be interpolated by the ideal D/C converter for both the input and the output sequence. Note in Figure 4.19(b) that the six-point moving-average filtering gives a sampled cosine signal such that the sample points have been shifted by 2.5 samples with respect to the sample points of the input. This can be seen from Figure 4.19(b) by comparing the positive peak at 10 in the interpolated cosine for the input to the positive peak at 12.5 in the interpolated cosine for the output. Thus, the six-point moving-average filter is seen to have a delay of $5/2 = 2.5$ samples.

4.6 CHANGING THE SAMPLING RATE USING DISCRETE-TIME PROCESSING

We have seen that a continuous-time signal $x_c(t)$ can be represented by a discrete-time signal consisting of a sequence of samples

$$x[n] = x_c(nT). \tag{4.69}$$

Alternatively, our previous discussion has shown that, even if $x[n]$ was not obtained originally by sampling, we can always use the bandlimited interpolation formula of Eq. (4.25) to find a continuous-time bandlimited signal $x_r(t)$ whose samples are $x[n] = x_c(nT)$.

It is often necessary to change the sampling rate of a discrete-time signal, i.e., to obtain a new discrete-time representation of the underlying continuous-time signal of the form

$$x'[n] = x_c(nT'), \tag{4.70}$$

where $T' \neq T$. One approach to obtaining the sequences $x'[n]$ from $x[n]$ is to reconstruct $x_c(t)$ from $x[n]$ using Eq. (4.25) and then resample $x_c(t)$ with period T' to obtain $x'[n]$. Often, however, this is not a desirable approach, because of the nonideal analog reconstruction filter, D/A converter, and A/D converter that would be used in a practical implementation. Thus, it is of interest to consider methods of changing the sampling rate that involve only discrete-time operations.

4.6.1 Sampling Rate Reduction by an Integer Factor

The sampling rate of a sequence can be reduced by "sampling" it, i.e., by defining a new sequence

$$x_d[n] = x[nM] = x_c(nMT). \tag{4.71}$$

Equation (4.71) defines the system depicted in Figure 4.20, which is called a *sampling rate compressor* (see Crochiere and Rabiner, 1983) or simply a *compressor*. From Eq. (4.71), it is clear that $x_d[n]$ is identical to the sequence that would be obtained from $x_c(t)$ by

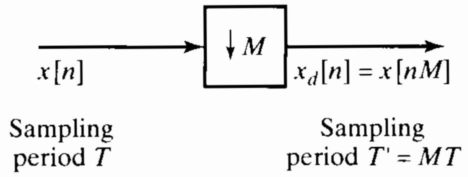

Figure 4.20 Representation of a compressor or discrete-time sampler.

sampling with period $T' = MT$. Furthermore, if $X_c(j\Omega) = 0$ for $|\Omega| \geq \Omega_N$, then $x_d[n]$ is an exact representation of $x_c(t)$ if $\pi/T' = \pi/(MT) \geq \Omega_N$. That is, the sampling rate can be reduced by a factor of M without aliasing if the original sampling rate was at least M times the Nyquist rate or if the bandwidth of the sequence is first reduced by a factor of M by discrete-time filtering. In general, the operation of reducing the sampling rate (including any prefiltering) will be called *downsampling*.

As in the case of sampling a continuous-time signal, it is useful to obtain a frequency-domain relation between the input and output of the compressor. This time, however, it will be a relationship between discrete-time Fourier transforms. Although several methods can be used to derive the desired result, we will base our derivation on the results already obtained for sampling continuous-time signals. First recall that the discrete-time Fourier transform of $x[n] = x_c(nT)$ is

$$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right). \tag{4.72}$$

Similarly, the discrete-time Fourier transform of $x_d[n] = x[nM] = x_c(nT')$ with $T' = MT$ is

$$X_d(e^{j\omega}) = \frac{1}{T'}\sum_{r=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T'} - \frac{2\pi r}{T'}\right)\right). \tag{4.73}$$

Now, since $T' = MT$, we can write Eq. (4.73) as

$$X_d(e^{j\omega}) = \frac{1}{MT}\sum_{r=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi r}{MT}\right)\right). \tag{4.74}$$

To see the relationship between Eqs. (4.74) and (4.72), note that the summation index r in Eq. (4.74) can be expressed as

$$r = i + kM, \tag{4.75}$$

where k and i are integers such that $-\infty < k < \infty$ and $0 \leq i \leq M-1$. Clearly, r is still an integer ranging from $-\infty$ to ∞, but now Eq. (4.74) can be expressed as

$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1}\left[\frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi k}{T} - \frac{2\pi i}{MT}\right)\right)\right]. \tag{4.76}$$

The term inside the square brackets in Eq. (4.76) is recognized from Eq. (4.72) as

$$X(e^{j(\omega - 2\pi i)/M}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega - 2\pi i}{MT} - \frac{2\pi k}{T}\right)\right). \tag{4.77}$$

Thus, we can express Eq. (4.76) as

$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X(e^{j(\omega/M - 2\pi i/M)}). \tag{4.78}$$

