

17

Interpolation

In developing the sampling theorem, we based the reconstruction procedure for recovering the original signal from its samples on the use of a lowpass filter. This follows naturally from the interpretation of the sampling process in the frequency domain. Correspondingly, in the time domain the reconstruction is represented by the convolution of the impulse train of samples with the impulse response of the lowpass filter. Convolution of an impulse response with an impulse train can be viewed as a superposition of weighted delayed impulse responses with amplitudes and positions corresponding to the impulses in the impulse train. This superposition represents an interpolation process between the samples. When the reconstruction filter is an ideal lowpass filter, the interpolating function is a sinc function. This is often referred to as *bandlimited interpolation* because it interpolates between sample points by explicitly assuming that the original signal is bandlimited to less than half the sampling frequency.

In addition to bandlimited interpolation, a variety of other interpolation procedures are commonly used. One, referred to as a *zero-order hold*, interpolates between sample points by holding each sample value until the next sampling instant. This generates a staircase-like approximation to the original signal. Linear interpolation, also commonly referred to as a *first-order hold*, corresponds to connecting the sample points by straight line segments. Both the zero-order hold and first-order hold can be alternatively viewed in much the same way as we have discussed ideal bandlimited interpolation. Specifically, the zero-order hold corresponds to convolving the impulse train of samples with a rectangular pulse of duration exactly equal to the sampling period. The first-order hold corresponds to an impulse response for the reconstruction filter that is a triangle of duration equal to twice the sampling period. In the frequency domain, then, the zero-order hold corresponds to processing the samples with an approximation to a lowpass filter corresponding to the Fourier transform of a rectangular pulse. With the first-order hold the approximate lowpass filter has a frequency response that is the Fourier transform of a triangle.

One of the important applications of the concept of sampling is its use in converting continuous-time signals to discrete-time signals corresponding to a sequence of sample values. This provides one basis for storing, coding, or

transmitting continuous-time signals. In addition, it offers the possibility for discrete-time processing of continuous-time signals. In many situations such processing is highly advantageous. For example, digital technologies for signal processing, which inherently are oriented toward discrete-time signals and systems, are extremely flexible and often lend themselves to implementing more sophisticated and flexible algorithms than a continuous-time system might. By exploiting the sampling theorem, a continuous-time signal to be processed can be converted to a discrete-time signal, processed by a discrete-time system, and then converted back to a continuous-time signal. In developing insight into this process, it is important to clearly understand in both the time and the frequency domains the process of converting from a continuous-time signal to a sequence of samples. This continuous-to-discrete-time conversion (abbreviated as C/D) is conveniently thought of in two stages. The first represents sampling of the continuous-time signal with a periodic impulse train to generate an impulse train of samples. This impulse train is then converted to a discrete-time sequence essentially by relabeling; that is, a discrete-time sequence is generated in which each impulse is represented by its area. After the first stage, the impulses in the impulse train occur at multiples of the sampling period. After the second stage, the discrete-time sequence representing the impulse values is indexed on sample number and consequently the sample spacing has been normalized to unity. For example, if a continuous-time signal were to be sampled and stored in a computer memory, it would first be sampled in time and the sample values converted through an analog-to-digital converter to digital numbers. These numbers would then be placed in memory. The resulting discrete-time sequence would be the sequence of numbers in successive memory locations, and the independent variable indexing the discrete-time sequence could in fact be thought of as memory location number.

