A Novel Implementation for Narrow-Band FIR Digital Filters

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

***Abstract*—In an earlier paper Crochiere and Rabiner [1] discuss the theory of using finite impulse response (FIR) digital filters for signal decimation, interpolation, and filtering. In this paper we expand on the ideas presented in the earlier paper on implementing narrow-band designs efficiently. It is shown how, using the techniques of decimation and interpolation, a desired narrow-band filter can be realized with a *greatly reduced* number of multiplications per second in the realization over standard direct form implementations. Further, it is shown that the proposed implementation can have less roundoff noise and less severe coefficient sensitivity problems than a standard direct form implementation. Several examples are presented to illustrate the applicability of this implementation to practical design problems.**

I. Introduction

ONE of the most difficult problems in digital filtering is the implementation of a narrow-band filter. The difficulty lies in the fact that such narrow-band filters inherently have sharp transitions in their frequency response, thereby requiring high-order designs to meet the desired frequency response specifications. These high-order designs are difficult to implement because of roundoff noise and coefficient sensitivity problems. Furthermore, they require a fairly large amount of computation in their realizations. In this paper we propose a novel implementation for narrow-band digital filters which has the following properties.

Property 1: The computation (in terms of multiplications per second) required to implement the filter is greatly reduced from that required for a standard, direct form implementation for an equivalent finite impulse response (FIR) digital filter.

Property 2: The computation is comparable to that required for optimum (elliptic) infinite impulse response (IIR) filters in a cascade realization.

Property 3: The phase response is linear.

Property 4: The roundoff noise generated in computing the output can be significantly less than for a standard direct form FIR implementation.

Property 5: The coefficient sensitivity problems can be less severe than for standard direct form FIR implementations.

The proposed implementation is based on using the techniques of decimation and interpolation, as discussed by Crochiere and Rabiner [1], to realize a narrow-band filter as a cascade of a decimator and an interpolator. Fig. 1 shows a block diagram of a general purpose system for decimating, or interpolating, a signal $x(n)$. The box labeled L_1 is a sample rate increase box which creates a signal $v(n)$ defined as

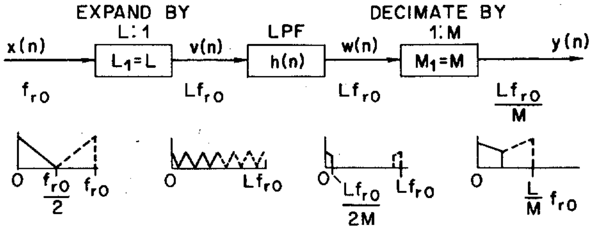

Fig. 1. Block diagram for a general purpose system to change the sampling rate by L/M with L and M integers.

$$
\begin{aligned}
v(n) &= x(n/L) \qquad n = 0, \pm L, \pm 2L, \cdots \\
&= 0 \qquad \text{otherwise,}
\end{aligned}
\tag{1}
$$

i.e., $v(n)$ contains samples of $x(n)$ spaced L samples apart; zero-valued samples being filled in between these samples. The box labeled M_1 is a decimation box which samples the input to the box once every Mth sample, i.e., $y(n)$ is defined as

$$
y(n) = w(nM) \qquad n = 0, \pm 1, \pm 2, \cdots . \tag{2}
$$

The box in the middle of Fig. 1 is the low-pass filter required to prevent aliasing when $w(n)$ is decimated by the factor of M:1 [1].

The general structure of Fig. 1 can be used to change the sampling rate of a signal by the factor L/M. Thus if $M = 1$, the structure acts as an interpolator at the rate of 1:L. If $L = 1$, the structure acts as a sample rate reducer by a factor of M:1. By cascading two structures of the type shown in Fig. 1, the structure of Fig. 2 can be obtained. In this case L_1 is set equal to 1 (for the first stage), and M_2 is set equal to 1 (for the second stage). Thus the structure is a one-stage decimator (with a decimation rate of D:1), followed by a one-stage interpolator (with an interpolation rate of 1:D). It can be seen that the input and output sampling rates are identical in this implementation. Thus the overall structure acts like a low-pass filter in terms of its input–output characteristics, and will be referred to as a one-stage decimation–interpolation filter because it consists of one stage of decimation followed by one stage of interpolation.

In Section II of this paper we will discuss the theory of implementation of the above one-stage decimation–interpolation filter for narrow-band FIR applications and show how it achieves the properties discussed above. Later we show how these results extend to the more general multistage decimation–interpolation filter designs. In Section III several examples of actual realizations are given with numerical comparisons of the efficiencies of various implementations.

Manuscript received January 24, 1975; revised May 16, 1975.

The authors are with Bell Laboratories, Murray Hill, N.J. 07974.

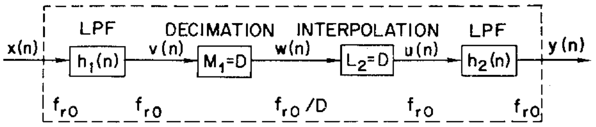

Fig. 2. Block diagram of a decimation–interpolation implementation of a narrow-band low-pass filter.

