Interpolation and Decimation of Digital Signals—A Tutorial Review

RONALD E. CROCHIERE, SENIOR MEMBER, IEEE, AND LAWRENCE R. RABINER, FELLOW, IEEE

Invited Paper

Abstract—The concepts of digital signal processing are playing an increasingly important role in the area of multirate signal processing, i.e. signal processing algorithms that involve more than one sampling rate. In this paper we present a tutorial overview of multirate digital signal processing as applied to systems for decimation and interpolation. We first discuss a theoretical model for such systems (based on the sampling theorem) and then show how various structures can be derived to provide efficient implementations of these systems. Design techniques for the linear-time-invariant components of these systems (the digital filter) are discussed, and finally the ideas behind multistage implementations for increased efficiency are presented.

I. Introduction

One of the most fundamental concepts of digital signal processing is the idea of sampling a continuous process to provide a set of numbers which, in some sense, is representative of the characteristics of the process being sampled. If we denote a continuous function from the process being sampled as $x_C(t)$, $-\infty \leqslant t \leqslant \infty$ where x_C is a continuous function of the continuous variable t (t may represent time, space, or any other continuous physical variable), then we can define the set of samples as $x_D(n)$, $-\infty \leqslant n \leqslant \infty$ where the correspondence between t and n is essentially specified by the sampling process, i.e.,

$$n = q(t) \tag{1a}$$

Many types of sampling have been discussed in the literature [1]-[3] including nonuniform sampling, uniform sampling, and multiple function uniform sampling. The most common form of sampling, and the one which we will refer to throughout this paper is uniform (periodic) sampling in which

$$q(t) = t/T = n \tag{1b}$$

i.e., the samples $x_D(n)$ are uniformly spaced in the dimension t, occurring nT apart. For uniform sampling we define the sampling period as T and the sampling rate as

$$F = 1/T \tag{2}$$

It should be clear from the above discussion that $x_C(t)$ can be sampled with any sampling period T. However, for a unique correspondence between the continuous function $x_C(t)$ and the discrete sequence $x_D(n)$, it is necessary that the sampling period T be chosen to satisfy the requirements of the Nyquist sampling theorem. This concept of a unique analog waveform corresponding to a digital sequence will often be used in the course of our discussion to provide greater intuitive insights into the nature of the processing algorithms that we will be considering.

The sampling period T is a fundamental consideration in many signal processing techniques and applications. It often determines the convenience, efficiency, and/or accuracy in which the signal processing can be performed. In some cases an input signal may already be sampled at some predetermined sampling period T and the goal is to convert this sampled signal to a new sampled signal at a different sampling period T' such that the resulting signal corresponds to the same analog function. In other cases it may be more efficient or convenient to perform different parts of a processing algorithm at different sampling rates in which case it may be necessary to convert the sampling rates of the signals in the system from one rate to another.

The process of digitally converting the sampling rate of a signal from a given rate $F = 1/T$ to a different rate $F' = 1/T'$ is called *sampling rate conversion.* When the new sampling rate is higher than the original sampling rate, i.e.,

$$F' > F \tag{3a}$$

or

$$T' < T \tag{3b}$$

the process is generally called *interpolation* since we are creating samples of the original physical process from a reduced set of samples. Historically the mathematical process of interpolation, or "reading between the lines," has received widespread attention from mathematicians who were interested in the problem of tabulating useful mathematical functions. The question was how often a given function had to be tabulated (sampled) so that someone could use a simple interpolation rule to obtain accurate values of the function at any higher sampling rate [4]. Not only did this early work lead to an appreciation of the sampling process, but it also led to several interesting classes of "interpolation functions" which could provide almost arbitrarily high accuracy in the interpolated values, provided that sufficient tabulated values of the function were available.

The process of digitally converting the sampling rate of a signal from a given rate F to a lower rate F', i.e.,

$$F' < F \tag{4a}$$

Manuscript received June 25, 1980.

The authors are with Bell Laboratories, Murray Hill, NJ 07974.

0018-9219/81/0300-0300$00.75 © 1981 IEEE

or

$$T' > T \tag{4b}$$

is called *decimation*.[1] It will be shown in Section III that decimation and interpolation of signals are dual processes—i.e., a digital system which implements a decimator can be transformed into a dual digital system which implements an interpolator using straightforward transposition techniques.

The techniques to be described in this paper have been applied in a wide variety of areas including:

1) communications systems [5], [6];
2) speech processing systems [7]–[9];
3) antenna systems [10];
4) radar systems [11], [12].

The above list contains only a few representative examples of multirate digital systems.

From a digital signal processing point of view, both the processes of interpolation and decimation can be well formulated in terms of linear filtering operations. This is the basic point of view we have taken in this paper. We begin in Section II with the mathematical (and signal processing) framework of sampling, interpolation, and decimation. In Section III we discuss digital networks (structures) which can be used to implement the conversion from one sampling rate to another. Included in this section is a brief review of signal-flowgraph representations of digital systems, and of structures for implementing the digital filters required for all sampling rate conversion systems. It is then shown how efficient implementations of sampling rate conversion systems can be obtained by simple manipulations of the proposed canonic structures.

In Section IV, we discuss the question of how to design the digital filter used in the systems presented in Sections II and III. It is shown that two general structures can be used to aid in the design of the special filters required in sampling rate conversion systems. Based on these structures, a number of special purpose design algorithms are described.

Finally Section V addresses the question of special structures for handling two special cases of sampling rate conversion, namely: 1) large changes in sampling rates within the system and 2) changes in sampling rate requiring large sampling rate changes internally in the structure—e.g., sampling rate conversion by a factor of 97/151. Each of these cases can be handled most efficiently in a multistage structure in which the sampling rate conversion occurs in a series of 2 or more distinct stages. Questions of computational, storage, and control efficiency are of paramount concern in the discussions in this section.

In this paper, we only consider decimation and interpolation systems based on finite impulse response (FIR) realizations. Another broad class of sampling rate conversion systems that can also be defined is based on infinite impulse response (IIR) realizations. However, they do not conveniently permit linear phase designs and a discussion of these issues was considered to be beyond the scope of this paper.

II. Basic Concepts of Sampling Rate Conversion

Fig. 1 provides a general description of a sampling rate conversion system. We are given the signal $x(n)$, sampled at the rate $F = 1/T$, and wish to compute the signal $y(m)$ with a new sampling rate $F' = 1/T'$. We will assume throughout this paper that the ratio of sampling periods of $y(m)$ and $x(n)$ can be expressed as a rational fraction, i.e.,

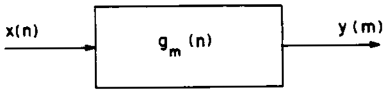


Fig. 1. Basic process of digital sampling rate conversion.

$$T'/T = F/F' = M/L \tag{5}$$

where M and L are integers.

A close examination of the structure of Fig. 1 shows that the systems we are dealing with for digital-to-digital sampling rate conversion are inherently linear time-varying systems, i.e., $g_m(n)$ is the response of the system at the output sample time m to an input at the input sample time $\lfloor mM/L \rfloor - n$ where $\lfloor u \rfloor$ denotes the integer less than or equal to u (this will become clearer in later discussion).

Since the system is linear, each output sample $y(m)$ can be expressed as a linear combination of input samples. A general form for this expression, which is used extensively in this paper, is [13]

$$y(m) = \sum_{n=-\infty}^{\infty} g_m(n)\, x\left(\left\lfloor \frac{mM}{L} \right\rfloor - n\right). \tag{6}$$

A derivation of (6) is given shortly where it is also seen that the system response $g_m(n)$ is periodic in m with period L, i.e.,

$$g_m(n) = g_{m+rL}(n), \qquad r = 0, \pm 1, \pm 2, \cdots. \tag{7}$$

Thus the system $g_m(n)$ belongs to the class of linear, *periodically* time-varying systems. Such systems have been extensively studied for a wide range of applications [14], [15].

In the trivial case when $T' = T$, or $L = M = 1$, equation (6) reduces to the simple time-invariant digital convolution equation

$$y(m) = \sum_{n=-\infty}^{\infty} g(n)\, x(m-n) \tag{8}$$

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

In the next few sections we study in some detail the structure and properties of systems that perform two special cases of sampling rate conversion, namely decimation by integer factors, and interpolation by integer factors [16]. We then consider the general case of a sampling rate change by a factor of L/M.

A. Sampling Rate Reduction—Decimation by an Integer Factor M

Consider the process of reducing the sampling rate of $x(n)$ by an integer factor M, i.e.,

$$T'/T = M/1. \tag{9}$$

Then the new sampling rate is $F' = F/M$. Assume that $x(n)$ represents a full band signal, i.e., its spectrum is nonzero for all frequencies f in the range $-F/2 \leqslant f \leqslant F/2$, with $\omega = 2\pi fT$, i.e.,

$$|X(e^{j\omega})| \neq 0, \qquad |\omega| = |2\pi fT| \leqslant \frac{2\pi FT}{2} = \pi \tag{10}$$

[1] Strictly speaking decimation means a reduction by 10 percent. In signal processing decimation has come to mean a reduction in sampling rate by any factor.

except possibly at an isolated set of points. Based on well known sampling theory, in order 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 low-pass filter which approximates the ideal characteristic

$$\tilde{H}(e^{j\omega}) = \begin{cases} 1, & |\omega| \leqslant 2\pi F'T/2 = \pi/M \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The sampling rate reduction is then achieved by forming the sequence $y(m)$ by extracting every Mth sample of the filtered output. This process is illustrated in Fig. 2(a). If we denote the actual low-pass filter unit sample response as $h(n)$, then we have

$$w(n) = \sum_{n=-\infty}^{\infty} h(k)\, x(n-k) \tag{12}$$

where $w(n)$ is the filtered output as seen in Fig. 2(a). The final output $y(m)$ is then obtained as

$$y(m) = w(Mm) \tag{13}$$

as denoted by the operation of the second box in Fig. 2(a). This block diagram symbol, which will be referred to as a *sampling rate compressor*, will be used consistently throughout this paper, and it corresponds to the resampling operation given by (13).

Fig. 2(b) shows typical spectra (magnitude of the Fourier transforms) of the signals $x(n)$, $h(n)$, $w(n)$, and $y(m)$ for an M to 1 reduction in sampling rate.

By combining (12) and (13) the relation between $y(m)$ and $x(n)$ is of the form

$$y(m) = \sum_{k=-\infty}^{\infty} h(k)\, x(Mm-k) \tag{14}$$

which is seen to be a special case of (6). Thus for decimation by integer factors of M we have

$$g_m(n) = g(n) = h(n), \qquad \text{for all } m \text{ and all } n. \tag{15}$$

Although $g_m(n)$ is *not* a function of m for this case, it can readily be seen that the overall system of (14) and Fig. 2(a) is not time-invariant by considering the output signal obtained when $x(n)$ is shifted by an integer number of samples. For this case, unless the shift is a multiple of M, the output is *not* a shifted version of the output for 0 shift, i.e.,

$$x(n) \rightarrow y(m) \tag{16a}$$

but

$$x(n-\delta) \not\rightarrow y(m-\delta/M) \text{ unless } \delta = rM. \tag{16b}$$

It is of value 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 the imperfect low-pass filter. To obtain this relationship we define the signal

$$w'(n) = \begin{cases} w(n), & n = 0, \pm M, \pm 2M, \cdots \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

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

$$w'(n) = w(n) \left\{ \frac{1}{M} \sum_{l=0}^{M-1} e^{j2\pi ln/M} \right\}, \quad -\infty < n < \infty \tag{18}$$

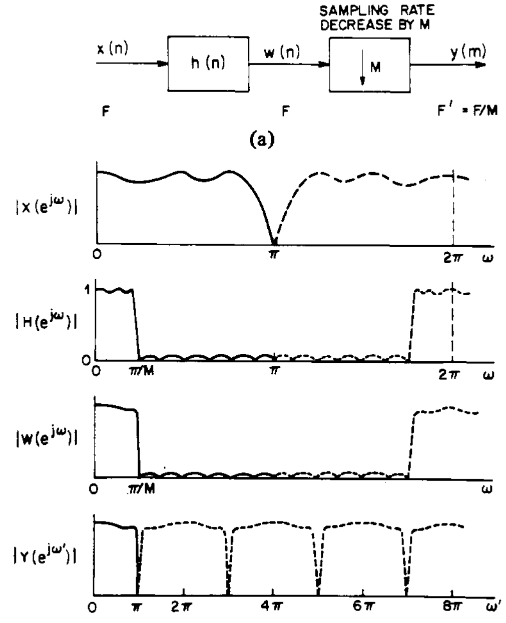


Fig. 2. Block diagram and typical spectra for sampling rate reduction by a factor of M

where the term in brackets corresponds to a discrete Fourier series representation of a periodic impulse train with a period of M samples. Thus we have

$$y(m) = w'(Mm) = w(Mm) \tag{19}$$

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

$$\begin{aligned} Y(z) &= \sum_{m=-\infty}^{\infty} y(m)\, z^{-m} \\ &= \sum_{m=-\infty}^{\infty} w'(Mm)\, z^{-m} \end{aligned} \tag{20}$$

and since $w'(m)$ is zero except at integer multiples of M, equation (20) becomes

$$\begin{aligned} Y(z) &= \sum_{m=-\infty}^{\infty} w'(m)\, z^{-m/M} \\ &= \sum_{m=-\infty}^{\infty} w(m) \left[\frac{1}{M} \sum_{l=0}^{M-1} e^{j2\pi lm/M} \right] z^{-m/M} \\ &= \frac{1}{M} \sum_{l=0}^{M-1} \left[\sum_{m=-\infty}^{\infty} w(m)\, e^{j2\pi lm/M} z^{-m/M} \right] \end{aligned}$$

$$Y(z) = \frac{1}{M} \sum_{l=0}^{M-1} W(e^{-j2\pi l/M} z^{1/M}). \tag{21}$$

Since

$$W(z) = H(z)\, X(z) \tag{22}$$

we can express $Y(z)$ as

$$Y(z) = \frac{1}{M} \sum_{l=0}^{M-1} H(e^{-j2\pi l/M} z^{1/M})\, X(e^{-j2\pi l/M} z^{1/M}). \tag{23}$$

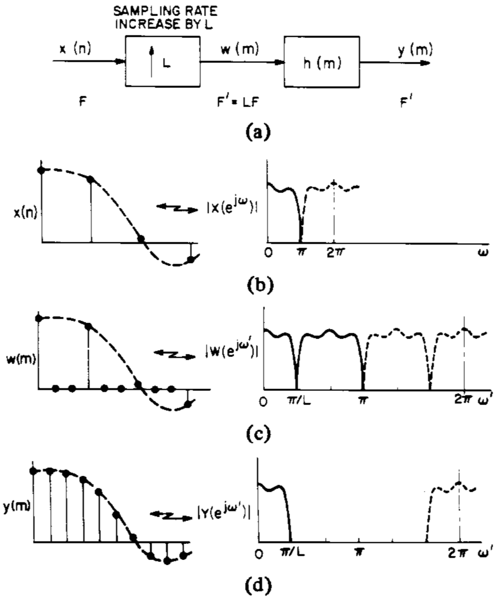


Fig. 3. Block diagram and typical waveforms and spectra for sampling rate increase by a factor of L.

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

$$Y(e^{j\omega'}) = \frac{1}{M} \sum_{l=0}^{M-1} H(e^{j(\omega' - 2\pi l)/M}) X(e^{j(\omega' - 2\pi l)/M}) \quad (24a)$$

where

$$\omega' = 2\pi f T' \quad \text{(in radians relative to sampling period } T'\text{)}. \quad (24b)$$

Equation (24) expresses the Fourier transform of the output signal $y(m)$ in terms of the transforms of the aliased components of the filtered input signal $x(n)$. By writing the individual components of (24) directly we see that

$$Y(e^{j\omega'}) = \frac{1}{M} \Big[H(e^{j\omega'/M}) X(e^{j\omega'/M}) + H(e^{j(\omega' - 2\pi)/M}) X(e^{j(\omega' - 2\pi)/M}) + \cdots \Big]. \quad (25)$$

The purpose of the low-pass filter $H(e^{j\omega})$ is to sufficiently filter $x(n)$ so that its components above the frequency $\omega = \pi/M$ are negligible. In terms of (24) this implies that all terms for $l \neq 0$ are removed and if the filter $H(e^{j\omega})$ closely approximates the ideal response of (11) then (24) becomes

$$Y(e^{j\omega'}) \cong \frac{1}{M} X(e^{j\omega'/M}), \qquad \text{for } |\omega'| \leqslant \pi. \quad (26)$$

B. Sampling Rate Increase–Interpolation by an Integer Factor L

If the sampling rate is increased by an integer factor L, then the new sampling period T' is

$$\frac{T'}{T} = \frac{1}{L} \quad (27)$$

and the new sampling rate F' is $F' = LF$. This process of increasing the sampling rate of a signal $x(n)$ by L implies that we must interpolate $L - 1$ new sample values between each pair of sample values of $x(n)$.

Fig. 3 illustrates this process of increasing the sampling rate 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

$$w(m) = \begin{cases} x(m/L), & m = 0, \pm L, \pm 2L, \cdots \\ 0, & \text{otherwise.} \end{cases} \quad (28)$$

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 (28) and it will be referred to as a *sampling rate expander.* The resulting signal $w(n)$ has the z-transform

$$W(z) = \sum_{m=-\infty}^{\infty} w(m) z^{-m} \quad (29a)$$

$$= \sum_{m=-\infty}^{\infty} x(m) z^{-mL} \quad (29b)$$

$$= X(z^L). \quad (29c)$$

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

$$W(e^{j\omega'}) = X(e^{j\omega' L}) \quad (30)$$

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(c) the spectrum of $w(n)$ 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 $\pm 2\pi/L, \pm 4\pi/L, \cdots$. To recover the baseband signal of interest and eliminate the unwanted higher frequency components it is necessary to filter the signal $w(m)$ with a digital low-pass filter which approximates the ideal characteristic

$$\tilde{H}(e^{j\omega'}) = \begin{cases} G, & |\omega'| \leqslant \dfrac{2\pi F T'}{2} = \pi/L \\ 0, & \text{otherwise.} \end{cases} \quad (31)$$

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

Letting $H(e^{j\omega'})$ denote the frequency response of an actual filter that approximates the characteristic in (31) it is seen that

$$Y(e^{j\omega'}) = H(e^{j\omega'}) X(e^{j\omega' L}) \quad (32)$$

and within the approximation of (31)

$$Y(e^{j\omega'}) \cong \begin{cases} G X(e^{j\omega' L}), & |\omega'| \leqslant \pi/L \\ 0, & \text{otherwise.} \end{cases} \quad (33)$$

It is easy to see why we need a gain of G in $\tilde{H}(e^{j\omega})$, whereas for the decimation filter a gain of 1 is adequate, by examining the zeroth sample of the sequences. From Fig. 2 it is clear that

$$y(0) = w(0)$$

$$= x(0)$$

if we assume that $|H(e^{j\omega})| = 1$ for $|\omega| < \pi/M$ and $|X(e^{j\omega})| = 0$ for $|\omega| > \pi/M$. Alternatively for the interpolator it is seen with the aid of Fig. 3 and (33) that

$$\begin{aligned}
y(0) &= \int_{-\pi}^{\pi} Y(e^{j\omega'})\, d\omega' \\
&= \int_{-\pi}^{\pi} H(e^{j\omega'}) X(e^{j\omega' L})\, d\omega' \\
&= G \int_{-\pi/L}^{\pi/L} X(e^{j\omega' L})\, d\omega' \\
&= G \int_{-\pi}^{\pi} X(e^{j\omega})\, d\omega/L \\
&= \frac{G}{L}\, x(0).
\end{aligned}$$

Therefore, a gain $G = L$ is required to match the amplitudes of the envelopes of the signals $y(m)$ and $x(n)$.

