PET ENGINEERING COLLEGE

An ISO 9001:2015 Certified Institution

Accredited by NAAC, Approved by AICTE, Recognized by Government of Tamil Nadu and Affiliated to Anna University

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

UNIT – V

DSP APPLICATIONS

CLASS : **S4 ECE**

SUBJECT CODE : **EC3492**

SUBJECT NAME : **DIGITAL SIGNAL PROCESSING**

REGULATION : **2021**

UNIT 5

DSP APPLICATIONS

INTRODUCTION

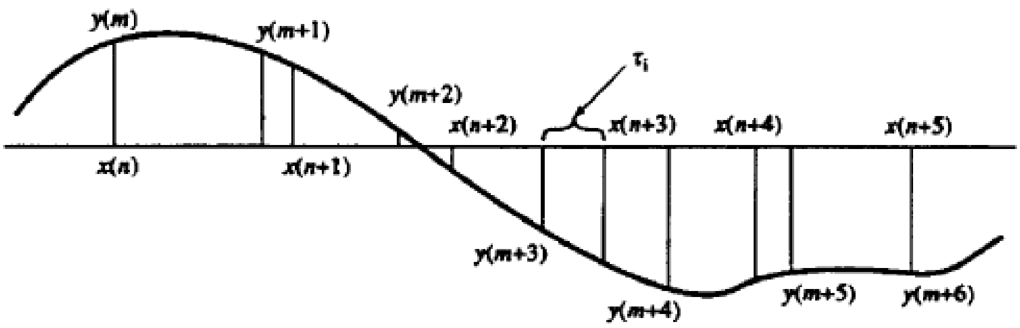
The process of sampling rate conversion in the digital domain can be viewed as a linear filtering operation, as illustrated in Fig. 10.1(a). The input signal $x(n)$ is characterized by the sampling rate $F_x = 1/T_x$ and the output signal $y(m)$ is characterized by the sampling rate $F_y = 1/T_y$, where T_x and T_y are the corresponding sampling intervals. In the main part of our treatment, the ratio F_y/F_x is constrained to be rational,

$$\frac{F_y}{F_x} = \frac{I}{D}$$

where D and I are relatively prime integers. We shall show that the linear filter is characterized by a time-variant impulse response, denoted as $h(n, m)$. Hence the input $x(n)$ and the output $y(m)$ are related by the convolution summation for time-variant systems.

The sampling rate conversion process can also be understood from the point of view of digital resampling of the same analog signal. Let $x(t)$ be the analog signal that is sampled at the first rate F_x to generate $x(n)$. The goal of rate conversion is to obtain another sequence $y(m)$ directly from $x(n)$, which is equal to the sampled values of $x(t)$ at a second rate F_y. As is depicted in Fig. 10.1(b), $y(m)$ is a time-shifted version of $x(n)$. Such a time shift can be

(a)



(b)

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. 10.2. 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

$$H_D(\omega) = \begin{cases} 1, & |\omega| \le \pi/D \\ 0, & \text{otherwise} \end{cases} \tag{10.2.1}$$

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.

The output of the filter is a sequence $v(n)$ given as

$$v(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \tag{10.2.2}$$

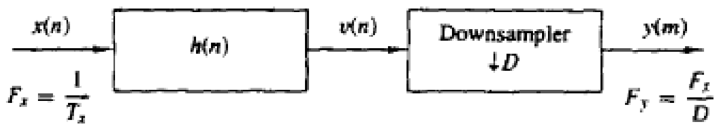


Figure 10.2 Decimation by a factor D.

which is then downsampled by the factor D to produce $y(m)$. Thus

$$\begin{aligned} y(m) &= v(mD) \\ &= \sum_{k=0}^{\infty} h(k)x(mD-k) \end{aligned} \tag{10.2.3}$$

Although the filtering operation on $x(n)$ is linear and time invariant, the downsampling operation in combination with the filtering results in a time-variant system. This is easily verified. Given the fact that $x(n)$ produces $y(m)$, we note that $x(n-n_0)$ does not imply $y(n-n_0)$ unless n_0 is a multiple of D. Consequently, the overall linear operation (linear filtering followed by downsampling) on $x(n)$ is not time invariant.

The frequency-domain characteristics of the output sequence $y(m)$ can be obtained by relating the spectrum of $y(\mathrm{m})$ to the spectrum of the input sequence $x(n)$. First, it is convenient to define a sequence $\tilde{v}(n)$ as

$$\tilde{v}(n) = \begin{cases} v(n), & n = 0, \pm D, \pm 2D, \dots \\ 0, & \text{otherwise} \end{cases} \tag{10.2.4}$$

Clearly, $\tilde{v}(n)$ can be viewed as a sequence obtained by multiplying $v(n)$ with a periodic train of impulses $p(n)$, with period D, as illustrated in Fig. 10.3. The discrete Fourier series representation of $p(n)$ is

$$p(n) = \frac{1}{D}\sum_{k=0}^{D-1} e^{j2\pi kn/D} \tag{10.2.5}$$

Hence

$$\tilde{v}(n) = v(n)p(n) \tag{10.2.6}$$

and

$$y(m) = \tilde{v}(mD) = v(mD)p(mD) = v(mD) \tag{10.2.7}$$

Now the z-transform of the output sequence $y(m)$ is

$$\begin{aligned} Y(z) &= \sum_{m=-\infty}^{\infty} y(m)z^{-m} \\ &= \sum_{m=-\infty}^{\infty} \tilde{v}(mD)z^{-m} \\ Y(z) &= \sum_{m=-\infty}^{\infty} \tilde{v}(m)z^{-m/D} \end{aligned} \tag{10.2.8}$$

where the last step follows from the fact that $\tilde{v}(m) = 0$, except at multiples of D. By making use of the relations in (10.2.5) and (10.2.6) in (10.2.8), we obtain

$$
\begin{aligned}
Y(z) &= \sum_{m=-\infty}^{\infty} v(m)\left[\frac{1}{D}\sum_{k=0}^{D-1} e^{j2\pi mk/D}\right] z^{-m/D} \\
&= \frac{1}{D}\sum_{k=0}^{D-1}\sum_{m=-\infty}^{\infty} v(m)(e^{-j2\pi k/D} z^{1/D})^{-m} \\
&= \frac{1}{D}\sum_{k=0}^{D-1} V(e^{-j2\pi k/D} z^{1/D}) \\
&= \frac{1}{D}\sum_{k=0}^{D-1} H_D(e^{-j2\pi k/D} z^{1/D}) X(e^{-j2\pi k/D} z^{1/D})
\end{aligned}
\tag{10.2.9}
$$

where the last step follows from the fact that $V(z) = H_D(z)X(z)$.