Finally, in Section IV we provide a general discussion of the properties of the structure, and give some further details on implementation methods.

II. Theory of Implementation of Narrow-Band FIR Designs

It is fairly straightforward to analyze the one-stage decimation-interpolation structure of Fig. 2. If $X(z)$, $H_1(z)$, and $V(z)$ are the z transforms of $x(n)$, $h_1(n)$, and $v(n)$, then

$$V(z) = H_1(z)\, X(z). \tag{3}$$

As shown in [2], $W(z)$, the z transform of $w(n)$; is related to $V(z)$ by the relation

$$W(z) = \frac{1}{D} \sum_{l=0}^{D-1} \cdot V[z^{1/D} \exp(-j2\pi l/D)] \tag{4}$$

and $U(z)$, the z transform of $u(n)$, is related to $W(z)$ by the relation

$$U(z) = W(z^D). \tag{5}$$

Finally $Y(z)$, the z transform of $y(n)$, can be written as

$$Y(z) = U(z)\, H_2(z) \tag{6}$$

where $H_2(z)$ is the z transform of $h_2(n)$. Combining (3)-(6) gives

$$Y(z) = \frac{H_2(z)}{D} \sum_{l=0}^{D-1} X[\exp(-j2\pi l/D)\, z] \cdot H_1[\exp(-j2\pi l/D)\, z]. \tag{7}$$

Evaluating (7) on the unit circle gives

$$Y(e^{j\omega}) = \frac{H_2(e^{j\omega})}{D} \sum_{l=0}^{D-1} X[\exp(j(\omega - 2\pi l/D))] \cdot H_1[\exp(j(\omega - 2\pi l/D))]. \tag{8}$$

Equation (7) can be inverse z transformed to solve for $y(n)$ as

$$y(n) = \frac{h_2(n)}{D} * \left[\sum_{l=0}^{D-1} (x(n) \exp(j2\pi l n/D)) * (h_1(n) \exp(j2\pi l n/D))\right] \tag{9}$$

where $*$ denotes discrete convolution. By performing the inner convolution, (9) can be put in the form

$$y(n) = \frac{h_2(n)}{D} * \left[\sum_{l=0}^{D-1} \sum_{m} x(m) \exp(j2\pi l m/D)\ h_1(n-m) \cdot \exp(j2\pi l(n-m)/D)\, l(n-m)\right] \tag{10}$$

and, by interchanging the summations over l and m, (10) can be written as

$$y(n) = \frac{h_2(n)}{D} * \left[\sum_{m} x(m)\, h_1(n-m) \sum_{l=0}^{D-1} \exp(j2\pi l n/D)\right] \tag{11}$$

$$= \frac{h_2(n)}{D} * \left[\sum_{m} x(m)\, h_1(n-m)\, s(n)\right] \tag{12}$$

where

$$s(n) = \begin{cases} D & n = 0, \pm D, \pm 2D, \cdots \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Equation (12) can be written as

$$y(n) = \frac{h_2(n)}{D} * [[x(n) * h_1(n)] \cdot s(n)]. \tag{14}$$

Thus the overall system can be realized as the convolution of $x(n)$ with $h_1(n)$, followed by a modulation by the impulse train $s(n)$, followed by a convolution with $h_2(n)/D$, as shown in Fig. 3.

If we consider the case where $x(n) = u_0(n)$, i.e., an impulse excitation, then from (14) we get

$$y(n) = \frac{h_2(n)}{D} * [h_1(n)\, s(n)] \tag{15}$$

whereas if $x(n)$ is a delayed impulse, i.e.,

$$x(n) = u_0(n-m) \tag{16}$$

then

$$y(n) = \frac{h_2(n)}{D} * [h_1(n-m)\, s(n)]. \tag{17}$$

It can be seen from (15) and (17) that the response of the system to a delayed impulse is not the same as the system response to an impulse which is delayed by the same amount—i.e., the system is not strictly shift invariant. This same result can be seen from (14) in that the modulator is of the form $s(n)$ and not $s(n-m)$ as would be required for a shift-invariant system. Thus the overall system of Figs. 2 and 3 is not adequately characterized by a simple impulse response.

Based on the above discussion, it is easily shown that there are D-distinct system responses to an impulse delayed by from 0 to $D - 1$ samples. Any further delays serve to repeat one of the set of D responses. These D responses are the result of the decimation stage in which the convolution of $x(n)$ and $h_1(n)$ is sampled once every Dth sample. There are D distinct ways of performing such sampling, depending on which sample of the convolution is chosen as the initial sample.