If $h(m)$ denotes the unit sample response of $H(e^{j\omega'})$, then $y(m)$ can be expressed as

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-k)\, w(k). \tag{34}$$

Combining (28) and (34) leads to

$$\begin{aligned}
y(m) &= \sum_{k=-\infty}^{\infty} h(m-k)\, x(k/L) \\
&= \sum_{r=-\infty}^{\infty} h(m-rL)\, x(r).
\end{aligned} \tag{35}$$

Next we introduce the change of variables

$$r = \left\lfloor \frac{m}{L} \right\rfloor - n \tag{36}$$

and the identity

$$mM - \left\lfloor \frac{mM}{L} \right\rfloor L = (mM) \oplus L \tag{37}$$

where $\lfloor u \rfloor$ denotes the integer less than or equal to u and $(i) \oplus L$ denotes the value of i modulo L. Applying (36) and (37) (with $M = 1$) to (35) then gives

$$\begin{aligned}
y(m) &= \sum_{n=-\infty}^{\infty} h\left(m - \left\lfloor \frac{m}{L} \right\rfloor L + nL\right) x\left(\left\lfloor \frac{m}{L} \right\rfloor - n\right) \\
&= \sum_{n=-\infty}^{\infty} h(nL + m \oplus L)\, x\left(\left\lfloor \frac{m}{L} \right\rfloor - n\right).
\end{aligned} \tag{38}$$

Equation (38) expresses the output $y(m)$ in terms of the input $x(n)$ and the filter coefficients $h(m)$ and it is again seen to be a special case of (6). Thus for interpolation by integer factors of L we have

$$g_m(n) = h(nL + m \oplus L), \qquad \text{for all } m \text{ and all } n \tag{39}$$

and it is seen that $g_m(n)$ is periodic in m with period L as indicated by (7).

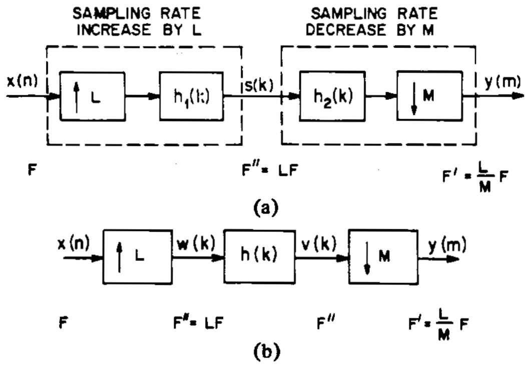


Fig. 4. (a) Cascade of an integer interpolator and an integer decimator for achieving sampling rate conversion by rational fractions. (b) A more efficient implementation of this process.

C. Sampling Rate Conversion by a Rational Factor M/L

In the previous two sections we have considered the cases of sampling rate reduction by an integer factor M and sampling rate increase by an integer factor L. In this section, we consider the general case of conversion by the ratio

$$\frac{T'}{T} = \frac{M}{L} \tag{40}$$

or

$$F' = \frac{L}{M} F. \tag{41}$$

This conversion can be achieved by a cascade of the two above processes of integer conversions by first increasing the sampling rate by L and then decreasing it by M. Fig. 4(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. 4(a) that the two filters $h_1(k)$ and $h_2(k)$ are operating in cascade at the same sampling rate LF. 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. 4(b). Since this digital filter $h(k)$ must serve the purposes of both the decimation and interpolation operations described in the previous two sections it is clear from (11) and (31) that it must approximate the ideal digital low-pass characteristic

$$\tilde{H}(e^{j\omega''}) = \begin{cases} L, & |\omega''| = |2\pi f T''| \leq \min \left| \dfrac{\pi}{L}, \dfrac{\pi}{M} \right| \\ 0, & \text{otherwise} \end{cases} \tag{42}$$

where

$$\omega'' = 2\pi f T'' = 2\pi f T/L \tag{43}$$

i.e., 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 filter is $F'' = LF$.

The time domain input-to-output relation for the general conversion circuit of Fig. 4(b) can be derived by considering the integer interpolation and decimation relations derived in Sections II-A and II-B, i.e., from (35) it can be seen that $v(k)$ can be expressed as

$$v(k) = \sum_{r=-\infty}^{\infty} h(k - rL)\, x(r) \tag{44}$$

and from (13) $y(m)$ can be expressed in terms of $v(k)$ as

$$y(m) = v(Mm). \tag{45}$$

Combining (44) and (45) gives

$$y(m) = \sum_{r=-\infty}^{\infty} h(Mm - rL)\,x(r) \tag{46}$$

and making the change of variables

$$r = \left\lfloor \frac{mM}{L} \right\rfloor - n \tag{47}$$

and applying (37) gives

$$\begin{aligned} y(m) &= \sum_{n=-\infty}^{\infty} h\left(Mm - \left\lfloor \frac{mM}{L} \right\rfloor L + nL\right) x\left(\left\lfloor \frac{mM}{L} \right\rfloor - n\right) \\ &= \sum_{n=-\infty}^{\infty} h(nL + mM \oplus L)\, x\left(\left\lfloor \frac{mM}{L} \right\rfloor - n\right). \end{aligned} \tag{48}$$

It is seen that (48) corresponds to the general form of the time-varying digital-to-digital conversion system described by (6) and that the time-varying unit sample response $g_m(n)$ can be expressed as

$$g_m(n) = h(nL + mM \oplus L), \qquad \text{for all } m \text{ and all } n \tag{49}$$

where $h(k)$ is the time-invariant unit sample response of the low-pass digital filter at the sampling rate LF [13].

Similarly, by considering the transform relationships of the individual integer decimation and interpolation systems, the output spectrum $Y(e^{j\omega'})$ can be determined in terms of the input spectrum $X(e^{j\omega})$ and the frequency response of the filter $H(e^{j\omega''})$. From (32) it is seen that $V(e^{j\omega''})$ can be expressed in terms of $X(e^{j\omega})$ and $H(e^{j\omega''})$ as

$$V(e^{j\omega''}) = H(e^{j\omega''})\, X(e^{j\omega'' L}) \tag{50}$$

and from (21) $Y(e^{j\omega'})$ can be expressed in terms of $V(e^{j\omega''})$as

$$\begin{aligned} Y(e^{j\omega'}) &= \frac{1}{M} \sum_{l=0}^{M-1} V(e^{j(\omega' - 2\pi l)/M}) \\ &= \frac{1}{M} \sum_{l=0}^{M-1} H(e^{j(\omega' - 2\pi l)/M})\, X(e^{j(\omega' L - 2\pi l)/M}). \end{aligned} \tag{51}$$

When $H(e^{j\omega'})$ closely approximates the ideal characteristic of (42) it is seen that this expression reduces to

$$Y(e^{j\omega'}) \cong \begin{cases} \dfrac{L}{M} X(e^{j\omega' L/M}), & \text{for } |\omega'| \leqslant \min(\pi, \pi M/L) \\ 0, & \text{otherwise} \end{cases} \tag{52}$$

Thus far, we have developed the general system for sampling rate conversion of low-pass signals by arbitrary rational factors L/M. It was shown that the process of sampling rate conversion could be modeled as a linear, periodically time-varying system, and that the unit sample response of this system, $g_m(n)$ could be expressed in terms of the unit sample response $h(k)$ of a time-invariant digital filter designed for the highest system sampling rate LF.

D. Sampling Rate Conversion of Bandpass Signals

In the preceding sections it was assumed that the signals that we are dealing with are low-pass signals and therefore the filters required for decimation and interpolation are low-pass filters which preserve the baseband signals of interest. In many practical systems it is also necessary to deal with bandpass signals and many of the results discussed in this paper can be logically extended to the bandpass case as well. While it is not our intention to go into detail on these issues in this paper we will briefly give an example in this section of one way in which this can be done. A more involved discussion of these issues can be found in [10].

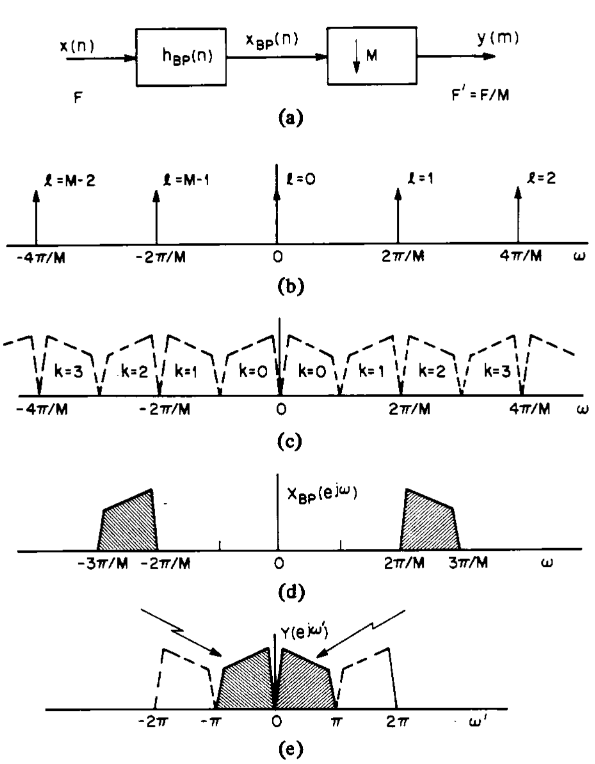


Fig. 5. Decimation of a bandpass signal and a spectral interpretation for the $k = 2$ band.

Perhaps the simplest and most direct approach to decimating or interpolating digital bandpass signals is to take advantage of the inherent frequency translating (i.e., aliasing) properties of decimation and interpolation. This property can be used to advantage when dealing with bandpass signals by associating the bandpass signal with one of these modulated "harmonics" instead of with the baseband. Fig. 5(a) illustrates an example of this process for the case of decimation 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 reduced in sampling rate by selecting one out of every M samples giving the final output $y(m)$. It is seen that this system is identical to that of the integer low-pass decimator with the exception that the filter is a bandpass filter instead of a low-pass filter. Thus the output signal $Y(e^{j\omega'})$ can be expressed as

$$Y(e^{j\omega'}) = \frac{1}{M} \sum_{l=0}^{M-1} H_{BP}(e^{j(\omega' - 2\pi l)/M})\, X(e^{j(\omega' - 2\pi l)/M}). \tag{53}$$

From (53) it is seen that $Y(e^{j\omega'})$ is composed of M aliased components of $X(e^{j\omega'}) H_{BP}(e^{j\omega'})$ modulated by factors of $2\pi l/M$. The function of the filter $H_{BP}(e^{j\omega})$ is to remove (attenuate) all aliasing components except those associated with the desired band of interest. Since the modulation is restricted to values of $2\pi l/M$ it can be seen that only specific frequency bands are allowed by this method. As a consequence the choice of the filter $H_{BP}(e^{j\omega})$ is restricted to approximate one of the M ideal characteristics

$$\tilde{H}_{BP}(e^{j\omega}) = \begin{cases} 1, & k\pi/M < |\omega| < (k+1)\pi/M \\ 0, & \text{otherwise} \end{cases} \tag{54}$$

where $k = 0, 1, 2, \cdots, M-1$, i.e., $H_{BP}(e^{j\omega})$ is restricted to bands $\omega = k\pi/M$ to $\omega = (k+1)\pi/M$ where π/M is the bandwidth.

Figs. 5(b)–(e) illustrate this approach. Fig. 5(b) shows the M possible modulating frequencies which are a consequence of the M to 1 sampling rate reduction, i.e., the digital sampling function (a periodic train of unit samples spaced M samples apart) has spectral components spaced $2\pi l/M$ apart. Fig. 5(c) shows the "sidebands" that are associated with these spectral components which correspond to the M choices of bands as defined by (54). They correspond to the bands that are aliased into the baseband of the output signal $Y(e^{j\omega'})$ according to (54). (As seen by (53) and (54) and Figs. 5(b) and (c), the relationship between k and l is nontrivial).

Fig. 5(d) illustrates an example in which the $k = 2$ band is used, such that $X_{BP}(e^{j\omega})$ is bandlimited to the range $2\pi/M < |\omega| < 3\pi/M$. Since the process of sampling rate reduction by M to 1 corresponds to a convolution of the spectra of $X_{BP}(e^{j\omega})$ (Fig. 5(d)) and the sampling function (Fig. 5(b)) this band is lowpass translated to the baseband of $Y(e^{j\omega'})$ as seen in Fig. 5(e). Thus, the processes of modulation and sampling rate reduction are achieved simultaneously by the M to 1 sampling rate reduction.

The process of bandpass interpolation is the inverse to that of bandpass decimation and it can be accomplished in a similar manner. Referring to Fig. 3(c) it is seen that we can use a bandpass filter with a characteristic similar to that described by (54) (with M replaced by L) to remove one of the harmonic images of the baseband signal rather than the baseband signal itself. The net result is that we achieve both an interpolation and a modulation of the input signal to one of its harmonic locations in the spectrum.

III. Signal Processing 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. 4(b) which can be used to convert the sampling rate of a signal by a factor of L/M. As discussed in Section II the theoretical model for this system is increasing the signal sampling rate by a factor of L (by filling in $L-1$ zero-valued samples between each sample of $x(n)$ to give the signal $w(k)$), filtering $w(k)$ to eliminate the images of $X(e^{j\omega})$ by a standard linear time-invariant low-pass filter, $h(k)$, to give $v(k)$, and sampling rate compressing $v(k)$ by a factor M (by retaining 1 of each M samples of $v(k)$). A direct implementation of the system of Fig. 4(b) is grossly inefficient since the low-pass 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, one can directly apply this knowledge in implementing the system of Fig. 4(b) in a more efficient manner as will be discussed in this section. Later in Section V we will extend these concepts to include multistage implementations which can achieve greater efficiencies than single stage designs when the conversion ratios are large.

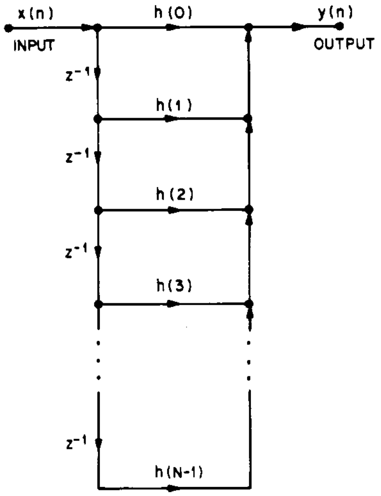


Fig. 6. Direct form structure for an FIR filter.

Before discussing specific classes of structures for sampling rate conversion we will first briefly review in Section III-A a number of fundamental network and signal-flowgraph concepts which will be used in developing these structures. We will then discuss three principle classes of FIR structures for realizing single stage interpolators and decimators and compare their properties.

A. Signal-Flowgraphs

In order to precisely define the sets of operations necessary to implement these digital systems we will strongly rely on the concepts of signal-flowgraph representation in this section [17]-[19]. Signal-flowgraphs provide a graphical representation of the explicit set of equations that are used to implement such systems. Furthermore, manipulating the flowgraphs in a pictorial way is equivalent to manipulation of the mathematical equations.

Fig. 6 illustrates an example of a signal-flowgraph of a direct-form 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 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.

Therefore from the signal-flowgraph (Fig. 6) 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-flowgraphs 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

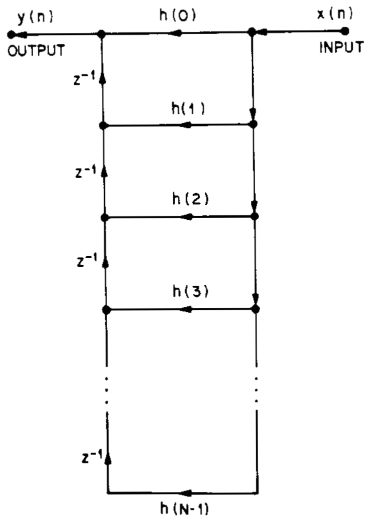


Fig. 7. Transposed direct form structure for an FIR filter.

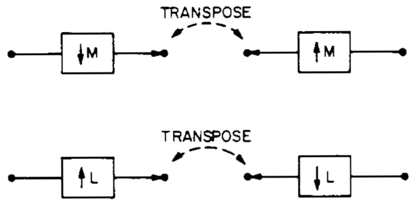


Fig. 8. Transpositions of the sampling rate compressor and expander.

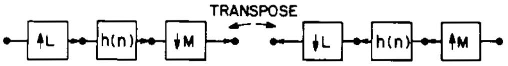


Fig. 9. Transpose of a generalized L/M sampling rate changer.

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 shall see shortly.

Another important network concept that we rely heavily on is that of transposition and duality [17]-[21]. Basically a dual system is one which 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. 2(a) and Fig. 3(a)) 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. 6 is the transposed direct form structure shown in Fig. 7. Also it can be shown [17]-[21] that for the case of linear time-invariant systems 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. 6 and Fig. 7 have identical system functions).

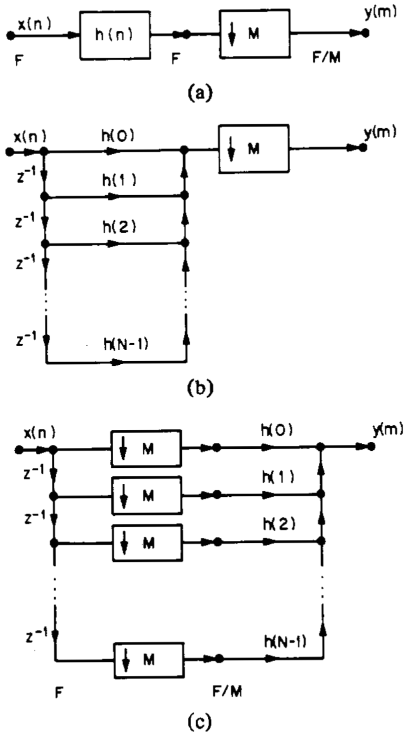


Fig. 10. Generation of an efficient direct form structure of an M to 1 decimator.

For 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. 8. Clearly these systems do not have the same system response.

By extending the concepts of transposition rigorously it can also be shown 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. 9.

B. Direct Form FIR Structures for Integer Changes in Sampling Rates

Consider the model of an M to 1 decimator as developed in Section II, and as shown in Fig. 10(a). According to this model the filter $h(n)$ operates at the high sampling rate F and $M - 1$ out of every M output samples of the filter 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. 10(b) results. The multiplications by $h(0)$, $h(1)$, $\cdots$, $h(N-1)$ and the associated summations in this network must be performed at the rate F.

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, the modified network of Fig. 10(c) results. The multiplications and additions associated with the coefficients $h(0)$ to $h(N-1)$ now occur at the low sampling rate F/M and therefore the total computation rate in the system has been reduced by a factor M. For every M samples of $x(n)$ which are shifted into the structure (the cascade of delays) one output sample $y(m)$ is

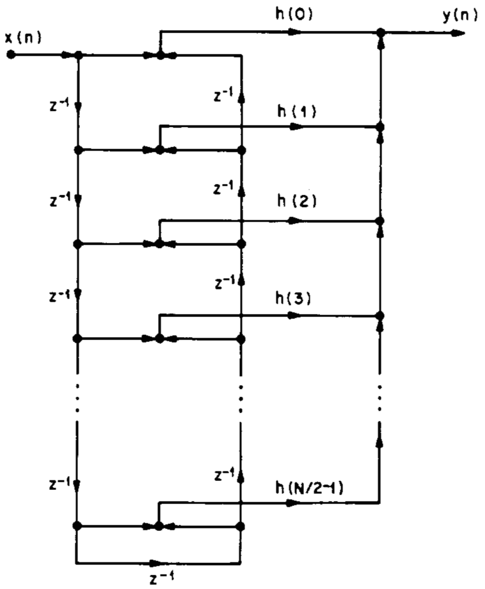


Fig. 11. Modified direct form FIR filter structure for exploiting impulse response symmetry.

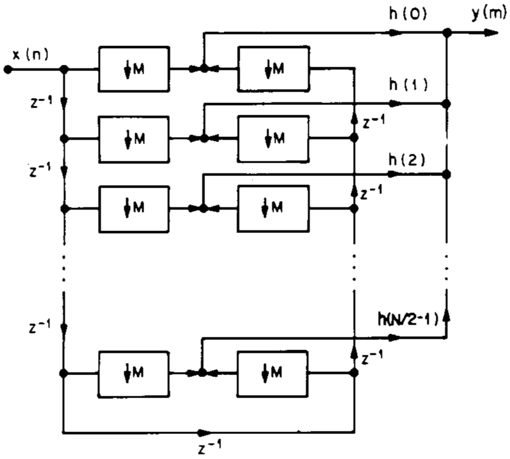


Fig. 12. Direct form realization of an M to 1 decimator that exploits symmetry in $h(n)$ for even values of N.

computed. Thus the structure of Fig. 10(c) is seen to be a direct realization of (14) in Section II-A.

An alternate form of this structure which can exploit symmetry in $h(n)$ (for linear phase designs) can be derived by using the modified direct form structure of Fig. 11 (for N even). This leads to the M to 1 decimator structure shown in Fig. 12 and it requires approximately a factor of 2 less multiplications than the structure of Fig. 10(c).

