CHAPTER

11

Multirate Digital Signal Processing

In many practical applications of digital signal processing, one is faced with the problem of changing the sampling rate of a signal, either increasing it or decreasing it by some amount. For example, in telecommunication systems that transmit and receive different types of signals (e.g., teletype, facsimile, speech, video, etc.), there is a requirement to process the various signals at different rates commensurate with the corresponding bandwidths of the signals. The process of converting a signal from a given rate to a different rate is called *sampling rate conversion*. In turn, systems that employ multiple sampling rates in the processing of digital signals are called *multirate digital signal processing systems*.

Sampling rate conversion of a digital signal can be accomplished in one of two general methods. One method is to pass the digital signal through a D/A converter, filter it if necessary, and then to resample the resulting analog signal at the desired rate (i.e., to pass the analog signal through an A/D converter). The second method is to perform the sampling rate conversion entirely in the digital domain.

One apparent advantage of the first method is that the new sampling rate can be arbitrarily selected and need not have any special relationship to the old sampling rate. A major disadvantage, however, is the signal distortion, introduced by the D/A converter in the signal reconstruction, and by the quantization effects in the A/D conversion. Sampling rate conversion performed in the digital domain avoids this major disadvantage.

In this chapter we describe sampling rate conversion and multirate signal processing in the digital domain. First we describe sampling rate conversion by a rational factor and present several methods for implementing the rate converter, including single-stage and multistage implementations. Then, we describe a method for sampling rate conversion by an arbitrary factor and discuss its implementation. We present several applications of sampling rate conversion in multirate signal processing systems, which include the implementation of narrowband filters, digital filter banks, subband coding, transmultiplexers, and quadrature mirror filters.

11.1 Introduction

The process of sampling rate conversion can be developed and understood using the idea of "resampling after reconstruction." In this theoretical approach, a discrete-time signal is ideally reconstructed and the resulting continuous-time signal is resampled at a different sampling rate. This idea leads to a mathematical formulation that enables the realization of the entire process by means of digital signal processing.

Let $x(t)$ be a continuous-time signal that is sampled at a rate $F_x = 1/T_x$ to generate a discrete-time signal $x(nT_x)$. From the samples $x(nT_x)$ we can generate a continuous-time signal using the interpolation formula

$$y(t) = \sum_{n=-\infty}^{\infty} x(nT_x)g(t - nT_x) \tag{11.1.1}$$

If the bandwidth of $x(t)$ is less than $F_x/2$ and the interpolation function is given by

$$g(t) = \frac{\sin(\pi t/T_x)}{\pi t/T_x} \overset{\mathcal{F}}{\longleftrightarrow} G(F) = \begin{cases} T_x, & |F| \le F_x/2 \\ 0, & \text{otherwise} \end{cases} \tag{11.1.2}$$

then $y(t) = x(t)$; otherwise $y(t) \neq x(t)$. In practice, perfect recovery of $x(t)$ is not possible because the infinite summation in (11.1.1) should be replaced by a finite summation.

To perform sampling rate conversion we simply evaluate (11.1.1) at time instants $t = mT_y$, where $F_y = 1/T_y$ is the desired sampling frequency. Therefore, the general formula for sampling rate conversion becomes

$$y(mT_y) = \sum_{n=-\infty}^{\infty} x(nT_x)g(mT_y - nT_x) \tag{11.1.3}$$

which expresses directly the samples of the desired sequence in terms of the samples of the original sequence and sampled values of the reconstruction function at positions $(mT_y - nT_x)$. The computation of $y(nT_y)$ requires (a) the input sequence $x(nT_x)$, (b) the reconstruction function $g(t)$, and (c) the time instants nT_x and mT_y of the input and output samples. The values $y(mT_y)$ calculated by this equation are accurate only if $F_y > F_x$. If $F_y < F_x$, we should filter out the frequency components of $x(t)$ above $F_y/2$ before resampling in order to prevent aliasing. Therefore, the sampling rate conversion formula (11.1.3) yields $y(mT_y) = x(mT_y)$ if we use (11.1.2) and $X(F) = 0$ for $|F| \geq \min\{F_x/2, F_y/2\}$.

