II. Sampling and Processing of Signals

Most discrete-time signals come from sampling a continuous-time signal, such as speech and audio signals, radar and sonar data, and seismic and biological signals. The process of converting these signals into digital form is called analog-to-digital (A/D) conversion. The reverse process of reconstructing an analog signal from its samples is known as digital-to-analog (D/A) conversion. This chapter examines the issues related to A/D and D/A conversion. Fundamental to this discussion is the sampling theorem, which gives precise conditions under which an analog signal may be uniquely represented in terms of its samples.

A. Analog-to-Digital Conversion

An A/D converter transforms an analog signal into a digital sequence. The input to the A/D converter, $x_a(t)$, is a real-valued function of a continuous variable, t. Thus, for each value of t, the function $x_a(t)$ may be any real number. The output of the A/D is a bit stream that corresponds to a discrete-time sequence, $x[n]$, with an amplitude that is quantised, for each value of n, to one of a finite number of possible values. The components of an A/D converter are shown in the Figure below.

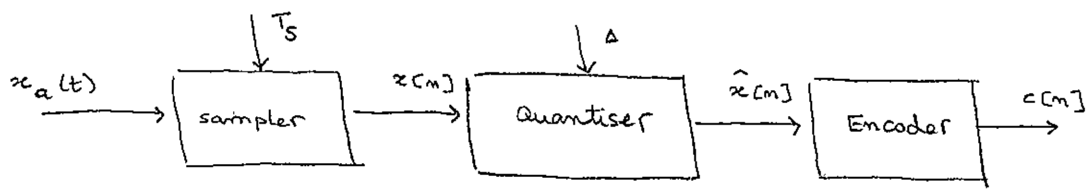


Fig. 1 The components of an analog-to-digital converter.

The first component in Fig. 1 is the sampler which is sometimes referred to as an ideal A/D converter. The sampler converts the continuous-time signal $x_a(t)$ into a discrete-time sequence $x[n]$ by extracting the values of $x_a(t)$ at integer multiples of the sampling period, T_s,

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

Because the samples $x_a(nT_s)$ have a continuous range of possible amplitudes, the second component of the A/D converter is the quantiser, which maps the continuous amplitude into a discrete set of amplitudes. For a uniform quantiser, the quantisation process is defined by the number of bits and the quantisation interval Δ. The last component is the encoder, which takes the digital signal $\hat{x}[n]$ and produces a sequence of binary codewords.

Typically, discrete-time signals are formed by periodically sampling a continuous-time signal as described by

$$x[n] = x_a(nT_s],$$

where the sample spacing T_s is the sampling period and $f_s = 1/T_s$ is the sampling frequency in samples per second.

The sampling process consists of two processing steps. First, the continuous-time signal $x_a(t)$ is multiplied by a periodic sequence of impulses:

$$s_a(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s),$$

which forms the sampled signal

$$x_s(t) = x_a(t)\, s_a(t) = \sum_{n=-\infty}^{\infty} x_a(nT_s)\, \delta(t - nT_s)$$

Then, the sampled signal is converted into a discrete-time signal by mapping the impulses that are spaced in time by T_s into a sequence $x[n]$ where the sample values are indexed by the integer value n:

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

This process is illustrated in the figure below.