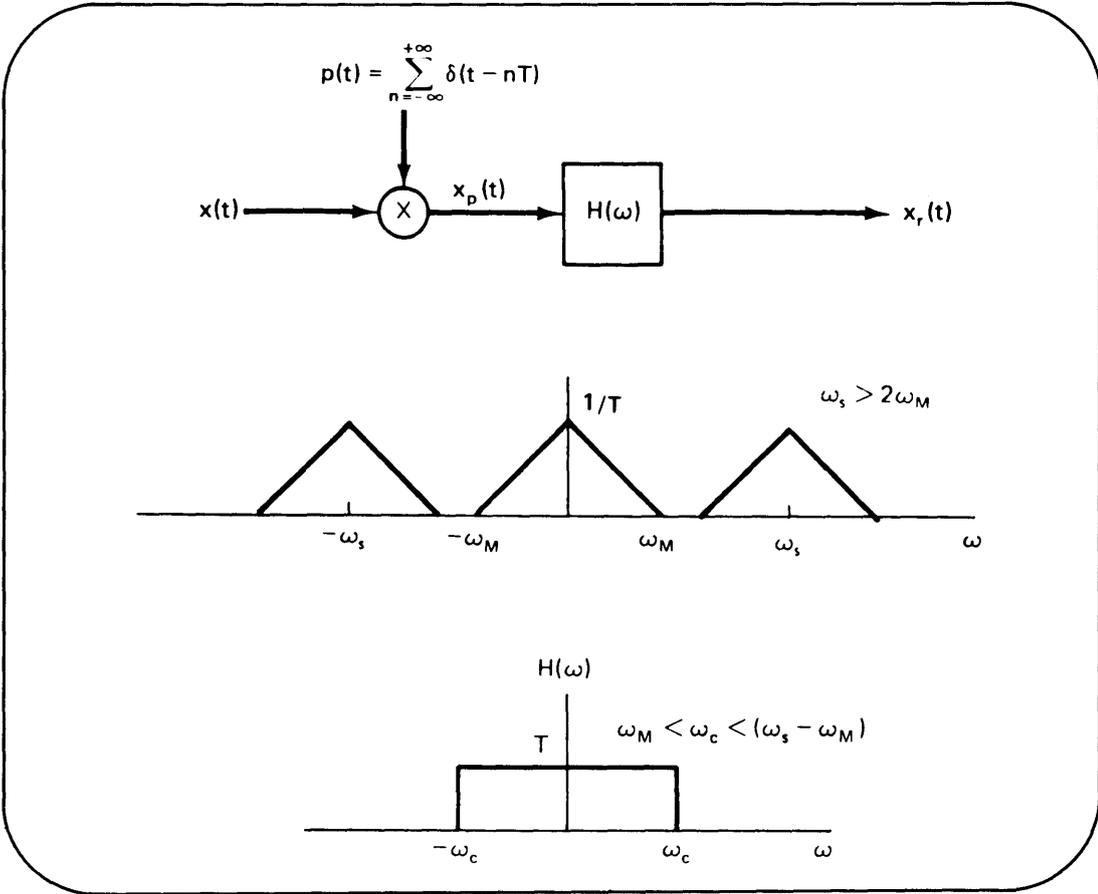
In the frequency domain, the two-step process described above has a relatively straightforward interpretation. Through the process of sampling, assuming that the continuous-time signal is bandlimited and the conditions of the sampling theorem are met, the spectrum of the continuous-time signal is periodically replicated at integer multiples of the sampling frequency. Conversion of the impulse train to a discrete-time sequence corresponds in the time domain to a time normalization, in effect normalizing out the sampling period. Correspondingly, in the frequency domain, the frequency axis is normalized with the sampling frequency being scaled to a discrete-time frequency of 2π . Thus, as we naturally expect, the Fourier transform of the discrete-time sequence is periodic with a period of 2π . The periodicity can be interpreted as being a consequence of the basic sampling process. The normalization of the period in frequency to 2π is a consequence of the inherent time normalization in converting the impulse train of samples to a discrete-time sequence.

Suggested Reading

Section 8.1.2, Sampling with a Zero-Order Hold, pages 519–521

Section 8.2, Reconstruction of a Signal from Its Samples Using Interpolation, pages 521–526

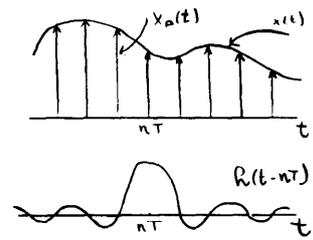
Section 8.4, Discrete-Time Processing of Continuous-Time Signals, pages 531 to mid-537



TRANSPARENCY
17.1
 System and an illustration of associated spectra for sampling and reconstruction.

MARKERBOARD
 17.1 (a)

$$\begin{aligned}
 x_p(t) &= x(t)p(t) \\
 &= x(t) \sum_{n=-\infty}^{+\infty} \delta(t - nT) \\
 &= \sum_{n=-\infty}^{+\infty} x(nT)\delta(t - nT)
 \end{aligned}$$



$$x_r(t) = x_p(t) * h(t)$$

$$x_r(t) = \sum_{n=-\infty}^{+\infty} x(nT)h(t - nT)$$

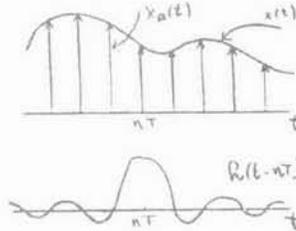
For $H(\omega)$ an ideal Lowpass filter with cutoff frequency ω_c ,

$$h(t) = T \frac{\omega_c}{\pi} \text{sinc}\left(\frac{\omega_c t}{\pi}\right)$$

MARKERBOARD

17.1 (b)

$$\begin{aligned}
 x_p(t) &= x(t)p(t) \\
 &= x(t) \sum_{n=-\infty}^{+\infty} \delta(t-nT) \\
 &= \sum_{n=-\infty}^{+\infty} x(nT)\delta(t-nT)
 \end{aligned}$$



