3

Multirate Processing of Digital Signals

Ronald E. Crochiere
Lawrence R. Rabiner
AT&T Bell Laboratories

3.0 INTRODUCTION

As more and more signals in the real world are represented, stored, and transmitted in digital formats, the importance of being able to process a signal digitally from its inception to its final destination grows. The general theory of digital signal processing has emerged in the past two decades and has grown to prominence in the engineering community, as evidenced by the publication of several key texts and reprint collections [1–9]. Almost all of the theory presented in those references deals with processing signals at a fixed sampling rate. However, in the past few years, there has been an increasing need for a deeper understanding of how to process digital signals in systems that require more than one sampling rate. An entire subfield of digital signal processing—multirate signal processing—has developed to meet this need [10].

This chapter presents the main ideas and concepts of multirate digital signal processing, with particular emphasis on digital techniques for changing the sampling rate of a signal. We begin our discussion with a thorough review of the Nyquist sampling theorem and its interpretations in terms of modulated signals. We then show how a continuous-time signal can be reconstructed from its digital samples. This discussion leads naturally to the topic of sampling rate conversion, in which we show that such conversion is essentially a digital "resampling" of the signal at the required rate. We show that we can readily implement such sampling rate conversion systems by paying careful attention to the structure for computation, to the digital filters that perform the antialiasing or anti-imaging functions, and to the use of cascade architectures, when appropriate. Finally, we conclude by discussing several applications of

Ronald E. Crochiere and Lawrence R. Rabiner are with AT&T Bell Laboratories, Murray Hill, NJ 07974.

sampling rate conversion to illustrate the power of the multirate signal processing techniques discussed in this chapter.

3.1 SAMPLING AND SIGNAL RECONSTRUCTION

3.1.1 The Sampling Theorem

Assume that we have a continuous-time signal $x_c(t)$, $-\infty < t < \infty$, for which we would like to obtain a discrete-time representation $x(n)$, $-\infty < n < \infty$. The signals $x(n)$ and $x_c(t)$ are related by sampling, that is,

$$x(n) = x_c(t)\big|_{t=q(n)} \tag{3.1}$$

where $q(n)$ is the "sampler" applied to $x_c(t)$. The most widely used sampler is the *uniform sampler*, in which

$$t = q(n) = nT \tag{3.2}$$

where T is the sampling period, and the quantity

$$F_s = 1/T \tag{3.3}$$

is the sampling rate. Figure 3.1 gives an illustration of uniform sampling. Part (a) of the figure depicts a typical waveform of the continuous-time signal $x_c(t)$. Parts (b) and (c) show the resulting digital signals (called $x(n)$) for two different choices of sampling period T. The most important consideration in sampling is determination of the proper

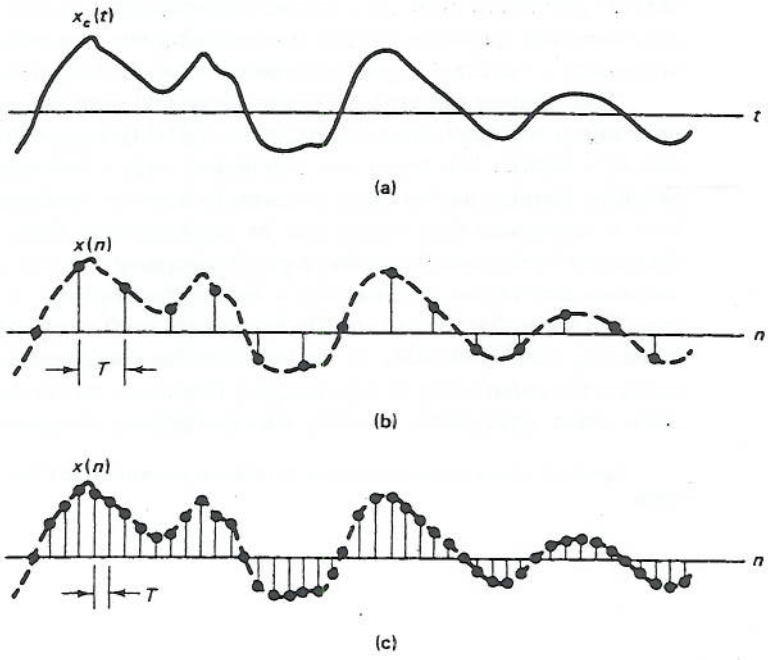


Figure 3.1 (a) Continuous-time signal and (b, c) two sampled versions of it.

value of T. We will discuss this point later in this section. First, however, we give a few simple interpretations of sampling.

The simplest interpretation of sampling is as a modulation process, as shown in Fig. 3.2(a). The continuous-time signal $x_c(t)$ (Fig. 3.2b) is first multiplied (modulated) by the sampling pulse train signal $s_c(t)$ (Fig. 3.2c), where $s_c(t)$ has the form

$$s_c(t) = \sum_{\ell=-\infty}^{\infty} \delta_c(t - \ell T) \tag{3.4}$$

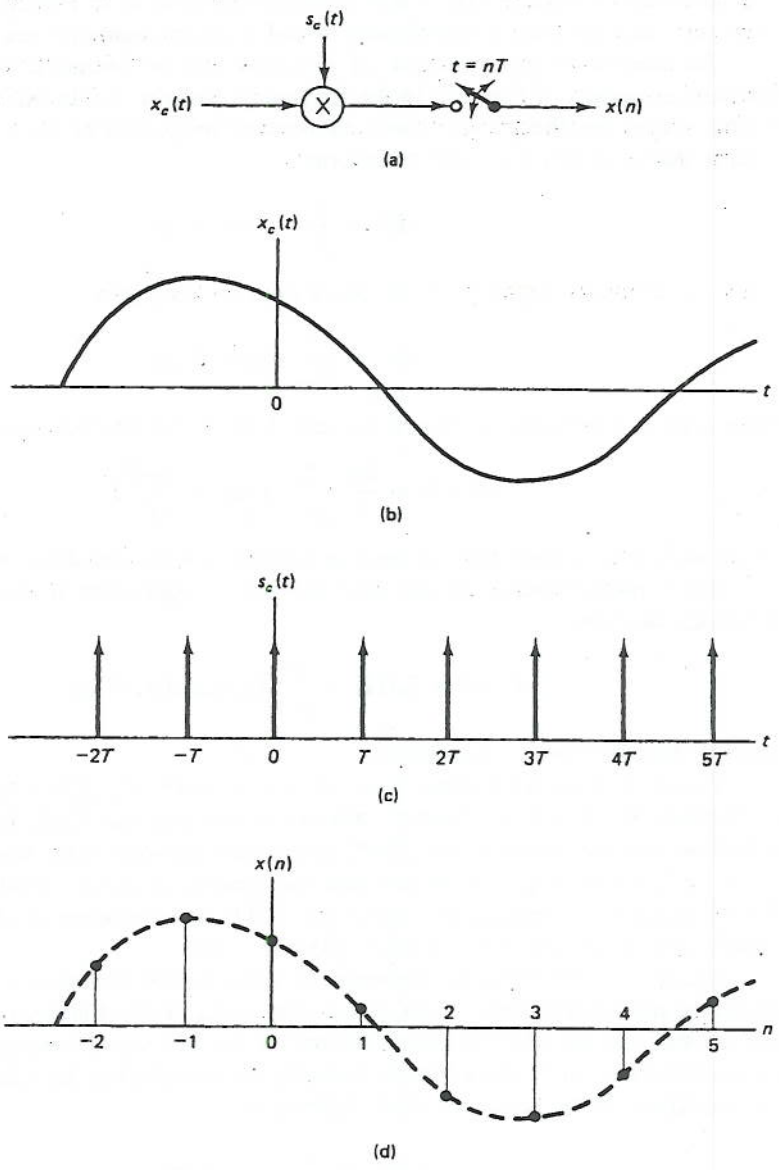


Figure 3.2 Periodic sampling of $x_c(t)$ via modulation to obtain $x(n)$.

and where $\delta_c(t)$ denotes an ideal unit impulse function. The resulting pulselike signal is measured for values of $t = nT$ via the switch, which closes for a brief instant once every T seconds and then remains open the rest of the time. The resulting digital signal $x(n)$ is shown in Fig. 3.2(d) and has the form

$$x(n) = \lim_{\epsilon \to 0} \int_{t=nT-\epsilon}^{nT+\epsilon} x_c(t)s_c(t)\, dt = x_c(nT) \tag{3.5}$$

The sampling network of Fig. 3.2(a) is often referred to as an analog-to-digital (A/D) converter since an analog signal goes in and a digital signal comes out.

An alternative interpretation of sampling can be obtained by examining the modulation system of Fig. 3.2 in the frequency domain. To do this we must rely on Fourier transform theory and some elementary properties of such transforms. We assume that $x_c(t)$ has a Fourier transform

$$X_c(\Omega) = \int_{-\infty}^{\infty} x_c(t)e^{-j\Omega t}\, dt \tag{3.6}$$

and $s_c(t)$, the sampling pulse train, has a Fourier transform

$$S_c(\Omega) = \int_{-\infty}^{\infty} s(t)\, e^{-j\Omega t}\, dt \tag{3.7}$$

Since $s_c(t)$ is a periodic impulse train (Eq. 3.4), $S_c(\Omega)$ has the form

$$S_c(\Omega) = \frac{2\pi}{T} \sum_{\ell=-\infty}^{\infty} \delta_c\left(\Omega - \frac{2\pi\ell}{T}\right) \tag{3.8}$$

i.e., a uniformly spaced impulse train in frequency with period $\Omega_F = 2\pi/T = 2\pi F_s$.

Since multiplication in the time domain is equivalent to convolution in the frequency domain,

$$X_c(\Omega) * S_c(\Omega) = \int_{-\infty}^{\infty} [x_c(t)s(t)]e^{-j\Omega t}\, dt \tag{3.9}$$

where $*$ denotes linear convolution.

Figure 3.3 shows typical plots of $X_c(\Omega)$ (part a), $S_c(\Omega)$ (part b), and the convolution $X_c(\Omega) * S_c(\Omega)$ (part c), where it is assumed that $X_c(\Omega)$ is bandlimited and its highest frequency component, $2\pi F_c$, is less than one-half of the sampling frequency $\Omega_F = 2\pi F_s$. From Fig. 3.3 we see that the process of pulse amplitude modulation (PAM) periodically repeats the spectrum $X_c(\Omega)$ at harmonics of the sampling frequency, due to the convolution of $X_c(\Omega)$ and $S_c(\Omega)$.

Because of the direct correspondence between the sequence $x(n)$ and the pulse amplitude modulated signal $x_c(t)s_c(t)$ (as seen in Eq. 3.5), it is clear that the information content and the spectral interpretations of the two signals are synonymous. This correspondence can be shown more formally by considering the (discrete-time) Fourier transform of the sequence $x(n)$, defined as

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \tag{3.10}$$

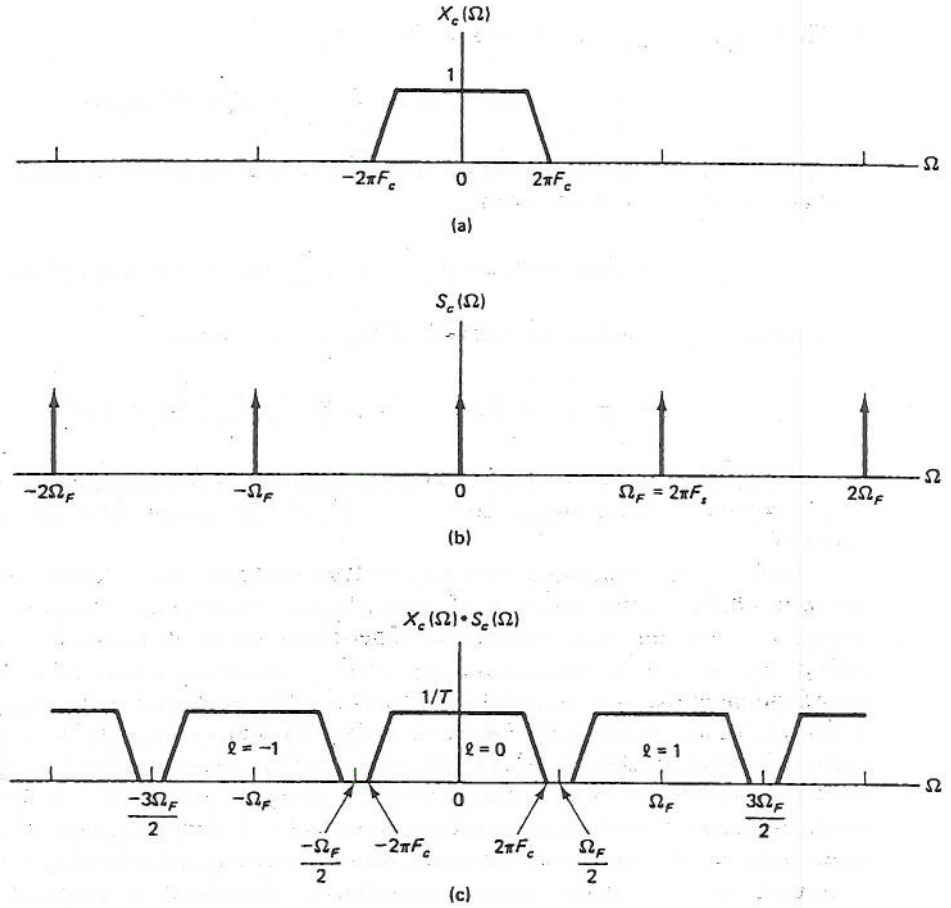


Figure 3.3 Spectra of signals obtained from periodic sampling via modulation.

where $\omega = \Omega T = \Omega/F_s$. Since $x_c(t)$ and $x(n)$ are related by Eq. (3.5), a relation can be derived between $X_c(\Omega)$ and $X(\omega)$ with the aid of Eqs. (3.6) and (3.10) as follows. The inverse Fourier transform of $X_c(\Omega)$ gives $x_c(t)$ as

$$x_c(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_c(\Omega)e^{j\Omega t}\,d\Omega \tag{3.11}$$

Evaluating Eq. (3.11) for $t = nT$, we get

$$x(n) = x_c(nT) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_c(\Omega)e^{j\Omega nT}\,d\Omega \tag{3.12}$$

The sequence $x(n)$ may also be obtained as the inverse (discrete-time) Fourier transform of $X(\omega)$:

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{j\omega n}\,d\omega \tag{3.13}$$

Thus, combining Eqs. (3.12) and (3.13), we get

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{j\omega n}\,d\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_c(\Omega)e^{j\Omega nT}\,d\Omega \tag{3.14}$$

By expressing the right-hand integral as a sum of integrals, each of width $2\pi/T$, we can put Eq. (3.14) in the form

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} [X(\omega)]e^{j\omega n}\,d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\frac{1}{T}\sum_{\ell=-\infty}^{\infty} X_c(\Omega + \ell\Omega_F)\right]e^{j\omega n}\,d\omega \tag{3.15}$$

By equating terms within the brackets of Eq. (3.15), we get

$$X(\omega) = \frac{1}{T}\sum_{\ell=-\infty}^{\infty} X_c(\Omega + \ell\Omega_F) = \frac{1}{T}\sum_{\ell=-\infty}^{\infty} X_c\left[\frac{1}{T}(\omega + 2\pi\ell)\right] \tag{3.16}$$

Equation (3.16) shows that the Fourier transform of the digital signal is the sum of frequency-shifted and scaled versions of the Fourier transform of the continuous signal.

We can now see quite clearly the effect of different choices of $\Omega_F = 2\pi/T$ on the resulting Fourier transform of the digital signal. These possibilities are illustrated in Fig. 3.4. Figure 3.4(a) shows the Fourier transform of the continuous signal and Fig. 3.4(b) shows the resulting digital Fourier transform when $\Omega_F > 4\pi F_c$. The individual terms for $\ell = 0$ and $\ell = \pm 1$ of Eq. (3.16) are shown in this figure. Figure 3.4(c) shows the resulting discrete-time Fourier transform when $\Omega_F = 4\pi F_c$. In this case the individual terms of Eq. (3.16) come right up to each other in frequency. This case is referred to as *critical sampling* of the signal. Finally, Fig. 3.4(d) shows the resulting discrete-time Fourier transform when $\Omega_F < 4\pi F_c$. In this case the individual terms in Eq. (3.16) overlap in frequency, and the resulting digital frequency response, in general, bears no simple, direct relationship to the continuous frequency response of Fig. 3.4(a). In this case we say that the digital signal is an *aliased* representation of the continuous signal.

The implications of the three cases of sampling discussed above are summarized in a simple and straightforward manner in the Nyquist sampling theorem:

> If a continuous-time signal $x_c(t)$ has a bandlimited Fourier transform $X_c(\Omega)$ that satisfies the condition $|X_c(\Omega)| = 0$ for $\Omega \geq 2\pi F_c$, then $x_c(t)$ can be *uniquely* reconstructed, without error, from equally spaced samples $x(n) = x_c(nT)$, $-\infty < n < \infty$, if $F_s \geq 2F_c$, where $F_s = 1/T$.

The sampling theorem tells us that the digital signals of Fig. 3.4(b) and (c) are suitable for exact reconstruction of the continuous-time signal from which they were obtained; however, the aliased digital signal of Fig. 3.4(d) cannot be used to reconstruct the continuous signal from which it was obtained.

There are several points worth noting about the sampling theorem and its practical implementation. The first point concerns the bandlimited requirement on $x_c(t)$. In practice there are no truly bandlimited signals for which $|X_c(\Omega)| = 0$ for $\Omega \geq 2\pi F_c$. Thus we substitute the more reasonable bandlimited criterion $|X_c(\Omega)| \leq \delta$ for $\Omega \geq 2\pi F_c$, where δ is some suitably small constant. For example, if the largest

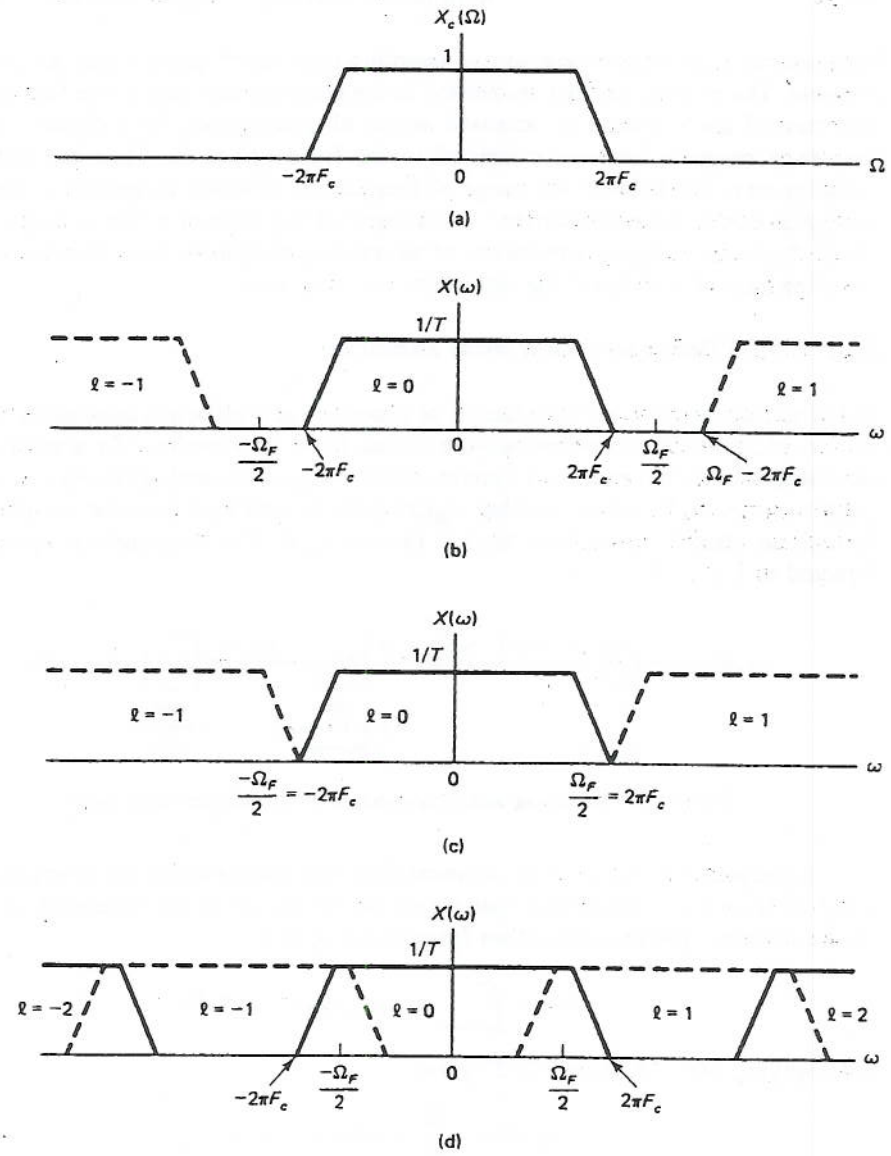


Figure 3.4 Spectral interpretations of the sampling theorem.

magnitude of $X_c(\Omega)$ is on the order of 1.0, values of δ from 0.0001 (80 dB down) to 0.01 (40 dB down) are generally used.

A second point of note in the practical use of the sampling theorem is that, in general, we are using a sampling rate F_s as close to the critical rate $2F_c$ as possible. In this manner the computation of the digital signal processing is kept as small as possible, consistent with the sampling requirements.

A final point about the sampling theorem concerns cases that violate the rule to keep the sampling rate as low as possible. For digital processing systems with non-

linearities it is often advisable to use sampling rates much greater than the minimum possible. The reasons for this are related to the nonlinearities that cause the frequency spectrum of the signal to be smeared across all frequencies. In a digital signal the frequency range is inherently limited to the band $|f| \leq F_c$. Thus the higher the sampling rate, the broader the range of frequencies in which to spread the energy of the signal due to the nonlinearity. An example of this type of effect is given in [14], which discusses a digital simulation of an ordinary telephone line. For this system a sampling rate of five times the minimum rate was used.

3.1.2 Signal Reconstruction from Samples

The sampling theorem gives us the set of necessary and sufficient conditions for being able to reconstruct a continuous-time signal from its samples. To actually do the reconstruction we have to use a system, called a digital-to-analog (D/A) converter, to transform $x(n)$ back to the sampled signal $x_c(t)s_c(t)$ and then pass the sampled signal through an ideal lowpass filter $h_I(t)$ to recover $x_c(t)$. This sequence of operations is depicted in Fig. 3.5.

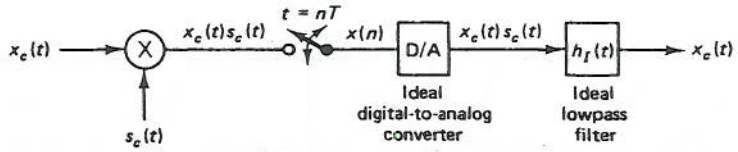


Figure 3.5 Sampling and reconstruction of a continuous-time signal.

To implement this process, an ideal digital-to-analog converter is required to get $x_c(t)s_c(t)$ from $x(n)$. Assuming that we do not worry about the reliability of such an ideal converter, the reconstruction formula for $x_c(t)$ is

$$x_c(t) = \int_{\tau=-\infty}^{\infty} x_c(\tau)s_c(\tau)h_I(t-\tau)\,d\tau \tag{3.17}$$

and applying Eqs. (3.4) and (3.5) gives

$$x_c(t) = \sum_{n=-\infty}^{\infty} x(n)h_I(t-nT) \tag{3.18}$$

If we use the reconstruction method, the relationship between the reconstructed signal $x_c(t)$ and the set of samples $x(n)$ is given by the convolutional formula

$$x_c(t) = \sum_{n=-\infty}^{\infty} x(n)h_I(t-nT) \tag{3.19}$$

The ideal lowpass filter $h_I(t)$ has the frequency-domain characteristics

$$|H_I(\Omega)| = \begin{cases} 1, & 0 \leq |\Omega| \leq \pi F_s = \pi/T \\ 0, & |\Omega| > \pi F_s = \pi/T \end{cases} \tag{3.20}$$

giving the ideal impulse response

$$h_I(t) = \frac{\sin(\pi t/T)}{(\pi t/T)}, \qquad -\infty < t < \infty \tag{3.21}$$

Combining Eqs. (3.19)–(3.21) gives the ideal reconstruction formula

$$x_c(t) = \sum_{n=-\infty}^{\infty} x(n)\left[\frac{\sin[\pi(t - nT)]/T}{\pi(t - nT)/T}\right] \tag{3.22}$$

Figure 3.6 illustrates the application of the reconstruction formula of Eq. (3.22). At each instant n_0 the digital sample $x(n_0)$ weights the ideal lowpass filter response. All such terms are then summed to give $x_c(t)$.

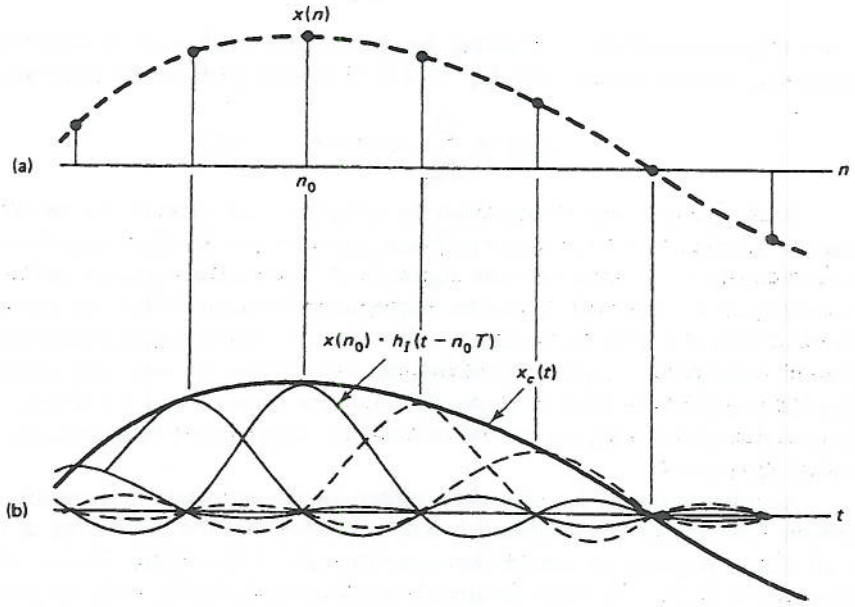


Figure 3.6 Illustration of a bandlimited reconstruction from shifted and scaled lowpass filter responses.

Unfortunately, the reconstruction formula of Eq. (3.22) cannot be implemented because the ideal lowpass filter is unrealizable. As such, in practical systems, the ideal lowpass filter is replaced with a real desampling filter giving a reasonable approximation to the desired response.

3.2 SAMPLING RATE CONVERSION

Once we have a signal in digital form, it is convenient to be able to process it digitally until it reaches its final destination and must ultimately be converted back to analog (continuous-time) form. When the sampling rate required in the digital processing remains constant, there is no problem in handling the digital signal. However, in many

practical systems, the sampling rates at different points in the system are different. Hence we must be able to change freely the sampling rate of a digital signal for maximum signal processing efficiency.

There is one very simple and straightforward approach to changing the sampling rate of a digital signal. This approach, called the analog approach, merely reconstructs the continuous-time signal from the original set of samples and then resamples the signal at the new rate (assuming no additional antialiasing lowpass filtering is required). Thus, using Eq. (3.19), if the original sampling period is T and if the new sampling period is T', we form

$$y(m) = x_c(t)\big|_{t=mT'} = \sum_{n=-\infty}^{\infty} x(n)h_l(mT' - nT) \tag{3.23}$$

If we restrict ourselves to practical lowpass filters (with a finite-duration impulse response), then we can rewrite Eq. (3.23) to express $y(m)$ as the finite summation

$$y(m) = \sum_{n=N_1}^{N_2} x(n)h_l(mT' - nT) \tag{3.24}$$

In theory the analog approach to sampling rate conversion works well. In practice it suffers from one major problem, namely, that the ideal operations required to reconstruct the continuous-time signal from the original samples and to resample the signal at the new rate cannot be implemented exactly. When we resort to using practical D/A and A/D converters, we find that the resulting signal has additive noise (due to resampling), signal-dependent distortions (due to nonideal samplers), and frequency distortions (due to nonideal frequency responses of the filters). Although these real-world distortions can be minimized, they cannot be eliminated using the analog approach.