If $T_y = T_x$, equation (11.1.3) becomes a convolution summation, which corresponds to an LTI system. To understand the meaning of (11.1.3) for $T_y \neq T_x$, we rearrange the argument of $g(t)$ as follows:

$$y(mT_y) = \sum_{n=-\infty}^{\infty} x(nT_x)g\left(T_x\left(\frac{mT_y}{T_x} - n\right)\right) \tag{11.1.4}$$

The term mT_y/T_x can be decomposed into an integer part k_m and a fractional part Δ_m, $0 \le \Delta_m < 1$, as

$$\frac{mT_y}{T_x} = k_m + \Delta_m \tag{11.1.5}$$

where

$$k_m = \left\lfloor \frac{mT_y}{T_x} \right\rfloor \tag{11.1.6}$$

and

$$\Delta_m = \frac{mT_y}{T_x} - \left\lfloor \frac{mT_y}{T_x} \right\rfloor \tag{11.1.7}$$

The symbol $\lfloor a \rfloor$ denotes the largest integer contained in a. The quantity Δ_m specifies the position of the current sample within the sample period T_x. Substituting (11.1.5) into (11.1.4) we obtain

$$y(mT_y) = \sum_{n=-\infty}^{\infty} x(nT_x)g((k_m + \Delta_m - n)T_x) \tag{11.1.8}$$

If we change the index of summation in (11.1.8) from n to $k = k_m - n$, we have

$$\begin{aligned} y(mT_y) &= y((k_m + \Delta_m)T_x) \\ &= \sum_{k=-\infty}^{\infty} g(kT_x + \Delta_m T_x)x((k_m - k)T_x) \end{aligned} \tag{11.1.9}$$

Equation (11.1.9) provides the fundamental equation for the discrete-time implementation of sampling rate conversion. This process is illustrated in Figure 11.1.1. We note that (a) given T_x and T_y the input and output sampling times are fixed, (b) the function $g(t)$ is shifted for each m such that the value $g(\Delta_m T_x)$ is positioned at $t = mT_y$, and (c) the required values of $g(t)$ are determined at the input sampling times. For each value of m, the fractional interval Δ_m determines the impulse response coefficients whereas the index k_m specifies the corresponding input samples.

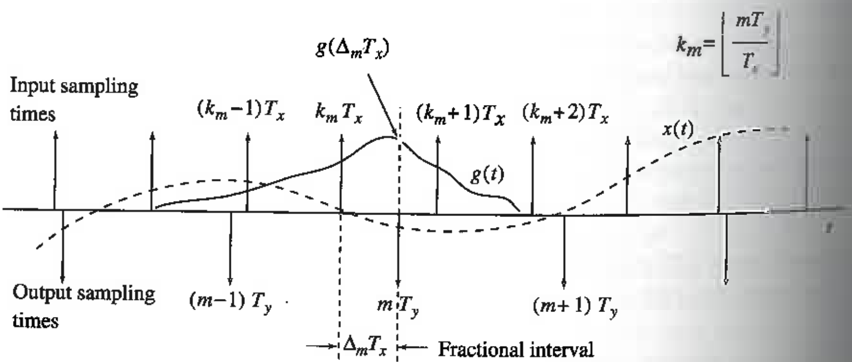

Figure 11.1.1 Illustration of timing relations for sampling rate conversion.

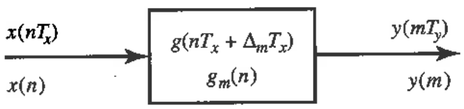

Figure 11.1.2 Discrete-time linear time-varying system for sampling rate conversion.

needed to compute the sample $y(mT_y)$. Since for any given value of m the index k_m is an integer number, $y(mT_y)$ is the convolution between the input sequence $x(nT_x)$ and an impulse response $g\left((n+\Delta_m)\,T_x\right)$. The difference between (11.1.8) and (11.1.9) is that the first shifts a "changing" reconstruction function whereas the second shifts the "fixed" input sequence.

The sampling rate conversion process defined by (11.1.9) is a discrete-time *linear and time-varying* system, because it requires a different impulse response

$$g_m(nT_x) = g((n+\Delta_m)T_x) \tag{11.1.10}$$

for each output sample $y(mT_y)$. Therefore, a new set of coefficients should be computed or retrieved from storage for the computation of *every* output sample (see Figure 11.1.2). This procedure may be inefficient when the function $g(t)$ is complicated and the number of required values is large. This dependence on m prohibits the use of recursive structures because the required past output values must be computed using an impulse response for the present value of Δ_m.

