

# Etapas esenciales en procesamiento discreto de señales

Dr. Ing. Rodrigo Gonzalez

`rodrazalez@frm.utn.edu.ar`

Control y Sistemas

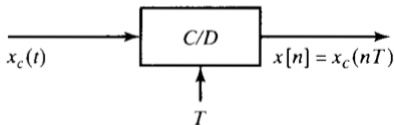
Universidad Nacional de Cuyo,  
Facultad de Ingeniería

- 1 Representación en el dominio de la frecuencia de señales sampleadas
- 2 Etapas esenciales de un sistema DSP
- 3 Filtro aliasing
- 4 Conversión A/D
- 5 Conversión D/A

La representación discreta de una señal continua  $x_c(nT)$  se puede obtener muestreando la misma en forma uniforme. Esto genera una secuencia de muestras  $x[n]$ .

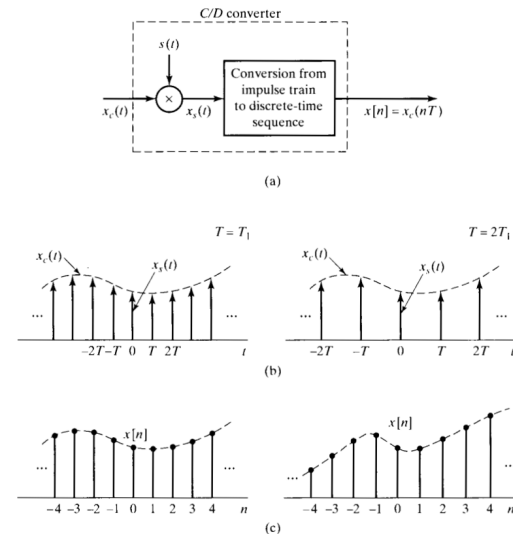
$$x[n] = x_c(nT), \quad -\infty < n < \infty, \quad (1)$$

donde  $T$  es el periodo de muestreo, y  $f_s = 1/T$  es la frecuencia de muestreo, o  $\Omega_s = 2\pi/T$  en radianes.



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

- Matemáticamente, es conveniente representar el muestreo de una señal continua en dos etapas.
- Se multiplica un tren de impulsos  $s(t)$  por la señal continua  $x_c(t)$ .
- La señal continua  $x_s(t)$  es transformada en una secuencia discreta  $x[n]$ .



**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.

$x_s(t)$  se obtiene modulando  $x_c(t)$  por un tren de impulsos  $s(t)$ ,

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

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

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

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

La transformada de Fourier de  $s(t)$  es un tren de impulsos periódico,

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

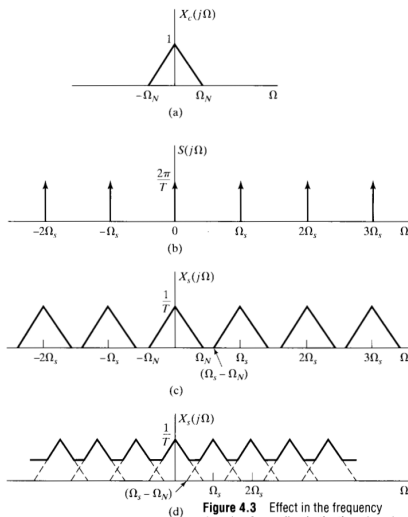
La transformada de Fourier de  $x_s(t)$  es la convolución en el dom. de la frecuencia de  $X_c(j\Omega)$  y  $S(j\Omega)$ ,

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

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

# Respuesta en frecuencia de señales muestreadas, cont'd

- $X_c(j\Omega)$  representa una señal BW limitado, con frec. máxima  $\Omega_N$ .
- Las réplicas de  $X_c(j\Omega)$  no se superponen cuando
  - $\Omega_S - \Omega_N > \Omega_N$ , o
  - $\Omega_S > 2\Omega_N$