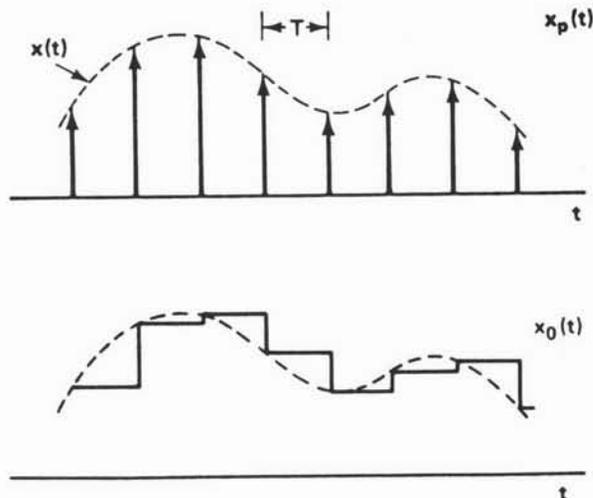
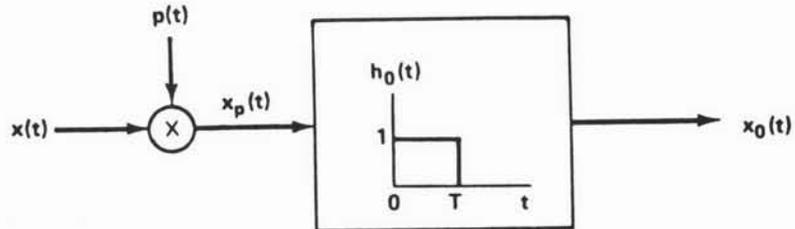
$$x_r(t) = x_p(t) * h(t)$$

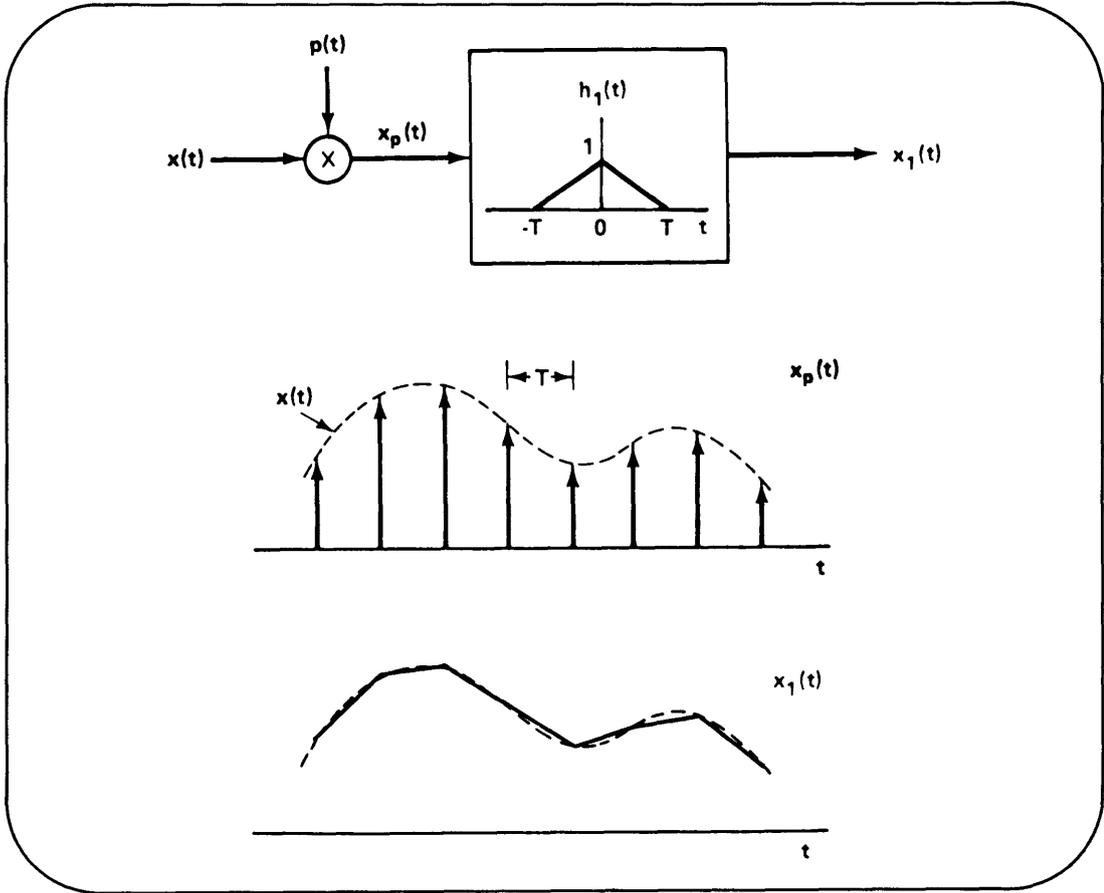
$$x_r(t) = \sum_{n=-\infty}^{+\infty} x(nT)h(t-nT)$$

For $H(\omega)$ an ideal Lowpass filter with cutoff frequency ω_c ,

$$h(t) = T \frac{\omega_c}{\pi} \text{sinc}\left(\frac{\omega_c t}{\pi}\right)$$

TRANSPARENCY
17.2
Reconstruction of a sampled signal with a zero-order hold.

