Optimum FIR Digital Filter Implementations for Decimation, Interpolation, and Narrow-Band Filtering

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

Abstract—In this paper a general theory of multistage decimators and interpolators for sampling rate reduction and sampling rate increase is presented. A set of curves and the necessary relations for optimally designing multistage decimators is also given. It is shown that the processes of decimation and interpolation are duals and therefore the same set of design curves applies to both problems. Further, it is shown that highly efficient implementations of narrow-band finite impulse response (FIR) filters can be obtained by cascading the processes of decimation and interpolation. Examples show that the efficiencies obtained are comparable to those of recursive elliptic filter designs.

I. Introduction

WITH the increasing number of applications of digital techniques in signal processing, the need for efficient translating between various sampling frequencies has become increasingly apparent. An example of such an application is the conversion of digital signal code formats, i.e., conversion from delta modulation to pulse-code modulation, which inherently operate at different sampling rates [1]. Another example is the transmission of speech via analysis–synthesis techniques (e.g., vocoders) where a sampling rate reduction is required for efficient transmission, and a sampling rate increase is required for the regeneration of the speech [2]. Yet another application is in narrow-band filtering where an efficient implementation of a digital filter can be realized using a sampling rate reduction followed by a sampling rate increase [3]-[6].

The processes of sampling rate reduction (often called decimation) and sampling rate increase (or interpolation) have been examined from several points of view. Crooke and Craig [7] have shown that for band-limiting applications computational efficiencies can be gained by sampling rate reduction. Nelson *et al.* [8] have found similar advantages using sampling rate reduction in conjunction with octave band filters. Schafer and Rabiner [4] have shown that sampling rate increase and sampling rate reduction are basically interpolation processes and can be efficiently implemented using finite impulse response (FIR) digital filters. Furthermore, they show that sampling rate conversion between any rational ratio of sampling frequencies can be efficiently implemented by a two-stage process consisting of an integer sampling rate increase followed by an integer sampling rate decrease. Bellanger *et al.* [5] found that efficient implementations of low-pass FIR filters could be obtained by a process of first reducing the sampling rate, filtering, and then increasing the sampling rate back to the original frequency. They proposed a multistage procedure for sampling rate reduction by alternately filtering with half-band filters and reducing the sampling rate by factors of 2 until a desired power of 2 sampling rate reduction was achieved. They then used a reverse process for interpolating back to the original sampling frequency. Shively [9] has also suggested a multistage process for integer sampling rate reductions for which the sampling rate reduction at each stage can be greater than two. He proposed a procedure for optimally choosing the reduction ratios at each stage such that the amount of computation in the overall sampling rate reduction process was minimized.

In this paper we present a general theory of the processes of sampling rate reduction and sampling rate increase. Hereafter, we will generally refer to these processes as decimation and interpolation, respectively. In Section II we review some of the basic principles of decimation and interpolation established by Schafer and Rabiner [4]. In Section III we use these results to extend the approach of Shively and present design curves and formulas for the optimum implementation of decimators with noninteger decimation ratios. In Section IV we address the problem of interpolation and show that there is a mathematical duality as well as a conceptual one between decimation and interpolation. This duality then enables a user to apply the design curves and formulas developed in Section III to the design of optimum interpolators. In Section V we discuss the design of low-pass narrow-band FIR filters which are implemented in terms of an optimal decimator in cascade with an optimal interpolator. We then show that this form of implementation corresponds to an optimal FIR filter implementation. For narrow-band filters very significant gains in efficiency can be derived by using this form of implementation over that of a direct form implementation. Finally, in Section VI some additional considerations in the design of optimum decimators and interpolators are given.

II. Preliminary Considerations of the Decimation and Interpolation Processes

Before proceeding to the design of optimum decimators and interpolators it will be convenient to review, in this section, some of the fundamental concepts of decimation and interpolation in the context of digital signal processing as discussed in [4].

The process of decimating a signal $x(n)$ by an integer ratio M is depicted in Fig. 1. The original sampling rate is denoted as f_r and the final rate is then f_r/M. To avoid aliasing at the lower sampling rate, f_r/M, it is necessary to filter the original

Manuscript received January 24, 1975; revised May 16, 1975.
The authors are with Bell Laboratories, Murray Hill, N.J. 07974.

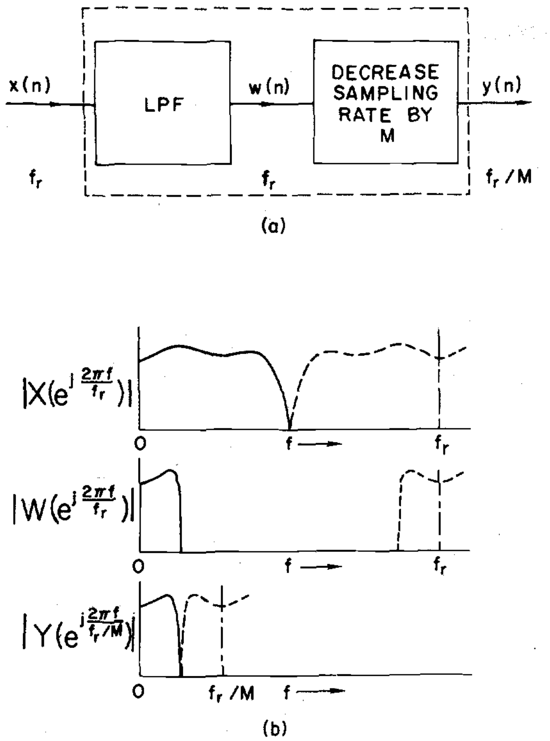
x(n)
LPF
w(n)
DECREASE SAMPLING RATE BY M
y(n)
f_r
f_r
f_r/M
(a)
$|X(e^{j\frac{2\pi f}{f_r}})|$
0
f ⟶
f_r
$|W(e^{j\frac{2\pi f}{f_r}})|$
0
f ⟶
f_r
$|Y(e^{j\frac{2\pi f}{f_r/M}})|$
0
f_r/M
f ⟶
(b)

Fig. 1. (a) Illustration of the decimation process and (b) frequency response interpretation.

signal $x(n)$ with a low-pass filter giving the signal $w(n)$. The sampling rate reduction is then achieved by forming a new sequence $y(n)$ by extracting every Mth sample of $w(n)$. A block diagram of these operations is given in Fig. 1(a). Fig. 1(b) shows typical Fourier transforms of the signals $x(n)$, $w(n)$, and $y(n)$.

The process of interpolating a signal $x(n)$ by an integer ratio L is, similarly, depicted in Fig. 2. In this case the sampling rate of a signal $x(n)$ is increased by the factor L by inserting L-1 zero-valued samples between each sample of $x(n)$. This creates a signal $w(n)$ (with sampling rate Lf_r) whose frequency components are periodic with period equal to the original sampling frequency f_r as shown in Fig. 2(b). To eliminate these periodic components and retain only the baseband frequencies it is necessary to filter the signal $w(n)$ with an appropriate low-pass filter.[1] The resulting signal $y(n)$ with sampling rate Lf_r is then the desired interpolated signal. At the bottom of Fig. 2(b) the Fourier transform of the signal $y(n)$ is shown.

Thus far, we have only considered sampling rate increases or decreases by integer ratios. If sampling rate conversions between rational integer ratios of sampling frequencies are desired this can be achieved by a combination of the two above processes. That is, if we desire a sampling rate conversion by a ratio L/M (where L and M are integers) we can achieve this by first interpolating by L and then decimating by M as shown in Fig. 3(a). It is important to recognize that the interpolation process *must* precede the decimation process to retain the desired frequency band in the final output (except in the case when the original signal is sufficiently band limited). If we

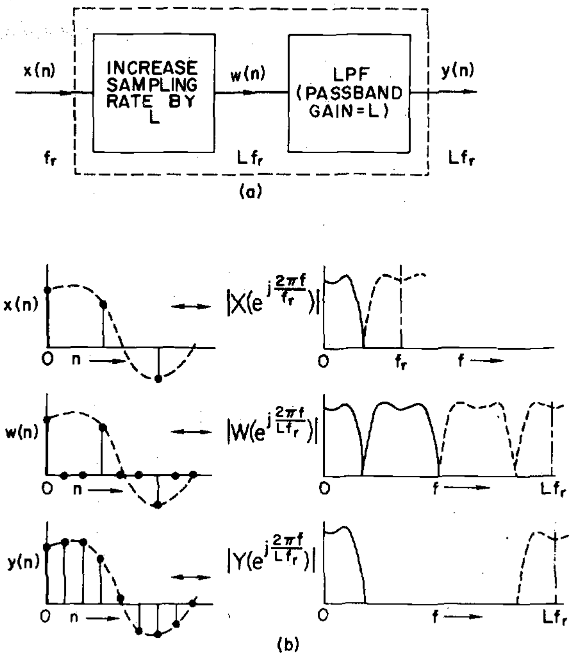
x(n)
INCREASE SAMPLING RATE BY L
w(n)
LPF (PASSBAND GAIN=L)
y(n)
f_r
Lf_r
Lf_r
(a)
x(n)
0 n ⟶
$|X(e^{j\frac{2\pi f}{f_r}})|$
0
f_r
f ⟶
w(n)
0 n ⟶
$|W(e^{j\frac{2\pi f}{Lf_r}})|$
0
f ⟶
Lf_r
y(n)
0 n ⟶
$|Y(e^{j\frac{2\pi f}{Lf_r}})|$
0
f ⟶
Lf_r
(b)

Fig. 2. (a) Illustration of the interpolation process and (b) frequency response and waveform interpretation.

