Sampling, Reconstruction, and DT Filtering

Jose Krause Perin

Stanford University

July 9, 2017

Last lecture

- The frequency response of a system tell us how much each frequency is scaled (magnitude response), and delayed (phase response) by the system.
- Poles increase magnitude and introduce phase lag (positive group delay)
- Zeros decrease the magnitude and introduce phase lead (negative group delay)
- All-pass systems have constant magnitude response. For each pole at e_k, there will be a zero at the conjugate reciprocal $1/e_k^*$
- Minimum phase systems have all zeros inside the unit circle
- Any system $H(z)$ can be decomposed into a cascade of a minimum phase system and an all-pass system $H(z) = H_{min}(z)H_{ap}(z)$
- For minimum phase systems, the phase response is given by the Hilbert transform of the log-magnitude response.
- The phase response of a generalized linear phase systems is an affine function of ω
- FIR systems are linear phase as long as their impulse response is either even or odd symmetric
- Linear phase rational IIR systems do not exist

Today's lecture

1. Sampling (continuous-to-discrete time conversion)
2. Reconstruction (discrete-to-continuous time conversion)
3. Discrete-time filtering of continuous-time signals

Digital processing of analog signals

Typical system

Analog-to-digital converter (ADC)

- Performs filtering, sampling, and quantization
- Sampling rate may be of tens of kHz (audio processing), or it may be of tens of GHz (optical communications)

Digital signal processor

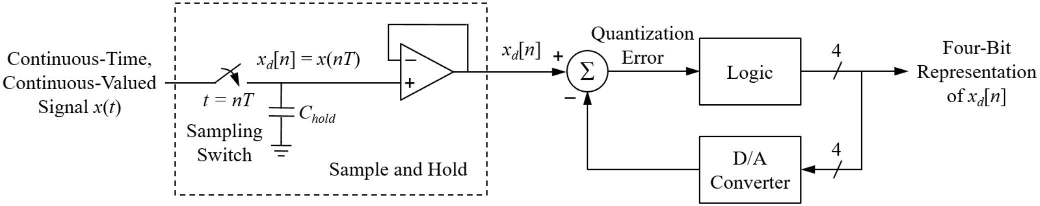
- Performs some operation e.g., filtering, FFT, etc
- May be implemented on PCs with 64-bit floating-point precision, or on ASICs with limited arithmetic precision (e.g., 6 bits).

Digital-to-analog converter (DAC)

- Performs quantization and reconstruction (filtering)
- Sampling rate could be similar to ADC

In practice

Example of successive-approximation analog-to-digital converter (SA-ADC)



Once every T seconds, the logic performs a successive approximation of $x_d[n]$, starting at the MSB, flipping bits to minimize the quantization error.

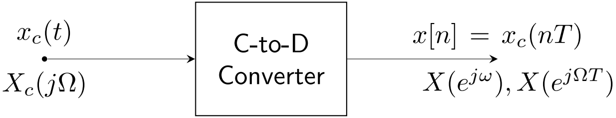
Taken from the lecture notes of EE 102B by Prof. Joseph Kahn

More about ADCs in EE 315: Analog-Digital Interface Circuits.

Analog-to-digital conversion

In this class

We'll model the ADC as an ideal continuous-to-discrete (C-to-D) time converter.



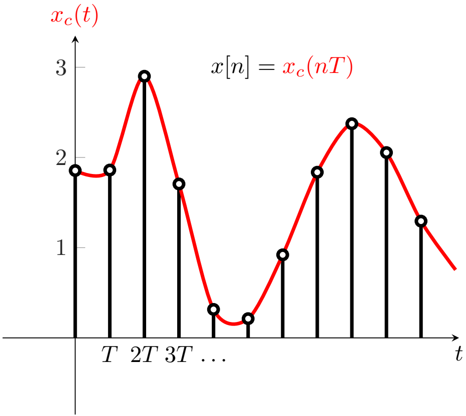
Notation:

$X_c(j\Omega)$ denotes the Fourier transform of the continuous-time signal $x_c(t)$, where Ω is the continuous-time frequency.

$X(e^{j\omega})$ denotes the discrete-time Fourier transform of a discrete-time signal $x[n]$, where ω is the normalized frequency.

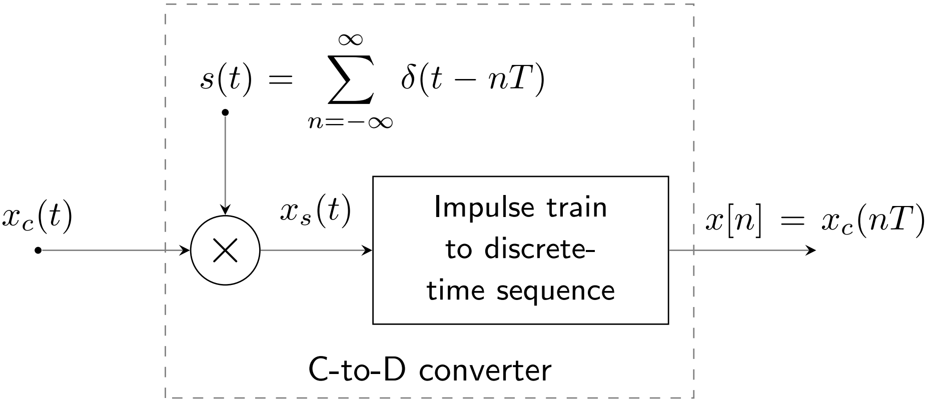
Continuous-to-discrete time conversion

The C-to-D converter simply samples the continuous-time signal every T seconds, where T is the **sampling period**.

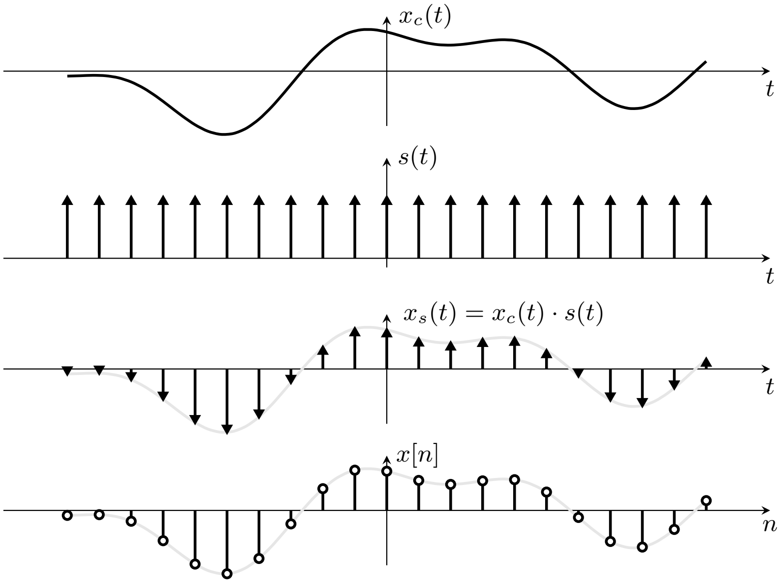


Question: How are $x_c(t)$ and $x[n]$ related in the frequency domain? That is, how to obtain the discrete-time Fourier transform $X(e^{j\omega})$ from the continuous-time Fourier transform $X_c(j\Omega)$?

Impulse sampling interpretation:
We can think of the C-to-D converter as multiplication by an **impulse train**, followed by an impulse-to-sequence converter.
This representation is purely for mathematical convenience.