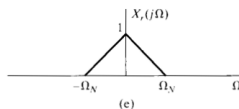
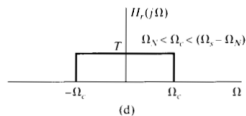
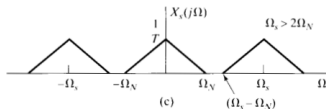
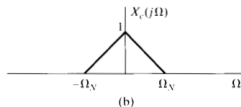
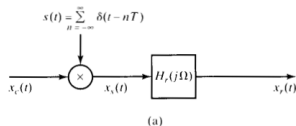
**Figure 4.3** Effect in the frequency domain of sampling in the time domain.  
(a) Spectrum of the original signal.  
(b) Spectrum of the sampling function.  
(c) Spectrum of the sampled signal with  $\Omega_S > 2\Omega_N$ . (d) Spectrum of the sampled signal with  $\Omega_S < 2\Omega_N$ .

# Transformación de una secuencia discreta en una señal continua

- Si  $X_r(j\Omega)$  es un filtro pasa bajos ideal con ganancia  $T$  y frecuencia de corte  $\Omega_c$  tal que,

$$\Omega_N < \Omega_c < \Omega_s - \Omega_N$$

$$\text{Entonces, } X_r(j\Omega) = X_c(j\Omega)$$



Sea  $x_c(t)$  una señal limitada en frecuencia con

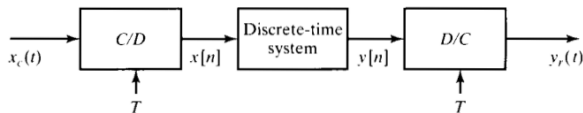
$$X_c(j\Omega) = 0 \text{ para } |\Omega| \geq \Omega_n. \quad (9)$$

Entonces  $x_c(t)$  está únicamente determinado por sus muestras  $x[n] = x_c(nT)$ ,  $n = 0, \pm 1, \pm 2, \dots$  si

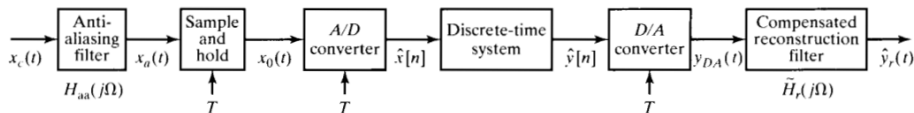
$$\Omega_s = \frac{2\pi}{T} \geq 2\Omega_n. \quad (10)$$

$\Omega_n$  se conoce como la frecuencia de Nyquist





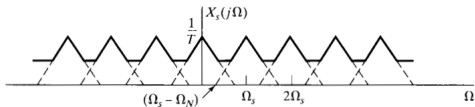
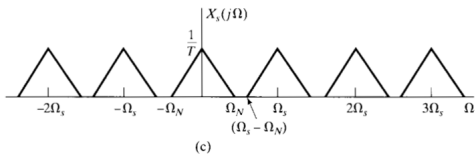
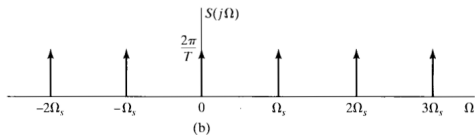
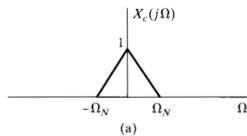
(a)



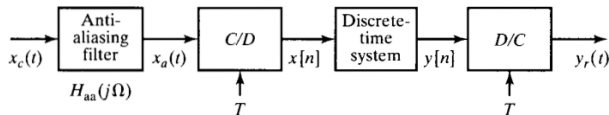
(b)

**Figure 4.41** (a) Discrete-time filtering of continuous-time signals. (b) Digital processing of analog signals.

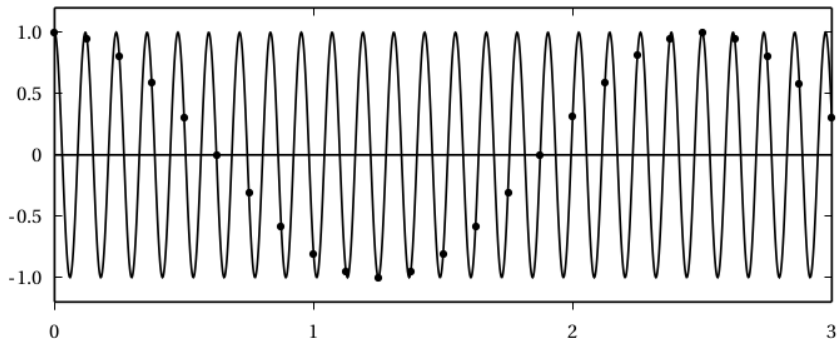
# Filtro antialiasing, motivación



- Aunque la señal esté naturalmente limitada en frecuencia (por ej., música), se debe limitar el AB para reducir el ruido blanco.
- (Reproducir video).