A significant simplification results when the ratio T_y/T_x is constrained to be a rational number, that is,

$$\frac{T_y}{T_x} = \frac{F_x}{F_y} = \frac{D}{I} \tag{11.1.11}$$

where D and I are relatively prime integers. To this end, we express the offset Δ_m as

$$\Delta_m = \frac{mD}{I} - \left\lfloor \frac{mD}{I} \right\rfloor = \frac{1}{I}\left(mD - \left\lfloor \frac{mD}{I} \right\rfloor I\right) = \frac{1}{I}(mD)_I \tag{11.1.12}$$

where $(k)_I$ denotes the value of k modulo I. From (11.1.12) it is clear that Δ_m can take on only I unique values $0, 1/I, \ldots, (I-1)/I$, so that there are only I distinct impulse responses possible. Since $g_m(nT_x)$ can take on I distinct sets of values, it is periodic in m; that is,

$$g_m(nT_x) = g_{m+rI}(nT_x), \quad r = 0, \pm 1, \pm 2, \ldots \tag{11.1.13}$$

Thus the system $g_m(nT_x)$ is a linear and *periodically time-varying* discrete-time system. Such systems have been widely studied for a wide range of applications (Meyers and Burrus, 1975). This is a great simplification compared to the *continuously* time-varying discrete-time system in (11.1.10).

To illustrate these concepts we consider two important special cases. We start with the process of reducing the sampling rate by an integer factor D, which is known as *decimation* or *downsampling*. If we set $T_y = DT_x$ in (11.1.3), we have

$$y(mT_y) = y(mDT_x) = \sum_{k=-\infty}^{\infty} x(kT_x)g((mD-k)T_x) \tag{11.1.14}$$

We note that the input signal and the impulse response are sampled with a period T_x. However, the impulse response is shifted at increments of $T_y = DT_x$ because we need to compute only one out of every D samples. Since $I = 1$, we have $\Delta_m = 0$ and therefore there is only one impulse response $g(nT_x)$, for all m. This process is illustrated in Figure 11.1.3 for $D = 2$.

We consider now the process of increasing the sampling rate by an integer factor I, which is called *upsampling* or *interpolation*. If we set $T_y = T_x/I$ in (11.1.3), we have

$$y(mT_y) = \sum_{k=-\infty}^{\infty} x(kT_x)g(m(T_x/I) - kT_x) \tag{11.1.15}$$

We note that both $x(t)$ and $g(t)$ are sampled with a period T_x; however, the impulse response is shifted at increments of $T_y = T_x/I$ for the computation of each output sample. This is required to "fill-in" an additional number of $(I - 1)$ samples within each period T_x. This is illustrated in Figure 11.1.4(a)–(b) for $I = 2$. Each "fractional shifting" requires that we resample $g(t)$, resulting in a new impulse response $g_m(nT_x) = g(nT_x + mT_x/I)$, for $m = 0, 1, \ldots, I-1$ in agreement with (11.1.14). Careful inspection of Figure 11.1.4(a)–(b) shows that if we determine an impulse response sequence $g(nT_y)$ and create a new sequence $v(nT_y)$ by inserting $(I - 1)$ zero samples between successive samples of $x(nT_x)$, we can compute $y(mT_y)$ as the convolution of the sequences $g(nT_y)$ and $x(nT_y)$. This idea is illustrated in Figure 11.1.4(c) for $I = 2$.

In the next few sections we discuss in more detail the properties, design, and structures for the implementation of sampling rate conversion entirely in the discrete-time domain. For convenience, we usually drop the sampling periods T_x and T_y from the argument of discrete-time signals. However, occasionally, it will be beneficial to the reader to reintroduce and think in terms of continuous-time quantities and units.

