19 Discrete-Time Sampling

In the previous lectures we discussed sampling of continuous-time signals. In this lecture we address the parallel topic of discrete-time sampling, which has a number of important applications. The basic concept of discrete-time sampling is similar to that of continuous-time sampling. Specifically, we multiply a discrete-time sequence by a periodic impulse train, thus retaining every Nth sample and setting the remaining ones to zero (where N denotes the period of the sampling impulse train). The consequences in the frequency domain and the constraints on the bandwidth of the original sequence such that it can be recovered from its samples parallel those for continuous time. Under the constraints of the sampling theorem, exact interpolation can again be implemented with an ideal lowpass filter.

Closely associated with, but not identical to, the concept of discrete-time sampling is that of decimation or downsampling. After sampling a sequence with an impulse train, we have obtained a new sequence that is nonzero only at multiples of the sampling period N. Consequently, in many practical situations there is no reason to explicitly retain these zero values since they can always be reinserted. Thus, somewhat distinct from the notion of sampling is the concept of decimation, whereby a new sequence is generated from the original sequence by selecting every Nth sample. This in effect results in a time compression. Although not typically implemented this way, it can be thought of as a two-step process, the first step consisting of periodic sampling and the second step corresponding to discarding the zero values between the samples. Decimation is also commonly referred to as downsampling since if the original sequence resulted from time sampling a continuous-time signal, the new sequence resulting from decimation would be exactly what would have been obtained had a lower sampling rate been used originally. If, for example, a continuous-time signal is sampled at or near the Nyquist rate and is then processed by a discrete-time system that provides some further band-limiting, downsampling or decimation is often used.

The reverse of downsampling is "upsampling," whereby we attempt to reconstruct the original sequence. The process is again best thought of in two stages, the first corresponding to converting the decimated sequence to a sampled sequence by reinserting the $(N-1)$ zero values between the sample points. The second stage is interpolation with a lowpass filter to construct the original sequence.

The processes of downsampling and upsampling have a number of practical implications. One, as indicated above, is sampling rate conversion after additional processing. Another very important one is converting a sequence from one sampling rate to another perhaps to generate compatibility between otherwise incompatible systems. For example, it is often important to convert between different digital audio systems that use different sampling rates.

In this lecture we also briefly discuss the concept of sampling in the frequency domain. Frequency-domain sampling typically arises when we would like to measure or explicitly evaluate numerically the Fourier transform. Although in general the Fourier transform for both continuous time and discrete time is a function of a continuous-frequency variable, the measurement or calculation must be made only at a set of sample frequencies. Because of the duality between the time and frequency domains for continuous time, the issues, analysis, and concepts related to frequency-domain sampling for continuous-time signals are exactly dual to those of time-domain sampling. Thus, for example, the Fourier transform can exactly be recovered from equally spaced samples in the frequency domain provided that the time-domain signal is timelimited (the dual of bandlimited). Basically, the same result applies in discrete time, i.e., the Fourier transform of a timelimited sequence can be exactly represented by and recovered from equally spaced samples provided that the sample spacing in frequency is sufficiently small in relation to the time duration of the signal in the time domain.

Suggested Reading

Section 8.6, Sampling of Discrete-Time Signals, pages 543–548

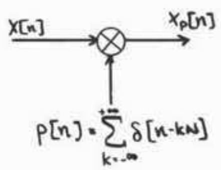
Section 8.7, Discrete-Time Decimation and Interpolation, pages 548–553

Section 8.5, Sampling in the Frequency Domain, pages 540–543

MARKERBOARD 19.1 (a)

Discrete-Time Sampling

- Resampling after Discrete-Time Filtering
- Sampling Rate Conversion
- decimation/interpolation



$$x_p[n] = p[n]\,x[n]$$

$$X_p(\Omega) = \frac{1}{2\pi}\int_{2\pi} P(\theta)\,X(\Omega-\theta)\,d\theta$$

$$p[n] = \sum_{k=-\infty}^{+\infty} \delta[n-kN]$$

$$P(\Omega) = \frac{2\pi}{N}\sum_{k=-\infty}^{+\infty} \delta\left(\Omega - k\frac{2\pi}{N}\right)$$

$k\Omega_s$

$$\Omega_s = \frac{2\pi}{N}$$

$$X_p(\Omega) = \frac{1}{N}\sum_{k=0}^{N-1} X\left(\Omega - k\frac{2\pi}{N}\right)$$

$k\Omega_s$

$$x_d[n] = x[nN] = x_p[nN]$$

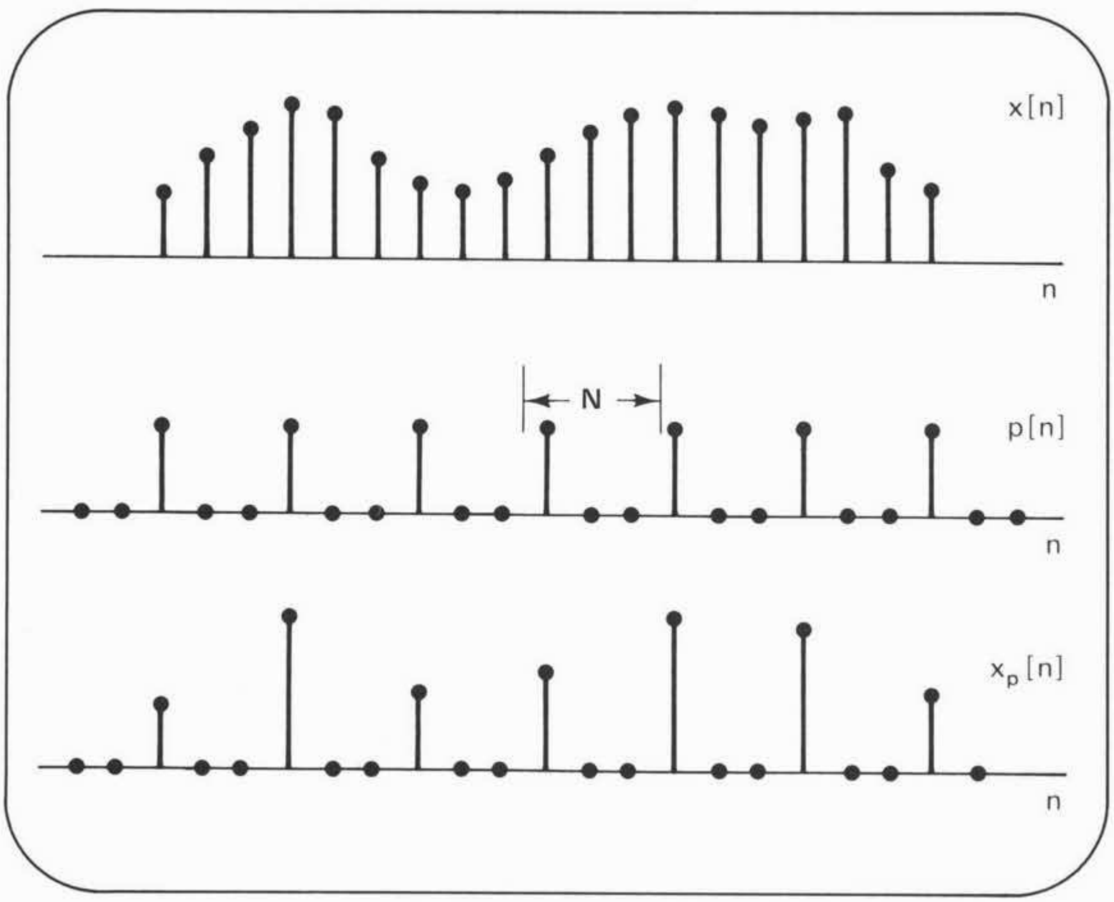
$$X_p(\Omega) = \sum_{n=-\infty}^{+\infty} x_p[n]\,e^{-j\Omega n}$$