An alternative is the so-called direct digital approach to sampling rate conversion. The key to this approach is the realization that the processing of Eq. (3.24) is all that is required to change the sampling rate of the signal. Hence, if the computations of Eq. (3.24) could be directly implemented, that is, without going through D/A or A/D conversion, in theory a distortion-free sampling rate conversion could be achieved. We now show how this is achieved in practice. Before presenting the general equations for the digital approach to sampling rate conversion, we first discuss three special cases that essentially provide a complete understanding of the nature of the digital approach:

1. Decimation (sampling rate decrease) by an integer factor M
2. Interpolation (sampling rate increase) by an integer factor L
3. Sampling rate conversion by a ratio of integer factors M/L

3.2.1 Decimation by an Integer Factor *M*

Consider the process of reducing the sampling rate (decimation) of $x(n)$ by an integer factor M. If we denote the sampling period and rate of $x(n)$ by T and $F_s = 1/T$ and

the sampling period and rate of the decimated signal $y(n)$ by T' and $F_s' = 1/T'$, then we have

$$\frac{T'}{T} = M = \frac{F_s}{F_s'} \tag{3.25}$$

Assume that $x(n)$ represents a fullband signal, i.e., its spectrum is nonzero for all frequencies in the range $|f| \leq F_s/2$, except possibly at an isolated set of points. Based on the analog interpretation of sampling, we see that to lower the sampling rate and to avoid aliasing at this lower rate, it is necessary to filter the signal $x(n)$ with a *digital* lowpass filter that approximates the ideal characteristic

$$H_I(\omega) = \begin{cases} 1, & |\omega| < \dfrac{\pi}{M} \\ 0, & \text{otherwise} \end{cases} \tag{3.26}$$

The sampling rate reduction is then achieved by forming the sequence $y(m)$ by saving only every Mth sample of the filtered output. This process is illustrated in Fig. 3.7(a). If we denote the actual lowpass filter impulse response as $h(n)$, then we have

$$w(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{3.27}$$

where $w(n)$ is the filtered output as seen in Fig. 3.7(a). The final output, $y(m)$, is then obtained as

$$y(m) = w(Mm) \tag{3.28}$$

as denoted by the operation of the second box in Fig. 3.7(a). The use of a down arrow followed by an integer is called a sampling rate compressor and corresponds to the resampling operation of Eq. (3.28).

Figure 3.7(b) shows typical spectra (magnitude of the discrete Fourier transforms) of the signals $x(n)$, $h(n)$, $w(n)$, and $y(m)$ for an M-to-1 reduction in sampling rate. Note that the frequencies $\omega = 2\pi fT$ and $\omega' = 2\pi f'T'$ are normalized with respect to the sampling frequencies F_s and F_s'.

When Eqs. (3.27) and (3.28) are combined, the relation between $y(m)$ and $x(n)$ is of the form

$$y(m) = \sum_{k=-\infty}^{\infty} h(k)x(Mm-k) \tag{3.29}$$

or, by a change of variables, it becomes

$$y(m) = \sum_{n=-\infty}^{\infty} h(Mm-n)x(n) \tag{3.30}$$

It is valuable to derive the relationship between the z-transforms of $y(m)$ and $x(n)$ so as to be able to study the nature of the errors in $y(m)$ caused by a practical (nonideal) lowpass filter. To obtain this relationship, we define the signal

$$w'(n) = \begin{cases} w(n), & n = 0, \pm M, \pm 2M, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{3.31}$$

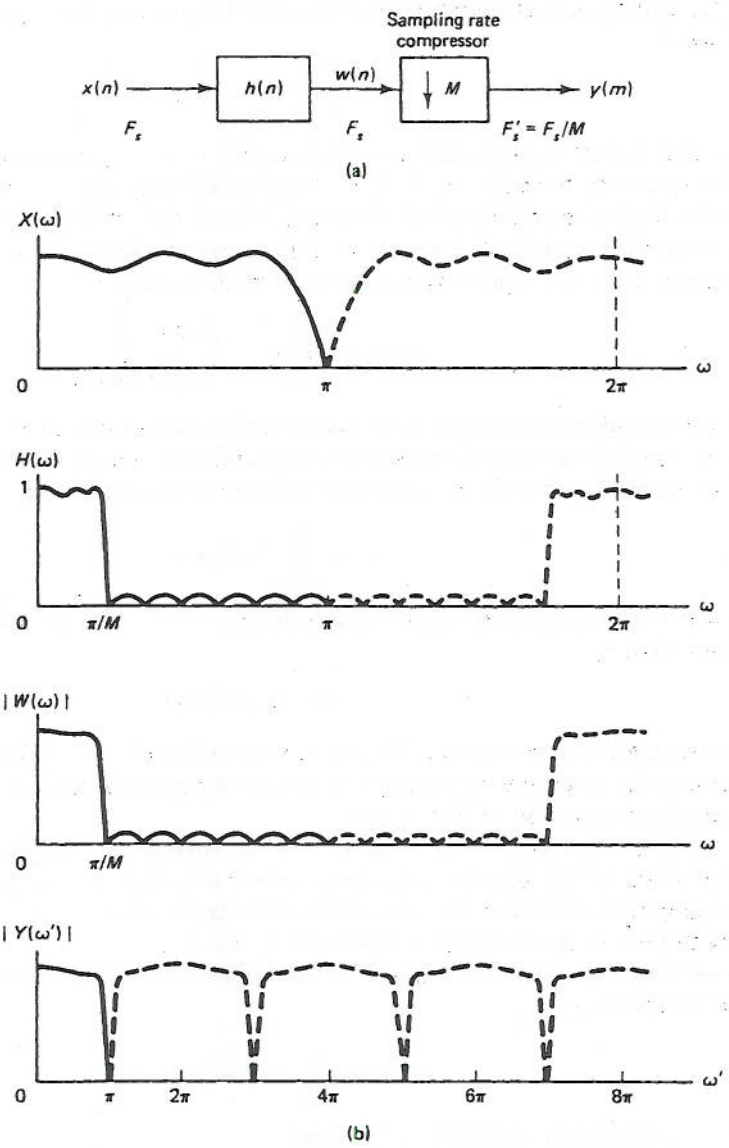


Figure 3.7 Block diagram and typical spectra for decimation by an integer factor M.

i.e., $w'(n) = w(n)$ at the sampling instants of $y(m)$ but is zero otherwise. A convenient and useful representation of $w'(n)$ is then

$$w'(n) = w(n)\left[\frac{1}{M}\sum_{\ell=0}^{M-1} e^{j(2\pi\ell n/M)}\right], \quad -\infty < n < \infty \tag{3.32}$$

where the term in brackets corresponds to a discrete Fourier series representation of a periodic impulse train with a period of M samples. (This pulse train is the digital

equivalent of the analog pulse amplitude modulated sampling function of Section 3.1.) Thus we have

$$y(m) = w'(Mm) = w(Mm) \tag{3.33}$$

We now write the z-transform of $y(m)$ as

$$Y(z) = \sum_{m=-\infty}^{\infty} y(m)z^{-m} = \sum_{m=-\infty}^{\infty} w'(Mm)z^{-m} \tag{3.34}$$

and since $w'(n)$ is zero, except at integer multiples of M, Eq. (3.34) becomes (after some simple manipulations)

$$Y(z) = \sum_{m=-\infty}^{\infty} w'(m)z^{-m/M} = \frac{1}{M}\sum_{\ell=0}^{M-1} W(e^{-j2\pi\ell/M}\, z^{\ell/M}) \tag{3.35}$$

Since

$$W(z) = H(z)X(z) \tag{3.36}$$

we can express $Y(z)$ as

$$Y(z) = \frac{1}{M}\sum_{\ell=0}^{M-1} H(e^{-j2\pi\ell/M}\, z^{1/M})X(e^{-j2\pi\ell/M}\, z^{1/M}) \tag{3.37}$$

Evaluating $Y(z)$ on the unit circle, $z = e^{j\omega'}$, leads to

$$Y(\omega') = \frac{1}{M}\sum_{\ell=0}^{M-1} H((\omega' - 2\pi\ell)/M)X((\omega' - 2\pi\ell)/M) \tag{3.38}$$

Equation (3.38) expresses the Fourier transform of the output signal $y(m)$ in terms of the transform of the aliased components of the filtered input signal $x(n)$. By writing out the individual terms, we get

$$Y(\omega') = \frac{1}{M}[H(\omega'/M)X(\omega'/M) + H((\omega' - 2\pi)/M)X((\omega' - 2\pi)/M) + \cdots] \tag{3.39}$$

The purpose of the lowpass filter, $H(\omega)$, is to filter $x(n)$ sufficiently so that its spectral components above the frequency $\omega = \pi/M$ are negligible (see Fig. 3.7). Thus it serves as an antialiasing filter. If the lowpass filter sufficiently removes all energy of $x(n)$ above the frequency $\omega = \pi/M$ (i.e., terms of Eq. 3.38 with $\ell \neq 0$) and if the filter $H(\omega)$ closely approximates the ideal response of Eq. (3.26), then Eq. (3.39) becomes

$$Y(\omega) = \frac{1}{M}X(\omega'/M), \qquad |\omega'| < \pi \tag{3.40}$$

i.e., the desired resampled signal.

One of the most interesting properties of the decimation system can be seen by comparing Eq. (3.38) and Eq. (3.16). Both equations express the frequency-domain content of the digital signal in terms of sums of shifted components of the frequency

contents of the input signal to the system. Hence the decimation system is truly a digital resampling of a signal with potential aliasing components. Much as with the sampling theorem, we must keep the level of the digital aliasing components low by applying a properly designed digital antialiasing lowpass filter.

3.2.2 Interpolation by an Integer Factor L

Consider the process of increasing the sampling rate (interpolation) of $x(n)$ by an integer factor L. If we again denote the sampling period and rate of $x(n)$ by T and $F_s = 1/T$, and the sampling period and rate of the interpolated signal $y(n)$ by T' and $F_s' = 1/T'$, then we have

$$\frac{T'}{T} = \frac{1}{L} = \frac{F_s}{F_s'} \tag{3.41}$$

This process of increasing the sampling rate (interpolation) of a signal $x(n)$ by the integer factor L implies that we must interpolate $L - 1$ new sample values between each pair of sample values of $x(n)$. The process is similar to that of digital-to-analog conversion in which all continuous-time values of a signal $x_c(t)$ must be interpolated from its sequence $x(n)$. (In this case only specific values must be determined.)

Figure 3.8 illustrates an example of interpolation by a factor $L = 3$. The input signal $x(n)$ is "filled in" with $L - 1$ zero-valued samples between each pair of samples of $x(n)$, giving the signal

$$w(m) = \begin{cases} x(m/L), & m = 0, \pm L, \pm 2L, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{3.42}$$

This process is the digital equivalent to the digital-to-PAM conversion process discussed in Section 3.1.2, and it is illustrated by the first box in the block diagram of Fig. 3.8(a). As with the resampling operation, the block diagram symbol of an up arrow with an integer corresponds to increasing the sampling rate as given by Eq. (3.42), and it will be referred to as a *sampling rate expander*. The resulting signal $w(n)$ has the z-transform

$$W(z) = \sum_{m=-\infty}^{\infty} w(m)z^{-m} \tag{3.43a}$$

$$= \sum_{m=-\infty}^{\infty} x(m)z^{-mL} \tag{3.43b}$$

$$= X(z^L) \tag{3.43c}$$

Evaluating $W(z)$ on the unit circle, $z = e^{j\omega'}$, gives the result

$$W(\omega') = X(\omega' L) \tag{3.44}$$

which is the Fourier transform of the signal $w(m)$ expressed in terms of the spectrum of the input signal $x(n)$ (where $\omega' = 2\pi f T'$ and $\omega = 2\pi f T$).

As illustrated by the spectral interpretation in Fig. 3.8(c), the spectrum of $w(m)$ contains not only the baseband frequencies of interest (i.e., $-\pi/L$ to π/L) but also *images* of the baseband centered at harmonics of the original sampling frequency $\pm 2\pi/L$, $\pm 4\pi/L$, ... To recover the baseband signal of interest and eliminate the

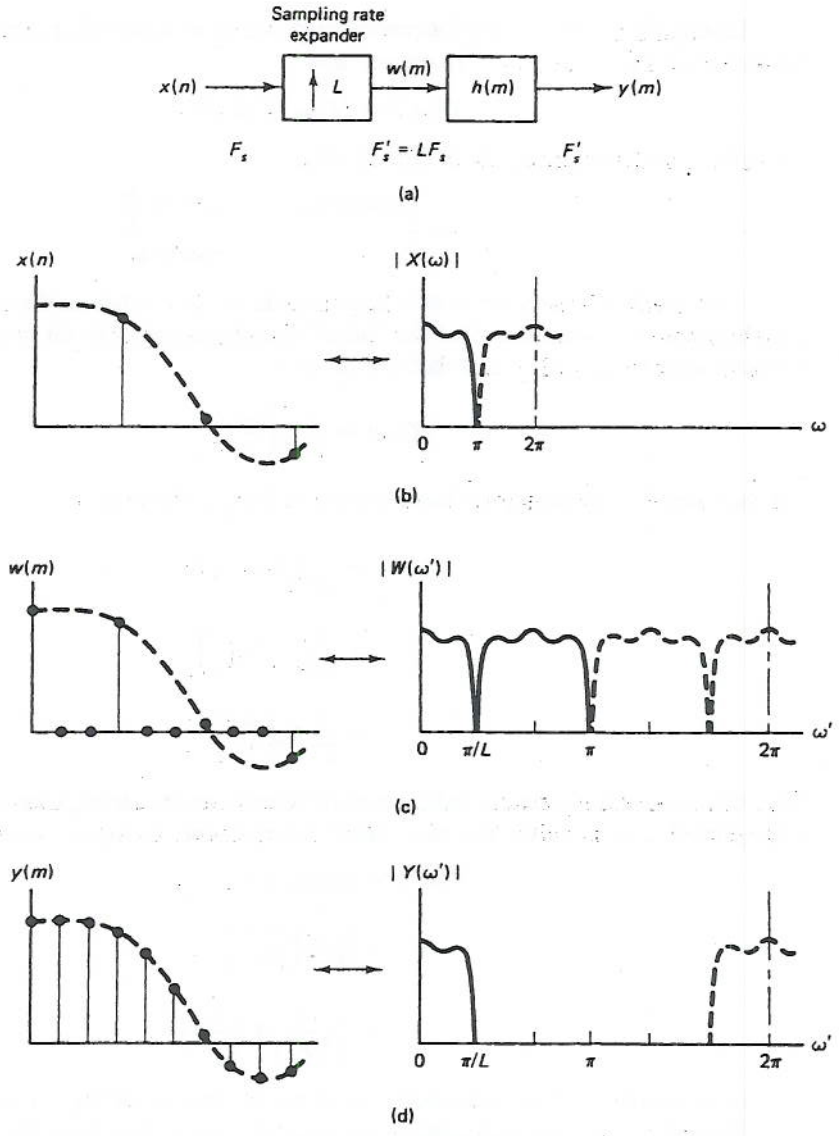


Figure 3.8 Block diagram and typical waveforms and spectra for interpolation by an integer factor L.

unwanted image components, it is necessary to filter the signal $w(m)$ with a digital lowpass (anti-imaging) filter that approximates the ideal characteristic

$$H_I(\omega') = \begin{cases} G, & |\omega'| \le \dfrac{2\pi F_s T'}{2} = \dfrac{\pi}{L} \\ 0, & \text{otherwise} \end{cases} \tag{3.45}$$

It will be shown that to ensure that the amplitude of $y(m)$ is correct, the gain of the filter, G, must be L in the passband.

Letting $H(\omega')$ denote the frequency response of an actual filter that approximates the characteristic in Eq. (3.45), we see that

$$Y(\omega') = H(\omega')X(\omega'L) \tag{3.46}$$

and within the approximation of Eq. (3.45),

$$Y(\omega') \approx \begin{cases} GX(\omega'L), & |\omega'| \le \dfrac{\pi}{L} \\ 0, & \text{otherwise} \end{cases} \tag{3.47}$$

It is easy to see why we need a gain of G in $H_I(\omega')$, whereas for the decimation filter a gain of 1 is adequate. For the "ideal" sampling system (with no aliasing error) we have seen from Eq. (3.16) that we desire

$$X(\omega) = \frac{1}{T}X_c\left(\frac{\omega}{T}\right)$$

For the "ideal" decimator, we have shown in Eq. (3.40) that

$$\begin{aligned} Y(\omega') &= \frac{1}{M}X(\omega'/M) \\ &= \frac{1}{MT}X_c\left(\frac{\omega'}{MT}\right) \\ &= \frac{1}{T'}X_c\left(\frac{\omega'}{T'}\right) \end{aligned}$$

Thus the necessary scaling is taken care of directly in the decimation process, and a filter gain of 1 is suitable. For the "ideal" interpolator, however, we have

$$\begin{aligned} Y(\omega') &= GX(\omega'L) \\ &= \frac{G}{T}X_c\left(\frac{\omega'L}{T}\right) \\ &= \frac{G}{L}\left(\frac{1}{T'}\right)X_c\left(\frac{\omega'}{T'}\right) \end{aligned}$$

Clearly, a gain $G = L$ is required to meet the conditions of Eq. (3.16).

If $h(m)$ denotes the unit sample response of $H(\omega')$, then from Fig. 3.8, $y(m)$ can be expressed as

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-k)w(k) \tag{3.48}$$

Combining Eqs. (3.42) and (3.48) leads to the time-domain input-to-output relation of the interpolator

$$\begin{aligned} y(m) &= \sum_{k=-\infty}^{\infty} h(m-k)x(k/L), \qquad k/L \text{ an integer} \\ &= \sum_{r=-\infty}^{\infty} h(m-rL)x(r) \end{aligned} \tag{3.49}$$

An alternative formulation of this equation can be obtained by introducing the change of variables

$$r = \left\lfloor \frac{m}{L} \right\rfloor - n \tag{3.50}$$

where $\lfloor u \rfloor$ denotes the integer less than or equal to u. Then by noting that

$$m - \left\lfloor \frac{m}{L} \right\rfloor L = ((m))_L \tag{3.51}$$

where $((m))_L$ denotes m modulo L.

$$\begin{aligned} y(m) &= \sum_{n=-\infty}^{\infty} h\left(m - \left\lfloor \frac{m}{L} \right\rfloor L + nL\right) x\left(\left\lfloor \frac{m}{L} \right\rfloor - n\right) \\ &= \sum_{n=-\infty}^{\infty} h(nL + ((m))_L) x\left(\left\lfloor \frac{m}{L} \right\rfloor - n\right) \end{aligned} \tag{3.52}$$

Equation (3.52) expresses the output $y(m)$ in terms of the input $x(n)$ and the filter coefficients $h(m)$.

3.2.3 Sampling Rate Conversion by a Rational Factor M/L

In Sections 3.2.1 and 3.2.2 we have considered the cases of decimation by an integer factor M and interpolation by an integer factor L. In this section we consider the general case of conversion by the ratio

$$\frac{T'}{T} = \frac{M}{L} = \frac{F_s}{F_s'} \tag{3.53}$$

This conversion can be achieved by a cascade of the two processes of integer conversions above by first increasing the sampling rate by L and then decreasing it by M. Figure 3.9(a) illustrates this process. It is important to recognize that the interpolation by L *must* precede the decimation process by M so that the width of the baseband of the intermediate signal $s(k)$ is greater than or equal to the width of the basebands of $x(n)$ or $y(m)$.

It can be seen from Fig. 3.9(a) that the two filters $h_1(k)$ and $h_2(k)$ are operating in cascade at the same sampling rate LF_s. Thus a more efficient implementation of the overall process can be achieved if the filters are combined into one composite lowpass filter, as shown in Fig. 3.9(b). Since this digital filter, $h(k)$, must serve the purposes of both the decimation and interpolation operations described in the preceding two sections, it is clear from Eqs. (3.26) and (3.45) that it must approximate the ideal digital lowpass characteristic

$$H_I(\omega'') = \begin{cases} L, & |\omega''| \le \min\left|\frac{\pi}{L}, \frac{\pi}{M}\right| \\ 0, & \text{otherwise} \end{cases} \tag{3.54}$$

where

$$\omega'' = 2\pi f T'' = 2\pi f \frac{T}{L} \tag{3.55}$$

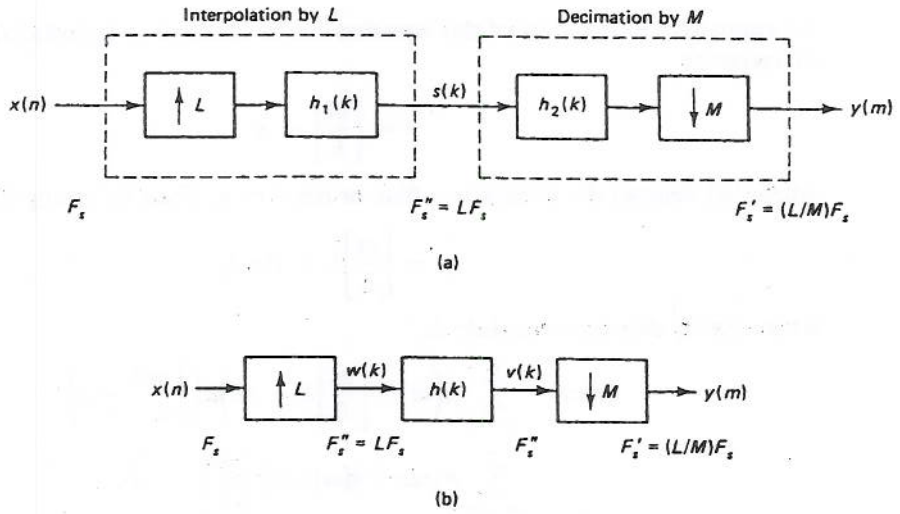


Figure 3.9 (a) Cascade of an integer interpolator and an integer decimator for achieving sampling rate changes by rational fractions; (b) a more efficient implementation of this process.

That is, the ideal cutoff frequency must be the minimum of the two cutoff frequency requirements for the decimator and interpolator, and the sampling rate of the filters is $F_s'' = LF_s$.

The time-domain input-to-output relation for the general conversion circuit of Fig. 3.9(b) can be derived by considering the integer interpolation and decimation relations derived in Sections 3.2.1 and 3.2.2; that is, from Eq. (3.49) it can be seen that $v(k)$ can be expressed as

$$v(k) = \sum_{r=-\infty}^{\infty} h(k - rL)x(r) \tag{3.56}$$

and from Eq. (3.28) $y(m)$ can be expressed in terms of $v(k)$ as

$$y(m) = v(Mm) \tag{3.57}$$

Combining Eqs. (3.56) and (3.57) gives the desired result

$$y(m) = \sum_{r=-\infty}^{\infty} h(Mm - rL)x(r) \tag{3.58}$$

Alternatively, by making the change of variables

$$r = \left\lfloor \frac{mM}{L} \right\rfloor - n \tag{3.59}$$

and using the relation

$$mM - \left\lfloor \frac{mM}{L} \right\rfloor L = ((mM))_L \tag{3.60}$$

we get

$$y(m) = \sum_{n=-\infty}^{\infty} h\left(Mm - \left\lfloor \frac{mM}{L} \right\rfloor L + nL\right) x\left(\left\lfloor \frac{mM}{L} \right\rfloor - n\right)$$
$$= \sum_{n=-\infty}^{\infty} h[nL + ((mM))_L] x\left(\left\lfloor \frac{mM}{L} \right\rfloor - n\right) \tag{3.61}$$

Similarly, by considering the transform relationships of the individual integer decimation and interpolation systems, the output spectrum $Y(\omega')$ can be determined in terms of the input spectrum $X(\omega)$ and the frequency response of the filter $H(\omega'')$. From Eq. (3.46) we see that $V(\omega'')$ can be expressed in terms of $X(\omega)$ and $H(\omega'')$ as

$$V(\omega'') = H(\omega'')X(\omega''L) \tag{3.62}$$

and from Eq. (3.35) $Y(\omega')$ can be expressed in terms of $V(\omega'')$ as

$$Y(\omega') = \frac{1}{M} \sum_{\ell=0}^{M-1} V((\omega' - 2\pi\ell)/M)$$
$$= \frac{1}{M} \sum_{\ell=0}^{M-1} H((\omega' - 2\pi\ell)/M) X((\omega' L - 2\pi\ell)/M) \tag{3.63}$$

When $H(\omega'')$ closely approximates the ideal characteristic of Eq. (3.54), we see that this expression reduces to

$$Y(\omega') \approx \begin{cases} \frac{L}{M} X(\omega' L/M), & \text{for } |\omega'| \le \min\left[\pi, \pi\frac{M}{L}\right] \\ 0, & \text{otherwise} \end{cases} \tag{3.64}$$

3.2.4 General Form of Digital Sampling Rate Conversion

It is possible to generalize the discussion of Sections 3.2.1–3.2.3 so as to give a canonic form of a digital system for sampling rate conversion. The form of this canonic system is illustrated in Fig. 3.10, in which an input signal $x(n)$, sampled at rate $F_s = 1/T$, is sent to a linear, time-varying, digital system with impulse response $g_m(n)$ to give the output signal $y(m)$, with new sampling rate $F_s' = 1/T'$. If we assume

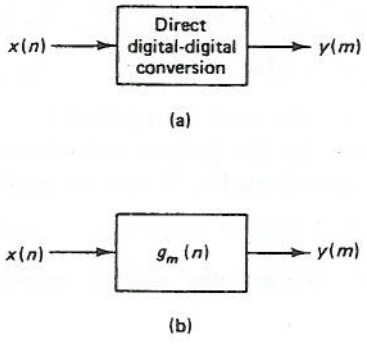


Figure 3.10 (a) Direct digital conversion of $x(n)$ to $y(m)$ and (b) a time-varying filter interpretation of the process.

that the ratio of sampling periods of $y(m)$ and $x(n)$ can be expressed as the rational fraction

$$\frac{T'}{T} = \frac{F_s}{F_s'} = \frac{M}{L} \tag{3.65}$$

with M and L integers, then the digital system response $g_m(n)$ becomes the response at output sample time m to an input at sample time $\lfloor mM/L \rfloor - n$, where $[u]$ again denotes the integer less than or equal to u.

Since the system is linear, each output sample $y(m)$ can be expressed as a linear combination of input samples. A general form [10] for this expression is

$$y(m) = \sum_{n=-\infty}^{\infty} g_m(n) x\left(\left\lfloor \frac{mM}{L} \right\rfloor - n\right) \tag{3.66}$$

where $g_m(n)$ is periodic in m with period L, i.e.,

$$g_m(n) = g_{m+rL}(n), \quad r = 0, \pm 1, \pm 2, \ldots \tag{3.67}$$

Thus the system $g_m(n)$ belongs to the class of linear, periodically time-varying systems.

Consider now several specific cases of sampling rate conversion systems. First consider the trivial case $T' = T$ or $L = M = 1$, in which case Eq. (3.66) reduces to the simple time-invariant digital convolution equation, i.e.,

$$y(m) = \sum_{n=-\infty}^{\infty} g(n) x(m - n) \tag{3.68}$$

since the period of $g_m(n)$ is 1 and the integer part of $m - n$ is the same as $m - n$.

Next consider the case of sampling rate reduction (decimation) by an integer factor M. In this case we get

$$y(m) = \sum_{n=-\infty}^{\infty} g_m(n) x(mM - n) \tag{3.69}$$

where $g_m(n) = g(n) = h(n)$ for all m and n, with $h(n)$ the lowpass filter impulse response of the system of Fig. 3.7. Although $g_m(n)$ is *not* a function of m for this case, it can readily be shown that the overall system of Eq. (3.69) is not time-invariant [10].

