Also, since there is no aliasing for the band-limited input in eq. (7.37), the output of the half-sample delay system is

$$y_c(t) = x_c(t - T/2) = \frac{\sin(\pi(t - T/2)/T)}{\pi(t - T/2)},$$

and the sequence $y_d[n]$ in Figure 7.24 is

$$y_d[n] = y_c(nT) = \frac{\sin(\pi(n - \frac{1}{2}))}{T\pi(n - \frac{1}{2})}.$$

We conclude that

$$h[n] = \frac{\sin(\pi(n - \frac{1}{2}))}{\pi(n - \frac{1}{2})}.$$

7.5 SAMPLING OF DISCRETE-TIME SIGNALS

Thus far in this chapter, we have considered the sampling of continuous-time signals, and in addition to developing the analysis necessary to understand continuous-time sampling, we have introduced a number of its applications. As we will see in this section, a very similar set of properties and results with a number of important applications can be developed for sampling of discrete-time signals.

7.5.1 Impulse-Train Sampling

In analogy with continuous-time sampling as carried out using the system of Figure 7.2, sampling of a discrete-time signal can be represented as shown in Figure 7.31. Here, the new sequence $x_p[n]$ resulting from the sampling process is equal to the original sequence $x[n]$ at integer multiples of the sampling period N and is zero at the intermediate samples; that is,

$$x_p[n] = \begin{cases} x[n], & \text{if } n = \text{an integer multiple of } N \\ 0, & \text{otherwise} \end{cases}. \tag{7.38}$$

As with continuous-time sampling in Section 7.1, the effect in the frequency domain of discrete-time sampling is seen by using the multiplication property developed in Section 5.5. Thus, with

$$x_p[n] = x[n]p[n] = \sum_{k=-\infty}^{+\infty} x[kN]\delta[n - kN], \tag{7.39}$$

we have, in the frequency domain,

$$X_p(e^{j\omega}) = \frac{1}{2\pi}\int_{2\pi} P(e^{j\theta})X(e^{j(\omega-\theta)})d\theta. \tag{7.40}$$

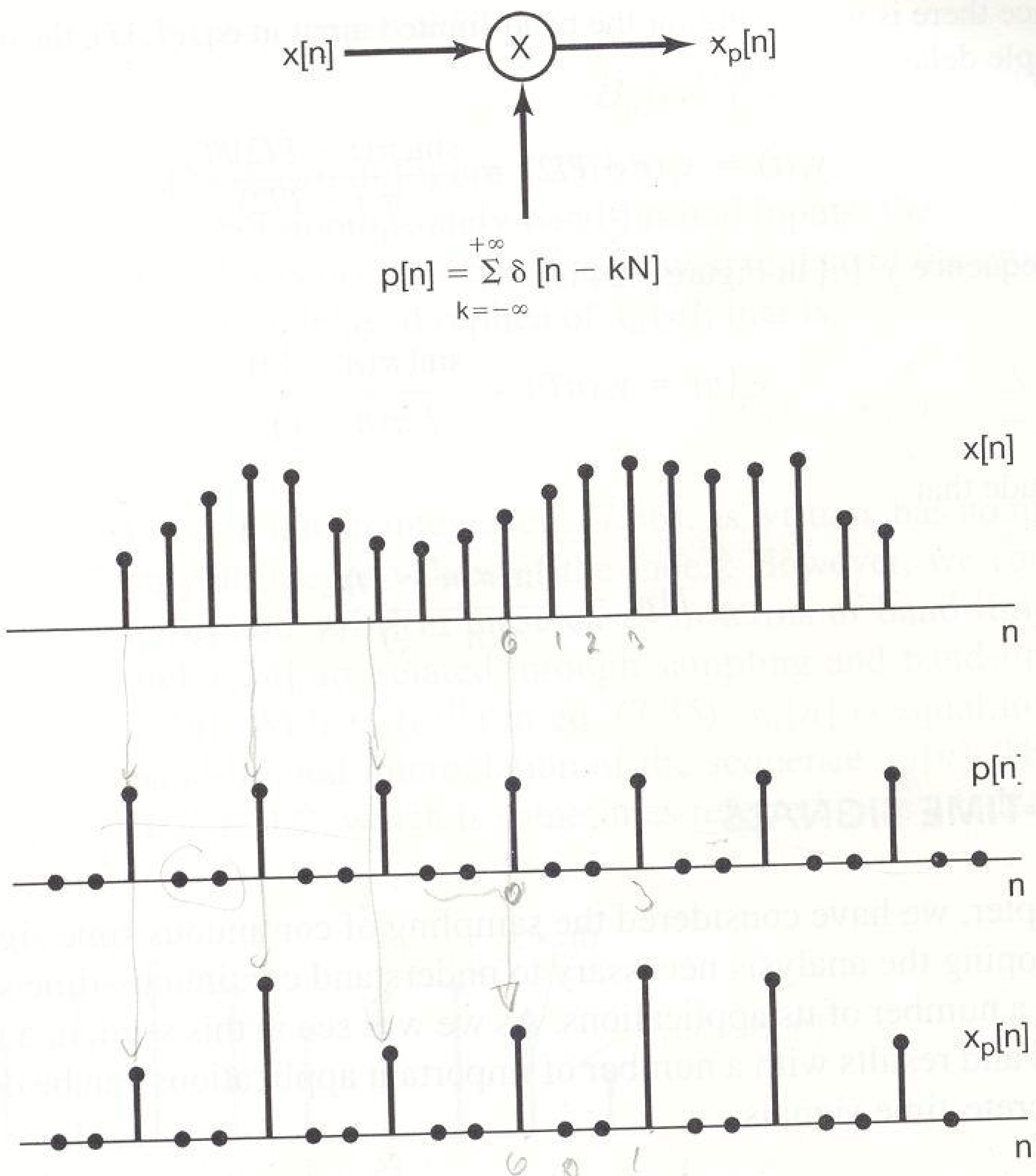

Figure 7.31 Discrete-time sampling.