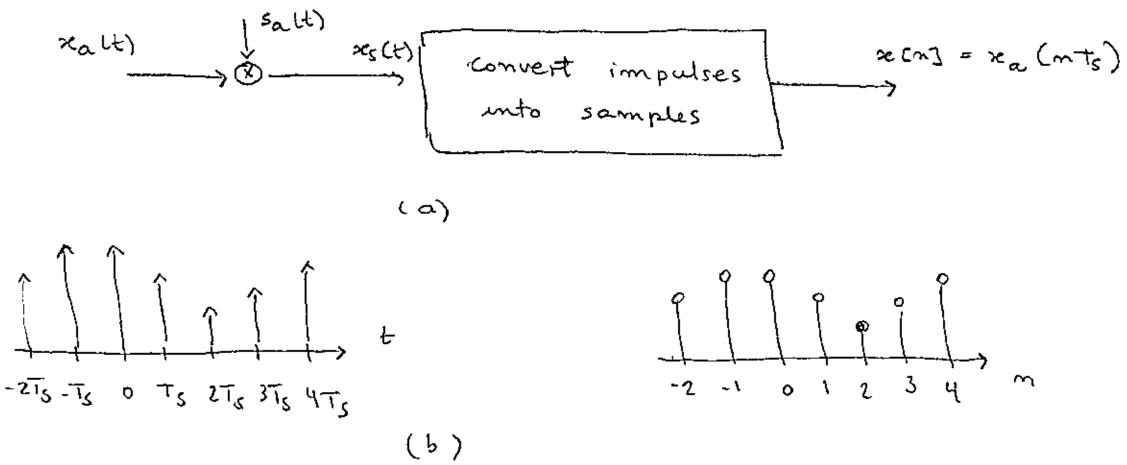


Fig. 2 (a) A model that consists of multiplying $x_a(t)$ by a sequence of impulses, followed by a system that converts impulses into samples. (b) An example that illustrates the conversion process.

The effect of the sampler can be analyzed in the frequency domain as follows. Because the Fourier transform of $\delta(t - nT_s)$ is $e^{-jn\Omega T_s}$, the Fourier transform of the sampled signal $x_s(t)$ is

$$X_s(j\Omega) = \sum_{n=-\infty}^{\infty} x_a(nT_s)\, e^{-jn\Omega T_s}$$

Another expression for $X_s(j\Omega)$ follows by noting that the Fourier transform of $s_a(t)$ is

$$S_a(j\Omega) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s),$$

where $\Omega_s = 2\pi/T_s$ is the sampling frequency in radians per second.

Therefore, we have

$$X_s(j\Omega) = \frac{1}{2\pi} X_a(j\Omega) * S_a(j\Omega) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_a(j\Omega - jk\Omega_s)$$

The discrete-time Fourier transform of $x[n]$ is

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\cdot e^{-jn\omega} = \sum_{n=-\infty}^{\infty} x_a(nT_s)\, e^{-jn\omega}$$

Comparing $X(e^{j\omega})$ with $X_s(j\Omega)$, it follows that

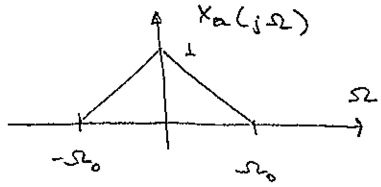
$$X(e^{j\omega}) = X_s(j\Omega)\Big|_{\Omega=\omega/T_s} = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X_a\left(j\frac{\omega}{T_s} - j\frac{2\pi k}{T_s}\right)$$

Thus, $X(e^{j\omega})$ is a frequency-scaled version of $X_s(j\Omega)$, with the scaling defined by

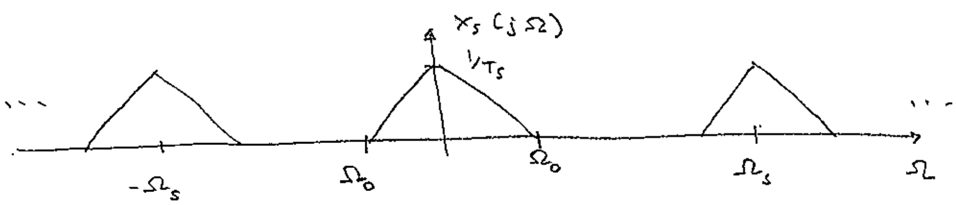
$$\omega = \Omega T_s$$

This scaling, which makes $X(e^{j\omega})$ periodic with a period of 2π, is a consequence of the time-scaling that occurs when $x_s(t)$ is converted to $x[n]$.