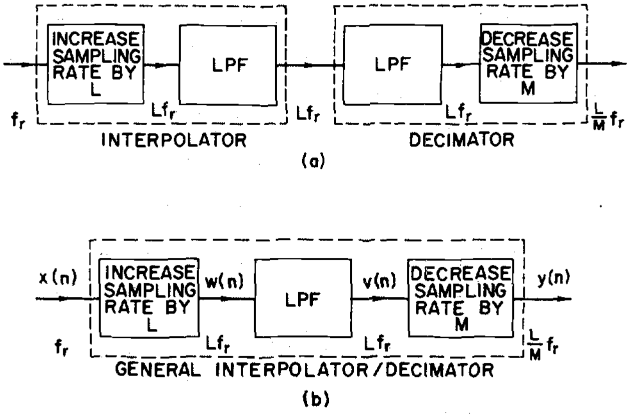
INCREASE SAMPLING RATE BY L
LPF
LPF
DECREASE SAMPLING RATE BY M
f_r
Lf_r
Lf_r
Lf_r
$\frac{L}{M}f_r$
INTERPOLATOR
DECIMATOR
(a)
x(n)
INCREASE SAMPLING RATE BY L
w(n)
LPF
v(n)
DECREASE SAMPLING RATE BY M
y(n)
f_r
Lf_r
Lf_r
$\frac{L}{M}f_r$
GENERAL INTERPOLATOR/DECIMATOR
(b)

Fig. 3. (a) Cascade of an interpolator and a decimator for achieving sampling rate changes by noninteger factors. (b) More efficient implementation of this process.

examine this process we recognize that the two low-pass filters are operating in cascade and at the same sampling rate. Thus a more efficient implementation of the overall process can be achieved if the filters are combined into one composite low-pass filter [4] as shown in Fig. 3(b). The system of Fig. 3(b) corresponds to a general interpolation or decimation module depending upon whether L is greater or less than M. If $L = 1$, this general module becomes the integer decimator of Fig. 1 and, if $M = 1$, it becomes the integer interpolator of Fig. 2. The design of the low-pass filter (i.e., specification of cutoff frequencies and ripples) in this general module for sampling rate change will be explained in Sections III and IV.

An extremely important consideration in the implementation of the sampling rate changing system of Fig. 3(b) is the choice of the type of low-pass filter [4]. For this type of system, a significant savings in computation can be obtained by using an FIR filter in a standard direct form implementation. For this case, the filter output $v(n)$ is computed from the in-

[1] To retain the same signal energy at the high sampling rate the passband gain of this filter must be equal to L [4].

put $w(n)$ by the relation

$$v(n) = \sum_{m=0}^{N-1} h(m)w(n-m) \tag{1}$$

where $h(m)$, $(m = 0, 1, \cdots, N-1)$ are the filter coefficients and N is the duration of the unit sample response of the filter. From (1) it is seen that the computation of an output point depends only upon past and present values of $w(n)$ and not upon past values of any internal filter variables. Thus the filter computations need only be made once for every Mth output point. Furthermore, it is known that $w(n)$ is nonzero only for every Lth input point. Thus only one multiplication and addition must be performed for every Lth input point. Therefore the effective number of multiplications and additions is $N/(LM)$ per output sample,[2] instead of N as predicted by a simple application of (1). For this and other reasons to be discussed later, we will assume for the remainder of this paper that the low-pass filter is a linear phase FIR design used in a direct form implementation.

III. Optimum Design of Multistage Decimators for Sampling Rate Reduction

In the previous section a general one-stage technique for changing the sampling rate of a signal by the factor L/M was discussed. For large changes in sampling rate, however, it is generally more efficient to reduce the sampling rate with a series of decimation stages rather than making the entire rate reduction with one stage. In this way the sampling rate is reduced gradually resulting in much less severe filtering requirements on the low-pass filters at each stage. Bellanger *et al.* [5] and Nelson *et al.* [8] also have implemented sampling rate reductions using several decimation stages; however they restricted their results by only using factors of 2 at each stage. Shively [9] considered a more general approach with integer decimation for a two-stage design similar to that of Fig. 1(b). He also suggested a procedure for optimizing the two-stage design by properly choosing the decimation ratios of each of the stages. In this section we extend and generalize the work of Shively using the more general decimation stage shown in Fig. 3(b). Design curves and formulas are presented for implementing optimum decimators for a wide range of parameters and conditions.

The basic multistage process for sampling rate reduction with K stages is illustrated in Fig. 4(a) and a frequency domain interpretation of this process is given in Fig. 4(b). The initial sampling rate is f_{r0} and the final sampling rate is f_{rK} with intermediate sampling frequencies designated as $f_{r1}, f_{r2}, \cdots, f_{r(K-1)}$. The sampling rate reduction achieved by each stage of the decimation process is denoted as $D_i, i = 1, 2, \cdots, K$ and therefore the intermediate sampling frequencies are given by

$$f_{ri} = \frac{f_{r(i-1)}}{D_i}, \quad i = 1, 2, \cdots, K. \tag{2}$$

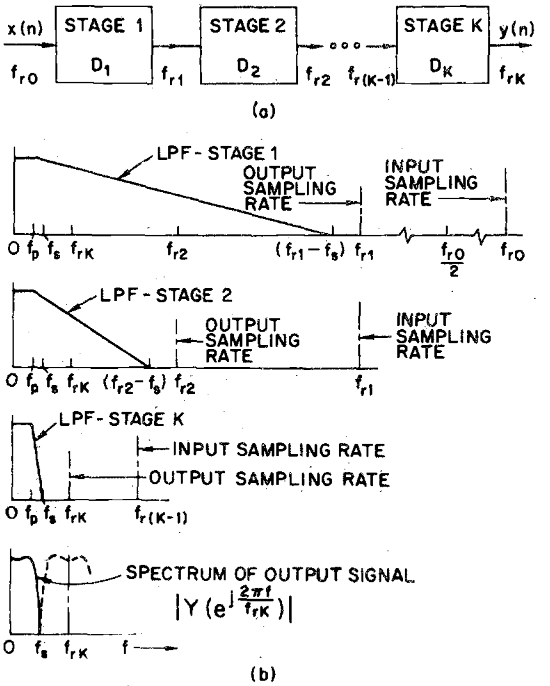

Fig. 4. (a) Illustration of a K-stage decimator and (b) frequency response interpretation of this process.

The overall sampling rate reduction achieved by this process is denoted as D and is given as

$$\begin{aligned} D &= f_{r0}/f_{rK} \\ &= \prod_{i=1}^{K} D_i. \end{aligned} \tag{3}$$

In Fig. 4(b) $|X[\exp j(2\pi f/f_{r0})]|$ represents the magnitude of the spectrum of $x(n)$ and $|Y[\exp j(2\pi f/f_{rK})]|$ corresponds to the magnitude of the spectrum of $y(n)$. From the sampling theorem, we recognize that the highest frequency in $y(n)$ is f_s which is

$$f_s \leqslant \frac{f_{rK}}{2}. \tag{4}$$

Because of practical considerations in the design of the low-pass filters used in the decimation process, this bandwidth can never be fully realized and the usable portion of the baseband will always be somewhat less than f_s. We denote this bandwidth as being from zero to f_p corresponding to the band over which the magnitude response of the composite of the low-pass filters remains flat within specified tolerance limits of $1 \pm \delta_p$. Also because of practical considerations a maximum tolerance must be allowed for the magnitude response of the low-pass filters in the stopband and this will be denoted as δ_s.

We can now focus on the design requirements for each stage. We will permit the most general form for each stage as discussed in the previous section [see Fig. 3(b)]. This form for stage i is illustrated in Fig. 5(a). L_i corresponds to the sampling rate increase and M_i corresponds to the sampling rate decrease for stage i. As there is a net sampling rate reduction this implies, $M_i > L_i$. Then

$$f_{ri} = \frac{L_i}{M_i} f_{r(i-1)} \tag{5}$$

[2] This does not imply that greater efficiencies can be obtained by simultaneously increasing L and M as both N and the sampling rate in (1) are proportional to L.