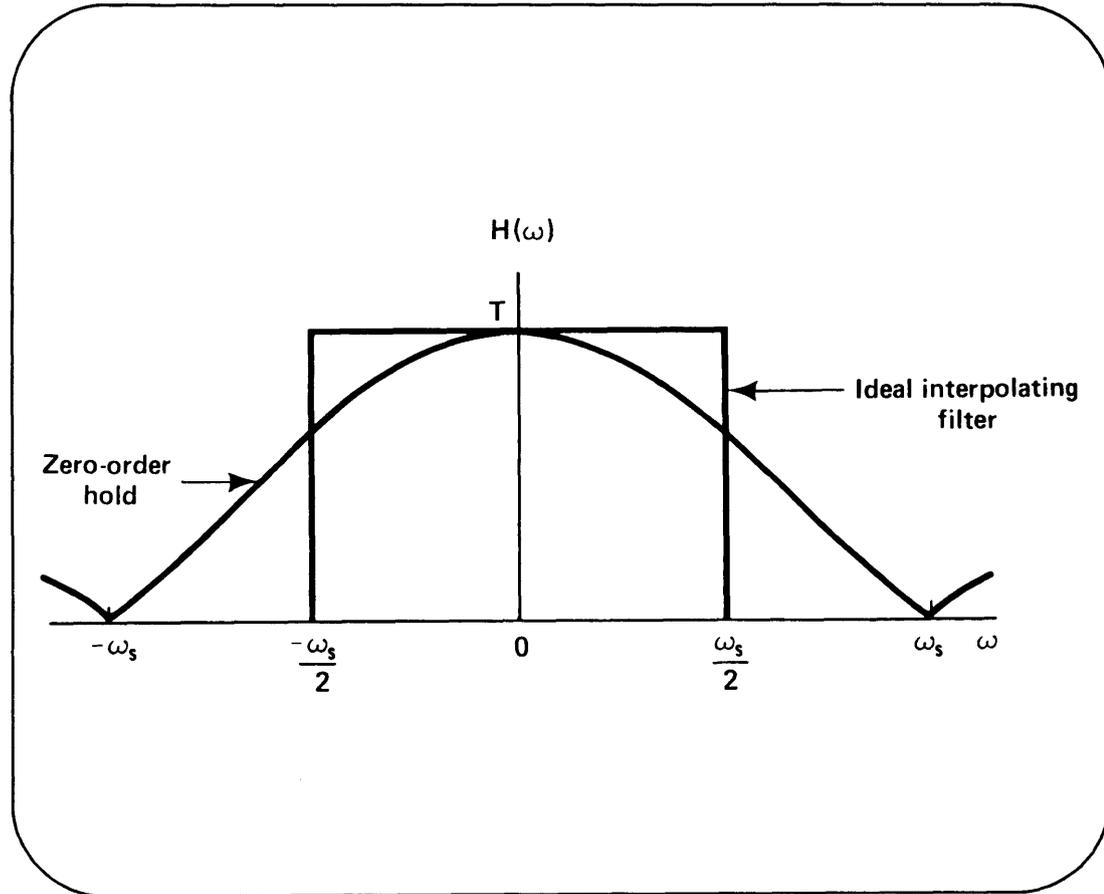




TRANSPARENCY

17.3

Reconstruction of a sampled signal with a first-order hold.