There is a strong analogy between Eqs. (4.72) and (4.78): Equation (4.72) expresses the Fourier transform of the sequence of samples, $x[n]$ (with period T), in terms of the Fourier transform of the continuous-time signal $x_c(t)$; Equation (4.78) expresses the Fourier transform of the discrete-time sampled sequence $x_d[n]$ (with sampling period M) in terms of the Fourier transform of the sequence $x[n]$. If we compare Eqs. (4.73) and (4.78), we see that $X_d(e^{j\omega})$ can be thought of as being composed of either an infinite set of copies of $X_c(j\Omega)$, frequency scaled through $\omega = \Omega T'$ and shifted by integer multiples of $2\pi/T'$ (Eq. (4.73)), or M copies of the periodic Fourier transform $X(e^{j\omega})$, frequency scaled by M and shifted by integer multiples of 2π (Eq. (4.78)). Either interpretation makes it clear that $X_d(e^{j\omega})$ is periodic with period 2π (as are all discrete-time Fourier transforms) and that aliasing can be avoided by ensuring that $X(e^{j\omega})$ is bandlimited, i.e.,

$$X(e^{j\omega}) = 0, \qquad \omega_N \le |\omega| \le \pi, \tag{4.79}$$

and $2\pi/M \ge 2\omega_N$.

Downsampling is illustrated in Figure 4.21. Figure 4.21(a) shows the Fourier transform of a bandlimited continuous-time signal, and Figure 4.21(b) shows the Fourier transform of the impulse train of samples when the sampling period is T. Figure 4.21(c) shows $X(e^{j\omega})$ and is related to Figure 4.21(b) through Eq. (4.18). As we have already seen, Figures 4.21(b) and (c) differ only in a scaling of the frequency variable. Figure 4.21(d) shows the discrete-time Fourier transform of the downsampled sequence when $M = 2$. We have plotted this Fourier transform as a function of the normalized frequency $\omega = \Omega T'$. Finally, Figure 4.21(e) shows the discrete-time Fourier transform of the downsampled sequence plotted as a function of the continuous-time frequency variable Ω. Figure 4.21(e) is identical to Figure 4.21(d), except for the scaling of the frequency axis through the relation $\Omega = \omega/T'$.

In this example, $2\pi/T = 4\Omega_N$; i.e., the original sampling rate is exactly twice the minimum rate to avoid aliasing. Thus, when the original sampled sequence is downsampled by a factor of $M = 2$, no aliasing results. If the downsampling factor is more than 2 in this case, aliasing will result, as illustrated in Figure 4.22.

Figure 4.22(a) shows the continuous-time Fourier transform of $x_c(t)$, and Figure 4.22(b) shows the discrete-time Fourier transform of the sequence $x[n] = x_c(nT)$, when $2\pi/T = 4\Omega_N$. Thus, $\omega_N = \Omega_N T = \pi/2$. Now, if we downsample by a factor of $M = 3$, we obtain the sequence $x_d[n] = x[3n] = x_c(n3T)$ whose discrete-time Fourier transform is plotted in Figure 4.22(c) with normalized frequency $\omega = \Omega T'$. Note that because $M\omega_N = 3\pi/2$, which is greater than π, aliasing occurs. In general, to avoid aliasing in downsampling by a factor of M requires that

$$\omega_N M < \pi \qquad \text{or} \qquad \omega_N < \pi/M. \tag{4.80}$$

If this condition does not hold, aliasing occurs, but it may be tolerable for some applications. In other cases, downsampling can be done without aliasing if we are willing to reduce the bandwidth of the signal $x[n]$ before downsampling. Thus, if $x[n]$ is filtered by an ideal lowpass filter with cutoff frequency π/M, then the output $\tilde{x}[n]$ can be downsampled without aliasing, as illustrated in Figures 4.22(d), (e), and (f). Note that the sequence $\tilde{x}_d[n] = \tilde{x}[nM]$ no longer represents the original underlying continuous-time signal $x_c(t)$. Rather, $\tilde{x}_d[n] = \tilde{x}_c(nT')$, where $T' = MT$, and $\tilde{x}_c(t)$ is obtained from $x_c(t)$ by lowpass filtering with cutoff frequency $\Omega_c = \pi/T' = \pi/(MT)$.