Next consider the case of sampling rate increase (interpolation) by an integer factor L. By comparing Eqs. (3.66) and (3.52), we see that the form of $g_m(n)$ is

$$g_m(n) = h[nL + ((m))_L], \qquad \text{for all } m \text{ and } n \tag{3.70}$$

and also that $g_m(n)$ is periodic in m with period L.

Finally, if we consider the general case of sampling rate conversion by the rational fraction M/L, then from Eq. (3.61) we get the result

$$g_m(n) = h[nL + ((mM))_L], \qquad \text{for all } m \text{ and } n \tag{3.71}$$

where $h(k)$ is the time-invariant unit sample response of the lowpass filter at the sampling rate LF_s.

3.3 PRACTICAL STRUCTURES FOR DECIMATORS AND INTERPOLATORS

It is easy to understand the need for studying structures for realizing sampling rate conversion systems by examining the simple block diagram of Fig. 3.9(b), which can be used to convert the sampling rate of a signal by a factor L/M. As discussed in Section 3.2, the theoretical model for this system first increases the signal sampling rate by a factor L (by filling in $L-1$ zero-valued samples between each pair of samples of $x(n)$ to give the signal $w(k)$), then filters $w(k)$ (to eliminate the images of $X(\omega)$) by a standard linear, time-invariant, lowpass filter $h(k)$ to give $v(k)$, and then compresses the sampling rate of $v(k)$ by a factor M (by retaining 1 of each M samples of $v(k)$). A direct implementation of this system is grossly inefficient since the lowpass filter $h(k)$ is operating at the high sampling rate on a signal for which $L-1$ out of each L input values are zero, and the values of the filtered output are required only once each M samples. For this example, we can directly apply this knowledge in implementing the system of Fig. 3.9(b) in a more efficient manner, as will be discussed in this section.

3.3.1 Signal Flow Graphs

To precisely define the sets of operations necessary to implement these digital systems, we will strongly rely on the concepts of signal flow graph representation [10]. Signal flow graphs provide a graphical representation of the explicit set of equations that are used to implement such systems. Furthermore, manipulating the flow graphs in a pictorial way is equivalent to manipulating the mathematical equations.

Figure 3.11 illustrates an example of a signal flow graph of a direct-form finite impulse response (FIR) digital filter. The input branch applies the external signal $x(n)$ to the network, and the output of the network $y(n)$ is identified as one of the node

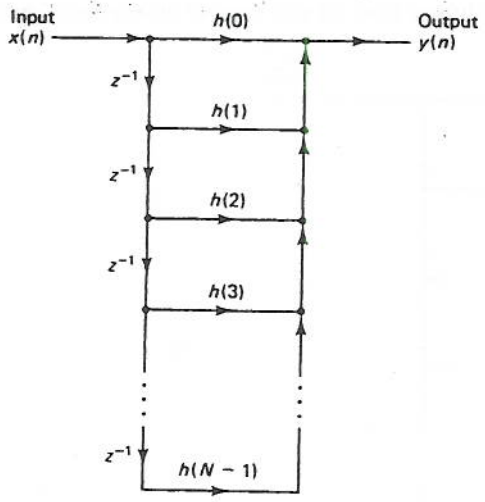


Figure 3.11 Direct-form structure for an FIR digital filter.

values. Branches define the signal operations in the structure such as delays, gains, and sampling rate expanders and compressors. Nodes define the connection points and summing points. The signal entering a branch is taken as the signal associated with the input node value of the branch. The node value of a branch is the sum of all branch signals entering the node.

From the signal flow graph in Fig. 3.11 we can immediately write down the network equation as

$$y(n) = x(n)h(0) + x(n-1)h(1) + \cdots + x(n-N+1)h(N-1)$$

An important concept in the manipulation of signal flow graphs is the principle of commutation of branch operations. Two branch operations commute if the order of their cascade operation can be interchanged without affecting the input-to-output response of the cascaded system. Thus interchanging commutable branches in a network is one way of modifying the network without affecting the desired input-to-output network response. This operation will be used extensively in constructing efficient structures for decimation and interpolation, as we will see shortly.

Another important network concept on which we rely heavily is that of transposition and duality [10]. Basically, a dual system is one that performs a complementary operation to that of an original system, and it can be constructed from the original system through the process of transposition. We have already seen an example of dual systems, namely, the integer decimator and interpolator (Fig. 3.7a and Fig. 3.8a) for the case $M = L$.

Basically the transposition operation is one in which the direction of all branches in the network are reversed and the roles of the input and output of the network are interchanged. Furthermore, all branch operations are replaced by their transpose operations. In the case of linear time-invariant branch operations, such as gains and delays, these branch operations remain unchanged. Thus, for example, the transpose of the direct-form structure of Fig. 3.11 is the transposed direct-form structure shown in Fig. 3.12. Also it can be shown that for the case of linear time-invariant systems

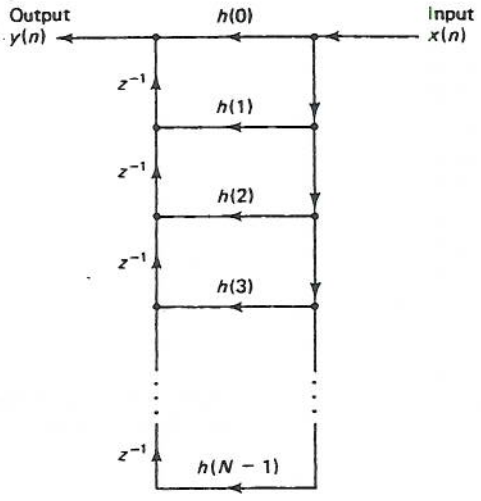


Figure 3.12 Transposed direct-form FIR filter structure.

the input-to-output system response of a system and its dual are identical (e.g., it can be verified that the networks of Fig. 3.11 and Fig. 3.12 have identical system functions).

For the time-varying systems this is not necessarily the case. For example, the transpose of a sampling rate compressor is a sampling rate expander, and the transpose of a sampling rate expander is a sampling rate compressor, as shown in Fig. 3.13. Clearly these systems do not have the same system response.

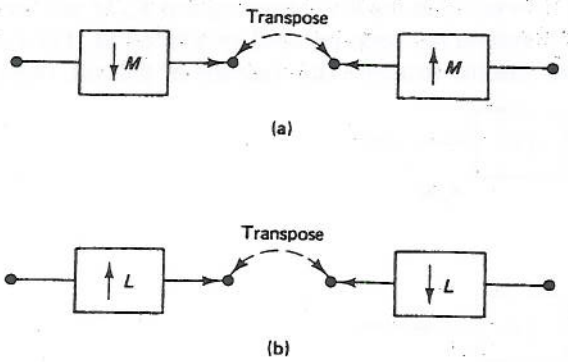


Figure 3.13 Transpositions of a sampling rate compressor and expander.

By extending the concepts of transposition rigorously, we can also show that the transposition of a network that performs a sampling rate conversion by the factor L/M is a network that performs a sampling rate conversion by the factor M/L. This is illustrated in Fig. 3.14.

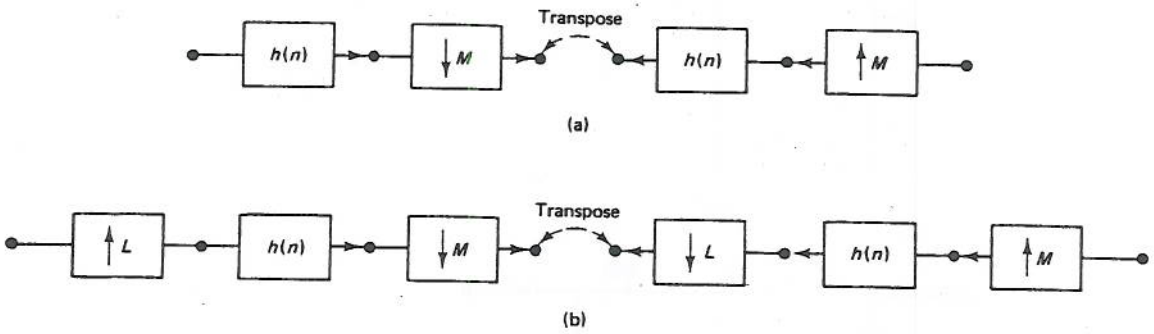


Figure 3.14 Transpositions of a decimator and a generalized L/M sampling rate changer.

3.3.2 Direct-Form FIR Structures for Integer Changes in Sampling Rates

Consider the model of an M-to-1 decimator as shown in Fig. 3.15(a). According to this model the filter $h(n)$ operates at the high sampling rate F_s, and $M - 1$ out of every M output samples of the filters are discarded by the M-to-1 sampling rate compressor. In particular, if we assume that the filter $h(n)$ is an N-point FIR filter realized with a direct-form structure, the network of Fig. 3.15(b) results. The multiplications by $h(0)$,

$h(1), \ldots, h(N-1)$ and the associated summations in this network must be performed at the rate F_s.

A more efficient realization of the above structure can be achieved by noting that the branch operations of sampling rate compression and gain can be commuted. By performing a series of commutative operations on the network, we obtain the modified network of Fig. 3.15(c). The multiplications and additions associated with the coefficients $h(0)$ to $h(N-1)$ now occur at the low sampling rate F_s/M and therefore the total computation rate in the system has been reduced by a factor M. For every M samples of $x(n)$ that are shifted into the structure (the cascade of delays), one output

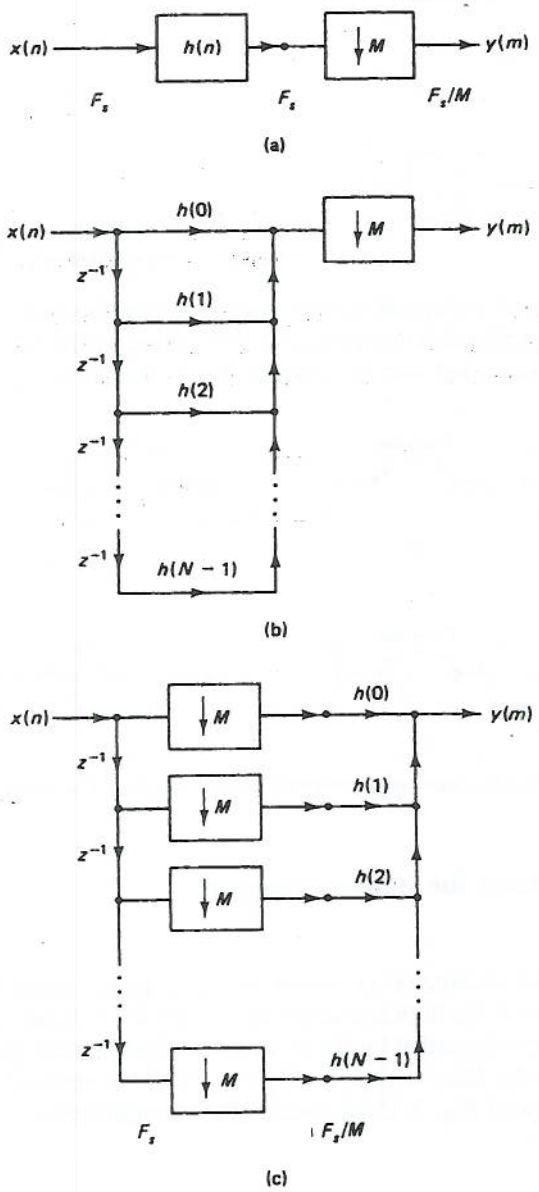


Figure 3.15 Generation of an efficient direct-form structure of an M-to-1 decimator.

sample $y(m)$ is computed. Thus the structure of Fig. 3.15(c) is seen to be a direct realization of Eq. (3.29).

An efficient structure for the 1-to-L integer interpolator, using an FIR filter, can be derived in a similar manner. We begin with the cascade model for the interpolator shown in Fig. 3.16(a). In this case however, if $h(m)$ is realized with the direct-form structure of Fig. 3.11 we are faced with the problem of commuting the 1-to-L sampling rate expander with a series of unit delays. One way around this problem is to realize $h(m)$ with the transposed direct-form FIR structure as shown in Fig. 3.12. The sampling rate expander can then be commuted into the network as shown by the series

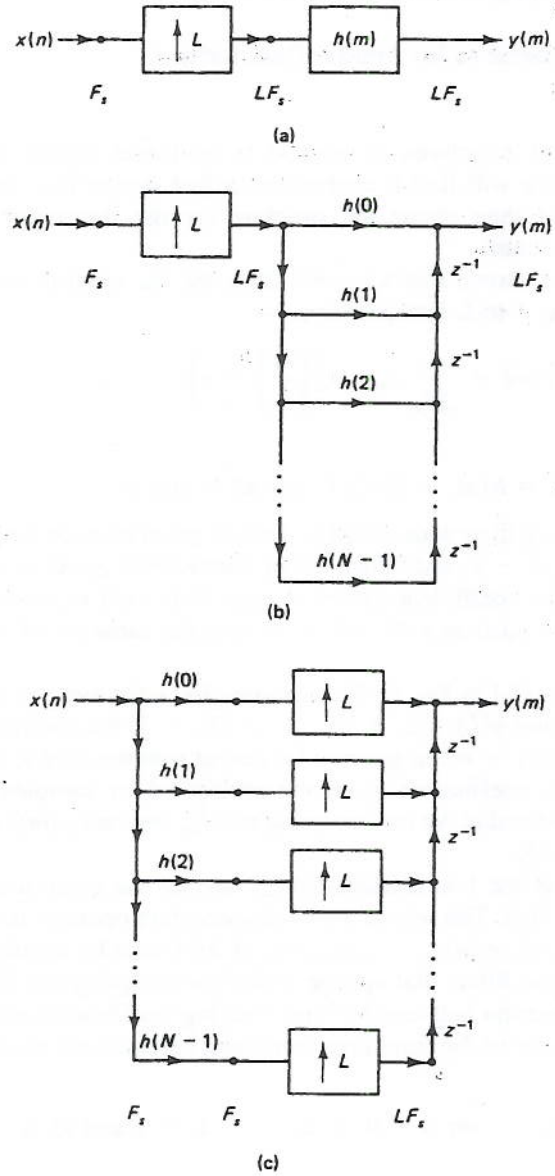


Figure 3.16 Steps in the generation of an efficient structure of a 1-to-L interpolator.

of operations in Fig. 3.16. Since the coefficients $h(0), h(1), \ldots, h(N-1)$ in Fig. 3.16(c) are now commuted to the low sampling rate side of the network, this structure requires a factor of L times less computation than the structure in Fig. 3.16(b).

An alternative way of deriving the structure of Fig. 3.16(c) is by a direct transposition of the network of Fig. 3.15(c) (letting $L = M$). This is a direct consequence of the fact that decimators and interpolators are duals. A further property of transposition is that for the resulting network, neither the number of multipliers nor the rate at which these multipliers operate will change [15]. Thus if we are given a network that is minimized with respect to its multiplication rate, then its transpose will also be minimized with respect to its multiplication rate.

3.3.3 Polyphase FIR Structures for Integer Decimators and Interpolators

A second general class of structures of interest in multirate digital systems is the polyphase networks. We will find it convenient to first derive this structure for the L-to-1 interpolator and then obtain the structure for the decimator by transposing the interpolator structure.

In Section 3.2 it was shown that a general form for the input-to-output time-domain relationship for the 1-to-L interpolator is

$$y(m) = \sum_{n=-\infty}^{\infty} g_m(n)x\left(\left\lfloor \frac{m}{L} \right\rfloor - n\right) \tag{3.72}$$

where

$$g_m(n) = h(nL + ((m))_L) \quad \text{for all } m \text{ and } n \tag{3.73}$$

is a periodically time-varying filter with period L. Thus to generate each output sample $y(m)$, $m = 0, 1, 2, \ldots, L-1$, a different set of coefficients $g_m(n)$ is used. After L outputs are generated, the coefficient pattern repeats; thus $y(L)$ is generated using the same set of coefficients $g_0(n)$ as $y(0)$, $y(L+1)$ uses the same set of coefficients $g_1(n)$ as $y(1)$, and so on.

Similarly the term $\lfloor m/L \rfloor$ in Eq. (3.72) increases by 1 for every L samples of $y(m)$. Thus for output samples $y(L), y(L+1), \ldots, y(2L-1)$ the coefficients $g_m(n)$ are multiplied by samples $x(1-n)$. In general, for output samples $y(rL), y(rL+1), \ldots, y(rL+L-1)$, the coefficients $g_m(n)$ are multiplied by samples $x(r-n)$. Thus we see that $x(n)$ is updated at the low sampling rate F_s, whereas $y(m)$ is evaluated at the high sampling rate LF_s.

An implementation of the 1-to-L interpolator based on the computation of Eq. (3.72) is shown in Fig. 3.17(a). The way in which this structure operates is as follows. The partitioned subsets $g_0(n), g_1(n), \ldots, g_{L-1}(n)$, of $h(m)$ can be identified with L separate linear time-invariant filters that operate at the low sampling rate F_s. To make this subtle notational distinction between the time-varying coefficients and the time-invariant filters, we will refer to the time-invariant filters respectively as $p_0(n), p_1(n), \ldots, p_{L-1}(n)$. Thus

$$p_\rho(n) = g_\rho(n), \qquad \text{for } \rho = 0, 1, 2, \ldots, L-1 \text{ and all } n \tag{3.74}$$

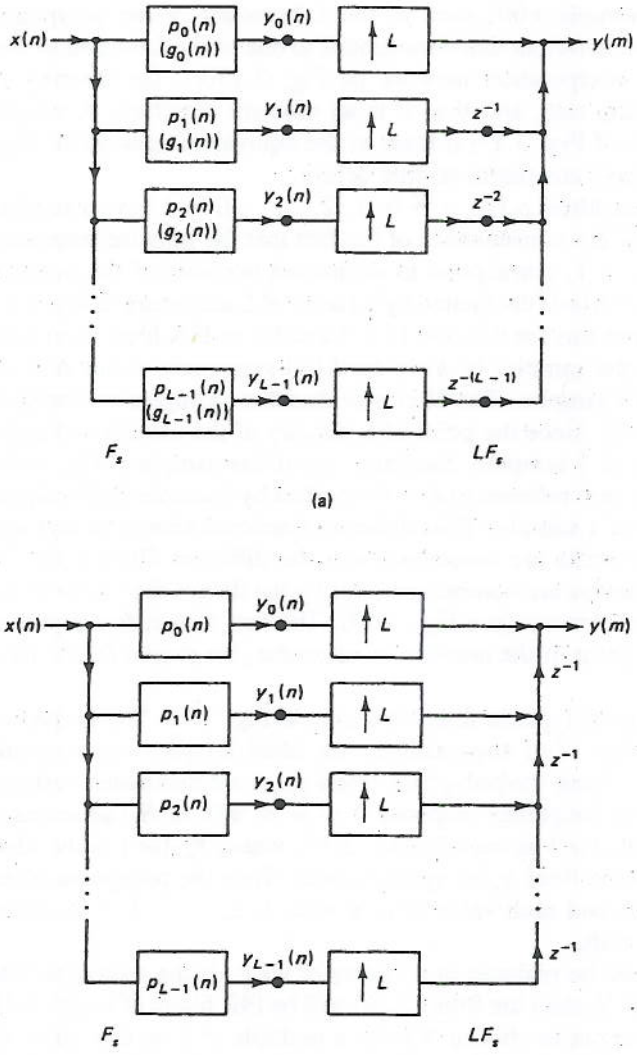


Figure 3.17 Polyphase structures for a 1-to-L interpolator.

These filters $p_\rho(n)$ will be referred to as the *polyphase filters*. Furthermore, by combining Eqs. (3.73) and (3.74) we see that

$$p_\rho(n) = h(nL + \rho), \qquad \text{for } \rho = 0, 1, 2, \ldots, L-1 \text{ and all } n \tag{3.75}$$

For each new input sample $x(n)$, there are L output samples (see Fig. 3.17). The output from the upper path $y_0(m)$ has nonzero values for $m = nL$, $n = 0, \pm 1, \pm 2, \ldots$, which correspond to system outputs $y(nL)$, $n = 0, \pm 1, \ldots$. The output from the next path $y_1(m)$ is nonzero for $m = nL + 1$, $n = 0, \pm 1, \pm 2, \ldots$ because of the delay of one sample at the high sampling rate. Thus $y_1(m)$ corresponds to the interpolation output samples $y(nL + 1)$, $n = 0, \pm 1, \ldots$. In general, the output of the

ρth path, $y_\rho(m)$ corresponds to the interpolation output samples $y(nL + \rho)$, $n = 0, \pm 1, \ldots$. For each input sample $x(n)$, each of the L branches of the polyphase network contributes one nonzero output that corresponds to one of the L outputs of the network. In the polyphase interpolation network of Fig. 3.17(a), the filtering is performed at the low sampling rate, and thus it is an efficient structure. A simple manipulation of the structure of Fig. 3.17(a) leads to the equivalent network of Fig. 3.17(b), in which all the delays are single sample delays.

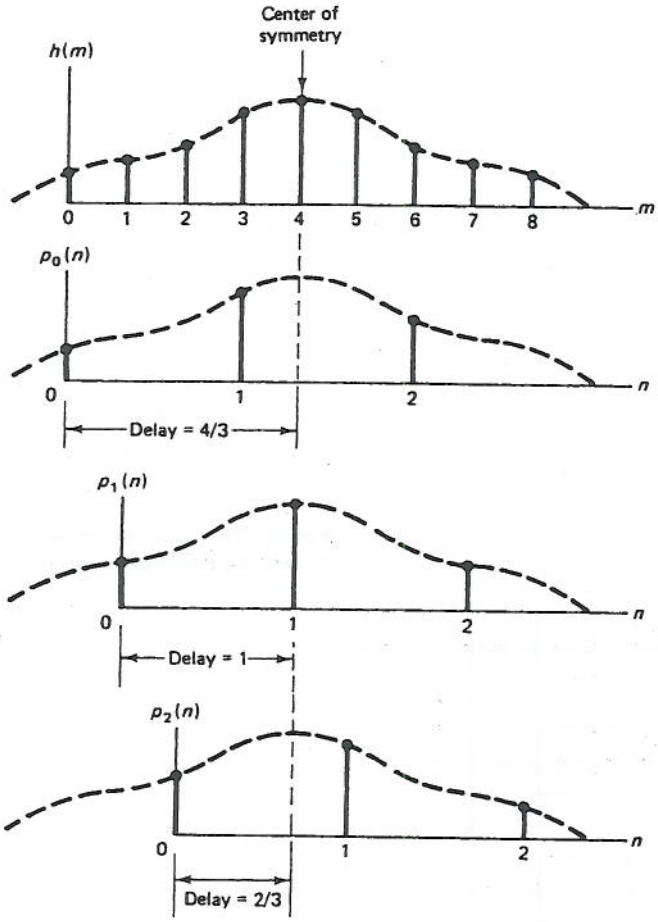
The individual polyphase filters $p_\rho(n)$, $\rho = 0, 1, 2, \ldots, L - 1$ have a number of interesting properties. This is a consequence of the fact that the impulse responses $p_\rho(n)$, $\rho = 0, 1, 2, \ldots, L - 1$, correspond to decimated versions of the impulse response of the prototype filter $h(m)$ (decimated by a factor of L according to Eq. 3.73 or 3.75). Figure 3.18 illustrates this for the case $L = 3$ and for an FIR filter $h(m)$ with $N = 9$ taps. Part (a) shows the samples of $h(m)$ for $h(m)$ symmetric about $m = 4$. Thus $h(m)$ has a flat delay of 4 samples. The filter $\rho_0(n)$ has three samples corresponding to $h(0)$, $h(3)$, $h(6) = h(2)$. Since the point of symmetry of the envelope of $\rho_0(n)$ is $n = \frac{4}{3}$, it has a flat delay of $\frac{4}{3}$ samples. Similarly, $\rho_1(n)$ has samples $h(1)$, $h(4)$, $h(7) = h(1)$, and because its zero reference ($n = 0$) is offset by $\frac{1}{3}$ sample (with respect to $m = 0$) it has a flat delay of 1 sample. Thus different fractional sample delays and consequently different phase shifts are associated with the different filters $p_\rho(n)$, as seen in Fig. 3.18(b). These delays are compensated for by the delays that occur at the high sampling rate LF in the network (see Fig. 3.17). The fact that different phases are associated with different paths of the network is, of course, the reason for the term *polyphase network*.

A second property of the polyphase filters is shown in Fig. 3.19. The frequency response of the prototype filter $h(m)$ approximates the ideal lowpass characteristic $H_I(\omega)$ shown in Fig. 3.19(a). Since the polyphase filters $p_\rho(n)$ are decimated versions of $h(m)$ (decimated by L), the frequency response $0 \leq \omega \leq \pi/L$ of $H_I(\omega)$ scales to the range $0 \leq \omega' \leq \pi$ for $P_{\rho,I}(\omega')$ as seen in Fig. 3.19, where $P_{\rho,I}(\omega')$ is the ideal characteristic that the polyphase filter $p_\rho(n)$ approximates. Thus the polyphase filters approximate allpass functions and each value of ρ, $\rho = 0, 1, 2, \ldots, L - 1$, corresponds to a different phase shift.

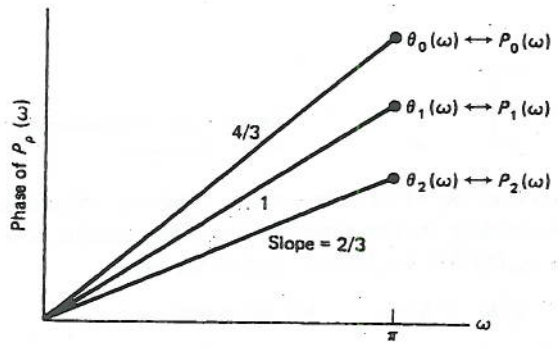
The polyphase filters can be realized in a variety of ways. If the prototype filter $h(m)$ is an FIR filter of length N, then the filters $p_\rho(n)$ will be FIR filters of length N/L. In this case it is often convenient to choose N to be a multiple of L so that all of the polyphase filters are of equal length. These filters may be realized by any of the conventional methods for implementing FIR filters such as the direct-form structure or the methods based on fast convolution [10]. If a direct-form FIR structure is used for the polyphase filters, the polyphase structure of Fig. 3.17 will require the same multiplication rate as the direct-form interpolator structure of Fig. 3.16.

By transposing the structure of the polyphase 1-to-L interpolator of Fig. 3.17(b), we get the polyphase M-to-1 decimator structure of Fig. 3.20, where L is replaced by M. Again the filtering operations of the polyphase filters occur at the low sampling rate side of the network, and they can be implemented by any of the conventional structures discussed above.

In the preceding discussion for the 1-to-L interpolator, we have identified the coefficients of the polyphase filters $p_\rho(n)$ with the coefficient sets $g_m(n)$ of the time-



(a)



(b)

Figure 3.18 Illustration of the properties of polyphase network filters.

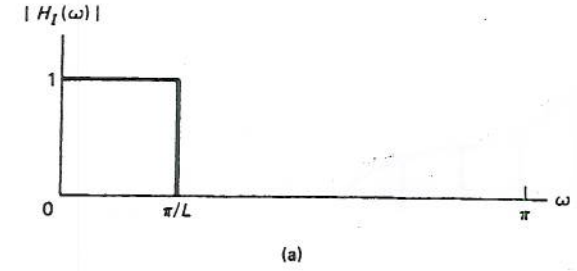


Figure 3.19 Ideal frequency response of the polyphase network filters.