TRANSPARENCY

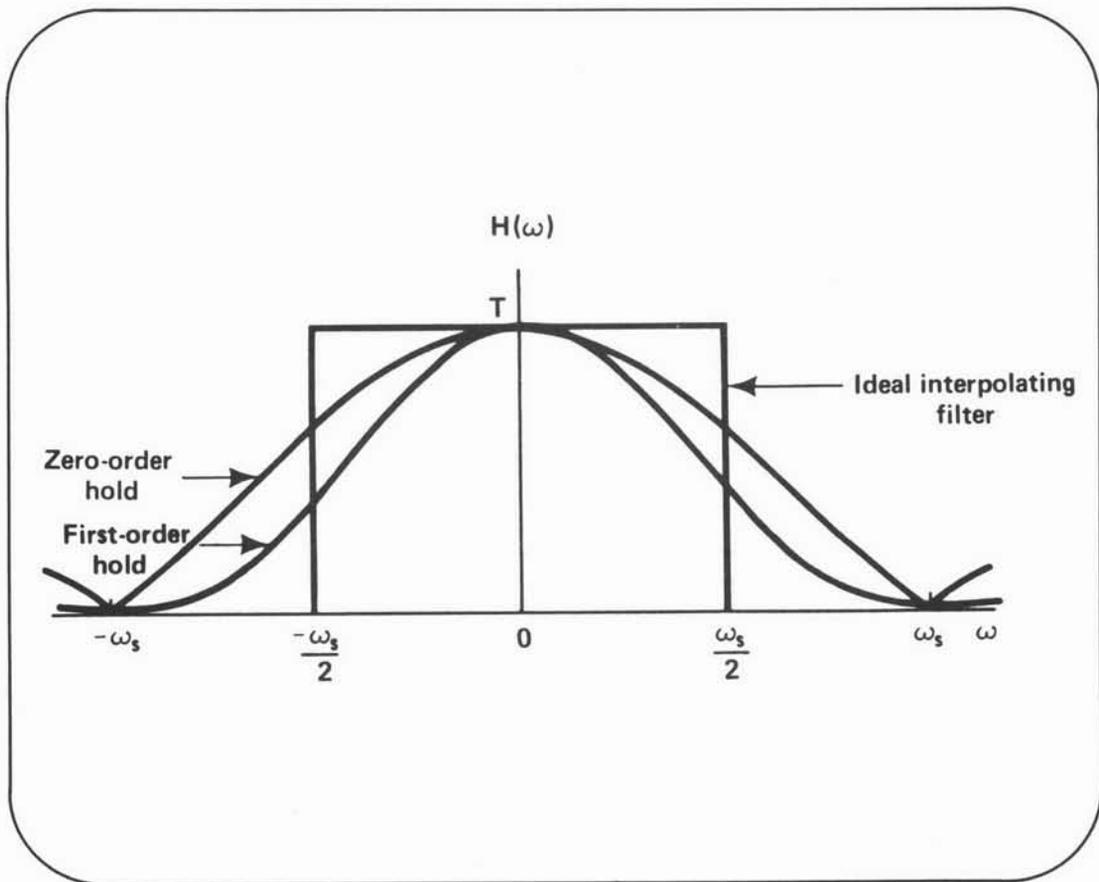
17.4

Transparencies 17.4 and 17.5 illustrate a comparison of the frequency responses of the ideal reconstruction filter and the reconstruction filter corresponding to a zero-order hold and a first-order hold. Shown here is the ideal reconstruction filter corresponding to a zero-order hold.

TRANSPARENCY

17.5

Ideal reconstruction filter, reconstruction filter corresponding to a first-order hold, and reconstruction filter corresponding to a zero-order hold.

