Further Considerations in the Design of Decimators and Interpolators

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

Abstract—In this paper several issues concerning the design and implementation of multistage decimators, interpolators, and narrow-band filters are discussed. In particular, the question of designing these systems in terms of minimum storage rather than minimum computation rate is examined. It is shown that a design which uses finite impulse response (FIR) filters for each stage, and which is minimized for storage is essentially minimized in terms of computation rate as well. The problem of further improvements in designing decimators and interpolators by taking advantage of DON'T CARE frequency bands is also discussed. For the early stages in a multistage design it is shown that fairly significant reductions in filter order can be achieved in this manner. A third issue in the design process is the question of practical schemes for efficient implementation of multistage decimators and interpolators in both hardware and software. One such efficient implementation is discussed in this paper. Finally, the problem of designing multistage decimators and interpolators using elliptic infinite impulse response (IIR) filters is discussed. It is shown that multistage IIR designs can be somewhat more efficient computationally than single-stage designs; however, the storage efficiency of the multistage IIR design is worse than that of the single-stage IIR design.

I. INTRODUCTION

IN earlier papers [1], [2] a general theory of multistage decimators and interpolators for sample-rate conversion and narrow-band filtering was discussed. In this paper we expand upon some of these ideas and discuss further issues in the design of decimators and interpolators. In particular, in Section II we address the issue of designing multistage decimators and interpolators for minimum storage as opposed to designing for minimum computation. It is shown that the two solutions are relatively close and that a design which is minimized on the basis of storage is essentially minimized in terms of computation as well. A complete set of design curves for minimum storage are given to complement similar curves in [1].

In Section III we discuss further improvements of multistage decimators by using multistopband filter designs in place of low-pass filters. It is shown that additional gains in efficiency up to about 25 percent are possible with this approach.

In Section IV we discuss further issues of implementing finite impulse response (FIR) decimators and interpolators in both software and hardware. A block diagram is presented illustrating a basic strategy for such an implementation. Finally, Section V deals with the use of infinite impulse response (IIR) filter designs in the implementation of multistage decimators and interpolators. Design curves are presented and compared with those for FIR designs.

Much of the discussion in this paper is a continuation and a direct extention of concepts developed in [1], [2]. For the sake of brevity we will not repeat any of that discussion, but will assume that the reader is familiar with this work. The notation used in this paper is that which was established in [1]. Also, as shown in [1], the design relations for decimators and interpolators are dual and the same set of design curves apply to both designs. Therefore, the subsequent design relations and curves in this paper, although formulated for decimators, apply equally to the design of interpolators as well.

II. MINIMIZATION OF STORAGE VERSUS MINIMIZATION OF MULTIPLICATIONS

In [1], [2] we considered the total number of multiplications and additions (MADS) as a criterion for optimization in a multistage decimator or interpolator design. It was also observed that a considerable savings in the total number of storage locations of a single-stage implementation could be accrued using a multistage implementation. In this section we consider the problem of designing multistage decimators and interpolators for minimum storage instead of minimum computation.

The development of the problem is nearly identical to that of minimizing the amount of computation. It can be assumed that the total number of storage locations (for both coefficient and data storage) in the multistage design is approximately proportional to the sum of the lengths of the filters in each of the stages. That is,

$$N_T \approx G \sum_{i=1}^{K} N_i, \tag{1}$$

where N_T is the total number of storage locations necessary, N_i is the length of the FIR filter for the ith stage, K is the total number of stages, and G is a proportionality constant which depends on whether we are implementing a decimator, interpolator, or low-pass filter, and on the particular manner in which it is implemented. From [1, eq. (20)][1] it was shown that N_i can be expressed as

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

where D is the total sampling rate reduction for the multistage decimator, $\Delta f = (f_s - f_p)/f_s$, f_p is the passband edge and f_s is one-half of the final sampling rate. K is the total number of

Manuscript received September 24, 1975; revised March 1, 1976.
The authors are with Bell Laboratories, Murray Hill, NJ 07974.

[1] *Note:* There is a typesetting error in [1, eq. (20)]. It should read as in (2) of this paper.

stages and $D_\infty(\ )$ is a function of δ_p and δ_s, the passband and stopband tolerances, respectively, for the filters in each decimation stage (see [1]).

We will assume in this development that $L_i = 1$, $i = 1, 2, \cdots, K$. That is, the decimator (or interpolator) has integer decimation (or interpolation) ratios, D_i, at each stage. For designs with noninteger ratios, the integer L_i must be greater than one and, as seen from (2), this implies a much larger value for N_i. This serves as a strong inducement against using stages with noninteger decimation ratios (in addition to the reasons given in [1]).

Using (1) and (2), N_T can be expressed as

$$N_T \approx GD_\infty\left(\frac{\delta_p}{K}, \delta_s\right) \sum_{i=1}^{K} \frac{D_i}{1 - \frac{2 - \Delta f}{2D} \prod_{j=1}^{i} D_j}. \tag{3}$$

Noting that $D = D_1, D_2, \cdots, D_K$, N_T can be further expressed in the form of a product of functions,

$$N_T \approx GD_\infty\left(\frac{\delta_p}{K}, \delta_s\right) T, \tag{4a}$$

where

$$T = T(D, \Delta f, K; D_1, D_2, \cdots, D_{k-1}) = \frac{2D}{\Delta f \prod_{j=1}^{K-1} D_j} + \sum_{i=1}^{K-1} \frac{D_i}{1 - \frac{2 - \Delta f}{2D} \prod_{j=1}^{i} D_j}. \tag{4b}$$

As shown in [1], $D_\infty(\delta_p/K, \delta_s)$ is a relatively weak function of K. The more interesting function is T. N_T can be minimized for a given K by minimizing T as a function of the D_i's and then choosing the value of K which minimizes N_T. To minimize T as a function of the D_i's an optimization routine (in this case the Hooke and Jeeves algorithm [3]) is used.

Fig. 1 gives plots of the minimum T as a function of D and Δf and Fig. 2 gives plots of the corresponding decimation ratios for minimum storage designs. For comparison, the dotted lines in Fig. 1 correspond to values of T when the design is minimized for multiplications. For a one-stage implementation the two designs are obviously identical and for a two-stage implementation they are essentially indistinguishable on the curves in Fig. 1. For three- and four-stage implementations, savings in storage of at most 2:1 are possible using a minimized storage design instead of a minimized multiplication design. Thus the two solutions are relatively close and their difference becomes smaller as Δf becomes small.

In comparing the decimation ratios of a minimized storage design (Fig. 2) to those of a minimized multiplication design [1, Fig. 7], it is seen that the minimized storage design favors slightly lower decimation ratios for the lower numbered stages and slightly larger ratios for the higher numbered stages.

In comparing the number of required multiplications in a minimum storage design to that of a minimum multiplication design it was found that the curves are essentially indistinguishable from those of [1, Fig. 6]. Thus a design that is minimized in terms of storage is essentially minimized in terms of the number of multiplications as well. This is a consequence of the fact that minima for S are relatively broad minima whereas minima for T are slightly narrower.

A final important conclusion which can be drawn in comparing curves of S and T ([1, Fig. 6] and Fig. 1) for large values of D is that, in terms of multiplication rate, only small gains in efficiency are obtainable by using more than two or three stages; however, substantial gains in efficiency of storage are still possible by going to three and four-stage designs.

Together, the curves in Figs. 1 and 2 and those in [1] provide a useful set of guidelines for choosing practical D_i values for a wide range of decimator or interpolator designs. All of these curves are based on the approximation to filter order given by [1, eq. (11)]. A similar set of curves was generated using [1, eq. (9)][2] and they were found to be essentially indistinguishable to those presented here and in [1], thus justifying the use of this approximation.

III. Further Improvement of Decimator or Interpolator Designs Using Multiple Stopband Filter Designs

In the original design procedure for multistage decimators and interpolators the digital filter at each stage was designed as a low-pass filter whose job was to eliminate the frequency bands which would either alias back into the baseband (for a decimator) or appear as images of the baseband (for an interpolator). In this section it is shown how, for some of the stages, the low-pass filter can be replaced by a multiband digital filter of lower order than the original low-pass filter—thereby reducing the overall computation for the multistage design.

Fig. 3 shows the filtering requirements for the ith stage of a multistage decimator. The initial sampling frequency is $f_{r(i-1)}$ and the final sampling frequency is f_{ri}, defined as

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

where D_i is the decimation rate for the ith stage. The baseband is defined as the region from $f = 0$ to $f = f_s$. The bands which are aliased back into the baseband after sample rate reduction are shown as dotted regions in Fig. 3. Each of these bands is centered at integer multiples of the final sampling frequency for the stage, i.e., $f = f_{ri}, 2f_{ri}, \cdots$, and is of width $2f_s$. The regions between the dotted bands are DON'T CARE regions (and are denoted with a ϕ in Fig. 3) in that the frequency response of the filter can be largely left unspecified and unconstrained in these bands.

When realized as a conventional low-pass filter the specifications of Fig. 3 are the following:

$$1 - \delta_p \leq |H_i(e^{j(2\pi f/f_{r(i-1)})})| \leq 1 + \delta_p \qquad 0 \leq f \leq f_p$$
$$0 \leq |H_i(e^{j(2\pi f/f_{r(i-1)})})| \leq \delta_s \qquad f_{ri} - f_s \leq f \leq 0.5 f_{r(i-1)}, \tag{6}$$

[2] These curves had to be generated for specific values of δ_p and δ_s as N_i could no longer be simply factored into a product of δ_p and δ_s and a product of the D_i's. The use of [1, eq. (9)] also greatly compounded the difficulty of the optimization routine in finding a minimum as undesired local minima outside of the range feasible solutions were introduced by inclusion of the term $f(\delta_1, \delta_2)\,\Delta F$.

