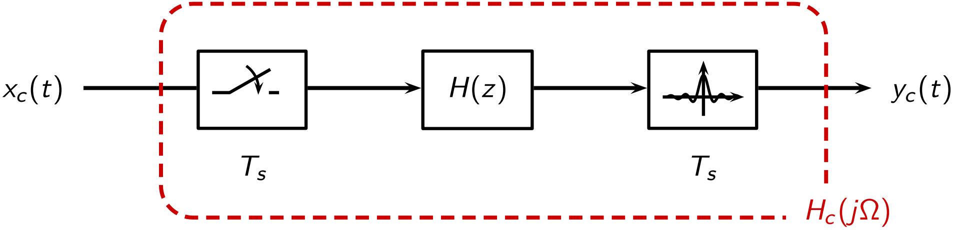
Overview:

- Impulse invariance
- Duality
- Examples

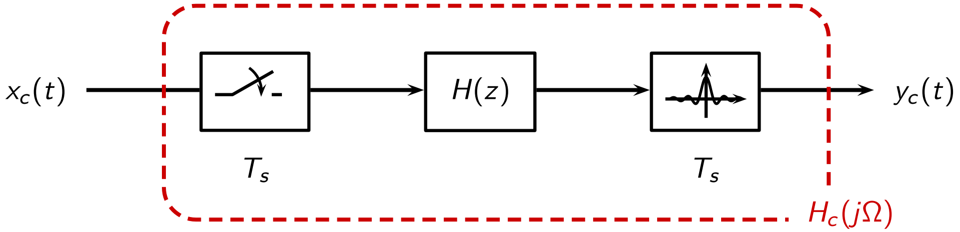
Basic setup



Basic setup

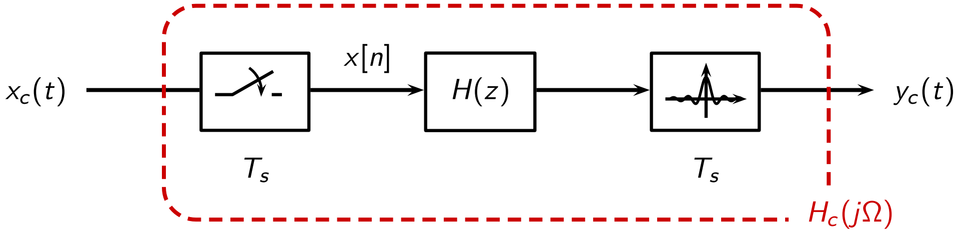


Basic setup



assume $x_c(t)$ is Ω_N-bandlimited:

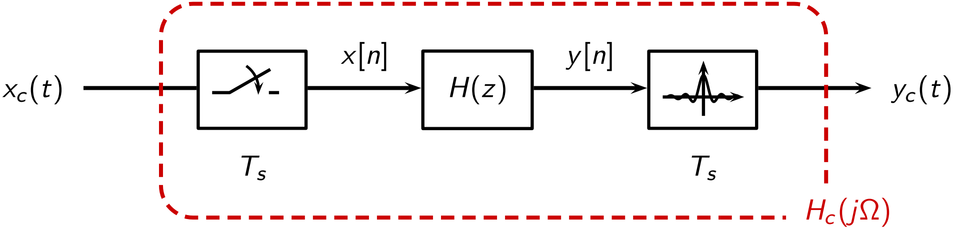
Basic setup



assume $x_c(t)$ is Ω_N-bandlimited:

- $X(e^{j\omega}) = \frac{1}{T_s} X_c\left(j\frac{\omega}{T_s}\right)$

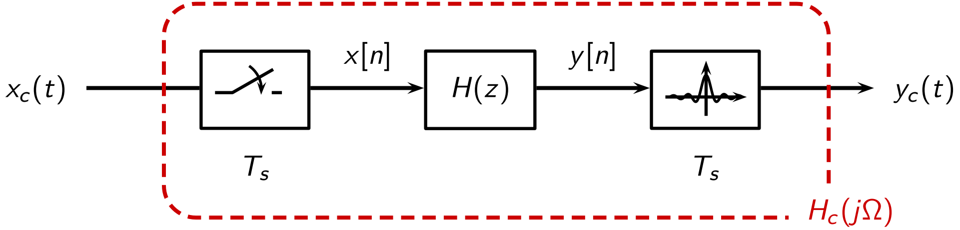
Basic setup



assume $x_c(t)$ is Ω_N-bandlimited:

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

Basic setup



assume $x_c(t)$ is Ω_N-bandlimited:

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

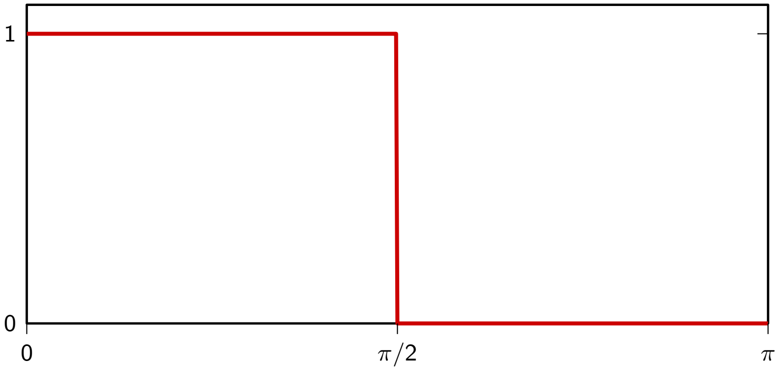
Impulse invariance

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

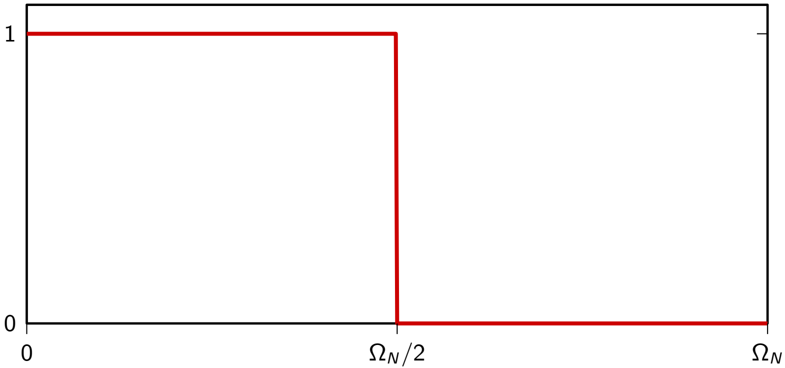
Impulse invariance

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

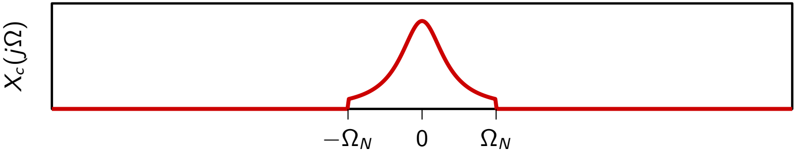
Impulse invariance



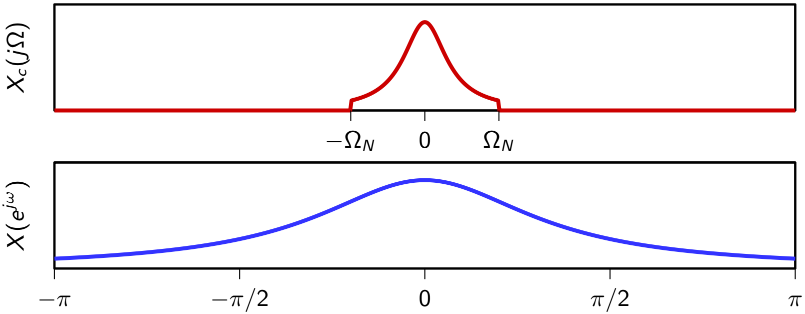
Impulse invariance



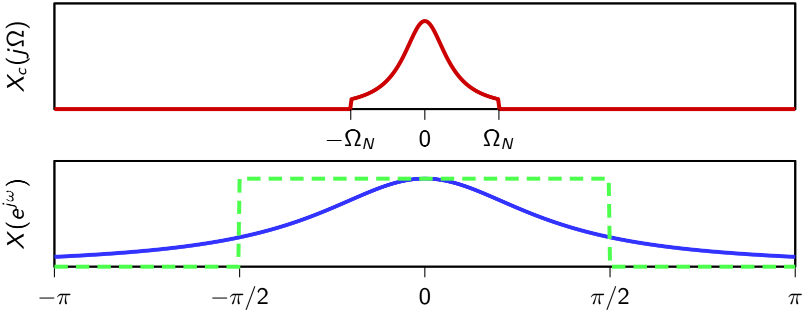
DT processing of CT signals



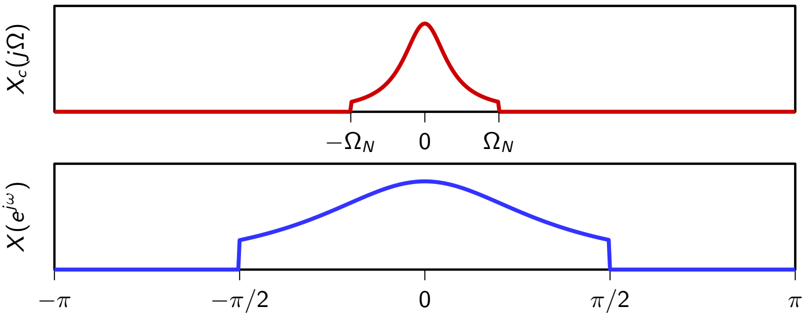
DT processing of CT signals



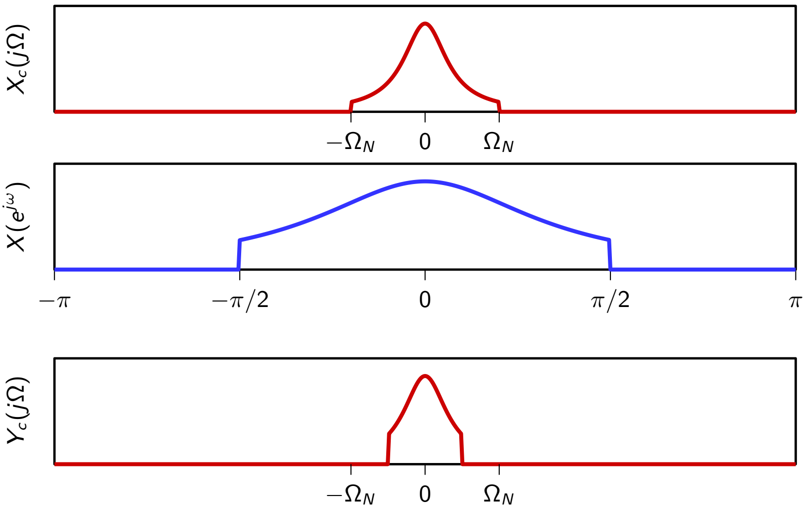
