

# Typical stages in digital signal processing

Dr. Ing. Rodrigo Gonzalez

`rodralez@frm.utn.edu.ar`

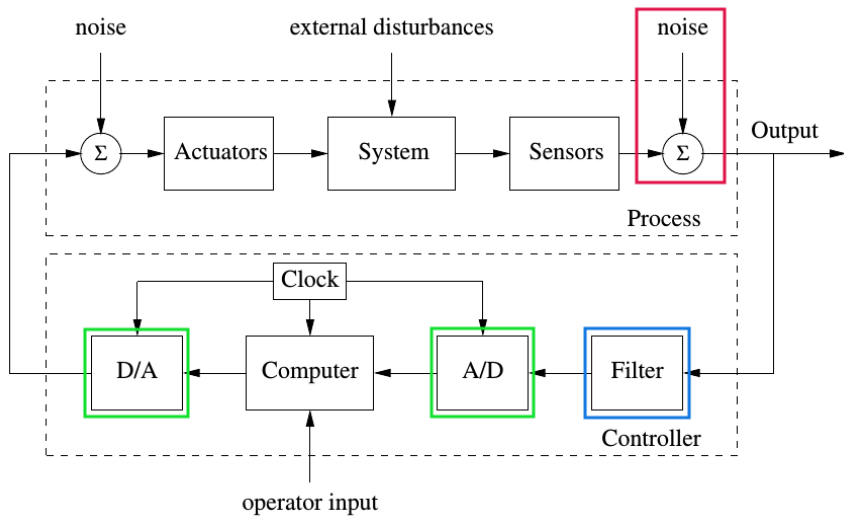
Técnicas Digitales III

Universidad Tecnológica Nacional,  
Facultad Regional Mendoza.



- 1 DSP in a the context of control systems
- 2 Sampling signals in the frequency domain
- 3 Digital processing of analog signals
- 4 Aliasing prefiltering
- 5 A/D Conversion
- 6 D/A Conversion

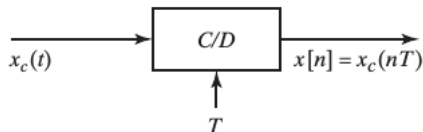
# DSP in the context of control systems



The discrete-time representation of a continuous-time signal is obtained through periodic sampling from a continuous-time signal  $x_c(t)$  according to,

$$x[n] = x_c(nT), \quad -\infty < n < \infty, \quad (1)$$

where  $T$  is the sampling period, and  $f_s = 1/T$  is the sampling frequency, or  $\Omega_s = 2\pi/T$  in radians/s

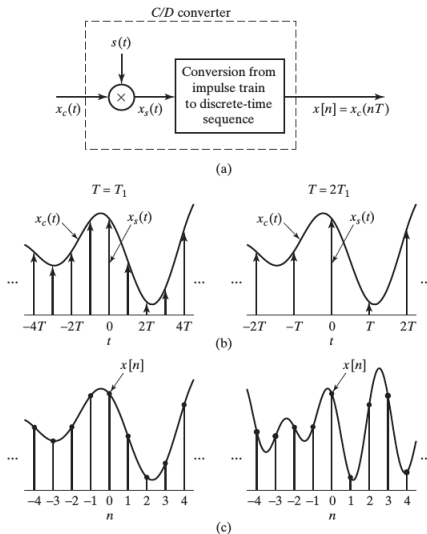


**Figure 4.1** Block diagram representation of an ideal continuous-to-discrete-time (C/D) converter.

# Sampling process

It is convenient to represent the sampling process mathematically in the two stages.

- 1 An impulse train  $s(t)$  is multiplied by a continuous-time signal  $x_c(t)$ .
- 2 The continuous-time signal  $x_s(t)$  is transformed to a discrete-time sequence  $x[n]$ .



**Figure 4.2** Sampling with a periodic impulse train, followed by conversion to a discrete-time sequence. (a) Overall system. (b)  $x_s(t)$  for two sampling rates. (c) The output sequence for the two different sampling rates.