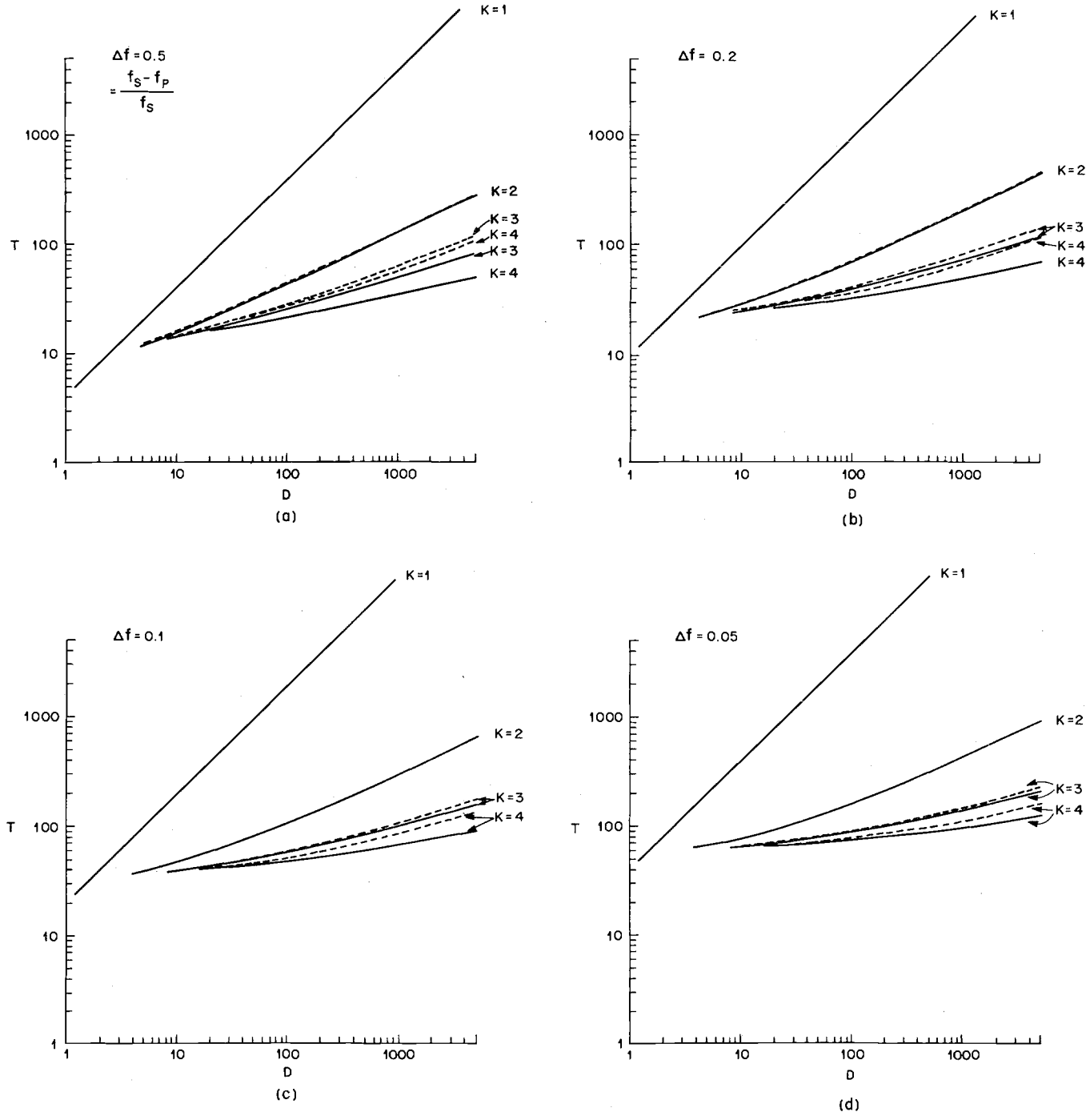


Fig. 1. Plots of minimized values of T as a function of K, D, and Δf where Δf is 0.5, 0.2, 0.1, 0.05, 0.02, and 0.01 for plots (a)-(f), respectively.

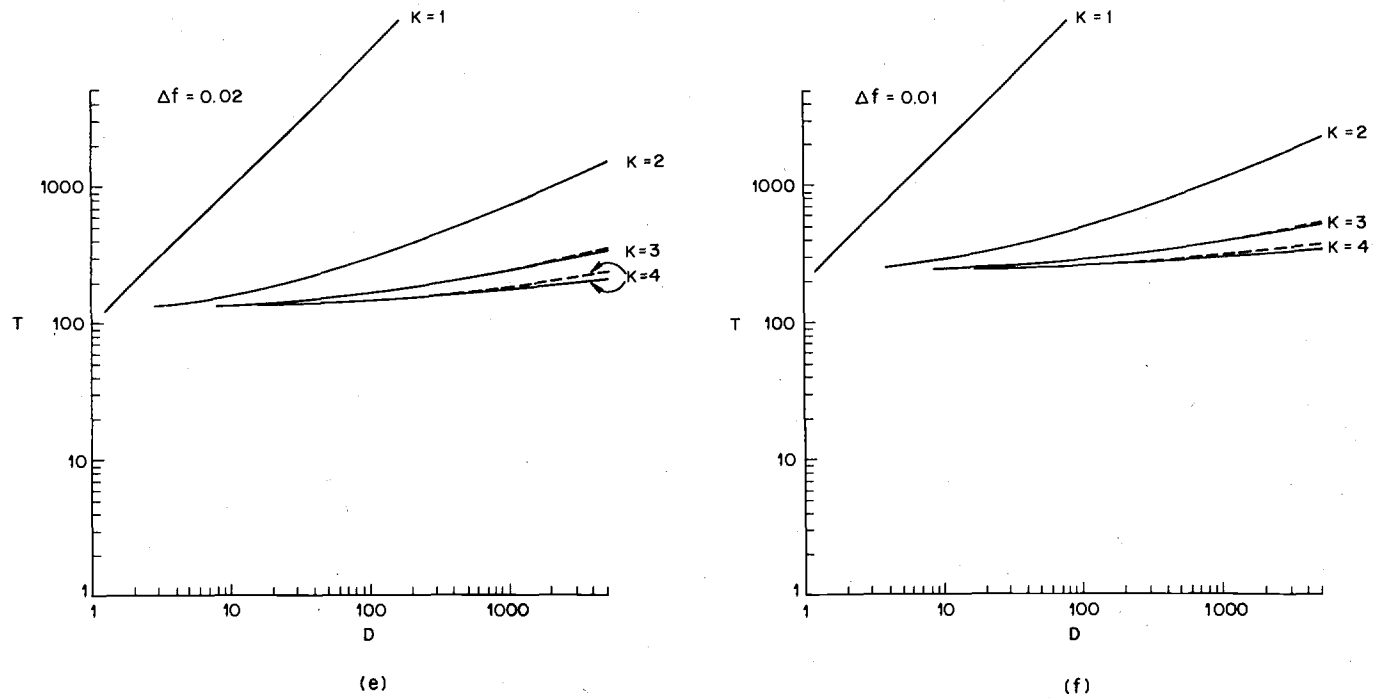


Fig. 1. (Continued.)

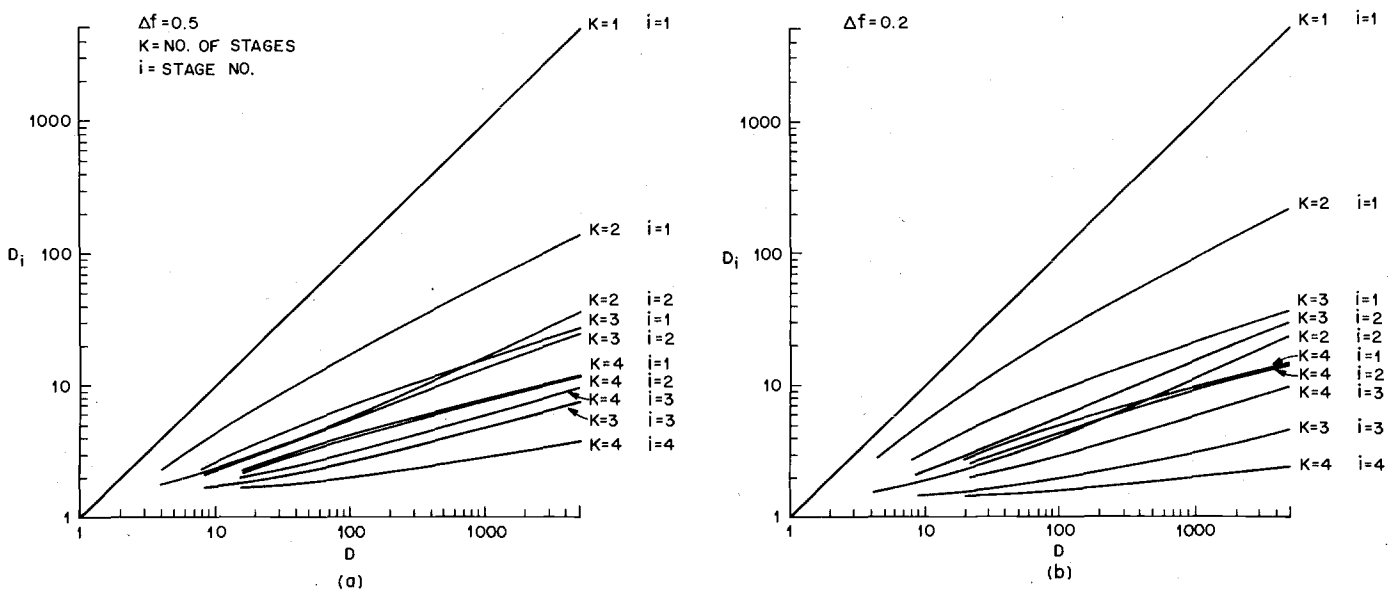


Fig. 2. Plots of optimum decimation ratios of minimized storage designs as a function of K, D, and Δf where Δf is 0.5, 0.2, 0.1, 0.05, 0.02, and 0.01 for plots (a)–(f), respectively.