Since the overall implementation is not characterized by a simple impulse response, the question of whether the overall structure retains the linear phase characteristic of FIR filters is not a simple one to answer. Strictly speaking, the phase response of the system is not exactly linear. However, if we examine (8) carefully, it can be seen that, under the appropriate conditions, the phase response of the system is essentially linear. Equation (8) shows that $Y(e^{j\omega})$ can be written

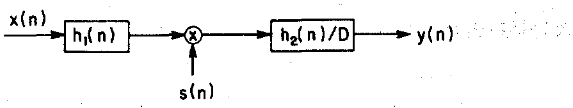

Fig. 3. Modulation structure realization of the system of Fig. 2.

in the form

$$Y(e^{j\omega}) = \frac{H_2(e^{j\omega})}{D} \sum_{l=0}^{D-1} B_l(e^{j\omega}) \tag{19}$$

with

$$B_l(e^{j\omega}) = X[\exp(j(\omega - 2\pi l/D))] \cdot H_1[\exp(j(\omega - 2\pi l/D))]. \tag{20}$$

Fig. 4 shows plots of $B_l(e^{j\omega})$ in the case where $H_1(e^{j\omega})$ is a narrow-band low-pass filter with high stopband loss. The terms $B_l(e^{j\omega})$ can be seen to occupy the frequency regions

$$\frac{2\pi}{D}(l) \leqslant \omega \leqslant \frac{2\pi}{D}(l+1) \tag{21a}$$

$$\frac{2\pi}{D}(D-l-1) \leqslant \omega \leqslant \frac{2\pi}{D}(D-l). \tag{21b}$$

If the interpolation filter passes only the frequency region $0 \leqslant \omega \leqslant 2\pi/D$, and $2\pi((D-1)/D) \leqslant \omega \leqslant 2\pi$, then (19) can be reduced to a single term (the $l = 0$ term), giving

$$Y(e^{j\omega}) \approx \frac{X(e^{j\omega})}{D} H_1(e^{j\omega}) H_2(e^{j\omega}). \tag{22}$$

Since $H_1(e^{j\omega})$ and $H_2(e^{j\omega})$ are FIR linear phase filters, to the extent that the approximations are valid, the overall system is linear phase. In practice, the approximations are valid if $H_1(e^{j\omega})$ and $H_2(e^{j\omega})$ are designed with small stopband tolerances. Finally, it is seen that the resulting filter has the frequency response

$$H(e^{j\omega}) \approx \frac{H_1(e^{j\omega}) H_2(e^{j\omega})}{D}. \tag{23}$$

Equation (23) is not valid exactly since in its derivation it was assumed that the stopband responses of both $H_1(e^{j\omega})$ and $H_2(e^{j\omega})$ were sufficiently small to prevent any aliasing. If aliasing is negligible for the passband of the resulting filter, the tolerance is essentially the sum of the tolerances of $H_1(e^{j\omega})$ and $H_2(e^{j\omega})$, whereas for the stopband, the tolerance is essentially the tolerance of either $H_1(e^{j\omega})$ or $H_2(e^{j\omega})$. Thus if the filter desired has passband tolerance δ_p, and stopband tolerance δ_s, a practical technique for implementing this filter is to design two identical filters ($H_1(e^{j\omega})$ and $H_2(e^{j\omega})$) with passband tolerances $\delta_p/2$, and stopband tolerances δ_s.[1]

In summary, we have just shown how a narrow-band low-pass filter can be realized as a one-stage decimation-interpolation process (i.e., one stage of decimation and one stage of interpolation) for which the decimation and interpolation rates are dependent on the bandwidth of the filter. We now show that such a structure can be made to operate much more efficiently than a standard direct form implementation.

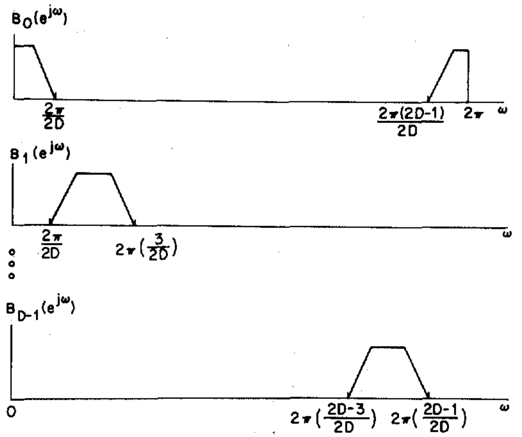

Fig. 4. Frequency domain interpretations of the individual components of the interpolated (but unfiltered) signal.

Assume that each of the low-pass filters $h_1(n)$ and $h_2(n)$ is an N-point linear phase FIR filter. For a direct implementation of a single N-point FIR filter, the number of multiplications per second is[2]

$$N_0 = \frac{N}{2} f_{ro} \tag{24}$$

where f_{ro} is the sampling rate of the input. For the structure of Fig. 2, the computation of $w(n)$ from $x(n)$ requires

$$N_D = \frac{N}{2} f_{ro} \cdot \frac{1}{D} \tag{25}$$

multiplications per second, whereas the computation of $y(n)$ from $u(n)$ requires

$$N_I = N f_{ro} \cdot \frac{1}{D} \tag{26}$$

multiplications per second. Thus the overall number of multiplications per second for the structure of Fig. 2 is

$$N_1 = N_D + N_I = \frac{3N}{2} f_{ro} \cdot \frac{1}{D}. \tag{27}$$

Thus the ratio of multiplications per second for the two implementations is

$$\frac{N_0}{N_1} = \frac{N/2 f_{ro}}{3N/2 f_{ro} \cdot 1/D} = \frac{D}{3}. \tag{28}$$

Equation (28) shows that if $D > 3$ the new structure is more efficient than the standard direct form structure.