DT processing of CT signals



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

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

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

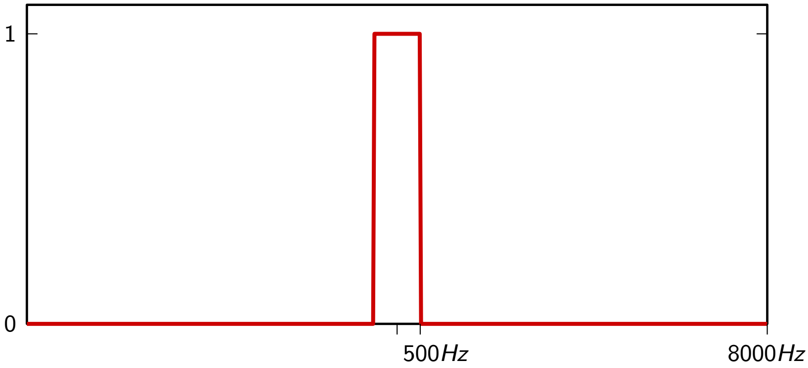
Example

- 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

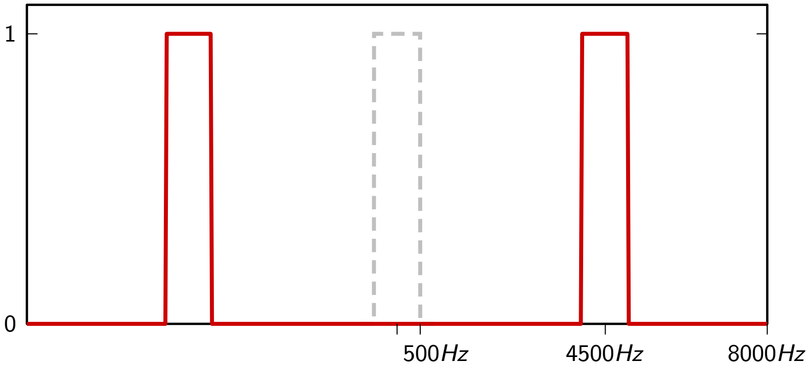
Example

- 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

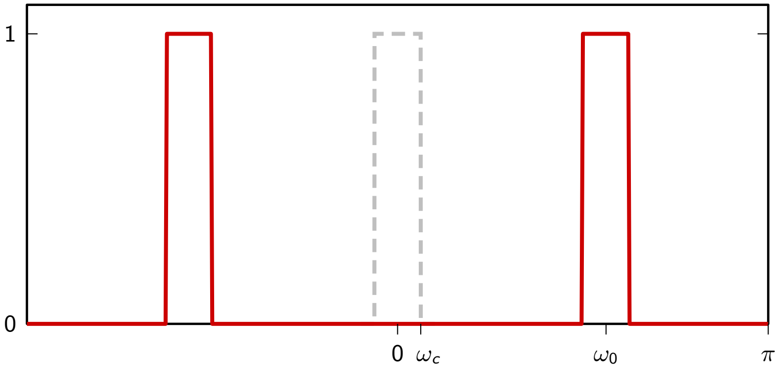
Impulse invariance



Impulse invariance



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 $2\cos\omega_0 n$

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 $2 \cos \omega_0 n$

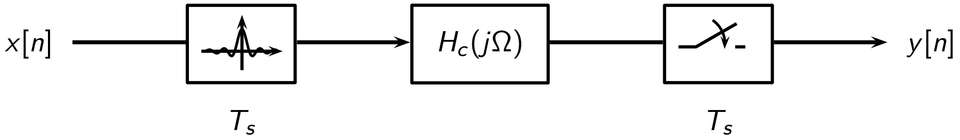
Example

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