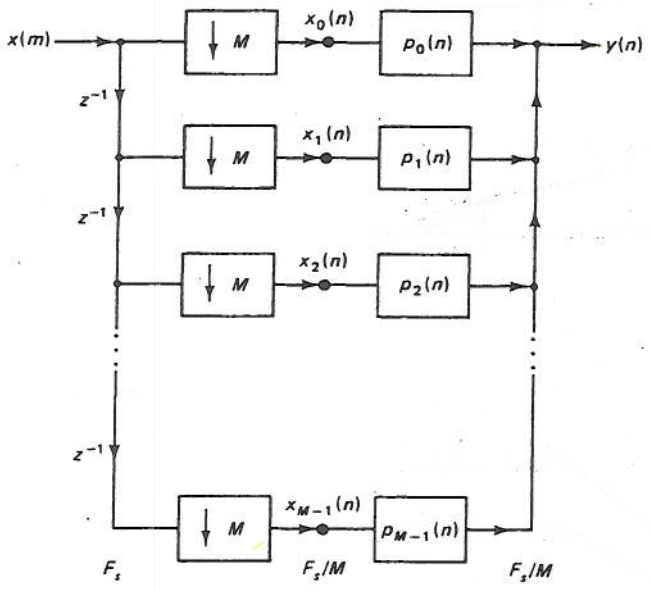


Figure 3.20 Polyphase structure for an M-to-1 decimator.

varying filter model. In the case of the M-to-1 decimator, however, this identification cannot be made directly. According to the time-varying filter model, discussed in Section 3.2, the coefficients $g_m(n)$ for the M-to-1 decimator are

$$g_m(n) = g(n) = h(n), \qquad \text{for all } n \text{ and } m \tag{3.76}$$

Alternatively, according to the transpose network of Fig. 3.20, the coefficients of the M-to-1 polyphase decimator are

$$p_\rho(n) = h(nM + \rho), \qquad \text{for } \rho = 0, 1, 2, \ldots, M - 1, \text{ and all } n \tag{3.77}$$

where ρ denotes the ρth polyphase filter. Thus the polyphase filters $p_\rho(n)$ for the M-to-1 decimator are equal to the time-varying coefficients $g_m(n)$ of the transpose (interpolator) of this decimator.

From a practical point of view it is often convenient to implement the polyphase structures in terms of a commutator model. By careful examination of the interpolator structure of Fig. 3.17 we can see that the outputs of each of the polyphase branches contributes samples of $y(m)$ for different time slots. Thus the 1-to-L sampling rate expander and delays can be replaced by a commutator, as shown in Fig. 3.21. The commutator rotates in a counterclockwise direction starting with the zeroth-polyphase branch at time $m = 0$.

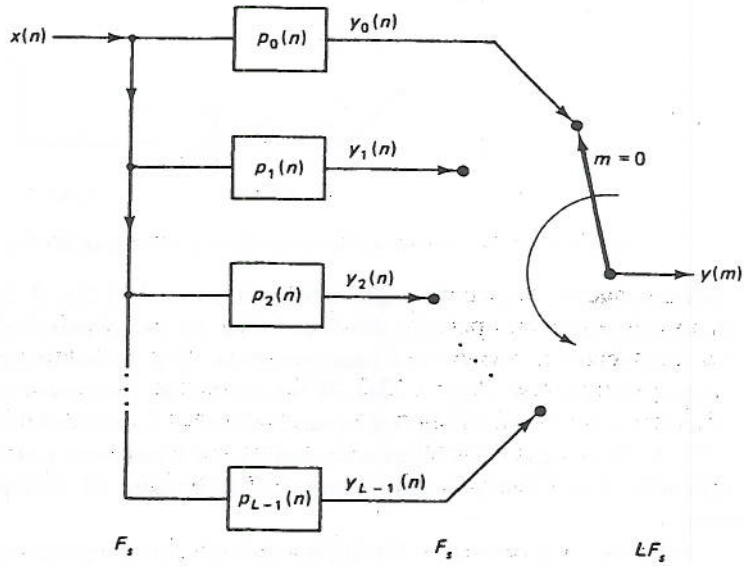


Figure 3.21 Counterclockwise commutator model for a 1-to-L interpolator.

A similar commutator model can be developed for the M-to-1 polyphase decimator by starting with the structure of Fig. 3.20 and replacing the delays and M-to-1 sampling rate compressors with a commutator. This leads to the structure of Fig. 3.22. Again the commutator rotates in a counterclockwise direction starting with the zeroth-polyphase branch at time $m = 0$.

3.3.4 FIR Structures with Time-Varying Coefficients for Interpolation/Decimation by a Factor of L/M

In the previous two sections we have considered implementations of decimators and interpolators using the direct-form and polyphase structures for the case of integer changes in the sampling rate. We obtained efficient realizations of these structures by commuting the filtering operations to occur at the low sampling rate. For the case of a network that realizes a change in sampling rate by a factor of L/M, it is difficult to

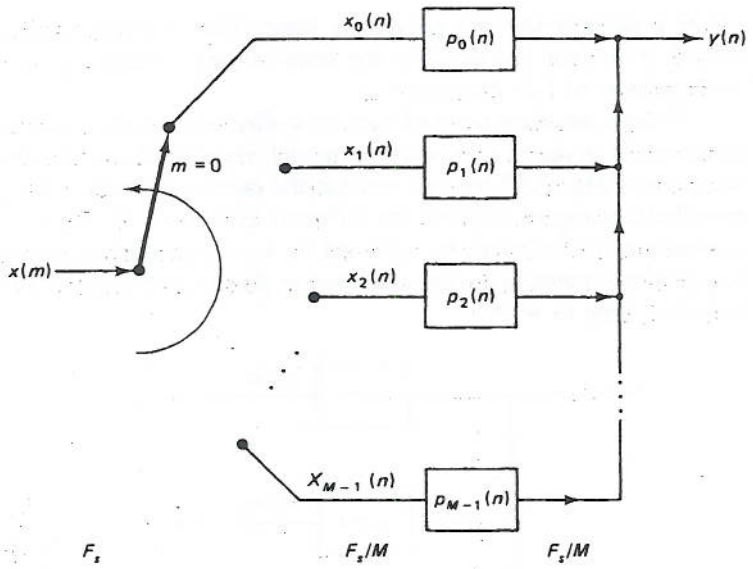


Figure 3.22 Counterclockwise commutator model for an M-to-1 decimator.

achieve such efficiencies. The difficulty is illustrated in Fig. 3.23. If we realize the 1-to-L interpolation part of the structure using the techniques described earlier, then we are faced with the problem of commuting the M-to-1 sampling rate compressor into the resulting network (Fig. 3.23a). If we realize the decimator part of the structure first, then the 1-to-L sampling rate expander must be commuted into the structure (Fig. 3.23b). In both cases difficulties arise and we are faced with a network that cannot be implemented efficiently simply using the techniques of commutation and transposition.

Efficient structures exist for implementing a sampling rate converter with a ratio in sampling rates of L/M, and in this section we discuss one such class of FIR structures with time-varying coefficients [10]. This structure can be derived from the time-domain input-to-output relation of the network, as derived in Section 3.2, namely

$$y(m) = \sum_{n=-\infty}^{\infty} g_m(n)x\left(\left\lfloor \frac{mM}{L} \right\rfloor - n\right) \tag{3.78}$$

where

$$g_m(n) = h(nL + ((mM))_L), \qquad \text{for all } m \text{ and all } n \tag{3.79}$$

and $h(k)$ corresponds to the lowpass (or bandpass) FIR prototype filter. It will be convenient for our discussion to assume that the length of the filter $h(k)$ is a multiple of L, i.e.,

$$N = QL \tag{3.80}$$

where Q is an integer. Then all of the coefficient sets $g_m(n)$, $m = 0, 1, 2, \ldots, L-1$ contain *exactly* Q coefficients. Furthermore, $g_m(n)$ is periodic in m with period L, i.e.,

$$g_m(n) = g_{m+rL}(n), \qquad r = 0, \pm 1, \pm 2, \ldots \tag{3.81}$$

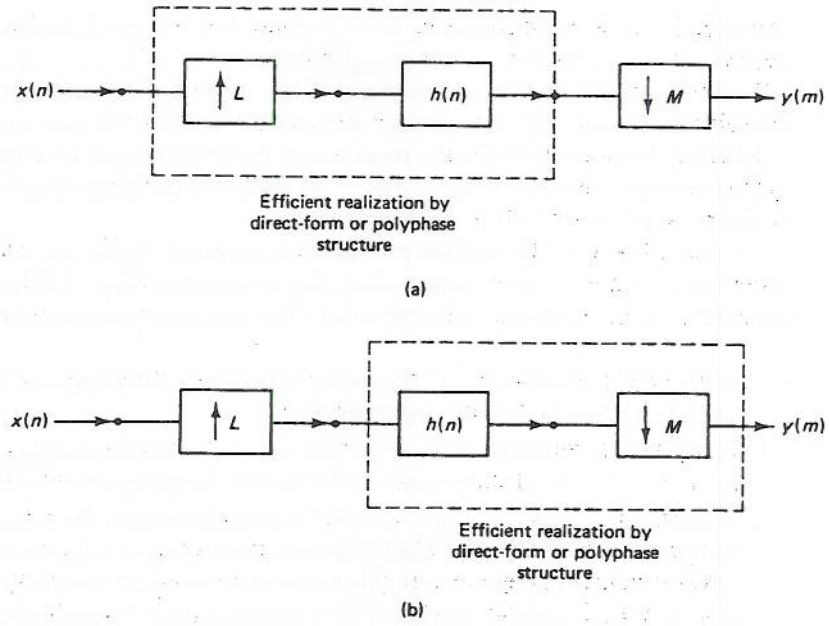


Figure 3.23 Possible realizations of an L/M sampling rate converter.

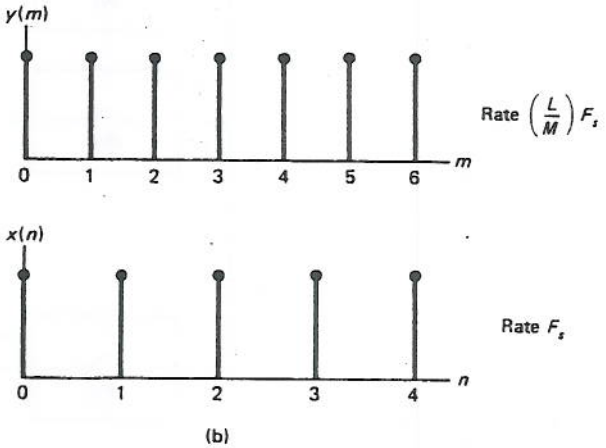
Therefore, Eq. (3.78) can be expressed as

$$y(m) = \sum_{n=0}^{Q-1} g_{((m))_L}(n)\, x\left(\left\lfloor \frac{mM}{L} \right\rfloor - n\right) \tag{3.82}$$

Equation (3.82) shows that the computation of an output sample $y(m)$ is obtained as a weighted sum of Q sequential samples of $x(n)$ starting at the $x(\lfloor mM/L \rfloor)$ sample and going backward in n sequentially. The weighting coefficients are periodically time-varying, so the $((m))_L$ coefficient set $g_{((m))_L}(n)$, $n = 0, 1, 2, \ldots, Q-1$, is used for the mth output sample. Figure 3.24 illustrates this timing relationship for the $n = 0$

$y(m)$	$x\left(\left\lfloor \frac{mM}{L} \right\rfloor\right)$	$g_{((m))_L}(0)$
m	$\left\lfloor \frac{2m}{3} \right\rfloor$	$((m))_L$
0	0	0
1	0	1
2	1	2
3	2	0
4	2	1
5	3	2
6	4	0

(a)



(b)

Figure 3.24 Timing relationships between $y(m)$ and $x(n)$ for the case $M = 2$, $L = 3$.

term in Eq. (3.82) and for the case $M = 2$ and $L = 3$. Figure 3.24(a) shows the index values of $y(m)$, $x(\lfloor mM/L \rfloor)$, and $g_{((m))_L}(0)$ for $m = 0, 1, 2, \ldots, 6$. Figure 3.24(b) illustrates the relative timing positions of the signals $y(m)$ and $x(n)$ drawn on an absolute time scale. By comparison of parts (a) and (b) we can see that the value $x([mM/L])$ always represents the most recent available sample of $x(n)$, i.e., $y(0)$ and $y(1)$ are computed on the basis of $x(0 - n)$. For $y(2)$ the most recent available value of $x(n)$ is $x(1)$, for $y(3)$ it is $x(2)$, and so on.

Based on Eq. (3.82) and the preceding description of how the input, output, and coefficients enter into the computation, the structure of Fig. 3.25 is suggested for realizing an L/M sampling rate converter. The structure consists of the following:

1. A Q sample "shift register" operating at the input sampling rate F_s, which stores sequential samples of the input signal
2. A direct-form FIR structure with time-varying coefficients ($g_{((m))_L}(n)$, $n = 0, 1, 2, \ldots, Q - 1$) that operates at the output sampling rate $(L/M)F_s$
3. A series of digital "hold-and-sample" boxes that couple the two sampling rates. The input side of the box "holds" the most recent input value until the next input value comes along; the output side of the box "samples" the input values at times $n = mM/L$. For times when mM/L is an integer (i.e., input and output sampling times are the same), the input changes first and the output samples the changed input.

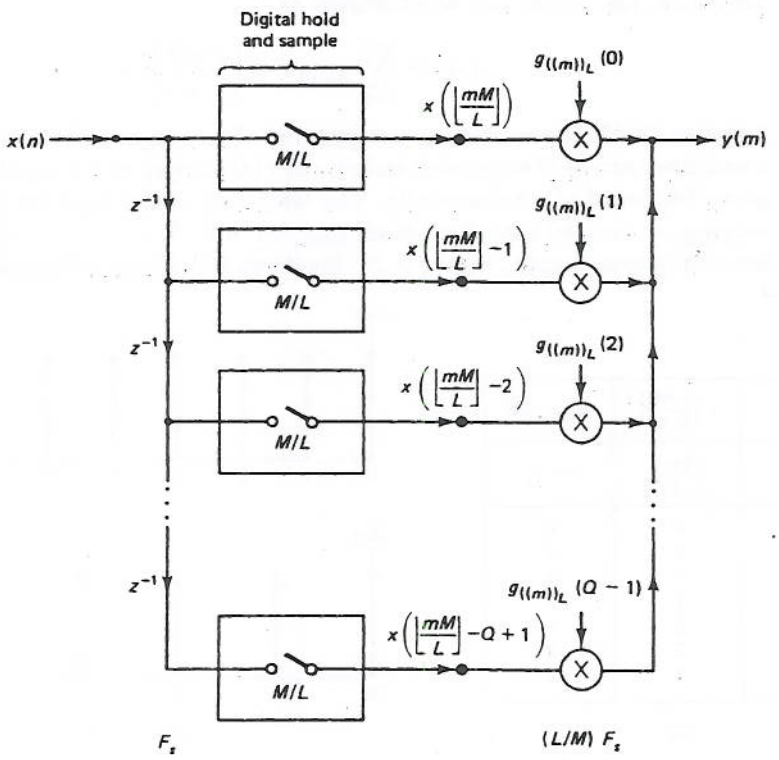


Figure 3.25 Efficient structure for realizing an L/M sampling rate converter.

It should be clear that the structure of Fig. 3.25 is an efficient one for implementing an (L/M) sampling rate converter since the filtering operations are all performed at the output sampling rate with the minimum required number of coefficients used to generate each output.

Figure 3.26 shows a diagram of a program configuration to implement this structure in a block-by-block manner. The program takes in a block of M samples of the input signal, denoted as $x(n')$, $n' = 0, 1, 2, \ldots, M-1$, and computes a block of L output samples $y(m')$ $m' = 0, 1, 2, \ldots, L-1$. For each output sample time m', $m' = 0, 1, 2, \ldots, L-1$, the Q samples from the state-variable buffer are multiplied respectively with Q coefficients from one of the coefficient sets $g_{m'}(n')$ and the products are accumulated to give the output $y(m')$. Each time the quantity $\lfloor m'M/L \rfloor$ increases by 1, one sample from the input buffer is shifted into the state-variable buffer. (This information can be stored in a control array.) Thus after L output values are computed, M input samples have been shifted into the state-variable buffer and the process can be repeated for the next block of data. In the course of processing one block of data (M input samples and L output samples), the state-variable buffer is sequentially addressed L times and the coefficient storage buffer is sequentially addressed once. A program that performs this computation can be found in [16].

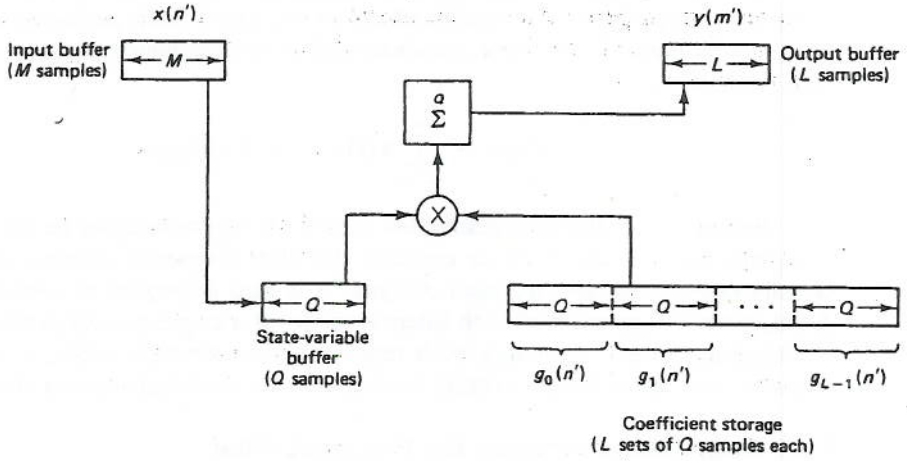


Figure 3.26 Block diagram of a program structure to implement the signal flow graph of Fig. 3.25 in a block-by-block manner.

3.3.5 Comparison of Structures

In Section 3.3 so far, we have discussed three principal classes of FIR structures for decimators and interpolators. In Section 3.5 we will discuss multistage cascades of these structures and show how such cascading can lead to additional gains in computational efficiency when conversion ratios are large. A natural question to ask at this point is which of these methods is most efficient. The answer, unfortunately, is nontrivial and is highly dependent on the application being considered. Some insight and direction, however, can be provided by observing a few general properties of the classes of structures discussed here.

The direct-form structures have the advantage that they can be easily modified to exploit symmetry in the system function to gain an additional reduction in computation by a factor of approximately two. The polyphase structures have the advantage that the filters $p_\rho(n)$ can be easily realized with efficient techniques such as the fast convolution methods based on the FFT. As such, this structure has been found useful for filter banks [10]. The structures with time-varying coefficients are particularly useful when considering conversions by factors of L/M.

3.4 DESIGN OF FIR FILTERS FOR DECIMATION AND INTERPOLATION

In the previous discussion we have assumed that the filter $h(k)$ approximates some ideal lowpass (or bandpass) characteristic. Consequently the effectiveness of these systems is directly related to the type and quality of design of this digital filter. The purpose of this section is to review digital filter design techniques that are especially applicable to the design of the digital filter in sampling rate-changing systems.

The filter design problem is essentially one of determining suitable values of $h(k)$ to meet given performance specifications on the filter. Such performance specifications can be made on the time response $h(k)$ or the frequency response of the filter $H(\omega)$ defined as

$$H(\omega) = \sum_{k=-\infty}^{\infty} h(k)e^{-j\omega k} = H(z)\big|_{z=e^{j\omega}} \tag{3.83}$$

Before proceeding to a discussion of filter design techniques for decimators and interpolators, it is important to consider the ideal frequency domain and the time-domain criteria that specify such designs. It is also important to consider, in more detail, the representation of such filters in terms of a single prototype filter or as a set of polyphase filters. Although both representations are equivalent, it is sometimes easier to view filter design criteria in terms of one representation or the other.

3.4.1 Relationship Between the Prototype Filter and Its Polyphase Representation

As discussed in Section 3.3, the coefficients, or impulse responses, of the polyphase filters correspond to sampled (and delayed) versions of the impulse response of the prototype filter. For a 1-to-L interpolator there are L polyphase filters and they are defined as (see Fig. 3.18)

$$p_\rho(n) = h(\rho + nL), \qquad \rho = 0, 1, 2, \ldots, L-1, \text{ and all } n \tag{3.84}$$

Similarly, for an M-to-1 decimator there are M polyphase filters in the polyphase structure and they are defined as

$$p_\rho(n) = h(\rho + nM), \qquad \rho = 0, 1, 2, \ldots, M-1, \text{ and all } n \tag{3.85}$$

Taken as a set, the samples $p_\rho(n)$ ($\rho = 0, 1, \ldots, L-1$, for an interpolator or

$\rho = 0, 1, \ldots, M-1$, for a decimator) represent all of the samples of $h(k)$. Since the development of the filter specifications is identical for both cases (1-to-L interpolators and M-to-1 decimators) we will consider only the case of interpolators. The results for decimators can then simply be obtained by replacing L by M in the appropriate equations.

The samples $h(k)$ can be recovered from $p_\rho(n)$ by sampling rate expanding the sequences $p_\rho(n)$ by a factor L. Each expanded set is then delayed by ρ samples and the L sets are then summed to give $h(k)$ (the reverse operation to that of Fig. 3.18). If we let $\hat{p}_\rho(k)$ represent the sampling rate expanded set

$$\hat{p}_\rho(k) = \begin{cases} p_\rho(k/L), & k = 0, \pm L, \pm 2L, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{3.86}$$

then $h(k)$ can be reconstructed from $\hat{p}_\rho(k)$ via the summation

$$h(k) = \sum_{\rho=0}^{L-1} \hat{p}_\rho(k-p) \tag{3.87}$$

The z-transform $H(z)$ of the prototype filter can similarly be expressed in terms of the z-transforms of the polyphase filters $P_\rho(z)$. It can be shown that

$$H(z) = \sum_{\rho=0}^{L-1} z^{-\rho} P_\rho(z^L) \tag{3.88}$$

Finally, the z-transform $P_\rho(z)$ can be expressed in terms of $H(z)$ according to the following derivation. If we define a sampling function $\delta_\rho(k)$ such that

$$\delta_\rho(k) = \begin{cases} 1, & k = \rho, \rho \pm L, \rho \pm 2L, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{3.89}$$

$$= \frac{1}{L} \sum_{\ell=0}^{L-1} e^{j2\pi\ell(k-\rho)/L} \tag{3.90}$$

then the sampling rate expanded sequences $\hat{p}_\rho(k)$ in Eq. (3.86) can be expressed as

$$\hat{p}_\rho(k) = \delta_\rho(k)h(k) = h(k)\frac{1}{L}\sum_{\ell=0}^{L-1} e^{j2\pi\ell(k-\rho)/L} \tag{3.91}$$

The z-transform $P_\rho(z)$ can then be expressed in the form

$$P_\rho(z) = \sum_{n=-\infty}^{\infty} p_\rho(n)z^{-n} = \sum_{n=-\infty}^{\infty} \hat{p}_\rho(\rho + nL)z^{-n} \tag{3.92}$$

and by the substitution of variables $k = \rho + nL$,

$$P_\rho(z) = \sum_{k=-\infty}^{\infty} \hat{p}_\rho(k)z^{-(k-\rho)/L} \tag{3.93}$$

Combining Eqs. (3.91) and (3.93), we get

$$P_\rho(z) = \frac{1}{L}\sum_{k=-\infty}^{\infty}\sum_{\ell=0}^{L-1} h(k)e^{j2\pi\ell(k-\rho)/L}z^{-(k-\rho)/L} \tag{3.94}$$

Letting $z = e^{j\omega}$ and rearranging terms gives

$$\begin{aligned} P_\rho(\omega) &= \frac{1}{L}\sum_{\ell=0}^{L-1} e^{j(\omega-2\pi\ell)\rho/L} \sum_{k=-\infty}^{\infty} h(k)e^{-j(\omega-2\pi\ell)k/L} \\ &= \frac{1}{L}\sum_{\ell=0}^{L-1} e^{j(\omega-2\pi\ell)\rho/L} H(\omega - 2\pi\ell)/L, \qquad \rho = 0, 1, 2, \ldots, L-1 \end{aligned} \tag{3.95}$$

Equation (3.95) shows the relationships of the Fourier transforms of the polyphase filters to the Fourier transform of the prototype filter.

3.4.2 Ideal Frequency-Domain Characteristics for Interpolation and Decimation Filters

In the previous sections we have assumed that the filter $h(k)$ approximates some ideal lowpass (or bandpass) characteristic. We will elaborate on these "ideal" characteristics in somewhat more detail in the next two sections. In practice it is also necessary to specify a performance criterion to measure (in a consistent manner) how closely an actual filter design approximates this ideal characteristic. Since different design techniques are often based on different criteria, we will consider these criteria as they arise.

Recall from the discussion in Section 3.2 that the interpolator filter $h(k)$ must approximate the ideal lowpass characteristic defined as

$$H_I(\omega') = \begin{cases} L, & |\omega'| < \pi/L \\ 0, & \text{otherwise} \end{cases} \tag{3.96}$$

where the subscript I refers to the "ideal" characteristic.

By combining Eqs. (3.95) and (3.96) it is possible to derive the equivalent ideal characteristics $P_{\rho,I}(\omega)$ that are implied in the polyphase filters. Note that the frequency variable ω' refers to $h(m)$ whereas the frequency variable $\omega = \omega' L$ refers to the polyphase filters $p_\rho(n)$. Because of the constraint imposed by Eq. (3.96), only the $\ell = 0$ term in Eq. (3.95) is nonzero, and the equation simplifies to the form

$$\begin{aligned} P_{\rho,I}(\omega) &= \frac{1}{L} e^{j\omega\rho/L} H_I(\omega/L) \\ &= e^{j\omega\rho/L}, \qquad \rho = 0, 1, 2, \ldots, L-1 \end{aligned} \tag{3.97}$$

Equation (3.97) shows that the "ideal" polyphase filters $P_{\rho,I}(n)$ should approximate allpass filters with linear phase shifts corresponding to fractional advances of ρ/L samples ($\rho = 0, 1, 2, \ldots, L-1$) (ignoring any fixed delays that must be introduced in practical implementations of such filters).

In some cases it is known that the spectrum of $x(n)$ does not occupy its full bandwidth. This property can be used to advantage in the filter design, and we will see examples of this in the next section on cascaded (multistage) implementations of sampling rate changing systems. If we define ω_c as the highest frequency of interest in $X(\omega)$, i.e.,

$$|X(\omega)| < \epsilon, \qquad \text{for } \pi > |\omega| > \omega_c \tag{3.98}$$

where ϵ is a small quantity (relative to the peak of $|X(\omega)|$), as shown in Fig. 3.27 (for $L = 5$). In this case, the ideal interpolator filter has to remove only the $(L-1)$ repetitions of the band of $X(\omega)$ where $|X(\omega)| > \epsilon$. Thus in the frequency domain, the ideal interpolator filter satisfies the constraints

$$H_I(\omega') = \begin{cases} L, & 0 \le |\omega'| \le \omega_c/L \\ 0, & (2\pi r - \omega_c)/L \le |\omega'| \le (2\pi r + \omega_c)/L, \; r = 1, 2, \ldots, L-1 \end{cases} \tag{3.99}$$

as illustrated in Fig. 3.27(c). The bands from $(2\pi r + \omega_c)/L$ to $[2\pi(r+1) - \omega_c]/L$, $r = 0, 1, \ldots$, are "don't care" (ϕ) bands in which the filter frequency response is essentially unconstrained. (In practice, however, $|H(\omega')|$ should not be very large in these ϕ bands, e.g., not larger than L, to avoid amplification of any noise (or tails of $X(\omega)$) that may exist in these bands.) We will see later how these ϕ bands can have a significant effect on the filter design problem. Figure 3.27(d) shows the response of the ideal polyphase filter that is converted from an all-pass to a lowpass filter with cutoff frequency ω_c. Of course, the phase response of each polyphase filter is unaltered by the "don't care" bands.