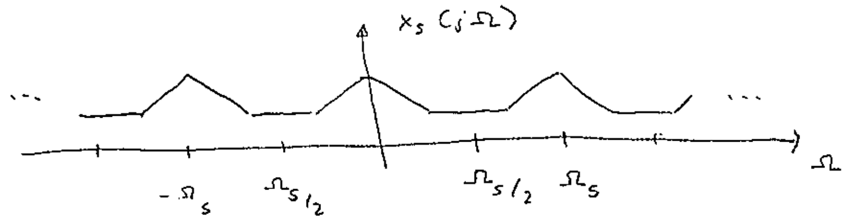
Ex: Suppose that $x_a(t)$ is strictly bandlimited so that $X_a(j\Omega) = 0$ for $|\Omega| > \Omega_0$ as shown below



If $x_a(t)$ is sampled with a sampling frequency $\Omega_s \geq 2\Omega_0$, the Fourier transform of $x_s(t)$ is formed by periodically replicating $X_a(j\Omega)$ as illustrated in the figure below



However, if $\Omega_s < 2\Omega_0$, the shifted spectra $X_a(j\Omega - jk\Omega_s)$ overlap and when these spectra are summed to form $X_s(j\Omega)$, the result is shown below



This overlapping of spectral components is called aliasing. When it occurs, the frequency content of $x_a(t)$ is corrupted and $X_a(j\Omega)$ cannot be recovered from $X_s(j\Omega)$.

In general, if $x_a(t)$ is strictly bandlimited so that the highest frequency in $x_a(t)$ is Ω_0, and if the sampling frequency is greater than $2\Omega_0$:

$$\Omega_s \geq 2\Omega_0$$

no aliasing occurs, and $x_a(t)$ may be uniquely recovered from its samples $x_a(nT_s)$ with a low-pass filter

8. Sampling Theorem: If $x_a(t)$ is strictly bandlimited

$$X_a(j\Omega) = 0 \quad , \quad |\Omega| > \Omega_0$$

then $x_a(t)$ may be uniquely recovered from its samples $x_a(nT_s)$ if

$$\boxed{\Omega_s = \frac{2\pi}{T_s} \geq 2\Omega_0}$$

The frequency Ω_0 is called the Nyquist frequency and the minimum sampling frequency, $\Omega_s = 2\Omega_0$, is called the Nyquist rate. Because the signals found in practice are seldom strictly bandlimited, an analog anti-aliasing filter is often used to filter the signal prior to sampling to reduce the aliasing that occurs in the A/D converter

c. Quantisation

A quantiser is a nonlinear and noninvertible system that transforms an input sequence $x[n]$ that has a continuous range of amplitudes into a sequence for which each value of $x[n]$ assumes one of a finite number of possible values. The quantisation operation is denoted by

$$\hat{x}[n] = Q[\, x[n] \,]$$

The quantiser has $L+1$ decision levels $x_1, x_2, \ldots, x_{L+1}$ that divide the amplitude range for $x[n]$ into L intervals:

$$I_k = [\, x_k,\ x_{k+1} \,] \quad , \quad k = 1, 2, \ldots, L$$

For an input $x[n]$ that falls within interval I_k, the quantiser assigns a value within this interval, $\hat{x}_k$, to $x[n]$. This process is illustrated in the Fig. below

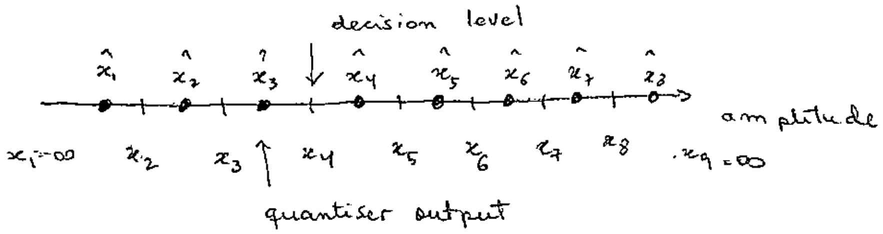


Fig. 3 A quantiser with nine decision levels that divide the input amplitudes into eight quantisation intervals and eight possible quantiser outputs, $\hat{x}_k$.