Until now we have only considered the simple decimation-interpolation realization of Fig. 2. However Crochiere and Rabiner [1] have shown that a decimation (or an interpolation) factor of D can often be more efficiently realized in a multistage process, rather than a single-stage process as shown in Fig. 2. Fig. 5 shows a general K-stage integer decimator followed by a K-stage integer interpolator. The resulting

[1] Note that from (23), in order for the overall gain of the filter to be unity in the passband, the combined passband gain of $H_1(e^{j\omega}) H_2(e^{j\omega})$ must be equal to D.

[2] We are assuming N is even until Section III for simplicity. When N is odd we replace $N/2$ by $(N+1)/2$ in the relevant equations.

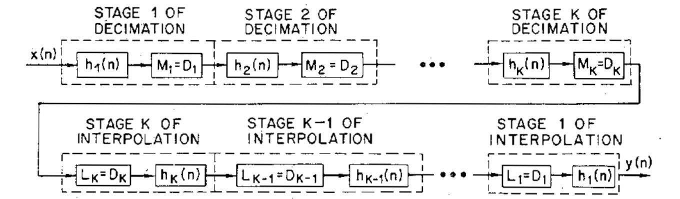

Fig. 5. Block diagram of a K-stage decimation–interpolation realization of a narrow-band low-pass filter.

multistage structure can be more efficient than the single-stage structure by as much as 500:1 for large values of D.

By a procedure similar to that above we can show that these multistage designs have linear phase, within the constraints of aliasing, and that they can be approximately characterized by the system function

$$\begin{aligned}H(e^{j\omega}) \approx\; & H_1(e^{j\omega})\, H_2\,[\exp(j\omega/D_1)] \\ & \cdots H_K\,[\exp(j\omega/D_1 D_2 \cdots D_{K-1})] \\ & \cdot H_K\,[\exp(j\omega/D_1 D_2 \cdots D_{K-1})] \\ & \cdots H_2\,[\exp(j\omega/D_1)]\, H_1(e^{j\omega}). \qquad (29)\end{aligned}$$

Because of aliasing they are also, strictly speaking, shift-variant systems and have a total of D distinct impulse responses, where $D = D_1 D_2 \cdots D_K$.

Since these multistage structures are thoroughly discussed by Crochiere and Rabiner [1], we will not say anything more about them here. Instead in the next section we present several examples illustrating the use of the proposed implementation for realizing narrow-band low-pass filters.

III. Filter Examples

To illustrate the application of the techniques discussed in Section II, we present three design examples for narrow-band filters.

Example 1

The first example (given only for illustrative purposes) is the design of a low-pass filter with specifications

$$f_p = \tfrac{1}{24} = 0.041666, \qquad \delta_p = 0.31$$

$$f_s = \tfrac{1}{12} = 0.083333, \qquad \delta_s = 0.155$$

$$f_{ro} = 1.$$

The decimation rate, D, for this filter is

$$D = \frac{f_{ro}}{2f_s} = \frac{1}{2 \cdot 1/12} = 6.$$

A direct form implementation (no decimation or interpolation) requires a value of N_0 of 11 as determined using the design program of McClellan *et al.* [3]. This implementation requires $(N_0 + 1)/2$ or 6 multiplications/s. When the desired filter is implemented as shown in Fig. 2 (using $D = 6$), two identical filters, $h_1(n)$ and $h_2(n)$,[3] are required with identical specifications to those listed above except the passband tolerance, δ_p, is halved for each filter. Using the design program, a length $N_1 = 17$ filter was determined adequate to meet the specifications. For this implementation a total of $(N_1 + 1)/(2 * D)$ or 3/2 multiplications/s are required to decimate the input to a rate f_{ro}/D, while a total of N_1/D or 17/6 multiplications/s are required to interpolate the signal back to the original rate. Thus the total number of multiplications per second is $4\frac{1}{3}$ for this one-stage implementation, as opposed to 6 for the direct implementation. Finally, if the filter is implemented in a multistage arrangement, as in Fig. 5, then using the design tables of Crochiere and Rabiner [1], two stages of decimation followed by two stages of interpolation yields a somewhat lower multiplication rate than the single stage. For the two-stage implementation, the optimum decimation factors are 3.3 for the first stage and 1.8 for the second stage. Since integer decimation factors are most convenient, and since Crochiere and Rabiner [1] have shown that the individual decimation factors can be varied over a fairly large range without strongly affecting the overall multiplication rate [1], the values $D_1 = 3$ and $D_2 = 2$ were chosen for the two-stage implementation. For the first stage, the filter specifications were

$$f_{p1} = \tfrac{1}{24} = 0.041666, \qquad \delta_{p1} = \frac{\delta_p}{4} = 0.0775$$

$$f_{s1} = \frac{f_{ro}}{D_1} - f_s = \tfrac{1}{3} - \tfrac{1}{12} = 0.25, \qquad \delta_{s1} = \delta_s = 0.155$$

$$f_{ro} = 1$$

$$D_1 = 3.$$

These specifications required a filter of length $N_{21} = 4$. For the second stage, the filter specifications were

$$f_{p2} = \tfrac{1}{24} = 0.041666, \qquad \delta_{p2} = \frac{\delta_p}{4} = 0.0775$$

$$f_{s2} = \tfrac{1}{12} = 0.083333, \qquad \delta_{s2} = \delta_s = 0.155$$

$$f_{r1} = \frac{f_{ro}}{3} = 0.3333$$

$$D_2 = 2.$$

These specifications required a filter of length $N_{22} = 8$. The total multiplication rate for implementing the filter is therefore