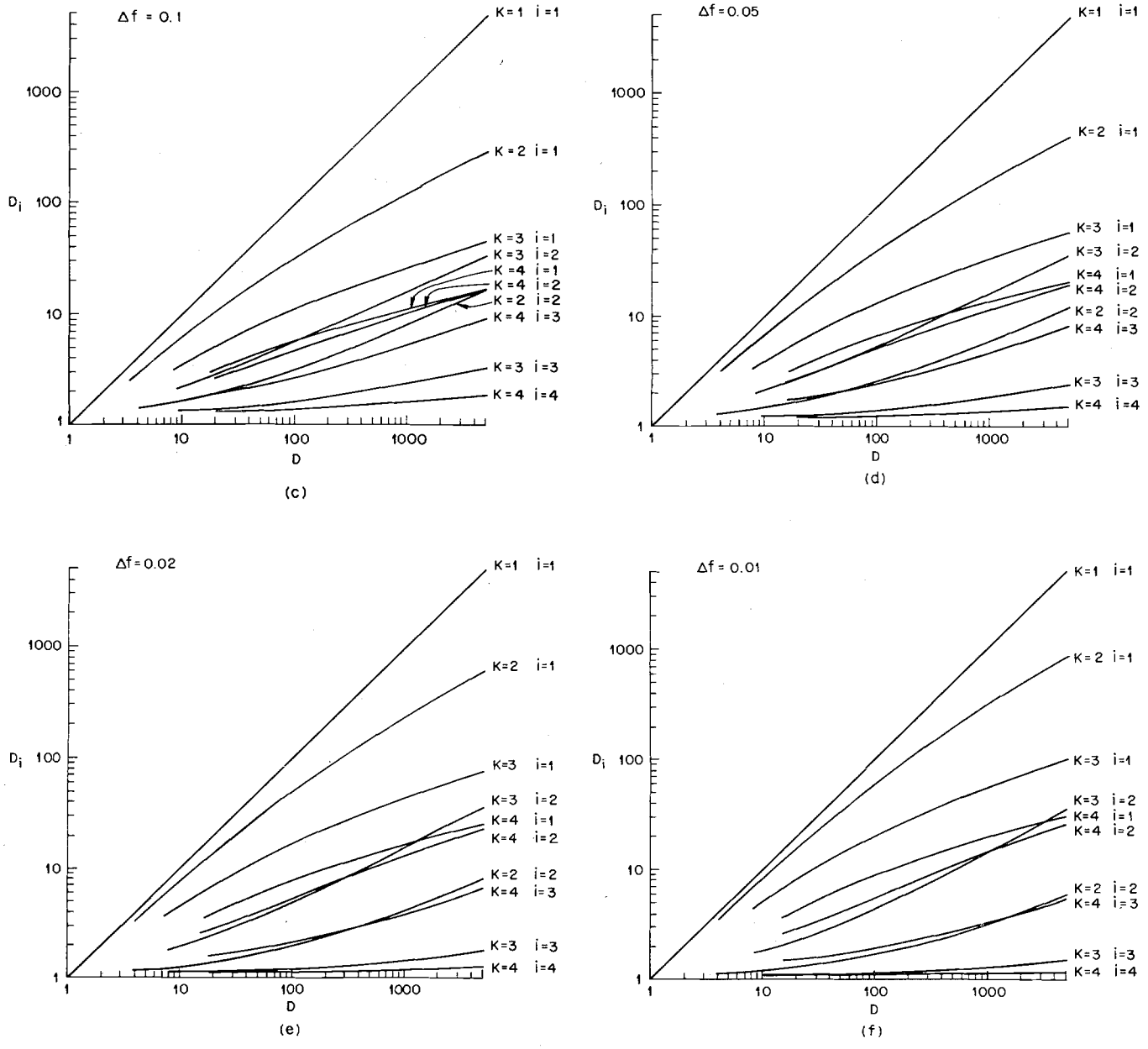


Fig. 2. (Continued.)

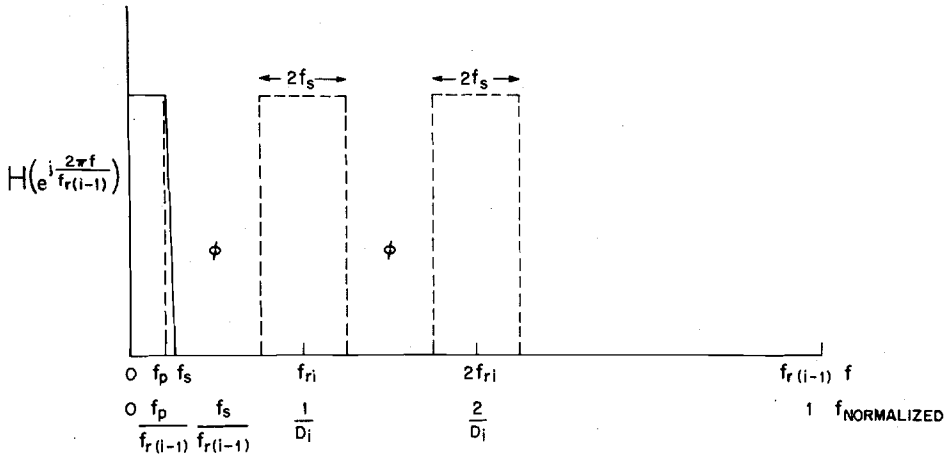


Fig. 3. Multiband filtering specifications for the ith stage of a multistage decimator or interpolator.

where $H_i(e^{j[2\pi f/f_{r(i-1)}]})$ is the required filter response. However, when the width of the DON'T CARE bands becomes comparable to the width of the individual stopbands of Fig. 3, fairly significant reductions in filter order can be achieved by designing a multiband filter with the specifications

$$\begin{aligned} 1-\delta_p &\leq |H_i(e^{j[2\pi f/f_{r(i-1)}]})| \leq 1+\delta_p \qquad 0 \leq f \leq f_p \\ 0 &\leq |H_i(e^{j[2\pi f/f_{r(i-1)}]})| \leq \delta_s \qquad f_{ri}-f_s \leq f \leq f_{ri}+f_s \\ 0 &\leq |H_i(e^{j[2\pi f/f_{r(i-1)}]})| \leq \delta_s \\ & \qquad\qquad 2f_{ri}-f_s \leq f \leq 2f_{ri}+f_s. \\ &\vdots \end{aligned} \tag{7}$$

As can be seen by comparing (6) and (7), the specifications for the passband and the transition band are identical. The differences occur primarily in the placement of the DON'T CARE bands following the initial stopband.

To illustrate the possible reductions in filter order which can be obtained by using this multiband approach to the design of the individual filters, Fig. 4 shows two curves of the percentage decrease in filter order (N) as a function of normalized stopband cutoff frequency for stages with decimation ratios of 5 [Fig. 4(a)] and 10 [Fig. 4(b)]. The percentage decrease is defined as the percentage reduction in N going from the low-pass case to the multiband case. The curves of Fig. 4 are measured curves (i.e., they are not theoretical computations) for actual designs with δ_p set to 0.001 and δ_s set to 0.0001. The data points on these curves are shown separately, and a smooth curve is used to show the trend in the data. The numbers to the right of each data point are the filter impulse response durations (in samples) for the low-pass, and multiband designs, respectively. For all of the designs of Fig. 4, the passband edge f_p was set equal to the stopband edge f_s to eliminate one variable in the plots. It is readily shown that the results of Fig. 4 are essentially independent of the exact relation between f_p and f_s for most practical ranges of f_p and f_s.

Fig. 4(a) shows that as the normalized stopband cutoff frequency becomes smaller (i.e., the width of the DON'T CARE bands becomes larger), the percentage decrease in N rises fairly rapidly. The interpretation of this result is that when the ith stage is an early stage in a multistage design, where the normal-

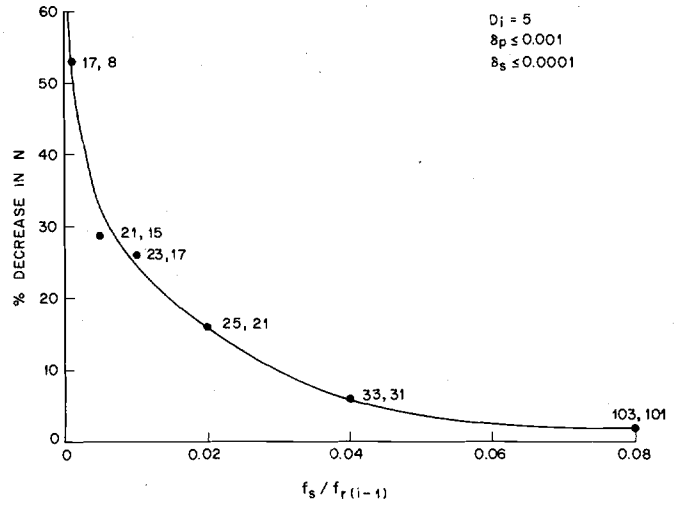


Fig. 4. Percentage decrease in filter order obtained by using a multiband design instead of a low-pass design as a function of $f_s/f_{r(i-1)}$ for $D_i = 5$ and $D_i = 10$ for (a) and (b), respectively.

ized stopband cutoff frequency will be small, there are significant reductions in filter order which can be achieved by using the multiband design approach. However, when the ith stage is a latter stage in the chain (i.e., the normalized stopband cutoff frequency is relatively large), the percentage reduction in filter order obtained by using a multiband design is small.

Fig. 4(b) shows similar results for a stage with a decimation ratio of 10. By extrapolating to the limits of the curves of Fig. 4 (as F_s becomes 0), it can be shown that a decimation ratio of D_i can be obtained using a simple filter–in particular a filter whose impulse response is a rectangular window of duration D_i samples will suffice for most practical cases. Goodman [4], [5] has exploited this result in proposing some extremely simple structures for realizing multistage decimators and interpolators.

To illustrate some specific design examples, Figs. 5 and 6 show comparisons between a low-pass filter and its equivalent multiband design for two different sets of specifications. For Fig. 5 the decimation ratio was $D_i = 5$, the passband cutoff frequency (normalized) was $f_p = 0.0475$, and the stopband cutoff frequency (normalized) was $f_s = 0.05$. Values of $\delta_p = 0.001$ and $\delta_s = 0.0001$ were used. Fig. 5(a) shows the low-pass filter meeting the design specifications, and Fig. 5(b) shows the multiband filter meeting the same specifications. The locations of the DON'T CARE bands are quite clear from this figure. In this example the reduction in filter order from the low-pass design ($N = 41$) to the multiband design ($N = 39$) was small because of the relatively wide width of the baseband.