# Frequency-domain representation of sampling

$x_s(t)$  is obtained multiplying  $x_c(t)$  by a periodic impulse train  $s(t)$ ,

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT), \quad (2)$$

$$x_s(t) = x_c(t) s(t), \quad (3)$$

$$= x_c(t) \sum_{n=-\infty}^{\infty} \delta(t - nT), \quad (4)$$

$$= \sum_{n=-\infty}^{\infty} x_c(nT) \delta(t - nT) \quad \text{by sifting property.} \quad (5)$$

The Fourier transform of the periodic impulse train  $s(t)$  is the periodic impulse train,

$$S(j\Omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s), \quad \text{where } \Omega_s = \frac{2\pi}{T}. \quad (6)$$

The Fourier transform of  $x_s(t)$  is the continuous-variable convolution of  $X_c(j\Omega)$  and  $S(j\Omega)$ ,

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega), \quad (7)$$

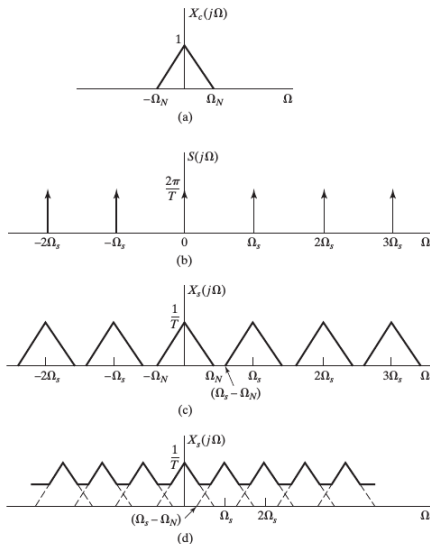
$$X_s(j\Omega) = \frac{1}{T} X_c[j(\Omega - k\Omega_s)]. \quad (8)$$

# Frequency-domain representation of sampling, II

- Fourier transform of  $x_s(t)$  consists of periodically repeated copies of  $X_c(j\Omega)$
- These copies are shifted by integer multiples of the sampling frequency.
- It is evident that

$$\Omega_s - \Omega_N \geq \Omega_N, \text{ or,}$$

$$\Omega_s \geq 2\Omega_N$$



**Figure 4.3** Frequency-domain representation of sampling in the time domain. (a) Spectrum of the original signal. (b) Fourier transform of the sampling function. (c) Fourier transform of the sampled signal with  $\Omega_s \geq 2\Omega_N$ . (d) Fourier transform of the sampled signal with  $\Omega_s < 2\Omega_N$ .

Let  $x_c(t)$  be a bandlimited signal with,

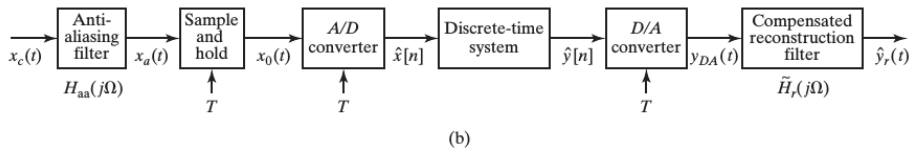
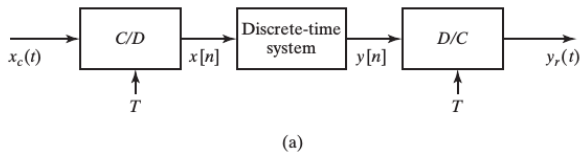
$$X_c(j\Omega) = 0 \text{ para } |\Omega| \geq \Omega_N. \quad (9)$$