As in Example 5.6, the Fourier transform of the sampling sequence $p[n]$ is

$$P(e^{j\omega}) = \frac{2\pi}{N}\sum_{k=-\infty}^{+\infty}\delta(\omega - k\omega_s), \tag{7.41}$$

where ω_s, the sampling frequency, equals $2\pi/N$. Combining eqs. (7.40) and (7.41), we have

$$X_p(e^{j\omega}) = \frac{1}{N}\sum_{k=0}^{N-1} X\left(e^{j(\omega - k\omega_s)}\right). \tag{7.42}$$

Equation (7.42) is the counterpart for discrete-time sampling of eq. (7.6) for continuous-time sampling and is illustrated in Figure 7.32. In Figure 7.32(c), with $\omega_s - \omega_M > \omega_M$, or equivalently, $\omega_s > 2\omega_M$, there is no aliasing [i.e., the nonzero portions of the replicas of $X(e^{j\omega})$ do not overlap], whereas with $\omega_s < 2\omega_M$, as in Figure 7.32(d), frequency-domain aliasing results. In the absence of aliasing, $X(e^{j\omega})$ is faithfully reproduced around $\omega = 0$ and integer multiples of 2π. Consequently, $x[n]$ can be recovered from $x_p[n]$ by means of a lowpass filter with gain N and a cutoff frequency greater than

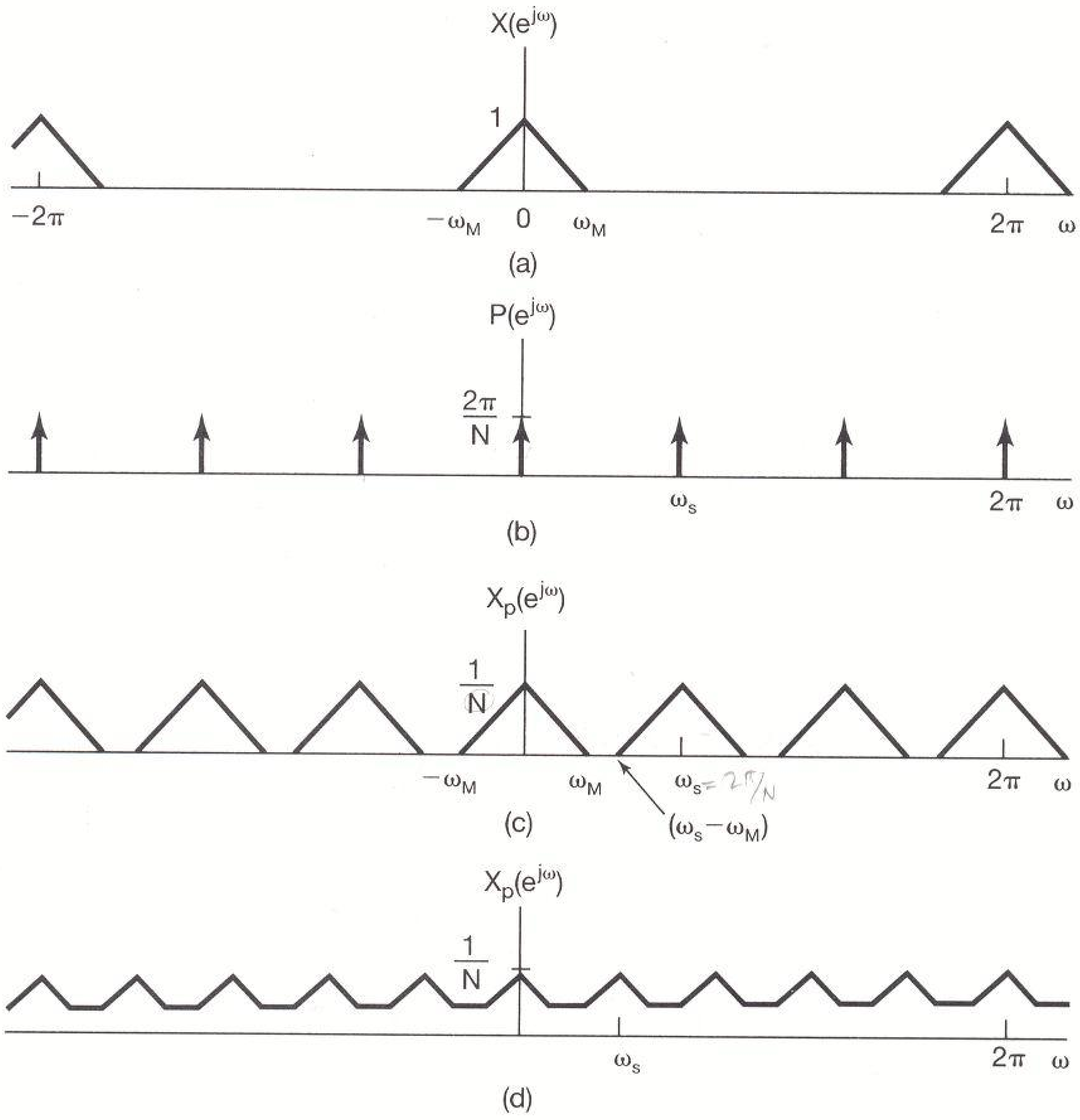

Figure 7.32 Effect in the frequency domain of impulse-train sampling of a discrete-time signal: (a) spectrum of original signal; (b) spectrum of sampling sequence; (c) spectrum of sampled signal with $\omega_s > 2\omega_M$; (d) spectrum of sampled signal with $\omega_s < 2\omega_M$. Note that aliasing occurs.