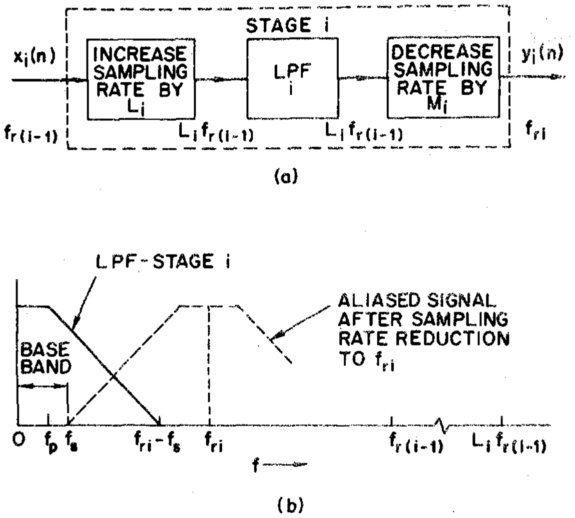

Fig. 5. (a) Stage i of a K-stage decimator. (b) Frequency response representation of the low-pass filter requirements for stage i.

and from (2) the net decimation ratio D_i for stage i is given as

$$D_i = \frac{M_i}{L_i}. \tag{6}$$

The requirements on the low-pass filter for stage i can be determined with the aid of Fig. 5(b). The passband requirement must be such that it is flat from zero to f_p in order to avoid distorting the baseband response. As it is desired that the overall passband ripple for the composite of K stages be maintained within $1 \pm \delta_p$, it is necessary to require more severe constraints on the individual filters in the cascade. A convenient choice which will satisfy this requirement is to specify the passband ripple constraints for each stage i to be within $1 \pm \delta_{pi}$ where $\delta_{pi} = \delta_p/K$. 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.[3] Finally, to assure that no high frequency components are aliased into the baseband at stage i, the stopband cutoff frequency of the low-pass filter i can be chosen to be $f_{ri} - f_s$. At the final stage, K, it can be seen that this cutoff frequency is then equal to f_s as desired since $f_{rK} = 2 f_s$.

Thus far, we have established the basic framework for a multistage decimator and have specified the requirements which must be met at each stage. Given that the requirements $f_{r0}, f_s, f_p, \delta_p$, and δ_s are known, the next consideration is that of optimizing the design for maximum efficiency. The flexible parameters that we have at our disposal are the decimation ratios D_i, $i = 1, 2, \cdots, K$, as well as K. The function to be minimized is the overall amount of required computation. A meaningful measure of this is the total number of multiplies and adds (MADS) which must be performed per second. If R_T is denoted as this value and R_i is the number of MADS/s for stage i, then

$$R_T = \sum_{i=1}^{K} R_i. \tag{7}$$

Let N_i denote the number of samples in the unit sample response of the direct form FIR filter in stage i. Then R_i can be expressed as the product of N_i times the sampling rate of the filter, $L_i f_{r(i-1)}$, divided by the savings factor $L_i M_i$ discussed in Section II. Therefore

$$\begin{aligned} R_i &= \frac{N_i L_i f_{r(i-1)}}{L_i M_i} \\ &= \frac{N_i f_{r(i-1)}}{M_i}. \end{aligned} \tag{8}$$

The duration of the unit sample response, N_i, can, in turn, be derived from the required filter specifications of the low-pass filter in stage i.

Design procedures and relations for designing optimal FIR filters of the above type have been well established. Given the desired ripple specifications and cutoff frequencies an approximate relation for the necessary unit sample response duration N is given as[4] [10], [11]

$$N \cong \frac{D_\infty(\delta_1, \delta_2)}{\Delta F} - f(\delta_1, \delta_2)\, \Delta F + 1 \tag{9}$$

where

ΔF width of the transition band normalized to the sampling frequency

$$\begin{aligned} D_\infty(\delta_1, \delta_2) = {} & [5.309 \times 10^{-3} (\log_{10} \delta_1)^2 \\ & + 7.114 \times 10^{-2} (\log_{10} \delta_1) \\ & - 0.4761] \log_{10} \delta_2 - [2.66 \times 10^{-3} (\log_{10} \delta_1)^2 \\ & + 0.5941 (\log_{10} \delta_1) + 0.4278] \end{aligned} \tag{10}$$

$$f(\delta_1, \delta_2) = 0.51244 \log_{10} (\delta_1/\delta_2) + 11.01217$$

δ_1 tolerance in the magnitude response in the passband

δ_2 tolerance in the magnitude response in the stopband.

In the particular application that we are considering here most of the stages will have relatively narrow-band filters, especially in cases where decimation ratios D are greater than 10 or so. Generally, ΔF in (9) will be relatively small and the second two terms on the right side of (9) will be relatively insignificant compared to that of the first term. With this in mind we can argue that a further approximation of (9) to the form

$$N \cong \frac{D_\infty(\delta_1, \delta_2)}{\Delta F} \tag{11}$$

is justified for our purposes. Also, it greatly simplifies the development of the following optimization procedure. Further justification for this step will be presented later.

From the preceding discussion of the required filter require-

[3] Since the aliasing generally does not add coherently, δ_s instead of δ_s/K is used in each stage for the stopband specification. The use of δ_s/K instead of δ_s does not change the results sufficiently from those given here because of the relative insensitivity of the design formulas to the deltas.

[4] $D_\infty(\delta_1, \delta_2)$ should *not* be confused with the decimation ratios D_i.

ments we can now determine the length of the unit sample response N_i for stage i. Letting

$$\delta_1 = \delta_p/K$$

$$\delta_2 = \delta_s$$

$$\Delta F = \frac{(f_{ri} - f_s) - f_p}{L_i f_{r(i-1)}}$$

and substituting into (11), we get

$$N_i \cong \frac{D_\infty(\delta_p/K, \delta_s)\, L_i f_{r(i-1)}}{f_{ri} - f_s - f_p}. \tag{12}$$

From (2), (6), (8), and (12) the computation rate R_i in MADS/s for stage i can be expressed in the form

$$R_i \cong D_\infty\left(\frac{\delta_p}{K}, \delta_s\right) \frac{D_i f_{ri}^2}{f_{ri} - f_s - f_p}. \tag{13}$$

By applying (7) the total computation for stages 1-K is given as

$$R_T \cong D_\infty\left(\frac{\delta_p}{K}, \delta_s\right) \sum_{i=1}^{K} \frac{D_i f_{ri}^2}{f_{ri} - f_s - f_p} \text{ (MADS/s)} \tag{14}$$

and with the aid of (2) and (3) this can be expressed in the form

$$R_T \cong D_\infty\left(\frac{\delta_p}{K}, \delta_s\right) f_{ro} \sum_{i=1}^{K} \frac{D_i}{\left(\prod_{j=1}^{i} D_j\right)\left(1 - \frac{f_s + f_p}{f_{ro}} \prod_{j=1}^{i} D_j\right)}. \tag{15}$$

This is almost in the form that we desire for a formal optimization procedure. That is, R_T is expressed only in terms of known parameters and the variables $D_1, D_2, \cdots, D_K$ over which it is to be minimized. It is important to recognize however that only $K - 1$ of these variables are independent. That is, we can express D_K in terms of the known parameter D and the variables $D_1, D_2, \cdots, D_{K-1}$ in the form

$$D_K = \frac{D}{\prod_{i=1}^{K-1} D_i}. \tag{16}$$

Therefore D_K must be eliminated from (15) before attempting to minimize R_T. Otherwise the optimization procedure will fail.

At this point it is also convenient for later notational purposes to define a new parameter

$$\Delta f = \frac{f_s - f_p}{f_s}. \tag{17}$$

This is simply an alternative to expressing both f_s and f_p. It has the interpretation that it is a relative measure of the transition band between f_s and f_p. For example if Δf is 0.25 then the passband, 0 to f_p, occupies three-quarters of the new baseband.

Upon substitution of (16) and (17) into (15) we get the final objective function

$$R_T \cong D_\infty\left(\frac{\delta_p}{K}, \delta_s\right) f_{ro}\, S \text{ (MADS/s)} \tag{18a}$$

where

$$S = S(D, \Delta f, K; D_1, D_2, \cdots, D_{K-1})$$
$$= \frac{2}{\left(\Delta f \prod_{j=1}^{K-1} D_j\right)} + \sum_{i=1}^{K-1} \frac{D_i}{\left(\prod_{j=1}^{i} D_j\right)\left(1 - \left(\frac{2 - \Delta f}{2D}\right) \prod_{j=1}^{i} D_j\right)}. \tag{18b}$$

This is the final form that we desire.

We can now minimize R_T by selecting the number of decimation stages K and then minimizing R_T in terms of the decimation ratios which can be assumed to be continuous variables. By comparing values of R_T for these various values of K we can then optimize the number of stages. Typically the optimum number of stages will be somewhere between 1 and 4.