Quantisers may have quantisation levels that are either uniformly or nonuniformly spaced. When the quantisation intervals are uniformly spaced, we have

$$\Delta = x_{k+1} - x_k \, ,$$

where Δ is called the quantisation step size or the resolution of the quantiser, and the quantiser is said to be a uniform or linear quantiser.

The number of levels in a quantiser is generally given by

$$L = 2^{B+1}$$

in order to make the most efficient use of a $(B+1)$-bit binary word. A 3-bit uniform quantiser in which the quantiser output is rounded to the nearest quantisation level is illustrated in the Fig. below

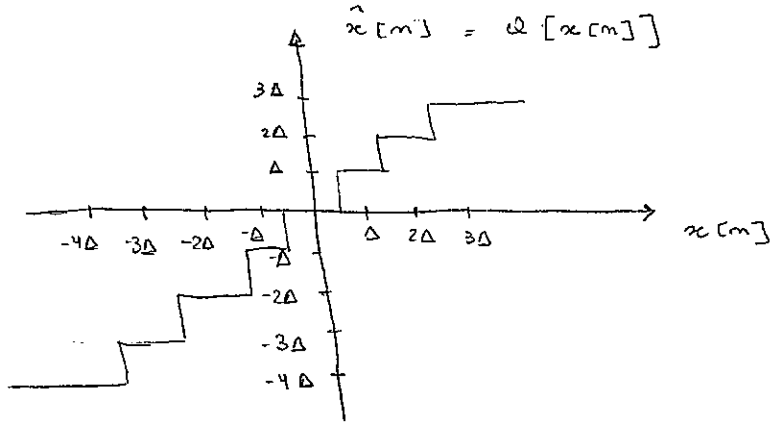


Fig. 4. A 3-bit uniform quantiser

With $L = 2^{B+1}$ quantisation levels and a step size Δ, the range of the quantiser is

$$R = 2^{B+1} \cdot \Delta$$

Therefore, if the quantiser input is bounded then we have

$$|x[n]| \leq X_{max},$$

where the range of possible input values may be covered with a step size described by

$$\Delta = \frac{X_{max}}{2^B}$$

With rounding, the quantisation error

$$e[n] = Q[x[n]] - x[n]$$

will be bounded by

$$-\frac{\Delta}{2} < e[n] < \frac{\Delta}{2}$$

However, if $|x[n]|$ exceeds X_{max}, then $x[n]$ will be clipped, and the quantisation error could be very large.

A useful model for the quantisation process is given in the Fig. below.

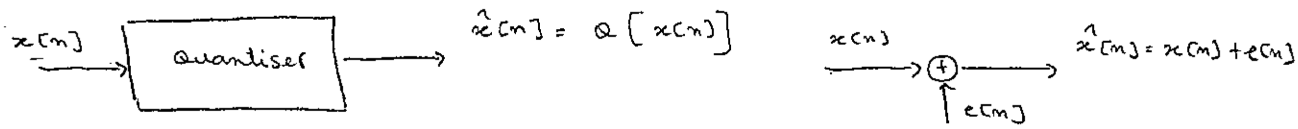


Fig. 5 A quantisation noise model.

Here, the quantisation error is assumed to be an additive noise source. Because the quantisation error is typically not known, the quantisation error is described statistically. It is generally assumed that $e[n]$ is a sequence of random variables where

1. The statistics of $e[n]$ do not change with time (the quantisation noise is a stationary random process)
2. The quantisation noise $e[n]$ is a sequence of uncorrelated random variables
3. The quantisation noise $e[n]$ is uncorrelated with the quantiser input $x[n]$.
4. The probability density function of $e[n]$ is uniformly distributed over the range of values of the quantisation error.