By evaluating $Y(z)$ in the unit circle, we obtain the spectrum of the output signal $y(m)$. Since the rate of $y(m)$ is $F_y = 1/T_y$, the frequency variable, which we denote as ω_y, is in radians and is relative to the sampling rate F_y,

$$\omega_y = \frac{2\pi F}{F_y} = 2\pi F T_y \tag{10.2.10}$$

Since the sampling rates are related by the expression

$$F_y = \frac{F_x}{D} \tag{10.2.11}$$

it follows that the frequency variables ω_y and

$$\omega_x = \frac{2\pi F}{F_x} = 2\pi F T_x \tag{10.2.12}$$

are related by

$$\omega_y = D\omega_x \tag{10.2.13}$$

Thus, as expected, the frequency range $0 \le |\omega_x| \le \pi/D$ is stretched into the corresponding frequency range $0 \le |\omega_y| \le \pi$ by the downsampling process.

We conclude that the spectrum $Y(\omega_y)$, which is obtained by evaluating (10.2.9) on the unit circle, can be expressed as

$$Y(\omega_y) = \frac{1}{D}\sum_{k=0}^{D-1} H_D\left(\frac{\omega_y - 2\pi k}{D}\right) X\left(\frac{\omega_y - 2\pi k}{D}\right) \tag{10.2.14}$$

With a properly designed filter $H_D(\omega)$, the aliasing is eliminated and, consequently, all but the first term in (10.2.14) vanish. Hence

$$
\begin{aligned}
Y(\omega_y) &= \frac{1}{D} H_D\left(\frac{\omega_y}{D}\right) X\left(\frac{\omega_y}{D}\right) \\
&= \frac{1}{D} X\left(\frac{\omega_y}{D}\right)
\end{aligned}
\tag{10.2.15}
$$

for $0 \le |\omega_y| \le \pi$. The spectra for the sequences $x(n)$, $v(n)$, and $y(m)$ are illustrated in Fig. 10.4.

INTERPOLATION BY A FACTOR I

An increase in the sampling rate by an integer factor of I can be accomplished by interpolating $I - 1$ new samples between successive values of the signal. The interpolation process can be accomplished in a variety of ways. We shall describe a process that preserves the spectral shape of the signal sequence $x(n)$.

Let $v(m)$ denote a sequence with a rate $F_y = IF_x$, which is obtained from $x(n)$ by adding $I - 1$ zeros between successive values of $x(n)$. Thus

$$v(m) = \begin{cases} x(m/I), & m = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{10.3.1}$$

and its sampling rate is identical to the rate of $y(m)$. This sequence has a z-transform

$$\begin{aligned} V(z) &= \sum_{m=-\infty}^{\infty} v(m)z^{-m} \\ &= \sum_{m=-\infty}^{\infty} x(m)z^{-mI} \\ &= X(z^I) \end{aligned} \tag{10.3.2}$$

The corresponding spectrum of $v(m)$ is obtained by evaluating (10.3.2) on the unit circle. Thus

$$V(\omega_y) = X(\omega_y I) \tag{10.3.3}$$

where ω_y denotes the frequency variable relative to the new sampling rate F_y (i.e., $\omega_y = 2\pi F/F_y$). Now the relationship between sampling rates is $F_y = IF_x$ and hence, the frequency variables ω_x and ω_y are related according to the formula

$$\omega_y = \frac{\omega_x}{I} \tag{10.3.4}$$

The spectra $X(\omega_x)$ and $V(\omega_y)$ are illustrated in Fig. 10.5. We observe that the sampling rate increase, obtained by the addition of $I - 1$ zero samples between successive values of $x(n)$, results in a signal whose spectrum $V(\omega_y)$ is an I-fold periodic repetition of the input signal spectrum $X(\omega_x)$.

Since only the frequency components of $x(n)$ in the range $0 \le \omega_y \le \pi/I$ are unique, the images of $X(\omega)$ above $\omega_y = \pi/I$ should be rejected by passing the sequence $v(m)$ through a lowpass filter with frequency response $H_I(\omega_y)$ that ideally has the characteristic

$$H_I(\omega_y) = \begin{cases} C, & 0 \le |\omega_y| \le \pi/I \\ 0, & \text{otherwise} \end{cases} \tag{10.3.5}$$

where C is a scale factor required to properly normalize the output sequence $y(m)$. Consequently, the output spectrum is

$$Y(\omega_y) = \begin{cases} CX(\omega_y I), & 0 \le |\omega_y| \le \pi/I \\ 0, & \text{otherwise} \end{cases} \tag{10.3.6}$$

The scale factor C is selected so that the output $y(m) = x(m/I)$ for $m = 0, \pm I, +2I, \ldots$. For mathematical convenience, we select the point $m = 0$. Thus

$$\begin{aligned} y(0) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} Y(\omega_y)d\omega_y \\ &= \frac{C}{2\pi}\int_{-\pi/I}^{\pi/I} X(\omega_y I)d\omega_y \end{aligned} \tag{10.3.7}$$

Since $\omega_y = \omega_x/I$, (10.3.7) can be expressed as

$$\begin{aligned} y(0) &= \frac{C}{I}\frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega_x)d\omega_x \\ &= \frac{C}{I}x(0) \end{aligned} \tag{10.3.8}$$

Therefore, $C = I$ is the desired normalization factor.

Finally, we indicate that the output sequence $y(m)$ can be expressed as a convolution of the sequence $v(n)$ with the unit sample response $h(n)$ of the lowpass filter. Thus

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-k)v(k) \tag{10.3.9}$$

Since $v(k) = 0$ except at multiples of I, where $v(kI) = x(k)$, (10.3.9) becomes

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-kI)x(k) \tag{10.3.10}$$

SAMPLING RATE CONVERSION BY A RATIONAL FACTOR I/D

Having discussed the special cases of decimation (downsampling by a factor D) and interpolation (upsampling by a factor I), we now consider the general case of sampling rate conversion by a rational factor I/D. Basically, we can achieve this sampling rate conversion by first performing interpolation by the factor I and then decimating the output of the interpolator by the factor D. In other words, a sampling rate conversion by the rational factor I/D is accomplished by cascading an interpolator with a decimator, as illustrated in Fig. 10.6.

We emphasize that the importance of performing the interpolation first and the decimation second, is to preserve the desired spectral characteristics of $x(n)$. Furthermore, with the cascade configuration illustrated in Fig. 10.6, the two filters with impulse response $\{h_u(l)\}$ and $\{h_d(l)\}$ are operated at the same rate, namely IF_x and hence can be combined into a single lowpass filter with impulse response $h(l)$ as illustrated in Fig. 10.7. The frequency response $H(\omega_v)$ of the combined filter must incorporate the filtering operations for both interpolation and decimation, and hence it should ideally possess the frequency response characteristic

$$H(\omega_v) = \begin{cases} I, & 0 \le |\omega_v| \le \min(\pi/D, \pi/I) \\ 0, & \text{otherwise} \end{cases} \quad (10.4.1)$$

where $\omega_v = 2\pi F/F_v = 2\pi F/I F_x = \omega_x/I$.