$$n = mN$$

$$X_p(\Omega) = \sum_{m=-\infty}^{+\infty} x_p[mN]\,e^{-j\Omega mN}$$

$$X_d(\Omega) = \sum_{m=-\infty}^{+\infty} \underbrace{x_d[m]}_{x_p[mN]}\,e^{-j\Omega m}$$

$$X_d(\Omega) = X_p(\Omega/N)$$



TRANSPARENCY 19.1
Discrete-time sampling with a periodic impulse train.

MARKERBOARD
19.1 (b)

Discrete-Time Sampling

- Resampling after Discrete-Time Filtering
- Sampling Rate Conversion
- decimation/interpolation

$$x[n] \rightarrow \otimes \rightarrow x_p[n]$$

$$p[n] = \sum_{k=-\infty}^{+\infty} \delta[n-kN]$$

$$x_p[n] = p[n]\, x[n]$$

$$X_p(\Omega) = \frac{1}{2\pi}\int_{2\pi} P(\theta)\, X(\Omega-\theta)\, d\theta$$

$$p[n] = \sum_{k=-\infty}^{+\infty} \delta[n-kN]$$

$$P(\Omega) = \frac{2\pi}{N}\sum_{k=-\infty}^{+\infty} \delta\Big(\Omega - \underbrace{k\frac{2\pi}{N}}_{k\Omega_s}\Big) \qquad \Omega_s = \frac{2\pi}{N}$$

$$X_p(\Omega) = \frac{1}{N}\sum_{k=0}^{N-1} X\Big(\Omega - \underbrace{k\frac{2\pi}{N}}_{k\Omega_s}\Big)$$

$$x_d[n] = x[nN] = x_p[nN]$$

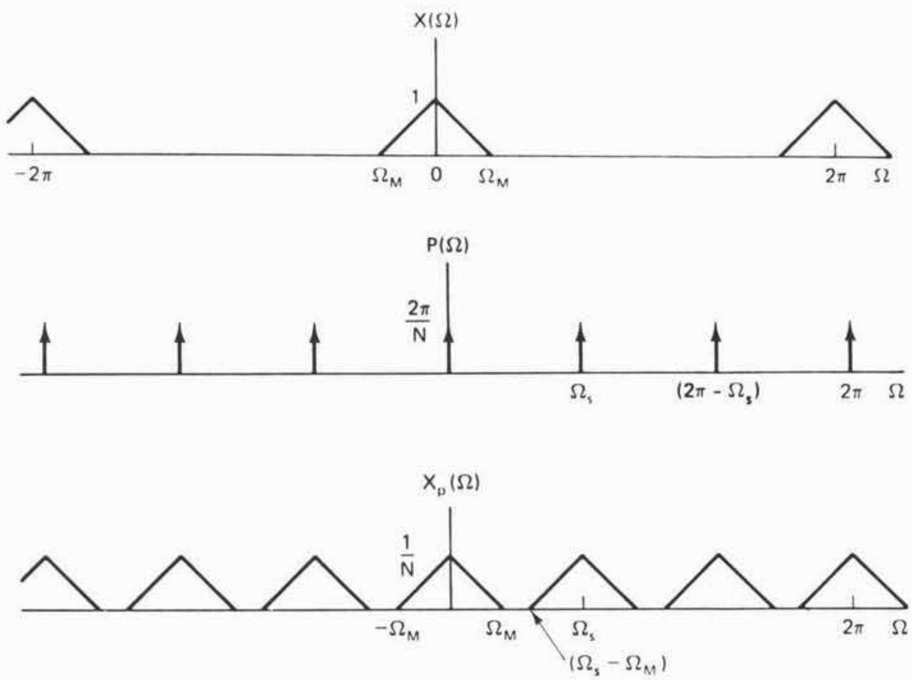
$$X_p(\Omega) = \sum_{n=-\infty}^{+\infty} x_p[n]\, e^{-j\Omega n}$$