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. 13(a). In this case however, if $h(m)$ is realized with the direct form structure of Fig. 6 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. 7 [22]. The sampling rate expander can then be commuted into the network as shown by the series of operations in Fig. 13. Since the coefficients $h(0)$, $h(1)$, $\cdots$, $h(N-1)$ in Fig. 13(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. 13(b).

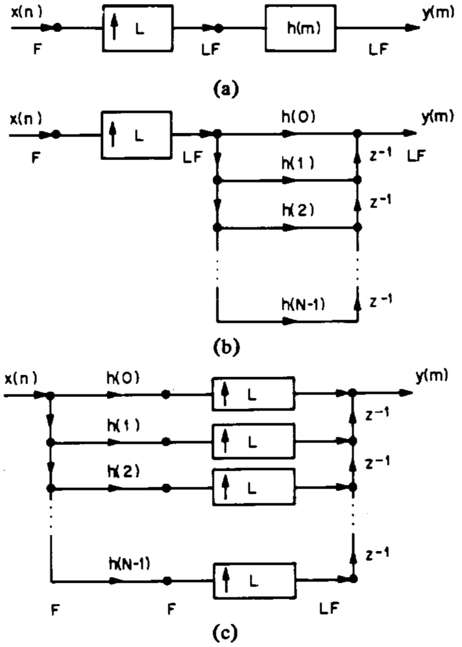


Fig. 13. Steps in the generation of an efficient structure of a 1 to L interpolator.

An alternative way of deriving the structure of Fig. 13(c) is by a direct transposition of the network of Fig. 10(c) (letting $L = M$). This is a direct consequence of the fact that decimators and interpolators are duals. Similarly an efficient direct form interpolator structure which can exploit symmetry in $h(n)$ can be obtained by transposing the structure of Fig. 12 and letting $L = M$. 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 [21]. 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.

C. Polyphase FIR Structures for Integer Decimators and Interpolators

A second general class of structures that are of interest in multirate digital systems are the polyphase networks (sometimes referred to as N path networks) [22], [23]. 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 II it was shown that a general form for the input-to-output time-domain relationship for the 1 to L interpolator is

$$y(m) = \sum_{n=-\infty}^{\infty} g_m(n)\, x\left(\left\lfloor \frac{m}{L} \right\rfloor - n\right) \qquad (55)$$

where

$$g_m(n) = h(nL + m \oplus L), \qquad \text{for all } m \text{ and } n \qquad (56)$$

is a periodically time-varying filter with period L. Thus to generate each output sample $y(m)$, $m = 0, 1, 2, \cdots, L - 1$,

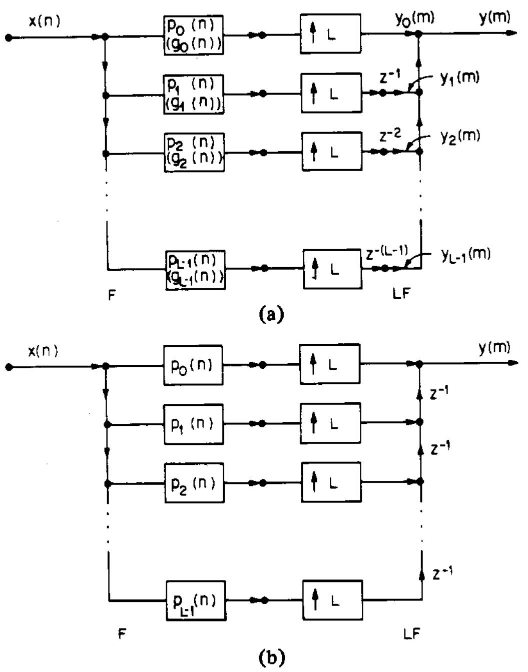


Fig. 14. Polyphase structures for a 1 to L interpolator.

a different set of coefficients $g_m(n)$ are 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 $y(1)$, etc.

Similarly the term $\lfloor m/L \rfloor$ in (55) increases by one for every L samples of $y(m)$. Thus for output samples $y(L), y(L+1), \cdots, y(2L-1)$ the coefficients $g_m(n)$ are multiplied by samples $x(1-n)$. In general, for output samples $y(rL)$, $y(rL+1), \cdots, y(rL+L-1)$ the coefficients $g_m(n)$ are multiplied by samples $x(r-n)$. Thus it is seen that $x(n)$ in (55) is updated at the low sampling rate F, whereas $y(m)$ is evaluated at the high sampling rate LF.

An implementation of the 1 to L interpolator based on the computation of (55) is shown in Fig. 14(a). The way in which this structure operates is as follows. The partitioned subsets, $g_0(n), g_1(n), \cdots, g_{L-1}(n)$, of $h(m)$ can be identified with L separate linear, time invariant filters which operate at the low sampling rate F. 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 $p_0(n), p_1(n), \cdots, p_{L-1}(n)$. Thus

$$p_\rho(n) = g_\rho(n), \quad \text{for } \rho = 0, 1, 2, \cdots, L-1 \text{ and all } n. \tag{57}$$

These filters $p_\rho(n)$ will be referred to as the polyphase filters. Furthermore by combining (56) and (57) it is apparent that

$$p_\rho(n) = h(nL+\rho), \quad \text{for } \rho = 0, 1, 2, \cdots, L-1 \text{ and all } n \tag{58}$$

For each new input sample $x(n)$ there are L output samples (see Fig. 14). The output from the upper path $y_0(m)$ has nonzero values for $m = nL$, $n = 0, \pm 1, \pm 2, \cdots$, which correspond to system outputs $y(nL)$, $n = 0, \pm 1, \cdots$. The output from the next path $y_1(m)$ is nonzero for $m = nL+1$, $n = 0, \pm 1, \pm 2, \cdots$ because of the delay of one sample at the high sampling rate. Thus $y_1(m)$ corresponds to the interpolation output samples $y(nL+1)$, $n = 0, \pm 1, \cdots$. In general the output of the ρth path, $y_\rho(m)$ corresponds to the interpolation output samples $y(nL+\rho)$, $n = 0, \pm 1, \cdots$. Thus for each input sample $x(n)$ each of the L branches of the polyphase network contributes one nonzero output which corresponds to one of the L outputs of the network. The polyphase interpolation network of Fig. 14(a) has the property that the filtering is performed at the low sampling rate and thus it is an efficient structure. A simple manipulation of the structure of Fig. 14(a) leads to the equivalent network of Fig. 14(b) in which all the delays are single sample delays.

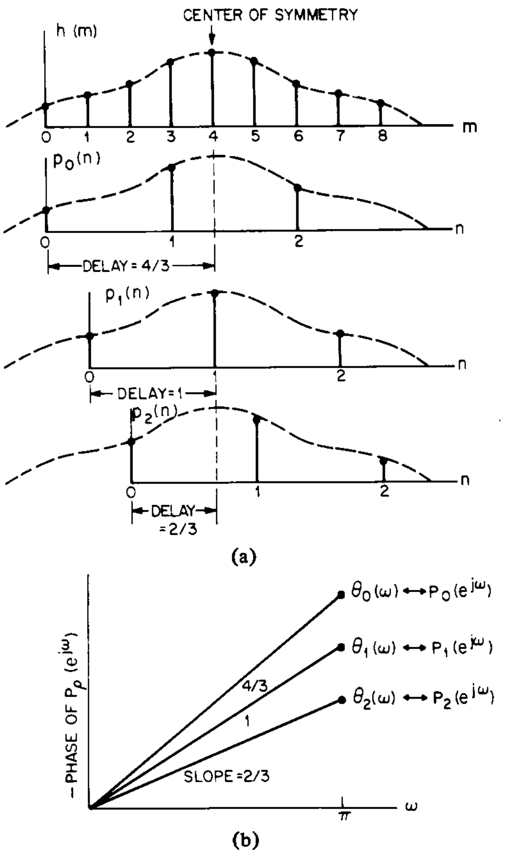


Fig. 15. Illustration of the properties of polyphase networks.

The individual polyphase filters $p_\rho(n)$, $\rho = 0, 1, 2, \cdots, L-1$ have a number of interesting properties. This is a consequence of the fact that the impulse responses $p_\rho(n)$, $\rho = 0, 1, 2, \cdots$, $L-1$, correspond to decimated versions of the impulse response of the prototype filter $h(m)$ (decimated by a factor of L according to (56) or (58)). Fig. 15 illustrates this for the case $L = 3$ and for an FIR filter $h(m)$ with $N = 9$ taps. The upper figure shows the samples of $h(m)$ where it is assumed that $h(m)$ is symmetric about $m = 4$. Thus $h(m)$ has a flat delay of 4 samples [17]. The filter $p_0(n)$ has three samples corresponding to $h(0)$, $h(3)$, $h(6) = h(2)$. Since the point of symmetry of the envelope of $p_0(n)$ is $n = \frac{4}{3}$ it has a flat delay of $\frac{4}{3}$ samples. Similarly $p_1(n)$ 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

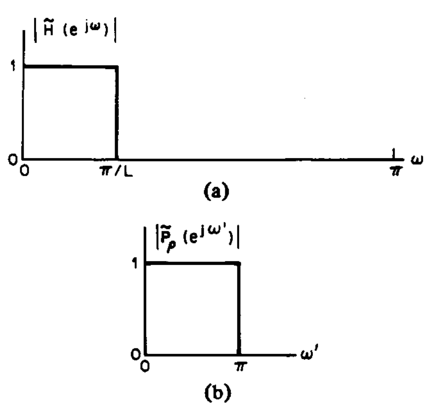


Fig. 16. Ideal frequency response of the polyphase networks.

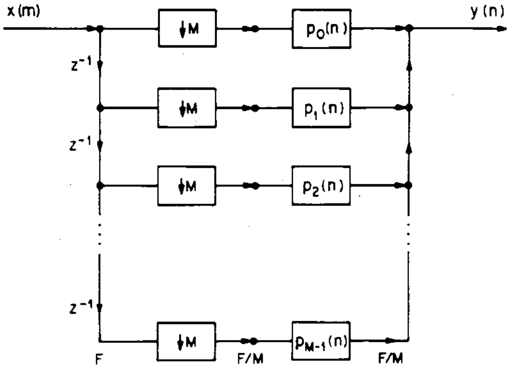


Fig. 17. Polyphase structure for an M to 1 decimator.

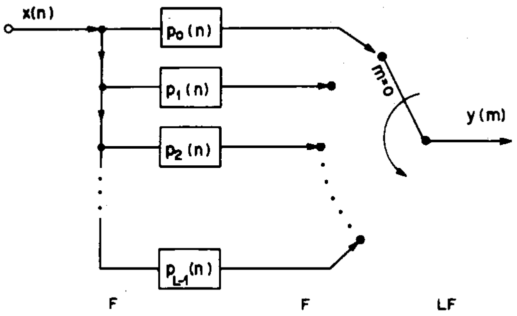


Fig. 18. Commutator model for the 1 to L polyphase interpolator.

$p_\rho(n)$ as seen in Fig. 15(b). These delays are compensated for by the delays which occur at the high sampling rate LF in the network (see Fig. 14). 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. 16. The frequency response of the prototype filter $h(m)$ approximates the ideal low-pass characteristic $\tilde{H}(e^{j\omega})$ shown in Fig. 16(a).[2] Since the polyphase filters $p_\rho(n)$ are decimated versions of $h(m)$ (decimated by L) the frequency response $0 \leqslant \omega \leqslant \pi/L$ of $\tilde{H}(e^{j\omega})$ scales to the range $0 \leqslant \omega' \leqslant \pi$ for $\tilde{P}_e(e^{j\omega'})$ as seen in Fig. 16 where $\tilde{P}_\rho(e^{j\omega'})$ is the ideal characteristic that the polyphase filter $p_\rho(n)$ approximates. Thus the polyphase filters approximate all-pass functions and each value of ρ, $\rho = 0, 1, 2, \cdots, L-1$, 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 [17], [24]. If a direct form FIR structure is used for the polyphase filters, the polyphase structure of Fig. 14 will require the same multiplication rate as the direct form interpolator structure of Fig. 13. Exploiting symmetry in $h(m)$ is more difficult in this class of structures since, at most, only one of the $p_\rho(n)$ subfilters are symmetric.

By transposing the structure of the polyphase 1 to L interpolator of Fig. 14(b), we get the polyphase M to 1 decimator structure of Fig. 17 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 above 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-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 II, the coefficients $g_m(n)$ for the M to 1 decimator are

$$g_m(n) = g(n) = h(n), \qquad \text{for all } n \text{ and } m. \tag{59}$$

Alternatively, according to the transpose network of Fig. 17, the coefficients of the M to 1 polyphase decimator are

$$p_\rho(n) = h(nM + \rho), \qquad \text{for } \rho = 0, 1, 2, \cdots, M-1, \text{and all } n \tag{60}$$

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. 14 it can be seen 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. 18. The commutator rotates in a counterclockwise direction starting with the zeroth-polyphase branch at time $m = 0$.

A similar commutator model can be developed for the M to 1 polyphase decimator by starting with the structure of Fig. 17 and replacing the delays and M to 1 sampling rate compressors with a commutator. This leads to the structure of Fig. 19. Again the commutator rotates in a counterclockwise direction starting with the zeroth-polyphase branch at time $m = 0$.

At this point the reader should be cautioned that an alternate formulation of these polyphase structures can be developed such that the commutators have clockwise rotations and a *different* but equivalent set of polyphase filters are defined.[3]

[2] Recall also that there is an additional gain of L required in the interpolator which we have ignored in this discussion (see Section II-B).

[3] This alternate formulation can be developed by defining a set of polyphase filters such that ρ is replaced by $-\rho$ on the right-hand side of (58).

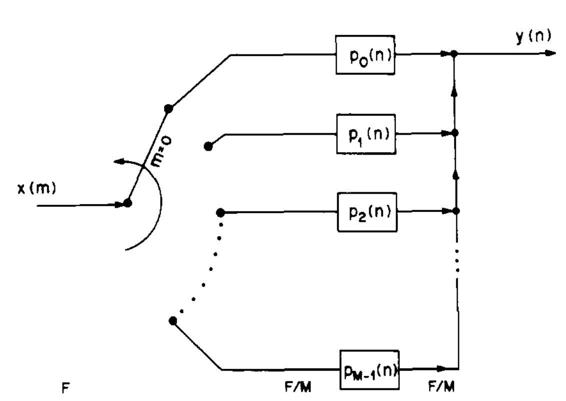
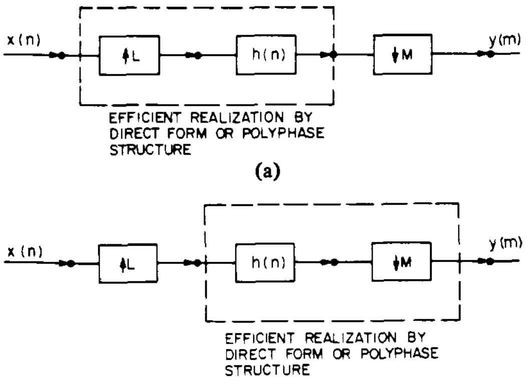


Fig. 19. Commutator model for the M to 1 polyphase decimator.


x(n)
↑L
h(n)
↓M
y(m)
EFFICIENT REALIZATION BY DIRECT FORM OR POLYPHASE STRUCTURE
x(n)
↑L
h(n)
↓M
y(m)
EFFICIENT REALIZATION BY DIRECT FORM OR POLYPHASE STRUCTURE

(a)

(b)

Fig. 20. Possible realization of an L/M sampling rate converter.

Both formulations have been used in the literature and they should not be confused.

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

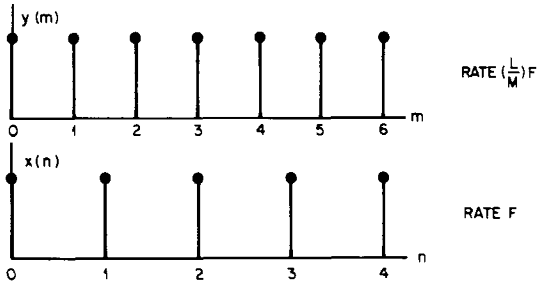
In the previous two sections we have considered implementations of decimators and interpolators using the direct form and polyphase structures for the case of integer changes in the sampling rate. Efficient realizations of these structures were obtained by commuting the filtering operations to occur at the low sampling rate. For the case of a network which realizes a change in sampling rate by a factor of L/M, it is difficult to achieve such efficiencies. The difficulty is illustrated in Fig. 20. 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. 20(a)). 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. 20(b)). In both cases difficulties arise and we are faced with a network which 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 [13]. This structure can be derived from the time domain input-to-output relation of the network, as derived in Section II, namely

$$y(m) = \sum_{n=-\infty}^{\infty} g_m(n)\, x\left(\left\lfloor \frac{mM}{L} \right\rfloor - n\right) \tag{61}$$

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

(a)



(b)

Fig. 21. Timing relationships between $y(m)$ and $x(n)$ for the case $M = 2, L = 3$.

where

$$g_m(n) = h(nL + mM \oplus L), \qquad \text{for all } m \text{ and all } n \tag{62}$$

and $h(k)$ corresponds to the low-pass (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.,

$$N = QL \tag{63}$$

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

$$g_m(n) = g_{m+rL}(n), \qquad r = 0, \pm 1, \pm 2, \cdots. \tag{64}$$

Therefore, equation (61) can be expressed as

$$y(m) = \sum_{n=0}^{Q-1} g_{m\oplus L}(n)\, x\left(\left\lfloor \frac{mM}{L} \right\rfloor - n\right). \tag{65}$$

Equation (65) 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 backwards in n sequentially. The weighting coefficients are periodically time varying so that the $m \oplus L$ coefficient set $g_{m\oplus L}(n)$, $n = 0, 1, 2, \cdots, Q - 1$, is used for the mth output sample. Fig. 21 illustrates this timing relationship for the $n = 0$ term in (65) and for the case $M = 2$ and $L = 3$. The table in Fig. 21(a) shows the index values of $y(m)$, $x(\lfloor mM/L \rfloor)$ and $g_{m\oplus L}(0)$ for $m = 0$, to $m = 6$. Fig. 21(b) illustrates the relative timing positions of the signals $y(m)$ and $x(n)$ drawn on an absolute time scale. By comparison of the table and the figure it can be seen that the value $x(\lfloor mM/L \rfloor)$ 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 $x(2)$, etc.

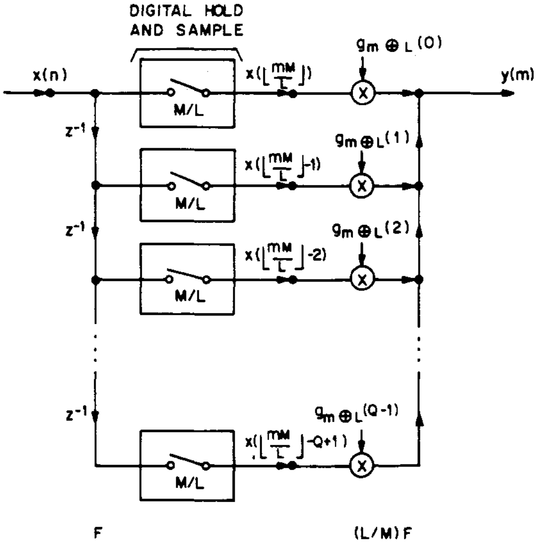

DIGITAL HOLD
AND SAMPLE
x(n)
y(m)
$g_{m \oplus L}(0)$
$g_{m \oplus L}(1)$
$g_{m \oplus L}(2)$
$g_{m \oplus L}(Q-1)$
$x(\lfloor \frac{mM}{L} \rfloor)$
$x(\lfloor \frac{mM}{L} \rfloor -1)$
$x(\lfloor \frac{mM}{L} \rfloor -2)$
$x(\lfloor \frac{mM}{L} \rfloor -Q+1)$
M/L
z^{-1}
F
(L/M)F

Fig. 22. Efficient structure for realizing an L/M sampling rate converter.

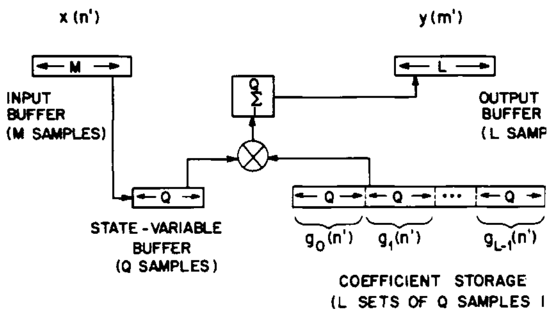


Fig. 23. Diagram of a program structure to implement the flowgraph of Fig. 22 in a block-by-block manner.