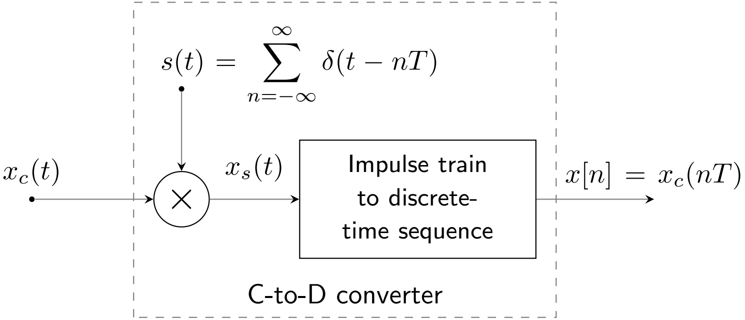


Impulse sampling example



Continuous-to-discrete time conversion

Question: How to obtain the discrete-time Fourier transform $X(e^{j\omega})$ from the continuous-time Fourier transform $X_c(j\Omega)$?



We'll calculate the continuous-time (CT) Fourier transform of $x_s(t)$ in two different ways. First, we'll calculate $X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega)$. Then, we'll calculate $X_s(j\Omega) = \mathcal{F}\{x_s(t)\}$. We'll use these two equations of $X_s(j\Omega)$ to obtain an equation for $X(e^{j\omega})$, the DTFT of $x[n]$.

Starting with $X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega)$, recall that the Fourier transform of the impulse train is given by

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \Longleftrightarrow S(j\Omega) = \sum_{k=-\infty}^{\infty} \frac{2\pi}{T} \delta\Big(\Omega - k\frac{2\pi}{T}\Big)$$

Now we can calculate $X_s(j\Omega)$:

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega)$$

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * \left(\sum_{k=-\infty}^{\infty} \frac{2\pi}{T} \delta\Big(\Omega - k\frac{2\pi}{T}\Big)\right)$$

$$\boxed{X_s(j\Omega) = \sum_{k=-\infty}^{\infty} \frac{1}{T} X_c\Big(j\Big(\Omega - k\frac{2\pi}{T}\Big)\Big)} \quad (1)$$

Note that $X_s(j\Omega)$ is equal to $X_c(j\Omega)$ scaled by $1/T$ and repeated every $2\pi/T$, which we define as the **sampling frequency** $\Omega_s \equiv 2\pi/T$

Now let's calculate $X_s(j\Omega) = \mathcal{F}\{x_s(t)\}$, where $x_s(t) = x_c(t) \cdot s(t)$:

$$\begin{aligned}
X_s(j\Omega) = \mathcal{F}\{x_c(t) \cdot s(t)\} &= \mathcal{F}\left\{ \sum_{n=-\infty}^{\infty} x_c(nT)\delta(t-nT) \right\} \\
&= \mathcal{F}\left\{ \sum_{n=-\infty}^{\infty} x[n]\delta(t-nT) \right\} \\
&= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n]\delta(t-nT)e^{-j\Omega t}dt && \text{(CT Fourier transform)} \\
&= \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} \delta(t-nT)e^{-j\Omega t}dt \\
&= \sum_{n=-\infty}^{\infty} x[n]e^{-jn\Omega T} && \text{(recall: } X(e^{j\omega}) \equiv \textstyle\sum_{n=-\infty}^{\infty} x[n]e^{-jn\omega}\text{)}
\end{aligned}$$

$$\boxed{X_s(j\Omega) = X(e^{j\omega})\Big|_{\omega=\Omega T}} \tag{2}$$

$X_s(j\Omega)$ is equal to $X(e^{j\omega})$ evaluated at $\omega = \Omega T$, or equivalently $X(e^{j\omega})$ is equal to $X_s(j\Omega)$ evaluated at $\Omega = \omega/T$.

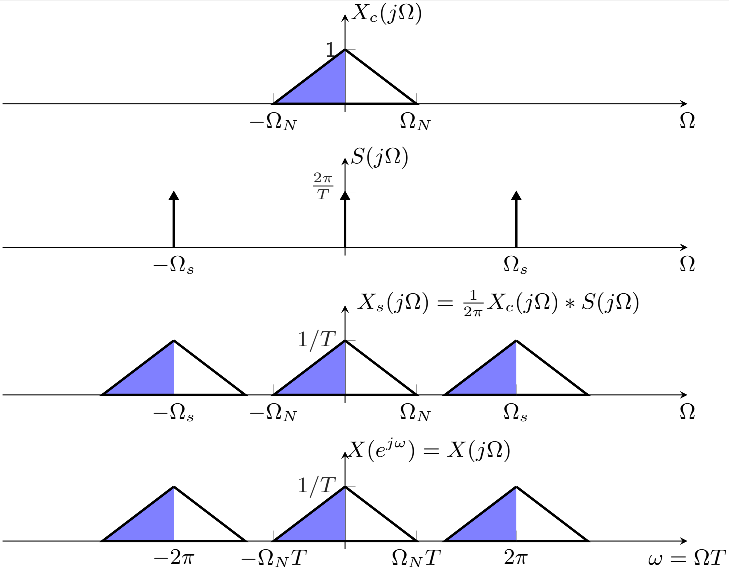
Substituting (1) in (2):

$$X(e^{j\Omega T}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c\Big(j\Big(\Omega - k\frac{2\pi}{T}\Big)\Big),$$

where we used the relation $\omega = \Omega T$.

- T is the **sampling period**, and $\Omega_s = \frac{2\pi}{T}$ is the **sampling frequency**
- This equation shows that in discrete time ($\omega = \Omega T$) replicas of the original spectrum appear with period 2π

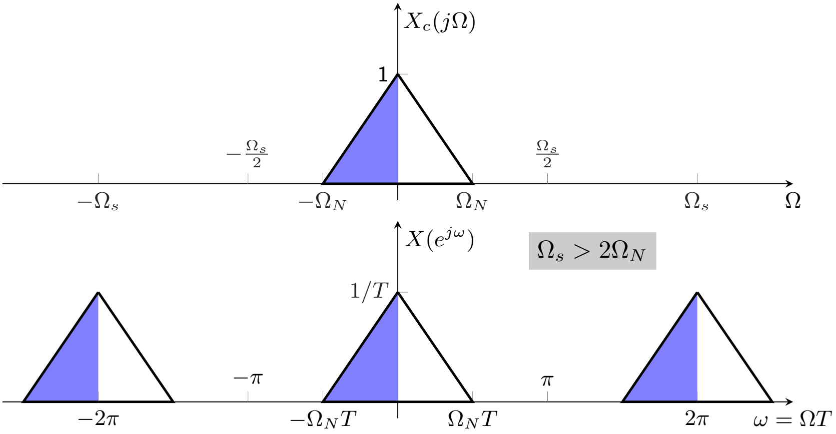
Graphically



Replicas of the original spectrum appear with period 2π

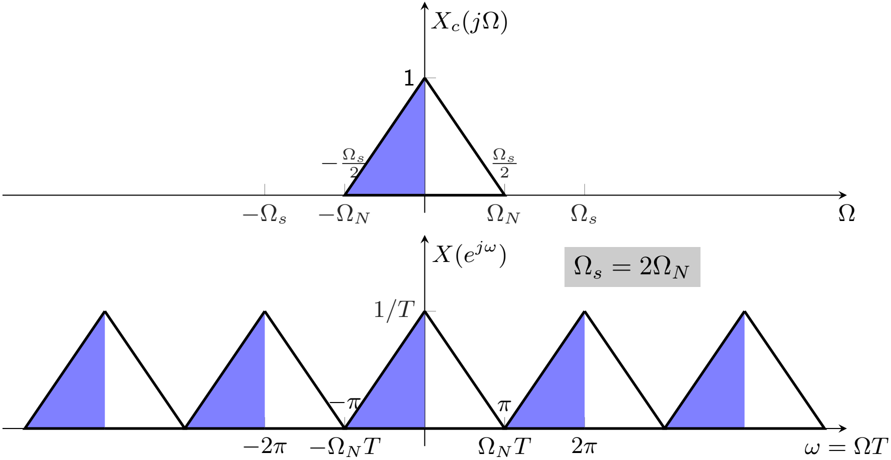
Oversampling

- A signal is **band limited** if $X_c(j\Omega) = 0$ for $|\Omega| > \Omega_N$. In this case, the signal has maximum frequency Ω_N and **bandwidth** $2\Omega_N$
- Sampling at $\Omega_s > 2\Omega_N$ is called **oversampling**
- Oversampling leads to gaps between the spectrum replicas