These assumptions often hold for time-varying signals with fine quantisation (Δ small).

Assuming rounding and the above statistical model for the quantisation noise, the quantisation noise is uniformly distributed over the interval $[-\Delta/2, \Delta/2]$, has zero mean ($E[e[n]]=0$) and its variance is

$$\sigma_e^2 = \int_{-\Delta/2}^{\Delta/2} e^2[n] \frac{1}{\Delta} \, de = \frac{\Delta^2}{12}$$

With a step size $\Delta = \frac{X_{max}}{2^B}$, the variance for a (B+1)-bit quantiser is given by

$$\sigma_e^2 = \frac{2^{-2B} X_m^2}{12}$$

The signal-to-quantisation noise ratio (SQNR), in decibels (dB), is

$$SQNR = 10 \log \frac{\sigma_x^2}{\sigma_e^2}$$

$$= 6.02B + 10.81 - 20 \log \frac{X_{max}}{\sigma_x},$$

where σ_x^2 is the signal power. The above expression indicates that the SQNR increases by 6 dB for each bit.

The output of the quantiser is sent to an encoder, which assigns a unique binary number (code word) to each quantisation level. Any assignment of codewords to levels may be used, and many encoding schemes exist. Most digital signal processing systems use the two's-complement representation. In this system, with a (B+1) bit codeword we have

$$C = [b_0 \; b_1 \ldots b_B],$$

where the leftmost or most significant bit, b_0, is the sign bit, and the remaining bits are used to represent either binary integers or fractions. Assuming binary fractions, the codeword $b_0 b_1 b_2 \ldots b_B$ has the value

$$x = [-1] b_0 + b_1 2^{-1} + b_2 2^{-2} + \ldots + b_B 2^{-B}$$

An example for a 3-bit codeword is given below

Table I

Binary Symbol	Numerical Value
011	3/4
010	1/2
001	1/4
000	0
111	-1/4
110	-1/2
101	-3/4
100	-1

c. Digital-to-Analog conversion

As stated in the sampling theorem, if $x_a(t)$ is strictly bandlimited so that $X_a(j\Omega) = 0$ for $|\Omega| > \Omega_0$, and if $T_s < \pi/\Omega_0$, then $x_a(t)$ may be uniquely reconstructed from its samples $x[n] = x_a[nT_s]$.

The reconstruction process involves two steps, as illustrated in the Figure below.

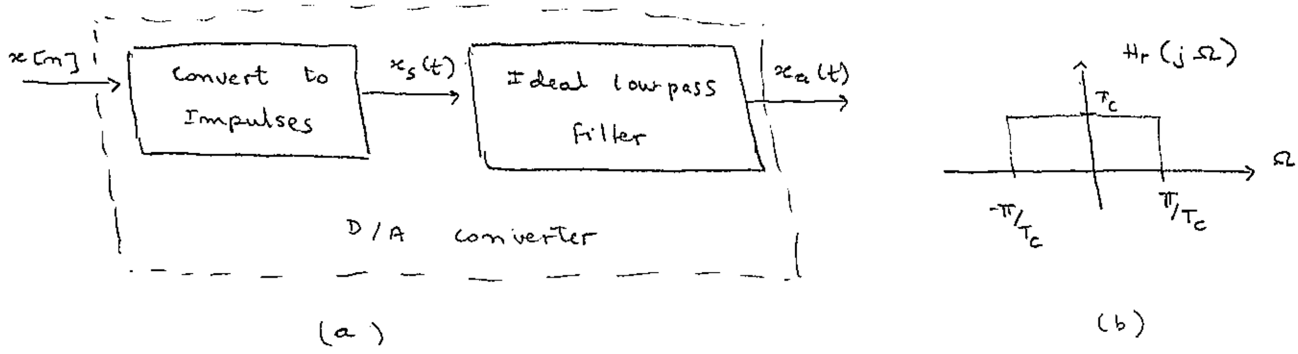


Fig. 5 (a) A D/A converter with an ideal low-pass filter
(b) The frequency response of the ideal reconstruction filter.