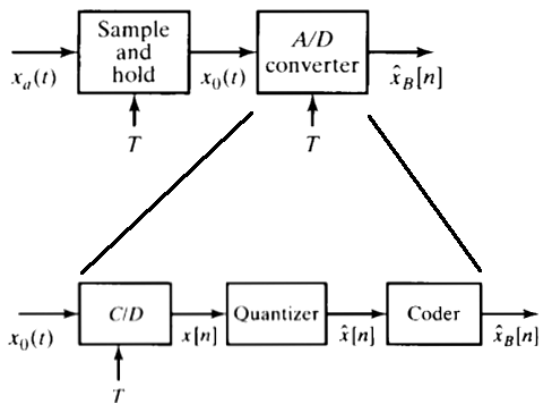


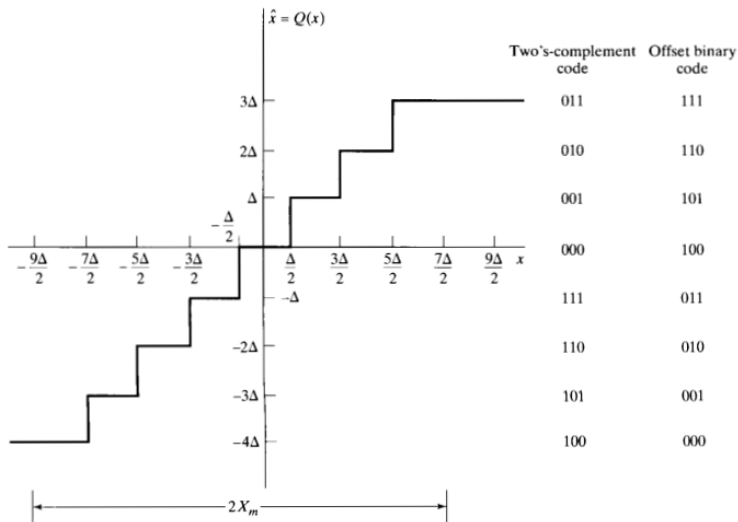
**Figure 4.42** Use of prefiltering to avoid aliasing.