$R_{1D} = \frac{2}{3}$	first decimation stage
$R_{2D} = \frac{4}{6}$	second decimation stage
$R_{2I} = \frac{8}{6}$	first interpolation stage
$R_{1I} = \frac{4}{3}$	second interpolation stage
$R_T = \frac{24}{6} = 4$ multiplications/s.	

Thus the total overall multiplications rate, R_T, for the two-stage implementation is only slightly smaller than for the one-stage implementation for this example. However, the two-stage multiplication rate can often be significantly smaller than the one-stage multiplication rate, as will be seen in a later example.

[3] A scaling factor of $D = 6$ is required in the implementation of $h_2(n)$ to preserve the overall signal level in the filter [2].

Fig. 6 shows a plot of the log-magnitude frequency response of the one-stage implementation. Due to the aliasing, the frequency response is not equiripple; however, the original filter specifications are still met, or exceeded, at all frequencies.

Fig. 7 shows a plot of the log-magnitude frequency response of the two-stage implementation. In this case the frequency response shows more variation than the one-stage case since the ripples from $H_1(e^{j\omega})$ do not line up with the ripples from $H_2(\exp j(\omega/D_1))$ which was then filtered by $H_1(e^{j\omega})$.

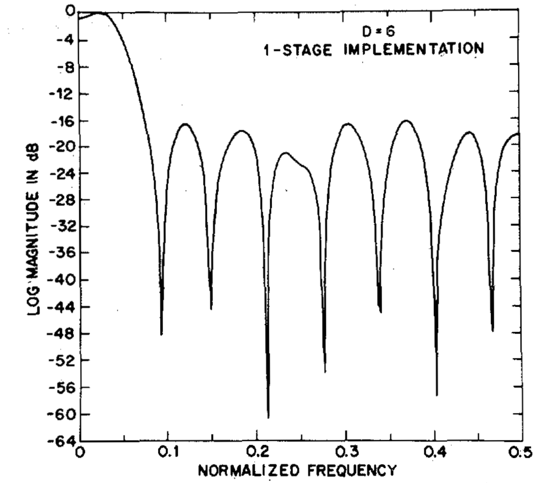

Fig. 6. Log-magnitude frequency response for a one-stage implementation of a $D = 6$ filter.

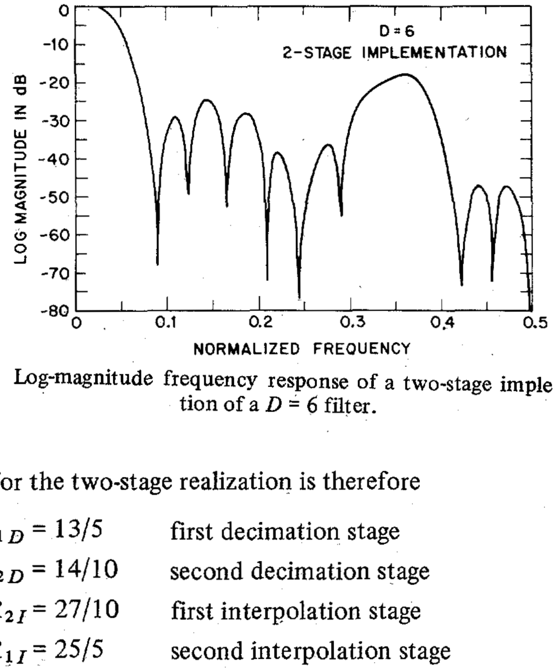

Fig. 7. Log-magnitude frequency response of a two-stage implementation of a $D = 6$ filter.

Example 2

As a more realistic example we consider the implementation of a narrow-band low-pass filter with specifications

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

For the straight direct form implementation, the filter specifications are met by an FIR filter with $N_0 = 110$–i.e., 55 multiplications/s are required. For one stage of decimation and one stage of interpolation, a value of $N_1 = 121$ is required (the specifications remain the same except δ_p is halved). Thus the total multiplication rate for this one-stage implementation is

$R_{1D} = 6.1$	decimation stage
$R_{1I} = 12.1$	interpolation stage
$R_T = 18.2$	multiplications/s.