As discussed in Section 3.2 for a decimator, the filter $H(\omega)$ should approximate the ideal lowpass characteristic

$$H_I(\omega) = \begin{cases} 1, & 0 \le \omega' \le \pi/M \\ 0, & \text{otherwise} \end{cases} \tag{3.100}$$

Alternatively, the polyphase filters should approximate the ideal all-pass characteristics

$$P_{\rho,I}(\omega) = \frac{1}{M} e^{j\omega\rho/M}, \qquad \rho = 0, 1, 2, \ldots, M-1 \tag{3.101}$$

If we are interested only in preventing aliasing in a band from 0 to ω_c, where $\omega_c < \pi/M$, and we are willing to tolerate aliased components for frequencies above ω_c, then we again have a situation where "don't care" bands are permitted in the filter design. The "don't care" regions are the same as those illustrated in Fig. 3.27(c) (with L replaced by M). In fact, all of the frequency-domain constraints that apply to the design of interpolation filters also apply to the design of decimation filters, a consequence of the property that they are transpose systems.

3.4.3 Time-Domain Properties of Ideal Interpolation and Decimation Filters

If we view the interpolation filter design problem in the time-domain, we obtain an alternative picture of the "ideal" interpolation filter. By taking the inverse transform of the ideal filter characteristic defined by Eq. (3.96) we get the well-known $\sin(x)/x$ characteristic

$$h_I(k) = \frac{\sin(\pi k/L)}{(\pi k/L)}, \qquad k = 0, \pm 1, \pm 2, \ldots \tag{3.102}$$

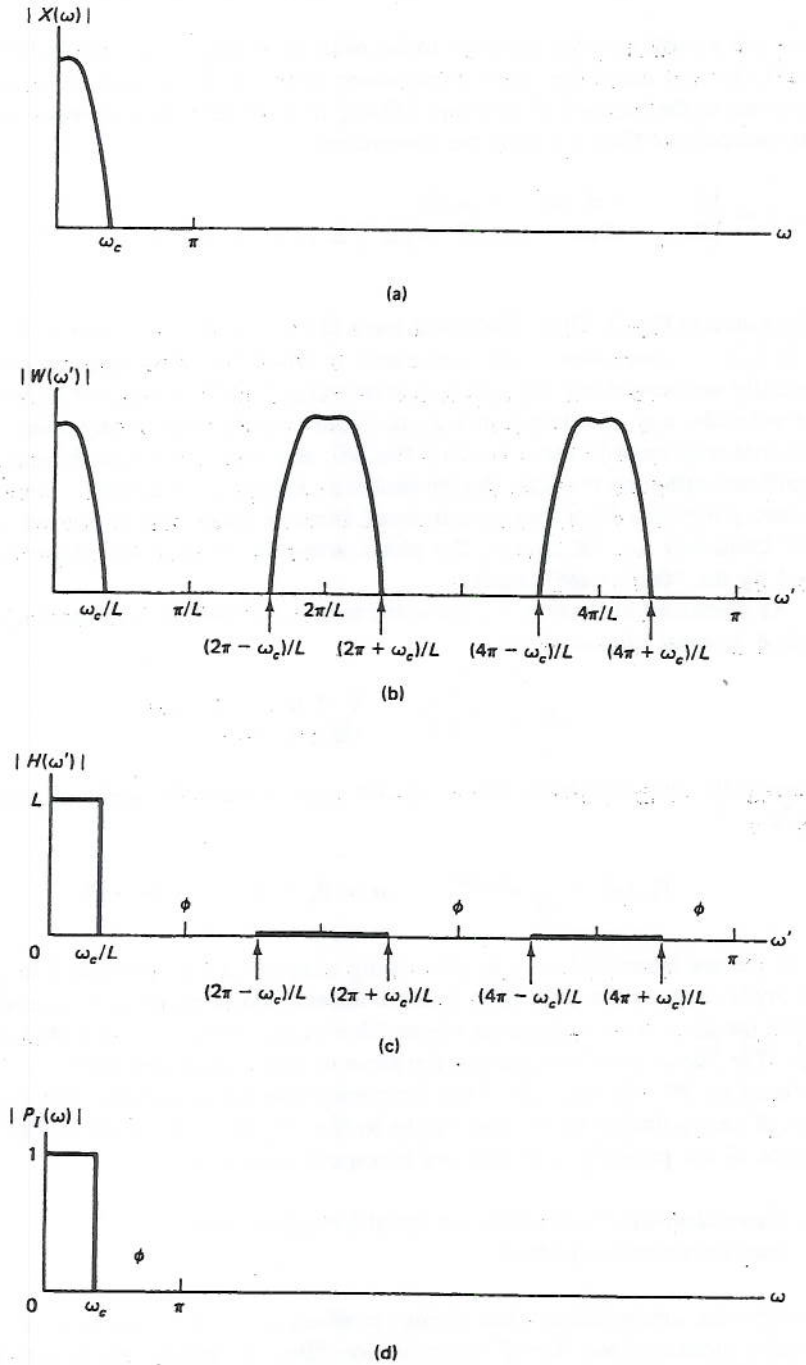


Figure 3.27 Illustration of ϕ bands in the specification of an interpolation filter ($L = 5$).

In a similar manner we can determine the ideal time responses of the polyphase filters, either by taking the inverse transform of Eq. (3.101) or by sampling the above time response $h_I(k)$ according to Eq. (3.84). The net result is that the ideal time responses of the polyphase filters are

$$p_{\rho,I}(n) = \frac{\sin[\pi(n + \rho/L)]}{\pi(n + \rho/L)}, \qquad \rho = 0, 1, 2, \ldots, L - 1, \quad \text{and all } n \tag{3.103}$$

A number of interesting observations can be made about the above ideal time responses. First we see that they constrain every Lth value of $h_I(k)$ such that

$$h_I(k) = \begin{cases} 1, & k = 0 \\ 0, & k = rL,\ r = \pm 1, \pm 2, \ldots \end{cases} \tag{3.104}$$

Alternatively, this implies the constraint that the zeroth-polyphase filter have an impulse response that is a unit pulse, i.e.,

$$p_{0,I}(n) = \delta(n), \qquad \text{for all } n \tag{3.105}$$

In terms of the polyphase structure of Fig. 3.17 and its signal processing interpretation in Fig. 3.18, the above constraint is easy to visualize. It simply implies that the output $y_0(m)$ of the zeroth polyphase branch is identical to the input $x(n)$ filled in with $L - 1$ zeros, i.e., these sample values are already known. The remaining $L - 1$ samples between these values must be interpolated by the polyphase filters $p_\rho(m)$, $\rho = 1, 2, \ldots, L - 1$. Since these filters are theoretically infinite in duration, they must be approximated, in practice, with finite-duration filters. Thus the interpolation "error" between the outputs of a practical system and an ideal system can be zero for $m = 0, \pm L, \pm 2L, \ldots$. However "in between" these samples, the error will always be nonzero.

By choosing a design that does not specifically satisfy the constraints of Eq. (3.104) or (3.105), we can make a tradeoff between errors that occur at sample times $m = 0, \pm L, \pm 2L, \ldots$, and errors that occur between these samples.

Another "time-domain" property that can be observed is that the ideal filter $h_I(k)$ is symmetric about zero, i.e.,

$$h_I(k) = h_I(-k) \tag{3.106}$$

(Alternatively, for practical systems it may be symmetric about some fixed nonzero delay.) This symmetry does not necessarily extend directly to the polyphase filters since they correspond to sample values of $h_I(k)$ offset by some fraction of a sample. Their envelopes, however, are symmetrical (see Fig. 3.18).

The ideal time responses for $h_I(k)$ and $p_{\rho,I}(n)$ for decimators are the same as those of Eqs. (3.102) and (3.103), respectively, with L replaced by M.

3.4.4 Filter Designs Based on Conventional Techniques

Many filter design techniques can be applied to the above ideal filter characteristics to achieve practical designs. They may be applied for a variety of reasons depending on the nature of the application and on the degree of accuracy necessary in meeting

desired error criteria. For example, window designs offer a simple, classical design procedure, and they have the property that they preserve the zero-crossing pattern of $h_I(k)$ in the actual design. They are limited, however, in the ability to control cutoff frequencies and stopband errors.

Equiripple filters have the advantage that highly efficient optimization techniques are available to design such filters with full control over the choice of error criteria and cutoff frequencies. They generally lead to very efficient designs, i.e., designs with a minimum filter length for a given tolerance specification. They can also be modified to accommodate the multiband design criteria illustrated in Fig. 3.27.

A third class of designs can be derived based on classical linear and Lagrange interpolation techniques. They are of interest from a historical point of view but are not widely used in modern digital signal processing.

Halfband filters are another class of designs of particular interest for interpolation or decimation by a factor of two. They are obtained by specifying the passband and stopband cutoff frequencies, ω_p, and ω_s, to be symmetrical about $\omega = \pi/2$, that is,

$$\omega_s = \pi - \omega_p \tag{3.107}$$

and the passband and stopband error tolerances δ_p and δ_s to be equal. This leads to filter designs with the symmetric property that

$$H(\omega) = 1 - H(\pi - \omega) \tag{3.108}$$

and with coefficient constraints

$$h(k) = \begin{cases} 1, & k = 0 \\ 0, & k = \pm 2, \pm 4, \ldots \end{cases} \tag{3.109}$$

The condition specified in Eq. (3.109) satisfies the zero-crossing criterion of ideal filters and results in efficient designs in that every other coefficient is zero and need not be computed in a practical implementation.

Comb filters are another class of designs that are used, particularly for multistage designs (to be discussed in Section 3.5). They are characterized by the impulse response

$$h(k) = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & \text{elsewhere} \end{cases} \tag{3.110}$$

where N is the length of the filter. N is usually chosen such that $N = M$ for decimation by M, or $N = L$ for interpolation by L. The frequency response of this class of filters can be shown to be

$$H(\omega) = \left| \frac{\sin(\omega N/2)}{\sin(\omega/2)} \right| \tag{3.111}$$

Under restricted conditions, comb filters can be applied to meet multiband requirements of the type illustrated in Fig. 3.27. When applicable, they are very useful in practical implementations since all coefficients are 1.0 and they can be efficiently implemented by simple sum-and-dump or sample-and-hold procedures.

Finally, a major class of filter designs that may be applied to decimation and

interpolation designs are the infinite impulse response (IIR) designs. They can be used when linear phase response is not required in the application. Numerous classical design techniques are available including well-known Butterworth, Bessel, Chebyshev, and elliptic designs.

3.4.5 Minimum Mean-Square Error Design of FIR Interpolator: Deterministic Signals

Thus far we have considered the design of decimation and interpolation filters based on conventional techniques applied to approximating the ideal choice of $h(k)$. In this section we consider an alternative point of view of filter design based on a minimum mean-square error criterion. In this approach the error criterion to be minimized is a function of the difference between the actual interpolated *signal* and its ideal value rather than a direct specification on the filter itself. We see in this section and in following sections that such an approach leads to a number of filter design techniques [17] that are capable of accounting directly for the spectrum of the signal being interpolated.

Figure 3.28(a) depicts the basic theoretical framework used for defining the above interpolator error criterion. We wish to design the FIR filter $h(m)$ such that it can be used to interpolate the signal $x(n)$ by a factor of L with minimum interpolation error. To define this error we need to compare the output of this actual interpolator with that of an ideal (infinite-duration) interpolator $h_I(m)$ whose characteristics were derived in Sections 3.4.2 and 3.4.3. This signal error is defined as

$$\Delta y(m) = y(m) - y_I(m) \tag{3.112}$$

where $y(m)$ is the output of the actual interpolator and $y_I(m)$ is the ideal output. We will consider interpolator designs that minimize the mean-square value of $\Delta y(m)$, defined as

$$E^2 = \|\Delta y(m)\|^2 = \lim_{K\to\infty} \frac{1}{2K+1} \sum_{m=-K}^{K} \Delta y(m)^2 \tag{3.113a}$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} |\Delta Y(\omega')|^2 \, d\omega' \tag{3.113b}$$

Alternative approaches lead to designs that minimize the maximum value of $|\Delta Y(\omega')|$ over a prescribed frequency range or to designs that minimize the maximum value of $|\Delta y((m))|$ in the time domain [10].

The above design problems are greatly simplified by considering them in the framework of the polyphase structures as illustrated in Fig. 3.17. Here we see that the signal $y(m)$ is actually composed of interleaved samples of the signals $u_\rho(n)$, $\rho = 0, 1, 2, \ldots, L-1$, as shown by Fig. 3.28(b), where $u_\rho(n)$ is the output of the ρth-polyphase filter. Thus the errors introduced by each polyphase branch are orthogonal to each other (since they do not coincide in time), and we can define the error in the ρth branch as the error between the actual output and the output of the ρth branch of an ideal polyphase interpolator as shown in Fig. 3.28(b), i.e.,

$$\Delta u_\rho(n) = u_\rho(n) - u_{\rho,I}(n) \tag{3.114}$$

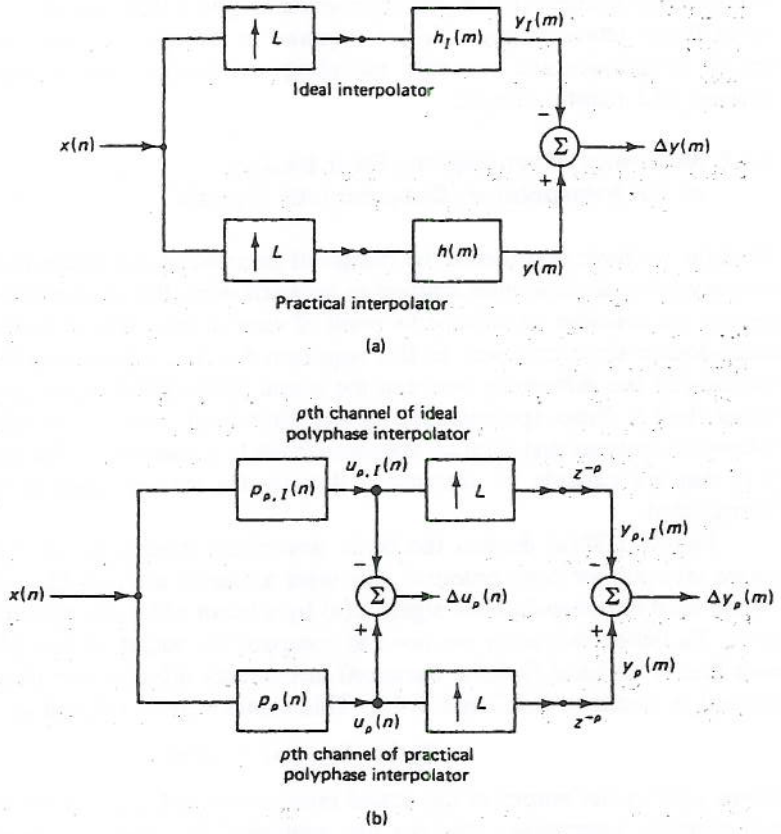


Figure 3.28 Framework for defining error criteria for interpolation filters.

Because of this orthogonality property we can *separately* and *independently* design each of the polyphase filters for minimum error and arrive at an overall interpolator design that minimizes the error $\|\Delta y(m)\|$. Thus a large (multirate) filter design problem can be broken down into L smaller (time-invariant) filter design problems.

In the case of the mean-square error criterion it can be seen that

$$E^2 = \|\Delta y(m)\|^2 = \frac{1}{L}\sum_{\rho=0}^{L-1} E_\rho^2 \tag{3.115}$$

where

$$E_\rho^2 = \|\Delta u_\rho(n)\|^2 \tag{3.116}$$

To minimize E^2 we then need to design L independent polyphase filters $p_\rho(n)$, $\rho = 0, 1, 2, \ldots, L-1$, which independently minimize the respective mean-square errors E_ρ^2.

To analytically set up the filter design problem, we can note that the ideal polyphase filter response is

$$P_{\rho,I}(\omega) = e^{j\omega\rho/L} \tag{3.117}$$

which then leads to the form

$$\begin{aligned} E_\rho^2 &= \|\Delta u_\rho(n)\|^2 \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |P_\rho(\omega) - e^{j\omega\rho/L}|^2 |X(\omega)|^2 \, d\omega \end{aligned} \tag{3.118}$$

Equation (3.118) reveals that in the minimum-mean-square error design, we are in fact attempting to design a polyphase filter such that the integral of the squared difference between its frequency response $P_\rho(\omega)$ and a linear (fractional sample) phase delay $e^{j\omega\rho/L}$, weighted by the spectrum of the input signal $|X(\omega)|^2$, is minimized. Note also that the integral from $-\pi$ to π in Eq. (3.118) is taken over the frequency range of the input signal of the interpolator, not the output signal.

In practice this error criterion is often modified slightly by specifying that $X(\omega)$ is bandlimited to the range $0 \le \omega \le \alpha\pi$, where $0 < \alpha < 1$, i.e.,

$$|X(\omega)| = 0, \qquad \text{for } |\omega| \ge \alpha\pi \tag{3.119}$$

Then Eq. (3.118) can be expressed as

$$E_{\rho,\alpha}^2 = \frac{1}{2\pi}\int_{-\alpha\pi}^{\alpha\pi} |P_\rho(\omega) - e^{j\omega\rho/L}|^2 |X(\omega)|^2 \, d\omega \tag{3.120}$$

where the subscript α is used to distinguish this norm from the one in Eq. (3.118). Alternatively we can consider the modification in Eq. (3.120) as a means of specifying that we want the design of $P_\rho(\omega)$ to be minimized only over the frequency range $0 \le \omega \le \alpha\pi$ and that the range $\alpha\pi \le \omega \le \pi$ is allowed to be a transition region. Then α can be used as a parameter in the filter design procedure.

The solution to the minimization problem of Eq. (3.120) involves expressing the norm $E_{\rho,\alpha}^2$ directly in terms of the filter coefficients $p_\rho(n)$. Then, since the problem is formulated in a classical mean-square sense, it can be seen that $E_{\rho,\alpha}^2$ is a quadratic function of the coefficients $p_\rho(n)$ and thus it has a single, unique minimum for some optimum choice of coefficients. At this minimum point, the derivative of $E_{\rho,\alpha}^2$ with respect to all the coefficients $p_\rho(n)$ is zero. Thus the second step in the solution is to take the derivative of $E_{\rho,\alpha}^2$ with respect to the coefficients $p_\rho(n)$ and set it equal to zero. This leads to a set of linear equations in terms of the coefficients $p_\rho(n)$, and the solution to this set of equations gives the optimum choice of coefficients that minimize $E_{\rho,\alpha}^2$. This minimization problem is solved for each value of ρ, $\rho = 0, 1, 2, \ldots, L-1$, and each solution provides the optimum solution for one of the polyphase filters. Finally, these optimum polyphase filters can be combined as in Eqs. (3.86) to (3.88) to obtain the optimum prototype filter $h(m)$ that minimizes the overall norm. The details for this approach can be found in [17]. Also, reference [17] contains a computer program that designs interpolation filters according to the above techniques and that greatly simplifies the task of designing these filters.

The minimum mean-square error interpolators designed using the procedure described have a number of interesting properties.

1. The resulting filters have the same symmetry properties as the ideal filters in Eq. (3.106).
2. The minimum error $\min E^2_{\rho,\alpha}$ for the polyphase filters also satisfies the symmetry condition

$$\min E^2_{\rho,\alpha} = \min E^2_{L-\rho,\alpha} \tag{3.121}$$

This error increases monotonically as ρ increases (starting with $E^2_{0,\alpha} = 0$) until $\rho = L/2$, at which point it decreases monotonically according to Eq. (3.121). Thus the greatest error occurs in interpolating sample values that are halfway between two given samples. This normalized error is closely approximated by the sine-squared function, i.e.,

$$\frac{\min E^2_{\rho,\alpha}}{\min E^2_{L/2,\alpha}} \approx \sin^2\left(\frac{\rho\pi}{L}\right) \tag{3.122}$$

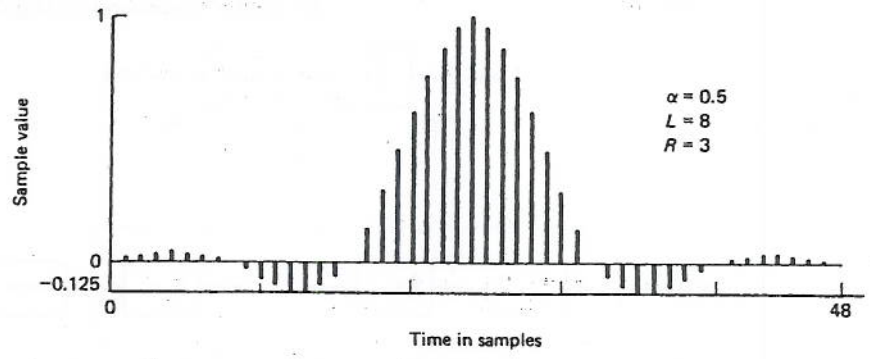
3. If an interpolator is designed for a given signal with a large value of L, all interpolators whose lengths are fractions of L are obtained by simply sampling the original filter; i.e., if we design an interpolator for $L = 100$, then for the same parameters α and R we can derive from this filter the optimum mean-square error interpolators for $L = 50$, 25, 20, 10, 5, and 2 by taking appropriate samples (or appropriate polyphase filters).

Figure 3.29 shows an example of the impulse response and frequency response for a minimum mean-square error interpolation filter with parameter values $\alpha = 0.5$, $N = 49$, $R = (N - 1)/2L = 3$, $L = 8$, and assuming that $|X(\omega)| = 1$.

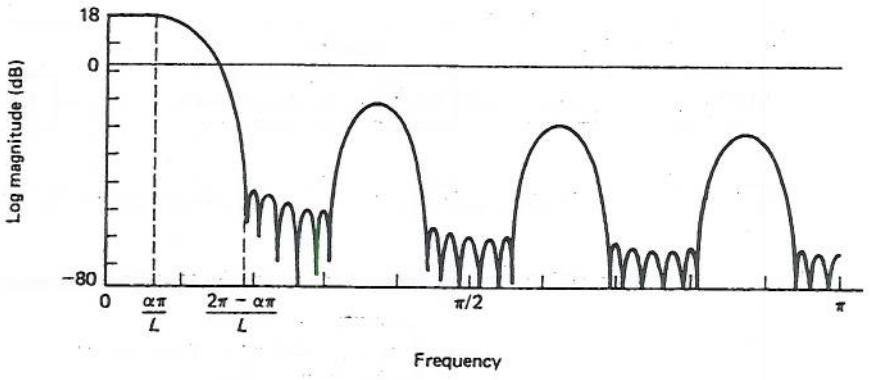
3.5 MULTISTAGE IMPLEMENTATIONS OF SAMPLING RATE CONVERSION

The concept of using a series of stages to implement a sampling rate conversion system can be extended to the case of simple interpolators and decimators [10], as shown in Fig. 3.30 and 3.31. Consider first a system for interpolating a signal by a factor of L as shown in Fig. 3.30(a). We denote the original sampling frequency of the input signal $x(n)$ as F_0, and the interpolated signal $y(m)$ has a sampling rate of LF_0. If the interpolation rate L can be factored into the product

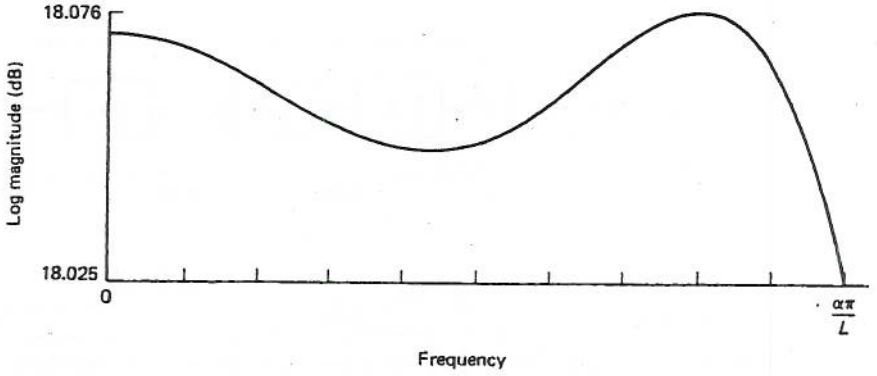
$$L = \prod_{i=1}^{I} L_i \tag{3.123}$$



(a)

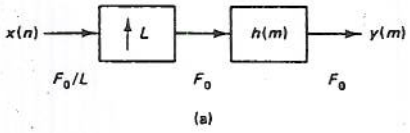


(b)

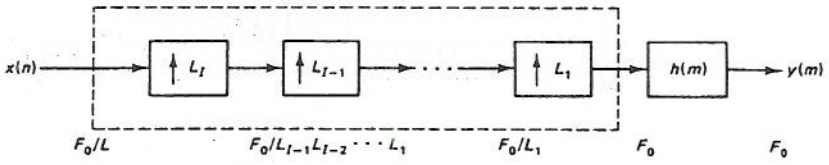


(c)

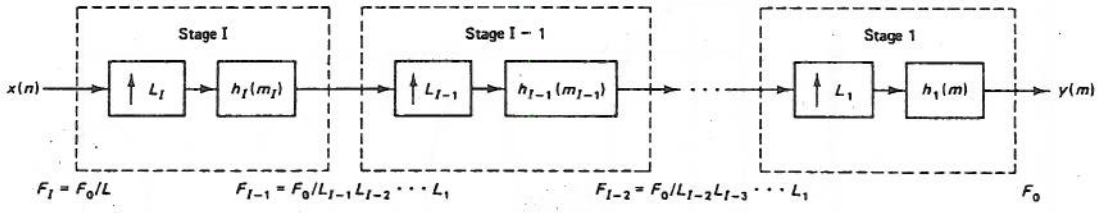
Figure 3.29 The impulse and frequency responses of a minimum mean-square error interpolation filter with $\alpha = 0.5$, $R = 3$, and $L = 8$.



(a)

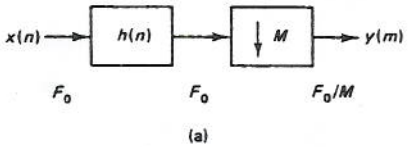


(b)

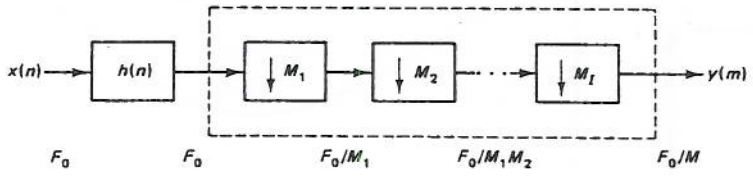


(c)

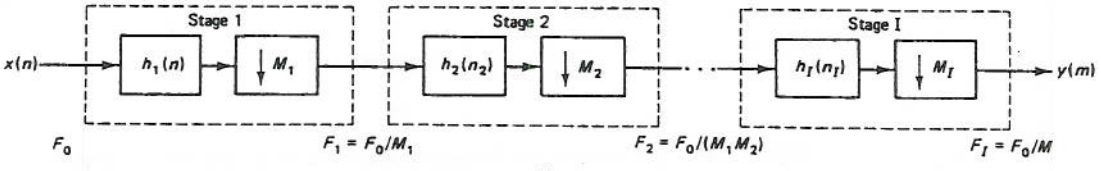
Figure 3.30 Steps in constructing a multistage interpolator for interpolation by a factor L.



(a)



(b)



(c)

Figure 3.31 Steps in constructing a multistage decimator for decimation by a factor M.

where each L_i is an integer, then we can express this network in the form shown in Fig. 3.30(b). This structure, by itself, does not provide any inherent advantage over the structure of Fig. 3.30(a). However, if we modify the structure by introducing a lowpass filter between *each pair* of the sampling rate increasing boxes, we produce the structure of Fig. 3.30(c). This structure has the property that the sampling rate increase occurs in a series of I stages, where each stage (shown within dashed boxes) is an *independent* interpolation stage.

Similarly, for an M-to-1 decimator, if the overall decimation rate M can be factored into the product

$$M = \prod_{j=1}^{J} M_j \tag{3.124}$$

then the general single-stage decimator structure of Fig. 3.31(a) can be converted into the multistage structure of Fig. 3.31(b). Again, each of the stages within the structure of Fig. 3.31(b) is an *independent* decimation stage.