Fig. 6 shows the resulting filters for an example with $D_i = 10$, $f_p = 0.00475$, $f_s = 0.005$. The reduction in filter order from the low-pass design ($N = 45$) to the multiband design ($N = 33$) is quite significant in this example.

The following examples illustrate the overall effect on the computation rate of substituting multiband filters for low-pass filters.

Example 1: Consider the design of a two-stage decimator ($D = 10$) with specifications

$$f_p = 0.025 \quad \delta_p = 0.01$$
$$f_s = 0.05 \quad \delta_s = 0.001$$
$$f_{ro} = 1.0 \quad D = 10.$$

In [2] it was shown that a practical implementation of this decimator had decimation rates of $D_1 = 5, D_2 = 2$ for the two stages. Thus the filter orders for a low-pass design were $N_1 = 25, N_2 = 27$ for the two stages. The total multiplication rate for the two-stage decimator was shown to be [2]

$R_{1D} = 13/5 = 2.6$ first decimation stage

$R_{2D} = 14/10 = 1.4$ second decimation stage

$R_T = 4.0$ multiplications/s.

If the low-pass filter in the first stage is replaced by a multiband filter, the required filter order is reduced to $N = 22$. For the second stage, no reduction in filter order is possible since $D_i = 2$ and for a 2:1 reduction the low-pass and multiband designs are identical. Thus for the overall decimator the total multiplication rate becomes

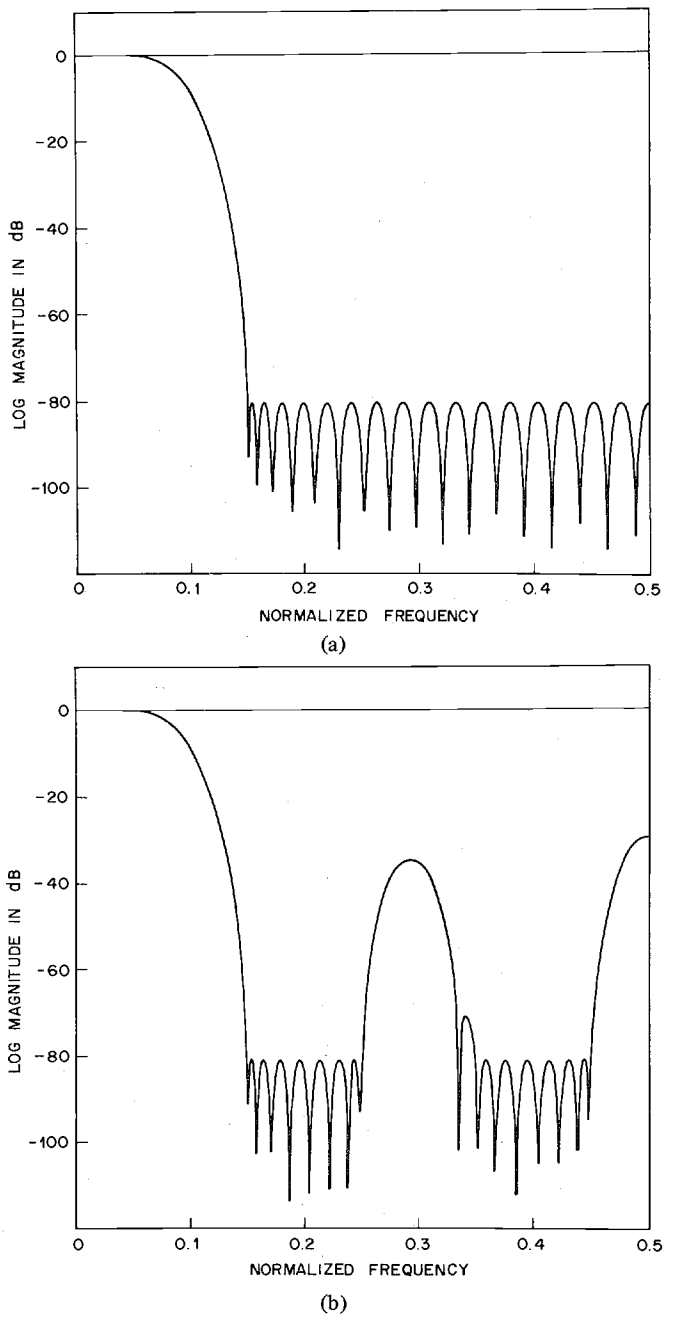


Fig. 5. Comparison between (a) a low-pass filter and (b) its equivalent multiband design for $D_i = 5$.

$R_{1D} = 11/5 = 2.2$ first decimation stage

$R_{2D} = 14/10 = 1.4$ second decimation stage

$R_T = 3.6$ multiplications/s.

The net savings in multiplications is 0.4 or a 10 percent savings.

Example 2: Consider the design of a three-stage decimator ($D = 100$) with specifications

$$f_p = 0.00475 \quad \delta_p = 0.001$$
$$f_s = 0.005 \quad \delta_s = 0.0001$$
$$f_{ro} = 1 \quad D = 100.$$

In [2] it was shown that a practical implementation of this

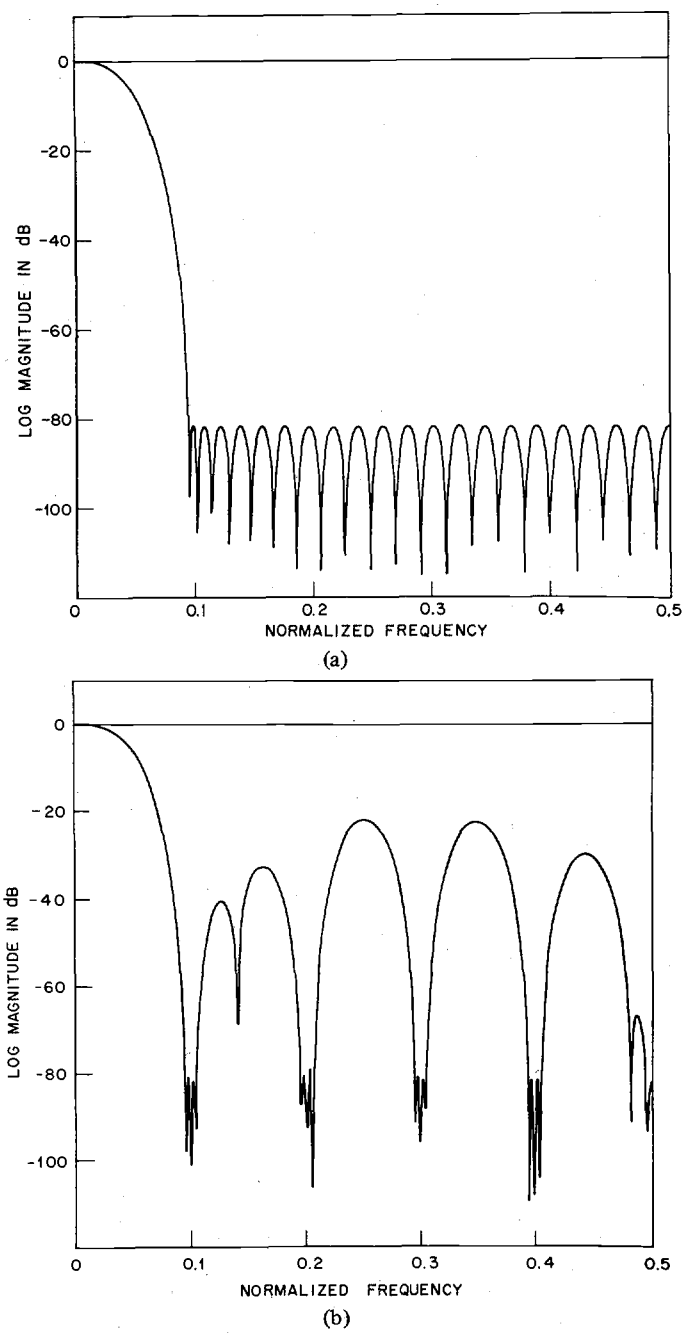


Fig. 6. Comparison between (a) a low-pass filter and (b) its equivalent multiband design for $D_i = 10$.

decimator had decimation rates of $D_1 = 10$, $D_2 = 5$, $D_3 = 2$, with filter order $N_1 = 50, N_2 = 44, N_3 = 356$ for the low-pass filters. The total multiplication rate for the decimator was shown to be

R_{1D} =	25/10 = 2.5	first decimation stage
R_{2D} =	22/50 = 0.44	second decimation stage
R_{3D} =	178/100 = 1.78	third decimation stage
R_T	= 4.72	multiplications/s.

By replacing the low-pass filters of the first two stages with equivalent multiband designs, the filter orders can be reduced to $N_1 = 38$ and $N_2 = 38$. Thus the overall multiplication rate becomes

R_{1D} =	19/10 = 1.9	first decimation stage
R_{2D} =	19/50 = 0.38	second decimation stage
R_{3D} =	178/100 = 1.78	third decimation stage
R_T	= 4.06	multiplications/s.

The net savings in multiplications is 0.66 or about 14 percent.

In summary, we have shown that for the early stages in a multistage design of decimators (or interpolators) a small, but significant, amount of savings in the required multiplication rate can be obtained by careful design of the digital filters required to remove the signal components which alias into the baseband. In particular, we have shown that the resulting design is a multiband digital filter with DON'T CARE bands between each of the bands of interest.

One interesting question which arises out of this discussion is whether or not it is beneficial to increase the number of stages in the design so as to exploit the computational advantages of the multiband approach to the fullest. This question is not a simple one to answer. Goodman [4], [5] has found that when using a large number of stages to implement a large decimator, the individual filters can be economically realized using simple prototypes, e.g., rectangular, or Hamming window designs. The argument against using a larger number of stages in the design is that the system complexity in terms of extra control and timing logic increases. Thus there are distinct tradeoffs here, and each specific application must be treated and studied independently.