Based on (65) and the above description of how the input, output, and coefficients enter into the computation, the structure of Fig. 22 is suggested for realizing an L/M sampling rate converter. The structure consists of:

1) a Q sample "shift register" operating at the input sampling rate F which stores sequential samples of the input signal;
2) a direct form FIR structure with time-varying coefficients ($g_{m \oplus L}(n)$, $n = 0, 1, \cdots, Q - 1$) which operates at the output sampling rate $(L/M)F$;
3) a series of digital "hold-and-sample" boxes which 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.

It should be clear that the structure of Fig. 22 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.

Fig. 23 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 $x(n')$, $n' = 0, 1, 2, \cdots, M - 1$, and computes a block of L output samples $y(m')$ $m' = 0, 1, 2, \cdots, L - 1$. For each output sample time m', $m' = 0, 1, \cdots, L - 1$, 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 one, 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 which performs this computation can be found in [25].

E. Comparisons of Structures

In this section we have discussed three principle classes of FIR structures for decimators and interpolators. In addition, in Section V we discuss multistage cascades of these structures and show how this 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 some general properties of the above classes of structures.

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 [24]. As such this structure has been found useful for filter banks [23]. The structures with time-varying coefficients are particularly useful when considering conversions by factors of L/M.

There are many other considerations which determine overall efficiency of these structures. Most of these considerations, however, are filter design ones and hence we must defer further comparisons of single stage structures for decimators and interpolators until we have discussed the filter design issues in some detail.

IV. Design of FIR Filters for Decimation and Interpolation

In the discussion in the previous chapters we have assumed that the filter $h(k)$ approximates some ideal low-pass (or bandpass) characteristic (see Figs 2-4). As such 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, and discuss those methods 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 $H(e^{j\omega})$ defined as

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h(k)\, e^{-j\omega k} \tag{66}$$

$$= H(z)|_{z=e^{j\omega}}. \tag{67}$$

The frequency response is, in general, a complex function of ω. Thus it is convenient to represent it in terms of its magnitude $|H(e^{j\omega})|$ and phase $\theta(\omega)$ as

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{j\theta(\omega)} \tag{68}$$

where

$$|H(e^{j\omega})| = \sqrt{\mathrm{Re}^2[H(e^{j\omega})] + \mathrm{Im}^2[H(e^{j\omega})]} \tag{69a}$$

$$\theta(\omega) = \tan^{-1}\left[\frac{\mathrm{Im}[H(e^{j\omega})]}{\mathrm{Re}[H(e^{j\omega})]}\right]. \tag{69b}$$

An important filter parameter is the group delay $\tau(\omega)$ defined as

$$\tau(\omega) = \frac{-d\theta(\omega)}{d\omega}. \tag{70}$$

The group delay is a measure of time delay as a function of frequency of a signal as it passes through the filter. Nondispersive filters have the property that $\tau(\omega)$ is a constant (i.e., a fixed delay) over the frequency range of interest.

Before proceeding to a discussion of filter design techniques for decimators and interpolators, it is important to consider the ideal frequency domain and 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. Also, some filter design techniques are directed at the design of a single prototype filter such as in the classical filter design methods, whereas other filter design techniques are directed at the design of the polyphase filters. Thus we will consider both representations in this section.

A. Relationship Between the Prototype Filter and its Polyphase Representation

As discussed in Section III, 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. 15).

$$p_\rho(n) = h(\rho + nL), \qquad \rho = 0, 1, 2, \cdots, L-1, \text{ and all } n. \tag{71}$$

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

$$p_\rho(n) = h(\rho + nM), \qquad \rho = 0, 1, 2, \cdots, M-1, \text{ and all } n. \tag{72}$$

Taken as a set, the samples $p_\rho(n)$ ($\rho = 0, 1, \cdots, L-1$ for an interpolator, or $\rho = 0, 1, \cdots, M-1$ 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 only consider 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. 15). If we let $\hat{p}_\rho(k)$ represent the sampling rate expanded set

$$\hat{p}_\rho(k) = \begin{cases} p_\rho(k/L), & k = 0, \pm L, \pm 2L, \cdots \\ 0, & \text{otherwise} \end{cases} \tag{73}$$

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

$$h(k) = \sum_{\rho=0}^{L-1} \hat{p}_\rho(k-\rho). \tag{74}$$

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

$$H(z) = \sum_{\rho=0}^{L-1} z^{-\rho} P_\rho(z^L). \tag{75}$$

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

$$\delta_\rho(k) = \begin{cases} 1, & k = \rho, \rho \pm L, \rho \pm 2L, \cdots \\ 0, & \text{otherwise} \end{cases} \tag{76a}$$

$$= \frac{1}{L} \sum_{l=0}^{L-1} e^{j2\pi l(k-\rho)/L} \tag{76b}$$

then the sampling rate expanded sequences $\hat{p}_\rho(k)$ in (73) can be expressed as

$$\hat{p}_\rho(k) = \delta_\rho(k)\, h(k) = h(k) \frac{1}{L} \sum_{l=0}^{L-1} e^{j2\pi l(k-\rho)/L}. \tag{77}$$

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

$$P_\rho(z) = \sum_{n=-\infty}^{\infty} p_\rho(n)\, z^{-n} = \sum_{n=-\infty}^{\infty} \hat{p}_\rho(\rho + nL)\, z^{-n} \tag{78}$$

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

$$P_\rho(z) = \sum_{k=-\infty}^{\infty} \hat{p}_\rho(k)\, z^{-(k-\rho)/L}. \tag{79}$$

By substituting (77) into (79) we get

$$P_\rho(z) = \frac{1}{L} \sum_{k=-\infty}^{\infty} \sum_{l=0}^{L-1} h(k)\, e^{j2\pi l(k-\rho)/L}\, z^{-(k-\rho)/L}. \tag{80}$$

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

$$P_\rho(e^{j\omega}) = \frac{1}{L} \sum_{l=0}^{L-1} e^{j(\omega - 2\pi l)\rho/L} \sum_{k=-\infty}^{\infty} h(k)\, e^{-j(\omega-2\pi l)k/L}$$

$$= \frac{1}{L} \sum_{l=0}^{L-1} e^{j(\omega-2\pi l)\rho/L}\, H(e^{j(\omega-2\pi l)/L}),$$

$$\rho = 0, 1, 2, \cdots, L-1. \tag{81}$$

Equation (81) shows the relationships of the Fourier transforms of the polyphase filters to the Fourier transform of the prototype filter.

Equations (71)-(74), therefore, illustrate the time-domain relationships between $h(k)$ and $p_\rho(k)$ and (75) and (81) show their frequency-domain relationships.

B. Ideal Frequency Domain Characteristics for Interpolation and Decimation Filters

In the previous sections we have assumed that the filter $h(k)$ approximates some ideal low-pass (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 II-B that the interpolator filter $h(k)$ must approximate the ideal[4] low-pass characteristic

$$\tilde{H}(e^{j\omega'}) = \begin{cases} L, & |\omega'| < \pi/L \\ 0, & \text{otherwise} \end{cases} \tag{82}$$

as illustrated in Fig. 3.

By applying (82) to (81) it is possible to derive the equivalent ideal characteristics, $\tilde{P}_\rho(e^{j\omega})$, that are implied in the polyphase filters.[5] Because of the constraint imposed by (82), only the $l = 0$ term in (81) is nonzero and thus it simplifies to the form

$$\begin{aligned} \tilde{P}_\rho(e^{j\omega}) &= \frac{1}{L}\, e^{j\omega\rho/L}\, \tilde{H}(e^{j\omega/L}) \\ &= e^{j\omega\rho/L}, \quad \rho = 0, 1, 2, \cdots, L-1. \end{aligned} \tag{83}$$

Equation (83) shows that the "ideal" polyphase filters $\tilde{p}_\rho(n)$ should approximate all-pass filters with linear phase shifts corresponding to fractional advances of ρ/L samples ($\rho = 0, 1, 2, \cdots, L-1$) (ignoring any fixed delays that must be introduced in practical implementations of such filters). A further interpretation of the reason for this phase advance can be found in Section III-C on the discussion of polyphase structures.

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 $X(e^{j\omega})$, i.e.,

$$|X(e^{j\omega})| < \epsilon, \quad \text{for } \pi > |\omega| > \omega_c \tag{84}$$

where ϵ is a small quantity (relative to the peak of $|X(e^{j\omega})|$), then $W(e^{j\omega'})$ is an L-fold periodic repetition of $X(e^{j\omega})$ as shown in Fig. 24 (for $L = 5$). In this case, the ideal interpolator filter only has to remove the $(L-1)$ repetitions of the band of $X(e^{j\omega})$ where $|X(e^{j\omega})| > \epsilon$. Thus in the frequency domain, the ideal interpolator filter satisfies the constraints

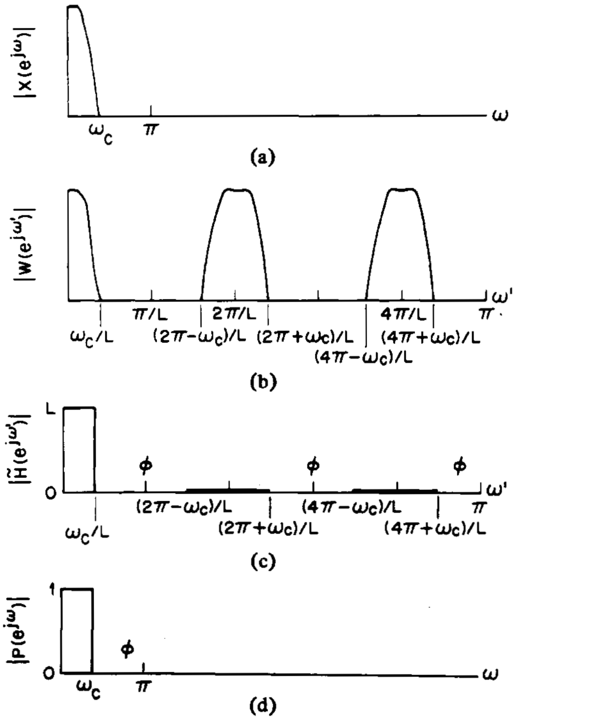


Fig. 24. Illustrations of ϕ bands in the specification of an interpolator filter ($L = 5$).

$$\tilde{H}(e^{j\omega'}) = \begin{cases} L, & 0 \leq |\omega'| \leq \omega_c/L \\ 0, & (2\pi r - \omega_c)/L \leq |\omega'| \leq (2\pi r + \omega_c)/L, \\ & \qquad r = 1, 2, \cdots, L-1 \end{cases} \tag{85}$$

as illustrated in Fig. 24(c). The bands from $(2\pi r + \omega_c)/L$ to $(2\pi(r+1) - \omega_c)/L$, ($r = 0, 1, \cdots$) are "don't care" (ϕ) bands in which the filter frequency response is essentially unconstrained. (In practice, however, $|H(e^{j\omega'})|$ should not be very large in these ϕ bands, e.g., not larger than L, to avoid amplification of any noise (or tails of $X(e^{j\omega})$) that may exist in these bands). We will see later how these ϕ bands can have a significant effect on the filter design problem. Fig. 24(d) shows the response of the ideal polyphase filter which is converted from an allpass to a low-pass 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 II-A for a decimator, the filter $H(e^{j\omega'})$ should approximate the ideal low-pass characteristic

$$\tilde{H}(e^{j\omega'}) = \begin{cases} 1, & 0 \leq \omega' \leq \pi/M \\ 0, & \text{otherwise.} \end{cases} \tag{86}$$

Alternatively, the polyphase filters should approximate the ideal allpass characteristics

$$\tilde{P}_\rho(e^{j\omega}) = \frac{1}{M}\, e^{j\omega\rho/M}, \quad \rho = 0, 1, 2, \cdots, M-1. \tag{87}$$

If we are only interested 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

[4] The "ideal" characteristic for a filter is denoted by a tilde over the variable throughout this section.

[5] Since the conventional filter $h(m)$ is implemented at the higher sampling rate, the frequency variable (in (82) for example) is ω', whereas since the polyphase filters are implemented at the lower sampling rate, the frequency variable is $\omega = \omega' L$.

have a situation where don't care bands are permitted in the filter design. The don't care regions are the same as those in (85) as illustrated in Fig. 24(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.

C. Time-Domain Properties of Ideal Interpolation and Decimation Filters

If we view the interpolation filter design problem in the time domain, an alternative picture of the "ideal" interpolation filter is obtained. By taking the inverse transform of the ideal filter characteristic defined by (82) we get the well-known $\sin(x)/x$ characteristic

$$\tilde{h}(k) = \frac{\sin(\pi k/L)}{(\pi k/L)}, \quad k = 0, \pm 1, \pm 2, \cdots. \tag{88}$$

In a similar manner we can determine the ideal time responses of the polyphase filters, either by taking the inverse transform of (87), or by sampling the above time response $\tilde{h}(k)$ according to (71). The net result is that the ideal time responses of the polyphase filters are

$$\tilde{p}_\rho(n) = \frac{\sin[\pi(n + \rho/L)]}{\pi(n + \rho/L)}, \quad \rho = 0, 1, 2, \cdots, L - 1, \text{ and all } n. \tag{89}$$

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

$$\tilde{h}(k) = \begin{cases} 1, & k = 0 \\ 0, & k = rL, r = \pm 1, \pm 2, \cdots. \end{cases} \tag{90}$$

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

$$\tilde{p}_0(n) = \delta(n), \quad \text{for all } n. \tag{91}$$

In terms of the polyphase structure of Fig. 14 and its signal processing interpretation of Fig. 15, 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 in between these values must be interpolated by the polyphase filters $p_\rho(m)$, $\rho = 1, 2, \cdots, L - 1$. 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 $m = 0, \pm L, \pm 2L, \cdots$. However "in-between" these samples, the error will always be nonzero.

By choosing a design that does not specifically satisfy the constraint of (90) or (91) a tradeoff can be made between errors that occur at sample times $m = 0, \pm L, \pm 2L, \cdots$ and errors that occur between these samples.

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

$$\tilde{h}(k) = \tilde{h}(-k). \tag{92}$$

(Alternatively, for practical systems it may be symmetrical about some fixed nonzero delay.) This symmetry does not necessarily extend directly to the polyphase filters since they correspond to sampled values of $\tilde{h}(k)$ offset by some fraction of a sample. Their envelopes, however, are symmetrical (see Fig. 15).

The above symmetry property does, however, imply a form of mirror image symmetry between pairs of polyphase filters $\tilde{p}_\rho(n)$ and $\tilde{p}_{L-\rho}(n)$. Applying (92) to (71) gives

$$\begin{aligned} \tilde{p}_\rho(n) &= \tilde{h}(-\rho - nL) \\ &= \tilde{h}(L - \rho - (n + 1)L). \end{aligned} \tag{93}$$

Also noting that

$$\tilde{p}_{L-\rho}(n) = \tilde{h}(L - \rho + nL) \tag{94}$$

it can be seen that this symmetry is of the form

$$\tilde{p}_\rho(n) = \tilde{p}_{L-\rho}(-n - 1). \tag{95}$$

In the case of decimators it is not possible to identify the outputs of specific polyphase branches with specific output samples of the network. All branches contribute to each output. Thus it is not as convenient to give meaningful time domain interpretations to the operation of the filter in a decimator.

The ideal time responses for $\tilde{h}(k)$ and $\tilde{p}_\rho(n)$ for decimators, however, are the same as those of (88) and (89), respectively, with L replaced by M.

D. Filter Design Procedures

In the remainder of this section we will discuss a number of filter design procedures which apply to the design of multirate systems. Since the filter design problem for such systems generally is a low-pass (or bandpass) design problem, nearly all of the work in digital signal processing filter theory can be brought to bear on this problem. We will not attempt to discuss all of these methods in detail, since they are well documented elsewhere [11], [17], but rather we will try to point to the relevant issues involved in these designs that particularly apply to multirate systems.

We will discuss five main categories of filter design procedures, namely:

1) window designs [11], [17], [26];
2) optimal, equiripple linear phase designs [27]-[30];
3) half-band designs [31]-[33];
4) special FIR interpolator designs based on time domain filter specifications [34]-[38], or stochastic properties of the signal [39];
5) classical interpolation designs, namely linear and Lagrangian interpolators [4], [16].

E. FIR Filters Based on Window Designs

One straightforward approach to designing FIR filters for decimators and interpolators is by the well-known method of windowing or truncating the ideal prototype response $\tilde{h}(k)$. A direct truncation (i.e., a rectangular windowing of $\tilde{h}(k)$), however, leads to the Gibbs phenomenon which manifests itself as a large (9 percent) ripple in the frequency behavior of the filter in the vicinity of filter magnitude discontinues (i.e., near the edges of the passband and stopband). Furthermore, the amplitude of this ripple does not decrease with increasing duration of the filter (it only becomes narrower in width).

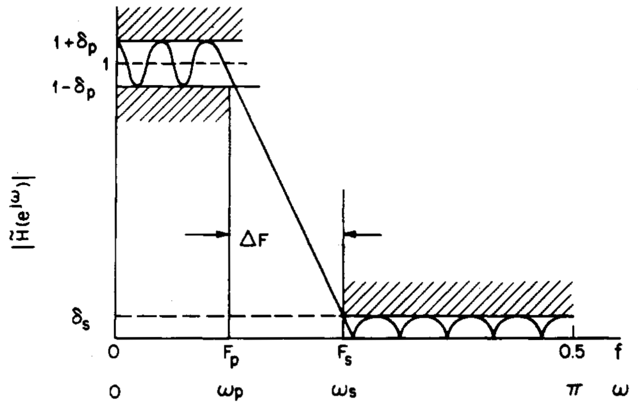


Fig. 25. Illustration of a tolerance scheme for a practical low-pass filter.

Thus a direct truncation, or a rectangular windowing, of $\tilde{h}(k)$ is rarely used in practice.

A more successful way of windowing the ideal characteristic $\tilde{h}(k)$ is by more gradually tapering its amplitude to zero near the ends of the filter with a weighting sequence $w(k)$ known as a window. The resulting filter design $h(k)$ is thus the product of the window $w(k)$ with the ideal response $h(k)$, i.e.,

$$h(k) = \tilde{h}(k)\, w(k), \qquad -(N-1)/2 \leq k \leq (N-1)/2 \tag{96}$$

where we assume that $w(k)$ is a symmetric N-point (N odd) window. A number of windows have been proposed in the literature for controlling the effects of the Gibbs phenomenon, and their properties are well understood [11], [17], [26]. Two commonly used types of windows are the "generalized" Hamming windows and the Kaiser windows.

The window designs have the property that they preserve the zero crossing pattern of $\tilde{h}(k)$ in the actual filter design $h(k)$. Thus if $\tilde{h}(k)$ is obtained from (88), then the time-domain properties discussed in Section IV-C apply to this class of filters.

Among the advantages of the window design approach is that it is simple, easy to use, and can readily be implemented in a direct manner (i.e., closed form expressions are available for the window coefficients, hence the filter responses can be obtained simply from the ideal filter response). Among the disadvantages are that there is only limited control in choosing cutoff frequencies and passband and stopband errors for most cases. The resulting filter designs are also suboptimal in that a smaller value of N can be found (using other design methods) such that all specifications on the filter characteristics are met or exceeded.

F. Equiripple (Optimal) FIR Designs

The windowing technique of the previous section represented a simple straightforward approach to the design of the digital filter required in all sampling rate conversion systems. However considerably more sophisticated design techniques have been developed for FIR digital filters [11], [27]–[30]. One such technique is the method of equiripple design based on Chebyshev approximation methods. The filters designed by this technique are optimal in the sense that the peak (weighted) approximation error in the frequency domain over the frequency range of interest is minimized.

To apply the method of equiripple design to the digital filter required for sampling rate conversion, a tolerance scheme on the filter must be defined. Fig. 25 shows an example of a tolerance scheme for a low-pass filter where

δ_p ripple (deviation) in the passband from the ideal response

δ_s ripple (deviation) in the stopband from the ideal response

F_p passband edge frequency $= \omega_p/2\pi$

F_s stopband edge frequency $= \omega_s/2\pi$

N number of taps in the FIR filter.