ω_M and less than $\omega_s - \omega_M$, as illustrated in Figure 7.33, where we have specified the cutoff frequency of the lowpass filter as $\omega_s/2$. If the overall system of Figure 7.33(a) is applied to a sequence for which $\omega_s < 2\omega_M$, so that aliasing results, $x_r[n]$ will no longer be equal to $x[n]$. However, as with continuous-time sampling, the two sequences *will* be equal at multiples of the sampling period; that is, corresponding to eq. (7.13), we have

$$x_r[kN] = x[kN], \quad k = 0, \pm 1, \pm 2, \ldots, \tag{7.43}$$

independently of whether aliasing occurs. (See Problem 7.46.)

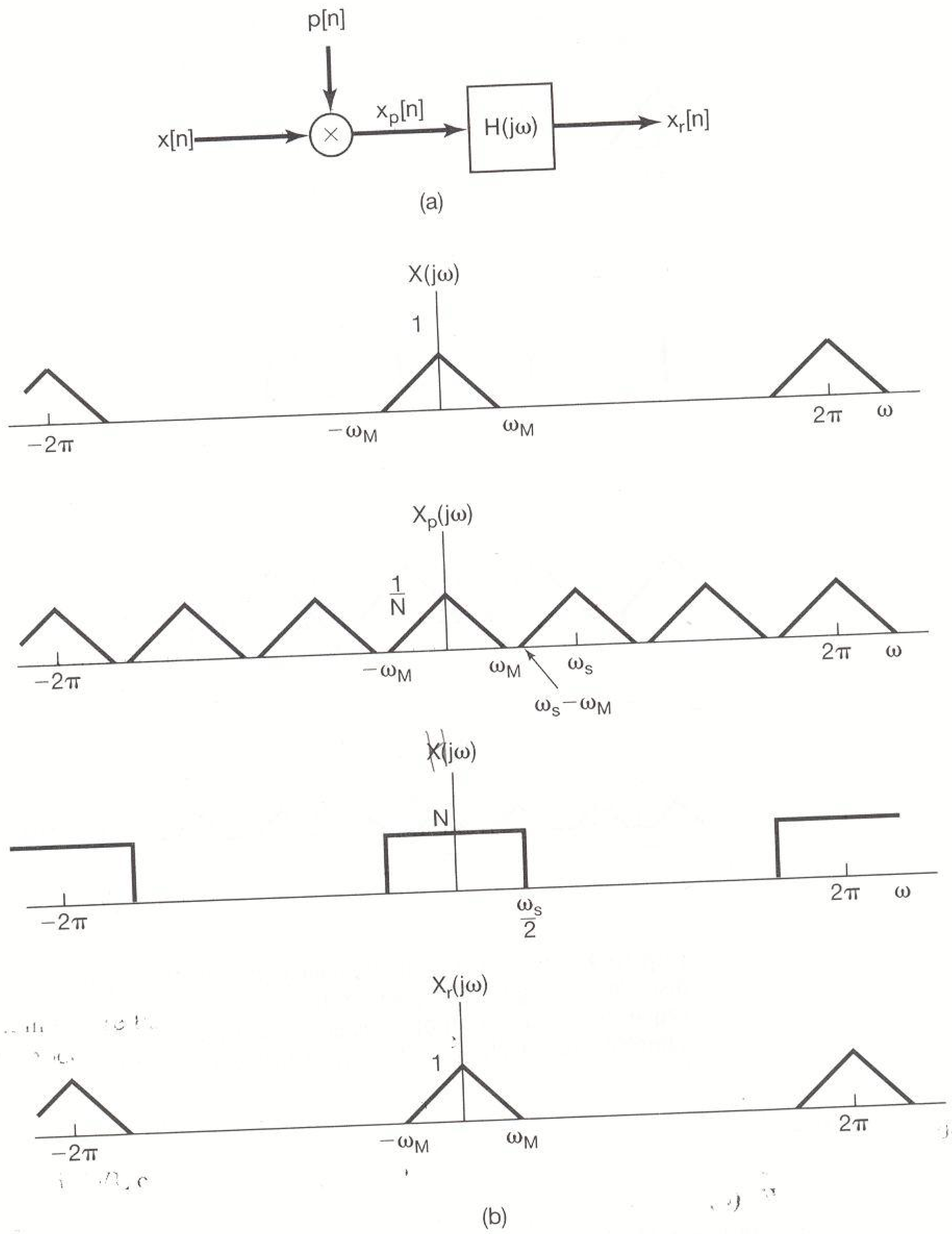

Figure 7.33 Exact recovery of a discrete-time signal from its samples using an ideal lowpass filter: (a) block diagram for sampling and reconstruction of a band-limited signal from its samples; (b) spectrum of the signal x[n]; (c) spectrum of $x_p[n]$; (d) frequency response of an ideal lowpass filter with cutoff frequency $\omega_s/2$; (e) spectrum of the reconstructed signal $x_r[n]$. For the example depicted here $\omega_s > 2\omega_M$ so that no aliasing occurs and consequently $x_r[n] = x[n]$.

Example 7.4

Consider a sequence $x[n]$ whose Fourier transform $X(e^{j\omega})$ has the property that

$$X(e^{j\omega}) = 0 \qquad \text{for} \quad 2\pi/9 \leq |\omega| \leq \pi.$$

To determine the lowest rate at which $x[n]$ may be sampled without the possibility of aliasing, we must find the largest N such that

$$\frac{2\pi}{N} \geq 2\left(\frac{2\pi}{9}\right) \Longrightarrow N \leq 9/2.$$

We conclude that $N_{\max} = 4$, and the corresponding sampling frequency is $2\pi/4 = \pi/2$.