IV. Additional Considerations in the Implementation of FIR Decimator and Interpolator Designs

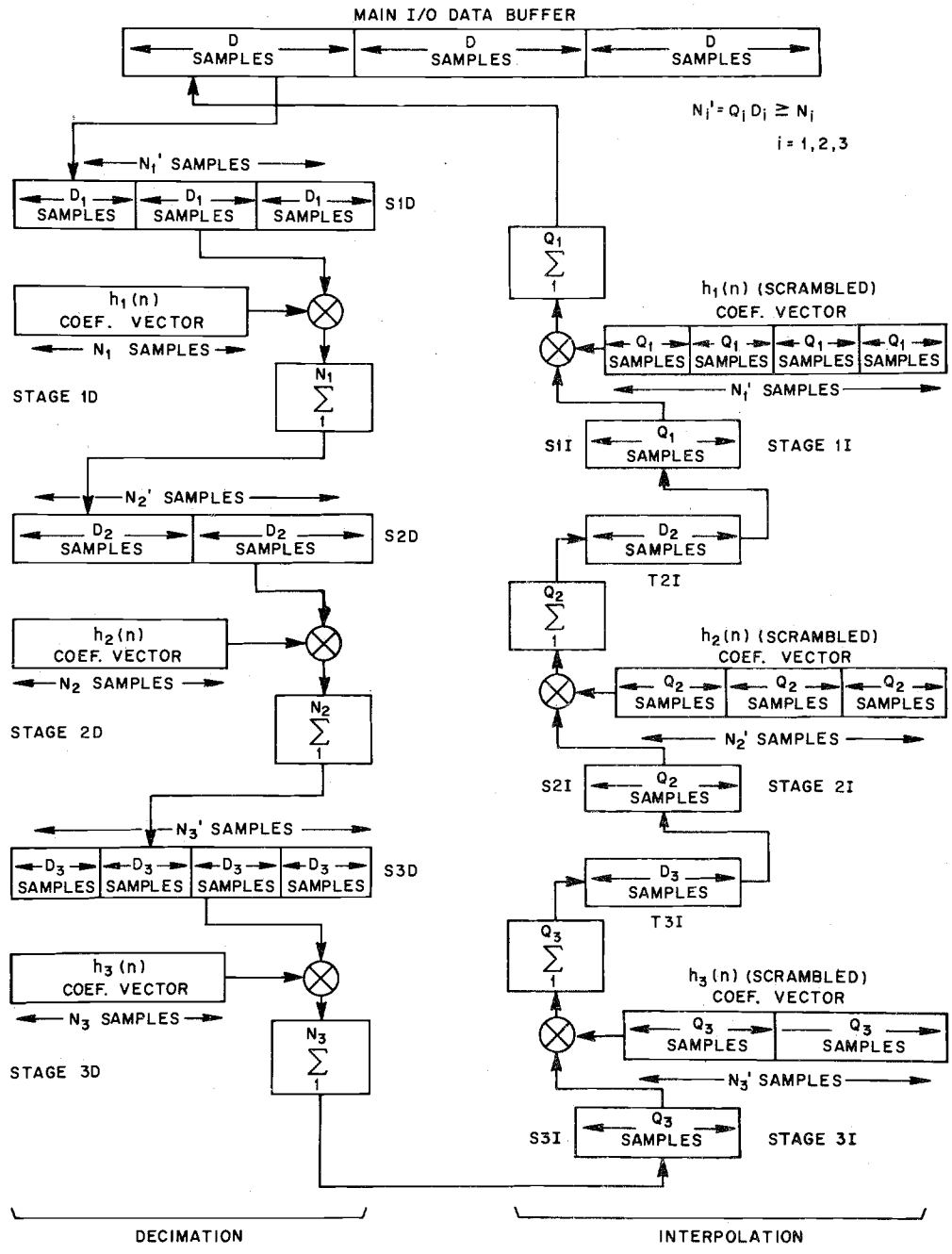
In [1] a practical scheme for implementing a general interpolator/decimator stage was presented. It involved the storage of the coefficients in a "scrambled" order such that both the data and coefficients could be *sequentially* addressed in the computation of the output samples. In this section we show how this scheme for a single stage can be incorporated into the overall structure of a multistage decimator or interpolator. We will restrict this discussion to the case of integer decimators and interpolators.

A general block diagram for the implementation of a three-stage decimator in cascade with a three-stage interpolator is shown in Fig. 7(a) and its corresponding control sequence is given in Fig. 7(b). Together, the cascade results in the implementation of a low-pass filter. To realize only a decimator or interpolator, appropriate parts of this structure can be partitioned off from the main structure.

The decimator has three data buffers (S1D, S2D, and S3D) for storage or internal data for its three stages. These storage buffers are of durations N_1', N_2', and N_3' words, respectively. These lengths are derived according to the relationship

$$N_i' = Q_i D_i \geq N_i \qquad i = 1, 2, 3, \tag{8}$$

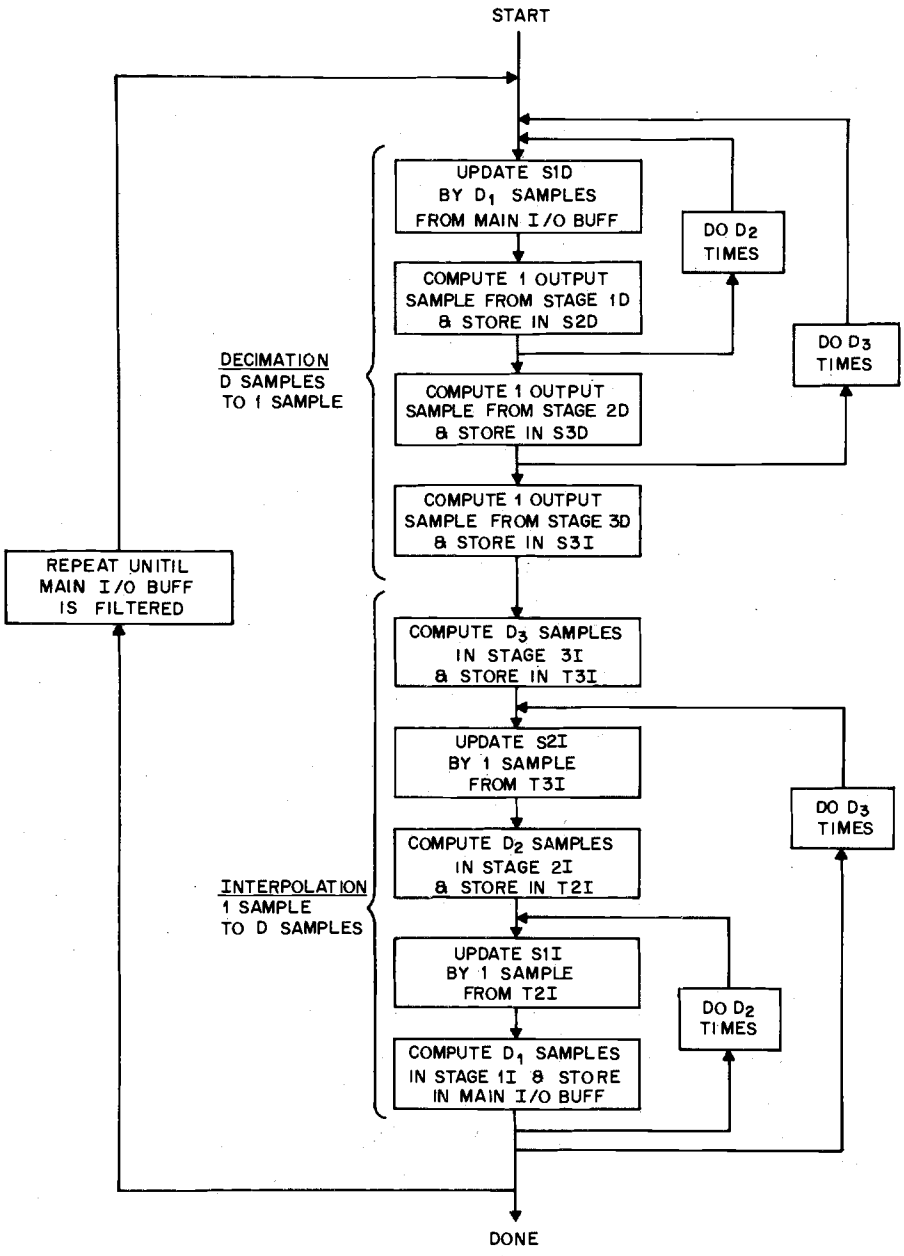
where Q_i is an integer (defined in [1]), N_i is the filter length for the ith stage and D_i is its decimation ratio. Each data



(a)

Fig. 7. (a) Block diagram for implementing a three-stage decimator followed by a three-stage interpolator and (b) its corresponding control sequence.

buffer is partitioned into Q_i blocks of data of length D_i. Three additional buffers hold the coefficients for the filters in each of the three stages. The interpolator has six data buffers associated with it, three (S1I, S2I, and S3I) of lengths Q_1, Q_2, and Q_3, respectively, and three (T1I, T2I, and T3I) of lengths D_1, D_2, and D_3, respectively. In addition, stage i has a buffer for the "scrambled" coefficients which are partitioned into D_i blocks of Q_i samples each. The operation of this structure is depicted by the control sequence in Fig. 7(b). The data buffers can be thought of as shift registers although they do not have to be implemented in this way. The process begins by reading D_1 samples from the main input/output (I/O) buffer into S1D. One output sample is then computed from stage one of the decimator and stored in S2D. This process is repeated D_2 times until D_2 samples have been computed and stored in S2D. One output is then computed for stage 2 of the decimator and stored in S3D. The above process is repeated D_3 times until D_3 samples have been stored in S3D at which point one output sample is calculated from stage 3 of the decimator. This completes one cycle of the decimator; $D = D_1 D_2 D_3$ samples have been read from the main I/O buffer and one output sample has been computed. A similar computation cycle can now proceed for the interpolator. The output sample from the decimator is stored into



(b)

Fig. 7. (Continued.)