$$n = mN$$

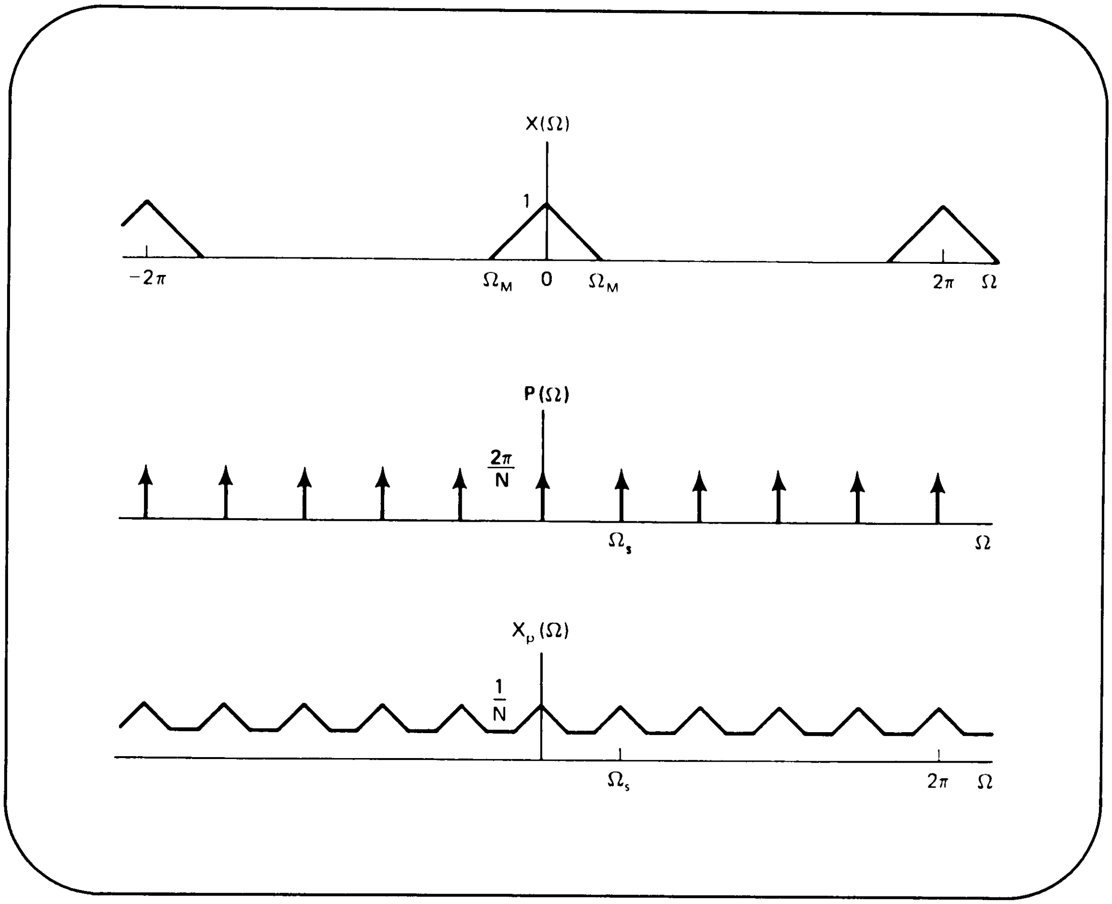
$$X_p(\Omega) = \sum_{m=-\infty}^{+\infty} x_p[mN]\, e^{-j\Omega mN}$$

$$X_d(\Omega) = \sum_{m=-\infty}^{+\infty} \underbrace{x_d[m]}_{x_p[mN]}\, e^{-j\Omega m}$$

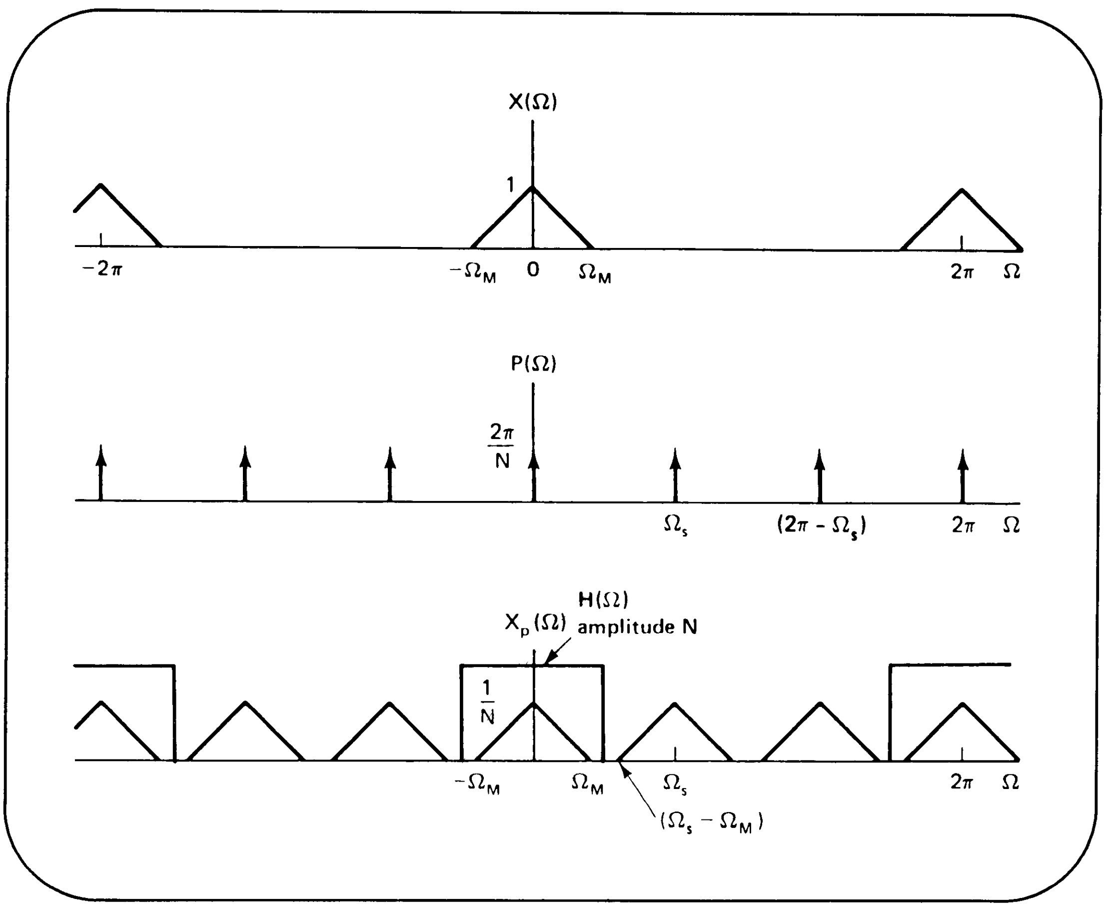
$$X_d(\Omega) = X_p(\Omega/N)$$



TRANSPARENCY
19.2
Illustration of spectra associated with discrete-time sampling. The sampling rate is sufficiently high to avoid aliasing.