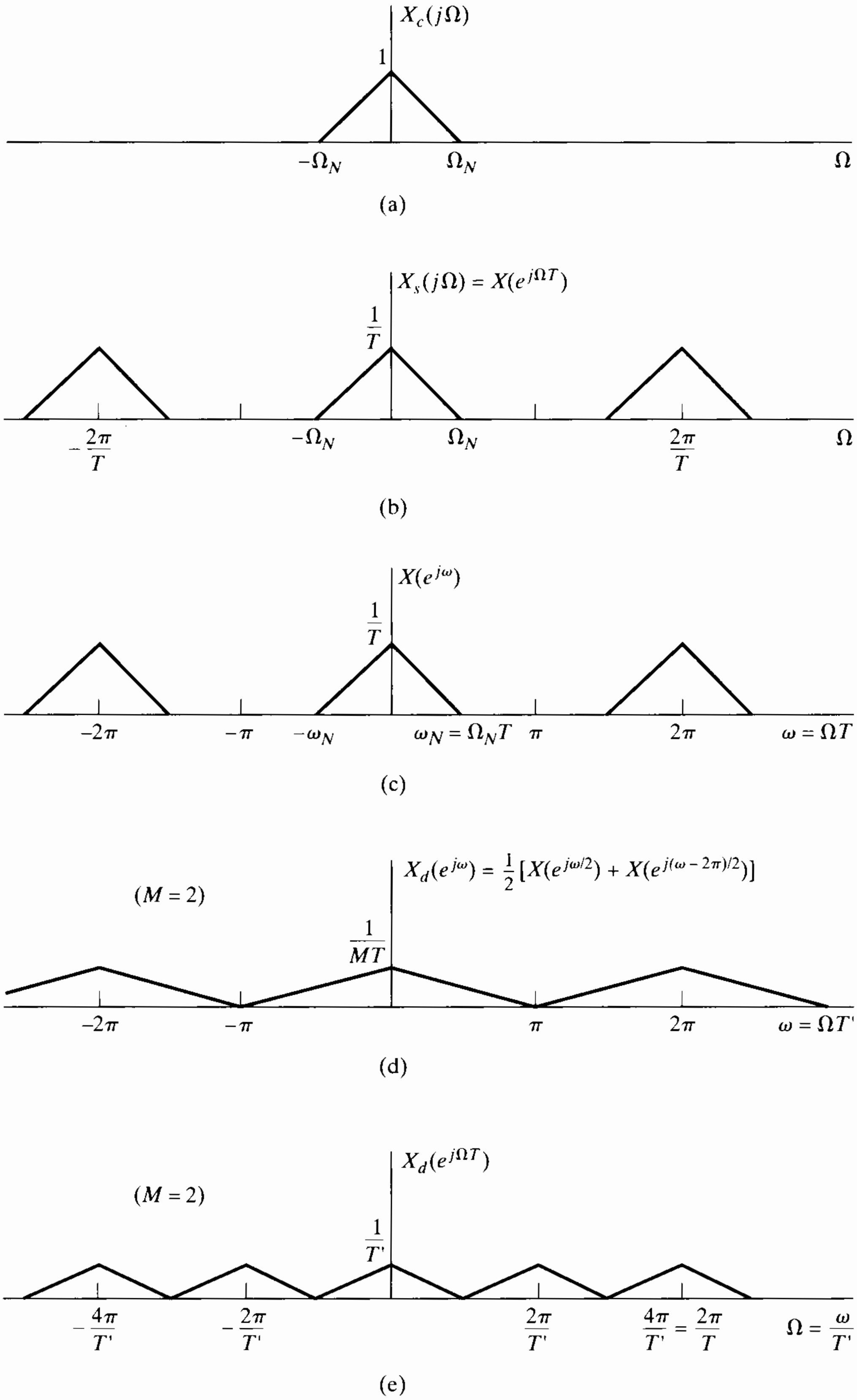

Figure 4.21 Frequency-domain illustration of downsampling.