Then  $x_c(t)$  is uniquely determined by its samples  $x[n] = x_c(nT)$ ,  $n = 0, \pm 1, \pm 2, \dots$  si

$$\Omega_s = \frac{2\pi}{T} \geq 2\Omega_N. \quad (10)$$

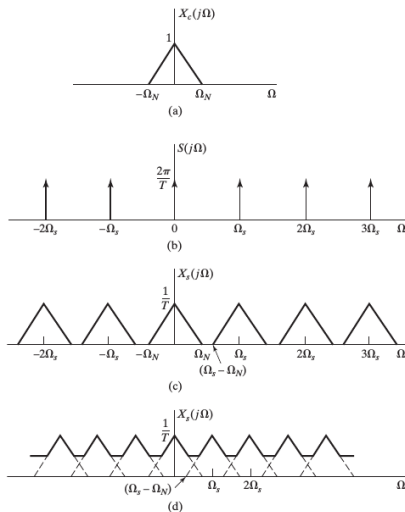
The frequency  $\Omega_N$  is commonly referred to as the **Nyquist frequency**, and the frequency  $2\Omega_N$  as the **Nyquist rate**.





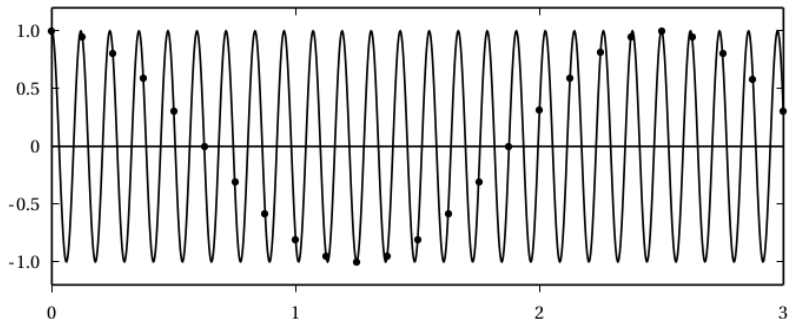
**Figure 4.47** (a) Discrete-time filtering of continuous-time signals. (b) Digital processing of analog signals.

# Aliasing prefiltering, motivation



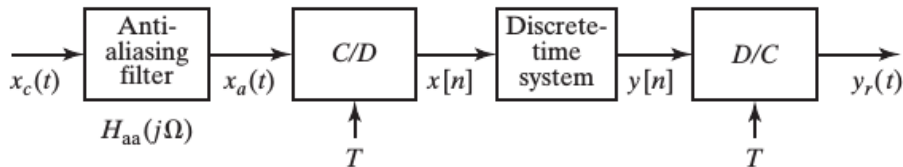
**Figure 4.3** Frequency-domain representation of sampling in the time domain. (a) Spectrum of the original signal. (b) Fourier transform of the sampling function. (c) Fourier transform of the sampled signal with  $\Omega_s > 2\Omega_N$ . (d) Fourier transform of the sampled signal with  $\Omega_s < 2\Omega_N$ .

## Aliasing prefiltering, example

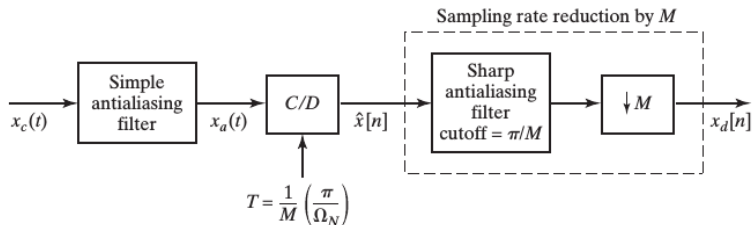


**Figure 9.8** Example of aliasing: a sinusoid at 8400 Hz,  $x(t) = \cos(2\pi \cdot 8400t)$  (solid line) is sampled at  $F_s = 8000$  Hz. The sampled values (dots) are indistinguishable from those of at 400 Hz sinusoid sampled at  $F_s$ .

- Even if the signal is naturally bandlimited (as music), wideband additive noise may fill in the higher frequency range, and as a result of sampling, these noise components would be aliased into the low-frequency band.
- (Play video).