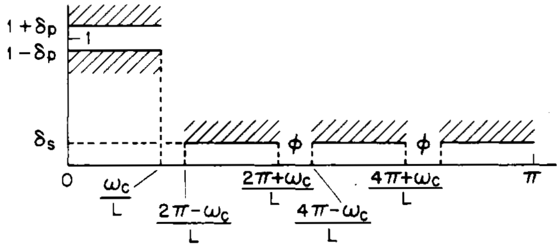


Fig. 26. Illustration of a tolerance scheme for multistopband interpolator filters.

Alternatively, Fig. 26 shows a practical tolerance scheme for a multistopband design of a 1 to L interpolator when it is known that the input spectrum does not occupy its full bandwidth (see Section IV-B and Fig. 24).

Given the tolerance schemes of Figs. 25 or 26, it is a simple matter to set up a filter approximation problem based on Chebyshev approximation methods. Several highly developed techniques have been presented in the literature for solving the Chebyshev approximation problem [27]–[29], including a well documented, widely used computer program [30]. The solutions are based on either a multiple exchange Remez algorithm, or a single exchange linear programming solution. We will not be concerned with details of the various solution methods.

For the case of the low-pass characteristic of Fig. 25 an empirical formula has been derived that relates the filter parameters. It can be expressed in the form

$$N \cong \frac{D_\infty(\delta_p, \delta_s)}{(\omega_s - \omega_p)/(2\pi)} - f(\delta_p, \delta_s)\frac{(\omega_s - \omega_p)}{2\pi} + 1 \tag{97a}$$

$$\approx \frac{D_\infty(\delta_p, \delta_s)}{(\omega_s - \omega_p)/(2\pi)} \text{ (when } \omega_s - \omega_p \text{ is small)} \tag{97b}$$

$$D_\infty(\delta_p, \delta_s) = D + f(\delta_p, \delta_s)(\Delta F)^2 \tag{98}$$

$$D = (N-1)\Delta F = (N-1)(\omega_s - \omega_p)/2\pi \tag{99}$$

$$D_\infty(\delta_p, \delta_s) = \log_{10}\delta_s[a_1(\log_{10}\delta_p)^2 + a_2\log_{10}\delta_p + a_3] + [a_4(\log_{10}\delta_p)^2 + a_5\log_{10}\delta_p + a_6] \tag{100}$$

$$f(\delta_p, \delta_s) = 11.012 + 0.512(\log_{10}\delta_p - \log_{10}\delta_s), \quad \text{for } |\delta_s| \leq |\delta_p| \tag{101}$$

and where $a_1 = 0.00539$, $a_2 = 0.07114$, $a_3 = -0.4761$, $a_4 = -0.00266$, $a_5 = -0.5941$, and $a_6 = -0.4278$.

Although the design relationships of (97)–(101) appear complex, they are fairly simple to apply. By way of example, Fig. 27 shows a series of plots of the quantity $D_\infty(\delta_p, \delta_s)$ as a function of δ_s for several values of δ_p. Through the use of

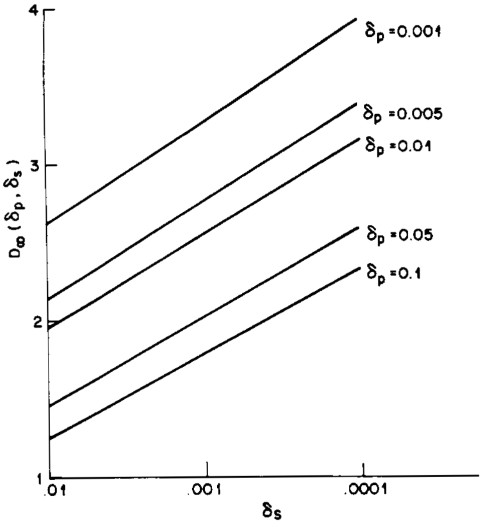


Fig. 27. Plot of $D_\infty(\delta_p, \delta_s)$ for practical values of δ_p and δ_s.

either the analytic form (e.g., (97)) or from widely available design charts and tables, it is a simple matter to determine the value of N needed for an FIR low-pass filter to meet any set of design specifications [11], [38].

The above design relationships are for low-pass filters. However, as shown in Fig. 24 and 26, digital filters for interpolation and decimation need not be strictly low-pass filters, but instead can also include ϕ, or don't care bands, which can influence the filter design problem. This is especially true when the total width of the ϕ bands is a significant portion of the total frequency band. For such cases no simple design formula such as (97) exists which relates the relevant filter parameters. Thus the simplest way of illustrating the effects of the ϕ bands on the filter design problem is by way of example.

Consider the design of a set of interpolators with specifications:

$$L = 5$$
$$\delta_p \leqslant 0.001$$
$$\delta_s \leqslant 0.0001$$
$$0 \leqslant \omega_c \leqslant \pi$$

i.e., the parameter that we allow to vary is ω_c (see Fig. 26). First we design a series of low-pass filters with passband cutoff frequencies $\omega_p = \omega_c/5$, and stopband cutoff frequencies $\omega_s = (2\pi - \omega_c)/5$. Next we design a series of multiband filters with a single passband (identical to that of the low-pass design), and a series of stopbands, separated by don't care bands (see Fig. 26). If we compare the required impulse response duration of the lowpass filter to the required impulse duration of the multiband filter we get a result of the type shown in Fig. 28(a) which gives the percentage decrease in N as a function of ω_c/π. The heavy dots shown in this figure are measured values (i.e., they are not theoretical computations), and the smooth curve shows the trend in the data. The numbers next to each heavy dot are the minimum required impulse response durations (in samples) for the low-pass, and multiband designs, respectively.

The trends in the curve are quite clear. For ω_c/π close to 1.0, there is essentially no gain in designing and using a multiband filter instead of a standard low-pass filter since the total width of the ϕ bands is small. However, as ω_c/π tends to 0, significant reductions in N are possible; e.g., for $\omega_c/\pi \approx 0.05$, a 50 percent reduction in impulse response duration is possible. Fig. 28(b) shows similar trends for a series of interpolators with a value of $L = 10$.

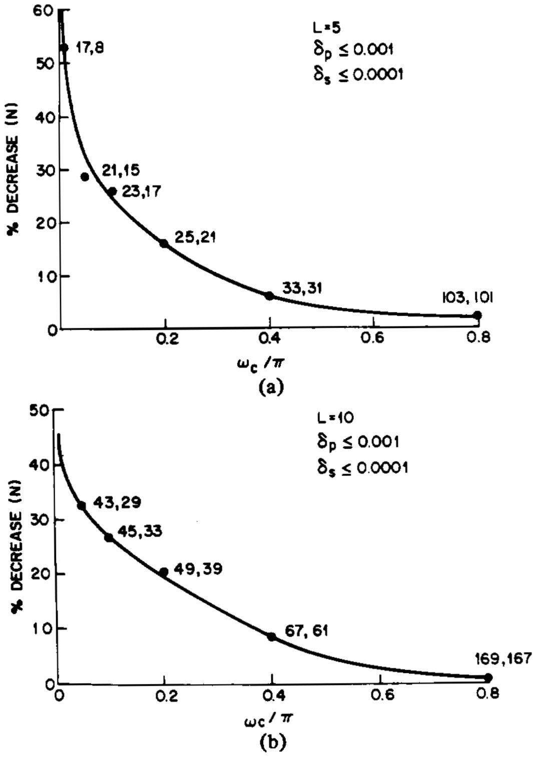


Fig. 28. Percentage decrease in required filter order by using a multiband design instead of a low-pass design, as a function of ω_c/π for (a) $L = 5$ and (b) $L = 10$, respectively.

Figs. 29 and 30 show typical frequency responses for low-pass and multiband interpolation filters. For the example of Fig. 29(a) the specifications were $L = 5$, $\delta_p = 0.001$, $\delta_s = 0.0001$, and $\omega_c = 0.5\pi$. The required values of N were 41 for the low-pass filter, and 39 for the multiband design. Thus for this case the reduction in filter order was insignificant. However, the change in filter frequency response (as seen in Fig. 29(b)) was highly significant. Fig. 30 shows the same comparisons for designs with $L = 10$, $\delta_p = 0.001$, $\delta_s = 0.0001$, and $\omega_c = 0.5\pi$. In this case a 28-percent reduction in filter order (from 45 for the low-pass filter to 33 for the multiband filter) was obtained since the frequency span of the stopbands was small compared to the frequency span of the don't care bands. In both these examples we see that, in the ϕ bands, the frequency response of the resulting filter is truly unconstrained, and can, in fact, become extremely large (as compared to the passband response). The practical implication is that some care must be taken to ensure that the amplitude response in the ϕ bands stays below some well specified level to guarantee that the noise in the input signal, in the ϕ bands, is not amplified so much that it becomes excessive in the output signal.

The above examples have shown that when ω_c/L, the cutoff frequency of the passband, is relatively small (compared to π/L), significant reductions in computation can be obtained by exploiting the don't care bands in the design of the interpolation (or decimation) digital filter. In practice such conditions

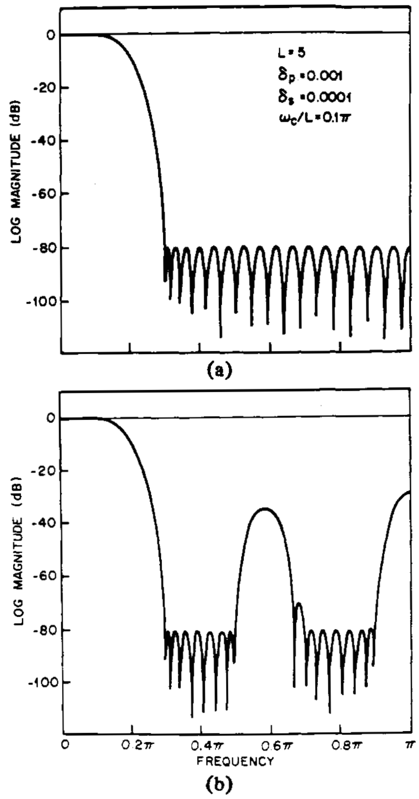


Fig. 29. Comparison between (a) low-pass filter, and (b) its equivalent multiband design for $L = 5$.

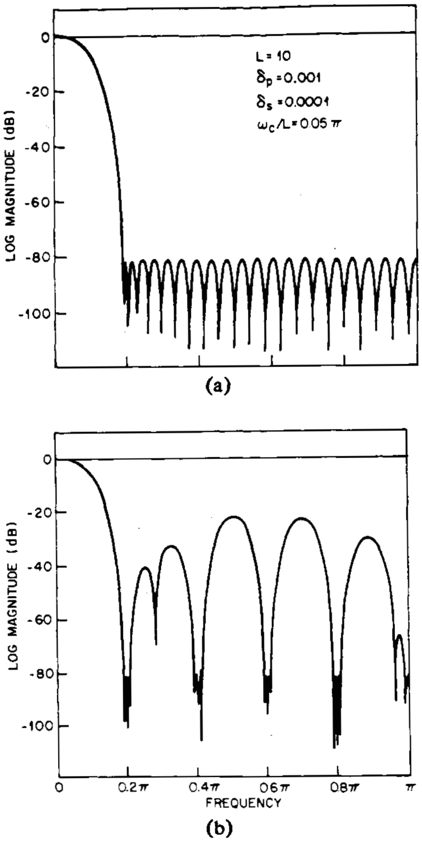


Fig. 30. Comparison between (a) a low-pass filter, and (b) its equivalent multiband design for $L = 10$.

rarely occur, i.e., we are usually dealing with signals where ω_c/L is relatively large. As such the question arises as to whether the multiband filter approach is of practical utility in the implementation of digital sampling rate changing systems. We will see in Section V that the techniques discussed here are of value in multistage implementation of decimators and interpolators involving large changes in sampling rates.

G. Half-Band FIR Filters–A Special Case of FIR Designs for Conversion by Factors of Two

Let us again consider the tolerance specifications of the ideal low-pass filter of Fig. 25. If we consider the special case

$$\delta_s = \delta_p = \delta \tag{102}$$

$$\omega_s = \pi - \omega_p \tag{103}$$

then the resulting equiripple optimal solution to the approximation problem has the property that

$$H(e^{j\omega}) = 1 - H(e^{j(\pi-\omega)}) \tag{104}$$

i.e., the frequency response of the optimal filter is symmetric around $\omega = \pi/2$, and at $\omega = \pi/2, H(e^{j\omega}) = 0.5$. It can also be readily shown that any symmetric FIR filter satisfying (104), also satisfies the ideal time-domain constraints discussed in Section IV-C, equation (90), i.e., every other impulse response coefficient (except for $k = 0$) is *exactly* 0.

Filters designed using the constraints of (102) and (103) have been called "half-band" filters [31]-[33], and their properties have been intensively studied. They can be designed in a variety of ways including the window designs and equiripple designs discussed previously.

H. Minimum Mean-Square-Error Design of FIR Interpolators–Deterministic Signals

Thus far we have considered the design of filters for interpolators and decimators from the point of view of designing the prototype filter $h(m)$ such that it satisfies a prescribed set of frequency-domain specifications. In the remainder of Section IV we consider an alternative point of view in designing filters (particularly for integer interpolators). 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 [34]-[37], [39] which are capable of accounting directly for the spectrum of the signal being interpolated.

Fig. 31(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 $\tilde{h}(m)$ whose characteristics were derived in Sections IV-B and IV-C. This signal error is defined as

$$\Delta y(m) = y(m) - \tilde{y}(m) \tag{105}$$

where $y(m)$ is the output of the actual interpolator and $\tilde{y}(m)$ is the ideal output.

In this section we will consider interpolator designs which minimize the mean-square value of $\Delta y(m)$, defined as

$$E^2 = \|\Delta y(m)\|^2 = \lim_{K\to\infty} \frac{1}{2K+1} \sum_{m=-K}^{K} \Delta y(m)^2 \tag{106a}$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} |\Delta Y(e^{j\omega'})|^2 \, d\omega'. \tag{106b}$$

Later in Section IV-I we will consider designs which minimize the maximum value of $|\Delta Y(e^{j\omega'})|$ over a prescribed frequency range, and in Section IV-J we will refer to designs which minimize the maximum value of $|\Delta y(m))|$ in the time domain.

The above design problems are greatly simplified by considering them in the framework of the polyphase structures as illustrated in Fig. 14. Here it is seen that the signal $y(m)$ is actually composed of interleaved samples of the signals $u_\rho(n)$, $\rho = 0, 1, 2, \cdots, L-1$, as shown by Fig. 31(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. 31(b), i.e.,

$$\Delta u_\rho(n) = u_\rho(n) - \tilde{u}_\rho(n). \tag{107}$$

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 which 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

$$E^2 = \|\Delta y(m)\|^2 = \frac{1}{L} \sum_{\rho=0}^{L-1} E_\rho^2 \tag{108a}$$

where

$$E_\rho^2 = \|\Delta u_\rho(n)\|^2. \tag{108b}$$

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

In order to analytically set up the filter design problem, it can be noted that the ideal polyphase filter response is

$$\tilde{P}_\rho(e^{j\omega}) = e^{j\omega\rho/L} \tag{109}$$

which then leads to the form

$$E_\rho^2 = \|\Delta u_\rho(n)\|^2$$
$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} |P_\rho(e^{j\omega}) - e^{j\omega\rho/L}|^2 |X(e^{j\omega})|^2 \, d\omega. \tag{110}$$

Equation (110) 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 $P_\rho(e^{j\omega})$ and a linear (fractional sample) phase delay $e^{j\omega\rho/L}$, weighted by the spectrum of the input signal $|X(e^{j\omega})|^2$, is minimized. Note also that the integral from $-\pi$ to π in (110) is taken over the frequency range of the input signal of the interpolator, not the output signal.

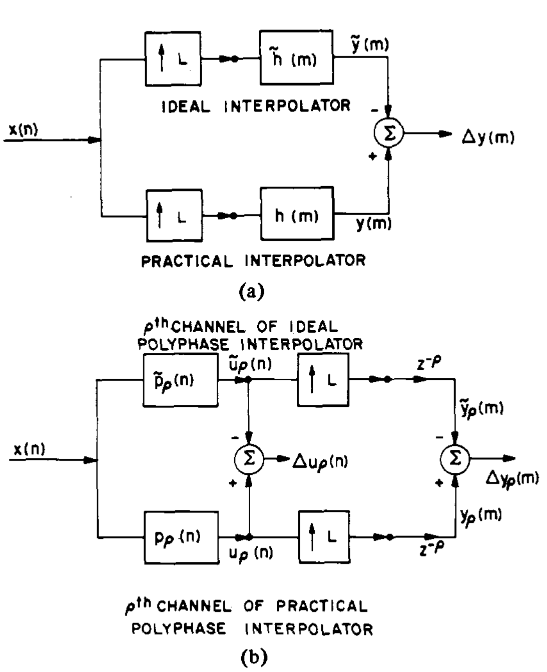


Fig. 31. Framework for defining error criteria for interpolation filters.

In practice this error criterion is often modified slightly [34]-[37], [39] by specifying that $X(e^{j\omega})$ is bandlimited to the range $0 \leqslant \omega \leqslant \alpha\pi$ where $0 < \alpha < 1$, i.e.,

$$|X(e^{j\omega})| = 0, \quad \text{for } |\omega| \geqslant \alpha\pi \tag{111}$$

Then (110) can be expressed as

$$E_{\rho,\alpha}^2 = \frac{1}{2\pi} \int_{-\alpha\pi}^{\alpha\pi} |P_\rho(e^{j\omega}) - e^{j\omega\rho/L}|^2 |X(e^{j\omega})|^2 \, d\omega \tag{112}$$

where the subscript α will be used to distinguish this norm from the one in (110). Alternatively we can consider the above modification as a means of specifying that we want the design of $P_\rho(e^{j\omega})$ to be minimized only over the frequency range $0 \leqslant \omega \leqslant \alpha\pi$, and that the range $\alpha\pi \leqslant \omega \leqslant \pi$ 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 (112) 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 of 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 which minimize $E_{\rho,\alpha}^2$. This minimization problem is solved for each value of ρ, $\rho = 0, 1, 2, \cdots, L-1$ and each solution provides the optimum solution for one of the polyphase filters. Finally, these optimum polyphase filters can be combined, as in (73)-(75), to obtain the optimum prototype filter $h(m)$ which minimizes the overall norm. The details for this approach can be found in [35]. Also, this same reference, or [40] by Oetken, Parks, and Schuessler, contains a computer program which designs minimum mean-square interpolators according to the above techniques and it 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 [36].

1) The resulting filters have the same symmetry properties as the ideal filters (92) and (95)).

2) The minimum error $\min E^2_{\rho,\alpha}$ for the polyphase filters also satisfies the symmetry condition

$$\min E^2_{\rho,\alpha} = \min E^2_{L-\rho,\alpha}. \tag{113a}$$

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 (113a). Thus the greatest error occurs in interpolating sample values which are halfway between two given samples. This normalized error is closely approximated by the sine-squared function [36], i.e.,

$$\frac{\min E^2_{\rho,\alpha}}{\min E^2_{L/2,\alpha}} \approx \sin^2\left(\frac{\rho\pi}{L}\right) \tag{113b}$$

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).

Fig. 32 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 $|X(e^{j\omega})| = 1$.