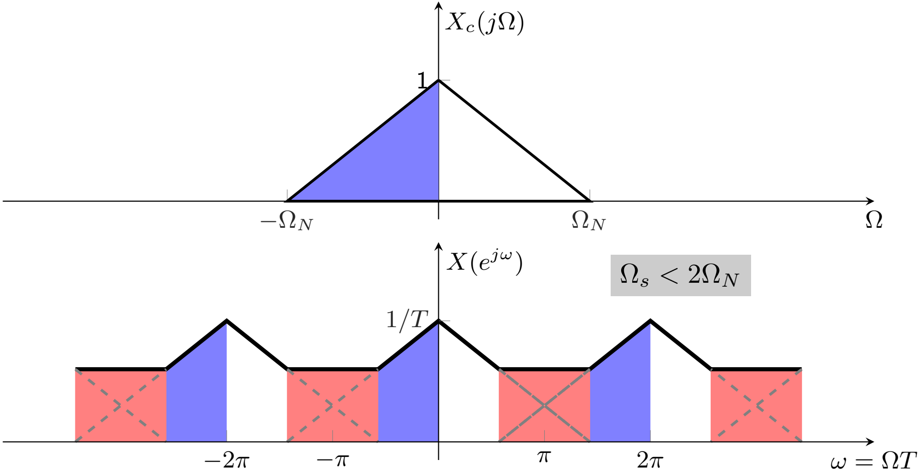
Nyquist sampling

- Sampling at $\Omega_s = 2\Omega_N$ is called **Nyquist sampling**
- Note that if Ω_s is any smaller than $2\Omega_N$, there will be overlapping of the spectrum replicas



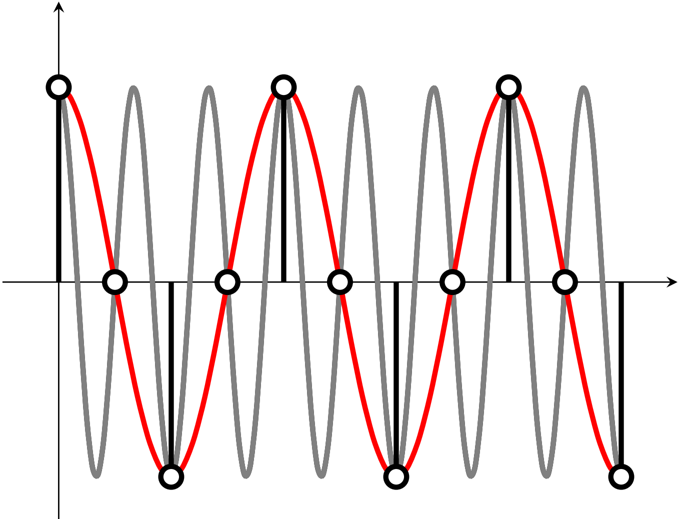
Undersampling

- Undersampling occurs when $\Omega_s < 2\Omega_N$
- In this case, the spectrum replicas overlap
- The overlapping causes **aliasing distortion**



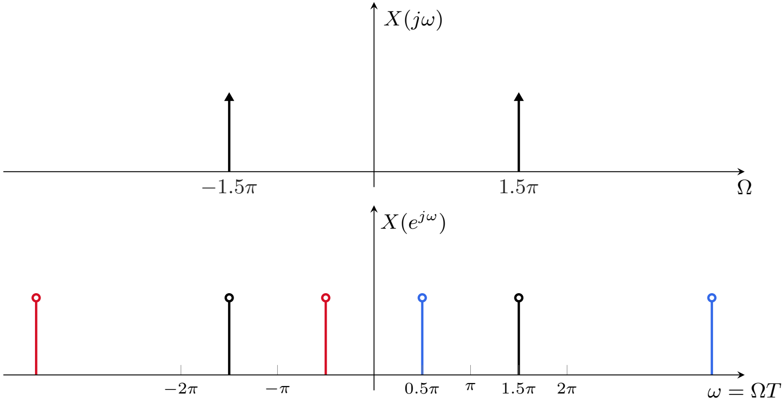
Aliasing: time domain

Samples taken at frequency $\Omega_s = 2\pi$ form an **alias signal** of frequency 0.5π, but **original signal** had frequency 1.5π



Aliasing: frequency domain

Same example, but now in the frequency domain



Blue components correspond to spectrum replica centered at 2π, while red components correspond to spectrum replica centered at -2π.
The final spectrum corresponds to $\cos(0.5\pi n)$

Aliasing examples

Cameras and our own visual system are sampling devices with a certain sampling frequency. Therefore, we can *see* aliasing.

Examples

- This video. The blades of the helicopter are spinning at the same frequency of the camera shutter
- Wheels of the car that appear to spin backwards
- Stroboscopic effect (search videos of this)

Sampling random signals

The same theory applies to random signals. Specifically, we we'll apply the same results to the autocorrelation function and PSD of random signals
Autocorrelation function of a continuous-time WSS process and its PSD:

$$\phi_{x_c x_c}(\tau) = \mathbb{E}(x_c(t+\tau)x_c^*(t)) \Longleftrightarrow \Phi_{x_c x_c}(j\Omega)$$

If we sample the random signal with sampling period $T = 2\pi/\Omega_s$, we obtain $x[n] = x_c(nT)$.
Calculating the autocorrelation function of $x[n]$:

$$\begin{aligned} \phi_{xx}[m] &= \mathbb{E}(x[n+m]x^*[n]) = \mathbb{E}(x_c((n+m)T)x_c^*(nT)) \\ &= \phi_{x_c x_c}(mT) \qquad \text{(the autocorrelation is sampled)} \end{aligned}$$

And for the PSD:

$$\begin{aligned} \Phi_{xx}(e^{j\Omega T}) &= \mathcal{F}(\phi_{xx}[m]) = \sum_{m=-\infty}^{\infty} \phi_{xx}[m]e^{-j\Omega T} \\ &= \frac{1}{T}\sum_{k=-\infty}^{\infty} \Phi_{x_c x_c}(j(\Omega - k\Omega_s)) \qquad \text{(the PSD is replicated with period } \Omega_s\text{)} \end{aligned}$$

Digital processing of analog signals

Typical system

Analog-to-digital converter (ADC)

- Performs filtering, sampling, and quantization
- Sampling rate may be of tens of kHz (audio processing), or it may be of tens of GHz (optical communications)