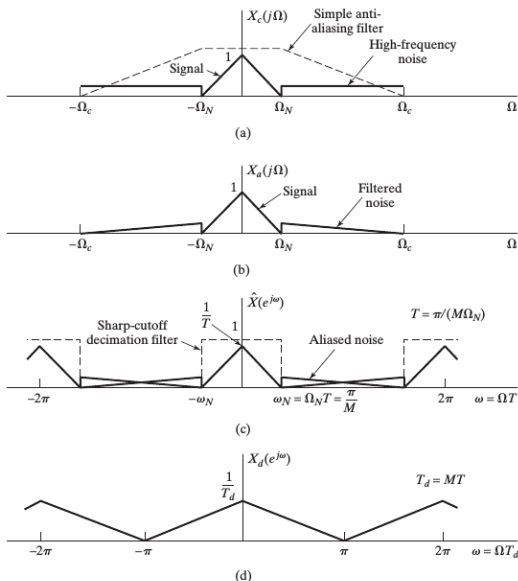


**Figure 4.48** Use of prefiltering to avoid aliasing.



**Figure 4.49** Using oversampled A/D conversion to simplify a continuous-time antialiasing filter.

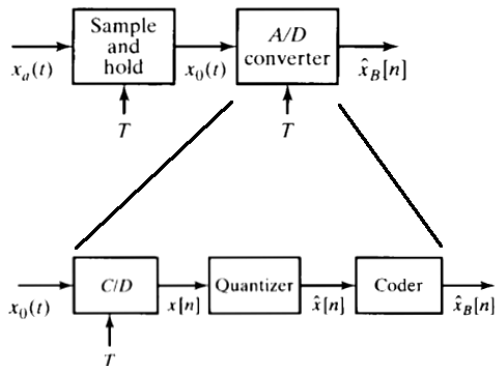
# Oversampling frequency response



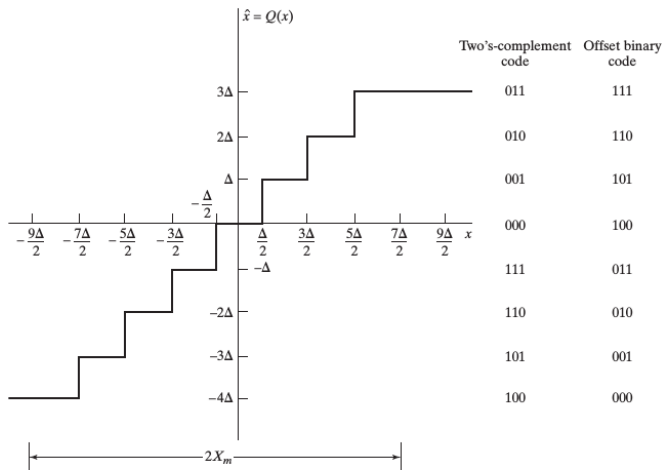
**Figure 4.50** Use of oversampling followed by decimation in C/D conversion.

# A/D Conversion Stages

- The A/D converter is a physical device that converts a voltage or current amplitude at its input into a binary code representing a quantized amplitude value closest to the amplitude of the input.
- The sample-and-hold stage can be a zero-order-hold.



- Uniformly spaced quantizer.
- The number of quantization levels will be a power of two ( $2^N$ ).