The reconstruction of $x[n]$ through the use of a lowpass filter applied to $x_p[n]$ can be interpreted in the time domain as an interpolation formula similar to eq. (7.11). With $h[n]$ denoting the impulse response of the lowpass filter, we have

$$h[n] = \frac{N\omega_c}{\pi} \frac{\sin \omega_c n}{\omega_c n}. \tag{7.44}$$

The reconstructed sequence is then

$$x_r[n] = x_p[n] * h[n], \tag{7.45}$$

or equivalently,

$$x_r[n] = \sum_{k=-\infty}^{+\infty} x[kN] \frac{N\omega_c}{\pi} \frac{\sin \omega_c(n - kN)}{\omega_c(n - kN)}. \tag{7.46}$$

Equation (7.46) represents ideal band-limited interpolation and requires the implementation of an ideal lowpass filter. In typical applications a suitable approximation for the lowpass filter in Figure 7.33 is used, in which case the equivalent interpolation formula is of the form

$$x_r[n] = \sum_{k=-\infty}^{+\infty} x[kN] h_r[n - kN], \tag{7.47}$$

where $h_r[n]$ is the impulse response of the interpolating filter. Some specific examples, including the discrete-time counterparts of the zero-order hold and first-order hold discussed in Section 7.2 for continuous-time interpolation, are considered in Problem 7.50.

7.5.2 Discrete-Time Decimation and Interpolation

There are a variety of important applications of the principles of discrete-time sampling, such as in filter design and implementation or in communication applications. In many of these applications it is inefficient to represent, transmit, or store the sampled sequence $x_p[n]$ directly in the form depicted in Figure 7.31, since, in between the sampling instants, $x_p[n]$ is *known* to be zero. Thus, the sampled sequence is typically replaced by a new sequence $x_b[n]$, which is simply every Nth value of $x_p[n]$; that is,

$$x_b[n] = x_p[nN]. \tag{7.48}$$

Also, equivalently,

$$x_b[n] = x[nN], \tag{7.49}$$

since $x_p[n]$ and $x[n]$ are equal at integer multiples of N. The operation of extracting every Nth sample is commonly referred to as *decimation.*[3] The relationship between $x[n]$, $x_p[n]$, and $x_b[n]$ is illustrated in Figure 7.34.

To determine the effect in the frequency domain of decimation, we wish to determine the relationship between $X_b(e^{j\omega})$—the Fourier transform of $x_b[n]$—and $X(e^{j\omega})$. To this end, we note that

$$X_b(e^{j\omega}) = \sum_{k=-\infty}^{+\infty} x_b[k]e^{-j\omega k}, \tag{7.50}$$

or, using eq. (7.48),

$$X_b(e^{j\omega}) = \sum_{k=-\infty}^{+\infty} x_p[kN]e^{-j\omega k}. \tag{7.51}$$

If we let $n = kN$, or equivalently $k = n/N$, we can write

$$X_b(e^{j\omega}) = \sum_{\substack{n=\text{integer}\\ \text{multiple of } N}} x_p[n]e^{-j\omega n/N},$$

and since $x_p[n] = 0$ when n is not an integer multiple of N, we can also write

$$X_b(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x_p[n]e^{-j\omega n/N}. \tag{7.52}$$

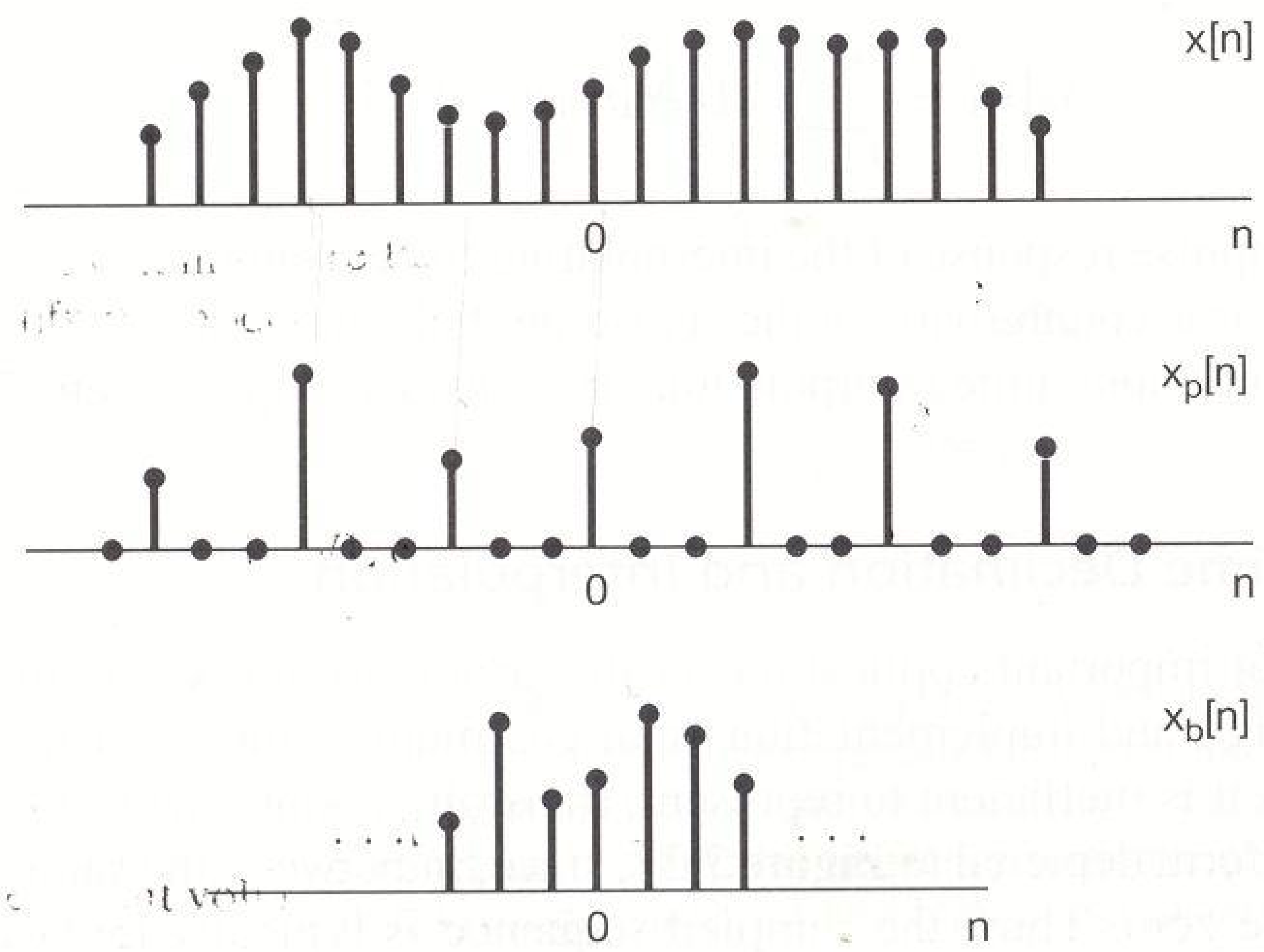

Figure 7.34 Relationship between $x_p[n]$ corresponding to sampling and $x_b[n]$ corresponding to decimation.