For a two-stage implementation, the optimum decimation ratios are $D_1 = 4.719$ for the first stage, and $D_2 = 2.119$ for the second stage. A convenient choice of integers for D_1 and D_2 are $D_1 = 5$ and $D_2 = 2$. The specifications for the low-pass filter for the first stage are

$$f_{p1} = 0.025, \quad \delta_{p1} = \frac{\delta_p}{4} = 0.0025$$
$$f_{s1} = 0.15, \quad \delta_{s1} = \delta_s = 0.001$$
$$f_{ro} = 1$$
$$D_1 = 5.$$

The impulse response duration of a linear phase FIR filter required to meet these specifications is $N_{21} = 25$. The specifications for the low-pass filter for the second stage are

$$f_{p2} = 0.025, \quad \delta_{p1} = \frac{\delta_p}{4} = 0.0025$$
$$f_{s2} = 0.05, \quad \delta_{s1} = \delta_s = 0.001$$
$$f_{r1} = \frac{f_{ro}}{D_1} = 0.2$$
$$D_2 = 2.$$

The duration of a linear phase FIR filter required to meet these specifications is $N_{22} = 27$. The total multiplication rate for the two-stage realization is therefore

$R_{1D} = 13/5$	first decimation stage
$R_{2D} = 14/10$	second decimation stage
$R_{2I} = 27/10$	first interpolation stage
$R_{1I} = 25/5$	second interpolation stage
$R_T = 11.7$	multiplications/s.

Table I summarizes the results on the efficiencies of realizing the narrow-band design of Example 2. It is seen that the two-stage implementation is almost five times as efficient (in terms of multiplication rate) as the direct form realization. Further, it is seen that the coefficient storage for the two-stage implementation is about half that of the direct form implementation since the filter orders are significantly lower for the two-stage implementation than the direct form.

Fig. 8 shows plots of the impulse responses of the one- [Fig. 8(a)] and two-stage implementations [Fig. 8(c)], as well as the log-magnitude frequency responses of the impulse responses from the one- [Fig. 8(b)] and two-stage [Fig. 8(d)] designs. The impulse response envelopes for the one- and two-stage designs are markedly similar (as expected); whereas the log-magnitude responses of the individual impulse responses differ because the aliasing from a one-stage design is different from the aliasing of a two-stage design.

TABLE I
COMPARISONS FOR D = 10 LOW-PASS FILTER

	Filter Lengths	Multiplication Rate	Efficiency	Coefficient Storage
Direct form realization	$N_0 = 110$	55	1:1	55 locations
One-stage decimation-interpolation	$N_1 = 121$	18.4	3.0:1	61 locations
Two-stages decimation-interpolation	$N_{21} = 25$ $N_{22} = 27$	11.7	4.7:1	27 locations

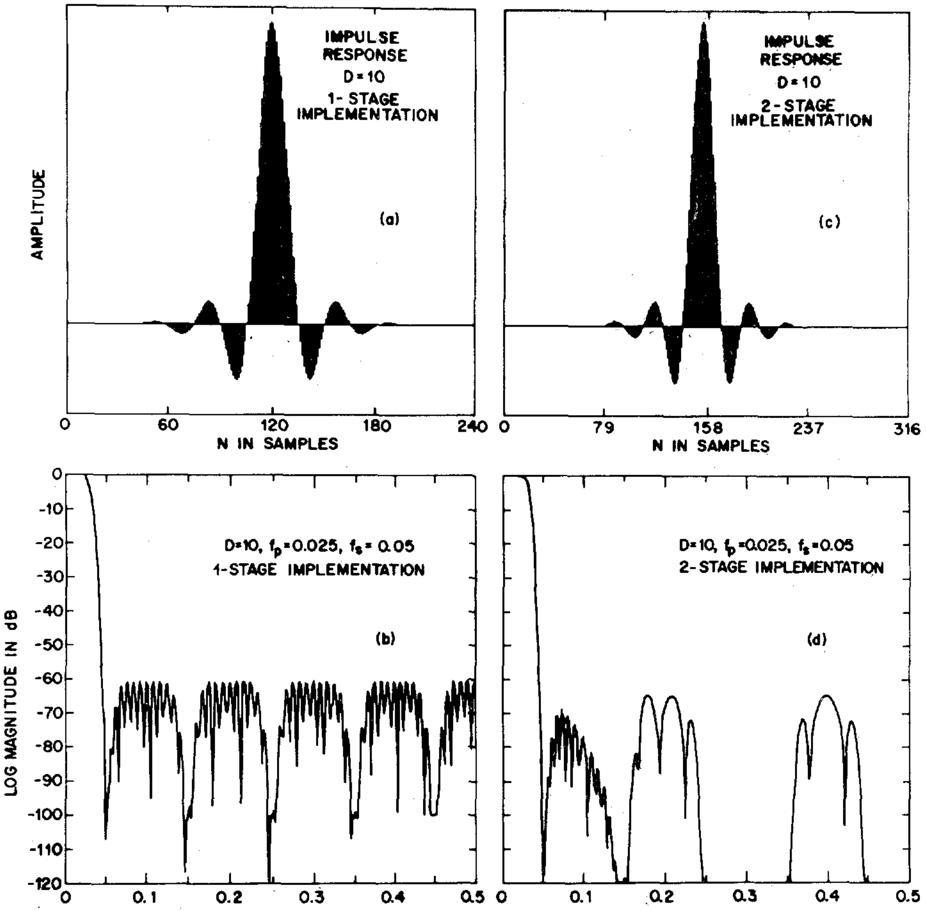

Fig. 8. Impulse responses and log-magnitude frequency responses for a single- and a two-stage implementation for Example 2.