Perhaps the most obvious question that arises from the preceding discussion is why we should consider such multistage structures. At first glance it would appear as if we are greatly increasing the overall computation (since we have inserted filters between each pair of stages) of the structure. This, however, is precisely the opposite of what occurs in practice. The reasons for considering multistage structures, of the types shown in Figs. 3.30(c) and 3.31(b), are as follows:

1. Significantly reduced computation to implement the system
2. Reduced storage in the system
3. Simplified filter design problem
4. Reduced finite word length effects, i.e., roundoff noise and coefficient sensitivity, in the implementations of the digital filters

These structures, however, are not without some drawbacks:

1. Increased control structure required to implement a multistage process
2. Difficulty in choosing the appropriate values of I (or J) of Eq. (3.123) and the best factors L_i (or M_j)

It is the purpose of this section to briefly show why and how a multistage implementation of a sampling rate conversion system can be (and generally is) more efficient than the standard single-stage structure for the following cases:

Case 1:	$L \gg 1$	$(M = 1)$
Case 2:	$M \gg 1$	$(L = 1)$
Case 3:	$L/M \approx 1$	but $L \gg 1, M \gg 1$

Cases 1 and 2 are high-order interpolation and decimation systems, and Case 3 occurs when a slight change in sampling rate is required (e.g., $L/M = 80/69$).

3.5.1 Computational Efficiency of a Two-Stage Structure: A Design Example

Since the motivation for considering multistage implementations of sampling rate conversion systems is the potential reduction in computation, it is worthwhile to present a simple design example that illustrates the manner in which the computational efficiency is achieved.

The design example is one in which a signal $x(n)$ with a sampling rate of 10,000 Hz is to be decimated by a factor of $M = 100$ to give the signal $y(m)$ at a 100-Hz rate. Figure 3.32(a) shows the standard single-stage decimation network that implements the desired process. It is assumed that the passband of the signal is from 0 to 45 Hz and that the band from 45 to 50 Hz is a transition band. Hence the specifications of the required lowpass filter are as shown in Fig. 3.32(b). We assume, for simplicity, that the design formula [10]

$$N \approx \frac{D(\delta_p, \delta_s)}{(\Delta F/F_s)} \tag{3.125}$$

can be used to give the order N of a symmetric FIR filter with maximum passband ripple δ_p, maximum stopband ripple δ_s, transition width ΔF, and sampling frequency F_s. For the lowpass filter of Fig. 3.32(b), we have

$$\Delta F = 50 - 45 = 5 \text{ Hz}$$
$$F_s = 10{,}000 \text{ Hz}$$
$$\delta_p = 0.01$$
$$\delta_s = 0.001$$
$$D(\delta_p, \delta_s) = 2.54$$

giving, from Eq. (3.125), $N \approx 5080$. The overall computation in multiplications per second (MPS) necessary to implement this system is

$$R = \frac{NF}{2M} = \frac{(5080)10{,}000}{2(100)} = 250{,}000 \text{ MPS}$$

i.e., a total of 250,000 MPS at the 10,000-Hz rate is required to implement the system of Fig. 3.32(a) (assuming the use of symmetry of $h(n)$).

Consider now the two-stage implementation shown in Fig. 3.32(c). The first stage decimates the signal by a factor of 50, and the second stage decimates the (already decimated) signal by a factor of 2, giving a total decimation factor of 100. The resulting filter specifications are illustrated in Fig. 3.32(d). For the first stage the passband is from 0 to 45 Hz, but the transition band extends from 45 to 150 Hz. Since the sampling rate at the output of the first stage is 200 Hz, the residual signal energy from 100 to 150 Hz gets aliased back into the range 50 to 100 Hz after decimation by the factor of 50. This aliased signal then gets removed in the second stage. For the second stage the passband extends from 0 to 45 Hz, and the transition band extends

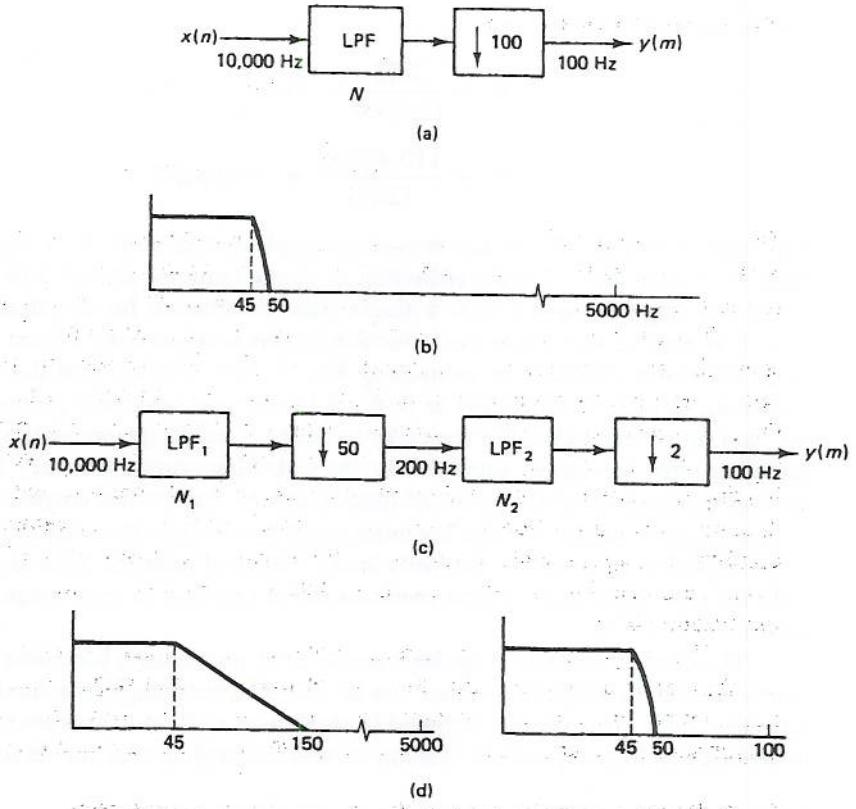


Figure 3.32 Simple example of (a) a one-stage and (c) a two-stage network for decimation by a factor of 100:1 and their respective filter design requirements (b and d).

from 45 to 50 Hz with a sampling rate of 200 Hz. One other change in the filter specifications occurs because we are using a two-stage filtering operation. The passband ripple specification of the two-stage structure is reduced to $\delta_p/2$ (since each stage can theoretically add passband ripple to each preceding stage). The stopband ripple specification does not change since the cascade of two lowpass filters reduces only the stopband ripples. Hence the $D(\delta_p, \delta_s)$ function in the filter design equation becomes $D(\delta_p/2, \delta_s)$ for the filters in the two-stage implementation. Since $D(\delta_p, \delta_s)$ is relatively insensitive to factors of 2, only slight changes occur (from 2.54 to 2.76) due to this factor. For the specific example of Fig. 3.32(c) we get (for the first stage)

$$N_1 = \frac{2.76}{(150 - 45)/10{,}000} = 263$$

$$R_1 = \frac{N_1 F}{2(M_1)} = \frac{263(10{,}000)}{(2)(50)}$$

$$= 26{,}300 \text{ MPS}$$

For the second stage we get

$$N_2 = \frac{2.76}{(5/200)} = 110.4$$

$$R_2 = \frac{110.4(200)}{(2)(2)} = 5500 \text{ MPS}$$

The total computation for the two-stage implementation is $R_1 + R_2 = 26{,}300 + 5500 = 31{,}800$ MPS. Thus a reduction in computation of almost 8 to 1 is achieved in the two-stage decimator over a single-stage decimation for this design example.

It is easy to see where the reduction in computation comes from for the multistage decimator structure by examining Eq. (3.125). We see that the required filter orders are directly proportional to $D(\delta_p, \delta_s)$ and F and inversely proportional to ΔF, the filter transition width. For the early stages of a multistage decimator, although the sampling rates are large, equivalently the transition widths are very large, thereby leading to relatively small values of filter length N. For the last stages of a multistage decimator, the transition width becomes small but so does the sampling rate, and the combination again leads to relatively small values of required filter lengths. We see from the preceding analysis that computation is kept low in each stage of the overall multistage structure.

The simple example presented above is by no means a complete picture of the capabilities and sophistication that can be found in multistage structures for sampling rate conversion. It is merely intended to show why such structures are of fundamental importance for many practical systems in which sampling rate conversion is required.

3.5.2 Design Considerations for Multistage Decimators and Interpolators

A large number of parameters and tradeoffs are involved in the design of multistage decimator and interpolator systems, including the following:

1. The number of stages to realize an overall decimation (or interpolation) factor M (or L) most efficiently
2. The choice of decimation ratios that are appropriate for each stage
3. The types of digital filters used in each stage
4. The structure used to implement filters in each stage
5. The required filter order in each stage
6. The resulting amount of computation, storage, and processing delay incurred in each stage and in the overall structure

As in most signal processing design problems, a number of factors influence each choice, and it is not a simple matter to make any one choice over all others. However, three general approaches and design philosophies have been applied to these designs. Each approach has slightly different advantages and disadvantages, and often a mix of these strategies is applied in practical applications to obtain the most effective design.

The first general approach is based on the formulation of the design problem in terms of a mathematically defined optimization problem with decimation (or interpolation) factors in each stage treated as continuous variables. The objective function to be minimized is then expressed as an analytical measure of the efficiency of the design. The design procedure is then performed by finding the most efficient solution (i.e., the best choice of decimation or interpolation factors for each stage) for each value of I (the number of stages) and then selecting that choice of I and its solution that give the best the overall solution [10].

A second approach is based on the use of halfband filter designs with a 2-to-1 decimation (or interpolation) factor for each stage. These filters have the advantage that approximately half of the filter coefficients are zero value (because of the halfband designs) and need not be implemented. This design procedure works best when the overall decimation or interpolation ratio is a power of two.

A third design approach combines the use of simple comb filters, where possible, in the initial stages of multistage decimators (or the final stages of multistage interpolators), followed by halfband filters and other special classes of filter designs. The idea here is to use a large number of stages to implement a large change in sampling rates and to use extremely simple linear phase FIR filters when possible.

In the preceding discussion we have very briefly outlined the issues and approaches used to design multistage decimation and interpolation systems. In addition to issues of overall computation rate, the cost of the control structure associated with multiple stages must be considered in the implementation of the design. As in most real-world problems there is no simple or universal answer about what design approach is best, and in practice a combination of these techniques often yields the most appropriate tradeoffs.

3.6 SIGNAL PROCESSING OPERATIONS BASED ON DECIMATION AND INTERPOLATION CONCEPTS

In the previous sections we discussed the basic concepts of sampling rate conversion and the issues involved in the efficient design of decimators and interpolators. In this section we show that these concepts not only are useful in designing efficient integer or ratio-of-integer sampling rate conversion systems but also can be applied more generally to the efficient design of a broad range of signal processing operations.

3.6.1 Sampling Rate Conversion Between Systems with Incommensurate Sampling Rates

So far we have considered multirate systems in which the various sampling rates within the system are related by exact integer or rational fraction ratios. That is, the sampling rates within the process are all generated from some common (higher-rate) clock. In practice, however, it is sometimes desired to interface digital systems that are controlled by *independent* (incommensurate) clocks. For example, it may be desired to exchange signals between two different systems, both of which are intended to be sampled at a rate F_s. However, due to practical limitations on the accuracy of the

clocks, the first system may be sampled at an actual rate of $F_s + \epsilon_1$ and the second system may be sampled at an actual rate of $F_s + \epsilon_2$ where ϵ_1 and ϵ_2 represent slowly varying components of drift as a function of time. If we simply exchange digital signals between these two systems (e.g., by means of a sample-and-hold process) samples may be either lost or repeated in the exchange due to the relative "slippage" of the two clocks. If the signals are highly uncorrelated from sample to sample (i.e., sampled at their Nyquist rates), this process can introduce large spikes or errors into the signals. The rate of occurrence of these errors is directly related to the amount of sampling rate slippage, that is, to the ratio $(\epsilon_1 - \epsilon_2)/F_s$.

One way to avoid the above problem is to interface the two digital systems through an analog connection, that is, to convert the signals to analog signals and then resample them with the new clock. In principle this process provides an error-free interface. In practice, however, it is limited by the practical capabilities and expense of the A/D and D/A conversion process as well as the dynamic range of the analog connection.

A more attractive all-digital approach to this problem can be accomplished by applying the multirate techniques discussed above to, in effect, duplicate the analog process in digital form. Figure 3.33 shows an example of a system for transferring a digital signal $x(n)$ from system 1 (with sampling rate $F_s + \epsilon_1$) to system 2 (with sampling rate $F_s + \epsilon_2$). The signal $x(n)$ is first interpolated by a 1-to-L interpolator (where $L \gg 1$) to produce the highly oversampled signal $y(m)$ (at sampling rate $L(F_s + \epsilon_1)$). This signal is then converted to a signal $\hat{y}(m)$ (at the sampling rate $L(F_s + \epsilon_2)$) through a digital sample-and-hold procedure that interfaces the two incommensurate sampling rates. It is then decimated by an L-to-1 decimator to produce the signal $\hat{x}(n)$ for input to system 2. The interface between the oversampled digital signals $y(m)$ and $\hat{y}(m)$ plays the same role as that of an analog interface and as $L \to \infty$ it is equivalent. The advantage is that the system is all-digital and the accuracy of the conversion can be designed with any degree of desired precision.

Although samples may be repeated or dropped in the sample-and-hold conversion process between $y(m)$ and $\hat{y}(m)$, it can be shown that the effects of these errors become small as L becomes large. This can be seen by considering the sample-to-sample difference of the signal $y(m)$ (or $\hat{y}(m)$) relative to its actual value as a function of L [10]. The sampling rate conversion can be efficiently realized using the structures and filter designs in Sections 3.3–3.5. In particular, for large values of L, the multi-

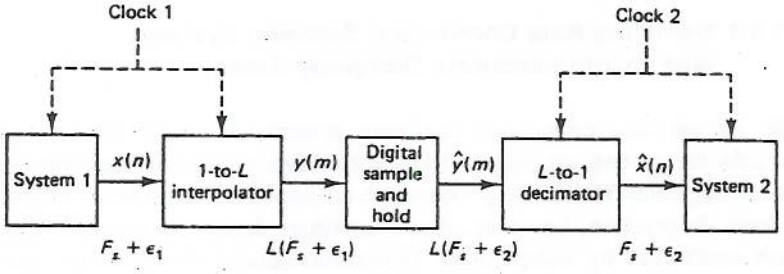


Figure 3.33 A multirate approach to sampling rate conversion between systems with incommensurate sampling rates.

stage designs are appropriate. Also it should be noted that although the system in Fig. 3.33 is described in terms of a conversion process between two "equivalent" but incommensurate sampling rates $F_s + \epsilon_1$ and $F_s + \epsilon_2$, it can be readily extended to a process of conversion between any two incommensurate sampling rates by choosing different ratios for the interpolator and decimator. It is necessary only that the signals $y(m)$ and $\hat{y}(m)$ have "equivalent" rates to minimize the amount of sample repeating or dropping in the sample-and-hold process.

3.6.2 Design of Fractional Sample Phase Shifters Based on Multirate Concepts

Many signal processing applications require a network that essentially delays the input signal by a fixed number of samples. When the desired delay is an integer number of samples, at the current sampling rate, such a network is trivially realized as a cascade of unit delays. However, when delays of a fraction of a sample are required, the processing required to achieve such a delay is considerably more difficult. We show here how multirate signal processing concepts can be used to greatly simplify the process required to design noninteger delay as long as the desired delay is a rational fraction of a sample.

Consider the ideal delay network $h_{AP}(n)$ of Fig. 3.34(a). The desired allpass network processes $x(n)$ to give the output $y(n)$ so that the relationship between their Fourier transforms is

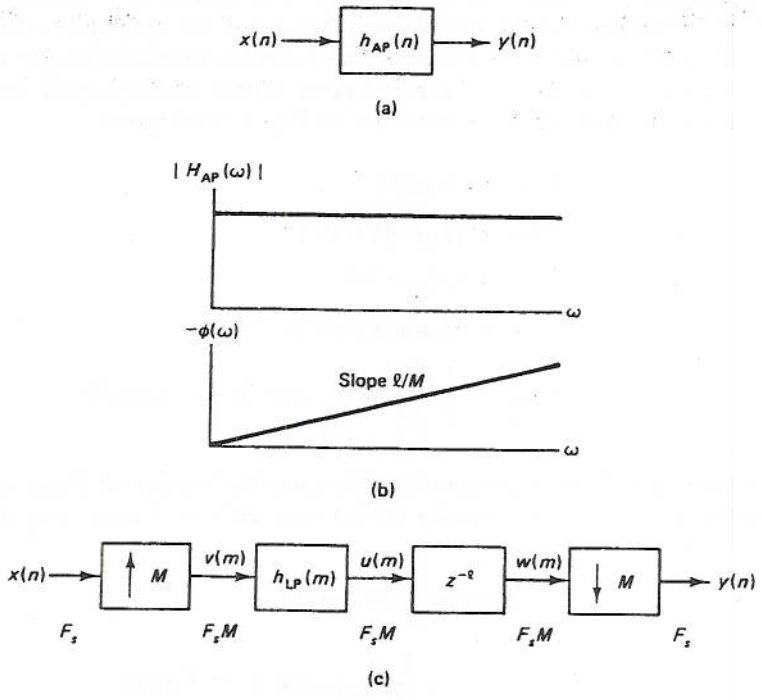


Figure 3.34 A multirate structure for realizing a fixed delay of ℓ/M samples.

$$Y(\omega) = e^{-j\omega\ell/M} X(\omega) \tag{3.126}$$

where ℓ and M are any integers. In the time domain this amounts to a delay of the envelope of the signal $x(n)$ by a fraction of a sample ℓ/M. The magnitude and phase responses of $h_{AP}(n)$, shown in Fig. 3.34, are of the form

$$H_{AP}(\omega) = |H_{AP}(\omega)| e^{j\phi(\omega)} \tag{3.127a}$$

where

$$|H_{AP}(\omega)| = 1 \tag{3.127b}$$

$$\phi(\omega) = -\frac{\ell\omega}{M} \tag{3.127c}$$

and where $\phi(\omega)$ denotes the phase of $H_{AP}(\omega)$. (At this point the observant reader should notice the similarity between the desired response and that of a polyphase filter. We return to this equivalence later in this section.) It should be clear that for arbitrary values of ℓ and M, the desired frequency response of Eqs. (3.127) cannot be achieved exactly by an FIR or an IIR filter. An FIR filter (all zeros) cannot achieve an exact allpass magnitude response, and an IIR filter (poles and zeros) cannot achieve an exact linear phase response. Thus the desired noninteger delay network $h_{AP}(n)$ cannot be realized exactly but can only be approximated through a design procedure.

Using multirate principles, this design problem can be clearly defined as illustrated in Fig. 3.34(c). The key to this procedure is the realization that a delay of ℓ/M samples at rate F_s is equivalent to a delay of ℓ samples (i.e., an integer delay) at rate F_sM. Hence the structure in Fig. 3.34(c) first raises the sampling rate of the signal to F_sM, filters the signal with a lowpass filter $h_{LP}(m)$ to eliminate images of $x(n)$, delays the signal by ℓ samples, and then decimates it back to the original sampling rate.

A simple analysis of the structure of Fig. 3.34(c) gives

$$V(\omega) = X(\omega M) \tag{3.128}$$

$$U(\omega) = H_{LP}(\omega) X(\omega M) \tag{3.129}$$

$$\begin{aligned} W(\omega) &= U(\omega) e^{-j\omega\ell} \\ &= H_{LP}(\omega) X(\omega M) e^{-j\omega\ell} \end{aligned} \tag{3.130}$$

$$Y(\omega) = \frac{1}{M} \sum_{r=0}^{M-1} W((-2\pi r/M) + (\omega/M)) \tag{3.131}$$

We assume that $H_{LP}(\omega)$ sufficiently attenuates the images of $X(\omega)$ so that they are negligible in Eq. (3.131), thereby giving only the $r = 0$ term, that is,

$$\begin{aligned} Y(\omega) &\approx \frac{1}{M} W(\omega/M) \\ &\approx \frac{1}{M} H_{LP}(\omega/M) e^{-j\omega\ell/M} X(\omega) \end{aligned} \tag{3.132}$$

We further assume that $H_{LP}(\omega)$ is an FIR filter with exactly linear phase, whose delay (at the high rate) is $(N-1)/2$ samples, and this value is chosen to be an integer delay at the low rate, that is,

$$\frac{N-1}{2} = IM \tag{3.133}$$

or

$$N = 2IM + 1 \tag{3.134}$$

We also require $H_{LP}(\omega)$ to have a magnitude response essentially equal to M (to within a small tolerance) in the passband, thereby giving for $Y(\omega)$

$$Y(\omega) \approx e^{-j\omega I} e^{-j\omega \ell/M} X(\omega) \tag{3.135}$$

or, as an equivalent network,

$$\frac{Y(\omega)}{X(\omega)} \approx e^{-j\omega I} e^{-j\omega \ell/M} \tag{3.136}$$

Thus the structure of Fig. 3.34(c) is essentially an allpass network with a fixed integer delay of I samples and a variable, noninteger delay of ℓ/M samples.

One efficient implementation of the multirate allpass filter of Fig. 3.34(c) is given in Fig. 3.35. A polyphase structure is used to realize the sampling rate increase and lowpass filtering based on the counterclockwise commutator structure where

$$p_p(n) = h_{LP}(nM + \rho), \qquad 0 \le \rho \le M-1 \tag{3.137}$$

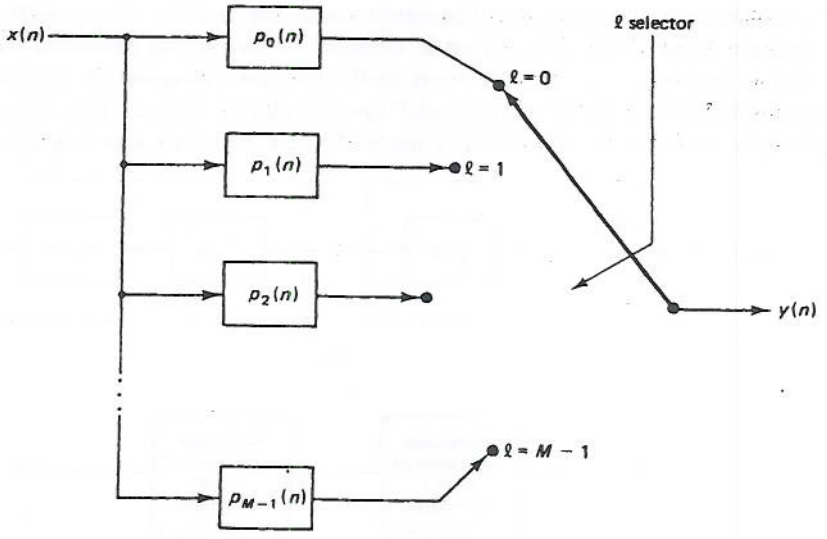


Figure 3.35 A polyphase network implementation of a fractional sample delay network.

The delay of ℓ samples is implemented as a new initial position of the commutator corresponding to the $m = 0$ sample. Finally, the decimation by M is implemented as a fixed arm position of the commutator, since it is back at the original position each M samples.

Thus for a single *fixed* delay of ℓ/M samples, only one branch (corresponding to the $\rho = \ell$th polyphase filter branch) is required, and for a network that requires all possible values of ℓ (from 0 to $M - 1$) the entire network of Fig. 3.35 is required, i.e., it represents a selectable choice of M different fractional delays.

A key point made in this section is that the design of an N-tap time-invariant filter with a noninteger delay (and flat magnitude response), such as $h_{AP}(n)$ in Fig. 3.34(a), can be readily transformed to that of an NM-tap lowpass filter design. This transformation is accomplished by means of an appropriate multirate interpretation of the problem and an application of the concept of polyphase filters.

3.6.3 Multirate Implementation of Lowpass Filters

In Section 3.2 we introduced the idea of cascading a 1-to-L interpolator with an M-to-1 decimator and showed that the resulting structure implemented a sampling rate conversion by a factor of L/M. Consider now cascading an M-to-1 decimator with a 1-to-M interpolator as shown in Fig. 3.36(a). Intuitively, we see that the overall system relating the output $y(n)$ to the input $x(n)$ acts like a lowpass filter (due to $h_1(n)$ and $h_2(n)$) with aliasing (due to the decimation) and imaging (due to the interpolation). When these components are appropriately removed by filters $h_1(n)$ and $h_2(n)$ it can be shown that the overall system acts like a well-behaved lowpass digital filter [10].

This same line of reasoning can be applied to the multirate lowpass system of Fig. 3.36(b). It can be shown that as the bandwidth of the resulting lowpass filter (relative to the sampling rate) becomes small, the benefits of the multistage approach become large. Since the design is identical to that of the cascaded multistage decimators and interpolators, the same methodologies discussed in Section 3.5 apply to the efficient design of narrowband lowpass digital filters. This approach leads to designs that can be significantly more efficient and also less sensitive to effects of

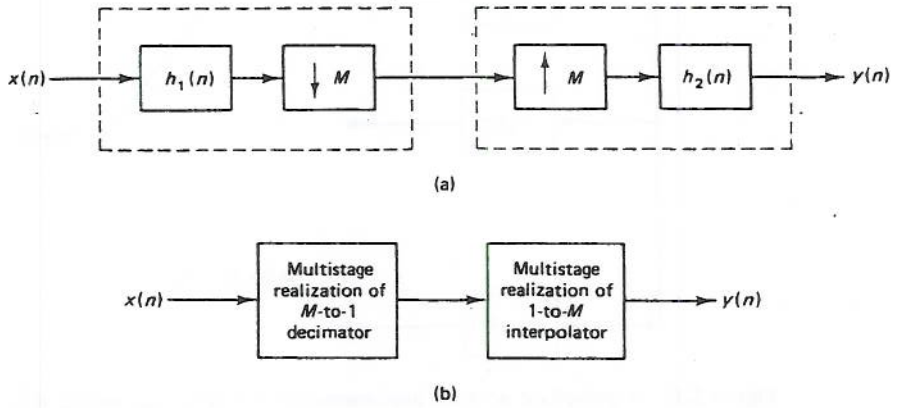


Figure 3.36 Block diagram of a multistage realization of a multirate lowpass filter structure

coefficient quantization and roundoff noise than classical implementations. To illustrate the above concepts more explicitly we conclude this section with a design example that illustrates the gains in efficiency attainable using a multirate, multistage implementation of a narrowband lowpass filter over a standard, direct-form implementation.

Consider the design of a narrowband lowpass filter with specifications $F_s = 1.0$, $f_p = \omega_p/2\pi = 0.00475$, $f_s = \omega_s/2\pi = 0.005$, $\delta_p = 0.001$, and $\delta_s = 0.0001$. Clearly, this filter is a very narrowband lowpass filter with very tight ripple specifications.

For the single-rate, standard, direct-form FIR implementation, the *estimate* of filter order, based on the standard equiripple design formula [2] is $N_0 = 15{,}590$; that is, an extremely high-order FIR filter is required to meet these filter specifications. Such a filter would never be designed in practice, and even if it could be designed, the implementation would yield excessive roundoff noise and therefore would not be useful. However, for theoretical purposes, we postulate such a filter and compute its multiplication rate (using symmetry of the impulse response) as

$$R_0 = \frac{N_0}{2} = 7795 \text{ MPS}$$

For a multirate implementation, a decimation ratio of $M = 100 = 0.5/0.005$ can be used. For a one-stage implementation of the decimator and interpolator as shown in Fig. 3.36(a), the (estimated) required filter order for $h_1(n)$ and $h_2(n)$ is $N_1 = 16{,}466$. Again, we could never really design such a high-order filter, but, for theoretical purposes, we can compute its multiplication rate (again employing symmetry in both the decimator and interpolator) to give

$$R_1 = \frac{N_1}{2(100)}(2) = 165 \text{ MPS}$$

resulting in a potential savings (of multiplications per second) of about 47.2 to 1 over the direct-form implementation.