[3]Technically, decimation would correspond to extracting every *tenth* sample. However, it has become common terminology to refer to the operation as decimation even when N is not equal to 10.

Furthermore, we recognize the right-hand side of eq. (7.52) as the Fourier transform of $x_p[n]$; that is,

$$\sum_{n=-\infty}^{+\infty} x_p[n]e^{-j\omega n/N} = X_p(e^{j\omega/N}). \tag{7.53}$$

Thus, from eqs. (7.52) and (7.53), we conclude that

$$X_b(e^{j\omega}) = X_p(e^{j\omega/N}). \tag{7.54}$$

This relationship is illustrated in Figure 7.35, and from it, we observe that the spectra for the sampled sequence and the decimated sequence differ only in a frequency scaling or normalization. If the original spectrum $X(e^{j\omega})$ is appropriately band limited, so that there is no aliasing present in $X_p(e^{j\omega})$, then, as shown in the figure, the effect of decimation is to spread the spectrum of the original sequence over a larger portion of the frequency band.

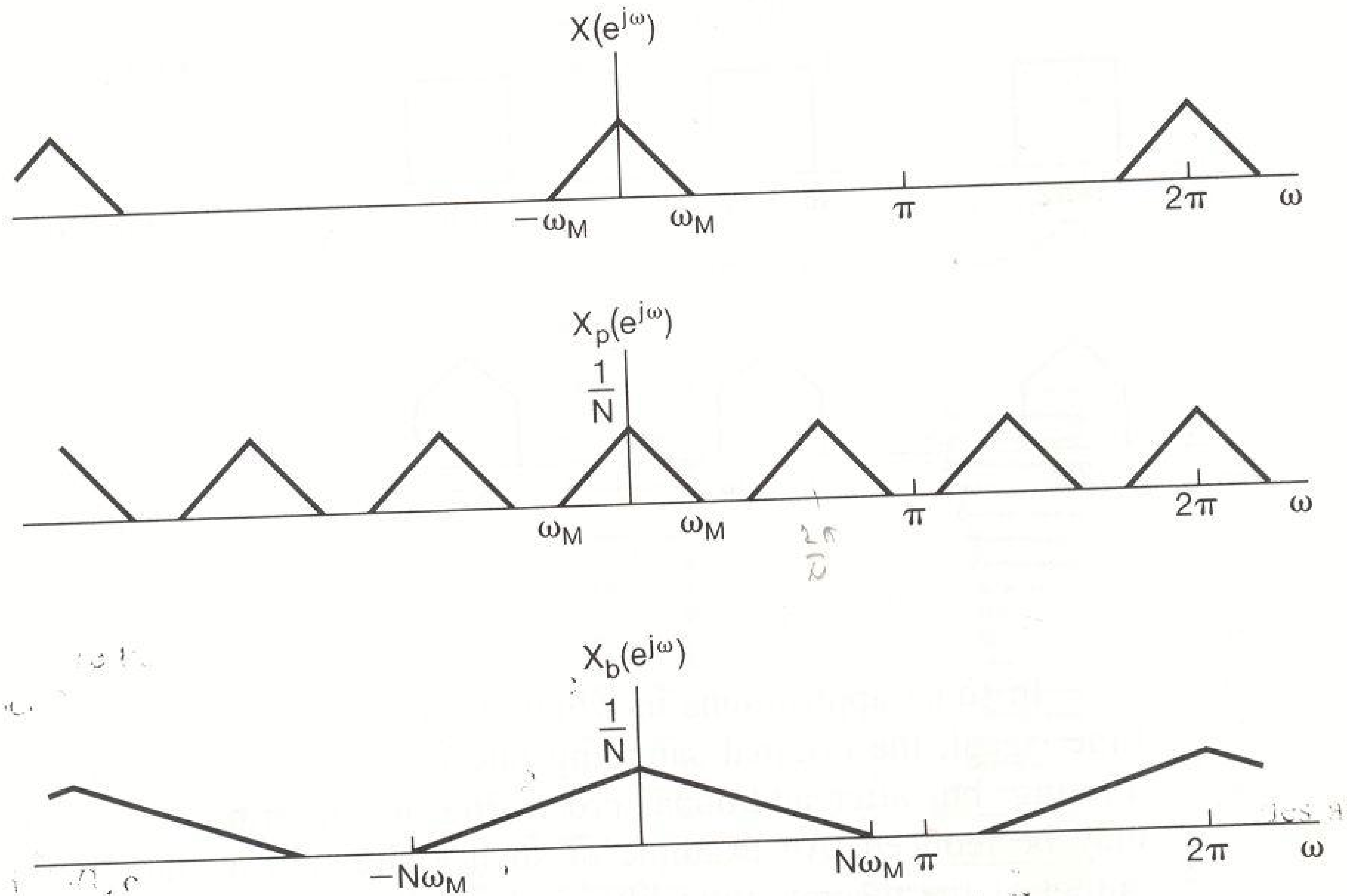

Figure 7.35 Frequency-domain illustration of the relationship between sampling and decimation.