**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 a 400 Hz sinusoid sampled at  $F_s$ .

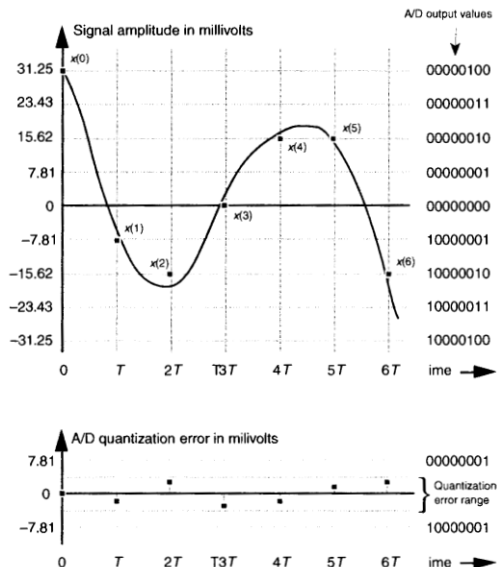
# Etapas de un conversor A/D

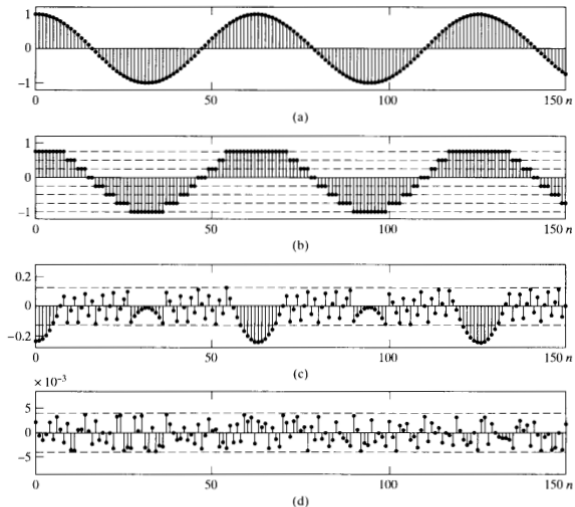




**Figure 4.48** Typical quantizer for A/D conversion.

# Cuantizador, cont'd





**Figure 4.51** Example of quantization noise. (a) Unquantized samples of the signal  $x[n] = 0.99 \cos(n/10)$ . (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).



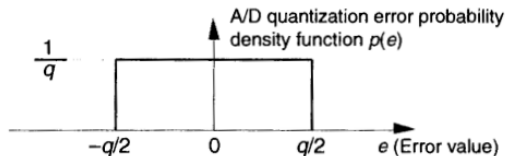
La precisión del cuantizador está dada por:

$$q = \frac{2V_p}{2^B} \cdot [mV] \quad (11)$$

$SNR = (P_{signal})/(P_{noise})$  relaciona potencias. Como  $q$  se define como una variable aleatoria, no se puede representar su potencia explícitamente. Se usa una versión estadística de SNR.

$$SNR_{ADC} = 10 \cdot \log_{10} \left( \frac{\text{varianza señal entrada}}{\text{varianza ruido cuantizador}} \right), \quad [dB] \quad (12)$$

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



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

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

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

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

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

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

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

### Consideraciones:

- Para aumentar  $SNR_{ADC}$  lo ideal es que  $rms_{signal} \gg V_p$ , pero esto producirá distorsión severa en la señal muestreada (saturación).
- Por otro lado, si  $rms_{signal} \ll V_p$  disminuye  $SNR_{ADC}$ .
- La ec. 19 es para un ADC ideal. No se consideran otras fuentes de error del ADC.
- Además, se considera que la tensión máxima de entrada coincide con  $V_p$  del ADC.
- Por tanto, en la práctica se asume que el  $SNR_{ADC}$  calculado es entre 3 y 6 dB menor.
- Se ve que  $SNR_{ADC}$  aumenta 6 dB por cada bit que posea el cuantizador del ADC.
- Entonces, mientras más bits tenga el ADC, mejor. ¿O no?.

Para una señal con distribución gaussiana, una solución de compromiso es  $rms_{signal} = V_p/2$  (demostración para el alumno).

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

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

Así, su máximo  $SNR_{ADC}$  está dado por,

$$SNR_{ADC} = 20 \cdot \log_{10} (0.5) + 4.77 + 6.02 \cdot B, \quad (22)$$

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

$$= -1.25 + 6.02 \cdot B. \text{ [dB]} \quad (24)$$

Considere el siguiente ejemplo:

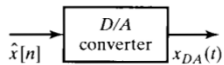
- La SNR de cierto amplificador de audio es de 110 dB.
- Se decide usar un ADC de 24 bits (audio profesional) para muestrear el amplificador.

$$SNR_{ADC} = -1.25 + 6.02 \cdot 24 - 3 = 140.23 \text{ dB}.$$

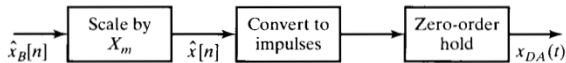
- ¡Se están usando  $(140 - 110)/6 \simeq 5$  bits para muestrear ruido!.
- En un lazo de control una mala estimación de la cantidad de bits del ADC puede llevar a resultados catastróficos.

*Conclusión: la cantidad bits de un ADC debe estar en función de la SRN de la señal que se desea muestrear.*

Regla: se debe elegir el ADC comercial con el número de bits inmediatamente superior al calculado. Luego, eliminar de las mediciones los bits adicionales con desplazamiento a derecha aritmético (`adc_read >> 5`).