For $K = 1$ there is no flexibility in choosing decimation ratios ($D_1 = D$). For $K = 2$, R_T is a function of one variable, D_1. To minimize R_T we can take its derivative with respect to D_1 and set it to zero. This yields a quadratic function in D_1. It leads to two solutions for D_1, only one of which is valid, and can be given as

$$D_{1_{\text{opt}}} = \frac{2D(1 - \sqrt{D\Delta f/(2 - \Delta f)})}{2 - \Delta f(D + 1)}, \quad K = 2 \tag{19a}$$

and

$$D_{2_{\text{opt}}} = D/D_{1_{\text{opt}}}, \quad K = 2. \tag{19b}$$

For larger values of K the above analytic procedure is not feasible and we must resort to more powerful computer-aided optimization techniques. For our purposes we found that the Hooke and Jeeves optimization routine [12], [13] worked well and did not require the evaluation of derivatives.

In the above procedure for optimizing the multistage decimator several interesting observations can be made. From (18a) we can see that R_T is a product of three factors, the initial frequency f_{ro}, $D_\infty(\delta_p/K, \delta_s)$ which is a function of δ_p, δ_s, and K, and S which is a function of Δf, D, K, and the decimation ratios. This factorization makes the problem much simpler to deal with and to grasp intuitively. For example, we can observe the effects that δ_p and δ_s have on R_T by examining the behavior of $D_\infty(\delta_p/K, \delta_s)$. In Table I this function is tabulated for some typical values of δ_p and δ_s and for $K = 1, 2, \cdots, 6$. From this table we can also observe the effect of K on $D_\infty(\delta_p/K, \delta_s)$. For example, in going from a one-stage design ($K = 1$) to a two-stage design ($K = 2$), $D_\infty(\delta_p/K, \delta_s)$ increases typically by a factor of approximately 10 percent. This is a reflection of the increased amount of computation that is required due to the fact that the passband ripples of the two cascaded filters in the two-stage design must be made smaller in order to retain a composite ripple of δ_p. As we will see shortly, this variation of $D_\infty(\delta_p/K, \delta_s)$ with respect to K is relatively insignificant compared to the variation of S with respect to K. In other words we conclude that the choice of the optimum number of stages is not strongly affected by the choice of δ_p and δ_s, but rather, on the choice of Δf and D.

Perhaps the more interesting function in (18a) is S. For a given number of stages R_T is minimized by minimizing S. To

TABLE I

TABULATION OF $D_\infty\left(\frac{\delta_p}{K}, \delta_s\right)$

δ_p	δ_s	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$
0.100	0.0100	1.25	1.46	1.58	1.67	1.74	1.79
0.100	0.0050	1.41	1.63	1.75	1.84	1.91	1.97
0.100	0.0010	1.80	2.02	2.15	2.25	2.32	2.38
0.100	0.0005	1.95	2.19	2.32	2.42	2.49	2.55
0.100	0.0001	2.33	2.58	2.72	2.82	2.90	2.96
0.050	0.0100	1.46	1.67	1.79	1.88	1.94	1.99
0.050	0.0050	1.63	1.84	1.97	2.06	2.12	2.18
0.050	0.0010	2.02	2.25	2.38	2.47	2.54	2.60
0.050	0.0005	2.19	2.42	2.55	2.65	2.72	2.78
0.050	0.0001	2.58	2.82	2.96	3.06	3.14	3.20
0.010	0.0100	1.94	2.15	2.27	2.35	2.42	2.47
0.010	0.0050	2.12	2.33	2.45	2.54	2.60	2.66
0.010	0.0010	2.54	2.76	2.89	2.98	3.04	3.10
0.010	0.0005	2.72	2.94	3.07	3.16	3.23	3.29
0.010	0.0001	3.14	3.37	3.50	3.60	3.67	3.73
0.005	0.0100	2.15	2.35	2.47	2.55	2.61	2.66
0.005	0.0050	2.33	2.54	2.66	2.74	2.81	2.86
0.005	0.0010	2.76	2.98	3.10	3.19	3.26	3.31
0.005	0.0005	2.94	3.16	3.29	3.38	3.45	3.50
0.005	0.0001	3.37	3.60	3.73	3.83	3.90	3.95
0.001	0.0100	2.61	2.81	2.92	3.00	3.07	3.11
0.001	0.0050	2.81	3.01	3.12	3.20	3.27	3.31
0.001	0.0010	3.25	3.46	3.58	3.67	3.73	3.78
0.001	0.0005	3.45	3.66	3.78	3.87	3.93	3.98
0.001	0.0001	3.90	4.12	4.24	4.33	4.40	4.45

show this behavior of S we chose six values of Δf, distributed from 0.01 to 0.5, and values of D ranging from 1 to 5000. We then minimized S for these examples with $K = 1, 2, 3$, and 4 using the Hooke and Jeeves algorithm when necessary. The resulting optimum values of S are plotted in Fig. 6 and the optimum decimation ratios which minimize S are plotted in Fig. 7. Six sets of graphs are given in each figure and they correspond to the six values of Δf. Each graph in Fig. 6 contains four curves for the appropriate values of K. In Fig. 7 curves for all values of K and i are given. Also indicated in the plots of Fig. 6 are the approximate ranges of D over which one, two, three, or four stages appear to be optimum based on the behavior of S. That is, one stage appears to be optimal over region I, two stages appear to be optimal over region II, etc. The dotted curves in Fig. 6 will be explained in Section V.

Several important conclusions can be drawn from the curves of Fig. 6. Generally, the largest gains in efficiency are possible in going from a one-stage to a two-stage design. This confirms a conjecture previously made by Shively [9]. Smaller gains are possible in going from a two-stage to a three-stage design and it is questionable as to whether any further improvement is possible with four stages.

Equations (10), (17)-(19), Table I, and Figs. 6 and 7 can be used as a set of design relations and curves for a wide range of designs of optimal decimators. To apply them we first calculate R_T in (18a) using (10) or Table I to determine $D_\infty(\delta_p/K, \delta_s)$ and the curves in Fig. 6 to determine the minimum S. The optimum number of stages can be determined by calculating R_T for $K = 1, 2, 3$, and 4 and choosing the one which corresponds to the minimum. When the optimum number of stages is known, the optimum values for the decimation ratios for each stage can be determined from the curves in Fig. 7 or from (19).

Generally, it has been observed that the range over which the choice of decimation ratios can be varied without appreciably increasing R_T is relatively large. This property further justifies our use of the approximation in (11) and is very desirable because it allows a great deal of flexibility in the choice of decimation ratios in an actual design without sacrificing efficiency. In fact it is often desired in practice to choose nearest integer values for the decimation ratios when possible. This allows us to take advantage of the symmetry of the impulse response in the implementation of the low-pass filters and achieve an additional factor of approximately 2 in savings in the required number of multiplications. This savings cannot, in general, be realized in stages where noninteger decimation ratios are implemented. Another reason for perhaps deviating from an optimal design is when the order N_i of a filter in one or more of the stages is excessively large. In this situation we may want to change the decimation ratios in such a way that it will relax the filter requirements of these particular stages at the expense of increasing the filter orders in other stages. It may also be desirable in such cases to increase the number of decimation stages in the design in order to reduce the orders of the filters in all of the stages. Because of these types of practical tradeoffs, it is suggested that the above design curves and formulas be applied rather flexibly and used only as a guide in obtaining a starting point for an optimal design. Practical considerations should then be applied to determine the final choice of the design.

An Example

To illustrate the above design approach we will choose to design a decimator with the following specifications.

$\delta_p = 0.05$
$\delta_s = 0.005$
$\Delta f = 0.1$
$D = 20$
$f_{r0} = 10$ kHz.

From Table I we can obtain values for $D_\infty(\delta_p/K, \delta_s)$ and from Fig. 6(c) we can obtain minimized values for S for $K = 1, 2, 3$, and 4. These results are compiled in Table II along with computed values of R_T using (18a). From this table we see that the minimum value for R_T occurs when $K = 3$. By referring to Fig. 7(c) we can determine the optimum decimation ratios for this three-stage design to be

$$D_1 \cong 5.9, D_2 \cong 2.4, D_3 \cong 1.4.$$

These values of decimation ratios appear to be somewhat impractical to use in a realistic design. A more convenient choice might be to choose a nearest design which achieves the overall decimation rate D using integer decimation ratios. One choice which is close to this optimum design might be $D_1 = 5, D_2 = 2, D_3 = 2$. By applying these values into (18a) and (18b) we can obtain an estimate of $R_T \cong 72730$ MAD/s. Because we have integer decimation ratios, we can take advantage of the sym-