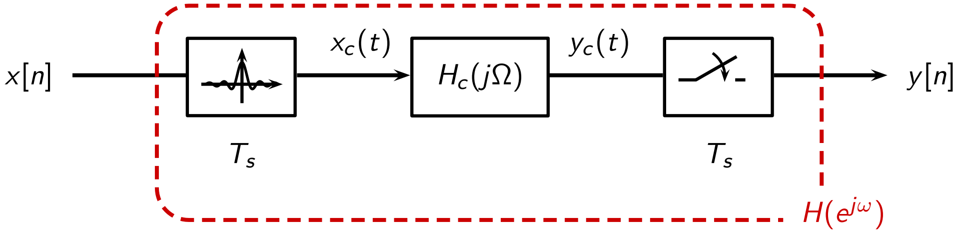
Example

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

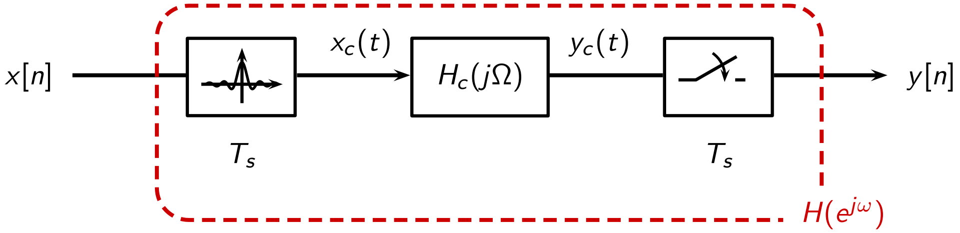
Duality



Duality

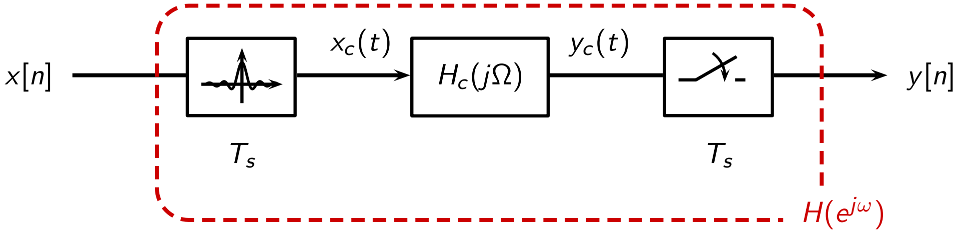


Duality



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

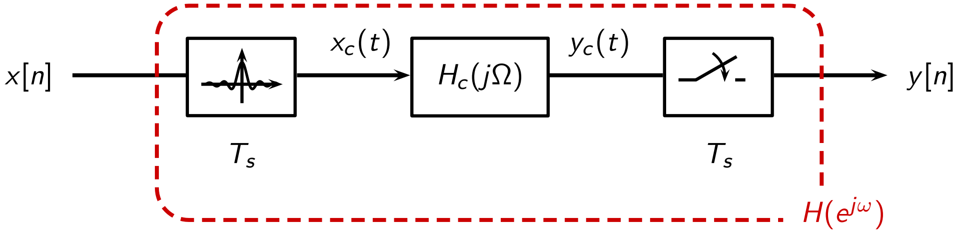
Duality



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

- $X_c(j\Omega) = X(e^{j\Omega})$

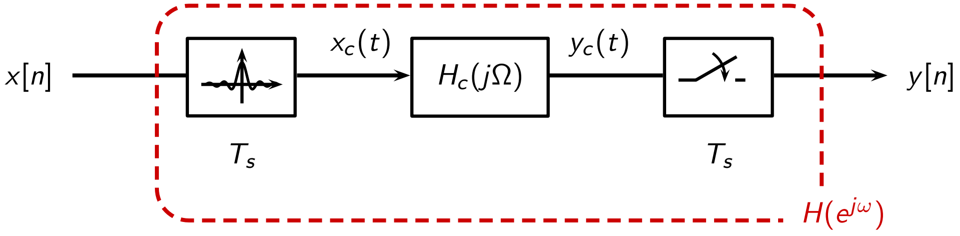
Duality



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

- $X_c(j\Omega) = X(e^{j\Omega})$
- $Y_c(j\Omega) = X_c(j\Omega)H_c(j\Omega)$

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 $\Rightarrow Y(e^{j\omega}) = Y_c(j\omega)$