Digital signal processor

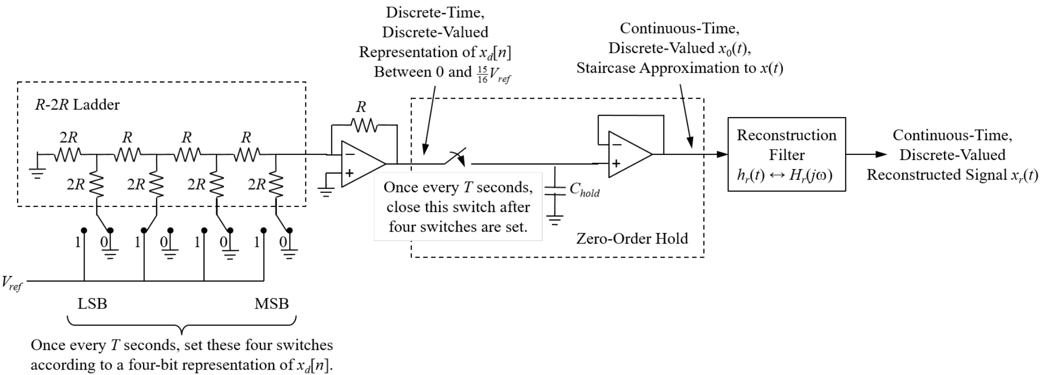
- Performs some operation e.g., filtering, FFT, etc
- May be implemented on PCs with 64-bit floating-point precision, or on ASICs with limited arithmetic precision (e.g., 6 bits).

Digital-to-analog converter (DAC)

- Performs quantization and reconstruction (filtering)
- Sampling rate could be similar to ADC

In practice

Example of digital-to-analog converter (DAC)

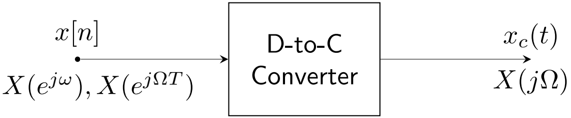


Taken from the lecture notes of EE 102B by Prof. Joseph Kahn

Digital-to-analog conversion

In this class

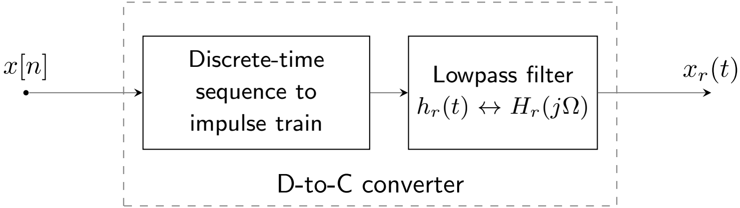
We'll model the ADC as an ideal discrete-to-continuous (D-to-C) time converter.



In essence, a D-to-C converter performs **interpolation**.

Discrete-to-continuous time conversion

For mathematical convenience we can model the D-to-C as

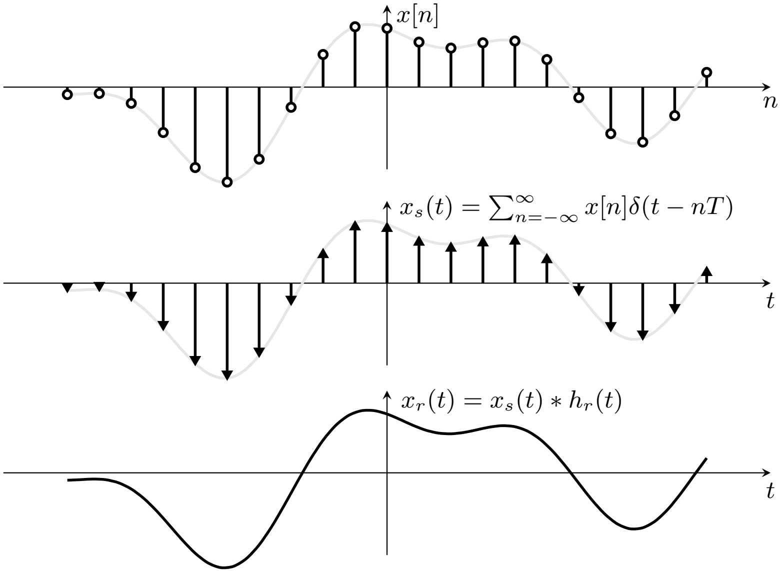


$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n] h_r(t - nT) \qquad \text{(D-to-C converter)}$$

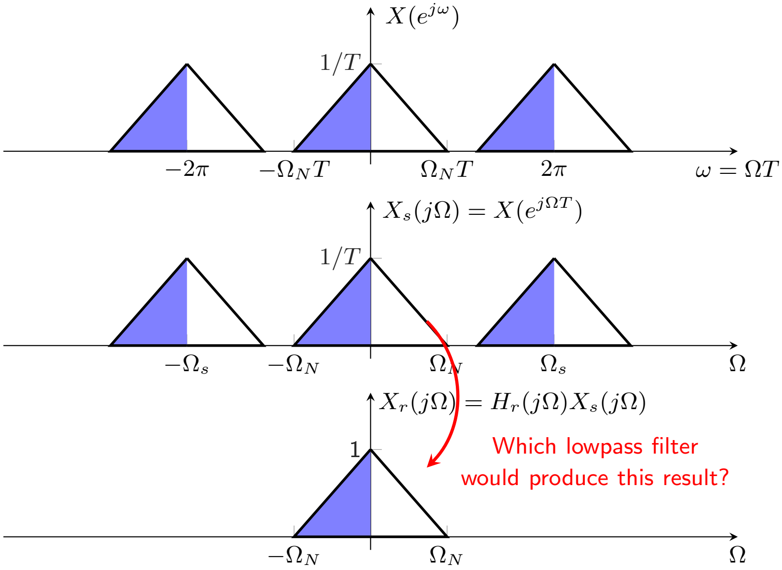
Important questions

1. How close to the original signal is $x_r(t)$?
2. What lowpass filter $H_r(j\Omega)$ will lead to the best performance?

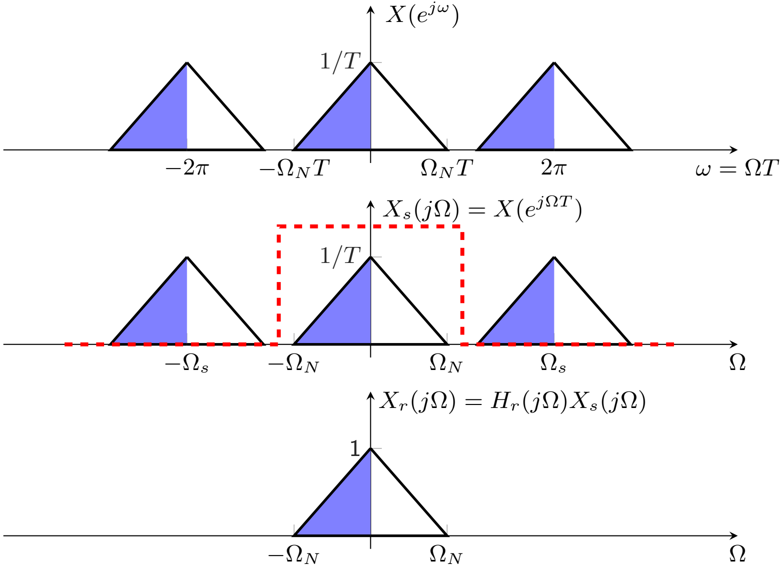
Reconstruction: time domain



Reconstruction: frequency domain



Reconstruction: frequency domain



Shannon-Nyquist sampling theorem

Shannon-Nyquist sampling theorem

A band-limited signal with highest frequency Ω_N can be **perfectly reconstructed** from samples taken with sampling frequency $\Omega_s = \frac{2\pi}{T} > 2\Omega_N$.