S3I. D_3 output samples are computed from stage 3 of the interpolator and temporarily stored in T3I. One sample from T3I is then stored in S2I and D_2 output samples are computed from stage 2 and stored in T2I. S1I is then updated by one sample from T2I and D_1 outputs are computed and stored in the main I/O buffer. This is repeated until all D_2 samples in T2I are removed. T2I is then refilled by updating S2I with one sample from T3I and computing D_2 more samples. Upon completion of the interpolator cycle one input was transferred into the interpolator and D output samples are computed and stored in the main I/O data buffer. The process can then be repeated on the next block of D samples.

The above process was described as a serial process and represents a straightforward approach for a software implementation. If high-speed and/or special-purpose hardware are used, the structure in Fig. 7(a) is also particularly attractive as it easily lends itself (with slight modification) to various degrees of parallel processing and pipelining. For example, in the interpolation stages all D_i outputs of a stage i can be computed in parallel [6]. Similar degrees of parallelism are possible in the implementation of the decimator stages. Pipelining is also an attractive possibility with this structure since it is essentially a completely feed-forward structure [7]. Each stage can be separately implemented in hardware. In this case it may be more attractive to choose the design such that the amount of computation is equally divided among the stages rather than minimizing the total computation or total storage. Another feature of this structure which could be used in a pipeline scheme is that, because of the duality of the decimation and interpolation stages, many of the same control and timing signals used for the decimator could also be used for the interpolator as well.

V. IIR Designs for Multistage Decimators and Interpolators

Until now we have only considered the use of FIR filters in the implementation of decimators and interpolators. One reason for this is that FIR filters can be designed to have exactly linear phase. However, if linear phase is not an important consideration, then IIR filters can be considered as an attractive alternative since they can generally be implemented more efficiently than equivalent FIR designs. In this section it is shown that an optimization procedure similar to the one discussed in [1] can be used to design an optimal IIR multistage decimator, interpolator, or narrow-band filter. In this discussion we will only consider the case of integer decimation or interpolation stages. It is fairly straightforward to extend the results to noninteger stages.

To formulate the problem it is useful to reexamine the design of decimator and interpolator stages with integral ratios of sampling rates. As shown in [1, Fig. 1], a decimator with a decimation rate of M can be implemented by filtering the input signal with a low-pass filter (in this case an IIR filter) and then decreasing the sampling rate by selecting one out of every M samples of the output $w(n)$ of the filter. Unlike the FIR implementation, the output $w(n)$ of the IIR filter (or all states of its recursive part) must be computed for *all values of n* prior to decimating by M. In a multistage decimator design the specifications on the low-pass filters are the same as those discussed in [1] with $D_i = M_i$ for the ith stage. Similarly, for a $1:L$ interpolator, the same design specifications as discussed in [1] apply to the low-pass IIR filters at each stage. As in the FIR case, duality also applies in the case of IIR designs of decimators and interpolators and, therefore, all design formulas and curves for multistage IIR decimators also apply to multistage IIR interpolators.

With the above concepts in mind we can now formulate the relations for the multistage IIR decimator (or interpolator). We will assume that the IIR filters are *optimal elliptic designs* (obtained from bilinearly transformed analog designs). Then the filter order for stage i can be shown to be of the form [8]

$$N_i = A\left(\frac{\delta_p}{K}, \delta_s\right) B_i(D, \Delta f, K; D_1, D_2, \cdots, D_{k-1}) \tag{9}$$

where $A(\ )$ is a function of the ripples and B_i is a function of the cutoff frequencies and decimation ratios. The function $A_i(\ )$ has the form

$$A_i = A\left(\frac{\delta_p}{K}, \delta_s\right) = \frac{K(\sqrt{1-r^2})}{K(r)} \tag{10}$$

where

$$r = \frac{2\delta_s\sqrt{\frac{\delta_p}{K}}}{\left(1-\frac{\delta_p}{K}\right)\sqrt{\left(1-\frac{\delta_p}{K}\right)^2 - \delta_s^2}} \tag{11}$$

$K(\ )$ the complete elliptic integral of the first kind
δ_p passband ripple tolerance
δ_s stopband ripple tolerance

and

K total number of stages.

The function $B_i(\ )$ has the form

$$B_i = B_i(D, \Delta f, K; D_1, D_2, \cdots, D_{k-1}) = \frac{K(t_i)}{K(\sqrt{1-t_i^2})} \tag{12}$$

where

t_i = transition ratio of the filter for the ith stage

$$= \frac{\tan\left(\frac{\pi f_p}{f_{r(i-1)}}\right)}{\tan\left(\frac{\pi(f_{ri} - f_s)}{f_{r(i-1)}}\right)} \tag{13a}$$

and using previous terminology

$$f_{ri} = f_{ro} \Big/ \prod_{j=1}^{i} D_j \tag{13b}$$

$$f_p = (1 - \Delta f) f_s \tag{13c}$$

$$f_s = 1/(2D) \tag{13d}$$

$$D_K = D \Big/ \prod_{j=1}^{K-1} D_j. \tag{13e}$$

Minimization of the Multiplication Rate

With the above expression for filter order, the problem of minimizing a multistage IIR decimator or interpolator design in terms of its multiplication rate can readily be formulated. The multiplication rate for a single stage i is approximately proportional to its IIR filter order N_i times its (output) sampling rate $f_{r(i-1)}$. For a K-stage design, the total multiplication rate of the decimator (or interpolator) R_T can then be expressed in the form

$$R_T \approx \sum_{i=1}^{K} G' N_i f_{r(i-1)} \approx G' f_{ro} \sum_{i=1}^{K} \frac{N_i D_i}{\prod_{j=1}^{i} D_j} \tag{14}$$

where G' is a proportionality factor which is dependent on the method of implementation of the IIR filter structure. For example, in a conventional cascade structure, three multiplications are required for the implementation of a second-order section and, therefore, for this structure G' is approximately $\frac{3}{2}$ (and we must add one multiplication for the gain constant).

With the aid of (9) the above expression can be written in the form

$$R_T \approx G' A\left(\frac{\delta_p}{K}, \delta_s\right) f_{ro} \tilde{S} \tag{15a}$$

where

$$\tilde{S} = \tilde{S}(D, \Delta f, K; D_1, D_2, \cdots, D_{k-1}) = \sum_{i=1}^{K} \frac{B_i D_i}{\prod_{j=1}^{i} D_j} \Bigg|_{D_K = D \Big/ \sum_{j=1}^{K-1} D_j}. \tag{15b}$$

The reader should note the similarity of this expression to that of [1, eq. (18a) and (18b)]. It is seen that R_T can be expressed as a product of a function of the ripples, the initial sampling frequency, and a function of the cutoff frequencies and decimation ratios. In a similar manner, R_T can be minimized by minimizing $\tilde{S}$ as a function of the decimation (interpolation) ratios for each value of K and then choosing the value of K which minimizes the product. The function $A(\delta_p/K, \delta_s)$ is tabulated in Table I for convenience.

The minimization of $\tilde{S}$ can again be performed by an optimization routine such as the Hooke and Jeeves algorithm. This optimization is not a trivial one. The evaluation of the elliptic functions[3] can be extremely sensitive numerically. Great care must be used in controlling the range of parameter values to avoid both arithmetic overflow and roundoff error, and also in constraining the optimization search to be within the region of realizable solutions.

Using the Hooke and Jeeves method, the above optimization problem was solved. Plots of minimized values of $\tilde{S}$ are given in Fig. 8 and the corresponding optimum decimation ratios and $B_i(\ )$ values are given in Figs. 9 and 10, respectively. These curves and Table I (which gives values for the function A as δ_p and δ_s vary) can be used in a manner similar to that of the FIR design curves as a guide toward selecting practical integer values of decimation ratios for decimator (or interpolator) designs.

Several interesting observations can be made from these curves and tables. As in the FIR case, the function of the ripples $A(\delta_p/K, \delta_s)$ (see Table I) is a weakly varying function of K and again most of the interesting observations can be made from curves of the $\tilde{S}$ function. From plots of $\tilde{S}$ in Fig. 8 we observe that improvements in efficiency of approximately two or three are possible for moderate values of D (20-50) by using a multistage design and gains of up to about eight are possible for large values of D and small values of Δf. Although these gains are not as striking as those for FIR designs (which can be orders of magnitude), they do represent modest improvements. Another conclusion that can be drawn from the curves in Fig. 8 is that little, if any, improvement in efficiency can be gained by using a three-stage IIR design over a two-stage IIR design and, therefore, two stages, at most, are sufficient for most purposes.

The curves in Fig. 10 represent B_i values for the optimized designs. By noting from (9) that $N_i = AB_i$, it is clear that the B_i values represent the normalized (e.g., N_i/A) filter orders. We can observe from these curves that the order of the final stage of a two- or three-stage design is essentially equal to that of a one-stage design. This occurs because the orders of the IIR filters are determined by the ratio f_p/f_s (the unwarped transition ratio) as opposed to FIR designs whose orders are determined by the difference $f_s - f_p$. As the sampling frequency is reduced it is clear that the transition ratio t_K [see (13a)] of the final stage K remains essentially unchanged (except for a slight warping due to the bilinear transformation) and, therefore, N_K and B_k remain essentially unchanged as K is increased. The filter orders for the earlier stages of a multistage design are of course lower as seen in Fig. 10.

It can also be seen that the sum of the filter orders for a multistage design, i.e., the total required storage, will always be greater than that of a one-stage design. Therefore, a multistage IIR decimator or interpolator is always less efficient, in terms of storage, than a single-stage IIR design. Thus, unlike the FIR design where both computation and storage could be reduced, the multistage IIR design represents a tradeoff between computation and storage.