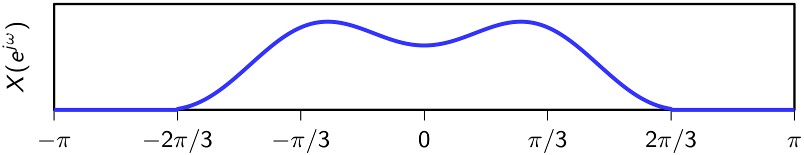
Duality

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

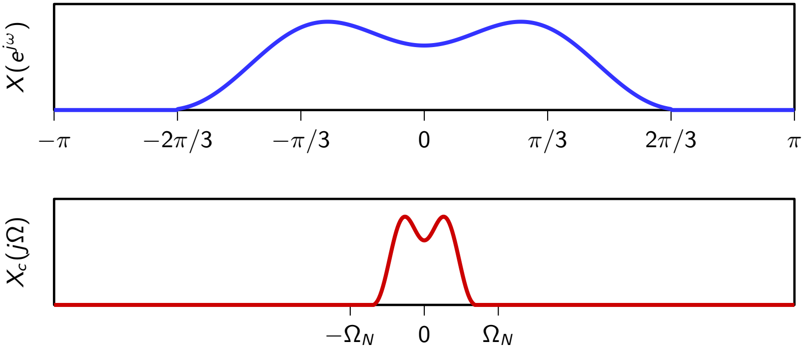
Duality

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

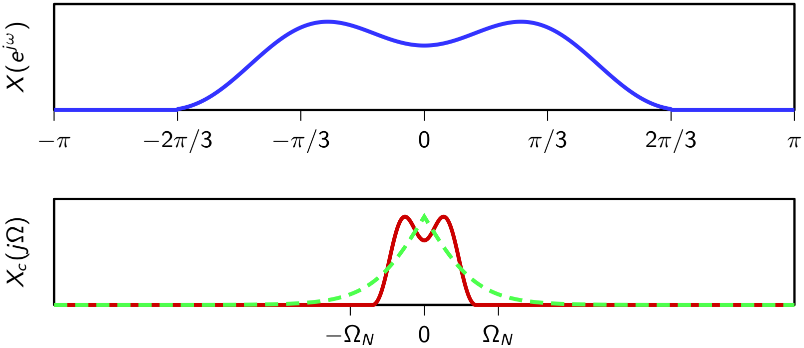
CT processing of DT signals



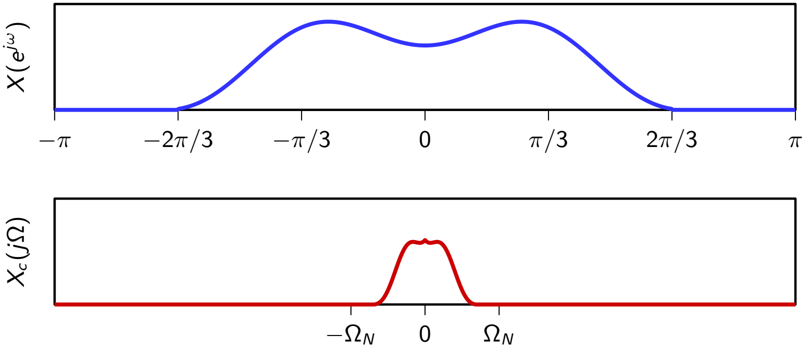
CT processing of DT signals



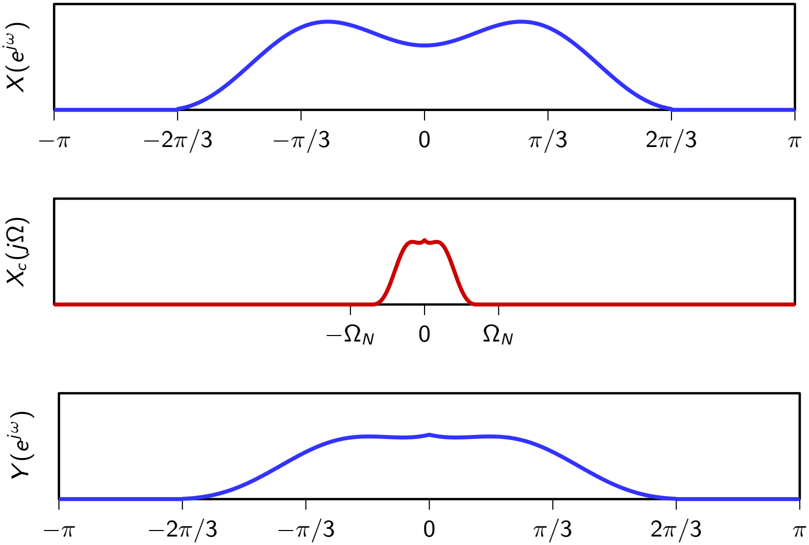
CT processing of DT signals



CT processing of DT signals