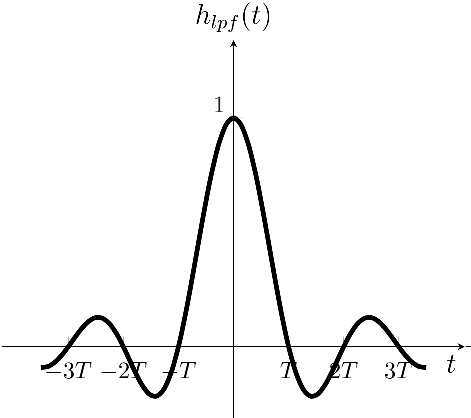
$$X_r(j\Omega) = H_r(j\Omega)X(e^{j\Omega T}) = X_c(j\Omega)$$

- Sampling above the Nyquist frequency $(2\Omega_N)$ avoids aliasing
- In practice, it is common to use an **anti-aliasing filter** to minimize aliasing when the analog signal is not band-limited.
- Perfect reconstruction is achieved if $H_r(j\Omega)$ is the ideal lowpass filter. In other words, the ideal lowpass filter (or sinc function in time domain) is the perfect *interpolator* for band-limited signals.

Ideal lowpass filter

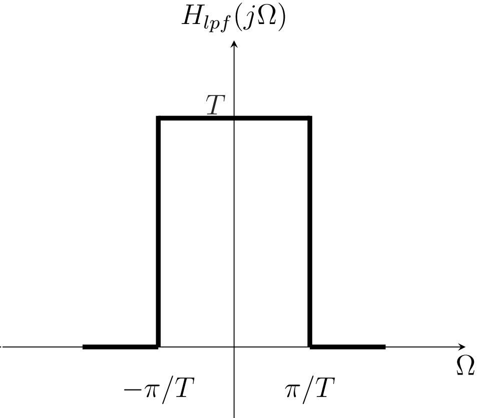
Time domain

$$h_{lpf}(t) = \frac{\sin \frac{\pi}{T} t}{\frac{\pi}{T} t} = \mathrm{sinc}\left(\frac{t}{T}\right)$$



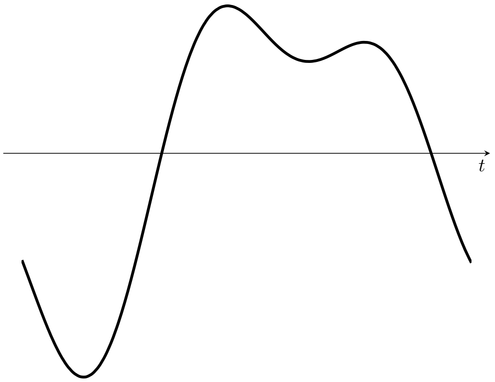
Frequency domain

$$H_{lpf}(j\Omega) = \begin{cases} T, & |\Omega| \le \frac{\pi}{T} \\ 0, & |\Omega| > \frac{\pi}{T} \end{cases}$$



Example of reconstruction with an ideal lowpass filter

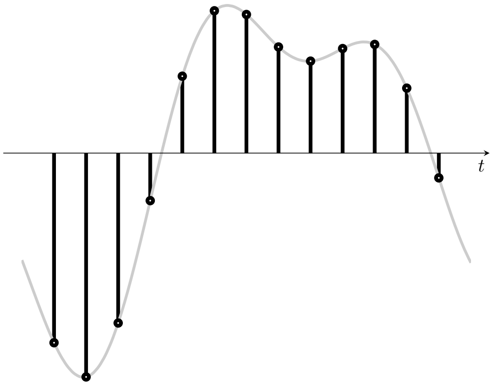
$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n] h_r(t - nT) = \sum_{n=-\infty}^{\infty} x[n] \text{sinc}(t - nT) \qquad \text{(reconstruction)}$$



Original continuous-time signal

Example of reconstruction with an ideal lowpass filter

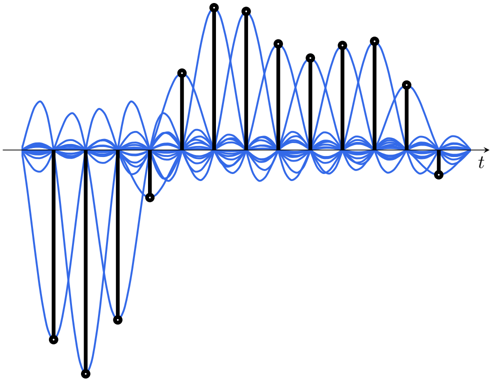
$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]h_r(t-nT) = \sum_{n=-\infty}^{\infty} x[n]\mathrm{sinc}(t-nT) \qquad \text{(reconstruction)}$$



Samples from original continuous-time signal

Example of reconstruction with an ideal lowpass filter

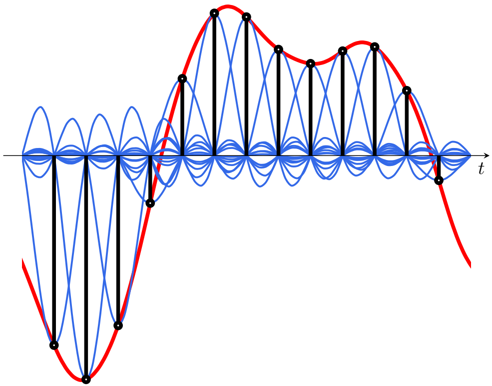
$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]h_r(t - nT) = \sum_{n=-\infty}^{\infty} x[n]\mathrm{sinc}(t - nT) \qquad \text{(reconstruction)}$$



At the nth sample, we have the sinc function $x[n]\mathrm{sinc}(t - nT)$

Example of reconstruction with an ideal lowpass filter

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]h_r(t-nT) = \sum_{n=-\infty}^{\infty} x[n]\text{sinc}(t-nT) \qquad \text{(reconstruction)}$$



The sum of all **sincs** results in the perfectly **reconstructed signal**.

Practical reconstruction

Problem

The ideal lowpass filter is not feasible, as it is non-causal and requires infinitely many samples.

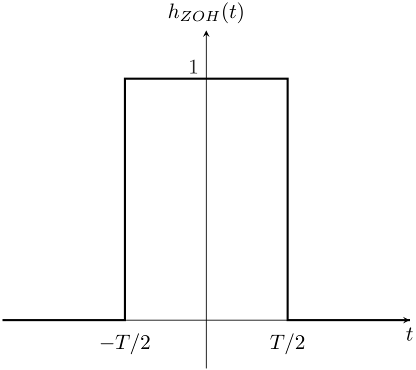
Common reconstruction filters

1. Zero-order hold (square pulse)
2. Linear interpolation (triangular pulse)
3. Cubic spline interpolation

Practical reconstruction: zero-order holder (ZOH)

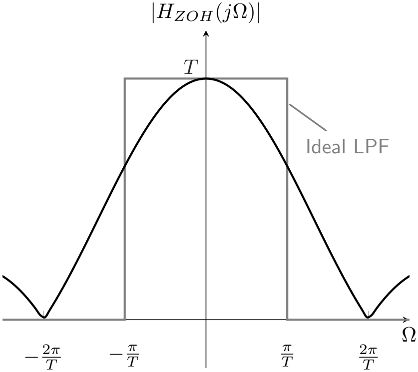
Impulse response

$$h_{ZOH}(t) = \begin{cases} 1, & -T/2 \leq t \leq T/2 \\ 0, & \text{otherwise} \end{cases}$$