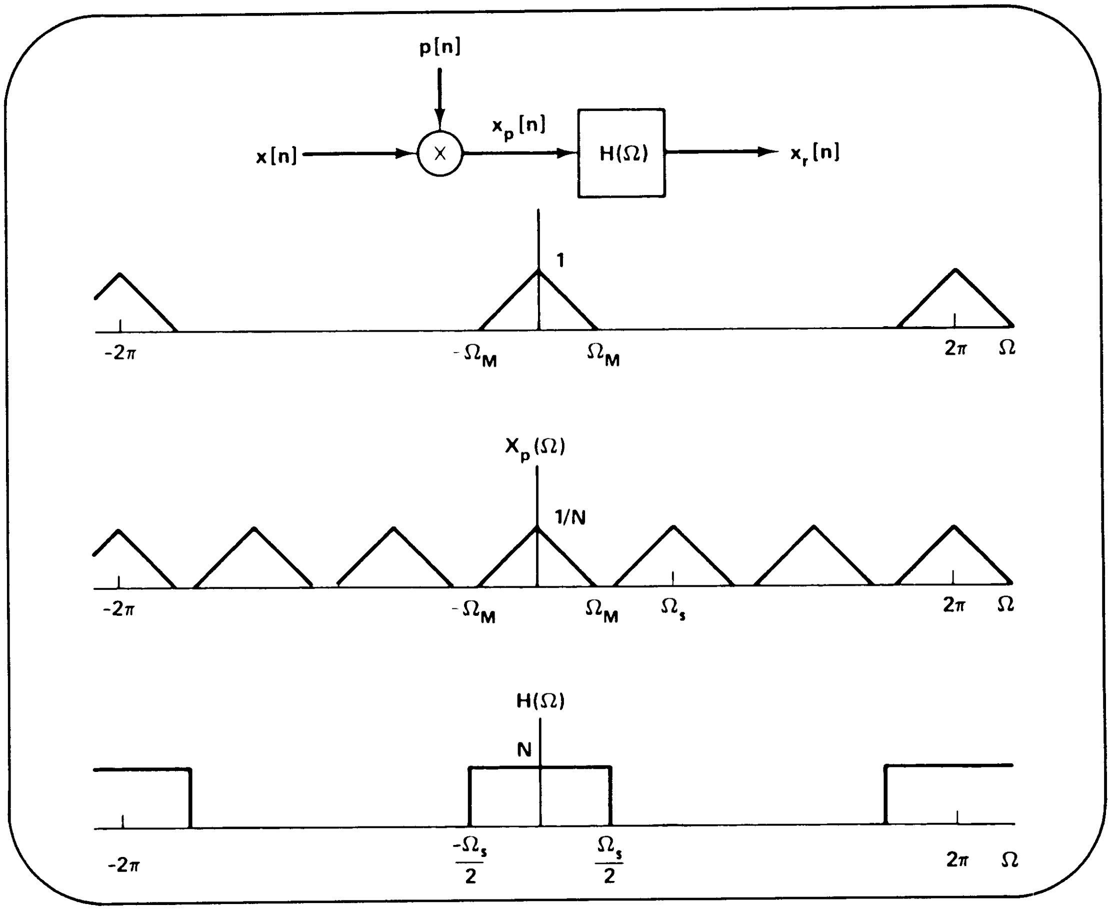


TRANSPARENCY 19.3
Illustration of spectra associated with discrete-time sampling when the sampling rate is too low to avoid aliasing.

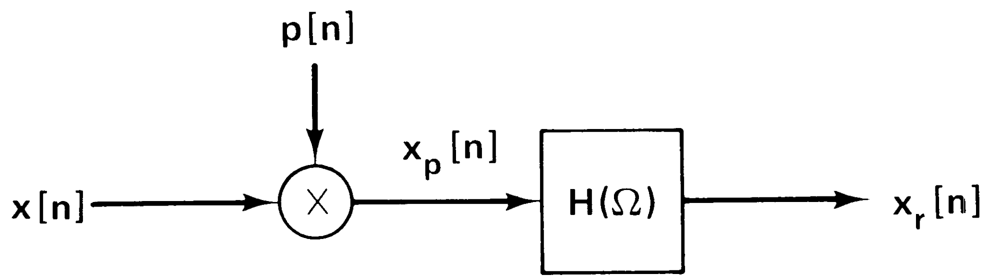


TRANSPARENCY 19.4
Illustration of the recovery of the original discrete-time spectrum from the spectrum of the sampled signal using an ideal lowpass filter.

TRANSPARENCY 19.5
Overall system for discrete-time sampling and reconstruction.



TRANSPARENCY 19.6
Discrete-time sampling and reconstruction with the reconstruction interpreted in the time domain as a process of interpolation.

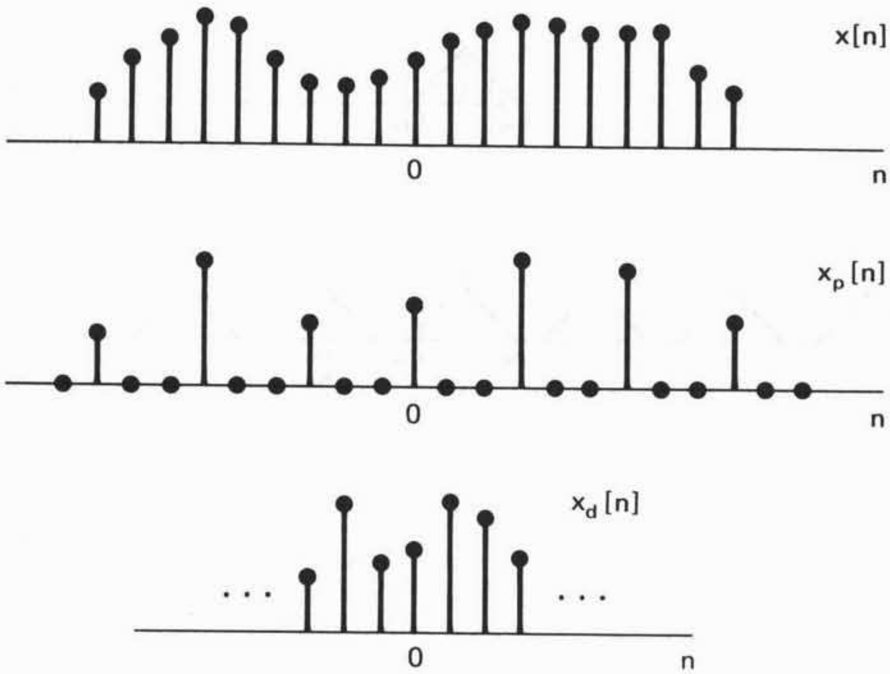


$$x_p[n] = x[n]\, p[n]$$

$$x_p[n] = \sum_{k=-\infty}^{+\infty} x[kN]\, \delta[n-kN]$$

$$x_r[n] = x_p[n] * h[n]$$

$$x_r[n] = \sum_{k=-\infty}^{+\infty} x[kN]\, h[n-kN]$$



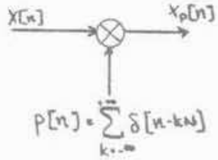
$$X_d(\Omega) = X_p\left(\frac{\Omega}{N}\right)$$

TRANSPARENCY 19.7
Relationship between sampling and decimation.