**Figure 4.54** Typical quantizer for A/D conversion.



# Quantizer Error example

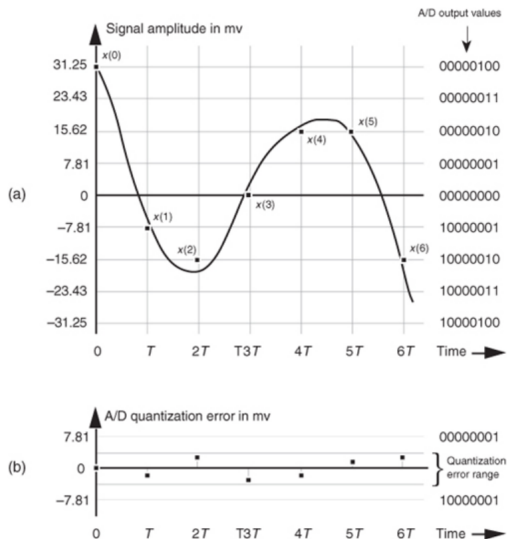
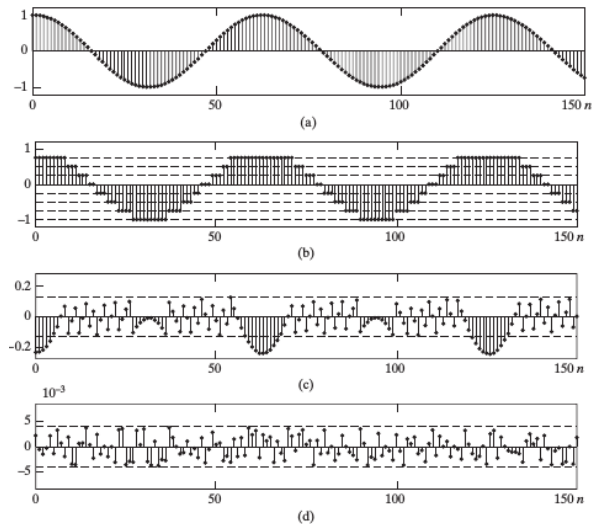
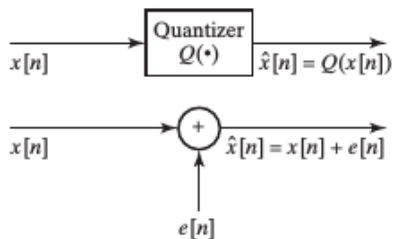


Figure: 12.1 [2]

## Quantizer Error example, II



**Figure 4.57** (continued) (b) Quantized samples of the cosine waveform in part (a) with a 3-bit quantizer. (c) Quantization error sequence for 3-bit quantization of the signal in (a). (d) Quantization error sequence for 8-bit quantization of the signal in (a).



**Figure 4.56** Additive noise model for quantizer.

The precision of the quantizer is given by [2]:

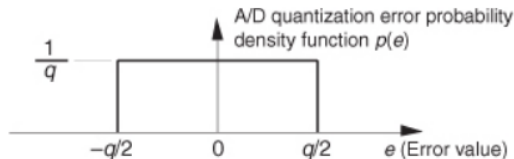
$$q = \frac{\text{full voltage range}}{2^{\text{word length}}} = \frac{2V_p}{2^B} \cdot [mV] \quad (11)$$

- $SNR = (P_{\text{signal}})/(P_{\text{noise}})$  relates two powers.
- Since  $q$  is defined as a random variable, its power cannot be represented explicitly.
- A statistical version of SNR is used,

$$SNR_{ADC} = 10 \cdot \log_{10} \left( \frac{\text{input signal variance}}{\text{A/D quantization noise variance}} \right), \quad [\text{dB}] \quad (12)$$

$$= 10 \cdot \log_{10} \left( \frac{\sigma_{\text{signal}}^2}{\sigma_{ADC}^2} \right). \quad (13)$$

## ADC Signal-to-Noise relationship, II



$$\sigma_{ADC}^2 = \int_{-q/2}^{q/2} (e - \mu)^2 p(e) de = \int_{-q/2}^{q/2} e^2 p(e) de = \frac{1}{q} \int_{-q/2}^{q/2} e^2 de = \frac{q^2}{12}, \quad (14)$$

$$\sigma_{ADC}^2 = \left( \frac{2V_p}{2^B} \right)^2 \cdot \frac{1}{12} = \boxed{\frac{V_p^2}{3 \cdot 2^{2B}}}, \quad (15)$$

$$LF = \frac{rms_{signal}}{V_p} = \frac{\sigma_{signal}}{V_p} \Rightarrow \sigma_{signal}^2 = \boxed{LF^2 \cdot V_p^2}, \quad (\text{Load Factor}) \quad (16)$$

$$SNR_{ADC} = 10 \cdot \log_{10} \left[ \left( LF^2 \cdot V_p^2 \right) \cdot \frac{3 \cdot 2^{2B}}{V_p^2} \right] = 10 \cdot \log_{10} \left[ \left( LF^2 \cdot 3 \cdot 2^{2B} \right) \right], \quad (17)$$