I. Design of FIR Interpolators with Minimax Error in the Frequency Domain

In the previous section we considered the design of FIR interpolators based on minimzing the mean-square error norm. In this section we will consider another class of designs of interpolation filters in which the maximum of the error $|\Delta Y(e^{j\omega'})|$ over the frequency range of interest is minimized. This type of design gives a greater degree of control over the errors at specific frequencies [36]. In minimax designs of this type, the error $\Delta Y(e^{j\omega'})$ oscillates (ripples) in the frequency domain between positive and negative values of this maximum error. If the number of taps in the overall filter is

$$N = 2RL + 1 \tag{114}$$

then each polyphase filter has $2R$ taps and therefore $2R$ degrees of freedom. These $2R$ degrees of freedom allow the error $|\Delta Y(e^{j\omega'})|$ to be exactly zero at $2R$ frequencies or R conjugate pairs of frequencies. Thus $\Delta Y(e^{j\omega'})$ will have $R+1$ extremal values in the range of positive frequencies $0 \leqslant \omega' \leqslant \alpha\pi/L$. The frequencies at which $|\Delta Y(e^{j\omega'})|$ is zero will be denoted as

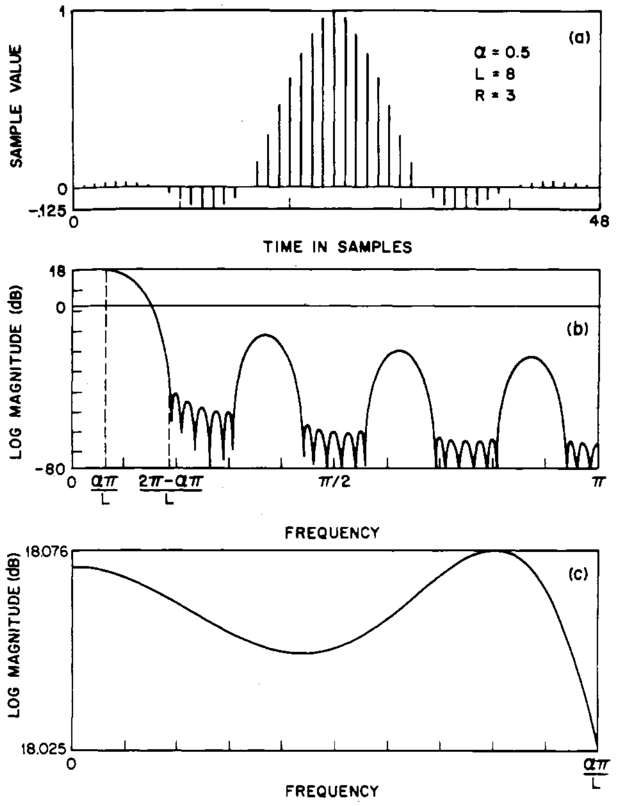


Fig. 32. The impulse and frequency responses of a minimum mean-squared error interpolation filter with $\alpha = 0.5$, $R = 3$, and $L = 8$.

$$0 \leqslant \omega_\lambda = \omega'_\lambda L \leqslant \alpha\pi, \qquad \lambda = 1, 2, \cdots, R \tag{115}$$

where the reader may recall that ω refers to frequencies specified with reference to the low sampling rate and ω' refers to frequencies specified with reference to the high sampling rate. From (110) it then follows that

$$P_\rho(e^{j\omega_\lambda}) = e^{j\omega_\lambda \rho/L}, \quad \text{for } \lambda = 1, 2, \cdots, R, \quad \text{and for } \rho = 0, 1, 2, \cdots, L-1. \tag{116}$$

Thus at these frequencies the polyphase filter responses $P_\rho(e^{j\omega_\lambda})$ are all equal to the ideal filter responses $\tilde{P}_\rho(e^{j\omega_\lambda})$, $\lambda = 1, 2, \cdots, R$.

The above problem has now been converted to a problem of finding the set of R frequencies ω_λ, $\lambda = 1, 2, \cdots, R$. The assumption that these frequencies be distinct is not essential to the validity of the result. Indeed if we assume that $\omega_\lambda = 0$ for all λ and $X(e^{j\omega}) =$ constant for $|\omega| \leqslant \alpha\pi$, then all of the zeros of $\|\Delta Y(e^{j\omega'})\|^2$ occur at $\omega' = 0$. In this case we get a maximally flat behavior of $\|\Delta Y(e^{j\omega'})\|^2$ such that its first $4R$ derivatives with respect to ω' are zero. The solution then leads to the well-known class of Lagrange interpolators [36].

For equiripple designs a procedure for finding the frequencies ω_λ has been proposed by Oetken [36]. First he showed that the error $\|\Delta Y(e^{j\omega'})\|^2$ could be expressed as the sum of the errors due to each polyphase filter. He then showed that the individual errors $\|\Delta Y_\rho(e^{j\omega'})\|^2$ are almost exactly proportional to each other over the whole frequency range. This proportionality has the form (similar to that of (113b))

$$\|\Delta Y_\rho(e^{j\omega'})\| \approx \|\Delta Y_{L/2}(e^{j\omega'})\| \sin(\rho\pi/L) \tag{117}$$

where $\|\Delta Y_{L/2}(e^{j\omega'})\|^2$ denotes the error for the polyphase filter $\rho = L/2$, i.e. it is the filter which interpolates samples exactly half way between the input samples. This form is a stronger condition than that of (113b) in that it applies to individual frequencies as opposed to the mean-square error integrated over the entire frequency range. The deviations to the approximation have been found to be smaller than 1 percent [36].

Because of the above condition, the design problem may be converted to a simpler problem of designing one of the polyphase filters such that it has the desired minimax error. The zeros of this filter can be calculated to obtain the frequencies ω_λ, $\lambda = 1, 2, \cdots, R$ and they can be applied to obtain the minimax solutions to the other polyphase filters. It is convenient to choose the polyphase filter $\rho = L/2$ which interpolates sample values half way between input samples. Also, as in the case of mean-square-error designs, the design of the polyphase filters is independent of L (assuming that there are always $2R$ taps for each polyphase filter). Therefore, it is convenient to choose $L = 2$ so that the above design requirements are those of a minimax half-band filter design. This design can be obtained using the techniques described in Section IV-G on half-band filters, with the appropriate weighting factor $|X(e^{j\omega})|$.

Fig. 33 shows an example of a minimax intepolator design for $\alpha = 0.5, R = 3, L = 8$, and assuming that $|X(e^{j\omega})| = 1$. Fig. 33(a) shows the individual errors $\|\Delta Y_\rho(e^{j\omega'})\|$ (note that $\|\Delta Y_\rho(e^{j\omega'})\| = \|\Delta Y_{L-\rho}(e^{j\omega'})\|$) and Fig. 33(b) shows the total error $\|\Delta Y(e^{j\omega'})\|$. Figs. 33(c)-(e) shows the impulse response and frequency response of the final prototype filter $H(e^{j\omega'})$. Note that although the error signal $\|\Delta Y(e^{j\omega'})\|$ exhibits an equal-ripple behavior (as specified by the design criterion) $H(e^{j\omega'})$ does not exhibit equal-ripple behavior.

J. Other Designs

In addition to the above design procedures a number of other methods have been proposed in the literature. In this section, we briefly refer to these methods.

Parks and Kolba [37] proposed a design technique for interpolators that minimizes the error $|\Delta y(m)|$ in a minimax sense in the time domain. Under the assumption that $|X(e^{j\omega})|$ is bandlimited and spectrally flat within this band this procedure leads to the same designs as the minimum mean-square designs of Section IV-H. Matrinex and Parks also investigated the use of IIR filters for interpolators. The interested reader is referred to [41] and [42] for details.

Other classical interpolation techniques are those based on linear and Lagrange methods [4], [16]. Linear interpolation has obvious limitations since it can be interpreted as only a two point filter. Lagrange interpolators are of historical importance and they can be shown to be related to the case of maximally flat designs at $\omega = 0$ as mentioned in Section IV-I.

This concludes our discussion of FIR design techniques for the filters for decimators and interpolators. As we have shown, there are many techniques for designing such filters, and they are all based on slightly different criteria. Some techniques are convenient because of their simplicity, some because they optimize a specific error criterion, and others are of interest strictly from a historical point of view. As in most filter design problems, it is up to the user to decide which of a set of alternative solutions to the filter design problem is most applicable to the problem at hand.

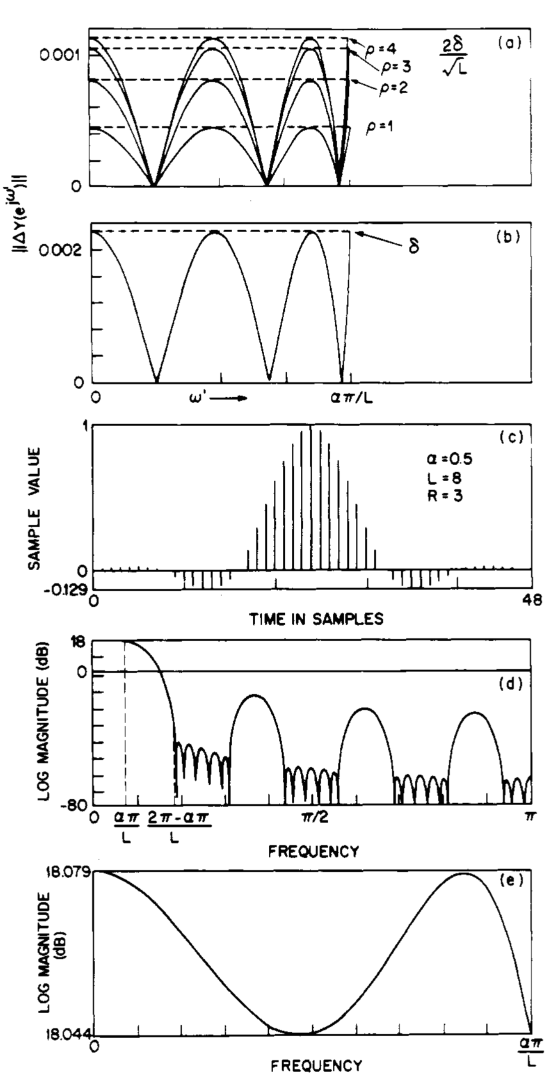


Fig. 33. The impulse and frequency responses of a minimax design interpolation filter with $\alpha = 0.5, R = 3$, and $L = 8$.

V. 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 [13], [31]-[33], [43]-[45] as shown in Figs. 34 and 35. Consider first a system for interpolating a signal by a factor of L as shown in Fig. 34(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

$$L = \prod_{i=1}^{I} L_i \tag{118}$$

where each L_i is an integer, then we can express this network in the form shown in Fig. 34(b). This structure, by itself, does not provide any inherent advantage over the structure of Fig. 34(a). However, if we modify the structure by introducing a lowpass filter between *each* of the sample rate increasing boxes, we produce the structure of Fig. 34(c). This structure has the property that the sampling rate increase occurs in a series of I

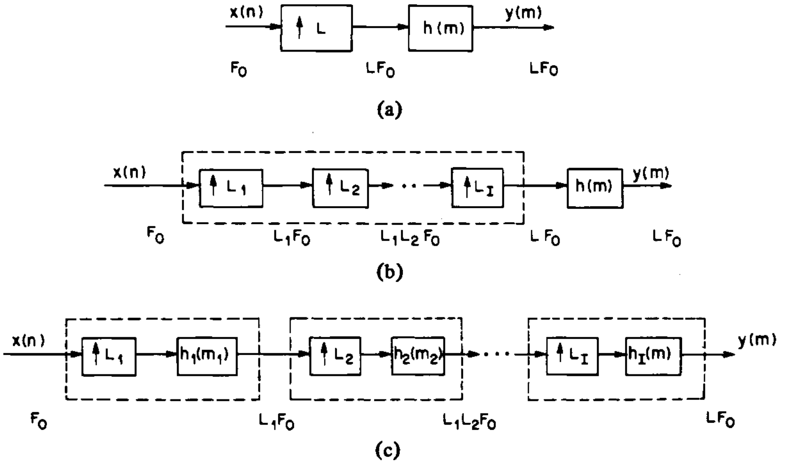


Fig. 34. Single stage and multistage structures for a 1 to L interpolator.

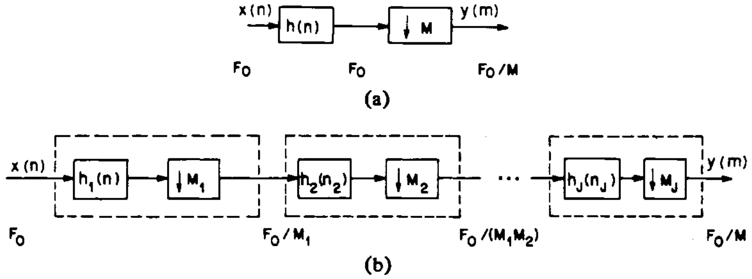


Fig. 35. Single stage and multistage structures for an M to 1 decimator.

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

$$M = \prod_{j=1}^{J} M_j \tag{119}$$

Then the general single-stage decimator structure of Fig. 35(a) can be converted into the multistage structure of Fig. 35(b). Again each of the stages within the structure of Fig. 35(b) is an *independent* decimation stage.

Perhaps the most obious question that arises from the above discussion is why 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 stage) 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. 34(c) and 35(b), are:

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, coefficient sensitivity, in the implementations of the digital filters.

These structures however are not without some drawbacks. These include:

1) increased control structure required to implement a multistage process;
2) difficulty in choosing the appropriate values of I (or J) of (118) and the best factors L_i (or M_j).

It is the purpose of this section to 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:

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

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

A. 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 presenting a simple deisign example which 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

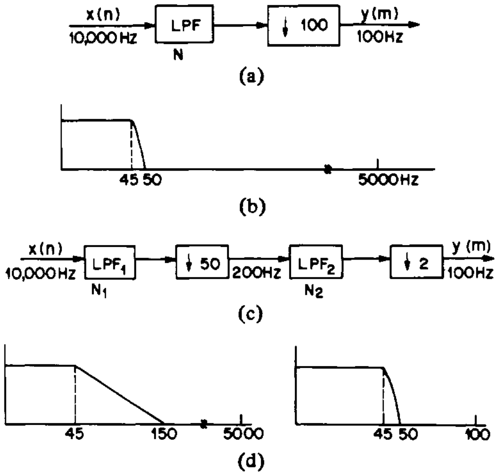


Fig. 36. Example of a one-stage and two-stage network for decimation by a factor of 100.

$M = 100$ to give the signal $y(m)$ at a 100-Hz rate. Fig. 36(a) shows the standard, single stage, decimation network which 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 low-pass filter are as shown in Fig. 36(b). We assume, for simplicity, that the design formula (97b)

$$N \approx \frac{D(\delta_p, \delta_s)}{(\Delta F/F)} \tag{120}$$

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. For the low-pass filter of Fig. 36(b) we have

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

giving, from (120), $N \approx 5080$ TAPS. The overall computation in multiplications per second necessary to implement this system is

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

i.e., a total of 250 000 multiplications per sample at the 10 000-Hz rate is required to implement the system of Fig. 36(a) (assuming the use of symmetry of $h(n)$).

Consider now the 2-stage implementation shown in Fig. 36(c). The first stage decimates the signal by a factor of 50,[6] 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. 36(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 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 low-pass filters only reduces 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 two, only slight changes occur (from 2.54 to 2.76) due to this factor. For the specific example of Fig. 36(c) we get (for the first stage)

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

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

$$= 26\,300 \text{ multiplications per second.}$$

For the second stage we get

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

$$R_2 = \frac{110.4(200)}{(2)(2)} = 5500 \text{ multiplications per second.}$$

The total computation for the two-stage implementation is

$$R_1 + R_2 = 26\,300 + 5500 = 31\,800 \text{ multiplications per second.}$$

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 (120). 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 above 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. In the next section we set up a formal structure for dealing with multistage sampling rate conversion networks, and show how it can be used in a variety of implementations.

[6] We will explain later in this section how the individual decimation factors of the stages are obtained.

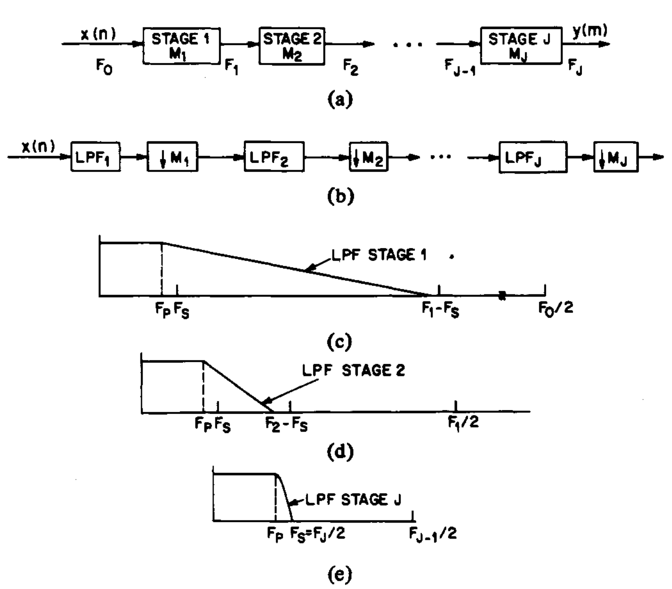


Fig. 37. Signal processing operations and filter specifications for a J-stage decimator.

B. Parameter Specifications for Multistage Implementations

Consider the J-stage decimator of Fig. 35(b) where the total decimation rate of the system is M. The sampling rate (frequency) at the output of the jth stage ($j = 1, 2, \cdots, J$) is

$$F_j = \frac{F_{j-1}}{M_j}, \quad j = 1, 2, \cdots, J \tag{121}$$

with initial input sampling frequency F_0, and final output sampling frequency F_J, where

$$F_J = \frac{F_0}{\prod_{j=1}^{J} M_j} = \frac{F_0}{M}. \tag{122}$$

Any of the structures discussed in Section III can be used for each stage of the network. Using the ideas developed above, we define the frequency range of the output signal $y(m)$ as

$$0 \leqslant f \leqslant F_p \qquad \text{Passband} \tag{123a}$$

$$F_p \leqslant f \leqslant F_s = \frac{F_J}{2} \qquad \text{Transition Band.} \tag{123b}$$

In each stage of the processing, the baseband from 0 to F_s must be protected from aliasing. Any other frequency band can be aliased in an early stage since subsequent processing will remove any signal components from the band as seen in Fig. 37. For the first stage the passband is defined from

$$0 \leqslant f \leqslant F_p \qquad \text{Stage 1 Passband} \tag{124a}$$

but the transition band is from

$$F_p \leqslant f \leqslant F_1 - F_s \qquad \text{Stage 1 Transition Band.} \tag{124b}$$

The transition band will alias back upon itself (after the decimation by M_1) only from $f = F_s$ up to $f = F_1/2$; hence the baseband of (123) is protected against aliasing. The stopband of the Stage 1 low-pass filter is from

$$F_1 - F_s \leqslant f \leqslant F_0/2 \qquad \text{Stage 1 Stopband.} \tag{124c}$$

For the second stage of the system, the specifications on the low-pass filter are defined for

$$0 \leqslant f \leqslant F_p \qquad \text{Stage 2 Passband} \tag{125a}$$

$$F_2 - F_s \leqslant f \leqslant F_1/2 \qquad \text{Stage 2 Stopband} \tag{125b}$$

as shown in Fig. 37(d). Again the band from $0 \leqslant f \leqslant F_s$ is protected from aliasing in the decimation stage.

For the rth stage of the system, the low-pass filter band specifications are

$$0 \leqslant f \leqslant F_p \qquad \text{Stage } r \text{ Passband} \tag{126a}$$

$$F_r - F_s \leqslant f \leqslant F_{r-1}/2 \qquad \text{Stage } r \text{ Stopband.} \tag{126b}$$

Alternatively if it is permissible to allow aliasing into the transition region F_p to F_s, F_s in (123)-(126) can be replaced by F_p.

It is readily seen from (122)-(126) and Fig. 37 that, for the last stage, the transition band of the low-pass filter is the same as the transition band of the one-stage implementation filter. However, the sampling rate of the system is substantially reduced in most cases.

Up to now we have been concerned solely with the regions of definition of the individual low-pass filter frequency bands. Another consideration in the design equations is the magnitude specifications on the filter response in each of the frequency bands. If it is desired that the overall passband response for the cascade of J stages be maintained within $1 \pm \delta_p$, it is necessary to require more severe constraints on the passband ripple of the individual filters in the cascade. A convenient choice which will satisfy this requirement is to specify the passband ripple constraints for each stage j to be within $1 \pm \delta_{pj}$ where $\delta_{pj} = \delta_p/J$. In the stopband the ripple constraint for the composite filter must be δ_s and this constraint must be imposed on each of the individual low-pass filters as well in order to suppress the effects of aliasing.

Fig. 38 illustrates the signal processing operations and low-pass filter specifications for an I-stage implementation of an interpolator with an overall change in sampling rates of 1 to L. One small change in notation is used in defining the I-stage interpolator—namely the stages are numbered backwards from I to 1. The reason we do this is to clearly show that the I-stage interpolator with interpolation factors L_i is a dual of the I-stage decimator with decimation factors L_i. This may be trivially seen by taking the transpose of the network of Fig. 38(a). The result is the network of Fig. 37(a) (with $M_i = L_i$ and $J = I$). Hence, in order to understand the behavior and properties of multistage structures for interpolators, we need only study the multistage decimator.