MARKERBOARD 19.1 (c)

Discrete-Time Sampling

- Resampling after Discrete-Time Filtering
- Sampling Rate Conversion
- decimation/interpolation



$$x_p[n] = p[n]\,x[n]$$

$$X_p(\Omega) = \frac{1}{2\pi}\int_{2\pi} P(\theta)\,X(\Omega-\theta)\,d\theta$$

$$p[n] = \sum_{k=-\infty}^{+\infty} \delta[n-kN]$$

$$P(\Omega) = \frac{2\pi}{N}\sum_{k=-\infty}^{+\infty} \delta\left(\Omega - k\frac{2\pi}{N}\right)$$

$k\Omega_s$ $\quad \Omega_s = \frac{2\pi}{N}$

$$X_p(\Omega) = \frac{1}{N}\sum_{k=0}^{N-1} X\left(\Omega - k\frac{2\pi}{N}\right)$$

$k\Omega_s$

$$x_d[n] = x[nN] = x_p[nN]$$

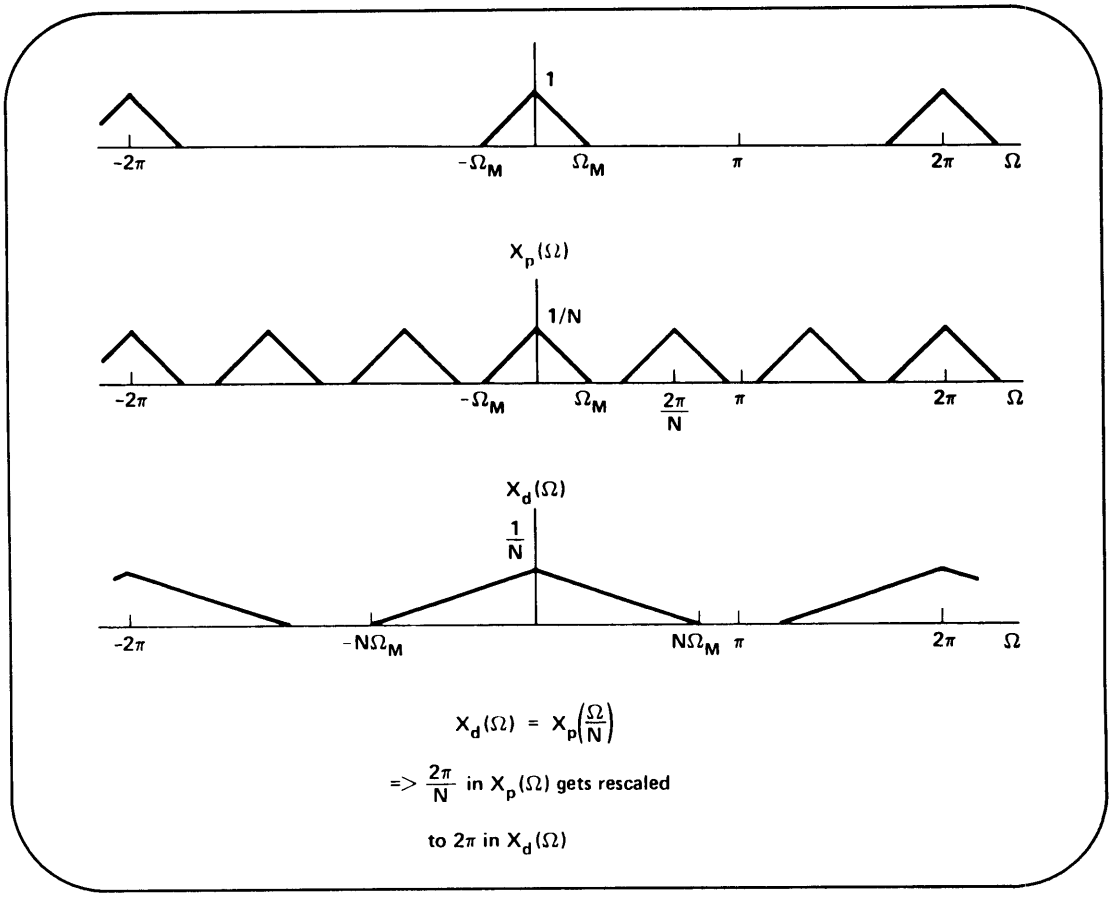
$$X_p(\Omega) = \sum_{n=-\infty}^{+\infty} x_p[n]\,e^{-j\Omega n}$$

$$n = mN$$

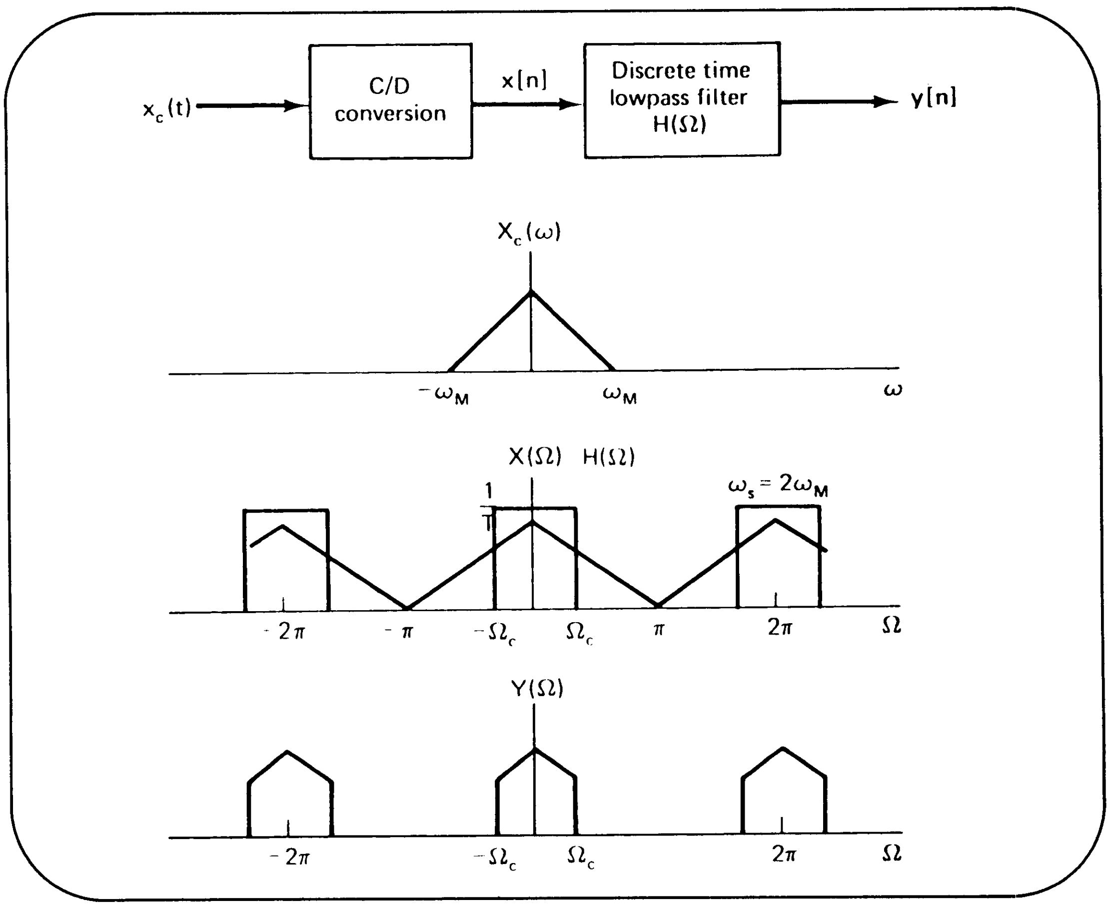
$$X_p(\Omega) = \sum_{m=-\infty}^{+\infty} x_p[mN]\,e^{-j\Omega mN}$$