First, the samples are converted into a sequence of impulses:

$$x_s(t) = \sum_{n=-\infty}^{\infty} x[n]\,\delta(t - nT_s)$$

and then $x_a(t)$ is filtered with a reconstruction filter, which is an ideal low-pass filter that has a frequency response given by

$$H_r(j\Omega) = \begin{cases} T_s, & |\Omega| \leq \frac{\pi}{T_s} \\ 0, & |\Omega| > \frac{\pi}{T_s} \end{cases}$$

This system is called an ideal D/A converter because the impulse response of the reconstruction filter is

$$h_r(t) = \frac{\sin(\pi t/T_s)}{\pi t/T_s}$$

The output of the filter is

$$x_a(t) = \sum_{n=-\infty}^{\infty} x[n]\, h_r(t - nT_s) = \sum_{n=-\infty}^{\infty} x[n] \frac{\sin \pi(t-nT_s)/T_s}{\pi(t-nT_s)/T_s}$$

This interpolation formula shows how $x_a(t)$ is reconstructed from its samples $x[n] = x_a(nT_s)$.

In the frequency domain, the interpolation formula becomes

$$X_a(j\Omega) = \sum_{n=-\infty}^{\infty} x[n] \cdot H_r(j\Omega)\, e^{-jn\Omega T_s}$$

$$= H_r(j\Omega) \sum_{n=-\infty}^{\infty} x[n]\, e^{-jn\Omega T_s}$$

$$= H_r(j\Omega)\, X(e^{j\Omega T_s})$$

which is equivalent to

$$X_a(j\Omega) = \begin{cases} T_s X(e^{j\Omega T_s}) & , \quad |\Omega| < \frac{\pi}{T_s} \\ 0 & , \quad \text{otherwise} \end{cases}$$

Thus, $X(e^{j\omega})$ is frequency scaled and then the low-pass filter removes all frequencies in the periodic spectrum $X(e^{j\Omega T})$ above the cutoff frequency $\Omega_c = \pi/T_s$.

Since it is not possible to implement an ideal low-pass filter, many D/A converters use a zero-order hold for the reconstruction filter. The impulse response of a zero-order hold is

$$h_0(t) = \begin{cases} 1 & , \quad 0 \leq t \leq T_s \\ 0 & , \quad \text{otherwise} \end{cases}$$

and the frequency response is

$$H_0(j\Omega) = e^{-j\Omega T_s/2} \, \frac{\sin(\Omega T_s/2)}{\Omega/2}$$

After a sequence of samples $x_a(nT_s)$ has been converted to impulses, the zero-order hold produces the stair case approximation to $x_a(t)$ as shown in the next figure.

Fig. 6 The use of a zero-order hold to interpolate between the samples in $x_s(t)$.

With a zero-order hold, it is common to postprocess the output with a reconstruction compensation filter that approximates the frequency response

$$H_c(j\Omega) = \begin{cases} \dfrac{\Omega T_s/2}{\sin(\Omega T_s/2)} e^{j\Omega T_s/2}, & |\Omega| \le \dfrac{\pi}{T_s} \\ 0, & |\Omega| > \dfrac{\pi}{T_s} \end{cases}$$

so that the cascade of $H_0(e^{j\omega})$ with $H_c(e^{j\omega})$ approximates a low-pass filter with a gain of T_s over the pass band, as depicted in the figure below.