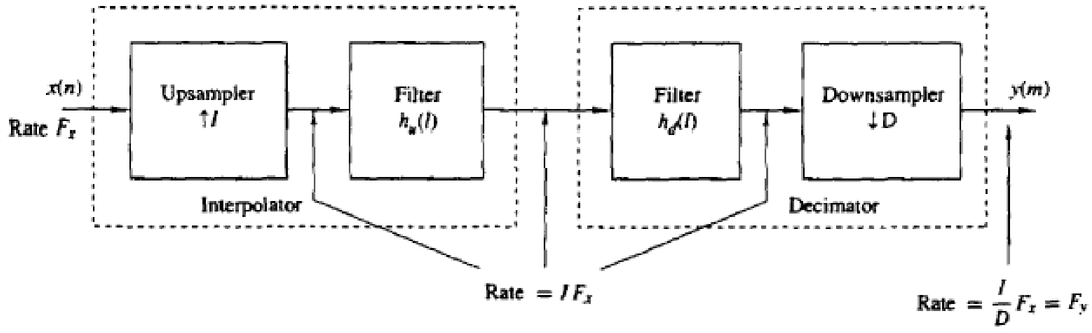


Figure 10.6 Method for sampling rate conversion by a factor I/D.

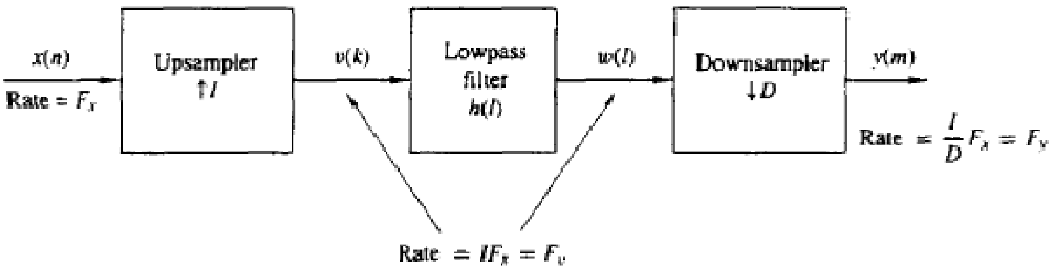


Figure 10.7 Method for sampling rate conversion by a factor I/D.

In the time domain, the output of the upsampler is the sequence

$$v(l) = \begin{cases} x(l/I), & l = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases} \quad (10.4.2)$$

and the output of the linear time-invariant filter is

$$\begin{aligned} w(l) &= \sum_{k=-\infty}^{\infty} h(l-k)v(k) \\ &= \sum_{k=-\infty}^{\infty} h(l-kI)x(k) \end{aligned} \quad (10.4.3)$$

Finally, the output of the sampling rate converter is the sequence $\{y(m)\}$, which is obtained by downsampling the sequence $\{w(l)\}$ by a factor of D. Thus

$$\begin{aligned} y(m) &= w(mD) \\ &= \sum_{k=-\infty}^{\infty} h(mD - kI)x(k) \end{aligned} \quad (10.4.4)$$

It is illuminating to express (10.4.4) in a different form by making a change in variable. Let

$$k = \left\lfloor \frac{mD}{I} \right\rfloor - n \tag{10.4.5}$$

where the notation $\lfloor r \rfloor$ denotes the largest integer contained in r. With this change in variable, (10.4.4) becomes

$$y(m) = \sum_{n=-\infty}^{\infty} h\left(mD - \left\lfloor \frac{mD}{I} \right\rfloor I + nI\right) x\left(\left\lfloor \frac{mD}{I} \right\rfloor - n\right) \tag{10.4.6}$$

We note that

$$\begin{aligned} mD - \left\lfloor \frac{mD}{I} \right\rfloor I &= mD \qquad \text{modulo } I \\ &= (mD)_I \end{aligned}$$

Consequently, (10.4.6) can be expressed as

$$y(m) = \sum_{n=-\infty}^{\infty} h(nI + (mD)_I) x\left(\left\lfloor \frac{mD}{I} \right\rfloor - n\right) \tag{10.4.7}$$

It is apparent from this form that the output $y(m)$ is obtained by passing the input sequence $x(n)$ through a time-variant filter with impulse response

$$g(n, m) = h(nI + (mD)_I) \qquad -\infty < m, n < \infty \tag{10.4.8}$$

where $h(k)$ is the impulse response of the time-invariant lowpass filter operating at the sampling rate IF_x. We further observe, that for any integer k,

$$\begin{aligned} g(n, m + kI) &= h(nI + (mD + kDI)_I) \\ &= h(nI + (mD)_I) \\ &= g(n, m) \end{aligned} \tag{10.4.9}$$

Hence $g(n, m)$ is periodic in the variable m with period I.

The frequency-domain relationships can be obtained by combining the results of the interpolation and decimation processes. Thus the spectrum at the output of the linear filter with impulse response $h(l)$ is

$$\begin{aligned} V(\omega_v) &= H(\omega_v) X(\omega_v I) \\ &= \begin{cases} I X(\omega_v I), & 0 \le |\omega_v| \le \min(\pi/D, \pi/I) \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{10.4.10}$$

The spectrum of the output sequence $y(m)$, obtained by decimating the sequence $v(n)$ by a factor of D, is

$$Y(\omega_y) = \frac{1}{D} \sum_{k=0}^{D-1} V\left(\frac{\omega_y - 2\pi k}{D}\right) \tag{10.4.11}$$

where $\omega_y = D\omega_v$. Since the linear filter prevents aliasing as implied by (10.4.10), the spectrum of the output sequence given by (10.4.11) reduces to

$$Y(\omega_y) = \begin{cases} \dfrac{I}{D} X\left(\dfrac{\omega_y}{D}\right), & 0 \le |\omega_y| \le \min\left(\pi, \dfrac{\pi D}{I}\right) \\ 0, & \text{otherwise} \end{cases} \tag{10.4.12}$$

SAMPLING-RATE CONVERSION BY AN ARBITRARY FACTOR

In the previous sections of this chapter, we have shown how to perform sampling rate conversion exactly by a rational number I/D. In some applications, it is either inefficient or, sometimes impossible to use such an exact rate conversion scheme. We first consider the following two cases.

Case 1. We need to perform rate conversion by the rational number I/D, where I is a large integer (e.g., $I/D = 1023/511$). Although we can achieve exact rate conversion by this number, we would need a polyphase filter with 1023 subfilters. Such an exact implementation is obviously inefficient in memory usage because we need to store a large number of filter coefficients.

Case 2. In some applications, the exact conversion rate is not known when we design the rate converter, or the rate is continuously changing during the conversion process. For example, we may encounter the situation where the input and output samples are controlled by two independent clocks. Even though it is still possible to define a nominal conversion rate that is a rational number, the actual rate would be slightly different, depending on the frequency difference between the two clocks. Obviously, it is not possible to design an exact rate converter in this case.

To implement sampling rate conversion for applications similar to these cases, we resort to nonexact rate conversion schemes. Unavoidably, a nonexact scheme will introduce some distortion in the converted output signal. (It should be noted that distortion exists even in an exact rational rate converter because the polyphase filter is never ideal.) Such a converter will be adequate, as long as the total distortion does not exceed the specification required in the application.

Depending on the application requirements and implementation constraints, we can use first-order, second-order, or higher-order approximations. We shall describe first-order and second-order approximation methods and provide an analysis of the resulting timing errors.