$$X_d(\Omega) = \sum_{m=-\infty}^{+\infty} \underbrace{x_d[m]}_{x_p[mN]}\,e^{-j\Omega m}$$

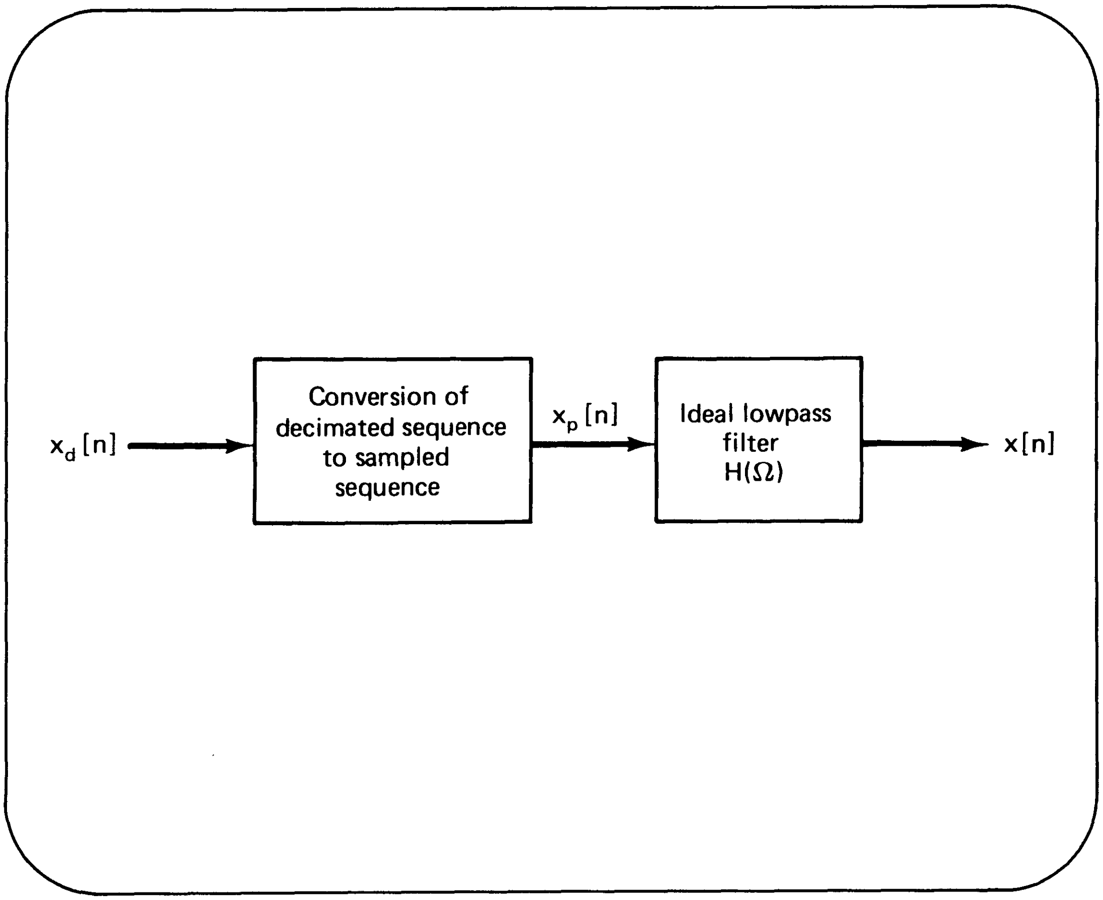
$$X_d(\Omega) = X_p(\Omega/N)$$



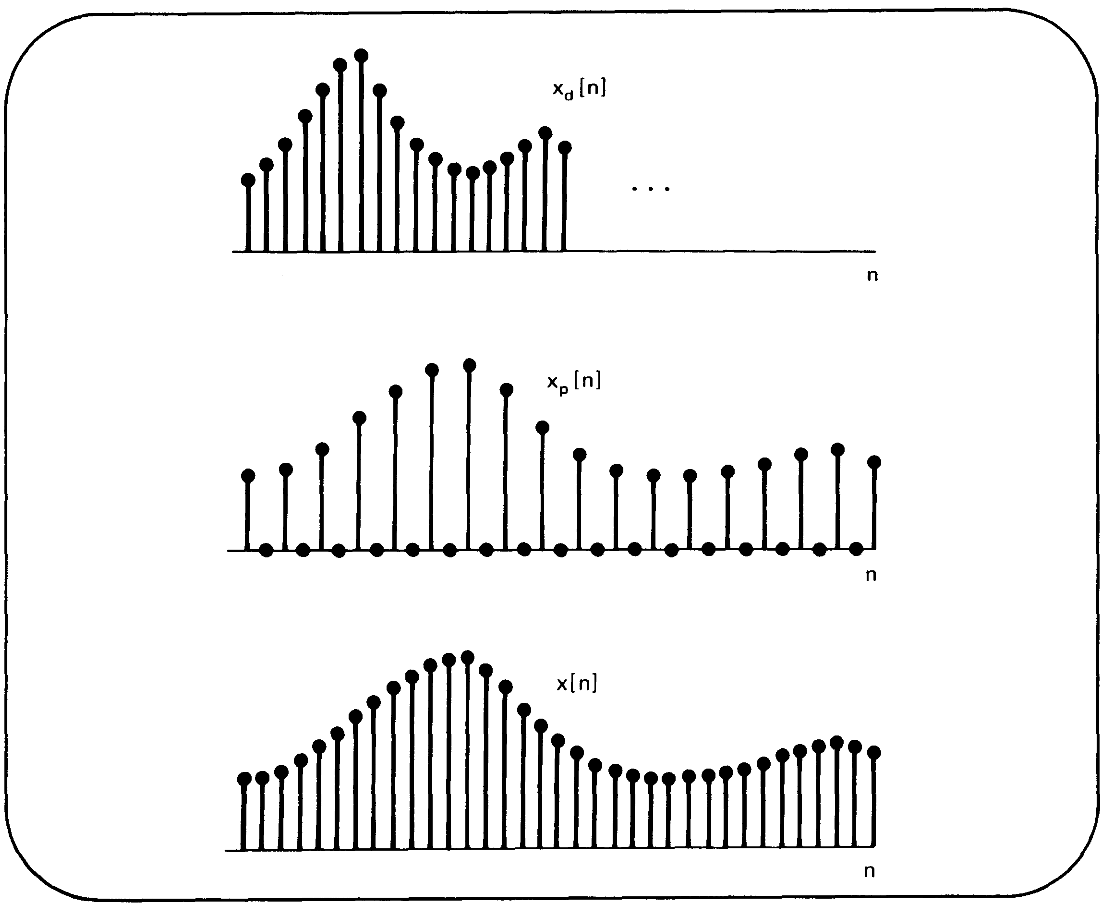
TRANSPARENCY 19.8
Illustration of the effect of decimation in the frequency domain.



TRANSPARENCY 19.9
Example of a context in which discrete-time decimation or downsampling might be used.