CT processing of DT signals

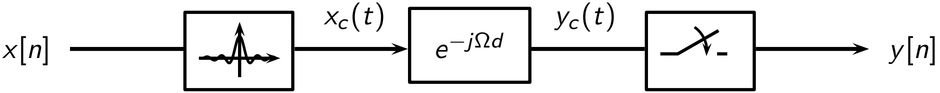


Example: fractional delay

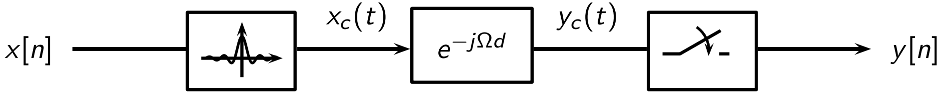
$$H(e^{j\omega}) = e^{-j\omega d}$$

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

By duality

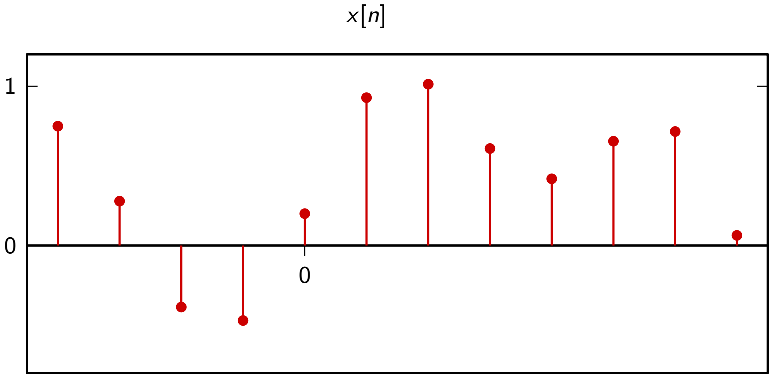


By duality

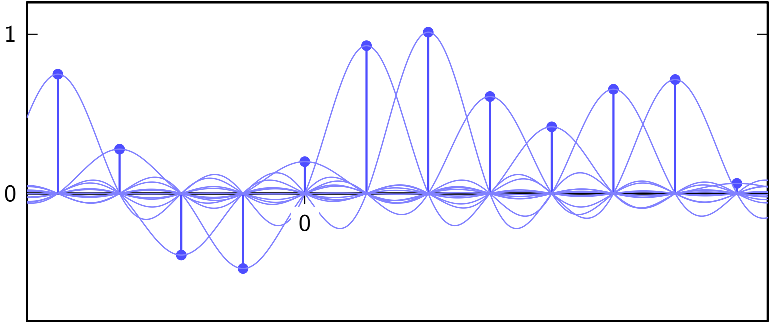


- $Y_c(j\Omega) = e^{-j\Omega d} X_c(j\Omega)$
- $y_c(t) = x_c(t - d)$
- $y[n]$ is the sampled interpolation of $x[n]$ delayed by d