10.8.1 First-Order Approximation

Let us denote the arbitrary conversion rate by r and suppose that the input to the rate converter is the sequence $\{x(n)\}$. We need to generate a sequence of output samples separated in time by T_x/r, where T_x is the sample interval for $\{x(n)\}$. By constructing a polyphase filter with a large number of subfilters as just described, we can approximate such a sequence with a nonuniformly spaced sequence. Without loss of generality, we can express $1/r$ as

$$\frac{1}{r} = \frac{k}{I} + \beta$$

where k and I are positive integers and β is a number in the range

$$0 < \beta < \frac{1}{I}$$

Consequently, $1/r$ is bounded from above and below as

$$\frac{k}{I} < \frac{1}{r} < \frac{k+1}{I}$$

I corresponds to the interpolation factor, which will be determined to satisfy the specification on the amount of tolerable distortion introduced by rate conversion. I is also equal to the number of polyphase filters.

For example, suppose that $r = 2.2$ and that we have determined, as we will demonstrate, that $I = 6$ polyphase filters are required to meet the distortion specification. Then

$$\frac{k}{I} \equiv \frac{2}{6} < \frac{1}{r} < \frac{3}{6} \equiv \frac{k+1}{I}$$

so that $k = 2$. The time spacing between samples of the interpolated sequence is T_x/I. However, the desired conversion rate $r = 2.2$ for $I = 6$ corresponds to a decimation factor of 2.727, which falls between $k = 2$ and $k = 3$. In the first-order approximation, we achieve the desired decimation rate by selecting the output sample from the polyphase filter closest in time to the desired sampling time. This is illustrated in Fig. 10.27 for $I = 6$.

In general, to perform rate conversion by a factor r, we employ a polyphase filter to perform interpolation and therefore to increase the frequency of the original sequence of a factor of I. The time spacing between the samples of the interpolated sequence is equal to T_x/I. If the ideal sampling time of the mth sample, $y(m)$, of the desired output sequence is between the sampling times of two samples of the interpolated sequence, we select the sample closer to $y(m)$ as its approximation.

Let us assume that the mth selected sample is generated by the (i_m)th subfilter using the input samples $x(n), x(n-1), \ldots, x(n-K+1)$ in the delay line. The normalized sampling time error (i.e., the time difference between the selected sampling time and the desired sampling time normalized by T_x) is denoted by t_m. The sign of t_m is positive if the desired sampling time leads the selected sampling time, and negative otherwise. It is easy to show that $|t_m| \leq 0.5/I$. The normalized time advance from the mth output $y(m)$ to the $(m+1)$st output $y(m+1)$ is equal to $(1/r) + t_m$.

To compute the next output, we first determine a number closest to $i_m/I + 1/r + t_m + k_m/I$ that is of the form $l_{m-1} + i_{m+1}/I$, where both l_{m+1} and i_{m+1} are integers and $i_{m+1} < I$. Then, the $(m+1)$st output $y(m+1)$ is computed using the (i_{m+1})th subfilter after shifting the signal in the delay line by l_{m+1} input samples. The normalized timing error for the $(m+1)$th sample is $t_{m+1} = (i_m/I + 1/r + t_m) - (l_{m+1} + i_{m+1}/I)$. It is saved for the computation of the next output sample.

By increasing the number of subfilters used, we can arbitrarily increase the conversion accuracy. However, we also require more memory to store the large number of filter coefficients. Hence it is desirable to use as few subfilters as possible while keeping the distortion in the converted signal below the specification. The distortion introduced due to the sampling-time approximation is most conveniently evaluated in the frequency domain.

Suppose that the input data sequence $\{x(n)\}$ has a flat spectrum from $-\omega_x$ to ω_x, where $\omega_x < \pi$, with a magnitude A. Its total power can be computed using Parseval's theorem, namely,

$$P_s = \frac{1}{2\pi} \int_{-\omega_x}^{\omega_x} |X(\omega)|^2 d\omega = \frac{A^2 \omega_x}{\pi} \tag{10.8.1}$$

APPLICATIONS OF MULTIRATE SIGNAL PROCESSING

There are numerous practical applications of multirate signal processing. In this section we describe a few of these applications.

10.9.1 Design of Phase Shifters

Suppose that we wish to design a network that delays the signal $x(n)$ by a fraction of a sample. Let us assume that the delay is a rational fraction of a sampling

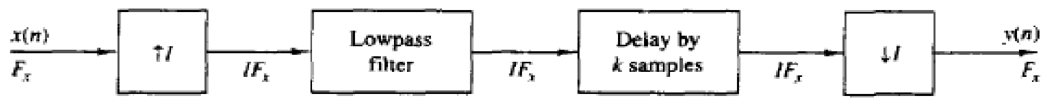


Figure 10.29 Method for generating a delay in a discrete-time signal.

interval T_x [i.e., $d = (k/I)T_x$, where k and I are relatively prime positive integers]. In the frequency domain, the delay corresponds to a linear phase shift of the form

$$\Theta(\omega) = -\frac{k\omega}{I} \tag{10.9.1}$$

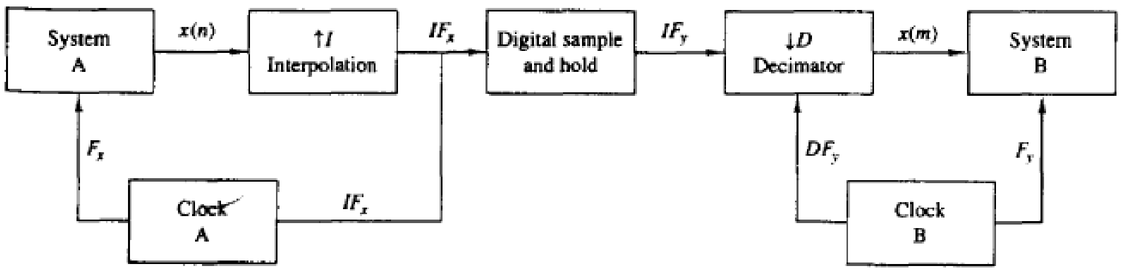
The design of an all-pass linear-phase filter is relatively difficult. However, we can use the methods of sample-rate conversion to achieve a delay of $(k/I)T_x$, exactly, without introducing any significant distortion in the signal. To be specific, let us consider the system shown in Fig. 10.29. The sampling rate is increased by a factor I using a standard interpolator. The lowpass filter eliminates the images in the spectrum of the interpolated signal, and its output is delayed by k samples at the sampling rate IF_x. The delayed signal is decimated by a factor $D = I$. Thus we have achieved the desired delay of $(k/I)T_x$.

10.9.2 Interfacing of Digital Systems with Different Sampling Rates

In practice we frequently encounter the problem of interfacing two digital systems that are controlled by independently operating clocks. An analog solution to this problem is to convert the signal from the first system to analog form and then resample it at the input to the second system using the clock in this system. However, a simpler approach is one where the interfacing is done by a digital method using the basic sample-rate conversion methods described in this chapter.