If the original sequence $x[n]$ is obtained by sampling a continuous-time signal, the process of decimation can be viewed as reducing the sampling rate on the signal by a factor of N. To avoid aliasing, $X(e^{j\omega})$ cannot occupy the full frequency band. In other words, if the signal can be decimated without introducing aliasing, then the original continuous-time signal was oversampled, and thus, the sampling rate can be reduced without aliasing. With the interpretation of the sequence $x[n]$ as samples of a continuous-time signal, the process of decimation is often referred to as *downsampling*.

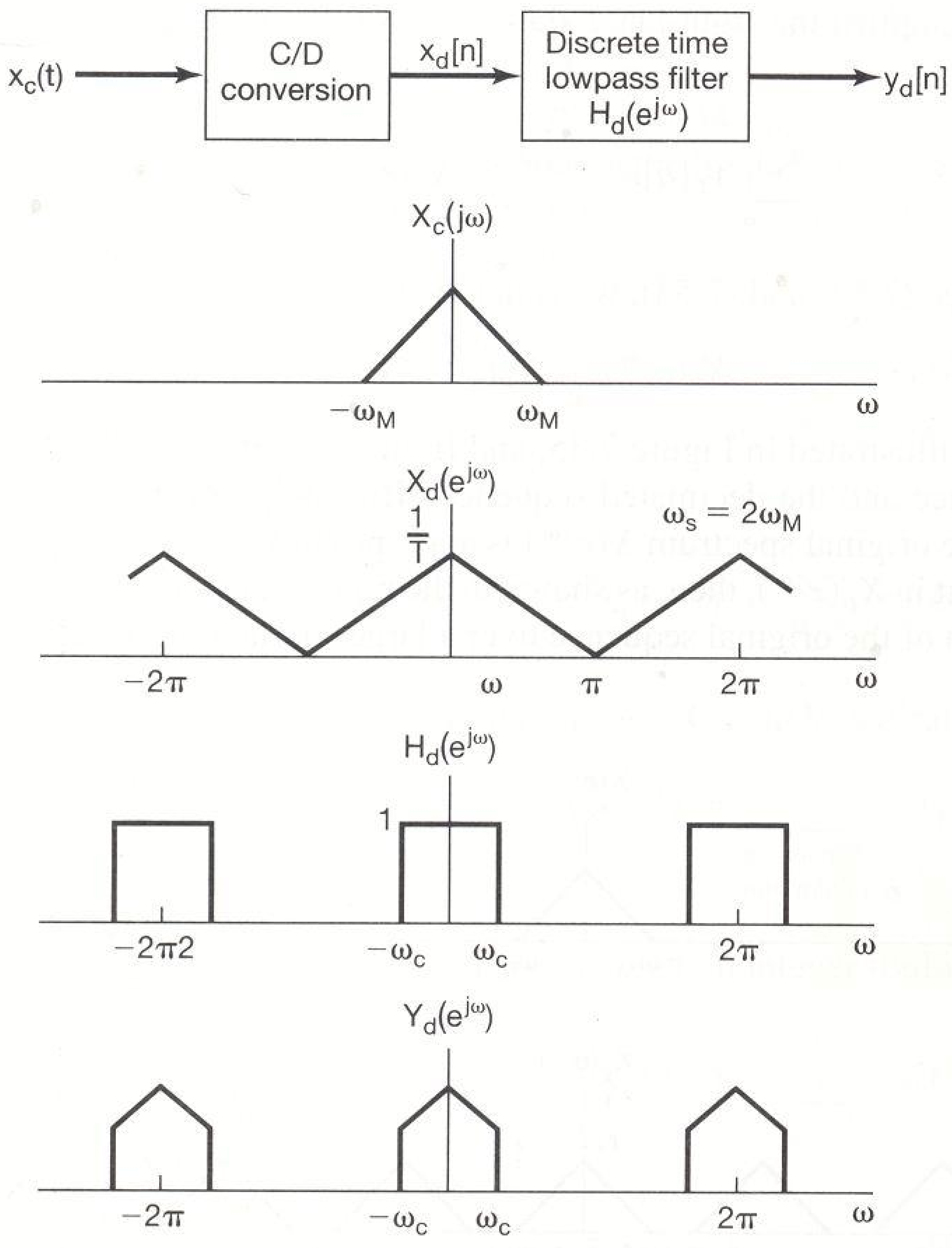

Figure 7.36 Continuous-time signal that was originally sampled at the Nyquist rate. After discrete-time filtering, the resulting sequence can be further downsampled. Here $X_c(j\omega)$ is the continuous-time Fourier transform of $x_c(t)$, $X_d(e^{j\omega})$ and $Y_d(e^{j\omega})$ are the discrete-time Fourier transforms of $x_d[n]$ and $y_d[n]$ respectively, and $H_d(e^{j\omega})$ is the frequency response of the discrete-time lowpass filter depicted in the block diagram.

In some applications in which a sequence is obtained by sampling a continuous-time signal, the original sampling rate may be as low as possible without introducing aliasing, but after additional processing and filtering, the bandwidth of the sequence may be reduced. An example of such a situation is shown in Figure 7.36. Since the output of the discrete-time filter is band limited, downsampling, or decimation can be applied.

Just as in some applications it is useful to downsample, there are situations in which it is useful to convert a sequence to a *higher* equivalent sampling rate, a process referred to as *upsampling* or *interpolation.* Upsampling is basically the reverse of decimation or downsampling. As illustrated in Figures 7.34 and 7.35, in decimation we first sample and then retain only the sequence values at the sampling instants. To upsample, we reverse the process. For example, referring to Figure 7.34, we consider upsampling the sequence $x_b[n]$ to obtain $x[n]$. From $x_b[n]$, we form the sequence $x_p[n]$ by inserting $N-1$ points with zero amplitude between each of the values in $x_b[n]$. The interpolated sequence $x[n]$ is then obtained from $x_p[n]$ by lowpass filtering. The overall procedure is summarized in Figure 7.37.