TABLE I
TABULATION OF $A(\delta_p/K, \delta_s)$

δ_p	δ_s	$K=1$	$K=2$	$K=3$	$K=4$
0.100	0.0100	4.10	4.32	4.45	4.55
0.100	0.0050	4.54	4.77	4.90	4.99
0.100	0.0010	5.57	5.79	5.92	6.01
0.100	0.0005	6.01	6.23	6.36	6.45
0.100	0.0001	7.03	7.26	7.39	7.48
0.050	0.0100	4.32	4.55	4.68	4.77
0.050	0.0050	4.77	4.99	5.12	5.21
0.050	0.0010	5.79	6.01	6.14	6.23
0.050	0.0005	6.23	6.45	6.58	6.67
0.050	0.0001	7.26	7.48	7.61	7.70
0.010	0.0100	4.84	5.06	5.19	5.28
0.010	0.0050	5.28	5.50	5.63	5.72
0.010	0.0010	6.30	6.53	6.65	6.75
0.010	0.0005	6.75	6.97	7.10	7.19
0.010	0.0001	7.77	7.99	8.12	8.21
0.005	0.0100	5.06	5.28	5.41	5.50
0.005	0.0050	5.50	5.72	5.85	5.94
0.005	0.0010	6.53	6.75	6.88	6.97
0.005	0.0005	6.97	7.19	7.32	7.41
0.005	0.0001	7.99	8.21	8.34	8.43
0.001	0.0100	5.57	5.79	5.92	6.01
0.001	0.0050	6.01	6.23	6.36	6.45
0.001	0.0010	7.04	7.26	7.39	7.48
0.001	0.0005	7.48	7.70	7.83	7.92
0.001	0.0001	8.50	8.72	8.85	8.94

An Example

To illustrate the use of the IIR design tables and curves we will choose a decimator with the following specifications:

$D = 100$ $\quad f_{ro} = 1$
$f_p = 0.00475$ $\quad f_s = 0.005$
$\delta_p = 0.001$ $\quad \delta_s = 0.0001$.

These are the same specifications that were used for the 100:1 filter example in [2] and in the 100:1 decimator example in Section III.

[3] See Hastings [9] for the evaluation of $K(\ )$.

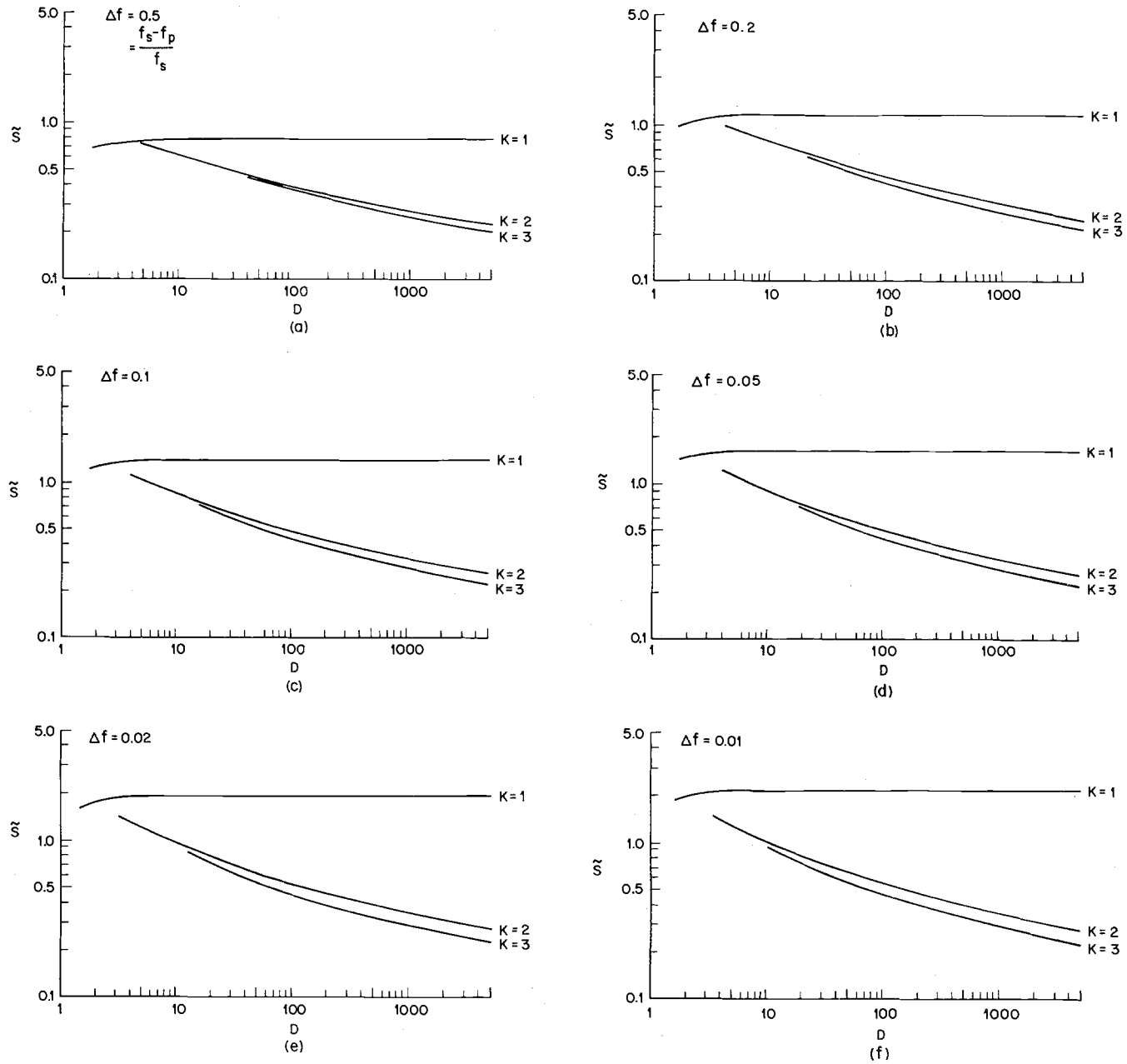


Fig. 8. Plots of minimized values of $\tilde{S}$ as a function of K, D, and Δf where Δf is 0.5, 0.2, 0.1, 0.05, 0.02, and 0.01 for plots (a)-(f), respectively.

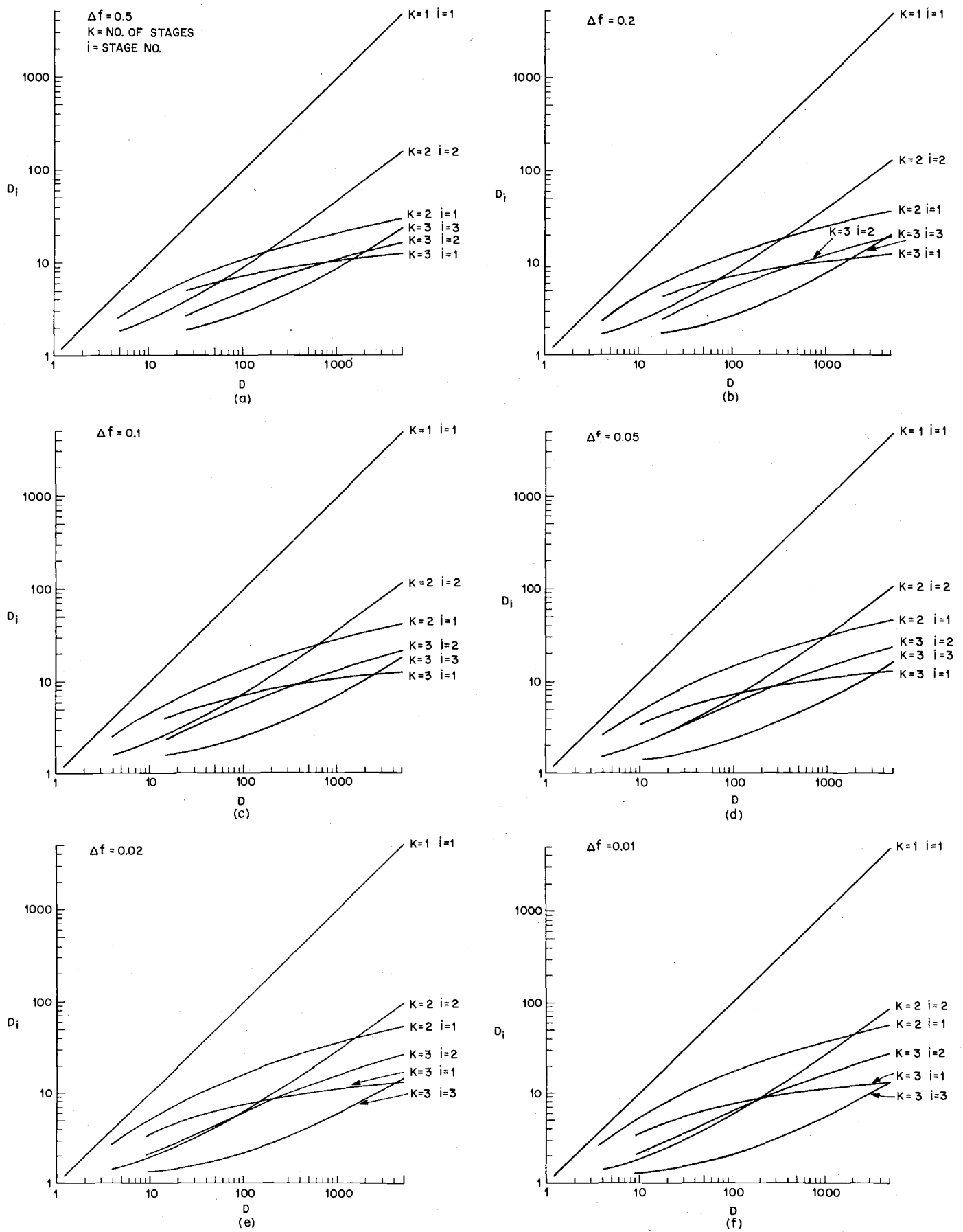


Fig. 9. Plots of optimum decimation ratios for IIR decimator and interpolator designs as a function of K, D, and Δf where Δf is 0.5, 0.2, 0.1, 0.05, 0.02, and 0.01 for plots (a)–(f), respectively.