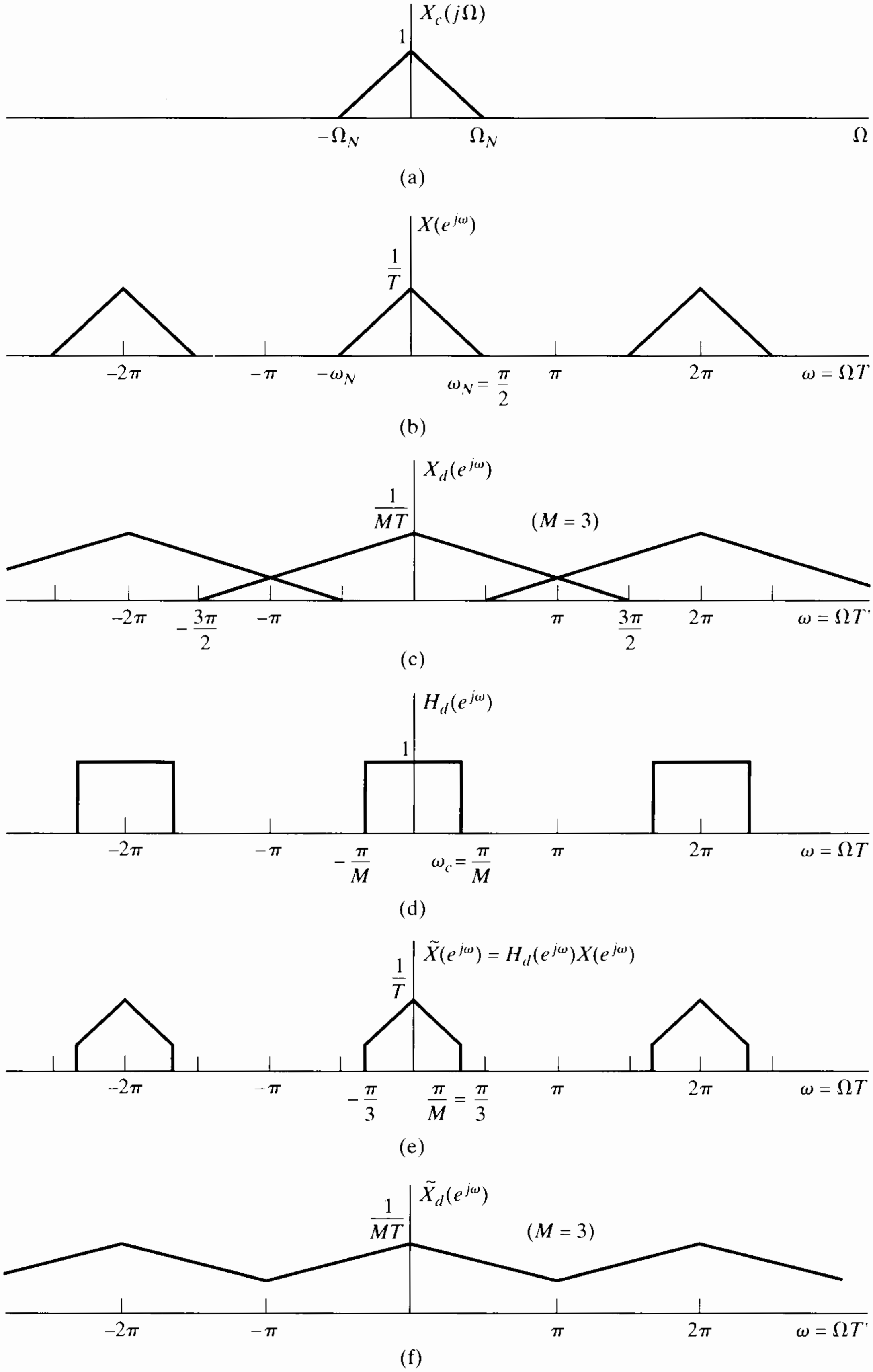

Figure 4.22 (a)–(c) Downsampling with aliasing. (d)–(f) Downsampling with prefiltering to avoid aliasing.

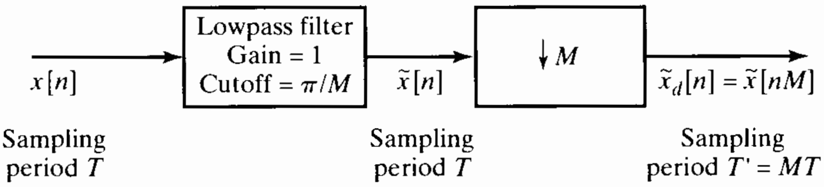

Figure 4.23 General system for sampling rate reduction by M.

From the preceding discussion, we see that a general system for downsampling by a factor of M is the one shown in Figure 4.23. Such a system is called a *decimator,* and downsampling by lowpass filtering followed by compression has been termed *decimation* (Crochiere and Rabiner, 1983).

4.6.2 Increasing the Sampling Rate by an Integer Factor

We have seen that the reduction of the sampling rate of a discrete-time signal by an integer factor involves sampling the sequence in a manner analogous to sampling a continuous-time signal. Not surprisingly, increasing the sampling rate involves operations analogous to D/C conversion. To see this, consider a signal $x[n]$ whose sampling rate we wish to increase by a factor of L. If we consider the underlying continuous-time signal $x_c(t)$, the objective is to obtain samples

$$x_i[n] = x_c(nT'), \tag{4.81}$$

where $T' = T/L$, from the sequence of samples

$$x[n] = x_c(nT). \tag{4.82}$$

We will refer to the operation of increasing the sampling rate as *upsampling*.

From Eqs. (4.81) and (4.82) it follows that

$$x_i[n] = x[n/L] = x_c(nT/L), \qquad n = 0, \pm L, \pm 2L, \ldots. \tag{4.83}$$

Figure 4.24 shows a system for obtaining $x_i[n]$ from $x[n]$ using only discrete-time processing. The system on the left is called a *sampling rate expander* (see Crochiere and Rabiner, 1983) or simply an *expander*. Its output is

$$x_e[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \ldots, \\ 0, & \text{otherwise}, \end{cases} \tag{4.84}$$

or equivalently,

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n - kL]. \tag{4.85}$$

The system on the right is a lowpass discrete-time filter with cutoff frequency π/L and gain L. This system plays a role similar to the ideal D/C converter in Figure 4.10(b). First we create a discrete-time impulse train $x_e[n]$, and then we use a lowpass filter to reconstruct the sequence.

The operation of the system in Figure 4.24 is most easily understood in the frequency domain. The Fourier transform of $x_e[n]$ can be expressed as

$$\begin{aligned} X_e(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} x[k]\delta[n - kL] \right) e^{-j\omega n} \\ &= \sum_{k=-\infty}^{\infty} x[k]e^{-j\omega Lk} = X(e^{j\omega L}). \end{aligned} \tag{4.86}$$

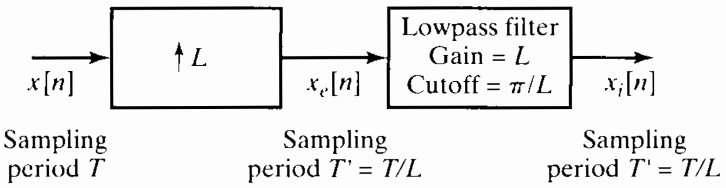

Figure 4.24 General system for sampling rate increase by L.

Thus, the Fourier transform of the output of the expander is a frequency-scaled version of the Fourier transform of the input; i.e., ω is replaced by ωL so that ω is now normalized by

$$\omega = \Omega T'. \tag{4.87}$$

This effect is illustrated in Figure 4.25. Figure 4.25(a) shows a bandlimited continuous-time Fourier transform, and Figure 4.25(b) shows the discrete-time Fourier transform of the sequence $x[n] = x_c(nT)$, where $\pi/T = \Omega_N$. Figure 4.25(c) shows $X_e(e^{j\omega})$ according to Eq. (4.86), with $L = 2$, and Figure 4.25(e) shows the Fourier transform of the desired signal $x_i[n]$. We see that $X_i(e^{j\omega})$ can be obtained from $X_e(e^{j\omega})$ by correcting the amplitude scale from $1/T$ to $1/T'$ and by removing all the frequency-scaled images of $X_c(j\Omega)$ except at integer multiples of 2π. For the case depicted in Figure 4.25, this requires a lowpass filter with a gain of 2 and cutoff frequency $\pi/2$, as shown in Figure 4.25(d). In general, the required gain would be L, since $L(1/T) = [1/(T/L)] = 1/T'$, and the cutoff frequency would be π/L.