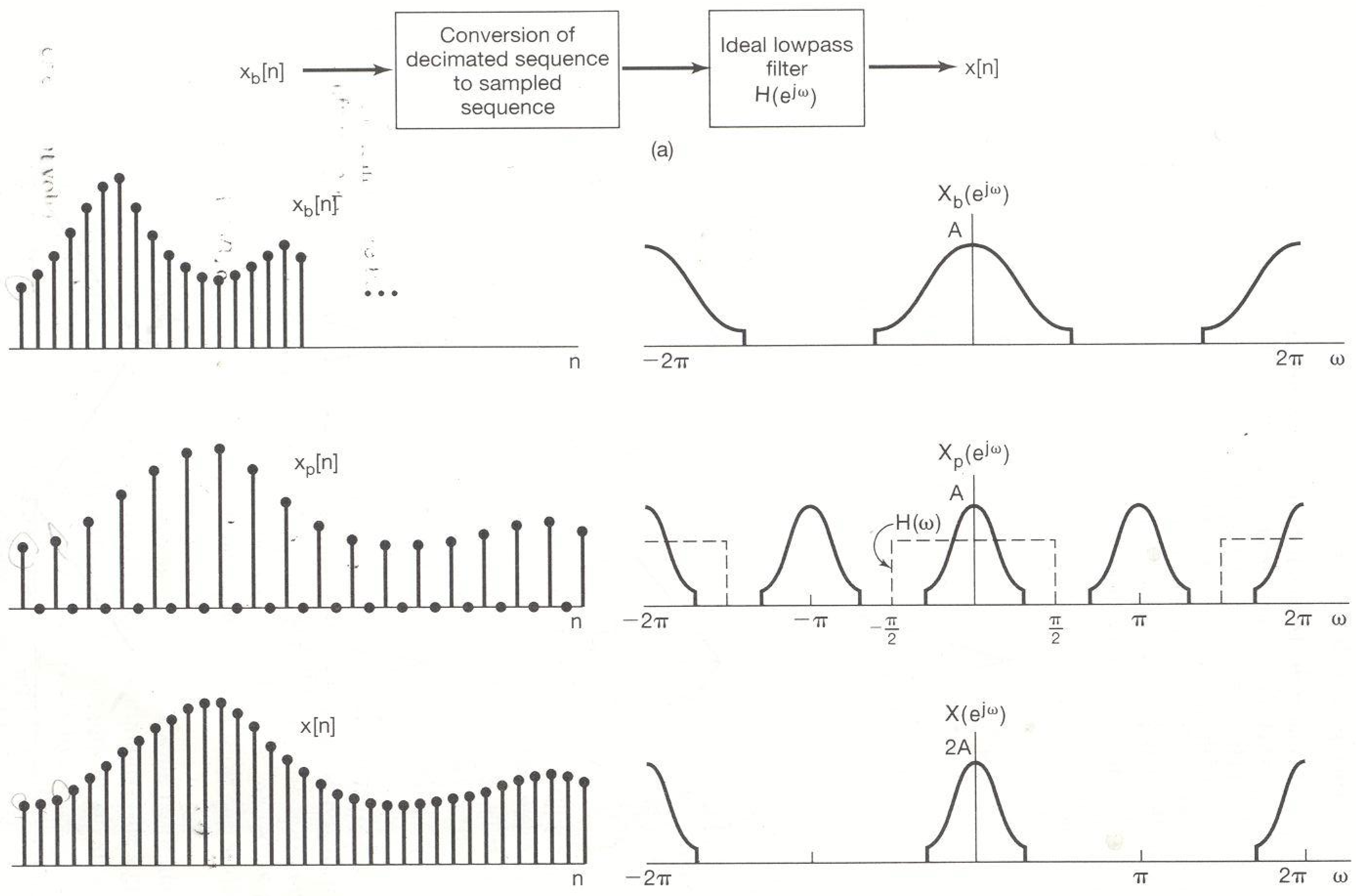

Figure 7.37 Upsampling: (a) overall system; (b) associated sequences and spectra for upsampling by a factor of 2.

Example 7.5

In this example, we illustrate how a combination of interpolation and decimation may be used to further downsample a sequence without incurring aliasing. It should be noted that maximum possible downsampling is achieved once the non-zero portion of one period of the discrete-time spectrum has expanded to fill the entire band from $-\pi$ to π.

Consider the sequence $x[n]$ whose Fourier transform $X(e^{j\omega})$ is illustrated in Figure 7.38(a). As discussed in Example 7.4, the lowest rate at which impulse-train sampling may be used on this sequence without incurring aliasing is $2\pi/4$. This corresponds to