$$= 10 \cdot \left[ \log_{10}(LF^2) + \log_{10}(3) + 2 \log_{10}(2) \cdot B \right], \quad (18)$$

$$= 20 \cdot \log_{10}(LF) + 4.77 + 6.02 \cdot B. \quad [\text{dB}] \quad (19)$$

$$\begin{aligned} SNR_{ADC} &= 20 \cdot \log_{10}(LF) + 4.77 + 6.02 \cdot B, \quad [\text{dB}] \\ &= 20 \cdot \log_{10} \left( \frac{rms_{signal}}{V_p} \right) + 4.77 + 6.02 \cdot B. \quad [\text{dB}] \end{aligned}$$

### Considerations about LF:

- Ideally, if  $rms_{signal} \gg V_p$ ,  $SNR_{ADC}$  increases, but this will produce a severe distortion in the sampling signal (saturation).
- On the other hand, if  $rms_{signal} \ll V_p$ ,  $SNR_{ADC}$  decreases.
- Eq. 19 was obtained for an ideal ADC. Other sources of error should be taken into account.
- Moreover, it was considered that ADC's  $V_{MAX} = V_p$ .
- Therefore, calculated  $SNR_{ADC}$  should be decreased by 3 or 6 dB.

### Considerations about numbers of bits:

- $SNR_{ADC}$  increases 6 dB by each bit in ADC's quantizer.
- So, the more bits the better, isn't it?

For a sinusoidal signal,  $rms_{signal} = V_p/\sqrt{2}$ .

$$SNR_{ADC} = 20 \cdot \log_{10} \left( \frac{rms_{signal}}{V_p} \right) + 4.77 + 6.02 \cdot B, \quad (20)$$

$$= 20 \cdot \log_{10} \left( \frac{V_p/\sqrt{2}}{V_p} \right) + 4.77 + 6.02 \cdot B. \quad (21)$$

Thus, the maximum  $SNR_{ADC}$  is,

$$SNR_{ADC} = 20 \cdot \log_{10} \left( 1/\sqrt{2} \right) + 4.77 + 6.02 \cdot B, \quad (22)$$

$$= -3.01 + 4.77 + 6.02 \cdot B, \quad (23)$$

$$= 1.76 + 6.02 \cdot B. \quad [\text{dB}] \quad (24)$$

# ADC resolution for a particular signal

Consider the following example:

- The SNR for an audio output amplifier is 110 dB.
- A 24-bits ADC is chosen to sample the output amplifier (professional audio).

$$SNR_{ADC} = 1.76 + 6.02 \cdot 24 - 3 = 143.24 \text{ dB}.$$

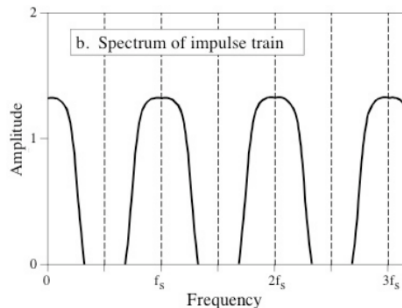
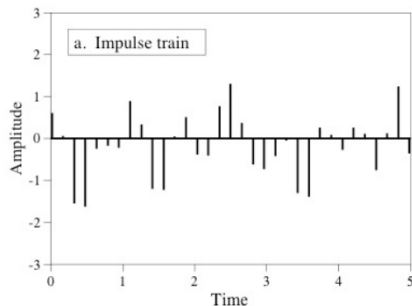
- How many bits are used to measure noise?  $(143 - 110)/6 \simeq 5.5$  bits!.
- In a control loop, picking a bad ADC resolution could lead to a catastrophic scenario.

*Summary: the number of bits in an ADC must match the SNR of the signal to be sampled.*

- Rule of thumb: the ADC resolution should be choose in order to provide 6 dB (1 bit) above the SNR of the signal to be sample.
- Additional bits (noisy bits) can be eliminated by right shifting (In C: `adc_read >> 5`).