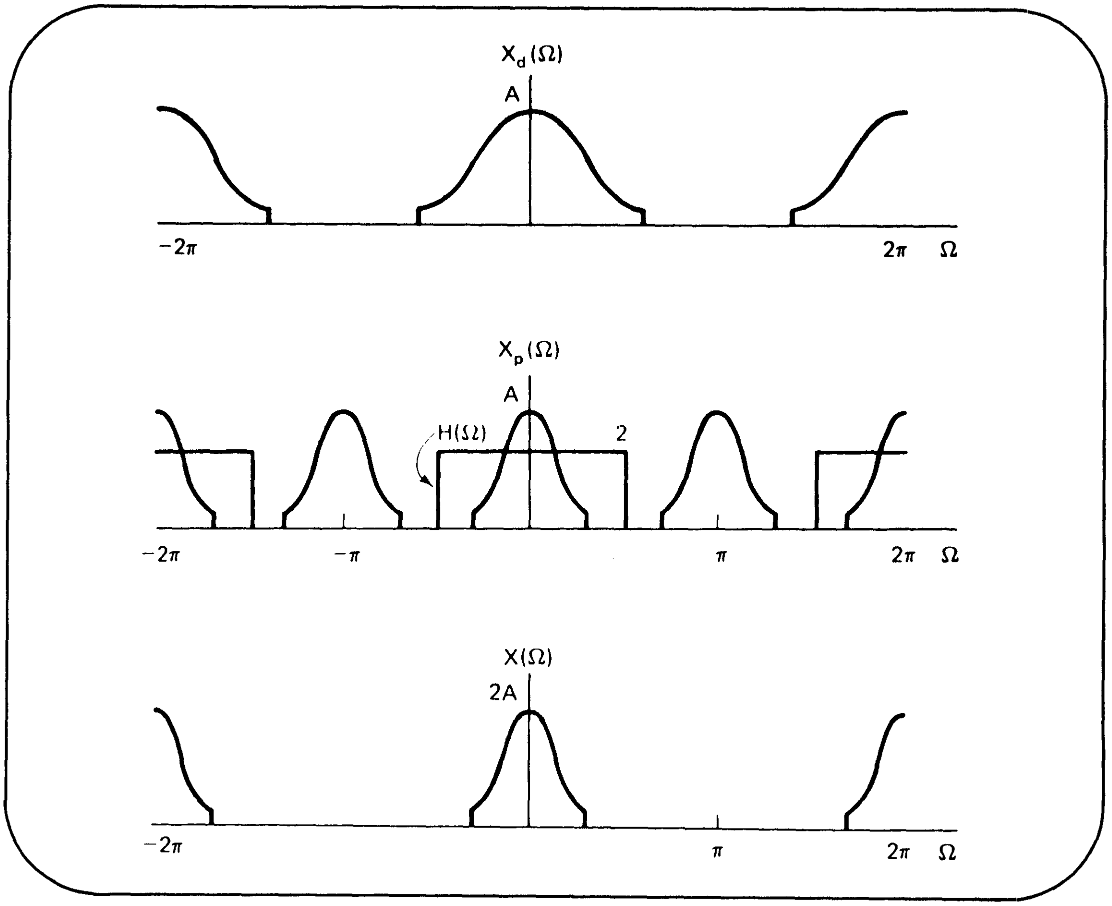


TRANSPARENCY 19.10
Steps involved in upsampling, i.e., recovering a signal after it has been decimated.

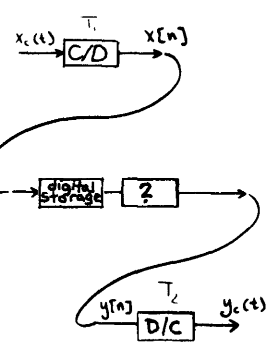


TRANSPARENCY 19.11
Time-domain illustration of upsampling.

TRANSPARENCY 19.12
Frequency-domain illustration of upsampling.