(a)



(b)

**Figure 4.53** (a) Block diagram of D/A converter. (b) Representation in terms of a zero-order hold.

Un conversor D/A toma una secuencia de números y los transforma en una señal continua  $x_{DA}(t)$ ,

$$x_{DA}(t) = \sum_{n=-\infty}^{\infty} \hat{x}[n]h_0(t - nT), \quad (25)$$

$$h_0(t) = \begin{cases} 1, & 0 < t < T \\ 0, & \text{en otro caso} \end{cases} \quad (26)$$

$$x_{DA}(t) = x_0(t) + e_0(t), \quad (27)$$

$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n]h_0(t - nT), \quad (28)$$

$$e_0(t) = \sum_{n=-\infty}^{\infty} e[n]h_0(t - nT). \quad (29)$$

$x[n]$  se relaciona con la señal de entrada como  $x[n] = x_a(nT)$ .

La transformada de Fourier de  $x_0(t)$ ,

$$X_0(j\Omega) = \sum_{n=-\infty}^{\infty} x[n]H_0(j\Omega)e^{-j\Omega nT}, \quad (30)$$

$$= \left( \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega nT} \right) H_0(j\Omega), \quad (31)$$

$$= X(e^{j\Omega T})H_0(j\Omega), \quad (32)$$

$$= \left[ \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a \left( j \left( \Omega - \frac{2\pi k}{T} \right) \right) \right] H_0(j\Omega). \quad (33)$$

La respuesta en frecuencia de un filtro ZOH,

$$H_0(j\Omega) = \frac{2 \sin(\Omega T/2)}{\Omega} e^{-j\Omega T/2}. \quad (34)$$

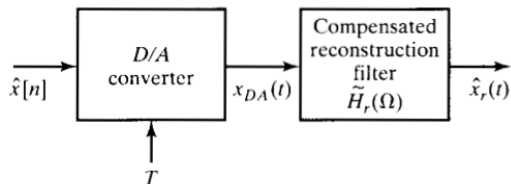


Por tanto, la respuesta en frecuencia de un filtro de reconstrucción compensado,

$$\tilde{H}_r(j\Omega) = \frac{H_r(j\Omega)}{H_0(j\Omega)}, \quad (35)$$

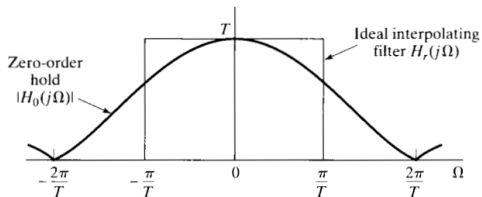
$$\tilde{H}_r(j\Omega)(t) = \begin{cases} \frac{\Omega T/2}{\sin(\Omega T/2)} e^{j\Omega T/2} & , |\Omega| < \pi/T \\ 0 & , |\Omega| > \pi/T \end{cases} \quad (36)$$

Así, la salida del DAC será  $x_a(t)$  si la entrada es  $x_o(t)$ .

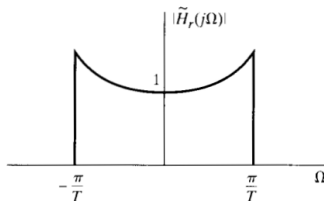


**Figure 4.55** Physical configuration for digital-to-analog conversion.

$|H_0(j\Omega)|$  atenúa -4 dB a  $\Omega = \pi/T$ .



(a)



(b)

**Figure 4.54** (a) Frequency response of zero-order hold compared with ideal interpolating filter. (b) Ideal compensated reconstruction filter for use with a zero-order-hold output.

- Alan V. Oppenheim and Ronald W. Schaffer. *Discrete-time signal processing, 2nd Ed.* Prentice Hall. 1999. Secciones 4.1, 4.2, 4.3 y 4.8.
- Richard G. Lyons. *Understanding Digital Signal Processing, 2nd Ed.* Prentice Hall. 2004. Sección 12.3.1.
- Paolo Prandoni and Martin Vetterli. *Signal processing for communications.* Taylor and Francis Group, LLC. 2008. Sección 9.6.