This example shows that the system of Figure 4.24 does indeed give an output satisfying Eq. (4.81) if the input sequence $x[n] = x_c(nT)$ was obtained by sampling without aliasing. That system is therefore called an *interpolator*, since it fills in the missing samples, and the operation of upsampling is therefore considered to be synonymous with *interpolation*.

As in the case of the D/C converter, it is possible to obtain an interpolation formula for $x_i[n]$ in terms of $x[n]$. First note that the impulse response of the lowpass filter in Figure 4.24 is

$$h_i[n] = \frac{\sin(\pi n/L)}{\pi n/L}. \tag{4.88}$$

Using Eq. (4.85), we obtain

$$x_i[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{\sin[\pi(n - kL)/L]}{\pi(n - kL)/L}. \tag{4.89}$$

The impulse response $h_i[n]$ has the properties

$$\begin{aligned} h_i[0] &= 1, \\ h_i[n] &= 0, \qquad n = \pm L, \pm 2L, \ldots. \end{aligned} \tag{4.90}$$

Thus, for the ideal lowpass interpolation filter, we have

$$x_i[n] = x[n/L] = x_c(nT/L) = x_c(nT'), \qquad n = 0, \pm L, \pm 2L, \ldots, \tag{4.91}$$

as desired. The fact that $x_i[n] = x_c(nT')$ for all n follows from our frequency-domain argument.

In practice, ideal lowpass filters cannot be implemented exactly, but we will see in Chapter 7 that very good approximations can be designed. (Also, see Schafer and

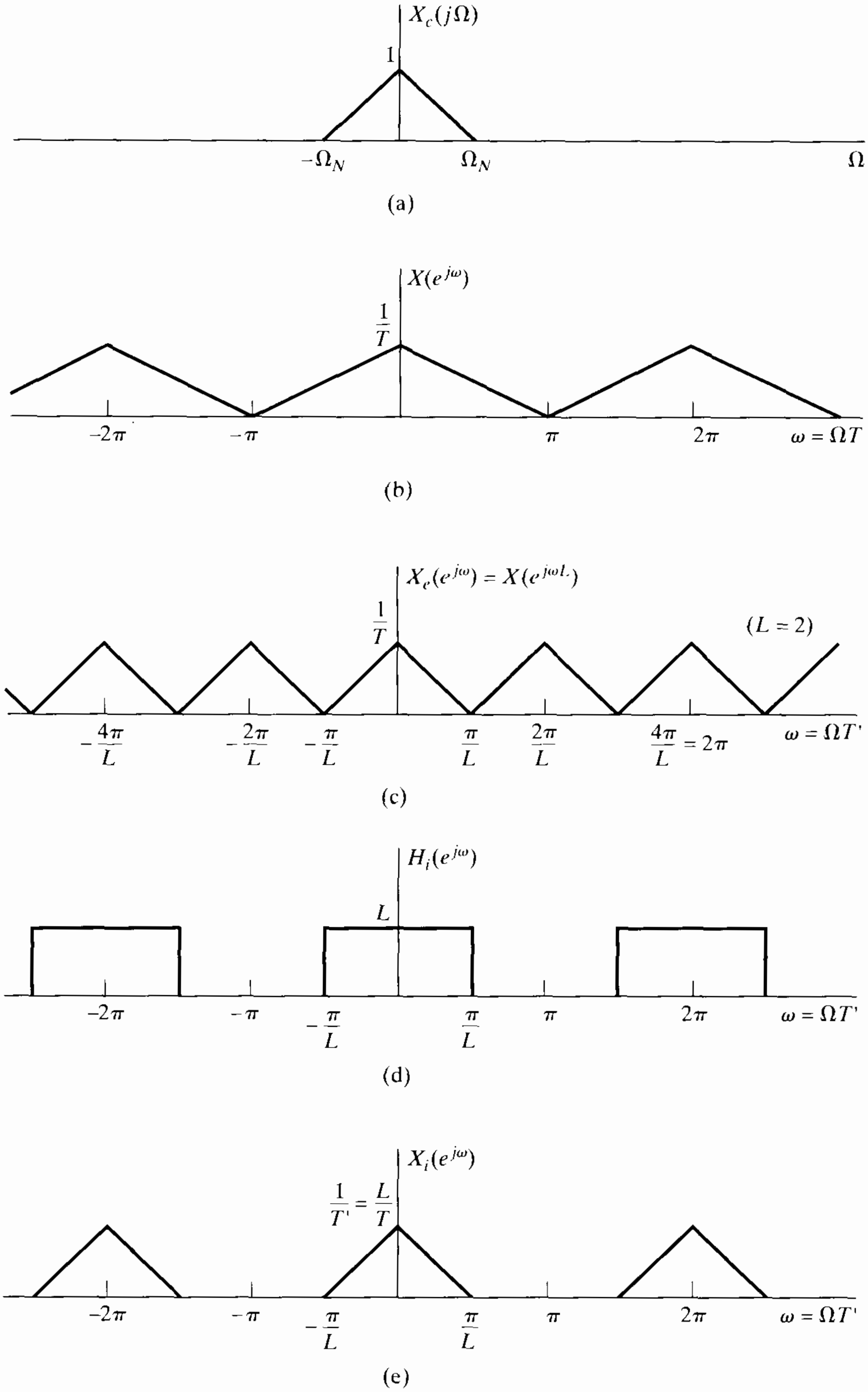

Figure 4.25 Frequency-domain illustration of interpolation.