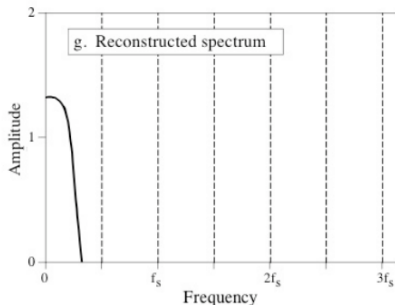
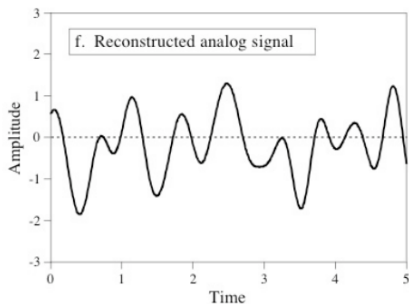


- The DAC reverses the ADC process.
- It decodes the signal making a conversion from a bit sequence to an impulse train.



## Ideal D/C Converter, II

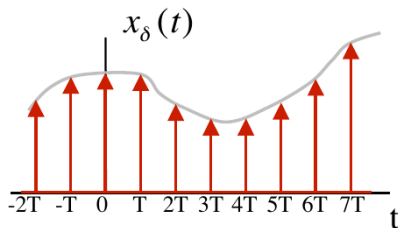
- The signal is reconstructed with an analog low-pass filter to remove the frequencies about 1/2 the sampling rate.
- After filtering the impulse train with a such a low-pass filter.



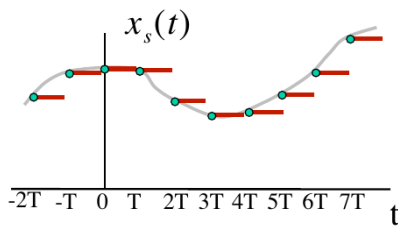
- However, the ideal operation just described assumes availability of infinitely many samples. Not realistic!

- Practical operation uses only a finite number of samples.
- Many techniques can be used to approximately reconstruct the signal.
- One such technique is a **zero-order holder** (ZOH).

Modulated impulses

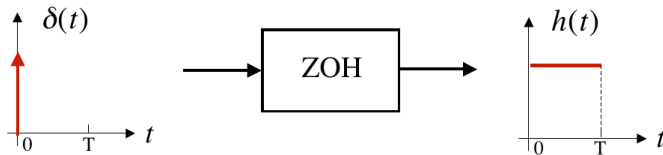


After Zero-Order Holder



# Sampling with Zero-order Holder

The impulse response of a zero-order holder.

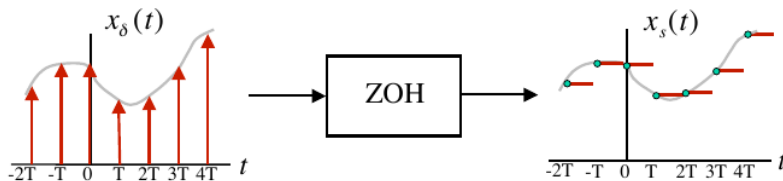


In the frequency domain:

$$H(f) = F \left\{ \text{rect} \left( \frac{t - T_s/2}{T_s} \right) \right\} = e^{-j\pi f T_s} \text{sinc}(f T_s) \quad (25)$$

**sinc**: cardinal sine function.

A sampled signal through a zero-order holder:

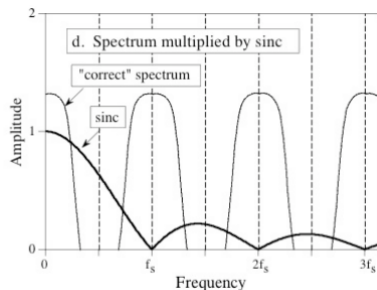
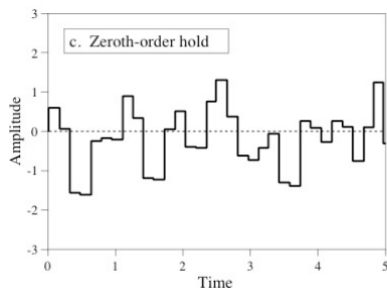