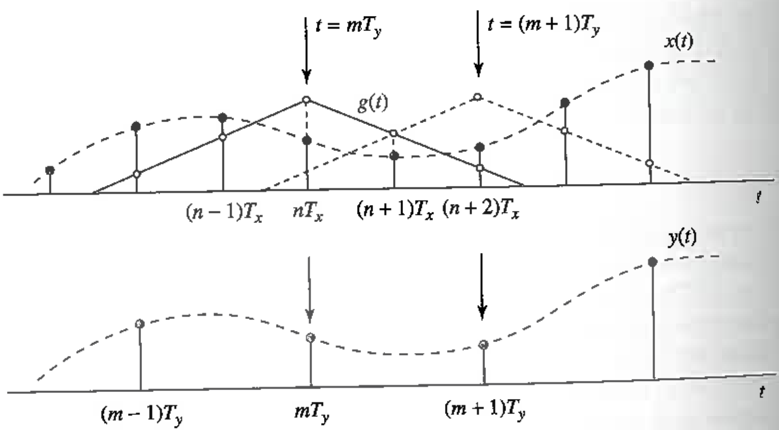

Figure 11.1.3 Illustration of timing relations for sampling rate decrease by an integer factor $D = 2$. A single impulse response, sampled with period T_x, is shifted at steps equal to $T_y = DT_x$ to generate the output samples.

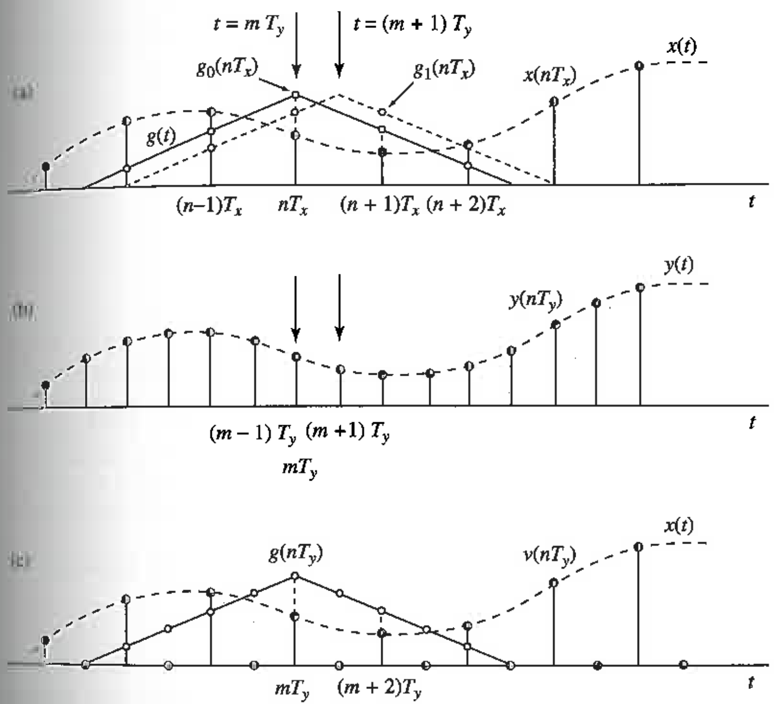

Figure 11.1.4 Illustration of timing relations for sampling rate increase by an integer factor $I = 2$. The approach in (a) requires one impulse response for the even-numbered and one for the odd-numbered output samples. The approach in (c) requires only one impulse response, obtained by interleaving the impulse responses in (a).

11.2 Decimation by a Factor D

Let us assume that the signal $x(n)$ with spectrum $X(\omega)$ is to be downsampled by an integer factor D. The spectrum $X(\omega)$ is assumed to be nonzero in the frequency interval $0 \le |\omega| \le \pi$ or, equivalently, $|F| \le F_x/2$. We know that if we reduce the sampling rate simply by selecting every Dth value of $x(n)$, the resulting signal will be an aliased version of $x(n)$, with a folding frequency of $F_x/2D$. To avoid aliasing, we must first reduce the bandwidth of $x(n)$ to $F_{\max} = F_x/2D$ or, equivalently, to $\omega_{\max} = \pi/D$. Then we may downsample by D and thus avoid aliasing.

The decimation process is illustrated in Fig. 11.2.1. The input sequence $x(n)$ is passed through a lowpass filter, characterized by the impulse response $h(n)$ and a frequency response $H_D(\omega)$, which ideally satisfies the condition

$$H_D(\omega) = \begin{cases} 1, & |\omega| \le \pi/D \\ 0, & \text{otherwise} \end{cases} \tag{11.2.1}$$

Thus the filter eliminates the spectrum of $X(\omega)$ in the range $\pi/D < \omega < \pi$. Of course, the implication is that only the frequency components of $x(n)$ in the range $|\omega| \le \pi/D$ are of interest in further processing of the signal.