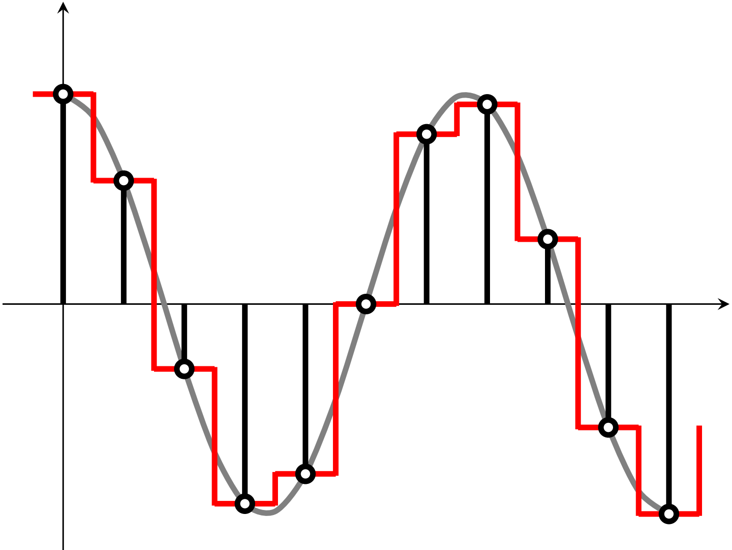
Frequency response

$$\begin{aligned} H_{ZOH}(j\Omega) &= T\frac{\sin(\Omega T/2)}{\Omega T/2} \\ &= T\,\text{sinc}\left(\frac{\Omega}{2\pi}T\right) \end{aligned}$$



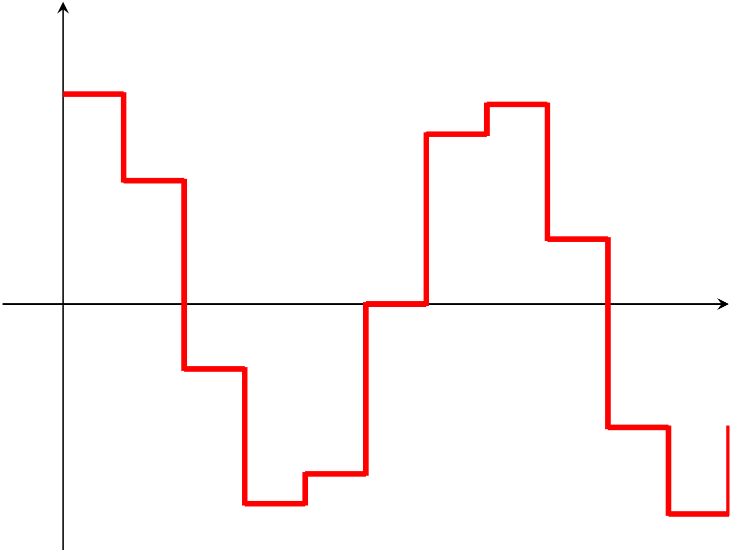
Practical reconstruction: zero-order holder (ZOH)

Example of reconstruction (or interpolation) using the ZOH



Practical reconstruction: zero-order holder (ZOH)

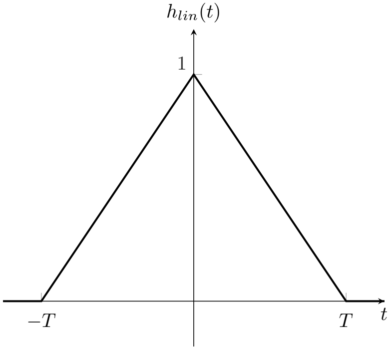
Example of reconstruction (or interpolation) using the ZOH



Practical reconstruction: linear interpolator

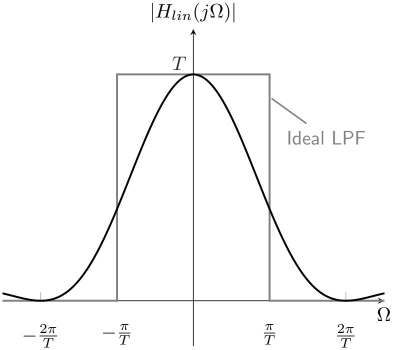
Impulse response

$$h_{lin}(t) = \frac{1}{T} h_{ZOH}(t) * h_{ZOH}(t)$$
$$= \begin{cases} 1 - |t|/T, & |t| < T/2 \\ 0, & \text{otherwise} \end{cases}$$



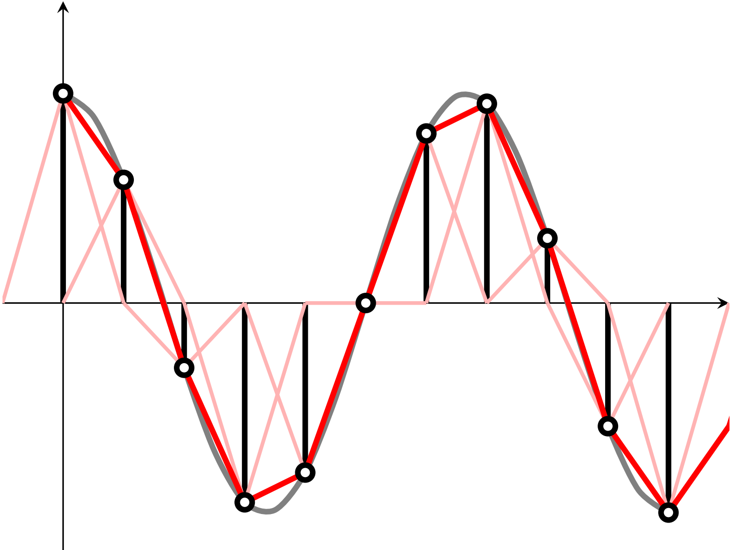
Frequency response

$$H_{lin}(j\Omega) = \frac{1}{T} H^2_{ZOH}(j\Omega)$$
$$= T \mathrm{sinc}^2\left(\frac{\Omega}{2\pi} T\right)$$



Practical reconstruction: linear interpolation

Example of reconstruction using linear interpolation



Practical reconstruction: cubic-spline interpolation

Impulse response

$$h_{spline}(t) = \begin{cases} (a+2)|t/T|^3 - (a+3)|t/T|^2 + 1, & 0 \leq |t| \leq T \\ a|t/T|^3 - 5a|t/T|^2 + 8a|t/T| - 4a, & T < |t| \leq 2T \\ 0, & \text{otherwise} \end{cases}$$

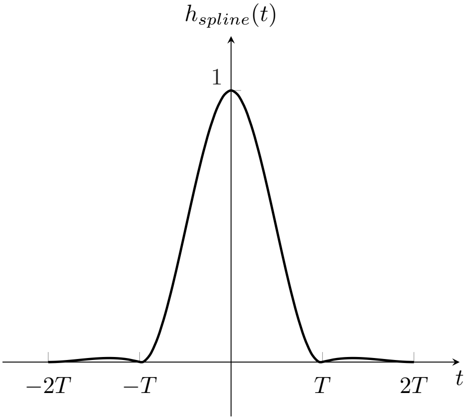
Frequency response

$$\begin{aligned} H_{spline}(j\Omega) &= \frac{12T}{(\Omega T)^2}\Big(\mathrm{sinc}^2(fT) - \mathrm{sinc}(2fT)\Big) \\ &+ \frac{8Ta}{(\Omega T)^2}\Big(3\mathrm{sinc}^2(2fT) - 2\mathrm{sinc}(2fT) - \mathrm{sinc}(4fT)\Big) \end{aligned}$$

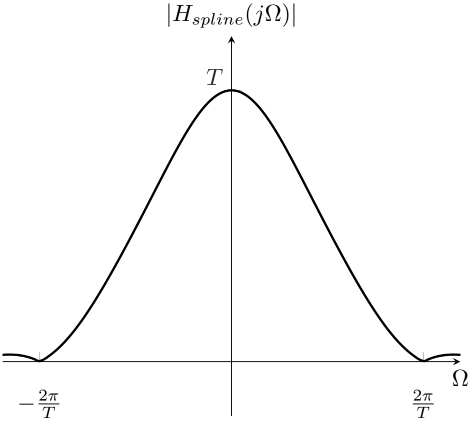
where $f = \Omega/(2\pi)$.