To be specific, let us consider interfacing the two systems with independent clocks as shown in Fig. 10.31. The output of system A at rate F_x is fed to an interpolator which increases the sampling rate by I. The output of the interpolator is fed at the rate IF_x to a digital sample-and-hold which serves as the interface to system B at the high sampling rate IF_x. Signals from the digital sample-and-hold are read out into system B at the clock rate DF_y of system B. Thus the output rate from the sample-and-hold is not synchronized with the input rate.

In the special case where $D = I$ and the two clock rates are comparable but not identical, some samples at the output of the sample-and-hold may be repeated or dropped at times. The amount of signal distortion resulting from this method can be kept small if the interpolator/decimator factor is large. By using linear interpolation in place of the digital sample-and-hold, as we described in Section 10.8, we can further reduce the distortion and thus reduce the size of the interpolator factor.



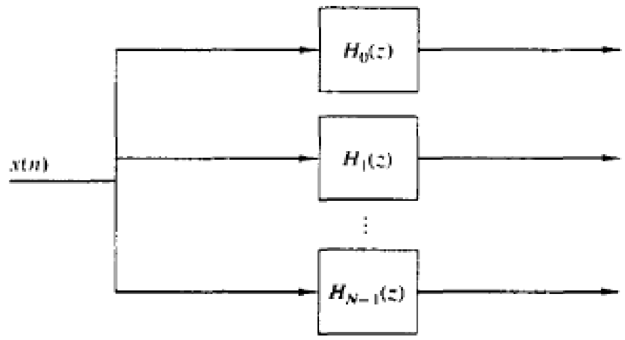
10.9.3 Implementation of Narrowband Lowpass Filters

In Section 10.6 we demonstrated that a multistage implementation of sampling-rate conversion often provides for a more efficient realization, especially when the filter specifications are very tight (e.g., a narrow passband and a narrow transition band). Under similar conditions, a lowpass, linear-phase FIR filter may be more efficiently implemented in a multistage decimator-interpolator configuration. To be more specific, we can employ a multistage implementation of a decimator of size D, followed by a multistage implementation of an interpolator of size I, where $I = D$.

10.9.4 Implementation of Digital Filter Banks

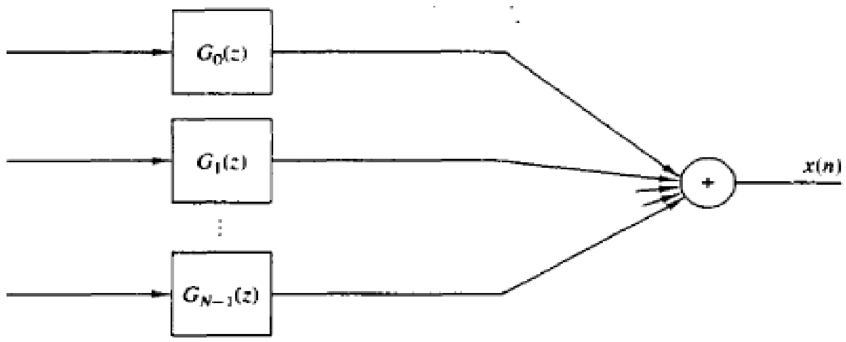
Filter banks are generally categorized as two types, *analysis filter banks* and *synthesis filter banks*. An analysis filter bank consists of a set of filters, with system functions $\{H_z(k)\}$, arranged in a parallel bank as illustrated in Fig. 10.32a. The frequency response characteristics of this filter bank splits the signal into a corresponding number of subbands. On the other hand, a synthesis filter bank consists of a set of filters with system functions $\{G_k(z)\}$, arranged as shown in Fig. 10.32b, with corresponding inputs $\{y_k(n)\}$. The outputs of the filters are summed to form the synthesized signal $\{x(n)\}$.

Filter banks are often used for performing spectrum analysis and signal synthesis. When a filter bank is employed in the computation of the discrete Fourier



Analysis filter bank

(a)



Synthesis filter bank

(b)

transform (DFT) of a sequence $\{x(n)\}$, the filter bank is called a DFT filter bank. An analysis filter bank consisting of N filters $\{H_k(z), k = 0, 1, \dots, N-1\}$ is called a uniform DFT filter bank if $H_k(z)$, $k = 1, 2, \dots, N-1$, are derived from a prototype filter $H_0(z)$, where

$$H_k(\omega) = H_0\left(\omega - \frac{2\pi k}{N}\right) \qquad k = 1, 2, \dots, N-1 \tag{10.9.2}$$

Hence the frequency response characteristics of the filters $\{H_k(z), k = 0, 1, \dots, N-1\}$ are simply obtained by uniformly shifting the frequency response of the prototype filter by multiples of $2\pi/N$. In the time domain the filters are characterized by their impulse responses, which can be expressed as

$$h_k(n) = h_0(n)e^{j2\pi nk/N} \qquad k = 0, 1, \dots, N-1 \tag{10.9.3}$$

where $\{h_0(n)\}$ is the impulse response of the prototype filter.

The uniform DFT analysis filter bank can be realized as shown in Fig. 10.33a, where the frequency components in the sequence $\{x(n)\}$ are translated in frequency to lowpass by multiplying $x(n)$ with the complex exponentials $\exp(-j2\pi nk/N)$, $k = 1, \dots, N-1$, and the resulting product signals are passed through a lowpass filter with impulse response $\{h_0(n)\}$. Since the output of the lowpass filter is relatively narrow in bandwidth, the signal can be decimated by a factor $D \le N$. The resulting decimated output signal can be expressed as

$$X_k(m) = \sum_n h_0(mD - n)x(n)e^{-j2\pi nk/N} \qquad \begin{aligned} k &= 0, 1, \dots, N-1 \\ m &= 0, 1, \dots \end{aligned} \tag{10.9.4}$$

where $\{X_k(m)\}$ are samples of the DFT at frequencies $\omega_k = 2\pi k/N$.

The corresponding synthesis filter for each element in the filter bank can be viewed as shown in Fig. 10.33b, where the input signal sequences $\{Y_k(m), k = 0, 1, \dots, N-1\}$ are upsampled by a factor of $I = D$, filtered to remove the images, and translated in frequency by multiplication by the complex exponentials $\{\exp(j2\pi nk/N), k = 0, 1, \dots, N-1\}$. The resulting frequency-translated signals from the N filters are then summed. Thus we obtain the sequence

$$\begin{aligned} v(n) &= \frac{1}{N}\sum_{k=0}^{N-1} e^{j2\pi nk/N}\left[\sum_m Y_k(m)g_0(n - mI)\right] \\ &= \sum_m g_0(n - mI)\left[\frac{1}{N}\sum_{k=0}^{N-1} Y_k(m)e^{j2\pi nk/N}\right] \\ &= \sum_m g_0(n - mI)y_n(m) \end{aligned} \tag{10.9.5}$$

where the factor $1/N$ is a normalization factor, $\{y_n(m)\}$ represent samples of the inverse DFT sequence corresponding to $\{Y_k(m)\}$, $\{g_0(n)\}$ is the impulse response of the interpolation filter, and $I = D$.