Example 3

As a final example we consider the design of a very narrowband low-pass filter (as is generally required in many speech processing systems) [4] with specifications

$$f_p = 0.00475, \quad \delta_p = 0.001$$

$$f_s = 0.005, \quad \delta_s = 0.0001$$

$$f_{ro} = 1$$

$$D = 100.$$

These specifications are those for a sharp cutoff (the transition width is $\Delta F = 0.00025$) with tight tolerances in both the passband and stopband, as is often required in actual applications.

For the direct form implementation an optimal linear phase FIR filter could not actually be designed to meet the above specifications because of numerical accuracy considerations in designing very high-order filters. Thus using the design formulas of Herrmann *et al.* [5] an estimate of the required filter length was computed, giving N_0 = 15 590–i.e., an extremely high-order filter is required to meet these tight specifications. In reality it is not reasonable to implement such a high-order filter directly because of roundoff noise and coefficient sensitivity problems. However we will now show that when a multistage decimation-interpolation realization is used, the required filter orders are substantially lower, and in fact such filters can easily be designed and implemented. The

TABLE II
COMPARISONS FOR $D = 100$ LOW-PASS FILTER

	Straight FIR Filter	One-Stage Dec. One-Stage Interp.	Two-Stage Dec. Two-Stage Interp.	Two-Stage Dec. Two-Stage Interp.	Three-Stage Dec. Three-Stage Interp.	Three-Stage Dec. Three-Stage Interp.
Optimum dec. ratios	–	$D = 100$	$D_1 = 39.428$ $D_2 = 2.536$	$D_1 = 39.428$ $D_2 = 2.536$	$D_1 = 16.094$ $D_2 = 4.554$ $D_3 = 1.364$	$D_1 = 16.094$ $D_2 = 4.554$ $D_3 = 1.364$
Actual dec. ratios used	–	$D = 100$	$D_1 = 40$ $D_2 = 2.5 = 5/2$	$D_1 = 50$ $D_2 = 2$	$D_1 = 15$ $D_2 = 5$ $D_3 = 4/3$	$D_1 = 10$ $D_2 = 5$ $D_3 = 2$
Filter orders	15 590	16 466	$N_1 = 284$ $N_2 = 866$	$N_1 = 423$ $N_2 = 347$	$N_1 = 78$ $N_2 = 83$ $N_3 = 712$	$N_1 = 50$ $N_2 = 44$ $N_3 = 356$
Total mult. rate using symmetry when possible	7795 mults/s	247 mults/s	19.3 mults/s	17.9 mults/s	14.23 mults/s	14.05 mults/s
Savings in the rate over the straight filter (Ratio)	–	31.6	404	435	548	554
Total number of stored coefficients	7795	8233	575	385	437	255

Dec. = decimation $f_p = 0.00475$ $\delta_p = 0.001$ $f_{ro} = 1$
Interp. = interpolation $f_s = 0.005$ $\delta_s = 0.0001$ $D = 100$

direct form realization would require approximately 7795 multiplications/s.

For a one-stage implementation, the estimated filter order is $N_1 = 16\,466$ since all specifications are the same except the required passband ripple is halved. The total multiplication rate for the one-stage design is $R_T \approx 247$ multiplications/s or a savings of a factor of 31.6 over direct form implementation.

For a two-stage implementation, the optimum decimation ratios are $D_1 = 39.4$ and $D_2 = 2.5$. A convenient choice of integer decimation ratios is $D_1 = 50$ and $D_2 = 2$. (It is possible to consider the choice $D_1 = 40$, $D_2 = 2.5$ but since noninteger ratios sacrifice the halving of the multiplication rate during the decimation stages, they generally are less efficient than integer ratios.) For this choice of decimation ratios, the specifications for the first low-pass filter are

$$f_{p1} = f_p = 0.00475 \qquad \delta_{p1} = \frac{\delta_p}{4} = 0.00025$$

$$f_{s1} = \frac{f_{ro}}{D_1} - f_s = 0.015 \qquad \delta_{s1} = \delta_s = 0.0001$$

$$f_{ro} = 1.0$$

$$D_1 = 50.$$

The estimated filter length to meet these specifications is $N_{21} = 423$. The specifications for the second low-pass filter are

$$f_{p2} = f_p = 0.00475 \qquad \delta_{p2} = \frac{\delta_p}{4} = 0.00025$$

$$f_{s2} = f_s = 0.005 \qquad \delta_{s2} = \delta_s = 0.0001$$

$$f_{r1} = 0.02$$

$$D_2 = 2.$$

The estimated filter length to meet these specifications is $N_{22} = 347$. The total multiplication rate for this two-stage implementation is $R_T = 17.91$ multiplications/s—a savings of a factor of 435.2 over direct form, and 13.8 over the one-stage implementation. Furthermore, the filters required can readily be designed and implemented as they are within the range of the FIR design algorithms.