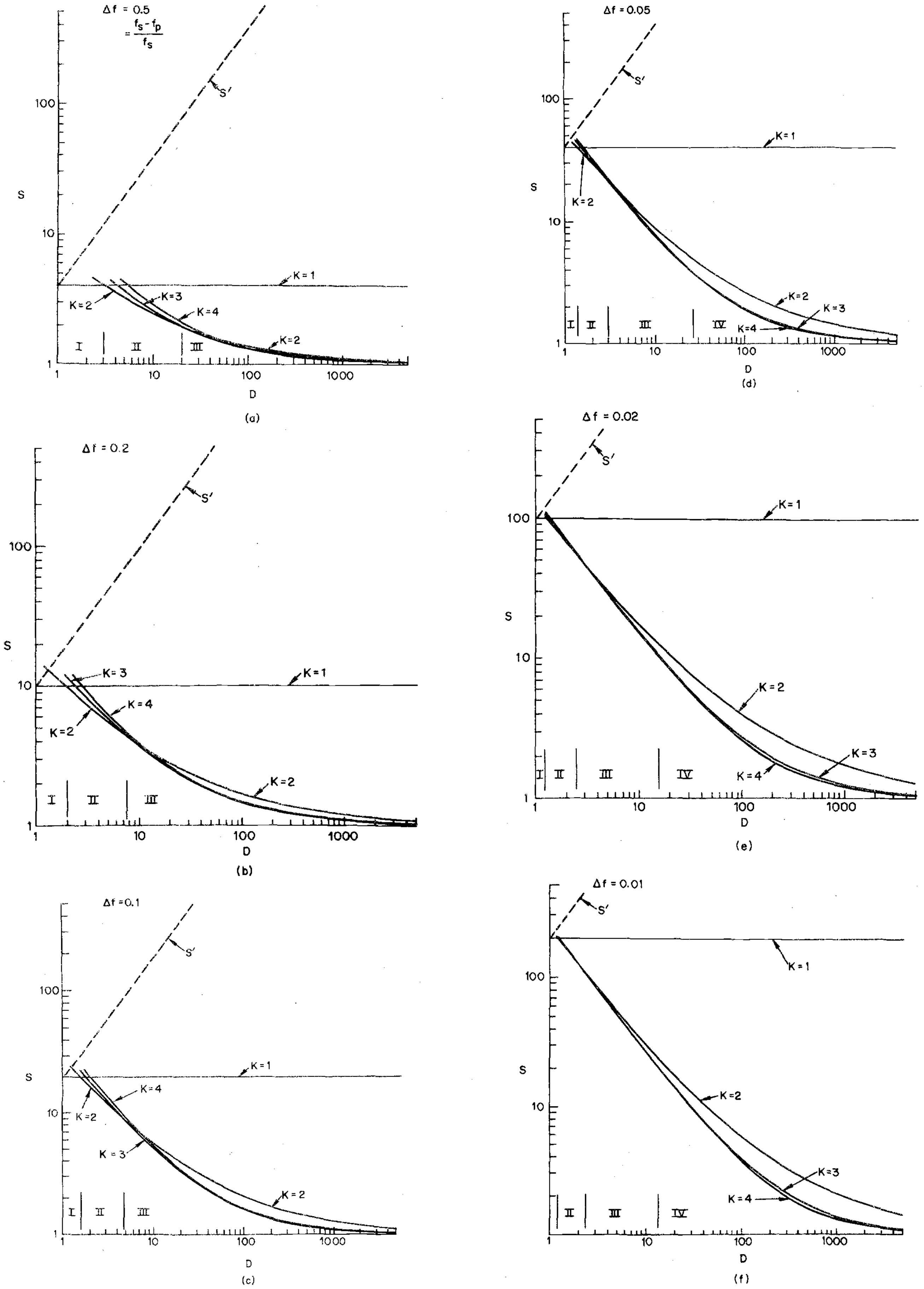

Fig. 6. Plots of minimized values of S as a function of K, D, and Δf where Δf is 0.5, 0.2, 0.1, 0.05, 0.02, and 0.01 for plots (a)-(f), respectively.

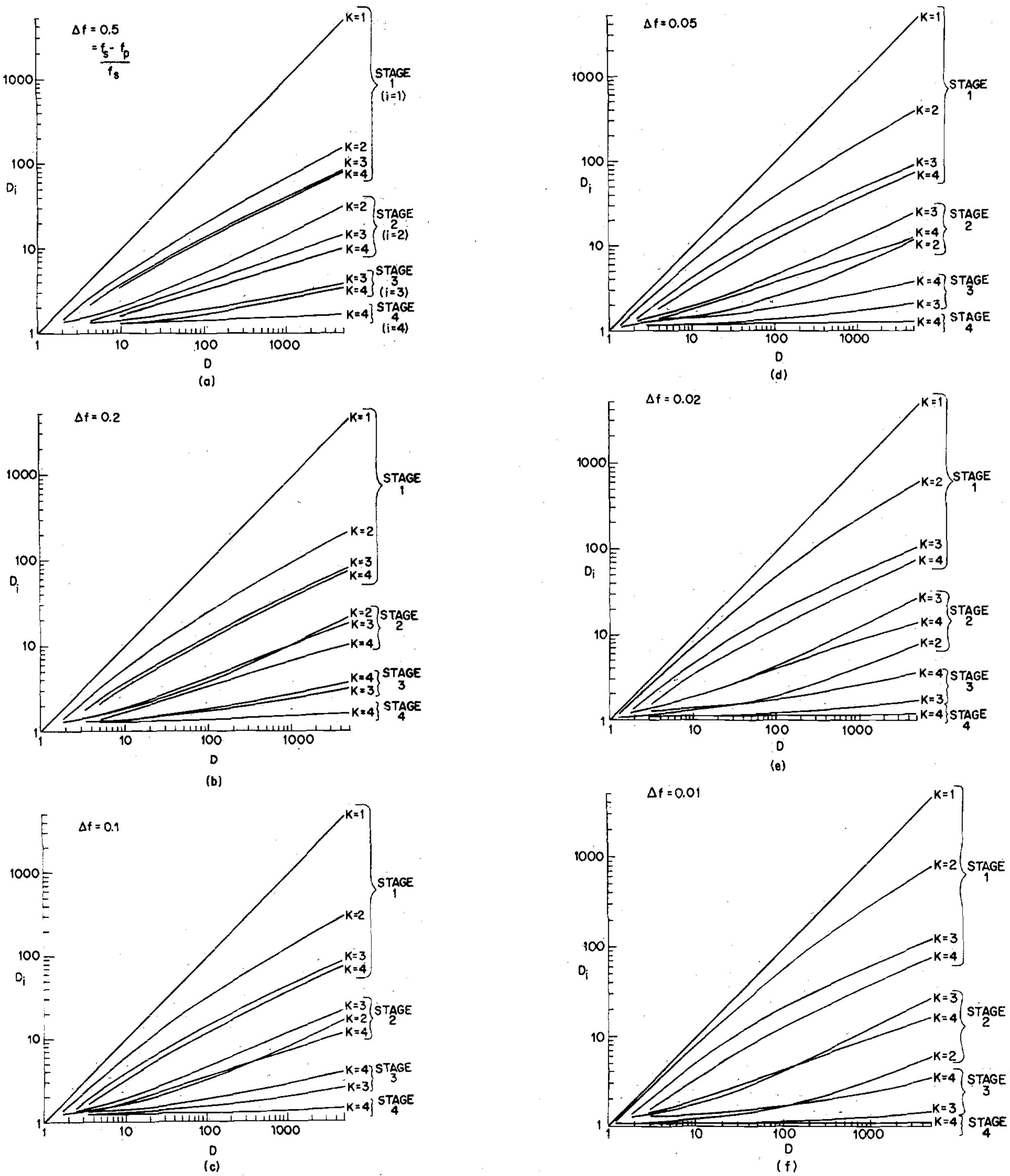

Fig. 7. Plots of optimum decimation ratios as a function of K, D, and Δf where Δf is 0.5, 0.2, 0.1, 0.05, 0.02 and 0.01 for plots (a)-(f), respectively.

metry of the impulse responses of the filters and require only half of the number of multiplications, or approximately 36 365 multiplications/s.

An alternative approach might be to consider a two-stage design as opposed to a three-stage design. In this case the optimum decimation ratios from Fig. 7(c) are $D_1 \cong 10.3$ and $D_2 \cong 1.95$. Practical values for these ratios would be $D_1 = 10$ and $D_2 = 2$. From (18a) and (18b) we can determine R_T for this design is to be $R_T \cong 71\,850$. Again we can take advantage of symmetry and implement the design with approximately 35 925 multiplications/s.

Therefore we have two alternatives which appear to be promising for the above design, both which require approximately the same amount of computation. Another consideration in comparing these designs might be the required filter order for each stage. By applying (2) and (17) into (12) an expression for this filter order can be given as

$$N_i \approx \frac{D_\infty(\delta_p/K, \delta_s)\, L_i D_i}{1 - (2 - \Delta f/2D) \prod_{j=1}^{i} D_j}. \tag{20}$$

TABLE II

K	$D_\infty\left(\frac{\delta_p}{K}, \delta_s\right)$	S	R_T
1	1.63	20.0	326 000
2	1.84	3.9	71 760
3	1.97	3.3	65 010
4	2.06	3.3	67 980

For the three-stage design these orders can be calculated to be $N_1 \approx 12.9, N_2 \approx 7.5$, and $N_3 \approx 78.8$ and for the two-stage design they are $N_1 \approx 35$ and $N_2 \approx 73.6$. The total amount of coefficient storage for the two designs, using symmetry, is therefore ≈50 for the three-stage design and ≈54 for the two-stage design. Therefore they are also comparable in this respect for this example.