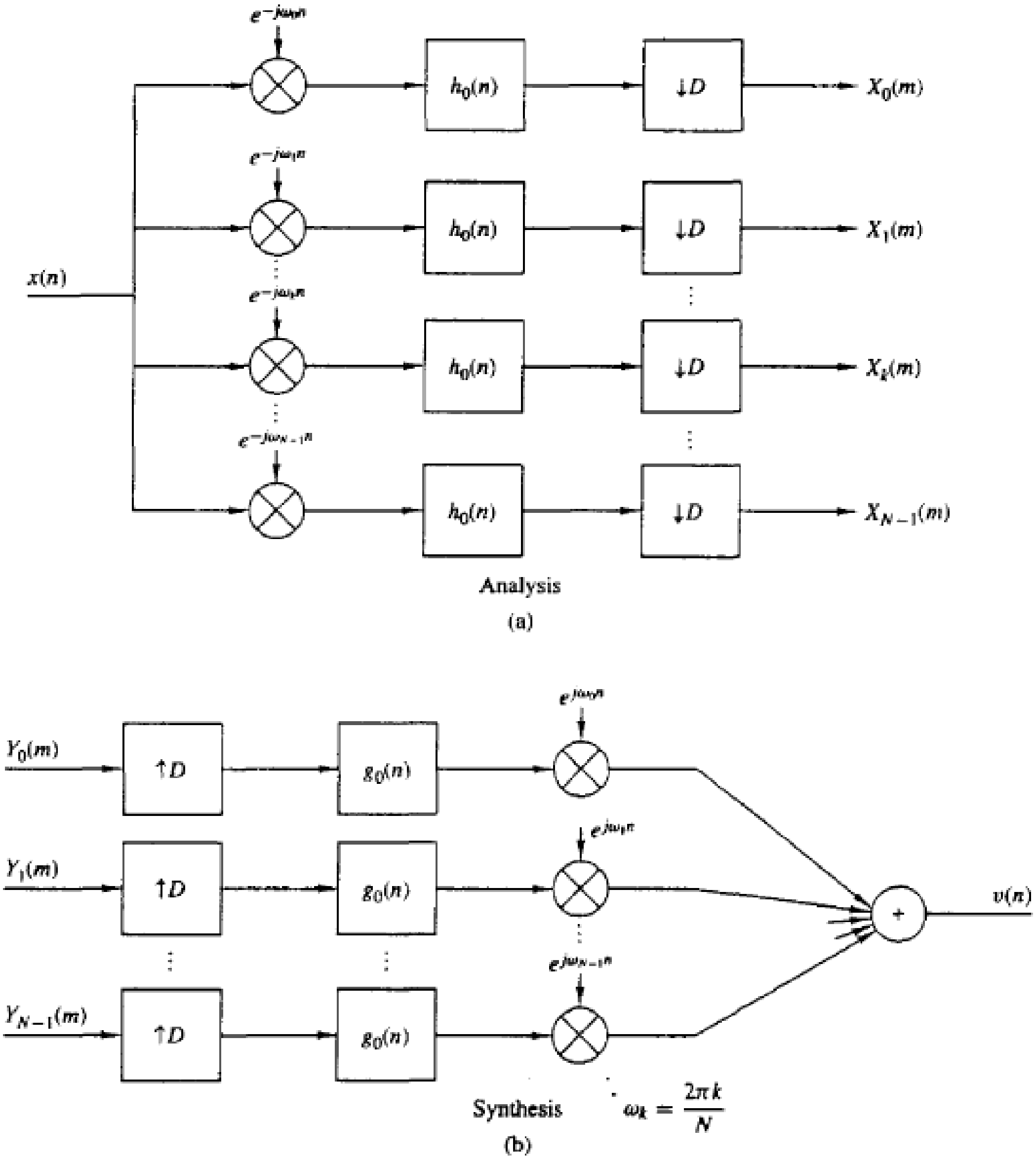


Figure 10.33 A uniform DFT filter bank.

10.9.5 Subband Coding of Speech Signals

A variety of techniques have been developed to efficiently represent speech signals in digital form for either transmission or storage. Since most of the speech energy is contained in the lower frequencies, we would like to encode the lower-frequency band with more bits than the high-frequency band. Subband coding is a method, where the speech signal is subdivided into several frequency bands and each band is digitally encoded separately.

An example of a frequency subdivision is shown in Fig. 10.37a. Let us assume that the speech signal is sampled at a rate F_s samples per second. The first frequency subdivision splits the signal spectrum into two equal-width segments, a lowpass signal ($0 \leq F \leq F_s/4$) and a highpass signal ($F_s/4 \leq F \leq F_s/2$). The second frequency subdivision splits the lowpass signal from the first stage into two equal bands, a lowpass signal ($0 < F \leq F_s/8$) and a highpass signal ($F_s/8 \leq F \leq F_s/4$). Finally, the third frequency subdivision splits the lowpass signal from the second stage into two equal bandwidth signals. Thus the signal is subdivided into four frequency bands, covering three octaves, as shown in Fig. 10.37b.

Decimation by a factor of 2 is performed after frequency subdivision. By allocating a different number of bits per sample to the signal in the four subbands, we can achieve a reduction in the bit rate of the digitalized speech signal.

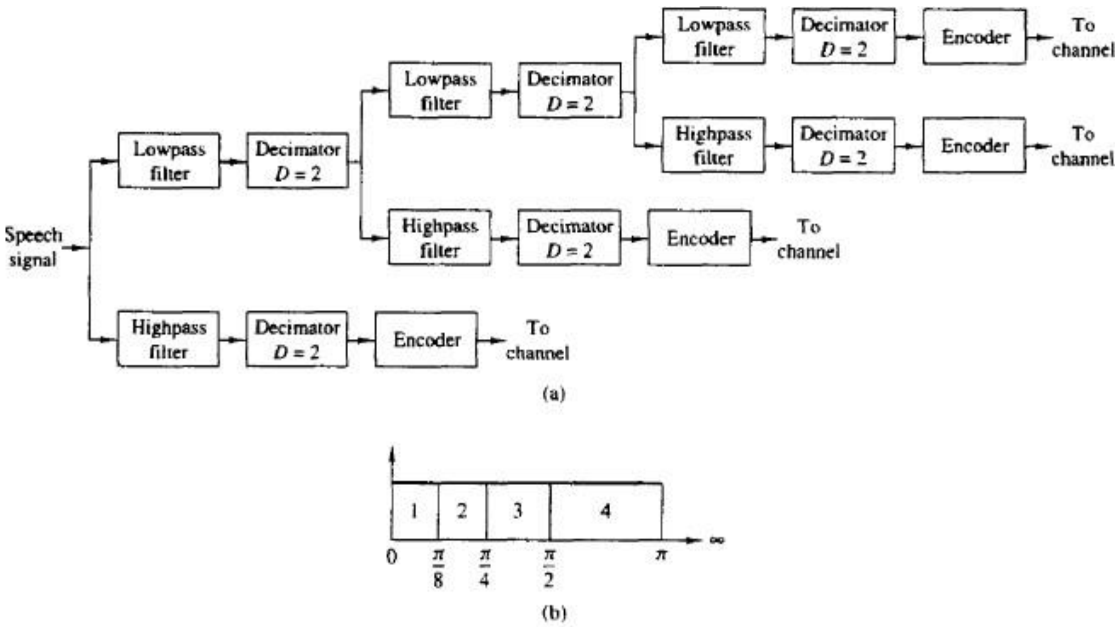


Figure 10.37 Block diagram of a subband speech coder.