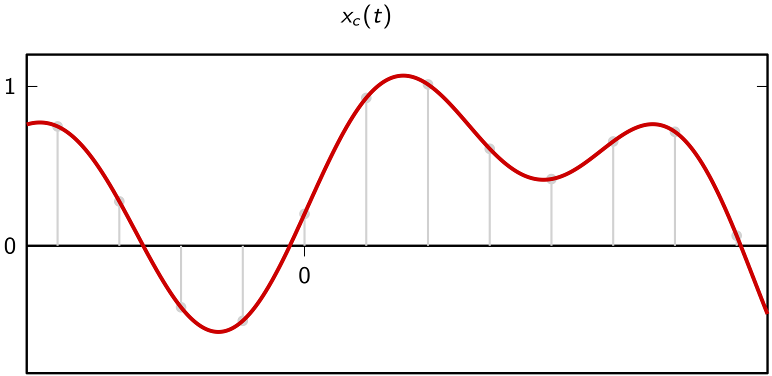
Example: fractional delay



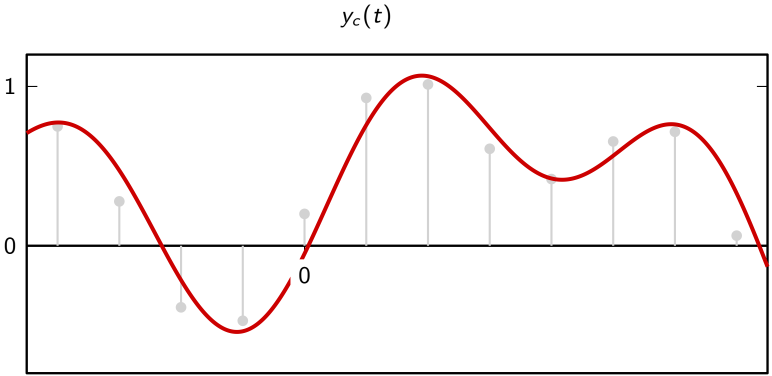
Example: fractional delay



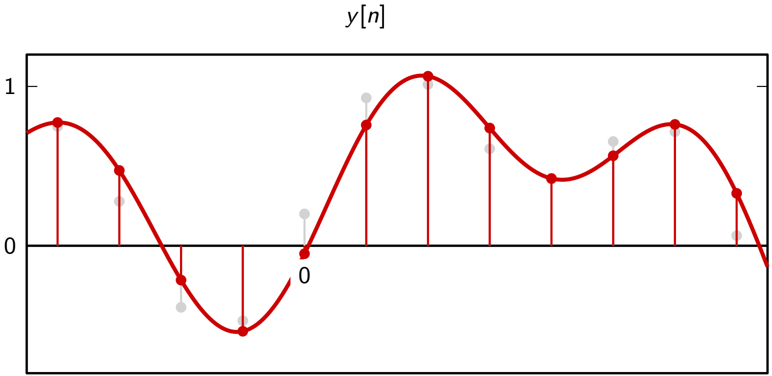
Example: fractional delay



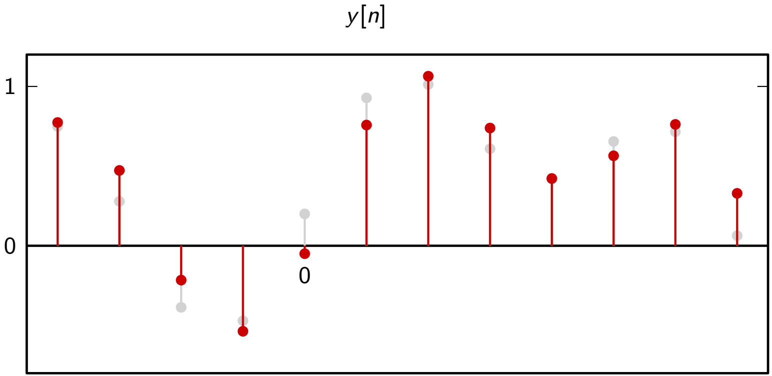
Example: fractional delay



Example: fractional delay



Example: fractional delay



Example: fractional delay

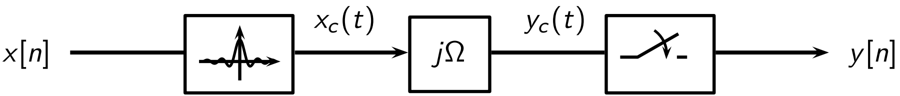
- to delay a discrete-time signal by a fraction of a sample we need an ideal filter!
- efficient approximations exist (e.g. cubic local interpolation)