A final consideration in choosing from the two alternatives is the number of stages. As both designs are roughly equivalent we would probably be inclined to choose the two-stage design as it would be somewhat simpler to implement. This example illustrates a typical method of applying the curves and relations in this section to the design of an optimal decimator.

IV. Optimal Design of Interpolators for Sampling Rate Increase

Interpolation is essentially the inverse or dual of decimation. That is, we start from a low sampling rate and convert to a high sampling rate. This duality is obvious conceptually and, as will be seen shortly, it can also be formulated in more rigorous terms. Since we have already shown that the decimation process can sometimes be implemented more efficiently as a multistage process, duality then implies that the interpolation process can also be implemented more efficiently as a multistage process.

With the above concepts in mind the multistage interpolation process can be formulated as shown in Fig. 8. Because the interpolation and decimation processes are dual, it is convenient to choose a notation that emphasizes this duality. Thus in Fig. 8, the notation used is the reverse of the notation used for decimation in Fig. 4. The input sampling rate, the low rate, is designated as f_{rK} and the high sampling rate at the output is designated as f_{r0}. The stages and intermediate frequencies are similarly labeled in reverse order. It is also convenient to assume that each of the stages of interpolation are of the general form shown in Fig. 3(b). However, in keeping with our notation of the dual process we will assign M_i to be the integer ratio by which the sampling rate is increased and L_i to be the integer ratio by which the sampling rate is decreased in stage i, as shown in Fig. 9. The requirement for the structure of Fig. 9 to be an interpolator is $M_i > L_i$. If D_i denotes the net ratio of sampling rate increase for stage i, then

$$D_i = \frac{M_i}{L_i} \qquad i = 1, 2, \cdots, K \tag{21}$$

and

$$f_{r(i-1)} = D_i f_{ri} \qquad i = 1, 2, \cdots, K. \tag{22}$$

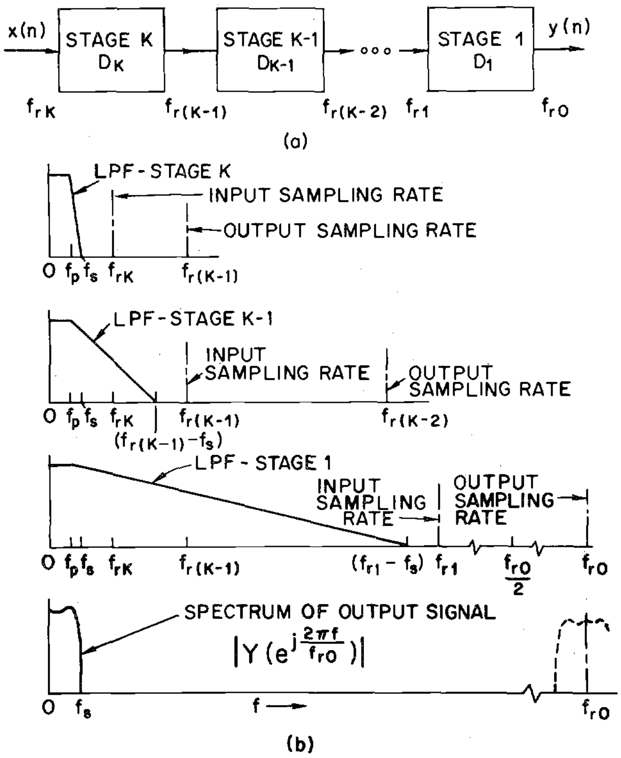

Fig. 8. (a) Illustration of a K-stage interpolator and (b) its frequency response interpretation.

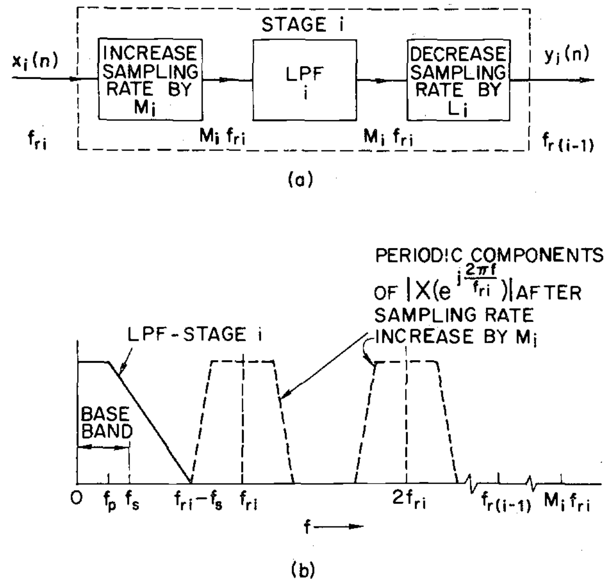
STAGE i
x_i(n)
INCREASE SAMPLING RATE BY M_i
LPF i
DECREASE SAMPLING RATE BY L_i
y_i(n)
f_{ri}
M_i f_{ri}
M_i f_{ri}
f_{r(i-1)}
(a)
PERIODIC COMPONENTS OF |X(e^{j2πf/f_{ri}})| AFTER SAMPLING RATE INCREASE BY M_i
LPF-STAGE i
BASE BAND
0 f_p f_s f_{ri}-f_s f_{ri} 2f_{ri} f_{r(i-1)} M_i f_{ri}
f ⟶
(b)

Fig. 9. (a) Stage i of a K-stage interpolator. (b) Frequency response representation of the low-pass filter requirements for stage i.

Finally the overall net sampling rate increase is given as

$$D = \prod_{i=1}^{K} D_i. \tag{23}$$

The frequency interpretation of this multistage interpolation process is depicted in Fig. 8(b). The initial spectrum $|X[\exp j(2\pi f/f_{r0})]|$ covers the baseband, 0 to f_s, where $f_s = f_{rK}/2$. We again assume for practical considerations that the useable portion of this baseband extends from 0 to f_p. The low-pass filters in the interpolation process eliminate the periodicity of $|X[\exp j(2\pi f/f_{r0})]|$ over the band f_s to $f_{r0} - f_s$ and the resulting spectrum after interpolation is shown as $|Y[\exp j(2\pi f/f_{r0})]|$. Again, for practical considerations, we specify that this composite magnitude response of the low-pass filters to have a tolerance in the passband of δ_p from 0 to f_p and a tolerance in the stopband of δ_s.

The requirements for an individual filter in stage i can now be obtained with the aid of Fig. 9(b). Essentially, the filter must have a passband tolerance of $1 \pm (\delta_p/K)$ and a stopband tolerance of δ_s. The passband extends from 0 to f_p. The stopband starts at $f_{ri} - f_s$ and eliminates all of the unwanted periodic components of $|X[\exp j(2\pi f/f_{r0})]|$ above this frequency. For all stages except the first we observe that between the periodic components of $|X[\exp j(2\pi f/f_{r0})]|$, there exist frequency bands with essentially zero signal level [see Fig. 9(b)]. Strictly speaking, the filter characteristics in these regions need not be as severe as for the regions where stopband behavior is required. It is worth noting that this property could be exploited to obtain a slightly more efficient implementation of the filter [4], [14], [15]. However, for convenience, we will assume that a standard low-pass filter is used in all cases since the design relations for this class of FIR filters are more readily understood than for the more general case.

The duration N_i of the unit sample response for the low-pass filter of stage i can now be determined with the aid of (11). Using the above specifications of the cutoff frequencies, the term ΔF in (11) is given as

$$\Delta F = \frac{f_{ri} - f_s - f_p}{M_i f_{ri}} \tag{24}$$

and N_i is

$$N_i \cong \frac{D_\infty(\delta_p/K, \delta_s) M_i f_{ri}}{f_{ri} - (f_s + f_p)}. \tag{25}$$

Similarly the amount of computation R_i for stage i in MADS/s is of the form

$$\begin{aligned} R_i &= \frac{N_i M_i f_{ri}}{L_i M_i} \\ &= \frac{N_i f_{ri}}{L_i} \end{aligned} \tag{26}$$

and with the aid of (21) and (25) this becomes

$$R_i \cong \frac{D_\infty(\delta_p/K, \delta_s) D_i f_{ri}}{f_{ri} - (f_s + f_p)}. \tag{27}$$

By applying (22), R_i can be expressed in terms of the interpolation ratios D_i and the known design parameters, giving

$$R_i \cong \frac{D_\infty(\delta_p/K, \delta_s) D_i f_{r0}}{\left(\prod_{j=1}^{i} D_j\right)\left(1 - \left(\frac{f_s + f_p}{f_{r0}}\right)\prod_{j=1}^{i} D_j\right)}. \tag{28}$$

Finally, the total amount of computation required for the overall multistage interpolator is

$$R_T \cong D_\infty\left(\frac{\delta_p}{K}, \delta_s\right) f_{r0} \cdot \sum_{i=1}^{K} \frac{D_i}{\left(\prod_{j=1}^{i} D_j\right)\left(1 - \left(\frac{f_s + f_p}{f_{r0}}\right)\prod_{j=1}^{i} D_j\right)}. \tag{29}$$

If we compare (29) and (15) we observe that they are identical, i.e., the total amount of computation required to interpolate from sampling rates f_{rK} to f_{r0} is the same as that required to decimate from f_{r0} to f_{rK} under the same set of filter specifications on Δf, δ_p, and δ_s. Furthermore, if we optimize this interpolator by minimizing R_T we obtain the identical solutions for the interpolation ratios D_i as for the decimation ratios D_i of the optimum decimator. Finally, comparing the filter requirements for the low-pass filters, it is seen that these also are identical. In fact there is a one-to-one correspondence or duality between the two designs. This duality is very significant because it means that all of the curves and design relations for optimum multistage decimator designs developed in the previous section can be applied equally well to the design of optimum multistage interpolators.