Filter design is particularly important in achieving good performance in subband coding. Aliasing resulting from decimation of the subband signals must be negligible. It is clear that we cannot use brickwall filter characteristics as shown in Fig. 10.38a, since such filters are physically unrealizable. A particularly practical solution to the aliasing problem is to use *quadrature mirror filters* (QMF), which have the frequency response characteristics shown in Fig. 10.38b. These filters are described in the following section.

The synthesis method for the subband encoded speech signal is basically the reverse of the encoding process. The signals in adjacent lowpass and highpass frequency bands are interpolated, filtered, and combined as shown in Fig. 10.39. A pair of QMF is used in the signal synthesis for each octave of the signal.

Subband coding is also an effective method to achieve data compression in image signal processing. By combining subband coding with vector quantization for each subband signal, Safranek et al. (1988) have obtained coded images with approximately $\frac{1}{2}$ bit per pixel, compared with 8 bits per pixel for the uncoded image.

In general, subband coding of signals is an effective method for achieving bandwidth compression in a digital representation of the signal, when the signal energy is concentrated in a particular region of the frequency band. Multirate signal processing notions provide efficient implementations of the subband encoder.

10.9.6 Quadrature Mirror Filters

The basic building block in applications of quadrature mirror filters (QMF) is the two-channel QMF bank shown in Fig. 10.40. This is a multirate digital filter structure that employs two decimators in the "signal analysis" section and two interpolators in the "signal synthesis" section. The lowpass and highpass filters in the analysis section have impulse responses $h_0(n)$ and $h_1(n)$, respectively. Similarly, the lowpass and highpass filters contained in the synthesis section have impulse responses $g_0(n)$ and $g_1(n)$, respectively.

The Fourier transforms of the signals at the outputs of the two decimators are

$$
\begin{aligned}
X_{a0}(\omega) &= \frac{1}{2}\left[X\left(\frac{\omega}{2}\right)H_0\left(\frac{\omega}{2}\right) + X\left(\frac{\omega-2\pi}{2}\right)H_0\left(\frac{\omega-2\pi}{2}\right)\right] \\
X_{a1}(\omega) &= \frac{1}{2}\left[X\left(\frac{\omega}{2}\right)H_1\left(\frac{\omega}{2}\right) + X\left(\frac{\omega-2\pi}{2}\right)H_1\left(\frac{\omega-2\pi}{2}\right)\right]
\end{aligned}
\qquad (10.9.17)
$$

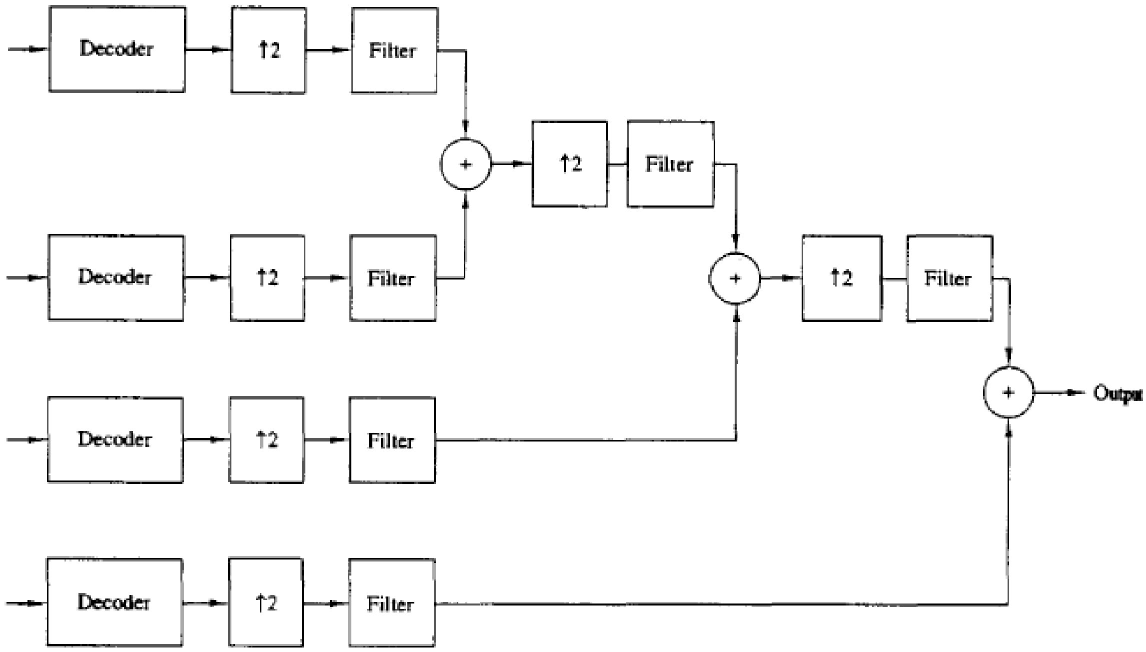


Figure 10.39 Synthesis of subband-encoded signals.

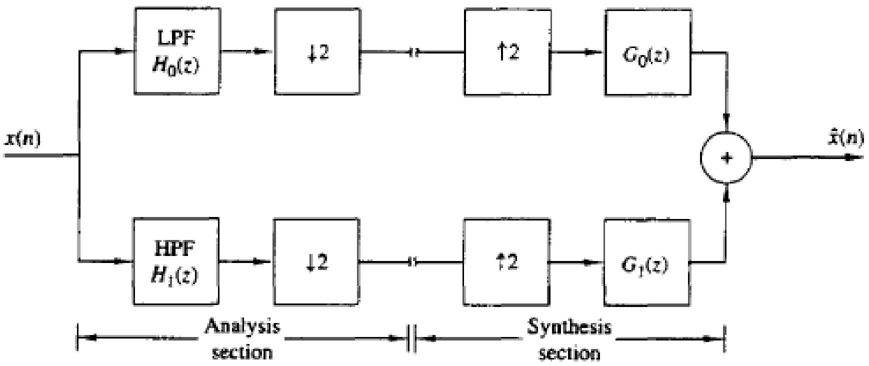


Figure 10.40 Two-channel QMF bank.

If $X_{s0}(\omega)$ and $X_{s1}(\omega)$ represent the two inputs to the synthesis section, the output is simply

$$\hat{X}(\omega) = X_{s0}(2\omega)G_0(\omega) + X_{s1}(2\omega)G_1(\omega)$$

10.9.7 Transmultiplexers

Another application of multirate signal processing is in the design and implementation of digital transmultiplexers which are devices for converting between time-division-multiplexed (TDM) signals and frequency-division-multiplexed (FDM) signals.