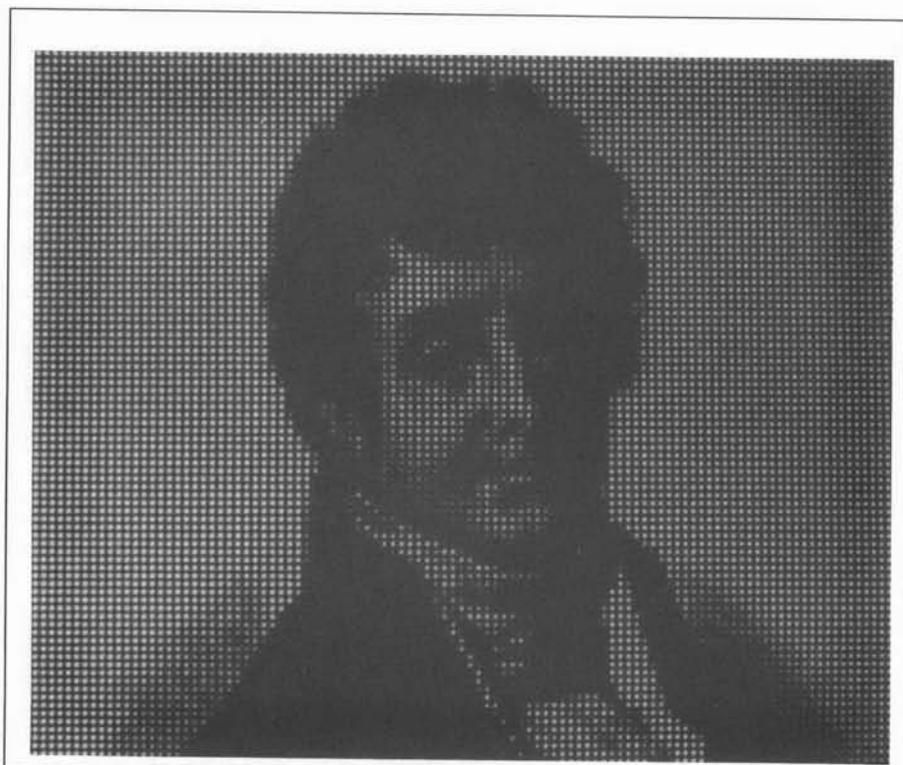


DEMONSTRATION

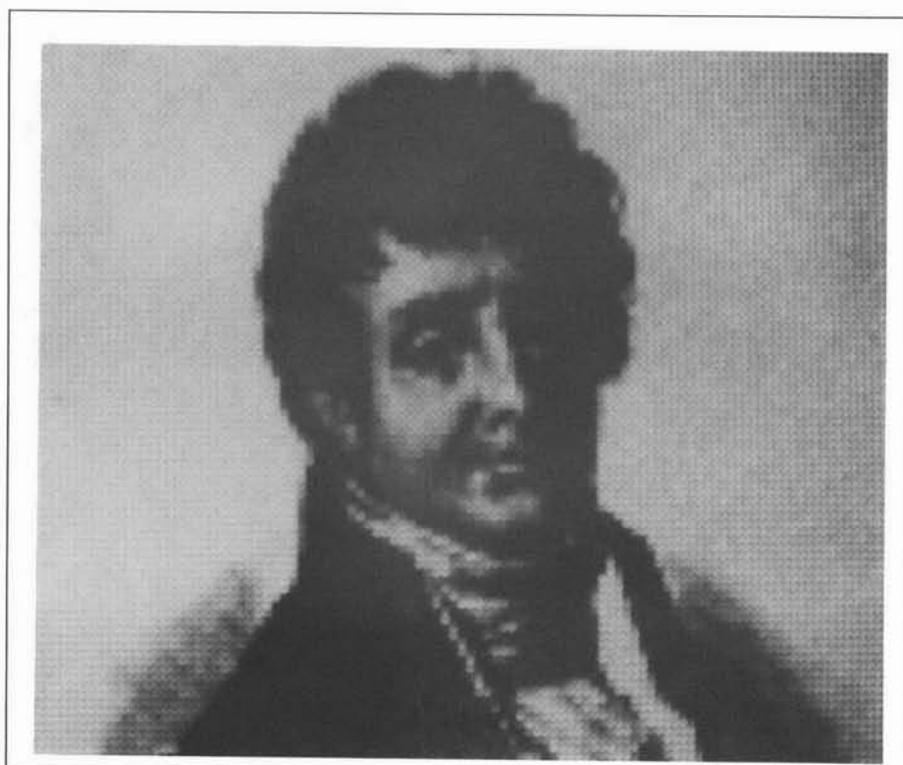
17.1

Original image.





**DEMONSTRATION
17.2**
Image of
Demonstration 17.1
spatially sampled.



**DEMONSTRATION
17.3**
Optical lowpass
filtering (defocusing)
of spatial sampling.

DEMONSTRATION
17.4
A zero-order hold
applied to the
sampled image of
Demonstration 17.1.



DEMONSTRATION
17.5
Defocusing of
Demonstration 17.4.





**DEMONSTRATION
17.6**
Close-up of
Demonstration 17.4.



**DEMONSTRATION
17.7**
Close-up of
Demonstration 17.4
defocused.

DEMONSTRATION

17.8

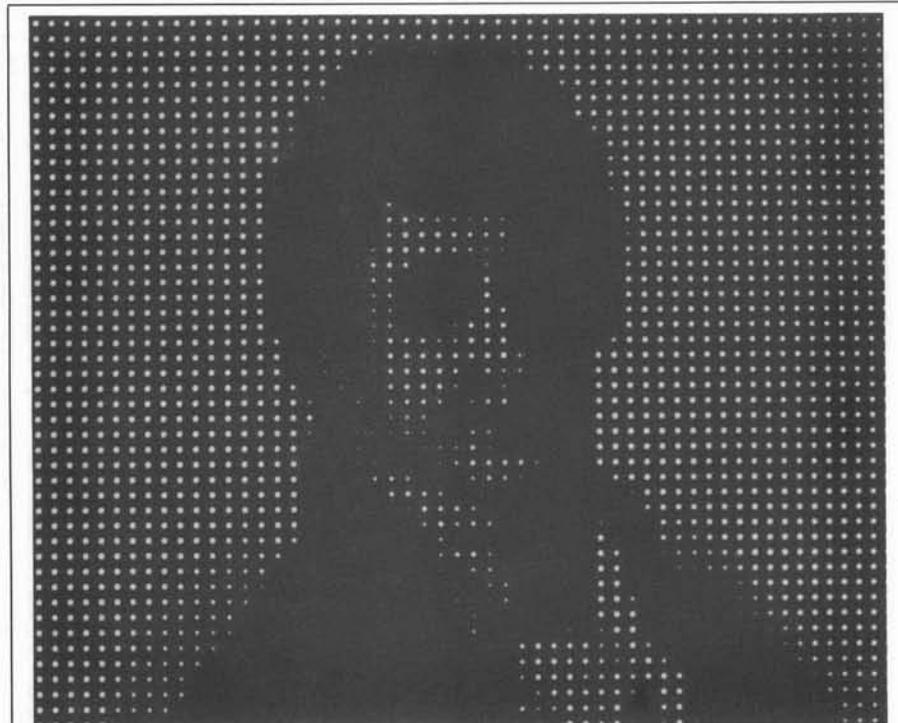
A first-order hold applied to the sampled image of Demonstration 17.1.



DEMONSTRATION

17.9

Image of Demonstration 17.1 spatially sampled with twice the sample spacing as in Demonstration 17.2.