The total multiplication rate for realizing this narrow-band filter can be reduced even further by using a three-stage implementation. Rather than giving the details here, Table II gives a summary of the required filter orders, decimation ratios, and multiplication rates for several implementations of the narrow-band design of Example 3. For a three-stage implementation, the total multiplication rate can be reduced to 14.05 multiplications/s. Using the same specifications, an elliptic filter was designed and implemented in cascade form. A 14th-order elliptic filter was required to meet the filter specifications. The cascade form realization of this filter required 22 multiplications/s. Thus the three-stage implementation is about *50 percent more efficient* than a cascade realization of an elliptic filter meeting the identical specifications. Furthermore, the three-stage realization is essentially a linear phase design, whereas the phase (or group delay) of the elliptic filter is highly nonlinear.

IV. DISCUSSION

In the preceding sections we have shown that a narrow-band low-pass filter can be realized using the processes of decimation and interpolation. The advantages of the proposed realization are primarily the following:

1) reduced total multiplication rate;
2) lower order filters required in implementing the design (for two or more stages);
3) linear phase;
4) lower roundoff noise (for two or more stages); and
5) lower coefficient sensitivity (for two or more stages).

We have already discussed and illustrated with examples the first three properties of the implementation. Properties 4 and

5 follow directly from Property 2. The lower the filter orders (with the concomitant relaxed filter specifications obtained inherently by decimating the interpolating in stages), the less roundoff noise in the realization (since roundoff noise is directly proportional to filter order), and the lower the coefficient sensitivity (since this is also directly related to filter order and tightness of specifications–i.e., the tolerances).

One disadvantage of the proposed implementation over a direct form FIR realization is that the system is not strictly time or shift invariant. However, we have shown that if the aliasing can be neglected (as is generally true for most designs) then the system is effectively a linear phase, linear, time-invariant system.

An important issue in implementing a digital filter by the methods discussed in this paper is what values of stopband attenuation are required to make the aliasing sufficiently negligible so that the results are usable. Unfortunately, there is no simple answer to this question because the minimum stopband attenuation is data dependent. Requirements for a speech processing system need not be the same as those for a picture processing system, etc. It should be emphasized, however, that the potential gains in speed are sufficiently large that one can tolerate making the stopband attenuation small enough to "guarantee" that the effects of aliasing are made negligible.

Although we have concentrated on minimizing the total computation in the implementation of the filter, another consideration in a multistage implementation of an FIR digital filter is the amount of storage required. Work in this area is currently under investigation.

Finally, it is possible to consider a mixed filter structure in which both FIR and IIR stages are used. The considerations and tradeoffs involved in the implementation of such a structure are currently being studied.

References

[1] R. E. Crochiere and L. R. Rabiner, "Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering," this issue, pp. 444–456.

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

[3] J. H. McClellan, T. W. Parks, and L. R. Rabiner, "A computer program for designing optimum FIR linear phase digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 506–526, Dec. 1973.

[4] R. W. Schafer and L. R. Rabiner, "Digital representations of speech waveforms," *Proc. IEEE (Special Issue on Digital Signal Processing)*, vol. 63, pp. 662–677, Apr. 1975.

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

Heuristic Optimization of the Cascade Realization of Fixed-Point Digital Filters

BEDE LIU, FELLOW, IEEE, AND ABRAHAM PELED, MEMBER, IEEE

Abstract—**In the cascade realization of fixed-point digital filters under dynamic range constraints, the output noise due to accumulation of roundoff errors is highly dependent upon the order of the sections. For recursive filters it also depends on the pole–zero pairing that forms the individual second-order sections. The output noise may vary over several orders of magnitude for different cascade realizations of high-order filters. Therefore an optimization procedure to find a good ordering and pairing is very desirable.**

We propose a heuristic optimization procedure for finding a "near optimal" solution. The procedure is completely automatic and does not require any knowledgeable judgment. The number of function evaluations required for a filter of N-cascaded sections is proportional to N^2. By using this procedure, "near optimal" solutions have been found for a 22nd-order recursive filter in 23 s, and for a 55th-order nonrecursive filter in 37.5 s, on an IBM 360-91 computer.

Manuscript received May 14, 1974; revised December 30, 1974 and March 17, 1975. This work was supported by the Air Force Office of Scientific Research, United States Air Force, under Grant AF-AFOSR 71-2101.

The authors are with the Department of Electrical Engineering, Princeton University, Princeton, N.J. 08540.

I. Introduction

The realization of digital filters by cascading second-order sections has many desirable features, such as better noise performance than the direct realization [1] and permitting a modular realization of high-order digital filter in a flexible manner.

When a fixed-point digital filter is realized by cascading its second-order sections under dynamic range constraints, the resulting roundoff error due to the use of finite word-length is highly dependent upon the pole–zero pairing and ordering of the sections. In this paper, we shall use the term assignment to denote a specific pole–zero pairing and a specific ordering. Jackson [2] has derived expressions for the roundoff error for such filters and has also shown that wide variations in the output noise can result from different assignments.

Based upon his extensive experimental analysis of several filters, Jackson [2] has proposed rules for determining good assignments. Lee [3] has suggested an optimization procedure