MARKERBOARD 19.2



- $T_2 = 2T_1$

  downsample & decimate 1:2

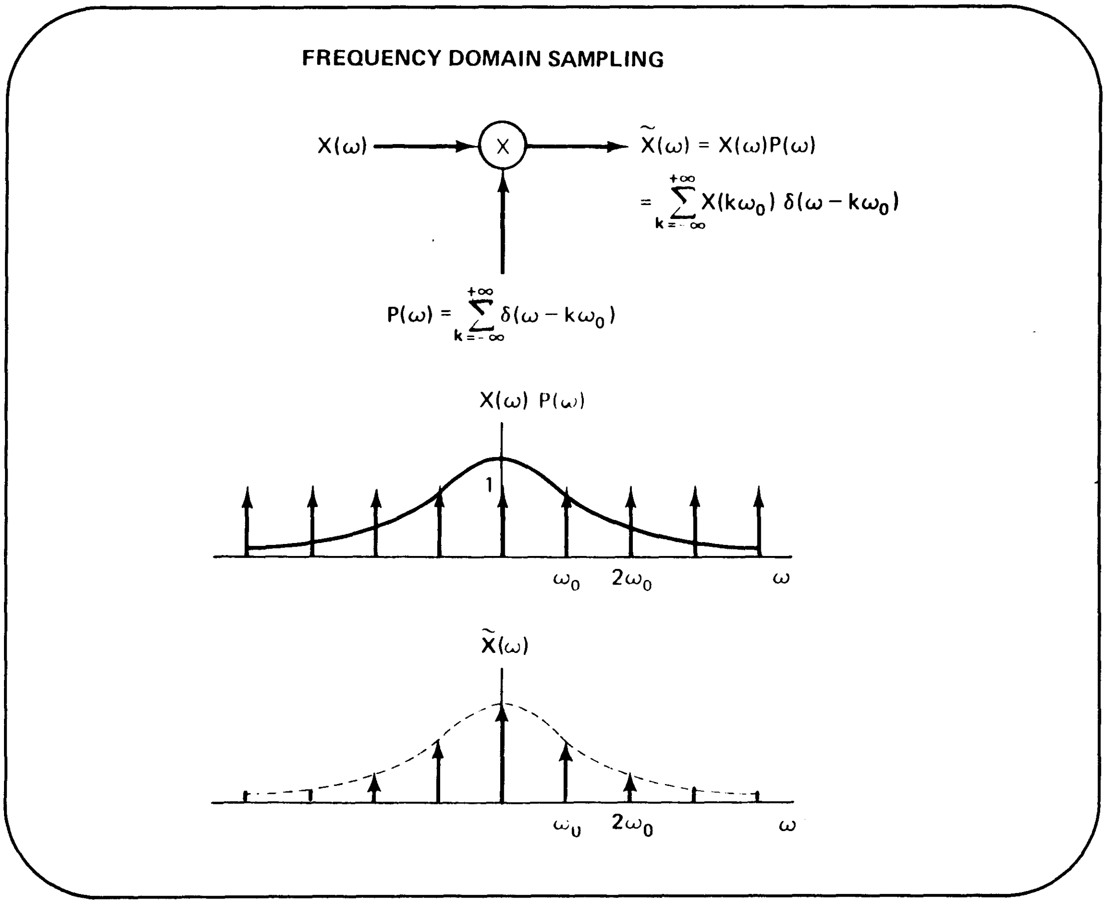
- $T_2 = \frac{1}{2}T_1$

  upsample & interpolate 2:1

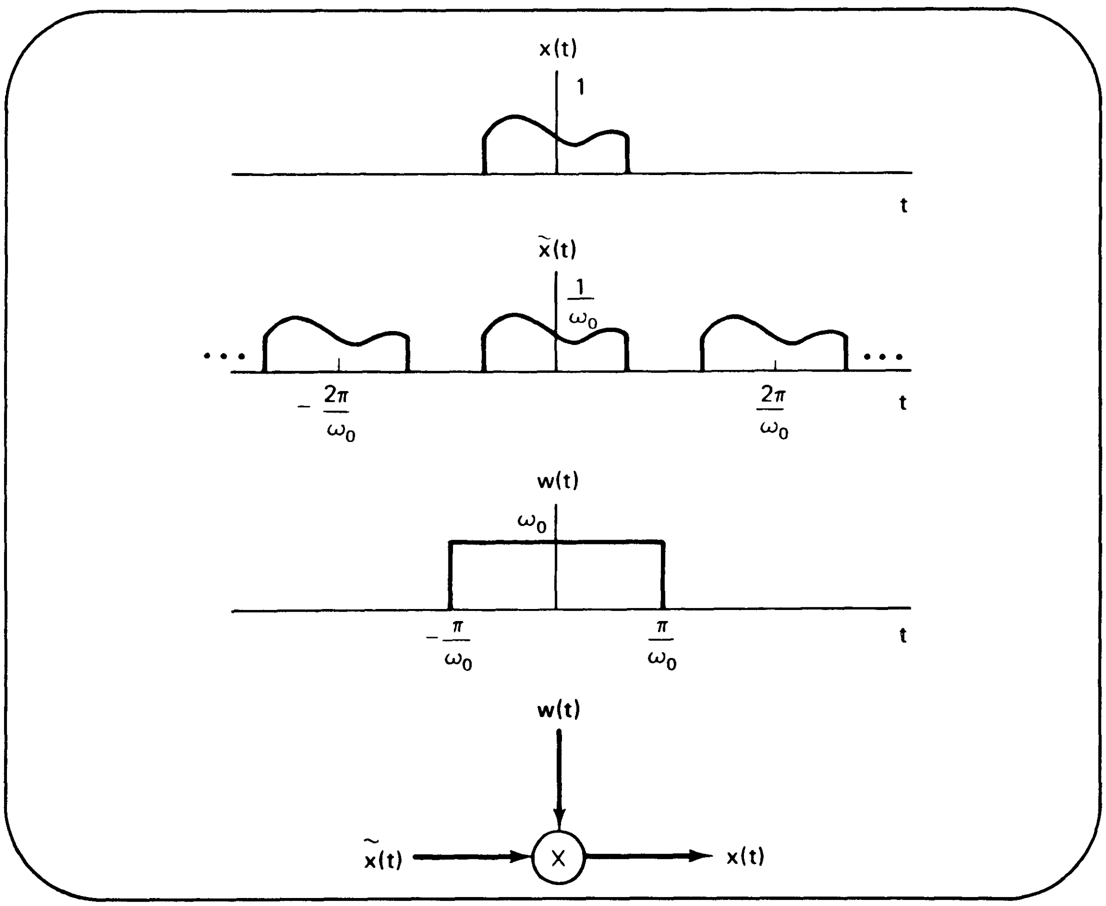
- $T_2 = \frac{3}{2}T_1$

  upsample & interpolate 2:1
  Then
  downsample & decimate 1:3

- $T_2 = \frac{P}{Q}T_1$



TRANSPARENCY 19.13
Frequency-domain sampling.



TRANSPARENCY 19.14
Signal recovery after frequency-domain sampling.

TRANSPARENCY 19.15
Recovery of a discrete-time signal after frequency-domain sampling.