Given the network structure of Fig. 37 for the J-stage decimator, we will be interested in the specification of the following:

1) the number of stages J to realize an overall decimation factor of M;
2) the choice of decimation factors $M_j, j = 1, 2, \cdots, J$, that are appropriate for the chosen implementation;
3) the types of digital filters used in each stage of the structure—e.g., FIR versus IIR designs, equiripple versus half-band versus specialized designs;
4) the structure used to implement the filter chosen for each stage;
5) the required filter order (impluse response duration) required in each stage of the structure;

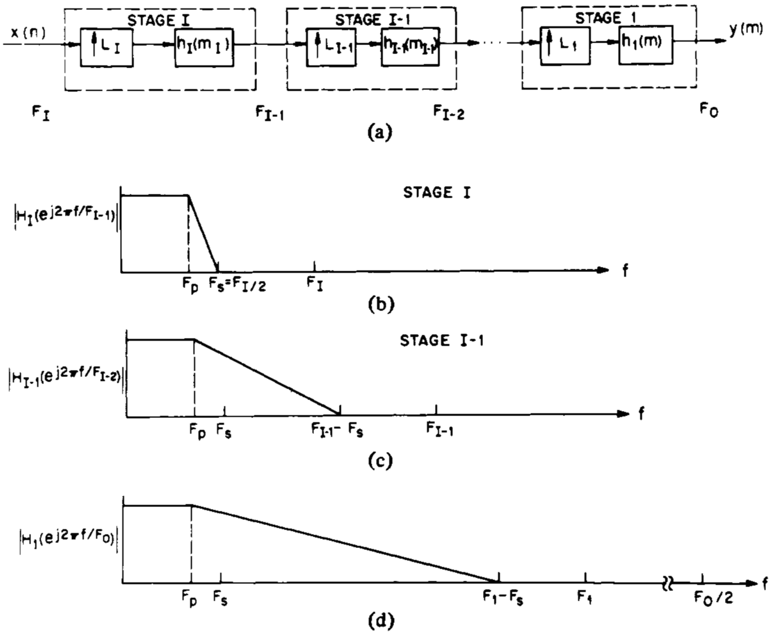


Fig. 38. Signal processing operations and filter specifications for an I-stage interpolator.

6) the resulting amount of computation and storage required for each stage, and for the overall structure.

As in most signal processing design problems, there are a number of factors that influence each of the above choices, and it is not a simple matter to select any one choice over all others.

Three distinct approaches to the design of multistage structures have been proposed by various authors. In one approach, suggested by Bellanger *et al.* [32], [33] and Rorabacher [31], a choice of factors of $M_j = 2$ for all stages is suggested in order to take advantage of the properties of half-band filters. Shively [44] and Crochiere and Rabiner [13], [49] have suggested an approach in which general equiripple filters are used and the choice of the factors M_j are chosen through an optimization procedure which minimizes the overall computation rate (or storage) and the number of stages. Finally, Goodman and Carey [45] have suggested a family of specific filter designs which can be applied to early stages of multistage decimators or final stages of multistage interpolators.

In practice any of the above approaches or any combination of them can be applied to obtain a multistage implementation. The tradeoffs are highly applications dependent. Furthermore any of the structures discussed in Section III can be be used to implement individual stages. In the following discussion we will briefly discuss the above approaches.

C. Half-Band Designs

The half-band filter structure is based on the symmetrical half-band FIR filter discussed in Section IV-G. Such filters naturally have the property that approximately half of the filter coefficients are exactly zero. Hence the number of multiplications in implementing such filters is half of that needed for a linear phase design, and a quarter of that needed for an arbitrary phase FIR filter. The half-band filter is appropriate only for sampling rate changes of 2 to 1. Hence the half-band multistage structure consists of a cascade of J-stages with 2 to 1 reductions in the sampling rate, and, if necessary, a final stage with a reduction of M_ϵ/L_ϵ in sampling rate [32], [33] where

$$M_\epsilon/L_\epsilon = M/2^J \tag{127}$$

This final stage can be realized by any of the structures for the general L/M sampling rate conversion discussed in Section III.

Several constraints must be observed with the half-band filters, as discussed in Section IV-G. First, the filter tolerances in the passband and stopband must be identical as seen from (102). Therefore, the smallest of the two required tolerances δ_p or δ_s, for the passband or stopband, respectively, must be used in the design for both bands. A second constraint is that the filter response for the half-band filter is symmetrical about one quarter of the sampling rate, as seen by (104) and the attenuation at this frequency is only 6 dB. Therefore, if the last stage of a decimator (or the first stage of an interpolator) is a 2 to 1 stage, a filter other than a half-band design may still be needed if this attenuation is not sufficient. We now consider a simple design to illustrate this method.

Design Example 1—Consider the design of a six stage structure to decimate a signal by a factor of $M = 2^6 = 64$, with filter specifications

$$\delta_p = 0.01, \quad \delta_s = 0.001, \quad F_p = 0.45 \text{ Hz}, \quad F_s = 0.5 \text{ Hz}$$

The input sampling frequency is $F_0 = 64$ Hz, and the output sampling frequency is $F_6 = F_0/64 = 1.0$ Hz. For a single-stage implementation we get

$$N = \frac{D_\infty(0.01, 0.001)}{(0.05)/64} = (2.54)\,(20)\,(64) = 3251 \text{ TAPS}$$

$$R = \frac{NF_0}{2M} = \frac{3251(64)}{2(64)} = 1625 \text{ multiplications per second}$$

where the factor of two in the above expression for R is due to the fact that the filter is symmetrical.

For a six stage half-band implementation we require:

$$\delta = \min\left(\frac{\delta_p}{6}, \delta_s\right) = 0.001$$

therefore,

$$D_\infty(0.001, 0.001) = 3.25.$$

In the implementation we will permit aliasing into the transition region F_p to F_s to permit the use of a half-band filter in the final stage. The filter order for the jth stage of the design can then be expressed as

$$N_j \cong \frac{D_\infty(0.001, 0.001)}{\Delta F_j / F_{j-1}}$$

where

$$\Delta F_j = F_j - 2F_p.$$

Therefore, the filter orders are approximately (based on the next largest odd order filter from the above estimate)

$$N_1 \simeq 7 \quad \text{TAPS}$$
$$N_2 \simeq 7 \quad \text{TAPS}$$
$$N_3 \simeq 9 \quad \text{TAPS}$$
$$N_4 \simeq 9 \quad \text{TAPS}$$
$$N_5 \simeq 13 \quad \text{TAPS}$$
$$N_6 \simeq 65 \quad \text{TAPS.}$$

The number of multiplications per second necessary to implement each of these filters is

$$R_j \simeq \tfrac{1}{4} N_j F_j$$

where the factor of $\frac{1}{4}$ is due to the fact that approximately one-half of the coefficients in the half-band filters are zero and the impulse response is symmetric. Then the multiplication rates for the stages are

$$R_1 \simeq \frac{(7)(32)}{4} = 56 \quad \text{multiplications per second}$$

$$R_2 \simeq \frac{(7)(16)}{4} = 28 \quad \text{multiplications per second}$$

$$R_3 \simeq 18 \quad \text{multiplications per second}$$

$$R_4 \simeq 9 \quad \text{multiplications per second}$$

$$R_5 \simeq 6.5 \quad \text{multiplications per second}$$

$$R_6 \simeq 16.25 \quad \text{mulitplications per second}$$

and the total rate is

$$R_T = \sum_{j=1}^{6} R_j \simeq 134 \text{ multiplications per second}$$

If we do not allow aliasing into the transition band from F_p to F_s, then all of the transition bands of the above filters become smaller by $2(F_s - F_p) = 0.1$ Hz. This does not significantly affect the orders of the filters N_1 to N_5. The transition band of the last filter N_6, however, is effectively reduced by a factor of two and it can no longer be a half-band design. Thus N_6 and R_6 become

$$N_6 \simeq 130 \text{ TAPS}$$

$$R_6 \simeq \tfrac{1}{2} N_6 F_6 = 65 \text{ multiplications per second.}$$

This increases the total rate to

$$R_T \simeq 183 \text{ multiplications per second}$$

Thus it is seen that, for the above example, the multistage structure is more efficient than the single-stage design by a factor 1625/183 = 8.9 in multiplications per second. If aliasing is permitted in the band from 0.45 to 0.5 Hz then the filter order for the single-stage design can be reduced by two and the multistage implementation is more efficient by a factor (1625/2)/134 = 6.1 in multiplications per second.

D. Designs Based on a Multistage Optimization Procedure

A second approach to the selection of parameters M_j, $j = 1, 2, \cdots, J$ and the number of stages J is based on setting up the problem in terms of an optimization procedure [13], [44], [49]. The order of the filters for each stage can be expressed as

$$N_j = \frac{D_\infty[(\delta_p/J), \delta_s]\, F_{j-1}}{(F_j - F_p - F_s)}, \qquad j = 1, 2, \cdots, J \tag{128}$$

and the computation rate (multiplications per second) can be expressed as

$$R_j = \tfrac{1}{2} N_j F_j \tag{129}$$

where the factor of $\frac{1}{2}$ is due to the assumption that the symmetry of the filter is used to reduce multiplications. (This optimization approach cannot, however, account for special features of specific filter designs such as half-band filters or those discussed in the next section.)

From (119)-(126), (128), (129) it can be shown [13] that the total multiplication rate R_T for the design can be expressed as the product of three terms

$$R_T = \frac{1}{2} \cdot D_\infty\left(\frac{\delta_p}{J}, \delta_s\right) \cdot S \cdot F_0 \text{ (multiplications per second)} \tag{130}$$

where S is a function of $J, \Delta f, M, M_1, M_2, \cdots, M_{J-1}$, i.e.,

$$S = S(J, \Delta f, M, M_1, M_2, \cdots, M_{J-1}) \tag{131}$$

and where

$$\Delta f = \frac{F_s - F_p}{F_s}. \tag{132}$$

This function can be minimized as a function of the parameters M_j, $j = 1, 2, \cdots, J$ for each choice of J by minimizing $S(\cdot)$. For $J = 2$ this can be achieved by taking the derivative of $S(\cdot)$ with respect to M_1 and setting it to zero. This leads to the ideal choice of M_1 and M_2 as

$$M_{1_{\text{opt}}} = \frac{2M(1 - \sqrt{M\Delta f/(2 - \Delta f)})}{2 - \Delta f(M + 1)} \tag{133a}$$

$$M_{2_{\text{opt}}} = \frac{M}{M_{1_{\text{opt}}}}. \tag{133b}$$

In practice the nearest integer values for M_1 and M_2 must be used.

For $J > 2$ an analytical approach is not possible. However, it is possible to find the values of M_j, $j = 1, 2, \cdots, J$ with the aid of a computer-aided optimization procedure. It was found in [13] that the Hooke and Jeaves [46], [47] procedure worked

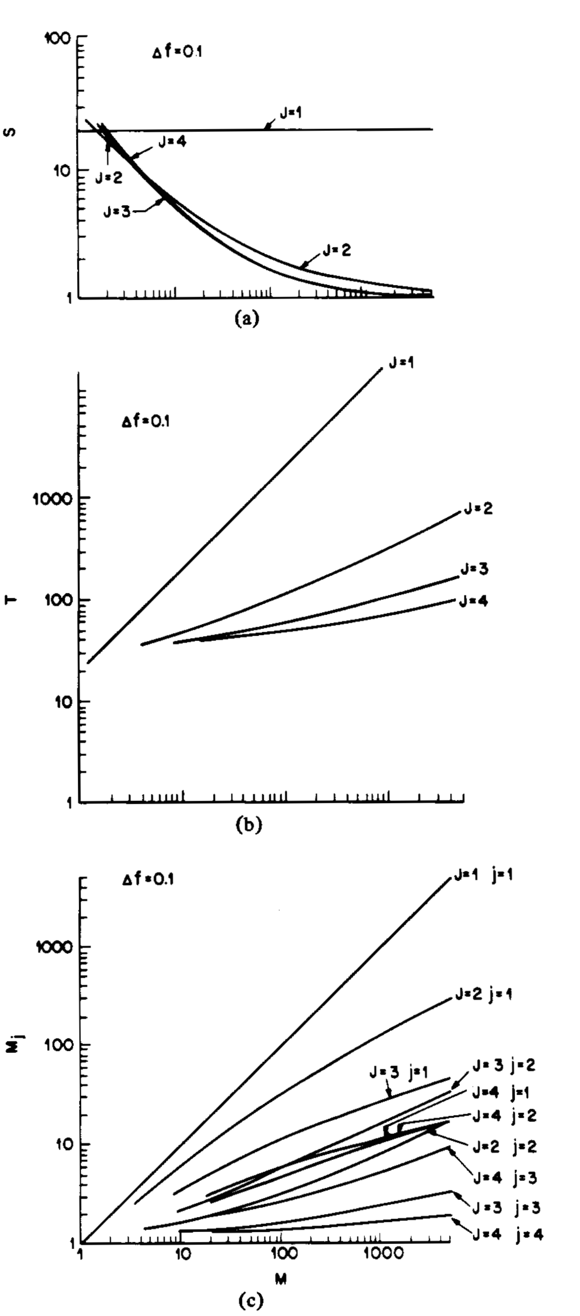


Fig. 39. Minimized values of S and T for ideal values of $M_j, j = 1, 2, \cdots, J$, and $\Delta f = 0.1$.

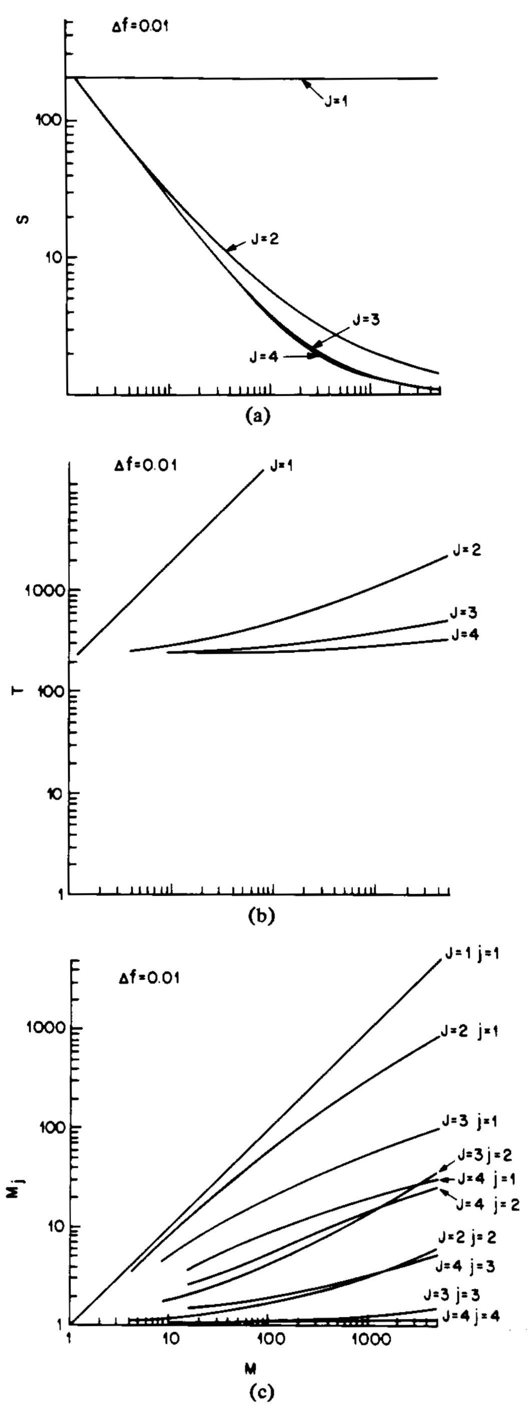


Fig. 40. Minimized values of S and T for ideal values of $M_j, j = 1, 2, \cdots, J$, and $\Delta f = 0.01$.

well on this problem and did not require the evaluation of derivatives.

A set of design curves was generated based on the above procedure and they can be found in [13]. Alternatively, a similar approach can be used for designs which minimize total storage, i.e.,

$$N_T = \sum_{j=1}^{J} N_j \tag{134a}$$

$$= D_\infty\left(\frac{\delta_p}{J}, \delta_s\right) \cdot T \tag{134b}$$

where T is another function of J, Δf, M, $M_1, M_2, \cdots, M_{J-1}$, i.e.,

$$T = T(J, \Delta f, M, M_1, M_2, \cdots, M_{J-1}). \tag{134c}$$

These design curves can be found in [49]. It turns out that designs based on minimizing N_T also result in designs that are essentially minimized for R_T as well since the minima for R_T are relatively broad whereas minima for N_T are slightly narrower. Therefore, the design curves in [49] are preferred.

Figs. 39 and 40 illustrate two example of these design curves for the cases $\Delta f = 0.1$ and $\Delta f = 0.01$, i.e., a ten percent and a one percent transition band. Figs. 39(a) and 40(a) show plots of minimized values of S in (130) and (131) as a function of M for each value of J. Figs. 39(b) and 40(b) show similar plots of minimized values of T in (134a)-(134c). Finally Figs. 39(c) and 40(c) show ideal values of M_j, $j = 1, 2, \cdots, J$ as a function of M for each J which result in the minimized values of S and T. Several important properties of this design procedure can be seen from these figures. They include the following.

1) For optimized computation, most of the gain in efficiency is achieved in a two stage structure ($J = 2$), with only small gains being achieved with three or four stage designs.

2) For optimized storage, substantial reductions in storage can still be achieved in going from two to three or four stages, although the largest decrease is obtained in going from one to two stages.

3) For optimized storage designs, the actual computation is essentially identical to that of optimized computation designs [49]. Thus a design that is minimized for storage is also minimized for computation. This result is a consequence of the fact that the function S has a broad minimum, whereas T has a somewhat narrower minimum.

4) The gain in efficiency (either computation or storage) in going from 1 to J stages increases dramatically as Δf gets smaller, and as M gets larger. For example a computational reduction of about 200 to 1 can be achieved with a three stage structure implementing a 1000 to 1 decimation for a normalized transition width of $\Delta f = 0.01$.

5) The decimation ratios of a J-stage optimized design follow the relation

$$M_j > M_{j+1} > \cdots > M_J. \tag{135}$$

6) The required computation (or storage) of an optimized J-stage design is relatively insensitive to small changes in the M_j for each stage. Thus nearest integer values of M_j can be chosen in practical designs with little loss in efficiency.

The curves of Figs. 39 and 40 provide a set of design guidelines for the optimized J-stage decimator. To illustrate how these curves can be applied we give a design example.

Design Example 2—Consider the design of a J-stage structure to decimate a signal by a factor of $M = 64$, with filter specifications $\delta_p = 0.01$, $\delta_s = 0.001$, $F_p = 0.45$ Hz, $F_s = 0.5$ Hz (this is identical to *Design Example 1*). The input sampling frequency is $F_0 = 64$ Hz, and the output sampling frequency is $F_{J6} = F_0/64 = 1.0$ Hz.

For a $J = 1$ stage design we get (see Example 1)

$N = 3251$	TAPS
$R = 1625$	multiplications per second.

For a $J = 2$ stage we can determine the ideal choices of M_1 and M_2 from (133). Alternatively, noting that

$$\Delta f = \frac{(0.5 - 0.45)}{0.5} = 0.1,$$

we may use the appropriate design curves for $\Delta f = 0.1$ in Fig. 39. Using either method we get ideal values $\tilde{M}_1 \cong 23$ and $\tilde{M}_1 \cong 2.7$. The nearest choice of integer values such that $M_1 M_2 = 64$ is either ($M_1 = 32$, $M_2 = 2$) or ($M_1 = 16$, $M_2 = 4$). We will use the second choice in this example. This leads to the design parameters:

$N_1 = 58$	TAPS
$N_2 = 221$	TAPS
$R_1 = 116$	multiplications per second
$R_2 = 111$	multiplications per second
$R_T = R_1 + R_2 = 227$	multiplications per second.

From the curves of S in Fig. 39(a) it is seen that a $J = 3$ stage design can be expected to result in a slightly lower overall computation rate. Thus following the same procedure we get

$$M_1 = 8, \quad M_2 = 4, \quad M_3 = 2$$
$$N_1 = 26 \quad R_1 = 104$$
$$N_2 = 22 \quad R_2 = 22$$
$$N_3 = 115 \quad R_3 = 57$$
$$R_T = R_1 + R_2 + R_3 = 183$$

Thus we see that a three stage design achieves a reduction in computation (multiplications/second) of a factor of 8.9 over a one-stage design or essentially the same as that of the 6-stage halfband approach.

By using the multistopband designs discussed in Section IV-F for early stages in the above two- or three-stage design it is possible to reduce the order of the filters in those stages. For example from Fig. 28 it can be estimated that for the three-stage design, N_1 can be reduced by a factor of about 22 percent or $N_1 \cong 20$ and N_2 can be reduced by a factor of about 10 percent or $N_2 \cong 20$. This results in an overall computation rate of $R_T = 157$ (multiplications per second).