In terms of practical implementations, it was shown that in the case of decimators, it was advantageous to choose integer decimation ratios and take advantage of the symmetry conditions of the FIR filters. For interpolators it is generally not possible to take advantage of such symmetry as we are already using the condition that $M_i - 1$ out of every M_i sample points entering the low-pass filter are zero valued (one exception to this occurs when $M_i = 2$ and N_i is odd). Therefore the condition that the interpolation ratios D_i be integer is not as important in terms of overall efficiency for interpolators as it was for the decimators.

V. Optimal Implementations of Narrow-Band FIR Filters by Decimation and Interpolation

In Sections III and IV we discussed the design of optimal implementations of decimators and interpolators. In this section we apply these concepts to the design of optimal multistage narrow-band filters. It is interesting to note that if a signal is decimated down to a low sampling rate and then interpolated back to the high rate, as shown in Fig. 10, we have, in effect, low-pass filtered the signal. Obviously the same result can be obtained directly with a low-pass filter operating at the high rate, f_{r0}. In this section it is shown that the first method, that of decimation and interpolation, is often a considerably more efficient way of implementing an FIR low-pass filter than the standard direct form implementation at the high sampling rate. This advantage becomes particularly significant when the bandwidth of the low-pass filter becomes small relative to the sampling rate.

It is convenient to first consider the case where the decimator and interpolator in Fig. 10 are one-stage designs. We assume that the desired specifications on the low-pass filter are f_s, f_p, δ_s, and δ_p as before. The amount of computation (MADS/s) required for the decimation stage can be obtained from (15). However, because there are two stages in the overall implementation (i.e., one for decimation and one for interpolation), $D_\infty(\delta_p/K, \delta_s)$ in (15) must be replaced by

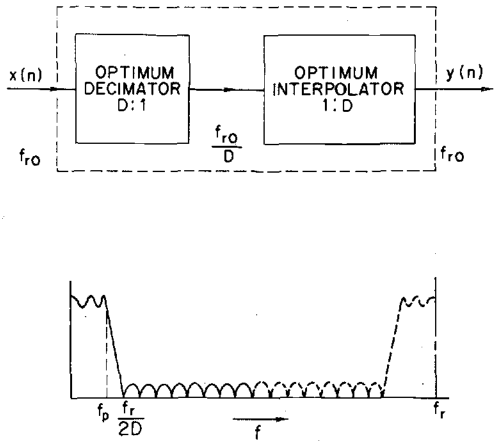

Fig. 10. Decimator-interpolator implementation of a narrow-band FIR low-pass filter.

$D_\infty(\delta_p/2K, \delta_s)$, i.e., the passband ripples must be reduced by an additional factor of 1/2 in order to keep the overall ripple within δ_p. The amount of computation R_D for the decimation stage (not using symmetry) is then

$$R_D \cong D_\infty\left(\frac{\delta_p}{2}, \delta_s\right) f_{ro} \cdot \frac{1}{1 - \left(\frac{f_s + f_p}{f_{ro}}\right) D}. \tag{30}$$

Because of duality, this quantity is also the amount of computation required for the interpolator. Therefore, the total amount of computation required is $2R_D$.

The amount of computation (MADS/s) required for a straightforward direct form FIR filter with the same specifications and not using symmetry is

$$R_{DF} = N f_{ro} \tag{31}$$

where N is the duration of the unit sample response. From (11) N can be approximated as

$$N \cong \frac{D_\infty(\delta_p, \delta_s)}{(f_s - f_p)/f_{ro}}. \tag{32}$$

Then R_{DF} is given as

$$R_{DF} \cong D_\infty(\delta_p, \delta_s)\, f_{ro}\, S' \text{ (MADS/s)} \tag{33a}$$

where

$$S' = \frac{f_{ro}}{f_s - f_p}. \tag{33b}$$

The net savings factor, SF, that can be achieved by using the decimator-interpolator implementation instead of the direct form implementation is then

$$\begin{aligned} \text{SF} &\cong \frac{R_{DF}}{2R_D} \\ &\cong \frac{D_\infty(\delta_p, \delta_s)}{D_\infty(\delta_p/2, \delta_s)} \cdot \frac{(f_s - f_p)/f_{ro}}{2(1 - ((f_s + f_p)/f_{ro}) D)}. \end{aligned} \tag{34}$$

From Table I we can recognize that for typical values of δ_p and δ_s

$$\frac{D_\infty(\delta_p, \delta_s)}{D_\infty(\delta_p/2, \delta_s)} \approx 1. \tag{35}$$

By applying this approximation in (34) and noting that $D = f_{ro}/2f_s$, we get

$$\text{SF} \approx \tfrac{1}{2} D \quad \text{(not using symmetry)}. \tag{36}$$

If symmetry is used in the direct form implementation, R_{DF} can be reduced by a factor of approximately 2 as can R_D for the decimator. However, as mentioned previously, no savings in multiplications due to symmetry are generally possible in the interpolator. Therefore, using symmetry, SF is approximately

$$\text{SF} \approx \tfrac{1}{3} D \quad \text{(using symmetry)}. \tag{37}$$

As D becomes large, or equivalently, the filter bandwidth becomes small, this savings becomes large. For example, for a filter with a bandwidth of 0.5 percent of the sampling rate, D is 100, and it can be implemented approximately 33 times more efficiently than a standard direct form FIR implementation.

For optimal multistage designs of the decimators and interpolators in Fig. 10 even greater efficiencies are possible. By arguments similar to those above it can be shown that the amount of computation (multiplications/s) required for a general multistage decimator-interpolator design, using symmetry in the decimation stage, is

$$R_{DI} \cong \frac{3}{2} D_\infty \left(\frac{\delta_p}{2K}, \delta_s\right) f_{ro} S \tag{38}$$

where S is plotted in the curves of Fig. 6. Similarly, the amount of computation (multiplications/s) required for a direct form FIR filter, using symmetry, is from (33a)

$$R_{DF} \cong \tfrac{1}{2} D_\infty(\delta_p, \delta_s)\, f_{ro} S' \tag{39a}$$

where S' can be expressed with the aid of (17) in the form

$$S' = \frac{2D}{\Delta f}. \tag{39b}$$

This relation is plotted as the dotted line in the curves of Fig. 6. The savings in using a general multistage design are then

$$\begin{aligned} \text{SF} &= \frac{R_{DF}}{R_{DI}} \\ &\cong \frac{1/2\, D_\infty(\delta_p, \delta_s)\, f_{ro} S'}{3/2\, D_\infty(\delta_p/2K, \delta_s)\, f_{ro} S} \\ &\approx \frac{1}{3} \frac{S'}{S} \quad \text{(using symmetry)}. \end{aligned} \tag{40}$$

By comparing the plots of S and S' in Fig. 6 an indication of these savings as a function of D and Δf can be obtained. For large values of D and small values of Δf savings factors in excess of 500 are possible.

We can further argue that this multistage approach for implementing narrow-band FIR filters, as shown in Fig. 10, is an optimal multistage design. That is, in the design of a multistage low-pass filter it is always most efficient to decimate as far down in sampling rate as possible, i.e., to $2f_s$, and then interpolate back rather than decimating to some intermediate sampling rate, low-pass filtering, and interpolating back. This is because the gain in efficiency monotonically increases as D increases (see Fig. 6). Therefore it is always preferable to make D as large as possible and to use optimal designs for the decimation and interpolation stages. By the same argument, if we wish to simultaneously low-pass filter a signal and perform a sampling rate conversion, it is always most efficient to optimally decimate to the lowest permissible sampling rate first and then interpolate to the new output sampling rate.

In a companion paper [6] some additional interesting properties of these decimator-interpolator filters are discussed.