Rabiner, 1973, and Oetken et al., 1975.) In some cases, very simple interpolation procedures are adequate. Since linear interpolation is often used (even though it is generally not very accurate), it is worthwhile to examine linear interpolation within the general framework that we have just developed.

Linear interpolation can be accomplished by the system of Figure 4.24 if the filter has impulse response

$$h_{\text{lin}}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise}, \end{cases} \tag{4.92}$$

as shown in Figure 4.26 for $L = 5$. With this filter, the interpolated output will be

$$x_{\text{lin}}[n] = \sum_{k=-\infty}^{\infty} x_e[k]h_{\text{lin}}[n-k] = \sum_{k=-\infty}^{\infty} x[k]h_{\text{lin}}[n-kL]. \tag{4.93}$$

Figure 4.27(a) depicts $x_e[n]$ and $x_{\text{lin}}[n]$ for the case $L = 5$. From this figure, we see that

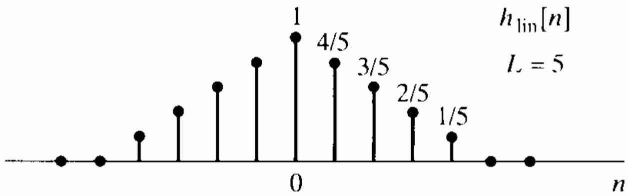

Figure 4.26 Impulse response for linear interpolation.

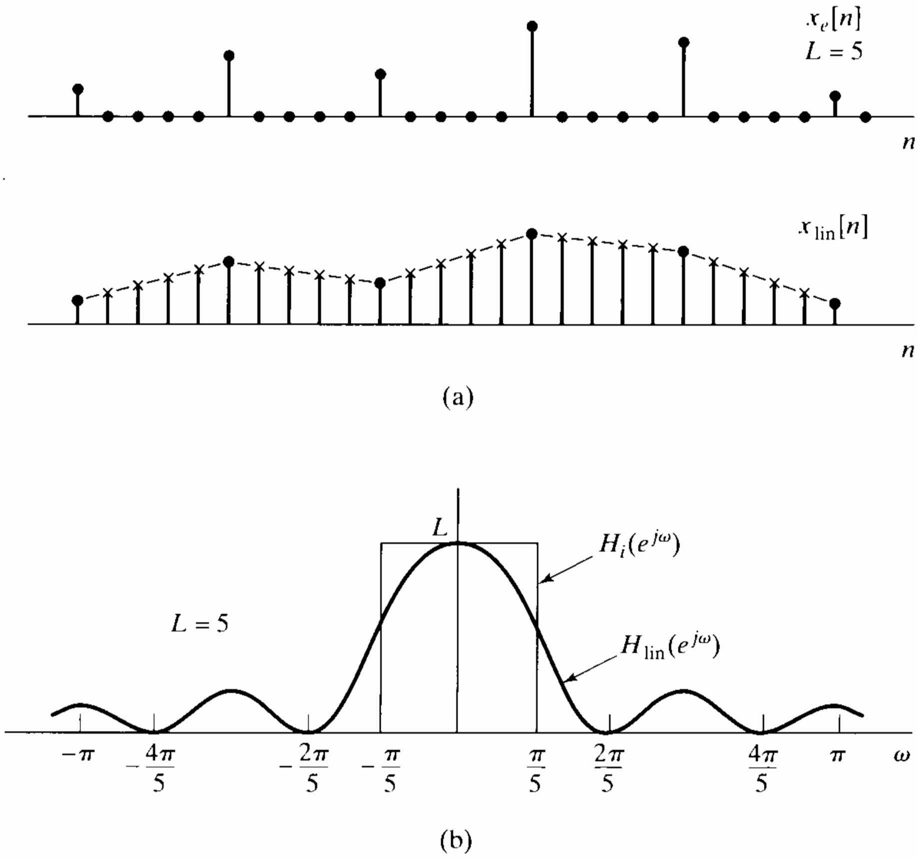

Figure 4.27 (a) Illustration of linear interpolation by filtering. (b) Frequency response of linear interpolator compared with ideal lowpass interpolation filter.

$x_{\text{lin}}[n]$ is identical to the sequence obtained by linear interpolation between the samples. Note that

$$\begin{aligned} h_{\text{lin}}[0] &= 1, \\ h_{\text{lin}}[n] &= 0, \qquad n = \pm L, \pm 2L, \ldots, \end{aligned} \tag{4.94}$$

so that

$$x_{\text{lin}}[n] = x[n/L] \quad \text{at} \quad n = 0, \pm L, \pm 2L, \ldots. \tag{4.95}$$

The amount of distortion in the intervening samples can be gauged by comparing the frequency response of the linear interpolator with that of the ideal lowpass interpolator for a factor-of-L interpolation. It can be shown (see Problem 4.50) that

$$H_{\text{lin}}(e^{j\omega}) = \frac{1}{L}\left[\frac{\sin(\omega L/2)}{\sin(\omega/2)}\right]^2. \tag{4.96}$$

This function is plotted in Figure 4.27(b) for $L = 5$, together with the ideal lowpass interpolation filter. From the figure we see that if the original signal is sampled at the Nyquist rate, linear interpolation will not be very good, since the output of the filter will contain considerable energy in the band $\pi/L < |\omega| \le \pi$. However, if the original sampling rate is much higher than the Nyquist rate, then the linear interpolator will be more successful in removing the frequency-scaled images of $X_c(j\Omega)$ at multiples of $2\pi/L$. This is because $H_{\text{lin}}(e^{j\omega})$ is small at these normalized frequencies and at higher sampling rates the shifted copies of $X_c(j\Omega)$ are more localized at these frequencies. This is intuitively reasonable, since, if the original sampling rate greatly exceeds the Nyquist rate, the signal will not vary significantly between samples, and thus, linear interpolation should be more accurate for oversampled signals.