E. Multistage Designs Based on a Specific Family of Filter Designs

The third approach [45] to multistage design is to use a specialized family of filter designs shown in Table I for early stages of the design and possibly a general FIR filter or half-band filter designed according to one of the methods in Section IV for the final stage. $F1$, the first filter in Table I is a filter with M_1 one's for coefficients and it can be used for arbitrary decimation factors M_1. Filters $F2$ through $F9$ are half-band filters for decimation by 2 to 1 with impulse response durations from 3 to 9 samples and short coefficient wordlengths. Filters $F2$, $F3$, and $F5$ have monotone passband frequency responses, and filters $F4$, $F6$–$F9$ have 2 or more ripples in the passband [45].

The way in which specific filters from Table I are selected for each stage of a J-stage design is as follows. The filter specifications F_p, F_s, δ_p, δ_s are defined as discussed previously. A level ripple factor D is then defined as

$$D = -20 \log (\delta_s) \qquad \text{for filters } F1, F2, F3, F5 \tag{136a}$$

or

$$D = -20 \log [\min (\delta_p/J, \delta_s)] \qquad \text{for filters } F4, F6\text{–}F9 \tag{136b}$$

and an "oversampling" frequency ratio O_j for each stage j is defined as

$$O_j = \frac{F_{j-1}}{F_J}. \tag{137}$$

The procedure at each stage is to determine the range of D and O_j such that each of the filters of Table I is applicable. This range is plotted in the chart of Fig. 41 which shows boundaries, in the plane, for which each of the filters meets specifications. The way the chart is used is as follows. For stage 1, the grid point (O_1, D) is located. If the boundary line of filter $F1$ lies

TABLE I
COEFFICIENTS OF SPECIALIZED FIR FILTERS

Filter	N	$h(0)$	$h(1)$	$h(3)$	$h(5)$	$h(7)$	$h(9)$
$F1$*	3	1	1				
$F2$	3	2	1				
$F3$	7	16	9	-1			
$F4$	7	32	19	-3			
$F5$	11	256	150	-25	3		
$F6$	11	346	208	-44	9		
$F7$	11	512	302	-53	7		
$F8$	15	802	490	-116	33	-6	
$F9$	19	8192	5042	-1277	429	-116	18

*$F1$ can be of any length with $h(i) = 1$ all i

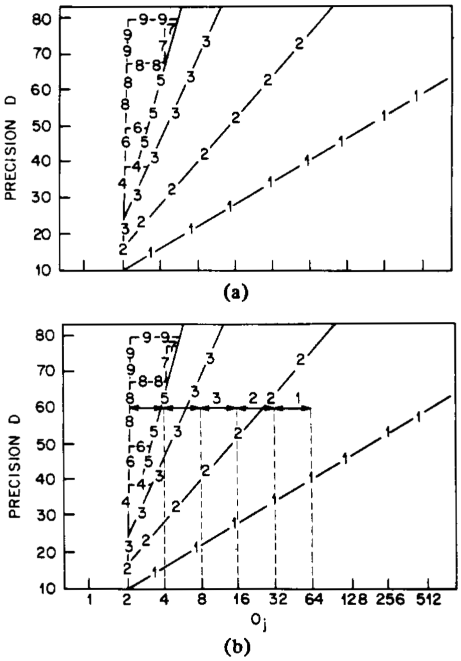


Fig. 41. Design chart showing filters required to meet specifications on D and O_j and a typical example of the use of the chart.

to the left of the grid point, filter $F1$ is used in the first stage, and the decimation rate of this stage M_1, is the largest integer such that the grid point $(O_1/M_1, D)$ also lies to the right of the $F1$ boundary line. For the second, and subsequent stages, decimation ratios of 2 are used, and the filter required is the filter whose boundary line lies to the left of the grid point $(O_j/2, D)$ for stage j. This procedure can be used until the next to last stage at which a general FIR filter is required to meet overall filter specifications.

We now illustrate this design method with the following example.

Design Example 3—To illustrate the above procedure, consider again the $M = 64$ decimator with $\delta_p = 0.01$, $\delta_s = 0.001$, $F_p = 0.45$, $F_s = 0.5$, $F_0 = 64$ Hz, and $F_J = 1$ Hz. We calculate first the ripple factor D as

$$D = -20 \log (0.001) = -60 \text{ dB } (F1, F2, F3, F5)$$

$$D = -20 \log (\min (0.01/6, 0.001)) = -60 \text{ dB } (F5, F5, F6\text{–}F9)$$

where we have used 6 as the largest possible number of stages in the design. The oversampling index for the first stage goes from 64 to 32, showing that an $F2$ filter is required, as shown in Fig. 41(b). For stages 2 and 3, filter $F3$ can be used as O_j goes from 32 to 16 (stage 2), and 16 to 8 (stage 3). For stage 4 (as O_j goes from 8 to 4) an $F5$ filter is required, whereas for stage 5 (as O_j goes from 4 to 2) an $F8$ filter is required. For the final stage, a 115 point FIR filter (the same used in Design Example 2, stage 3 of the three-stage design) is required.

The total computation (using symmetry and excluding multiplication by 1, -1, or 2) of the resulting six-stage design is:

Stage	M_j	Filter	Multiplications per second
$j = 1$	2	$F2$	$R_1 = 0$
2	2	$F3$	$R_2 = 32$
3	2	$F3$	$R_3 = 16$
4	2	$F5$	$R_4 = 16$
5	2	$F8$	$R_5 = 10$
6	2	$N_6 = 115$	$R_6 = 57$
			$R_T = 131$

From this example it can be seen that some savings in computation (multiplications per second) can be achieved over previous designs discussed. However, this is achieved at the expense of extra manipulations of the data to avoid coefficients of 2, 1, and -1.

F. Other Considerations in Multistage Designs

In the preceding sections we have considered three approaches to multistage designs of decimators or, equivalently, interpolators by use of transposition. In practice there are many factors besides computation rate in multiplications/sec that determine the tradeoffs in these designs.

One consideration is that of the cost of control of data flow in the structure when many stages are involved. Fig. 42(a) illustrates an example of a block diagram of a three-stage decimation structure and Fig. 42(b) shows the corresponding control sequence that must be used to implement this structure [49]. In practice a block of M samples of $x(n)$ are entered into the main input buffer and, after processing, one sample $y(m)$ is obtained at the output. Three state-variable buffers $S1$, $S2$, and $S3$ are used to hold internal data for the filters. The control structure consists of three loops, one within the other, to maintain the proper timing and flow of data in the structure. The actual operations are self-explanatory. In [50] a program is available to implement this structure.

Many other factors must also be considered in the design of multistage decimators and interpolators. As in most real world situations there is no simple or universal answer as to what approach is best and in practice a combination of the above techniques often yields the most appropriate tradeoffs.

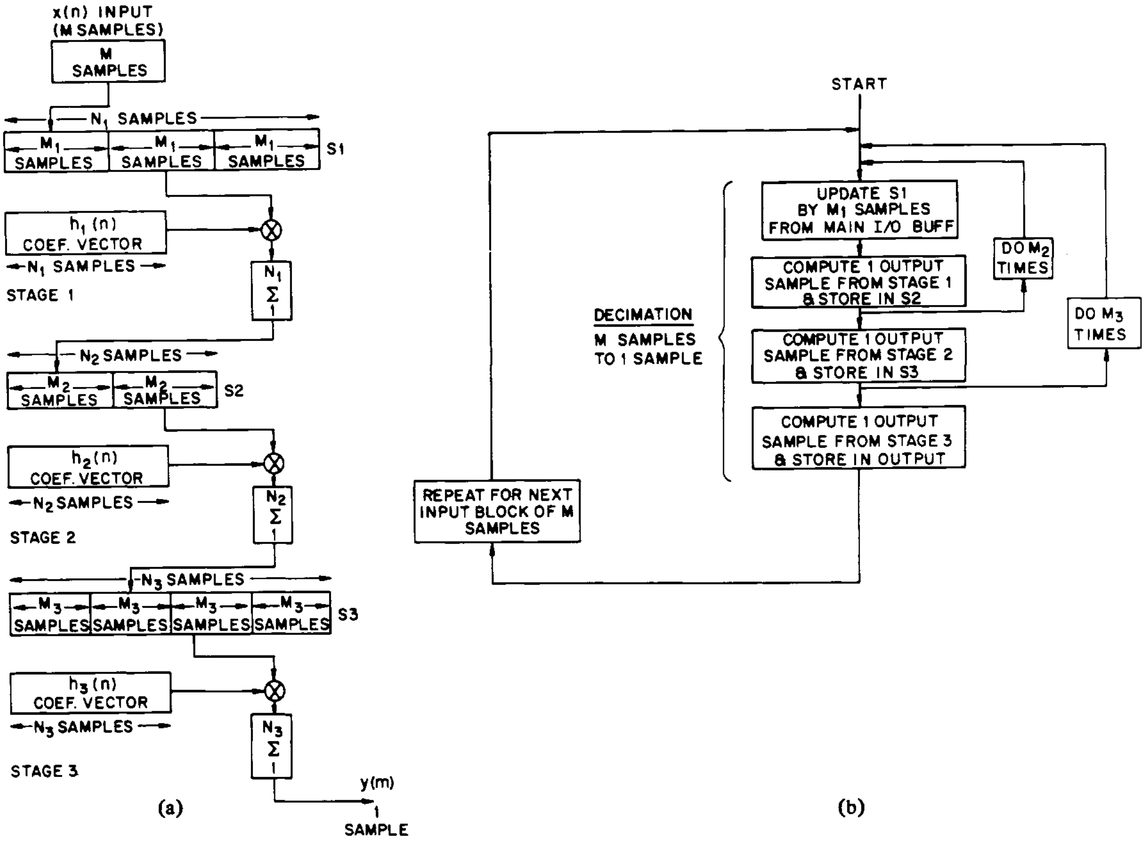


Fig. 42. Computational operations and control structure for a three-stage decimator.

VI. Discussion

In this paper, we have attempted to present a tutorial coverage of the topic of decimation and interpolation of digital signals. We began in Section II by introducing the basic concepts of sampling rate conversion and its relationship to periodically time-varying linear systems. In Section III we discussed three categories of structures, direct-form, polyphase, and time-varying structures for implementing single stage designs of decimators or interpolators. In Section IV we examined properties of ideal filters for sampling rate conversion systems and then briefly discussed a variety of filter design techniques for designing practical filters for these systems. Finally in Section V we considered multistage structures and showed that under certain circumstances they could be considerably more efficient than single stage designs. Three approaches were then presented for designing such multistage structures.

While we have covered a large scope of material in this paper there were numerous important topics that had to be left out. For example a large body of literature is developing on the use of IIR designs for sampling rate conversion, particularly where linear phase response is not of concern. References [23], [41], [42], [49] illustrate examples of some of this work.

Another area of work that has not been covered in this paper is the use of decimation and interpolation concepts for the efficient implementation or design of other digital signal processing operations. For example, the polyphase concept is very useful in designing phase shifters with fractional sample delays since each polyphase filter component by itself approximates an allpass system [51]. The cascade of a decimator followed by an interpolator also leads to an efficient approach to the implementation of narrow-band low-pass or bandpass filters [13], [32], [52], [53] and these concepts have not been covered in this paper.

References

[1] C. E. Shannon, "Communications in the presence of noise," *Proc. IRE*, vol. 37, pp. 10-21, Jan. 1949.

[2] D. A. Linden, "A discussion of sampling theorems," *Proc. IRE*, vol. 47, pp. 1219-1226, July 1959.

[3] H. Nyquist, "Certain topics in telegraph transmission theory," *Trans. AIEE*, vol. 47, pp. 617-664, Feb. 1928.

[4] R. W. Hamming, *Numerical Methods for Scientists and Engineers.* New York: McGraw-Hill, 1962.

[5] S. L. Freeny, R. B. Kieburtz, K. V. Mina, and S. K. Tewksbury, "Design of digital filters for an all digital frequency division multiplex-time division multiplex translator," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 702-711, Nov. 1971.

[6] S. L. Freeny, J. F. Kaiser, and H. S. McDonald, "Some applications of digital signal processing in telecommunications," in *Applications of Digital Signal Processing*, A. V. Oppenheim, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1978, ch. 1, pp. 1-28.

[7] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals.* Englewood Cliffs, NJ: Prentice-Hall, 1978.

[8] R. E. Crochiere, S. A. Webber, and J. L. Flanagan, "Digital coding of speech in subbands," *Bell Syst. Tech. J.*, vol. 55, no. 8, pp. 1069-1085, Oct. 1976.

[9] D. J. Goodman and J. L. Flanagan, "Direct digital conversion between linear and adaptive delta modulation formats," *Proc. IEEE Int. Communications Conf.*, (Montreal, P.Q., Canada), June 1971.

[10] R. G. Pridham and R. A. Mucci, "Digital interpolation beam forming for lowpass and bandpass signals," *Proc. IEEE*, vol. 67, no. 6, pp. 904-919, June 1979.

[11] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1975, ch. 13.

[12] J. H. McClellan and R. J. Purdy, "Applications of digital signal processing to radar," in *Applications of Digital Signal Processing*, A. V. Oppenheim, Ed., Englewood Cliffs, NJ: Prentice-Hall, 1978, pp. 239-330.

[13] R. E. Crochiere and L. R. Rabiner, "Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-23, no. 5, pp. 444-456, Oct. 1975.

[14] R. A. Meyer and C. S. Burrus, "A unified analysis of multirate and periodically time-varying digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 162-168, Mar. 1975.

[15] B. Liu and P. A. Franaszek, "A class of time-varying digital filters," *IEEE Trans. Circuit Theory*, vol. CT-16, pp. 467-471, Nov. 1969.

[16] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE*, vol. 61, pp. 692-702, June 1973.

[17] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[18] R. E. Crochiere and A. V. Oppenheim, "Analysis of linear digital networks," *Proc. IEEE*, vol. 63, pp. 581-595, Apr. 1975.

[19] A. Fettweis, "A general theorem for signal-flow networks, and applications," *Arch. Elek. Übertragung*, vol. 25, pp. 557-561, Dec. 1971. (Also in *Digital Signal Processing*, L. R. Rabiner and C. M. Rader, Eds. New York: IEEE Press, pp. 126-130, 1972.)

[20] L. B. Jackson, "On the interaction of roundoff noise and dynamic range in digital filters," *Bell Syst. Tech. J.*, vol. 49, no. 2, pp. 159-184, Feb. 1970.

[21] 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.

[22] M. G. Bellanger and G. Bonnerot, "Premultiplication scheme for digital FIR filters with application to multirate filtering," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-26, pp. 50-55, Feb. 1978.

[23] M. G. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital filtering by polyphase network: Application to sample rate alteration and filter banks," *IEEE Trans. Acous., Speech, Signal Proc.*, vol. ASSP-24, pp. 109-114, Apr. 1976.

[24] R. A. Meyer and C. S. Burrus, "Design and implementation of multirate digital filters," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-24, pp. 53-58, Feb. 1976.

[25] R. E. Crochiere, "A general program to perform sampling rate conversion of data by rational ratios," in *Programs for Digital Signal Processing*. New York: IEEE Press, 1979, pp. 8.2-1 to 8.2-7.

[26] J. F. Kaiser, "Nonrecursive digital filter design using the I_0 –Sinh window function," in *Proc. 1974, IEEE Int. Symp. Circuits and Systems*, pp. 20-23, Apr. 1974. (Also reprinted in *Digital Signal Processing, II*. New York: IEEE Press, 1975, pp. 123-126.

[27] T. W. Parks and J. H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 189-194, Mar. 1972.

[28] E. Hofstetter, A. V. Oppenheim, and J. Siegel, "A new technique for the design of nonrecursive digital filters," in *Proc. 5th Annu. Princeton Conf. Information Sciences Systems*, pp. 64-72, 1971.

[29] L. R. Rabiner, "The design of finite impulse response digital filters using linear programming techniques," *Bell Syst. Tech. J.*, vol. 51, no. 6, pp. 1177-1198, July-Aug. 1972.

[30] J. H. McClellan, T. W. Parks, and L. R. Rabiner, "A computer program for designing optimum FIR linear phase digital filters," *IEEE Trans. Audio Electracoust.*, vol. AU-21, pp. 506-526, Dec. 1973. (See also *Programs for Digital Signal Processing*. New York: IEEE Press, 1979, pp. 5.1-1 to 5.1-13.

[31] D. W. Rorabacher, "Efficient FIR filter design for sample rate reduction or interpolation," in *Proc. 1975 Int. Symp. Circuits and Systems*, Apr. 1975.

[32] M. G. Bellanger, J. L. Daguet and G. P. Lepagnol, "Interpolation, extrapolation, and reduction of computation speed in digital filters," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-22, pp. 231-235, Aug. 1974.

[33] M. G. Bellanger, "Computation rate and storage estimation in multirate digital filtering with halfband filters," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-25, pp. 344-346, Aug. 1977.

[34] G. Oetken and H. W. Schuessler, "On the design of digital filters for interpolation," *Arch. Elek. Übertragung*, vol. 27, pp. 471-476, 1973.

[35] G. Oetken, T. W. Parks and H. W. Schuessler, "New results in the design of digital interpolators," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-23, pp. 301-309, June 1975.

[36] G. Oetken, "New approaches for the design of digital interpolating filters," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-27, pp. 637-643, Dec. 1979.

[37] T. W. Parks and D. P. Kolba, "Interpolation minimizing maximum normalized error for band-limited signals," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-26, pp. 381-384, Aug. 1978.

[38] O. Herrmann, L. R. Rabiner, and D. S. Chan, "Practical design rules for optimum finite impulse response lowpass digital filters," *Bell Syst. Tech. J.*, vol. 52, no. 6, pp. 769-799, July-Aug. 1973.

[39] A. Polydoros and E. N. Protonotarios, "Digital interpolation of stochastic signals," in *Proc. Zurich Seminar on Communications*, pp. 349-367, Mar. 1978.

[40] G. Oetken, T. W. Parks, and H. W. Schuessler, "A computer program for digital interpolator design," in *Programs for Digital Signal Processing*. New York: IEEE Press, 1979, pp. 8.1-1 to 8.1-6.

[41] H. G. Martinez and T. W. Parks, "A class of infinite-duration response digital filters for sampling rate reduction," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-27, pp. 154-162, Apr. 1979.

[42] —, "Design of recursive digital filters with optimum magnitude attenuation poles on the unit circle," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-26, pp. 150-156, Apr. 1978.

[43] G. A. Nelson, L. L. Pfeifer, and R. C. Wood, "High-speed octave band digital filtering," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 58-65, Mar. 1972.

[44] R. R. Shively, "On multistage FIR filters with decimation," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-23, pp. 353-357, Aug. 1975.

[45] D. J. Goodman and M. J. Carey, "Nine digital filters for decimation and interpolation," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-25, pp. 121-126, Apr. 1977.

[46] R. Hooke and T. A. Jeaves, "Direct search solution of numerical and statistical problems," *J. Assoc. Comput. Mach.*, vol. 8, no. 4, pp. 212-229, Apr. 1961.

[47] J. L. Kuester and J. H. Mize, *Optimization Techniques with Fortran*. New York: McGraw-Hill, 1973.

[48] L. R. Rabiner, J. F. Kaiser, O. Herrmann, and M. T. Dolan, "Some comparisons between FIR and IIR digital filters," *Bell Syst. Tech. J.*, vol. 53, no. 2, pp. 305-331, Feb. 1974.

[49] R. E. Crochiere and L. R. Rabiner, "Further considerations in the design of deimators and interpolators," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-24, pp. 296-311, Aug. 1976.

[50] —, "A program for multistage decimation, interpolation, and narrow band filtering," in *Programs for Digital Signal Processing*. New York: IEEE Press, 1979, pp. 8.3-1 to 8.3-14.

[51] R. E. Crochiere, L. R. Rabiner, and R. R. Shively, "A novel implementation of digital phase shifters," *Bell Syst. Tech. J.*, vol. 54, pp. 1497-1502, Oct. 1975.

[52] L. R. Rabiner and R. E. Crochiere, "A novel implementation for narrow-band FIR digital filters," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-23, pp. 457-464, Oct. 1975.

[53] F. Mintzer and B. Liu, "The design of optimal multirate bandpass and bandstop filters," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-26, pp. 534-543, Dec. 1978.