VI. Additional Practical Considerations in the Implementation of Decimation and Interpolation Stages

In Section II we demonstrated that each stage of a decimator or interpolator can generally be in the form given in Fig. 3(b), and that for the specific cases of integer decimations and integer interpolations this form reduces to those of Fig. 1(b) and Fig. 2(b), respectively. In practice the implementation of these modules can be very complicated [particularly the general module in Fig. 3(b)], since it involves signals at several different sampling rates and also involves keeping track of when multiplications must be performed. In this section we present a straightforward method of implementing this general stage which alleviates many of these problems. For the case of integer decimations or interpolations the approach simplifies accordingly.

Assume that the length of the unit sample response of the low-pass filter in Fig. 3(b) is N. $L - 1$ out of every L samples of $w(n)$ are of zero value, and therefore the computation is proportional to N/L since the unit sample response spans approximately N/L nonzero samples. Schafer and Rabiner[5] [4] have observed that if N is chosen to be

$$N = QL \tag{41}$$

where Q is an integer, then the unit sample response will span *exactly* Q nonzero samples of $w(n)$ for each filter cycle. We assume for convenience that N is chosen such that it satisfies this condition. It is then possible to express the computation of an output point $y(n)$ in the form

$$y(n) = \sum_{k=0}^{Q-1} h(kL + (nM) \oplus L)\, x\left(\left[\frac{nM}{L}\right] - k\right) \tag{42}$$

where $(\ )\oplus L$ implies the quantity in parentheses modulo L and $[\ ]$ corresponds to the integer part of the number in brackets. The sequence $h(n)$, $n = 0, 1, \cdots, N-1$ contains the coefficients of the direct form FIR filter or equivalently the samples of its unit sample response. Equation (42) was determined empirically from several examples. It has the general characteristics of a convolution and when $M = L = 1$ it becomes the convolution equation as expected.

To illustrate the relationship depicted by (42) we will examine the case where $L = 2$, $M = 3$, and $Q = 4$. The first output point $y(0)$ is computed by noting that

$$y(0) = v(0) = \sum_{m=0}^{N-1} h(m)\, w(0 - m)$$

where $N = QL$. Fig. 11 (a) and (b) show $h(m)$ and $w(0 - m)$ respectively where $w(m)$ is constructed from $x(n)$ by inserting $L - 1$ zero-valued samples between the samples of $x(n)$. It is then easy to see that

$$y(0) = h(0)\,x(0) + h(2)\,x(-1) + h(4)\,x(-2) + h(6)\,x(-3).$$

To compute the point $y(1)$ we note that

$$y(1) = v(3) = \sum_{m=0}^{N-1} h(m)\, w(3 - m).$$

The sequence $w(3 - m)$ is illustrated in Fig. 11(c). Comparing the curves of (a) and (c) we see that $y(1)$ can be computed as

$$y(1) = h(1)\,x(1) + h(3)\,x(0) + h(5)\,x(-1) + h(7)\,x(-2).$$

Similarly, referring to Fig. 11(a) and (d), $y(2)$ can be computed as

$$y(2) = v(6) = \sum_{m=0}^{N-1} h(m)\, w(6 - m)$$

$$= h(0)\,x(3) + h(2)\,x(2) + h(4)\,x(1) + h(6)\,x(0).$$

In the computation of each output point $y(n)$, $x(n)$ is *sequentially* addressed for Q of its values. According to (42), it must then be advanced by $[nM/L]$ for the computation of the next output point. Also $h(n)$ must be addressed by $(kL + (nM) \oplus L)$. Alternatively, $h(n)$ can be addressed sequentially if the samples of $h(n)$ are stored in an appropriate scrambled order. That is, to compute $y(0)$ the sequence of Q samples $g_0(n) = h(0), h(L), \cdots, h((Q-1)L)$ is required. To compute $y(1)$ the sequence of Q samples $g_1(n) = h(M \oplus L)$, $h(L + M \oplus L), \cdots, h((Q-1)L + M \oplus L)$ is required. To compute $y(L-1)$, the sequence of Q samples $g_{L-1}(n) = h(((L-1)M) \oplus L), h(L + ((L-1)M) \oplus L), \cdots, h((Q-1)L + ((L-1)M) \oplus L)$ is needed. Finally, to compute $y(L)$ the sequence $g_0(n)$ is again required and the cycle repeats. Therefore, if the coefficients of $h(n)$ are arranged in the scrambled order, as shown in Fig. 12, they can be addressed in a *sequential* manner for the calculation in (42). On each computation of an output point $y(n)$, one block of Q coefficients is sequentially addressed. For the computation of the next output points $y(n+1), \cdots, y(n+2), \cdots, y(n+L-1)$, the address pointer is indexed sequentially until it has cycled through all L sequences $g_0(n), g_1(n), \cdots, g_{L-1}(n)$ at which point the address pointer is reset and the cycle begins again.

This approach provides an efficient way of handling the data in a general section. For $L = 1$ or $M = 1$ it simplifies accordingly and, if $L = M = 1$, then it is equivalent to a straightforward direct form convolution. If the assumption on the length of the impulse response (41) is not made, a similar formulation results in which the sequences $g_0(n)$-$g_{L-1}(n)$ are not all of equal length.

[5] In [4] the additional constraint that N be an odd integer was imposed; here this constraint need not be applied.

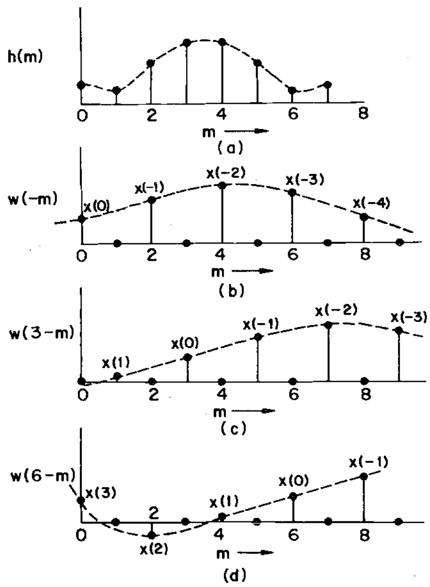

Fig. 11. Illustration of (42).

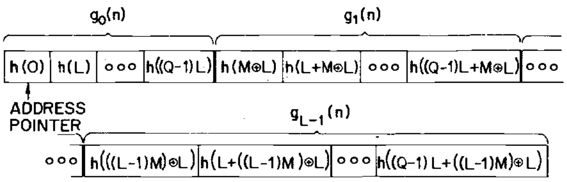

Fig. 12. Illustration of the scrambled order for storing the coefficients $h(n)$ so that they can be accessed in a sequential manner.

VII. Conclusions

In this paper we have discussed issues in the design of multistage decimators and interpolators and have presented curves and relations for optimal implementations. It was shown that the two processes, decimation and interpolation, are duals and that the same set of design curves can be applied to both. We then applied these techniques to the design of narrow-band FIR filters by a process of decimation and interpolation and showed that this led to a highly efficient way of implementing such filters. In the last section a practical approach toward implementing the individual stages of these multistage sampling rate changers was presented. In a companion paper [6] additional properties of the implementation of narrow-band FIR filters using these techniques are discussed, and illustrated with several design examples.

References

[1] D. J. Goodman, "The application of delta modulation to analog-to-digital PCM encoding," *Bell Syst. Tech. J.*, vol. 48, pp. 321-344, Feb. 1969.

[2] J. L. Flanagan, *Speech Analysis, Synthesis, and Perception*. New York: Springer-Verlag, 1972.

[3] S. L. Freeney, 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 (Special Issue on Active and Digital Networks)*, vol. CT-18, pp. 702-711, Nov. 1971.

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

[5] 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 Processing*, vol. ASSP-22, pp. 231-235, Aug. 1974.

[6] L. R. Rabiner and R. E. Crochiere, "A novel implementation for narrow-band FIR digital filters," this issue, pp. 457-464.

[7] A. W. Crooke and J. W. Craig, "Digital filters for sample-rate reduction," *IEEE Trans. Audio Electroacoust. (Special Issue on Digital Filtering)*, vol. AU-20, pp. 308-315, Oct. 1972.

[8] 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.

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

[10] O. Herrmann, L. R. Rabiner, and D. S. K. Chan, "Practical design rules for optimum finite impulse response low-pass digital filters," *Bell Syst. Tech. J.*, vol. 52, pp. 769-799, July-Aug. 1973.

[11] L. R. Rabiner, "Approximate design relationships for low-pass FIR digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 456-460, Oct. 1973.

[12] R. Hooke and T. A. Jeeves, "'Direct search' solution of numerical and statistical problems," *J. Ass. Comput. Mach.*, vol. 8, pp. 212-229, Apr. 1961.

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

[14] G. Oetken and H. W. Schüssler, "On the design of digital filters for interpolation," *Arch Elek. Übertragung.*, vol. 2, pp. 471-476, 1973.

[15] G. Oetken, T. W. Parks, and H. W. Schüssler, "New results in the design of digital interpolators," *IEEE Trans. Acoust., Speech, Signal Processing (Special Issue on 1974 Arden House Workshop on Digital Signal Processing)*, vol. ASSP-23, pp. 301-309, June 1975.