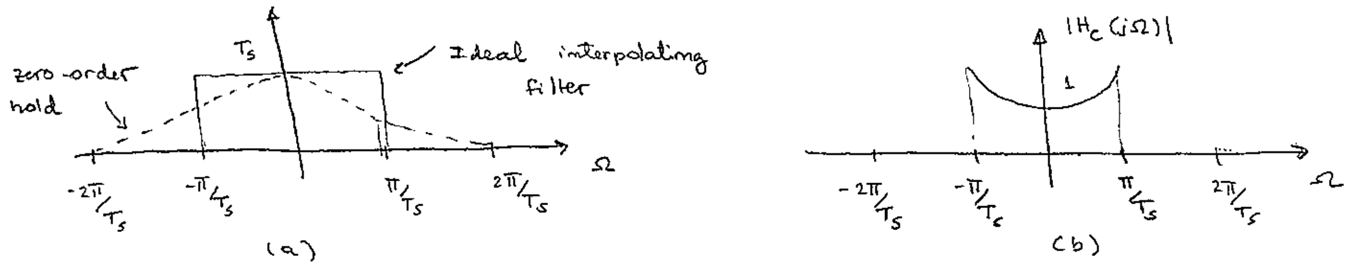


Fig. 7. (a) The magnitude of the frequency response of a zero-order hold compared to the ideal interpolating filter.

(b) The ideal reconstruction compensation filter.

Fig. 7 above shows the frequency response of a zero-order hold, the ideal interpolating filter and the ideal reconstruction compensation filter. Interestingly, the cascade of $H_c(j\Omega)$ with the zero-order hold is an ideal low-pass filter.

D. Discrete-Time Processing of Analog Signals

One of the important applications of A/D and D/A converters is the processing of analog signals with a discrete-time system. In the ideal case, the overall system consists of the cascade of an A/D converter, a discrete-time system and a D/A converter, as illustrated in the figure below.

$$x_a(t) \longrightarrow \boxed{\text{A/D converter}}_{\downarrow T_s} \xrightarrow{x[n]} \boxed{H(e^{j\omega})} \xrightarrow{y[n]} \boxed{\text{D/A converter}}_{\downarrow T_s} \longrightarrow y_a(t)$$

Fig. 8. Processing an analog signal using a discrete-time system.

We assume that the sampled signal is not quantised and that an ideal low-pass filter is used for the reconstruction filter in the D/A converter. Because the input $x_a(t)$ and the output $y_a(t)$ are analogous system, the overall system corresponds to a continuous-time system. To analyze this system, consider the discrete-time signal $x[n]$ at the output of the A/D converter, which has a DTFT given by

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

If the discrete-time system is linear and shift-invariant with a frequency response $H(e^{j\omega})$ then we have

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

The D/A converter produces the continuous-time signal $y_a(t)$ from the samples $y[n]$ as follows:

$$y_a(t) = \sum_{n=-\infty}^{\infty} y[n]\,\frac{\sin \pi(t-nT_s)/T_s}{\pi(t-nT_s)/T_s}$$

Either using the relation $X_a(j\Omega) = H_r(j\Omega)\cdot X(e^{j\Omega T_s})$ or by taking the DTFT directly, in the frequency domain this relationship becomes

$$Y_a(j\Omega) = H_r(j\Omega)\, Y(e^{j\Omega T_s})$$
$$= H_r(j\Omega)\, H(e^{j\Omega T_s})\, X(e^{j\Omega T_s})$$

or

$$Y_a(j\Omega) = H_r(j\Omega)\, H(e^{j\Omega T_s})\ \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X_a\left(j\Omega - j\frac{2\pi k}{T_s}\right)$$

If $x_a(t)$ is bandlimited with $X_a(j\Omega)=0$ for $|\Omega| > \pi/T_s$, the low-pass filter $H_r(j\Omega)$ eliminates all terms in the sum except the first one and results in

$$Y_a(j\Omega) = \begin{cases} H(e^{j\Omega T_s})\, X_a(j\Omega) & , \ |\Omega| \leq \dfrac{\pi}{T_s} \\ 0 & , \ |\Omega| > \dfrac{\pi}{T_s} \end{cases}$$

Therefore, the overall system behaves as a linear time-invariant continuous-time system with a frequency response given by

$$H_a(j\Omega) = \begin{cases} H(e^{j\Omega T_s}) & , \ |\Omega| \leq \dfrac{\pi}{T_s} \\ 0 & , \ \text{otherwise} \end{cases}$$

We can also implement a discrete-time system in terms of a continuous-time system as illustrated below.