4.6.3 Changing the Sampling Rate by a Noninteger Factor

We have shown how to increase or decrease the sampling rate of a sequence by an integer factor. By combining decimation and interpolation, it is possible to change the sampling rate by a noninteger factor. Specifically, consider Figure 4.28(a), which shows an interpolator that decreases the sampling period from T to T/L, followed by a decimator that increases the sampling period by M, producing an output sequence $\tilde{x}_d[n]$ that has an effective sampling period of $T' = TM/L$. By choosing L and M appropriately, we can approach arbitrarily close to any desired ratio of sampling periods. For example, if $L = 100$ and $M = 101$, then $T' = 1.01T$.

If $M > L$, there is a net increase in the sampling period (a decrease in the sampling rate), and if $M < L$, the opposite is true. Since the interpolation and decimation filters in Figure 4.28(a) are in cascade, they can be combined as shown in Figure 4.28(b) into one lowpass filter with gain L and cutoff equal to the minimum of π/L and π/M. If $M > L$, then π/M is the dominant cutoff frequency, and there is a net reduction in sampling rate. As pointed out in Section 4.6.1, if $x[n]$ was obtained by sampling at the Nyquist rate, the sequence $\tilde{x}_d[n]$ will be a lowpass-filtered version of the original underlying bandlimited signal if we are to avoid aliasing. On the other hand, if $M < L$, then π/L is the dominant cutoff frequency, and there will be no need to further limit the bandwidth of the signal below the original Nyquist frequency.

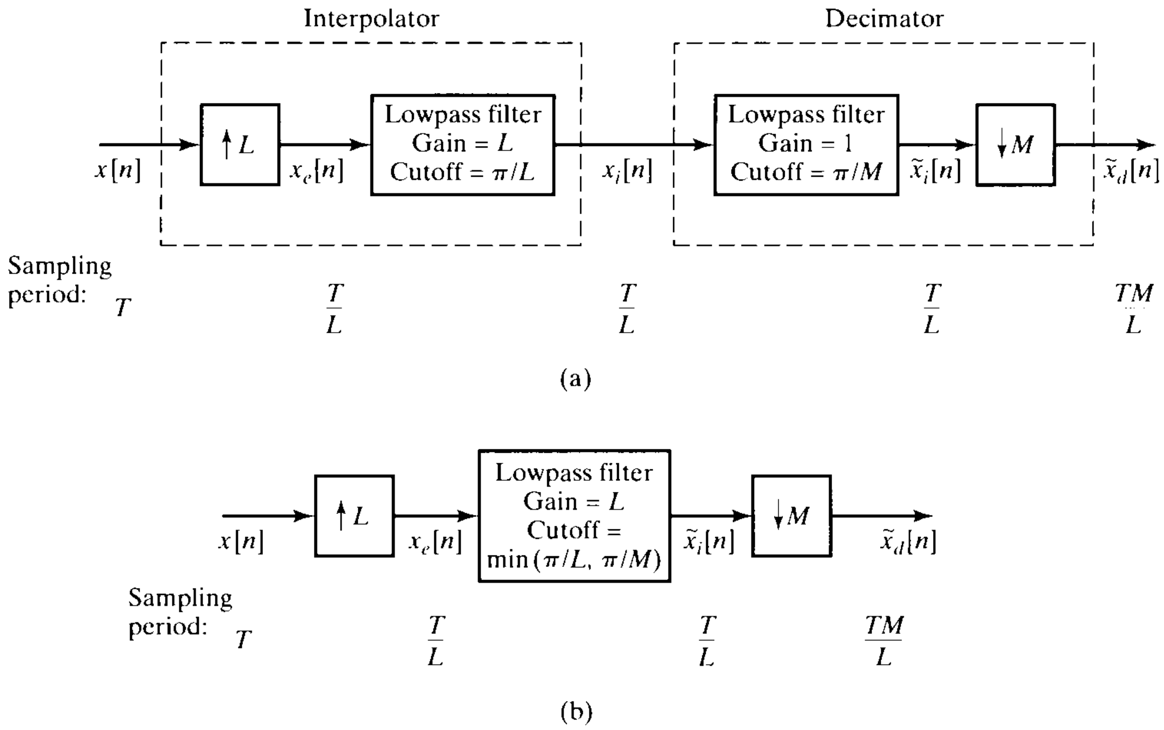

Figure 4.28 (a) System for changing the sampling rate by a noninteger factor. (b) Simplified system in which the decimation and interpolation filters are combined.