**DEMONSTRATION
17.10**
A zero-order hold
applied to the
sampled image of
Demonstration 17.9.

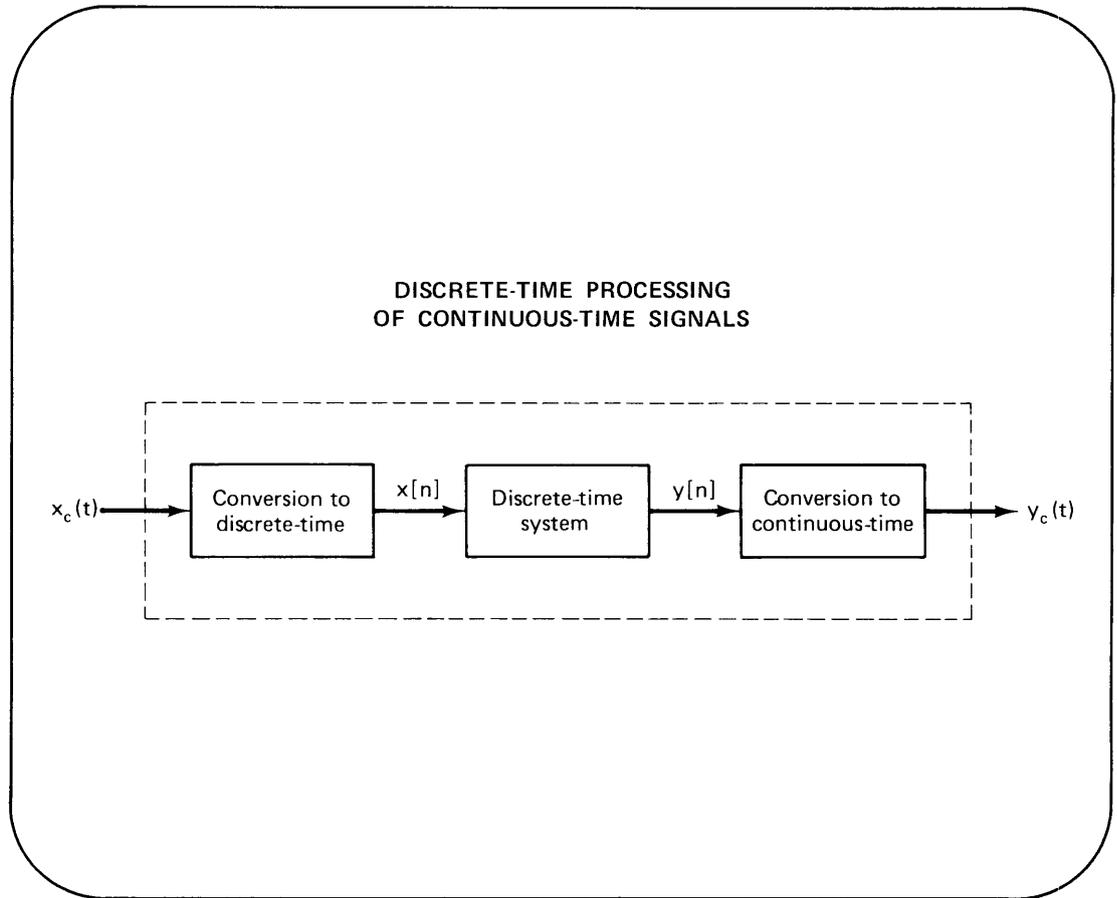


**DEMONSTRATION
17.11**
A first-order hold
applied to the
sampled image of
Demonstration 17.9.

TRANSPARENCY

17.6

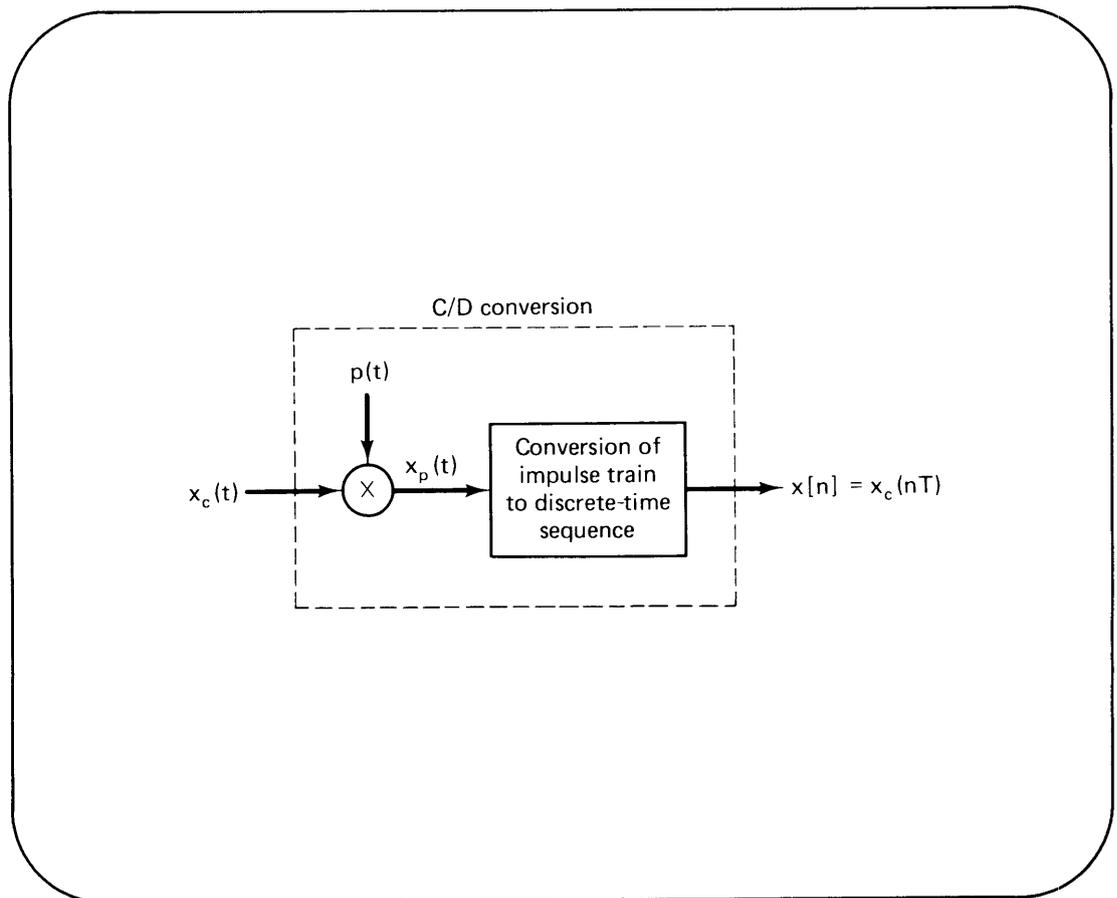
Discrete-time processing of continuous-time signals.



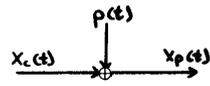
TRANSPARENCY

17.7

Interpretation of the conversion of a continuous-time signal to a discrete-time sequence as impulse train sampling followed by conversion of the impulse train to a discrete-time sequence.



MARKERBOARD
17.2



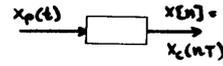
$$x_p(t) = x_c(t) \sum_{n=-\infty}^{+\infty} \delta(t-nT)$$

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

$$X_p(\omega) = \frac{1}{2\pi} [X_c(\omega) * P(\omega)]$$

$$P(\omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{+\infty} \delta(\omega - k\frac{2\pi}{T})$$

$$X_p(\omega) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} X_c(\omega - k\frac{2\pi}{T})$$