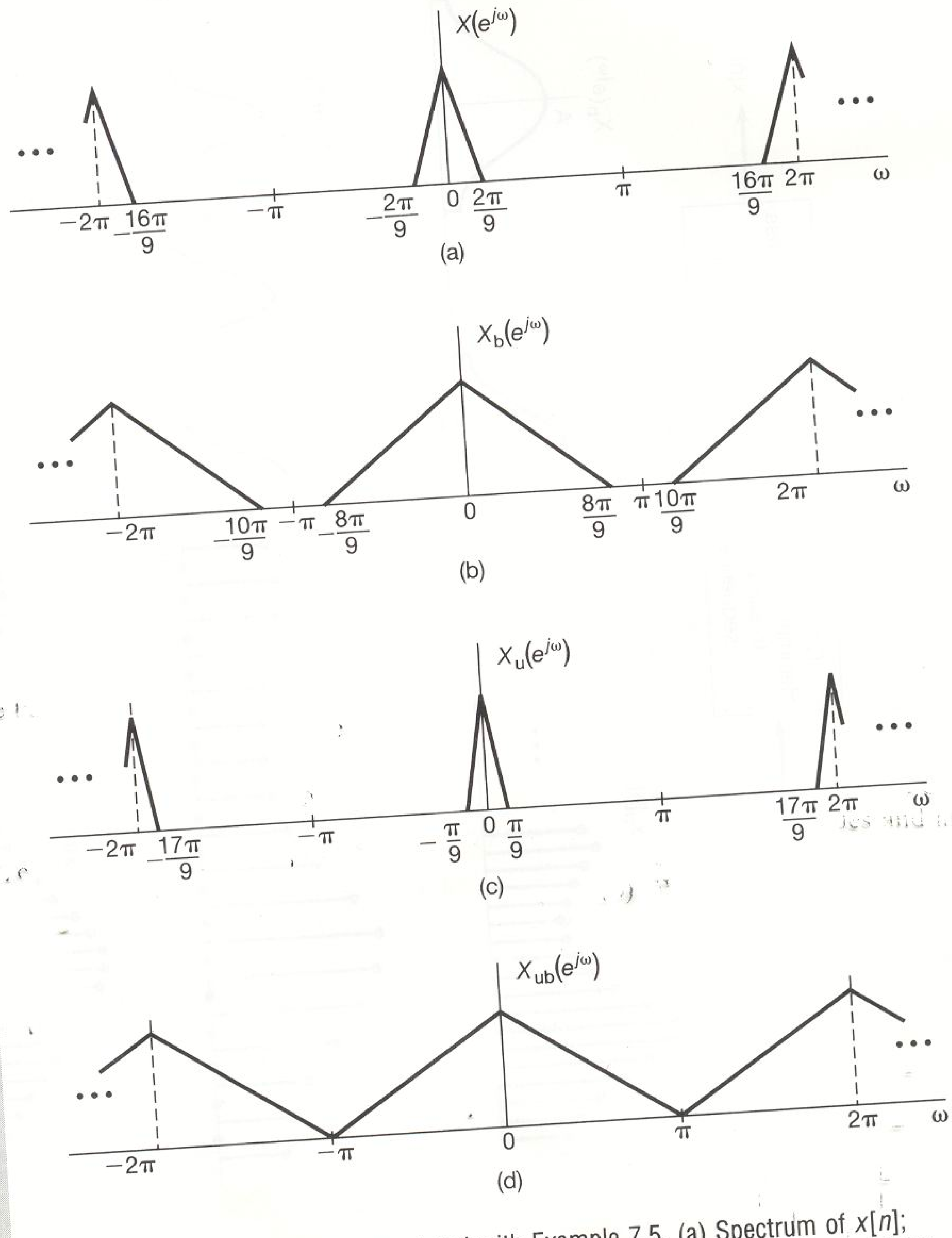

Figure 7.38 Spectra associated with Example 7.5. (a) Spectrum of $x[n]$; (b) spectrum after downsampling by 4; (c) spectrum after upsampling $x[n]$ by a factor of 2; (d) spectrum after upsampling $x[n]$ by 2 and then downsampling by 9.

sampling every 4th value of $x[n]$. If the result of such sampling is decimated by a factor of 4, we obtain a sequence $x_b[n]$ whose spectrum is shown in Figure 7.38(b). Clearly, there is still no aliasing of the original spectrum. However, this spectrum is zero for $8\pi/9 \le |\omega| \le \pi$, which suggests there is room for further downsampling.

Specifically, examining Figure 7.38(a) we see that if we could scale frequency by a factor of 9/2, the resulting spectrum would have nonzero values over the entire frequency interval from $-\pi$ to π. However, since 9/2 is not an integer, we can't achieve this purely by downsampling. Rather we must first upsample $x[n]$ by a factor of 2 and then downsample by a factor of 9. In particular, the spectrum of the signal $x_u[n]$ obtained when $x[n]$ is upsampled by a factor of 2, is displayed in Figure 7.38(c). When $x_u[n]$ is then downsampled by a factor of 9, the spectrum of the resulting sequence $x_{ub}[n]$ is as shown in Figure 7.38(d). This combined result effectively corresponds to downsampling $x[n]$ by a noninteger amount, 9/2. Assuming that $x[n]$ represents unaliased samples of a continuous-time signal $x_c(t)$, our interpolated and decimated sequence represents the maximum possible (aliasing-free) downsampling of $x_c(t)$.

7.6 SUMMARY

In this chapter we have developed the concept of sampling, whereby a continuous-time or discrete-time signal is represented by a sequence of equally spaced samples. The conditions under which the signal is exactly recoverable from the samples is embodied in the sampling theorem. For exact reconstruction, this theorem requires that the signal to be sampled be band limited and that the sampling frequency be greater than twice the highest frequency in the signal to be sampled. Under these conditions, exact reconstruction of the original signal is carried out by means of ideal lowpass filtering. The time-domain interpretation of this ideal reconstruction procedure is often referred to as ideal band-limited interpolation. In practical implementations, the lowpass filter is approximated and the interpolation in the time domain is no longer exact. In some instances, simple interpolation procedures such as a zero-order hold or linear interpolation (a first-order hold) suffice.

If a signal is undersampled (i.e., if the sampling frequency is less than that required by the sampling theorem), then the signal reconstructed by ideal band-limited interpolation will be related to the original signal through a form of distortion referred to as aliasing. In many instances, it is important to choose the sampling rate so as to avoid aliasing. However, there are a variety of important examples, such as the stroboscope, in which aliasing is exploited.

Sampling has a number of important applications. One particularly significant set of applications relates to using sampling to process continuous-time signals with discrete-time systems, by means of minicomputers, microprocessors, or any of a variety of devices specifically oriented toward discrete-time signal processing.

The basic theory of sampling is similar for both continuous-time and discrete-time signals. In the discrete-time case there is the closely related concept of decimation, whereby the decimated sequence is obtained by extracting values of the original sequence at equally spaced intervals. The difference between sampling and decimation lies in the fact that, for the sampled sequence, values of zero lie in between the sample values, whereas in the decimated sequence these zero values are discarded, thereby compressing the sequence in time. The inverse of decimation is interpolation. The ideas of decima-