In a transmultiplexer for TDM-to-FDM conversion, the input signal $\{x(n)\}$ is a time-division multiplexed signal consisting of L signals, which are separated by a commutator switch. Each of these L signals are then modulated on different carrier frequencies to obtain an FDM signal for transmission. In a transmultiplexer for FDM-to-TDM conversion, the composite signal is separated by filtering into the L signal components which are then time-division multiplexed.

In telephony, single-sideband transmission is used with channels spaced at a nominal 4-kHz bandwidth. Twelve channels are usually stacked in frequency to form a basic group channel, with a bandwidth of 48 kHz. Larger bandwidth FDM signals are formed by frequency translation of multiple groups into adjacent frequency bands. We shall confine our discussion to digital transmultiplexers for 12-channel FDM and TDM signals.

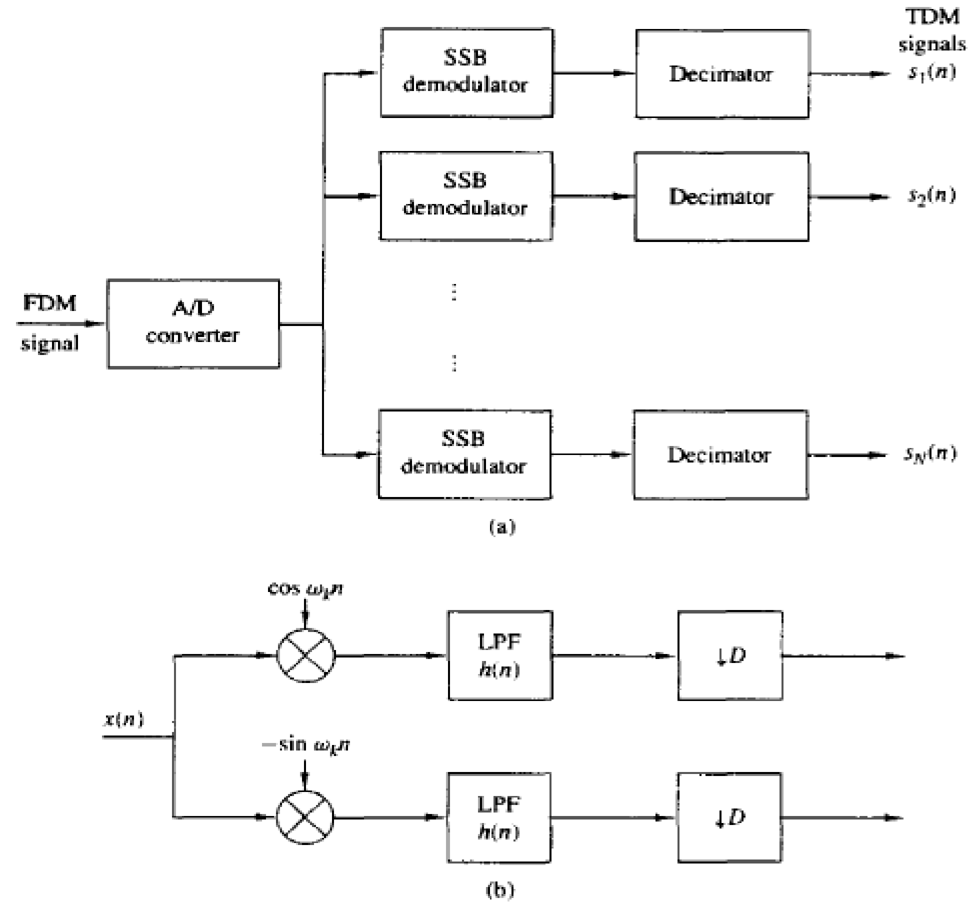


Figure 10.44 Block diagram of FDM-to-TDM transmultiplexer.

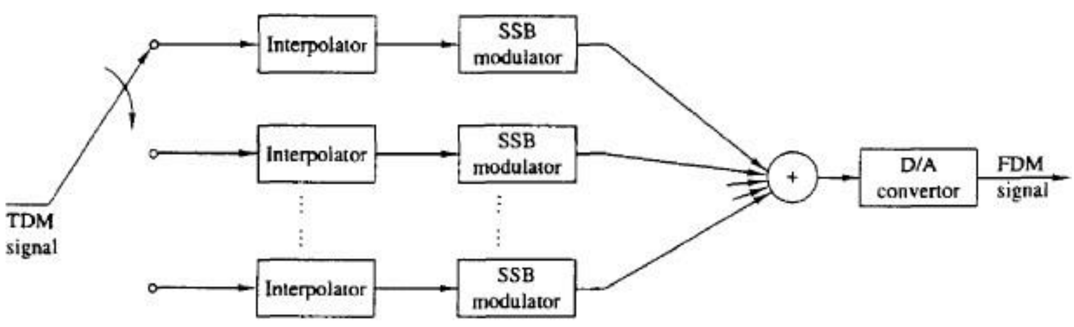


Figure 10.45 Block diagram of TDM-to-FDM transmultiplexer.

10.9.8 Oversampling A/D and D/A Conversion

Our treatment of oversampling A/D and D/A converters in Chapter 9 provides another example of multirate signal processing. Recall that an oversampling A/D converter is implemented by a cascade of an analog sigma-delta modulator (SDM) followed by a digital antialiasing decimation filter and a digital highpass filter as shown in Fig. 10.46. The analog SDM produces a 1-bit per sample output at a very high sampling rate. This 1-bit per sample output is passed through a digital lowpass filter, which provides a high-precision (multiple-bit) output that is decimated to a lower sampling rate. This output is then passed to a digital highpass filter that serves to attenuate the quantization noise at the lower frequencies.

The reverse operations take place in an oversampling D/A converter, as shown in Fig. 10.47. As illustrated in this figure, the digital signal is passed through a highpass filter whose output is fed to a digital interpolator (upsampler and anti-imaging filter). This high-sampling-rate signal is the input to the digital SDM that provides a high-sampling-rate 1-bit per sample output. The 1-bit per sample output

Figure 10.46 Diagram of oversampling A/D converter

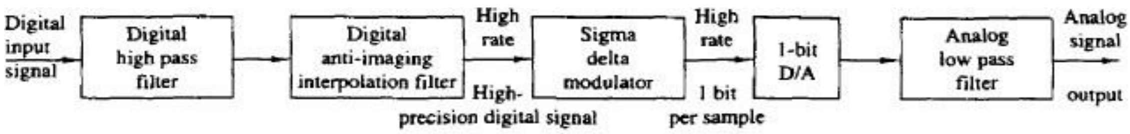


Figure 10.47 Diagram of oversampling D/A converter

is then converted to an analog signal by lowpass filtering and further smoothing with analog filters.

Figure 10.48 illustrates the block diagram of a commercial (Analog Devices ADSP-28 msp02) codec (encoder and decoder) for voice-band signals based on sigma-delta A/D and D/A converters and analog front-end circuits needed as an interface to the analog voice-band signals. The nominal sampling rate (after decimation) is 8 kHz and the sampling rate of the SDM is 1 MHz. The codec has a 65-dB SNR and harmonic distortion performance.

****