In the frequency domain:

$$H(f) = F \left\{ x_{\delta}(t) * \text{rect} \left( \frac{t - T_s/2}{T_s} \right) \right\} = \frac{e^{-j\pi f T_s} \text{sinc}(f T_s)}{T_s} \sum_{k=-\infty}^{\infty} X(f - n f_s) \quad (26)$$

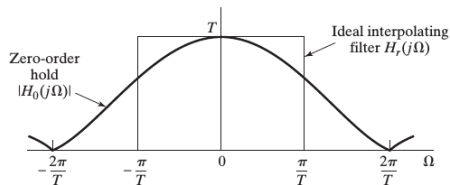
# Sampling with Zero-order Holder, III

In the frequency domain, the zero-order holder translates into a multiplication of the real signal spectrum by a sinc function!

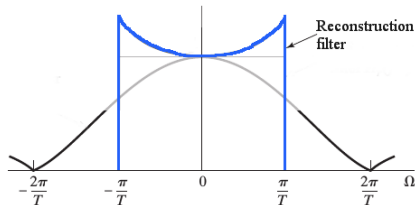


The zero-order holder produce a gain drop!

- The reconstruction filter compensates the gain drop.
- It is also known as invert sinc filter (inv sinc).



(a)



# Reconstruction Filtering Strategies

- 1) Ignore the effect of the zero-order holder and accept the consequences.
- 2) Design a digital filter to remove the sinc effect (pre-equalizing) [4].
- 3) Design an analog filter to remove the sinc effect (post-equalizing) [4].

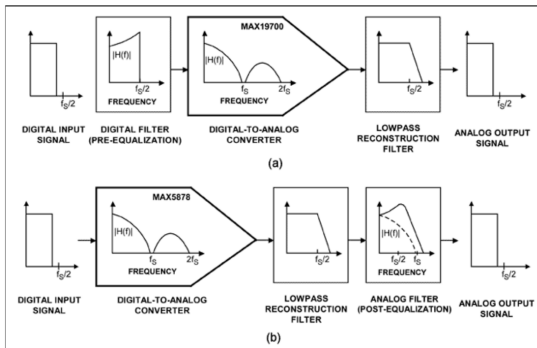


Figure 4. A pre-equalization digital filter is used to cancel the effect of sinc rolloff in a DAC (a). As an alternative, you can use a post-equalization analog filter for the same purpose (b).

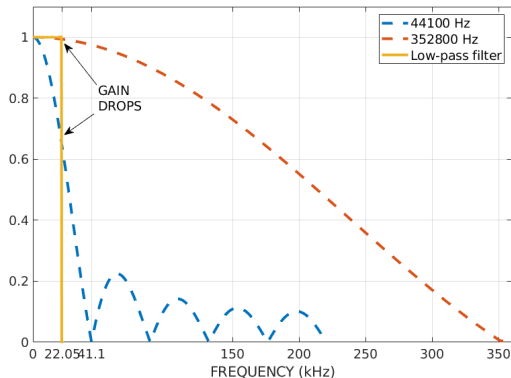


# Reconstruction Filtering Strategies, II

## 4) Oversampling [4].

CD players example:

- In CD players, data sampling rate is 44.1 kHz.
- At 80 % of Nyquist frequency, the output amplitude is attenuated by 2.42dB.
- Data is oversampled by a factor of 8 to 352.8 kHz.
- Now, distortion for ZOH is effectively eliminated.



- 1 Alan V. Oppenheim and Ronald W. Schaffer. *Discrete-time signal processing*, 3rd Ed. Prentice Hall. 2010. Sections 4.1, 4.2, 4.3 and 4.8.
- 2 Richard G. Lyons. *Understanding Digital Signal Processing*, 3rd Ed. Prentice Hill. 2010. Section 12.3.1.
- 3 Paolo Prandoni and Martin Vetterli. *Signal processing for communications*. Taylor and Francis Group, LLC. 2008. Section 9.6.
- 4 Maxim Integrated. *Equalizing Techniques Flatten DAC Frequency Response*. Application Note 3853. August 2012.