Example 4.11 Sampling Rate Conversion by a Noninteger Rational Factor

Figure 4.29 illustrates sampling rate conversion by a rational factor. Suppose that a bandlimited signal with $X_c(j\Omega)$ as given in Figure 4.29(a) is sampled at the Nyquist rate; i.e., $2\pi/T = 2\Omega_N$. The resulting discrete-time Fourier transform

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

is plotted in Figure 4.29(b). If we wish to change the sampling period to $T' = (3/2)T$, we must first interpolate by a factor $L = 2$ and then decimate by a factor of $M = 3$. Since this implies a net decrease in sampling rate, and the original signal was sampled at the Nyquist rate, we must incorporate additional lowpass filtering in order to avoid aliasing.

Figure 4.29(c) shows the discrete-time Fourier transform of the output of the $L = 2$ upsampler. If we were interested only in interpolating by a factor of 2, we could choose the lowpass filter to have a cutoff frequency of $\omega_c = \pi/2$ and a gain of $L = 2$. However, since the output of the filter will be decimated by $M = 3$, we must use a cutoff frequency of $\omega_c = \pi/3$, but the gain of the filter should still be 2 as in Figure 4.29(d). The Fourier transform $\tilde{X}_i(e^{j\omega})$ of the output of the lowpass filter is shown in Figure 4.29(e). The shaded regions indicate the part of the signal spectrum that is removed due to the lower cutoff frequency for the interpolation filter. Finally, Figure 4.29(f) shows the discrete-time Fourier transform of the output of the downsampler by $M = 3$. Note that the shaded regions show the aliasing that would have occurred if the cutoff frequency of the interpolation lowpass filter had been $\pi/2$ instead of $\pi/3$.

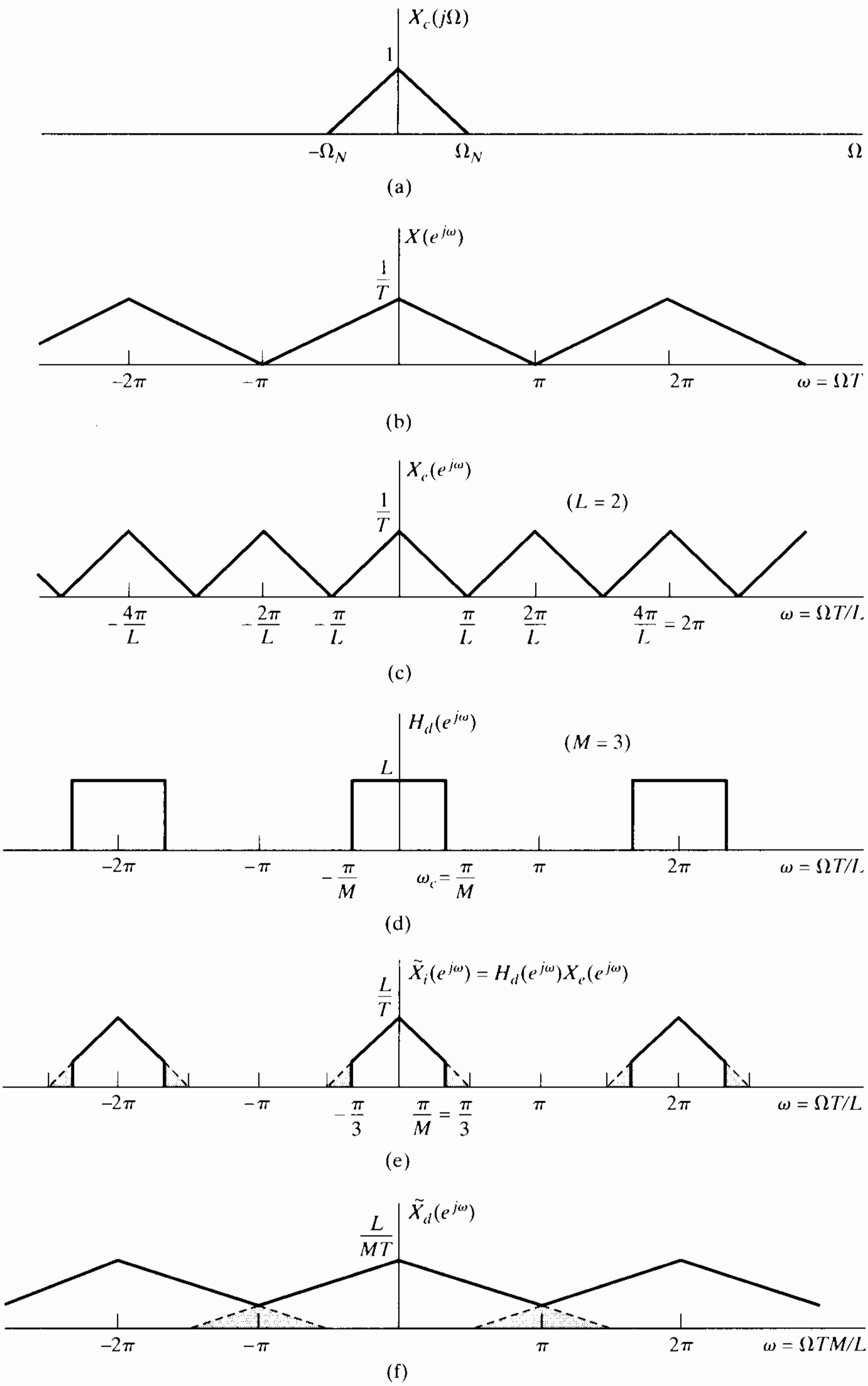

Figure 4.29 Illustration of changing the sampling rate by a noninteger factor.