Example: differentiator

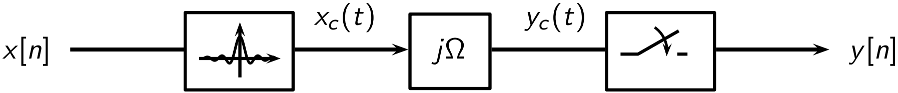
$$H(e^{j\omega}) = j\omega$$

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

By duality



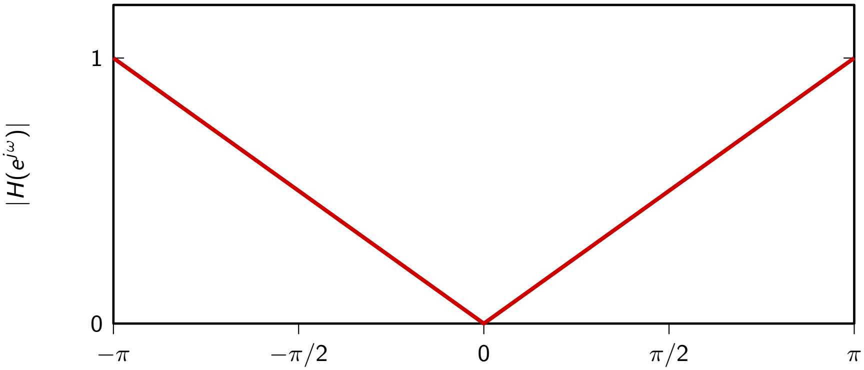
By duality



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

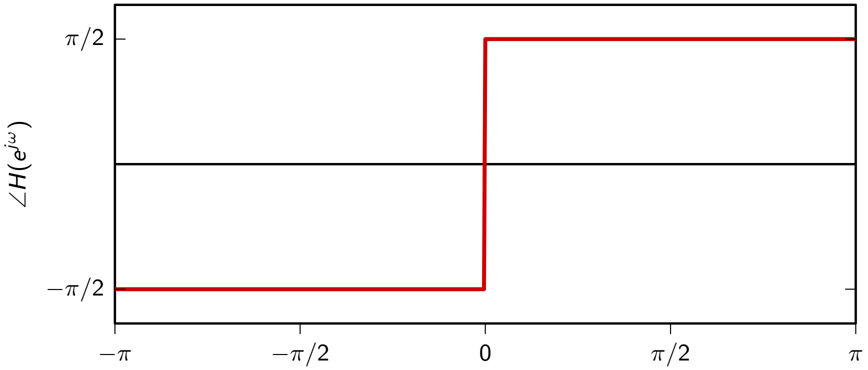
Digital differentiator, magnitude response

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



Digital differentiator, phase response

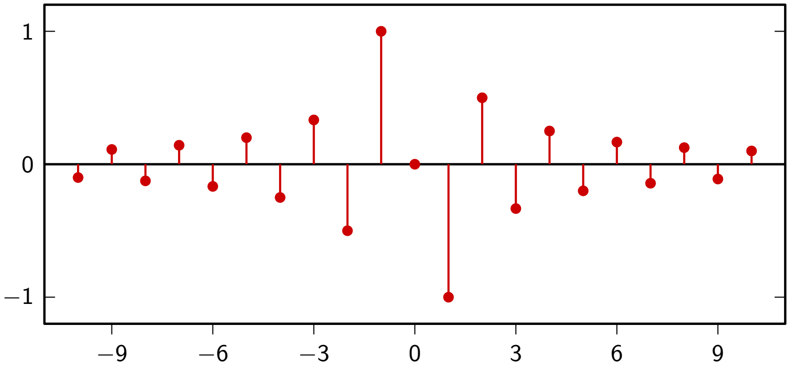
$$H(e^{j\omega}) = j\omega$$



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!