$$x_p(t) = \sum_{k=-\infty}^{+\infty} x_c(nT) \delta(t-nT)$$

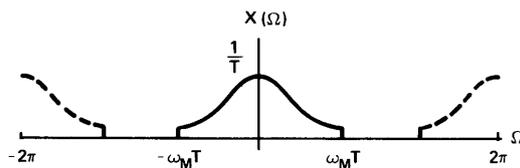
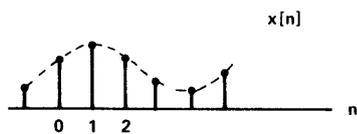
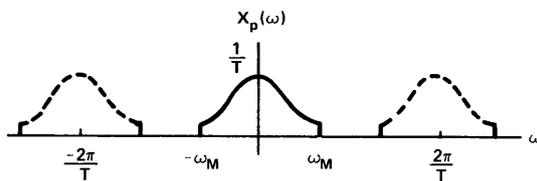
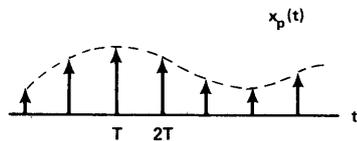
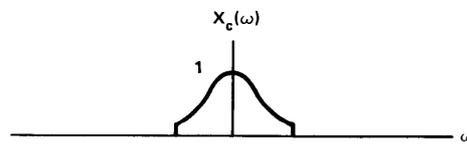
$$X_p(\omega) = \sum_{n=-\infty}^{+\infty} x_c(nT) e^{-j\omega nT}$$

$$X(\Omega) = \sum_{n=-\infty}^{+\infty} x[n] e^{-j\Omega n}$$

$\Omega = \omega T$

$$X(\Omega) = X_p(\omega) \Big|_{\omega T = \Omega}$$

$$X(\Omega) = X_p(\Omega/T)$$



TRANSPARENCY

17.8

Illustration in the time domain and frequency domain of conversion of a signal from continuous time to discrete time.

MIT OpenCourseWare
<http://ocw.mit.edu>

Resource: Signals and Systems
Professor Alan V. Oppenheim

The following may not correspond to a particular course on MIT OpenCourseWare, but has been provided by the author as an individual learning resource.

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.