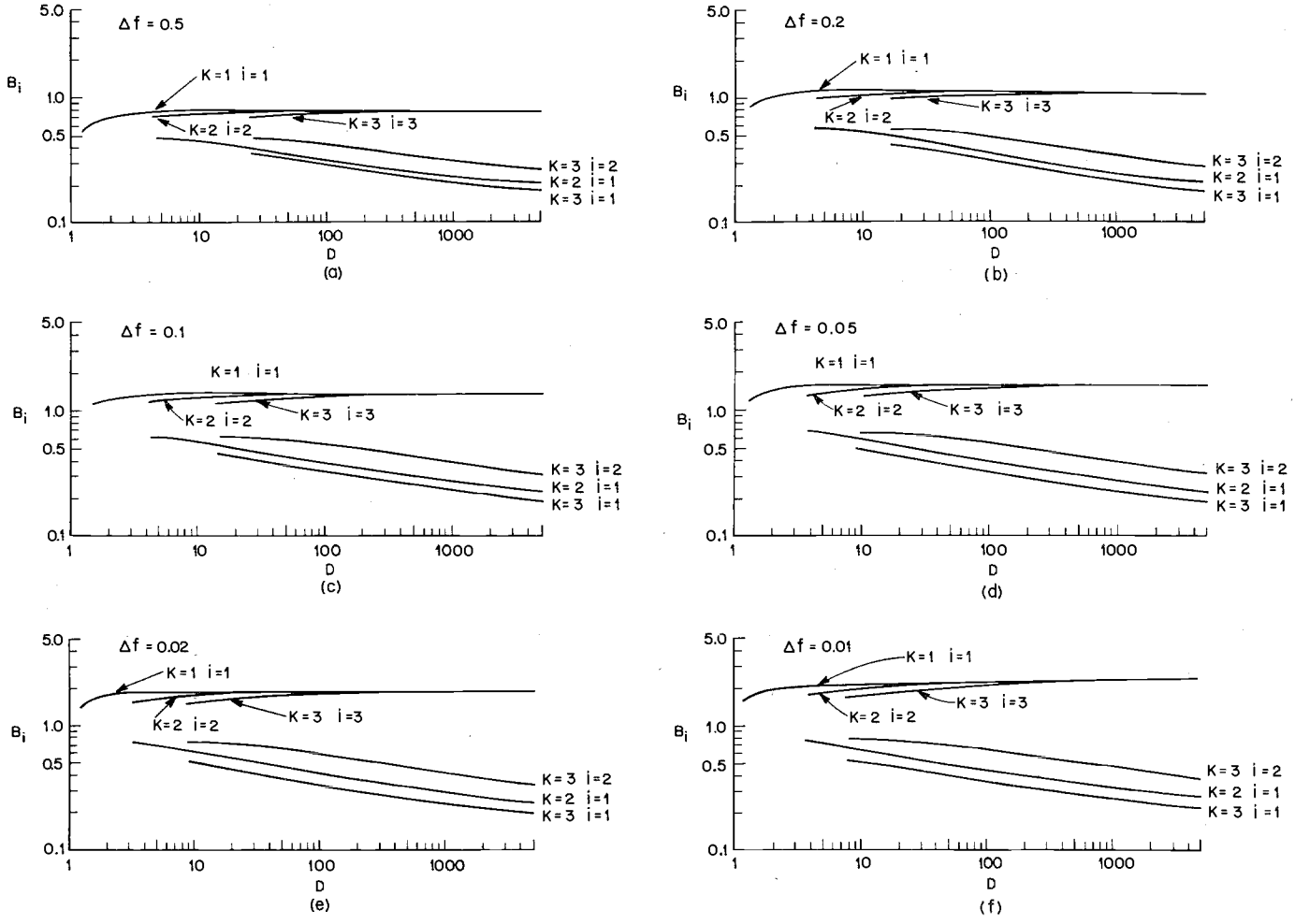


Fig. 10. Plots of B_i values for IIR designs as a function of K, D, and Δf where Δf is 0.5, 0.2, 0.1, 0.05, 0.02, and 0.01 for plots (a)–(f), respectively.

For a one-stage IIR design it can be seen from Table I that $A = 8.50$ and from Fig. 10(d), $B_1 = 1.61$. Therefore the theoretical filter order is 13.7 [from (9)] and actual value of 14 must be used. For a cascade implementation this will require a multiplication rate of $(1 + 14 \times 3/2)f_{ro}$ or 22 mults./s.

For a two-stage IIR design it is seen from Fig. 9(d) that the optimum (theoretical) decimation ratios are $D_1 = 14.5$ and $D_2 = 6.9$. From Table I and Fig. 10(d) we find that $A = 8.72$, $B_1 = 0.39$ and $B_2 = 1.6$ giving theoretical filter orders of $N_1 \cong 3.4$ and $N_2 \cong 13.9$. One practical choice of decimation ratios is $D_1 = 20$ and $D_2 = 5$. This leads to theoretical filter orders of $N_1 = 3.8$ and $N_2 = 13.8$ (these values cannot be obtained from Fig. 10; see instead the tables in [8]). Actual filter orders for this design are, therefore, 4 and 14 for N_1 and N_2, respectively.

Another practical choice of decimation ratios might be $D_1 = 10$ and $D_2 = 10$. This results in theoretical filter orders of $N_1 = 3.1$ and $N_2 = 13.97$ and actual values of 4 and 14, respectively. With careful design, a third-order filter might be substituted for N_1, however.

The results for these three IIR designs for the $D = 100$ decimation are tabulated in Table II. As seen in this table, a good choice for the IIR design might be the two-stage approach with $D_1 = 20$ and $D_2 = 5$. This results in a savings in computation of 2.72 over that of a one-stage design. It is achieved at the cost of four extra storage locations and the added complexity of a two-stage design.

In order to compare the IIR designs to FIR designs a two-stage and a three-stage FIR decimator design are also included in Table II. The design of the two-stage FIR decimator was taken from [2] and the three-stage design is that given in Section III. From this comparison we can observe that the FIR designs are more efficient in terms of computation than the IIR designs, however, they require considerably more storage for both data and coefficients.

VI. Summary

In this paper we have discussed several of the important issues which concern the detailed design and implementation of multistage decimators, interpolators, and narrow-band filters. It was shown that, when using FIR filters, multistage decimators and interpolators which are minimized for storage were essentially minimized in terms of computation rate as well. This is because the minimum on the computation rate was a broad minimum of the objective design function and

TABLE II
COMPARISONS FOR A $D = 100$ DECIMATOR

	One Stage (IIR)	Two Stage (IIR)	Two Stage (IIR)	Two Stage (FIR)	Three Stage (FIR)
Optimum Decimation Ratios	$D_1 = 100$	$D_1 = 14.5$ $D_2 = 6.9$	$D_1 = 14.5$ $D_2 = 6.9$	$D_1 = 39.4$ $D_2 = 2.54$	$D_1 = 16.09$ $D_2 = 4.55$ $D_3 = 1.36$
Actual Decimation Ratios Used	$D_1 = 100$	$D_1 = 20$ $D_2 = 5$	$D_1 = 10$ $D_2 = 10$	$D_1 = 50$ $D_2 = 2$	$D_1 = 10$ $D_2 = 5$ $D_3 = 2$
Theoretical Filter Orders	$N_1 = 13.7$	$N_1 = 3.8$ $N_2 = 13.8$	$N_1 = 3.1$ $N_2 = 13.97$	–	–
Actual Filter Orders Used	$N_1 = 14$	$N_1 = 4$ $N_2 = 14$	$N_1 = 4$ $N_2 = 14$	$N_1 = 423$ $N_2 = 347$	$N_1 = 38$ $N_2 = 38$ $N_3 = 356$
Total Multiplication Rate	22 mults./s	8.1 mults./s	9.2 mults./s	5.98 mults./s	4.06 mults./s
Savings in Rate over One-Stage IIR	1.0	2.72	2.39	3.68	5.42
Total Data Storage	14	18	18	798	436

did not vary significantly as the decimation ratios of the individual stages were varied. Thus the optimum design based on storage coincided with the broad optimum based on computation.

A second issue in the design of optimal decimators and interpolators was the question of minimizing FIR filter order at each stage. It was shown that, for the early stages in a multistage design, one could take advantage of the DON'T CARE frequency bands which lay between each of the relevant frequency bands to reduce the filter order required to meet the given design specifications. It was shown that, for the early stages in the design, reductions in filter order of up to 50 percent were achievable in this manner.

Another issue in the implementation of multistage designs was the question of how to efficiently implement multistage decimators and interpolators in both hardware and software. In Section IV a modular structure was discussed which was particularly suited for both hardware and software implementations. Techniques for pipelining the hardware structure for maximum efficiency were also discussed.

Finally, the question of the suitability of using IIR filters in the implementation of multistage decimators and interpolators was discussed. It was shown that a multistage IIR design is only slightly more efficient computationally than a single-stage IIR design, and that it was always less efficient in terms of storage than the single stage design. In comparing the FIR and IIR implementations of decimators and interpolators, it was shown that the storage required for IIR implementations was considerably less than for FIR implementations; however, the computation rates were comparable. In addition, the FIR designs were linear phase designs whereas the IIR designs were elliptic designs whose phase was highly nonlinear.

In summary, we have tried to discuss several of the issues which affect the design and implementation of multistage decimators and interpolators. We have tried to point out the advantages and disadvantages of each of the alternatives which can be used to design and implement these systems.

REFERENCES

[1] R. E. Crochiere and L. R. Rabiner, "Optimum FIR digital filter implementation for decimation, interpolation, and narrow-band filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 444-456, Oct. 1975.
[2] L. R. Rabiner and R. E. Crochiere, "A novel implementation for narrow-band FIR digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 457-464, Oct. 1975.
[3] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE*, vol. 61, pp. 692-702, June 1973.
[4] D. J. Goodman, "Digital filters for code format conversion," *Electron. Lett.*, vol. 11, Feb. 1975.
[5] D. J. Goodman, to be published.
[6] H. Urkowitz, "Parallel realizations of digital interpolation filters for increasing the sampling rate," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 146-154, Feb. 1975.
[7] R. E. Crochiere and A. V. Oppenheim, "Analysis of linear digital networks," *Proc. IEEE*, vol. 63, Apr. 1975.
[8] 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, pp. 305-331, Feb. 1974.
[9] C. Hastings, Jr., *Approximations for Digital Computers*. Princeton, NJ: Princeton Univ. Press, 1955.