For a two-stage implementation of the decimator and interpolator (i.e., two stages of decimation followed by two stages of interpolation), a reasonable set of ratios for decimation is $M_1 = L_1 = 50$, $M_2 = L_2 = 2$, resulting in filter orders of $N_1 = 423$ and $N_2 = 347$. The total multiplication rate for the two-stage structure is

$$R_2 = 2\left[\frac{N_1}{2(50)} + \frac{N_2}{2(100)}\right] = 11.9 \text{ MPS}$$

resulting in a potential savings of about 655 to 1 over the direct-form structure.

Finally, if we use a three-stage implementation of the decimator and interpolator, a reasonable set of ratios is $M_1 = L_1 = 10$, $M_2 = L_2 = 5$, $M_3 = L_3 = 2$, resulting in filter orders of $N_1 = 50$, $N_2 = 44$, and $N_3 = 356$. The total multiplication rate is then

$$R_3 = 2\left[\frac{N_1}{2(10)} + \frac{N_2}{2(50)} + \frac{N_3}{2(100)}\right] = 9.4 \text{ MPS}$$

resulting in a savings of 829 to 1 over the direct-form implementation.

By way of comparison, an elliptic filter meeting the given filter design specifications is of 14th order and, in a cascade realization, requires 22 MPS. This shows that for the given design example, the three-stage FIR design is about 2.3 times more efficient than a single-stage, fixed-rate IIR filter (and it has linear phase). However, it requires a significantly larger amount of storage for coefficients and data than does the IIR design.

A summary of these results (for the FIR filters) is presented in Table 3.1. The key point to note is the spectacular gains in efficiency that are readily achievable by a multistage, multirate implementation over a direct FIR filter implementation. Furthermore, we see that, for the three-stage structure, the resulting FIR designs can be readily achieved and implemented without problem due to excessive length of the impulse response or the excessive roundoff noise in the implementation. Thus we have achieved both an efficient implementation *and* an efficient method of designing very long, very narrow-bandwidth lowpass filters with tight ripple tolerances.

TABLE 3.1 COMPARISONS OF FILTER CHARACTERISTICS FOR SEVERAL MULTISTAGE IMPLEMENTATIONS OF A LOWPASS FILTER WITH SPECIFICATIONS $F_s = 1.0$, $f_p = 0.00475$, $f_s = 0.005$, $\delta_p = 0.001$, $\delta_s = 0.0001$

	Direct Form	One-Stage	Two-Stage	Three-Stage
Decimation rates	—	100	50 2	10 5 2
Filter lengths	15,590	16,466	423 347	50 44 356
MPS	7795	165	11.9	9.4
Rate reduction (MPS)	1	47.2	655	829
Total storage for filter coefficients	7795	8233	385	225

3.6.4 Sampling Rate Conversion Applied to Bandpass Signals

Until now we have assumed that the signals with which we are dealing are lowpass signals and that the filters required for decimation and interpolation are therefore lowpass filters that preserve the baseband signals of interest. In many practical systems, however, it is often necessary to deal with bandpass signals as well as lowpass signals. In this section we show how the concepts of decimation and interpolation can be applied to systems in which bandpass signals are present.

Figure 3.37(a) shows an example of the discrete-time Fourier transform of a digital bandpass signal $S(2\pi fT)$ that contains spectral components only in the frequency range $f_\ell < |f| < f_\ell + f_\Delta$. If we apply directly the concepts of lowpass sam-

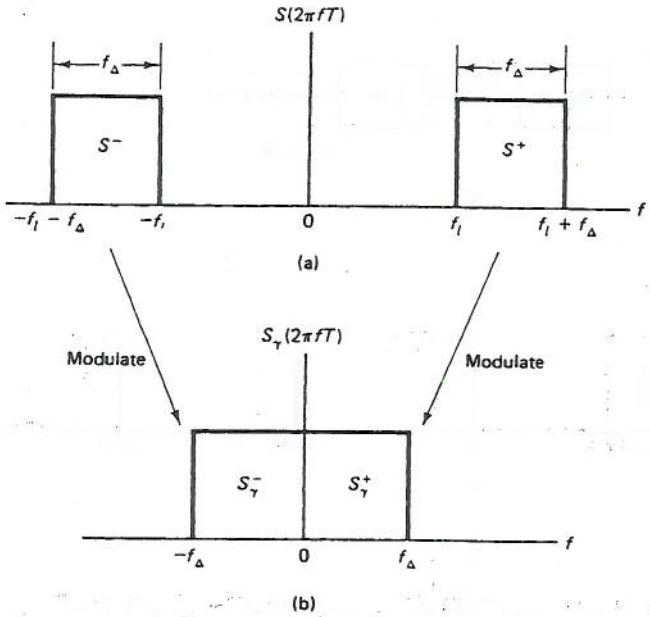


Figure 3.37 Bandpass signal and its lowpass translated representation.

pling, we see that the sampling rate, F_w, necessary to represent this signal must be twice that of the highest-frequency component in $S(2\pi fT)$, that is, $F_w \geq 2(f_l + f_\Delta)$. Alternatively, let S^+ denote the component of $S(2\pi fT)$ associated with $f > 0$ and S^- denote the component of $S(2\pi fT)$ associated with $f < 0$, as seen in Fig. 3.37. Then, by lowpass translating (modulating) S^+ to the band 0 to f_Δ and S^- to the band $-f_\Delta$ to 0, as illustrated by Fig. 3.37(b), we see that a new signal $S_\gamma(2\pi fT)$ can be generated that is "equivalent" to $S(2\pi fT)$ in the sense that $S(2\pi fT)$ can uniquely be reconstructed from $S_\gamma(2\pi fT)$ by the inverse process of bandpass translation. (Actually, we see that $S(2\pi fT)$ is the "single-sideband" modulated version of $S_\gamma(2\pi fT)$.) By applying concepts of lowpass sampling to $S_\gamma(2\pi fT)$, however, we can see that the sampling frequency necessary to represent this signal is now $F_\Delta \geq 2f_\Delta$, which can be much lower than the value of F_w specified above (if $f_l \gg f_\Delta$). Thus we see that by an appropriate combination of modulation followed by lowpass sampling, any real bandpass signal with (positive-frequency) bandwidth f_Δ can be uniquely sampled at a rate $F_\Delta \geq 2f_\Delta$ (i.e., such that the original bandpass signal can be uniquely reconstructed from the sampled representation).

Perhaps the simplest and most direct approach to decimating or interpolating digital bandpass signals, sometimes referred to as *integer-band sampling*, is to take advantage of the inherent frequency translating (i.e., aliasing or imaging) properties of decimation and interpolation. As discussed in Section 3.1, sampling and sampling rate conversion can be viewed as a modulation process in which the spectrum of the digital signal contains periodic repetitions of the baseband signal (images) spaced at harmonics of the sampling frequency. This property can be used to advantage when dealing with bandpass signals by associating the bandpass signal with one of these images instead of with the baseband.

Figure 3.38(a) illustrates an example of this process for the use of decimation

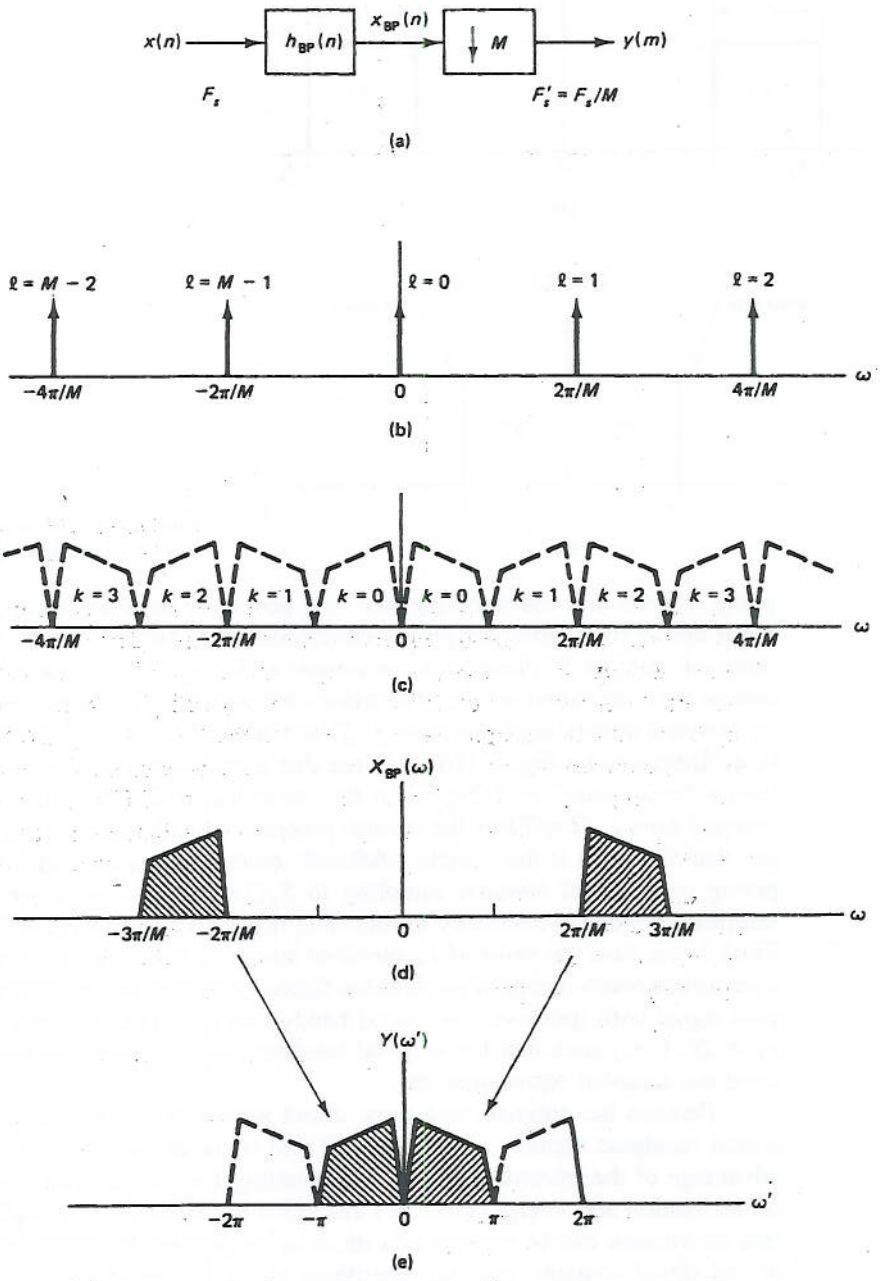


Figure 3.38 Integer-band decimation and a spectral interpretation for the $k = 2$ band.

by the integer factor M. The input signal $x(n)$ is first filtered by the bandpass filter $h_{BP}(n)$ to isolate the frequency band of interest. The resulting bandpass signal, $x_{BP}(n)$, is then directly recduced in sampling rate by an M-sample compressor giving the final output, $y(m)$. We see that this system is identical to that of the integer lowpass decimator discussed in Section 3.2.1, with the exception that the filter is a bandpass filter rather than a lowpass filter. Thus the output signal $Y(\omega')$ can be expressed as

$$Y(\omega') = \frac{1}{M}\sum_{\ell=0}^{M-1} H_{BP}((\omega' - 2\pi\ell)/M)X((\omega' - 2\pi\ell)/M) \tag{3.138}$$

From Eq. (3.138) we see that $Y(\omega')$ is composed of M aliased components of $X(\omega')H_{BP}(\omega')$ modulated by factors of $2\pi\ell/M$. The function of the filter $H_{BP}(\omega)$ is to remove (attenuate) all aliasing components except those associated with the desired band of interest. Since the modulation is restricted to values of $2\pi\ell/M$, we can see that only specific frequency bands are allowed by this method. As a consequence, the choice of the filter $H_{BP}(\omega)$ is restricted to approximate one of the M ideal characteristics

$$H_{BP,I}(\omega) = \begin{cases} 1, & k\frac{\pi}{M} < |\omega| < (k+1)\frac{\pi}{M} \\ 0, & \text{otherwise} \end{cases} \tag{3.139}$$

where $k = 0, 1, 2, \ldots, M-1$; that is $H_{BP}(\omega)$ is restricted to bands $\omega = k\pi/M$ to $\omega = (k+1)\pi/M$, where π/M is the bandwidth.

Figures 3.38(b)–(e) illustrate this approach. Figure 3.38(b) shows the M possible modulating frequencies that are a consequence of the M-to-1 sampling rate reduction; that is, the digital sampling function (a periodic train of unit samples spaced M samples apart) has spectral components spaced $2\pi\ell/M$ apart. Figure 3.38(c) shows the "sidebands" that are associated with these spectral components, which correspond to the M choices of bands as defined by Eq. (3.139). They correspond to the bands that are aliased into the baseband of the output signal $Y(\omega')$ according to Eq. (3.139).

Figure 3.38(d) illustrates an example in which the $k = 2$ band is used, such that $X_{BP}(\omega)$ is bandlimited to the range $2\pi/M < |\omega| < 3\pi/M$. Since the process of sampling rate compression by M to 1 corresponds to a convolution of the spectra of $X_{BP}(\omega)$ (Fig. 3.38d) and the sampling function (Fig. 3.38b), this band is lowpass translated to the baseband of $Y(\omega')$ as seen in Fig. 3.38(e). Thus the processes of modulation and sampling rate reduction are achieved simultaneously by the M-to-1 compressor.

Figure 3.39 illustrates a similar example for the $k = 3$ band such that $X_{BP}(\omega)$ is bandlimited to the band $3\pi/M < |\omega| < 4\pi/M$. In this case it is seen that the spectrum is inverted in the process of lowpass translation. If the noninverted representation of $y(m)$ is desired, it can easily be achieved by modulating $y(m)$ by $(-1)^m$ (i.e., $\bar{y}(m) = (-1)^m y(m)$), which corresponds to inverting the signs of odd samples of $y(m)$. In general, bands associated with even values of k are directly lowpass translated to the baseband of $Y(\omega')$, whereas bands associated with odd values of k are translated and inverted. This is a consequence of the fact that even-numbered bands (k even) correspond to "upper sidebands" of the modulation frequencies $2\pi\ell/M$, whereas odd-numbered bands (k odd) correspond to "lower sidebands" of the modulation frequencies.

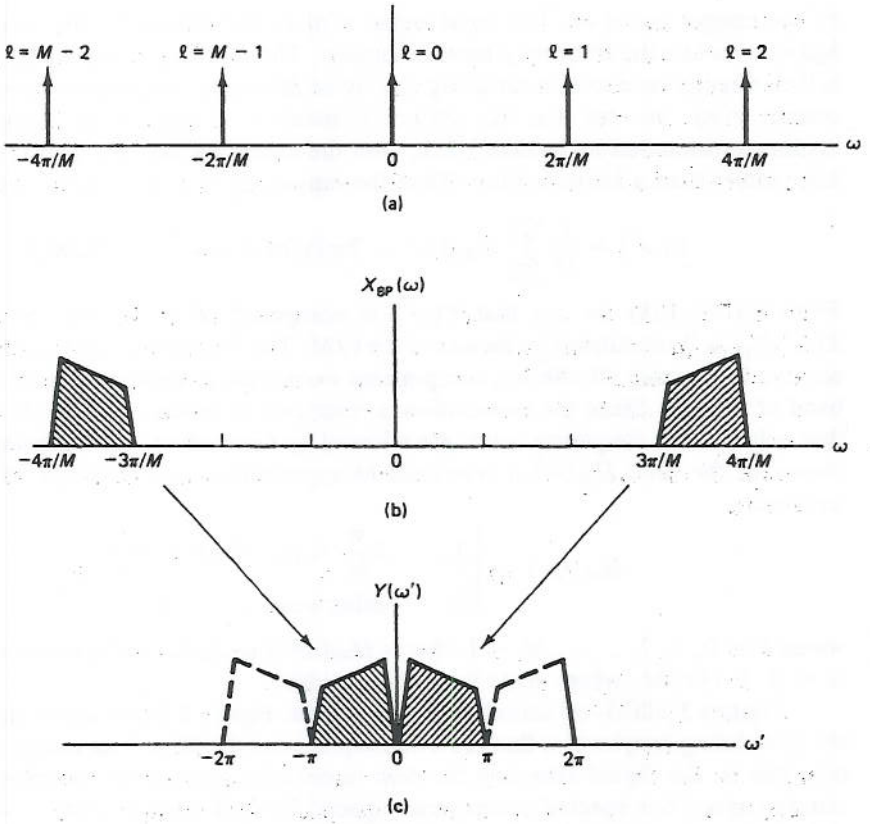


Figure 3.39 Spectral interpretation of integer-band decimation for the band $k = 3$.

The process of integer-band interpolation is the transpose of that of integer-band decimation; that is, it performs the reconstruction (interpolation) of a bandpass signal from its integer-band decimated representation. Figure 3.40(a) illustrates this process. The input signal, $x(n)$, is sampling rate expanded by L to produce the signal $w(m)$. The spectrum of $w(m)$ can be expressed as

$$W(\omega') = X(\omega' L) \tag{3.140}$$

and it corresponds to periodically repeated images of the baseband of $X(\omega)$ centered at the harmonics $\omega' = 2\pi\ell/L$. A bandpass filter $h_{BP}(m)$ is then used to select the appropriate image of this signal. We can see that to obtain the kth image, the bandpass filter must approximate the ideal characteristic

$$H_{BP,I}(\omega') = \begin{cases} L, & k\dfrac{\pi}{L} < |\omega'| < (k+1)\dfrac{\pi}{L} \\ 0, & \text{otherwise} \end{cases} \tag{3.141}$$

where $k = 0, 1, 2, \ldots, L - 1$. Figure 3.40(d) shows an example of the output spectrum of the bandpass signal $Y(\omega')$ for the $k = 2$ band, and Fig. 3.40(e) illustrates

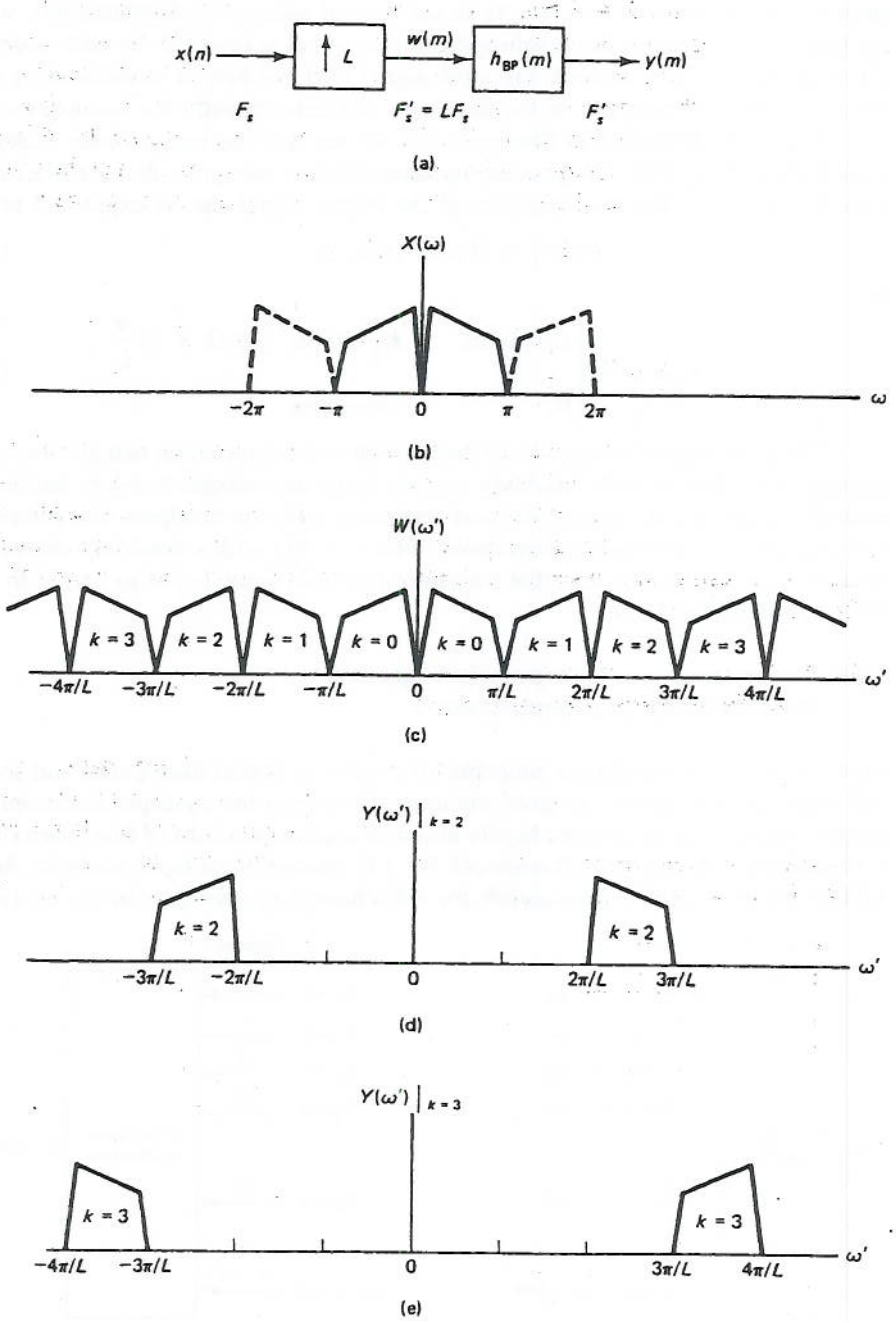


Figure 3.40 Spectral interpretation of integer-band interpolation of bandpass signals.

an example for the $k = 3$ band. As in the case of integer-band decimation, we also see that the spectrum of the resulting bandpass signal is inverted for odd values of k. If this inversion is not desired, the input signal $x(n)$ can first be modulated by $(-1)^n$, which inverts the spectrum of the baseband and consequently the bandpass signal.

From the discussion in Section 3.2.2 we see that the integer-band interpolator is identical to that of the lowpass interpolator with the exception that the filter may be a bandpass filter. Thus the spectrum of the output signal can be expressed as

$$Y(\omega') = H_{BP}(\omega')X(\omega'L) \tag{3.142}$$

or

$$Y(\omega') \approx \begin{cases} LX(\omega'L), & k\frac{\pi}{L} < |\omega'| < (k+1)\frac{\pi}{L} \\ 0, & \text{otherwise} \end{cases} \tag{3.143}$$

The processes of integer-band decimation and interpolation can also be used for translating or modulating bandpass signals from one integer band to another. For example, a cascade of an integer-band decimator with the bandpass characteristic for $k = 2$ and an integer-band interpolator (with $L = M$) with a bandpass characteristic for $k = 3$ results in a system that translates the band from $2\pi/M$ to $3\pi/M$ to a band from $3\pi/M$ to $4\pi/M$.

3.6.5 Decimation and Interpolation Applied to Filter Bank Implementations

Decimation and interpolation concepts often arise in digital filter banks and spectrum analyzers. In this section we give one example of how the concepts discussed in this chapter can be used to achieve highly efficient implementations of filter banks. Figure 3.41(a) illustrates the basic framework for a K-channel filter bank analyzer, and Fig. 3.41(b) shows a similar framework for a K-channel synthesizer. In the analyzer the

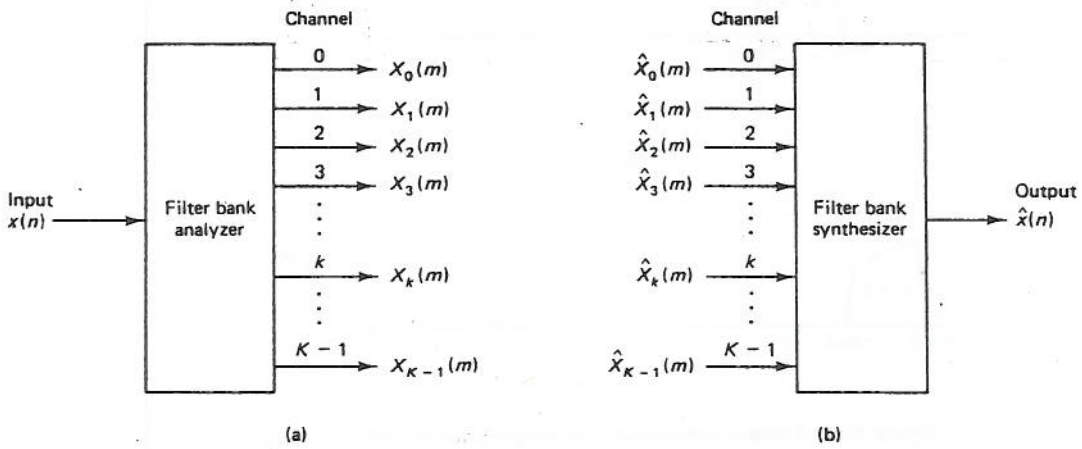


Figure 3.41 Basic framework for (a) a K-channel filter bank analyzer and (b) a K-channel filter bank synthesizer.

input signal $x(n)$ is divided into a set of K spectral components or channel signals denoted as $X_k(m)$, $k = 0, 1, \ldots, K-1$. In the synthesizer a similar set of spectral components $\hat{X}_k(m)$ can be recombined to form a single output signal $\hat{x}(n)$. In practice the signals $X_k(m)$ and $\hat{X}_k(m)$ are often reduced in sampling rate for efficiency and therefore these systems are inherently multirate systems.

An important class of filter banks are those based on the framework of the discrete Fourier transform (DFT). Perhaps the simplest interpretation of the DFT filter bank, shown in Fig. 3.42(a), is that in which each channel, k, is separately bandpass modulated by a complex modulation signal $e^{-j\omega_k n}$, lowpass filtered by a filter $h(n)$, and then reduced in sampling rate by a factor M. Signal paths with double lines denote complex signals. Figure 3.42(b) shows the equivalent transpose operation for channel k in the synthesizer. In this framework the filter $h(n)$ is often referred to as the *analysis filter* and $f(n)$ is often referred to as the *synthesis filter*.

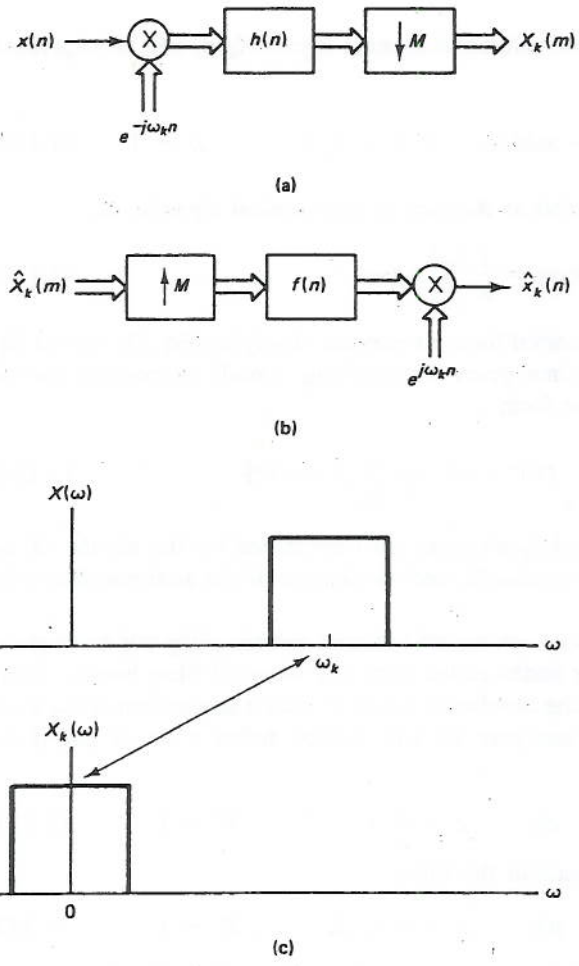


Figure 3.42 (a) Single channel of a DFT filter bank analyzer; (b) single channel of a DFT filter bank synthesizer; (c) a spectral interpretation.