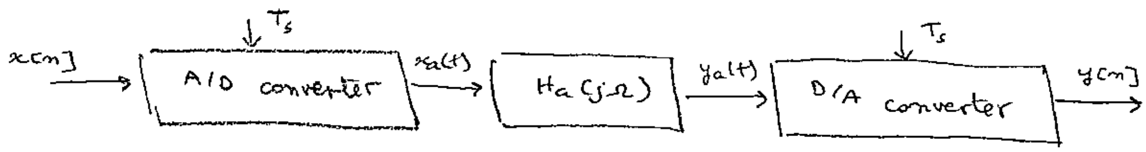


Fig. 9. Processing a discrete-time signal using a continuous-time system.

The signal $x_a(t)$ is related to the sequence values $x[n]$ as follows:

$$x_a(t) = \sum_{n=-\infty}^{\infty} x[n]\ \frac{\sin \pi(t-nT_s)/T_s}{\pi(t-nT_s)/T_s}$$

Because $x_a(t)$ is bandlimited, $y_a(t)$ is also bandlimited and can be represented in terms of its samples as follows:

$$y_a(t) = \sum_{n=-\infty}^{\infty} y[n]\ \frac{\sin \pi(t-nT_s)/T_s}{\pi(t-nT_s)/T_s}$$

The relationship between the Fourier transform of $x_a(t)$ and the DTFT of $x[n]$ is given by

$$X_a(j\Omega) = \begin{cases} T_s\, X(e^{j\Omega T_s}) , & |\Omega| < \frac{\pi}{T_s} \\ 0 , & \text{otherwise} \end{cases}$$

and the relationship between the Fourier transforms of $x_a(t)$ and $y_a(t)$ is described by

$$Y_a(j\Omega) = \begin{cases} H_a(j\Omega)\, X_a(j\Omega) , & |\Omega| < \frac{\pi}{T_s} \\ 0 , & \text{otherwise} \end{cases}$$

Therefore, we have

$$Y(e^{j\omega}) = \frac{1}{T_s} Y_a\left(\frac{j\omega}{T_s}\right) , \quad |\omega| < \pi$$

and the frequency response of the equivalent discrete-time system is

$$H(e^{j\omega}) = H_a\left(\frac{j\omega}{T_s}\right) , \quad |\omega| < \pi$$

Ex: Consider the discrete-time signal $x[n] = \cos\left(\frac{n\pi}{8}\right)$. Find two different continuous-time signals that would produce this sequence when sampled at a frequency of $f_s = 10$ kHz.

A continuous-time sinusoid $x_a(t) = \cos(\Omega_0 t) = \cos(2\pi f_0 t)$ that is sampled with a sampling frequency of f_s results in the sequence

$$x[n] = x_a(nT_s) = \cos\left(2\pi \frac{f_0}{f_s} n\right)$$

However, note that for any integer k we have

$$\cos\left(2\pi \frac{f_0}{f_s} n\right) = \cos\left(2\pi \frac{f_0 + kf_s}{f_s} n\right)$$

Therefore, any sinusoid with a frequency $f = f_0 + kf_s$ will produce the same sequence when sampled with f_s. With $x[n] = \cos(n\pi/8)$, we have

$$2\pi \frac{f_0}{f_s} = \frac{\pi}{8} \quad \text{or} \quad f_0 = \frac{1}{16} f_s = 625 \text{ Hz}$$

Thus, two signals that produce $x[n] = \cos\left(\frac{n\pi}{8}\right)$ are

$$x_1(t) = \cos(1250\pi t) \quad \text{and} \quad x_2(t) = \cos(21250\pi t)$$