Practical reconstruction: cubic-spline interpolation

Impulse response



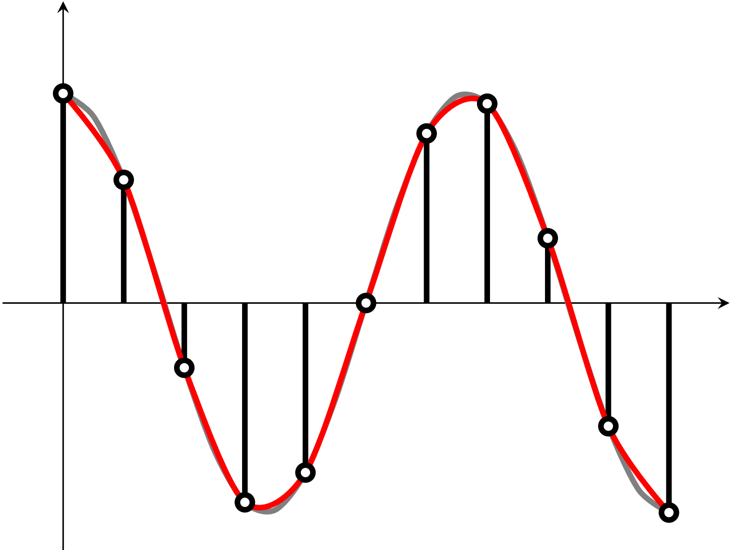
Frequency response



This assumes $a = 0.1$ in the previous slide.

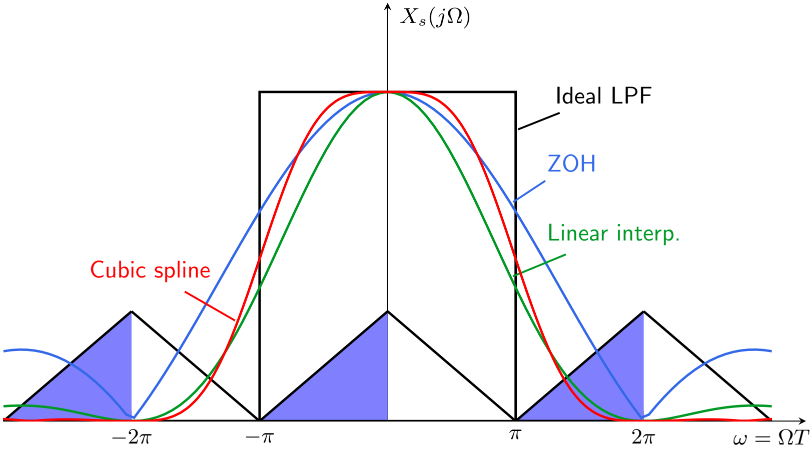
Practical reconstruction: cubic-spline interpolation

Example of reconstruction using cubic-spline interpolation



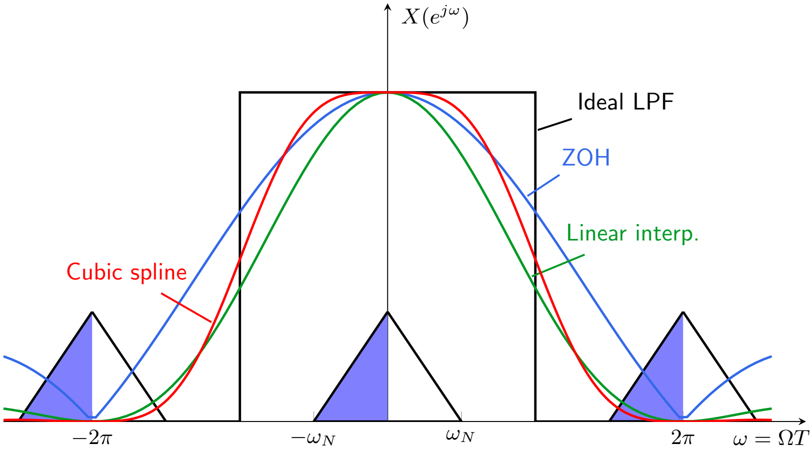
Comparison of reconstruction filters

The interpolation filter must suppress the spectrum replicas without distorting the spectrum centered at the origin



Comparison of reconstruction filters

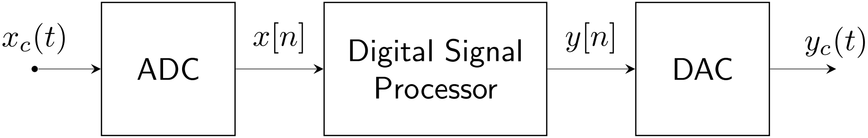
Oversampling makes the job of the interpolation filter much easier.



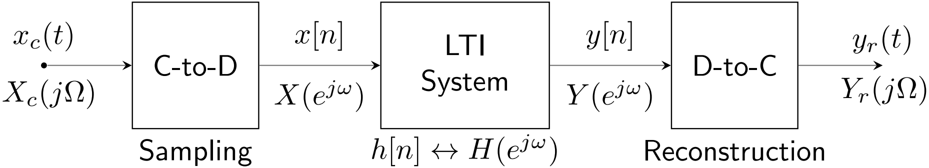
With oversampling, the interpolation filters look approximately flat around $[-\omega_N, \omega_N]$, and they suppress the spectrum replicas more strongly.

Discrete-time filtering of continuous-time signals

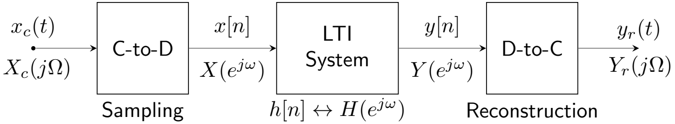
In practice



DSP theory



Discrete-time filtering of continuous-time signals



For the C-to-D converter (sampling):

$$X(e^{j\Omega T}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s))$$

For the discrete-time LTI system:

$$Y(e^{j\Omega T}) = H(e^{j\Omega T})X(e^{j\Omega T}) \qquad \text{(using } \omega = \Omega T\text{)}$$

For the D-to-C converter (reconstruction or interpolation):

$$Y_r(j\Omega) = H_r(j\Omega)Y(e^{j\Omega T})$$

Putting it all together:

$$Y_r(j\Omega) = H_r(j\Omega)H(e^{j\Omega T})\frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s))$$

Discrete-time filtering of continuous-time signals

$$Y_r(j\Omega) = H_r(j\Omega)H(e^{j\Omega T})\frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s))$$

We can simplify this equation by making two assumptions:

1. **No aliasing**. That is, assuming that the signal $X_c(j(\Omega - k\Omega_s))$ is bandlimited such that

   $$X_c(j\Omega) = 0 \text{ for } |\Omega| \geq \Omega_N$$

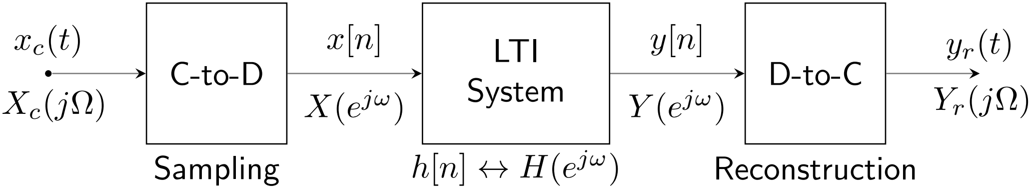
   and that the sampling frequency is such that $\Omega_s > 2\Omega_N$.
2. **Ideal reconstruction**. That is, $H_r(j\Omega)$ is the ideal lowpass filter:

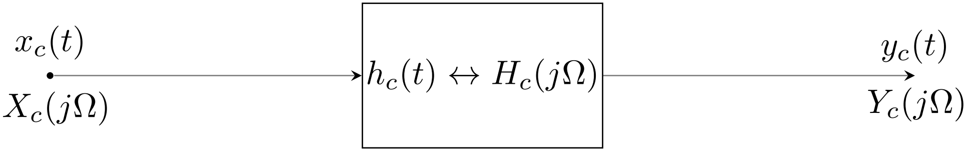
With these assumptions:

$$Y_r(j\Omega) = H(e^{j\Omega T})X_c(j\Omega), \text{ for } |\Omega| < \Omega_N$$

Discrete-time filtering of continuous-time signals

The simplified equation $Y_r(j\Omega) = H(e^{j\Omega T})X_c(j\Omega),$ for $|\Omega| < \Omega_N$ implies that the following two systems are equivalent

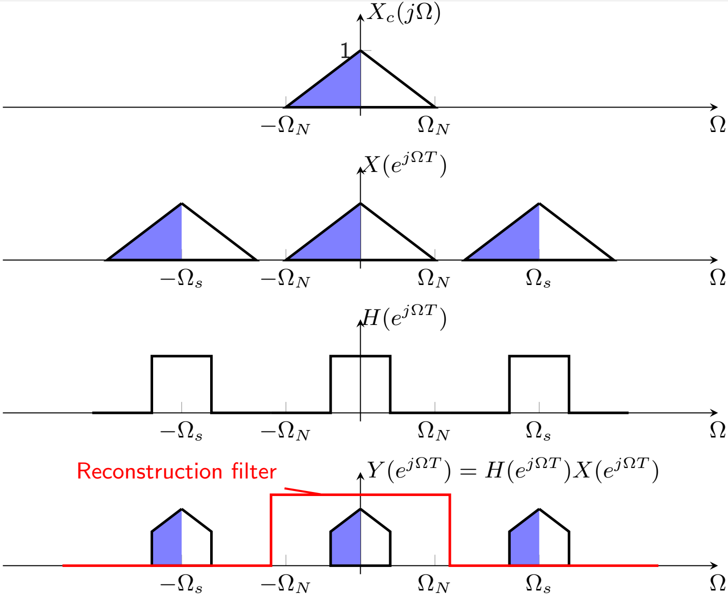




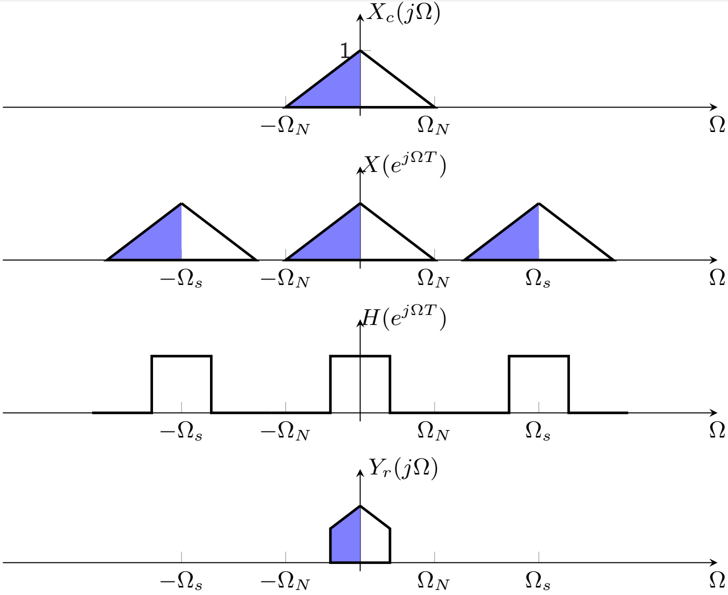
$$H_c(j\Omega) = H(e^{j\Omega T}), |\Omega| \leq \Omega_s/2$$

Conclusion: in theory, we can perform any LTI continuous-time filtering in discrete-time.

Graphically



Graphically



- We start with a bandlimited analog signal $x_c(t) \leftrightarrow X_c(j\Omega)$ (top plot)
- After sampling with frequency $\Omega_s = \frac{2\pi}{T}$, we obtain the discrete-time signal (second plot from the top). As usual we have the spectrum replicas in discrete-time. Note that for plotting we used the continuous-time frequency Ω. As a result, the spectrum replicas are centered around multiples of Ω_s. If we had used the discrete-time frequency $\omega = \Omega T$, the signal replicas would appear around multiples of 2π.
- The Third plot corresponds to some operation that we'll perform in discrete time. It could be any sort of LTI system, but for this illustration we'll use an ideal lowpass filter whose cutoff frequency is smaller than Ω_N. As a result, some part of the signal will be cut off.
- Filtering by $H(e^{j\Omega T})$ yields the output discrete-time signal $Y(e^{j\Omega T})$.
- After signal reconstruction with the ideal reconstruction filter (lowpass filter with cutoff frequency $\Omega_s/2$), we obtain the analog signal (no spectrum replicas) shown in the bottom plot.
- Note that the output analog signal is the same that we'd have obtained if we had filtered the original analog signal with an ideal (analog) lowpass filter of cutoff frequency smaller than Ω_N.
- Therefore, this example illustrates that we achieved continuous-time filtering by performing discrete-time filtering by $H(e^{j\Omega T})$. In fact, any continuous-time filtering can be performed in discrete-time, provided that there is no aliasing and that the reconstruction filter is the ideal lowpass filter. Although the latter condition is unfeasible, we can use practical interpolation filters to achieve very similar results.

Summary

- Sampling a continuous-time signal results in replicas of the spectrum at multiples of the sampling frequency Ω_s (or 2π of the normalized frequency ω)
- A band-limited signal has highest frequency Ω_N $(X_c(j\Omega) = 0, |\Omega| > \Omega_N)$
- If a band-limited signal is oversampled $(\Omega_s > 2\Omega_N)$ there'll be gaps between the spectrum replicas
- If the signal is undersampled $(\Omega_s < 2\Omega_N)$ the spectrum replicas will overlap resulting in aliasing distortion
- We can perfectly reconstruct a signal from its samples, provided that there is no aliasing and that we use the ideal lowpass filter as reconstruction filter
- In practice, we use different reconstruction filters, since the ideal lowpass filter is unfeasible.
- Oversampling relaxes the reconstruction filter specifications
- In theory, we can perform any LTI continuous-time filtering in discrete-time (in DSP), provided that there is no aliasing and that we use the ideal reconstruction filter