In the DFT filter bank model the center frequencies of the channels are further defined to be uniformly spaced so that the modulation frequencies are

$$\omega_k = \frac{2\pi k}{K}, \qquad k = 0, 1, 2, \ldots, K-1 \tag{3.144}$$

It is convenient to define

$$W_K = e^{j(2\pi/K)} \tag{3.145}$$

and the complex modulation function as

$$e^{-j\omega_k n} = e^{-j(2\pi kn/K)} = W_K^{-kn} \tag{3.146}$$

The channel signals can then be expressed as

$$X_k(m) = \sum_{n=-\infty}^{\infty} h(mM - n)x(n)W_K^{-kn}, \qquad k = 0, 1, 2, \ldots, K-1 \tag{3.147}$$

Similarly, in the synthesizer the reconstructed channel signals $\hat{x}_k(n)$ can be expressed as

$$\hat{x}_k(n) = W_K^{kn} \sum_{m=-\infty}^{\infty} \hat{X}_k(m)f(n - mM), \qquad k = 0, 1, 2, \ldots, K-1 \tag{3.148}$$

and the final output, $\hat{x}(n)$, is defined as the sum of the channel signals, i.e.,

$$\hat{x}(n) = \frac{1}{K}\sum_{k=0}^{K-1} \hat{x}_k(n) \tag{3.149}$$

where the scale factor of $1/K$ is inserted for convenience. Applying Eq. (3.148) to Eq. (3.149) and interchanging sums then gives the resulting overall expression for the DFT filter bank synthesizer in the form

$$\hat{x}(n) = \sum_{m=-\infty}^{\infty} f(n - mM)\frac{1}{K}\sum_{k=0}^{K-1} \hat{X}_k(m)W_K^{kn} \tag{3.150}$$

The properties of the analyzer and synthesizer are determined by the choice of the number of bands K, the decimation ratio M, and the designs of the analysis filter $h(n)$ and the synthesis filter $f(n)$.

For the purposes of this discussion we will restrict and simplify our example to an important subset of DFT filter banks called critically sampled filter banks, where the decimation ratio M is equal to the number of bands K. (For a discussion of the more general case, see [10].) For the analyzer we will further define a set of polyphase analysis filters of the form

$$\overline{p}_\rho(m) = h(mM - \rho), \qquad \rho = 0, 1, 2, \ldots, M-1 \tag{3.151}$$

and a set of decimated input signals of the form

$$x_\rho(m) = x(mM + \rho), \qquad \rho = 0, 1, 2, \ldots, M-1 \tag{3.152}$$

By carefully applying these definitions and the condition $M = K$ to Eq. (3.147), we can derive an expression for the filter bank analyzer in the form

$$X_k(m) = \sum_{\rho=0}^{M-1} \sum_{r=-\infty}^{\infty} \bar{p}_\rho(r) W_M^{-k\rho} x_\rho(m-r)$$
$$= \sum_{\rho=0}^{M-1} W_M^{-k\rho} [\bar{p}_\rho(m) * x_\rho(m)] \tag{3.153}$$

where $*$ denotes discrete convolution [10]. This form suggests the structure of Fig. 3.43(a) for a single filter bank channel k.

Equation (3.153) further suggests that $X_k(m)$ is of the form of a discrete Fourier transform of the convolved outputs $\bar{p}_\rho(m) * x_\rho(m)$ of the polyphase branches. In fact, these output signals are *independent* of the filter channel number k. Therefore, it is clear from Eq. (3.153) or Fig. 3.43(a) that computations involving the polyphase

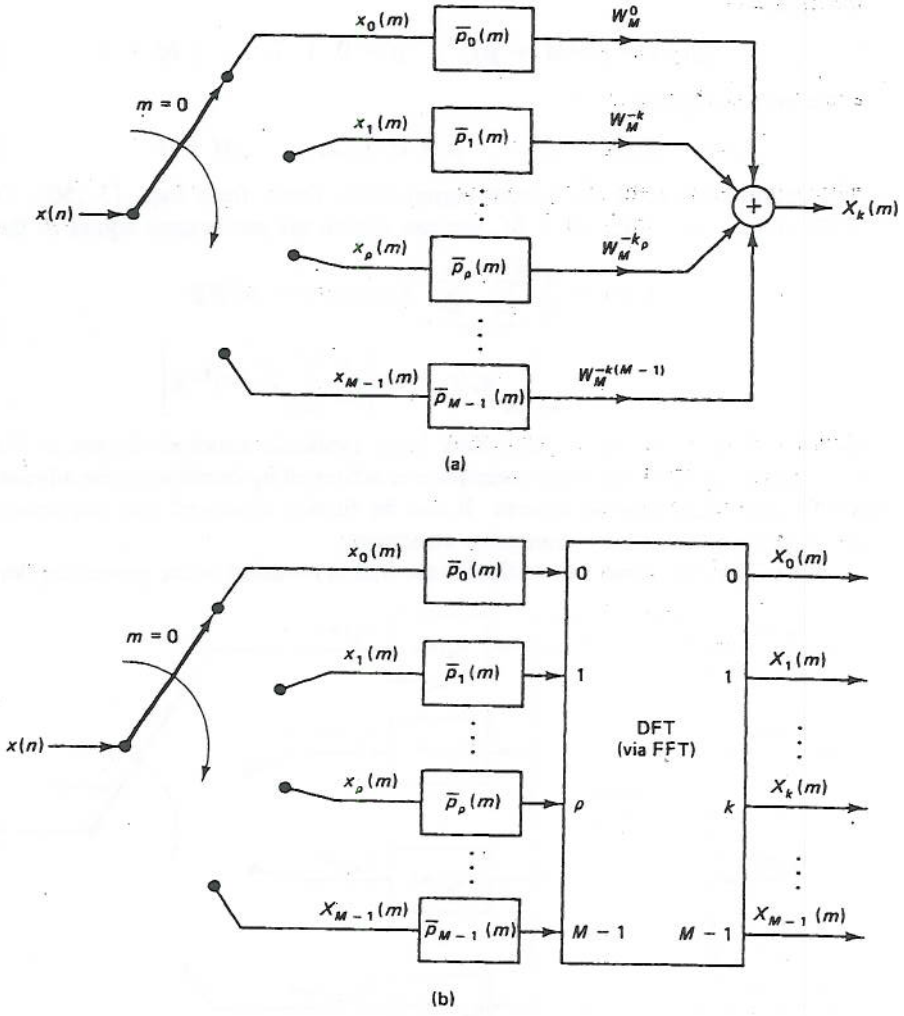


Figure 3.43 (a) Polyphase structure for the kth channel of a DFT filter bank analyzer; (b) the total polyphase DFT filter bank structure with an FFT modulator.

filters $\bar{p}_\rho(m)$ can be shared among all the filter bank channels, saving a factor of M in the total computation. An additional factor of M is gained from the polyphase form of the structure, as discussed in Section 3.3.3. Finally, we can recognize that the DFT in Eq. (3.153) can be performed with $M \log_2 M$ efficiency (as opposed to M^2 efficiency) by using the fast Fourier transform (FFT) algorithm. This then leads to the highly efficient filter bank structure shown in Fig. 3.43(b). In this implementation the input signal $x(n)$ is divided into a decimated set of branch signals, each of which is filtered by a separate polyphase filter (a decimated set of the coefficients $h(n)$). After one sweep of the commutator, the filtered outputs are transformed by the FFT to produce the desired filter bank signals. The overall structure is therefore very simple and elegant to implement.

A similar form for the synthesizer can be derived by defining the polyphase synthesis filters

$$q_\rho(m) = f(mM + \rho), \qquad \rho = 0, 1, 2, \ldots, M-1 \tag{3.154}$$

and the branch signals

$$\hat{x}_\rho(r) = \hat{x}(rM + \rho), \qquad \rho = 0, 1, 2, \ldots, M-1 \tag{3.155}$$

which define subsets of the output signal $\hat{x}(n)$. Then, from Eqs. (3.150), (3.154), (3.155) and the condition $K = M$, we can derive the ρth branch signal in the form

$$\begin{aligned} \hat{x}_\rho(r) &= \frac{1}{M} \sum_{k=0}^{M-1} \sum_{m=-\infty}^{\infty} \hat{X}_k(m) q_\rho(r - m) W_M^{k\rho} \\ &= \sum_{m=-\infty}^{\infty} q_\rho(r-m) \left[\frac{1}{M} \sum_{k=0}^{M-1} \hat{X}_k(m) W_M^{k\rho} \right] \end{aligned} \tag{3.156}$$

This form suggests the polyphase filter bank synthesis structure shown in Fig. 3.44 where, again, an efficient implementation is achieved by combining the advantages of the FFT and polyphase structures. It can be further observed that the structures of Figs. 3.43(b) and 3.44 are transpose structures.

An additional subtle but critical issue that is revealed in the preceding derivation

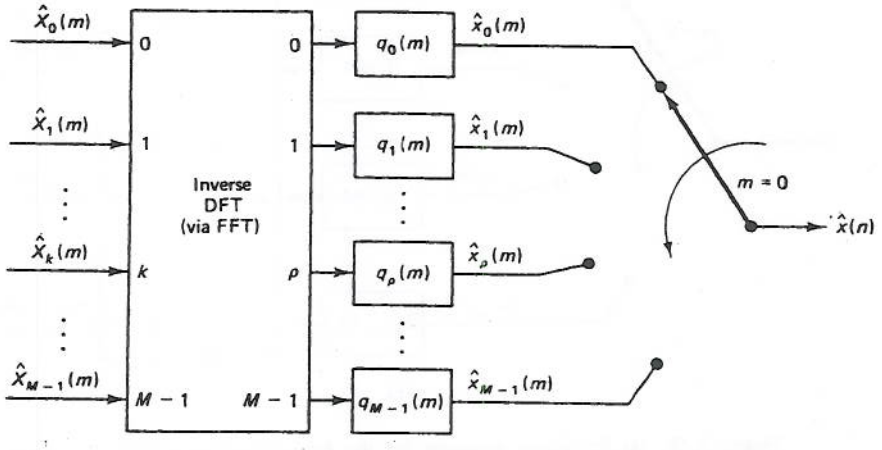


Figure 3.44 Synthesis structure for the polyphase DFT filter bank.

is that the transpose of a clockwise commutator is a counterclockwise commutator. This accounts for the fact that forms of the polyphase filters in Eqs. (3.151) and (3.154) must be defined with different signs to reflect the direction of commutation implied by the polyphase structure [10].

This concludes our discussion of filter bank implementations. The example derived above represents one of a class of filter bank structures that can be derived to take full advantage of efficient decimation and interpolation structures combined with the efficiency of fast transform algorithms. The issues are further complicated when we consider back-to-back arrangements of filter banks in which the processes of both analysis and synthesis must be performed. The interested reader is referred to Chapter 7 in [10]. For a discussion of the related topic of short-time Fourier transform, see Chapter 6 of this book.

3.7 CONCLUSIONS AND PRACTICAL EXAMPLES OF MULTIRATE DIGITAL SYSTEMS

In this chapter we have described the basic concepts of sampling rate conversion and have shown how these concepts can be efficiently utilized and extended to a broad range of signal processing operations. We conclude this chapter by showing specific examples of how these concepts have been utilized in practical applications and systems.

3.7.1 Sampling Rate Conversion in Digital Audio Systems

An important practical application of multirate digital signal processing and sampling rate conversion is in the field of professional digital audio. A variety of different types of digital processing systems have emerged for storage, transmission, and processing of audio program material. For a number of reasons such systems may have different sampling rates depending on whether they are used for broadcasting, digital storage, consumer products, or other professional applications. Also in digital processing of audio material, signals may be submitted to different types of digital rate control for varying the speed of the program material. This process can inherently vary the sampling frequency of the digital signal.

In practice it is often desired to convert audio program material from one digital format to another. One way to achieve this format conversion is to convert the audio signal back to analog form and digitize it in the new format. This process inherently introduces noise at each stage of conversion because of the limited dynamic range of the analog circuitry associated with the D/A and A/D conversion processes. Furthermore, this noise accumulates at each new interface.

An alternative and more attractive approach is to convert directly between the two digital formats by a process of waveform interpolation. This process is depicted in Fig. 3.45 and it is seen to be basically a sampling rate conversion problem. Since the accuracy of this sampling rate conversion can be maintained with any desired degree of precision (by controlling the wordlengths and the interpolator designs), essentially a noise free interconnection between the two systems can be achieved.

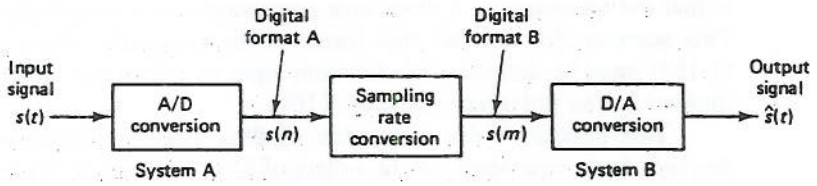


Figure 3.45 Example of a digital-to-digital translation between two audio signal formats.

3.7.2 Conversion Between Delta Modulation and PCM Signal Coding Formats

A second application of sampling rate conversion is in the area of digital communications. In communication networks a variety of different coding formats may be used in different parts of the network to achieve flexibility and efficiency. Conversion between these coding formats often involves a conversion of the basic sampling rate.

By way of example, delta modulation (DM) is sometimes used in A/D conversion or in voice terminals because of its simplicity and low cost. DM is basically a 1-bit/sample coding technique in which only the sign of the sample-to-sample difference of a highly oversampled signal is encoded. This approach eliminates the need for expensive antialiasing filters and allows the signal to be manipulated in a simple unframed serial bit stream format.

Alternatively, in long-distance transmission or in signal processing operations, such as digital filtering, it is generally desired to have the signal in a pulse code modulation (PCM) format. Thus it is necessary to convert between the high-sampling-rate, single-bit format of DM and the low-sampling-rate, multiple-bit format of PCM. Figure 3.46(a) shows an example of this process for the DM-to-PCM conversion, and Fig. 3.46(b) shows the reverse process of converting from a PCM to a DM format. When used as a technique for A/D conversion, this approach can combine the advantages of both the DM and the PCM signal formats.

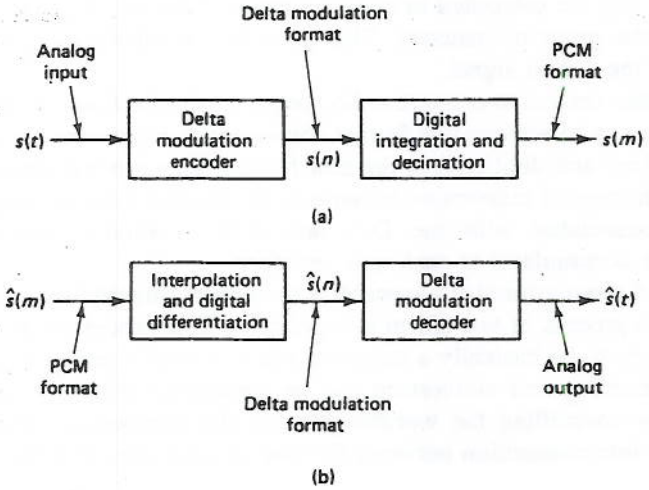


Figure 3.46 Illustration of (a) a DM-to-PCM conversion and (b) a PCM-to-DM conversion.

3.7.3 Digital Time-Division Multiplexing (TDM) to Frequency-Division Multiplexing (FDM) Translation

A third example of multirate digital systems is the translation of signals in a telephone system between time-division-multiplexed (TDM) and frequency-division-multiplexed (FDM) formats. The FDM format is often used for long-distance transmission, whereas the TDM format is more convenient for digital switching.

Figure 3.47 illustrates the basic process of translating a series of 12 TDM digital speech signals, $s_1(n)$, $s_2(n)$, . . . , $s_{12}(n)$, to a single FDM signal $r(m)$, and Fig. 3.47(b) illustrates the reverse (FDM-to-TDM) translation process. The sampling rate of the TDM speech signals is 8 kHz, whereas the sampling rate of the FDM signal is much higher to accommodate the increased total bandwidth. In each channel of the TDM-to-FDM translator the sampling rate is effectively increased (by interpolation) to the higher FDM sampling rate. The signal is then modulated by single-sideband modulation techniques to its appropriate frequency-band location in the range 56 to

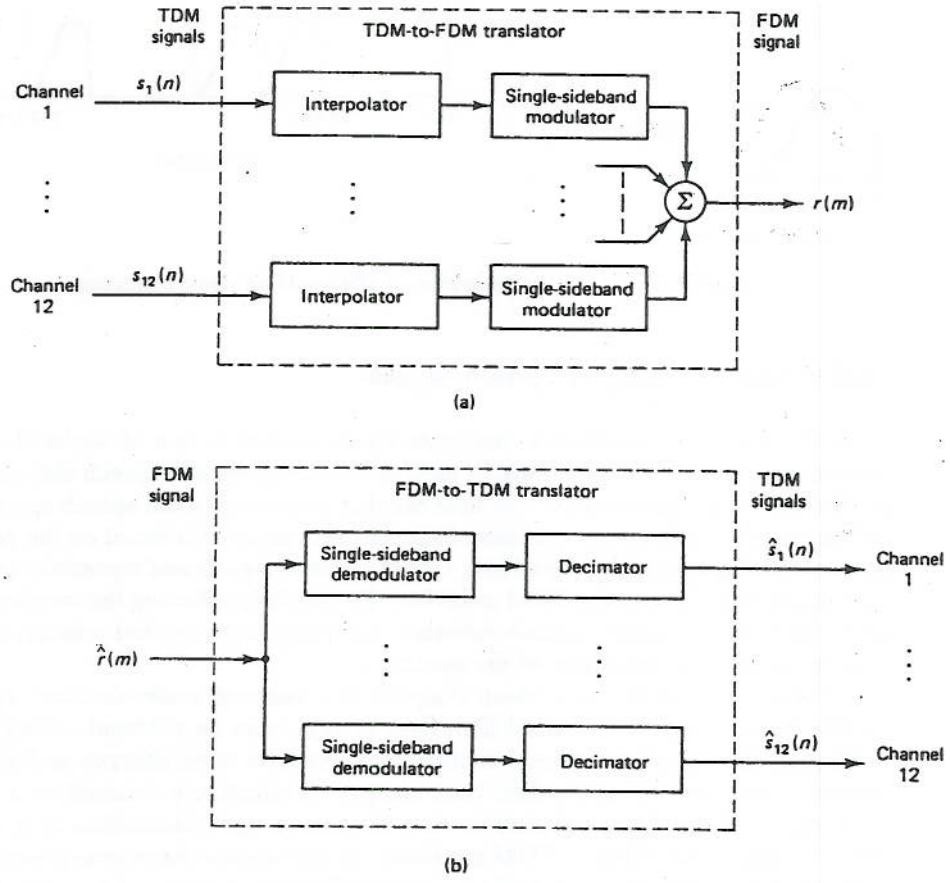


Figure 3.47 Illustration of (a) a TDM-to-FDM translator and (b) an FDM-to-TDM translator.

112 kHz, as illustrated in Fig. 3.48. The interpolated and modulated channel signals are then digitally summed to give the desired FDM signal. In the FDM-to-TDM translator, the reverse process takes place. As seen in Fig. 3.47 the process of translation between TDM and FDM formats involves sampling rate conversion, and therefore these systems are inherently multirate systems.

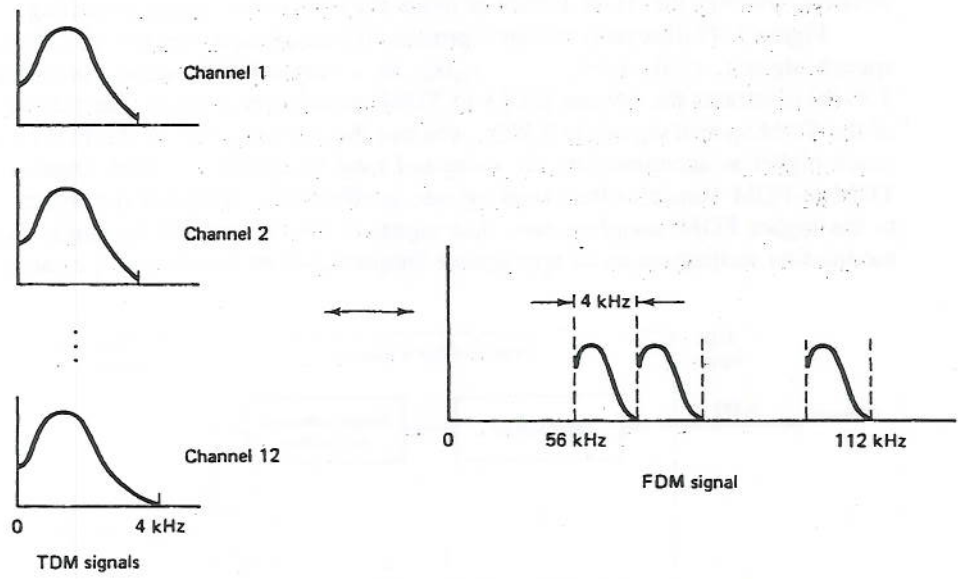


Figure 3.48 Spectral interpretation of TDM-to-FDM signal translation.

3.7.4 Subband Coding of Speech Signals

A fourth example of a practical multirate digital system is that of subband coding. Subband coding is a technique that is used to efficiently encode speech signals at low bit rates by taking advantage of the time-varying properties of the speech spectrum as well as some well-known properties of speech perception. It is based on the principle of decomposing the speech signal into a set of subband signals and separately encoding each signal with an adaptive PCM quantizer. By carefully selecting the number of bits per sample used to quantize each subband, according to perceptual criteria, one can achieve an efficient encoding of the speech.

Figure 3.49(a) shows a block diagram of a subband coder-decoder, and Fig. 3.49(b) shows a typical five-band filter bank arrangement for subband coding. A key element in this design is the implementation of the filter bank analysis and synthesis systems. Each subband in the filter bank analyzer is effectively obtained by a process of bandpass filtering, modulation to zero frequency, and decimation in a manner similar to that of the FDM-to-TDM translator. In the receiver the reverse process takes place, with the filter bank synthesizer reconstructing an output signal from the decoded subband signals. The process is similar to that of the TDM-to-FDM translator.

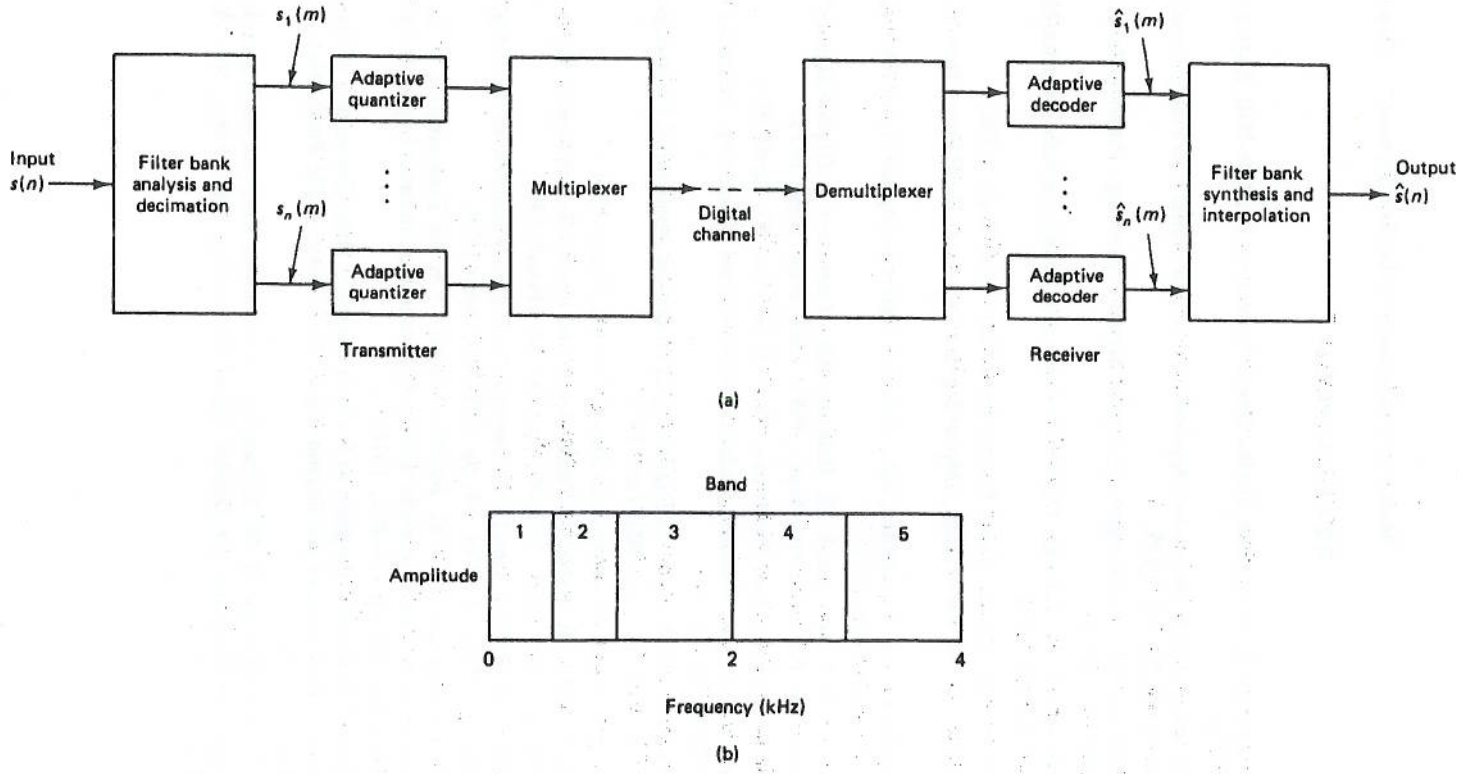


Figure 3.49 (a) Illustration of an N-band subband coder; (b) an example of the frequency bands for a 5-band subband coder design.

REFERENCES

1. A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
2. L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
3. A. V. Oppenheim, Ed., *Applications of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1978.
4. L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, NJ, 1978.
5. A. Peled and B. Liu, *Digital Signal Processing*, Wiley, New York, 1976.
6. L. R. Rabiner and C. M. Rader, Eds., *Digital Signal Processing*, IEEE Press, New York, 1972.
7. Digital Signal Processing Committee, Eds., *Selected Papers in Digital Signal Processing II*, IEEE Press, New York, 1976.
8. H. D. Helms, J. F. Kaiser, and L. R. Rabiner, Eds., *Literature in Digital Signal Processing: Author and Permuted Title Index*, IEEE Press, New York, 1975.
9. Special Issue on Digital Signal Processing, *Proc. IEEE*, Vol. 63, April 1975.
10. R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
11. R. W. Schafer and L. R. Rabiner, "A Digital Signal Processing Approach to Interpolation," *Proc. IEEE*, Vol. 61, pp. 692–702, June 1973.
12. M. Bellanger, *Traitment Numérique du Signal*, Masson, Paris, 1981.
13. R. E. Crochiere and L. R. Rabiner, "Interpolation and Decimation of Digital Signals: A Tutorial Review," *Proc. IEEE*, Vol. 69, pp. 300–331, March 1981.
14. C. E. Schmidt, L. R. Rabiner, and D. A. Berkley, "A Digital Simulation of the Telephone System," *Bell Syst. Tech. J.*, Vol. 58, pp. 839–855, April 1979.
15. T. A. C. M. Claasen and W. F. G. Mecklenbrauker, "On the Transposition of Linear Time-Varying Discrete-Time Networks and Its Application to Multirate Digital Systems," *Philips J. Res.*, Vol. 23, pp. 78–102, 1978.
16. R. E. Crochiere, "A General Program to Perform Sampling Rate Conversion of Data by Rational Ratios," in *Programs for Digital Signal Processing*, IEEE Press, New York, 1979, pp. 8.2-1–8.2-7.
17. G. Oetken, T. W. Parks, and H. W. Schuessler, "A Computer Program for Digital Interpolator Design," in *Programs for Digital Signal Processing*, IEEE Press, New York, 1979, pp. 8.1-1–8.1-6.