THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 59, No. 9, November 1980
Printed in U.S.A.

Finite Precision Design of Linear-Phase FIR Filters

By V. B. LAWRENCE and A. C. SALAZAR

(Manuscript received April 14, 1980)

The FIR (finite impulse response) filter is an essential tool for a large number of applications in communication. In this paper we consider the design of linear-phase FIR digital filters with finitely precise coefficients. Coefficient inaccuracy is known to degrade the frequency response of band-select FIR filters, especially in the stopband region. We derive a bound on the attainable stopband attenuation, and we also develop techniques for designing FIR filters with finitely precise coefficients. Mixed-integer programming algorithms are presented to select finitely precise coefficients for a filter that best approximates an arbitrary magnitude characteristic in the minimax sense. Our method generates a number of possible solutions including that of simple rounding or truncation and then selects the best finitely precise coefficients from this set. In this way, significant improvement in the filter performance is gained over methods that simply round or truncate the infinitely precise coefficients. We also show how integer programming can be used to design filters with powers of two coefficients. Such filters are easier to mechanize since they do not require multipliers.

I. INTRODUCTION AND SUMMARY

In this paper, we develop techniques for designing linear-phase finite impulse response (FIR) digital filters with finitely precise coefficients. The FIR filter has wide applications in communications.[1] For example, FIR filters have been used as band-select filters in FDM/TDM translators[2] and in *Touch-Tone*® receivers.[3] They have been used as adaptive equalizers,[4] as matched filters in radar, and as echo cancellers in satellite communications.[5] In addition, they are widely used in speech synthesis and analysis.[6] Many techniques are available for the design of infinite-precision FIR filters. Such techniques include (i) the use of

windows,[7] (*ii*) the linear programming approach,[8-10] (*iii*) the Remez exchange algorithm by Parks and McClellan,[11] (*iv*) the interpolation techniques by Hofstetter et al,[12] (*v*) Hamming-Kaiser twicing algorithms[13] and (*vi*) the nonlinear optimization procedure by Herrmann.[14] The above techniques are appropriate for filters with infinitely precise coefficients, i.e., for sampled data systems. If a sufficient number of such coefficients are used, FIR filters can be designed to approximate virtually any frequency response as closely as desired.

FIR filters are commonly implemented either as charge-coupled devices (CCDs), as surface acoustic wave (SAW) devices, or as digital filters. The first two allow the realization of FIR filters without the need for analog-to-digital conversion. However, there are fundamental limitations on the attainable coefficient accuracy for CCDs and SAW devices.[15] Current CCD technology can only mechanize FIR filters with coefficient accuracies of 10 bits. Finite-precision CCD filters have been designed using integer programming techniques.[16] In digital filtering,* FIR filters, as depicted in Fig. 1, and other digital signal-processing algorithms are implemented either with special-purpose hardware or as programs in digital computers or signal processors. In either case, the data sequence values $\{x(n)\}$ and coefficients $\{h(n)\}$ are usually stored in finite-length registers or memory elements. Register length is an important economic factor in hardware implementations. Recently, innovations in hardware have emphasized the importance of efficiently designing digital filters with finite register length coefficients. Of particular interests are the advances in microprocessors,[17] bit-slice technology,[18] and programmable digital signal processors.[19,28]

In this paper, we discuss only one effect of finite register length in digital filters, i.e., quantization of the coefficients. Usually the coefficients of a digital filter are obtained by some theoretical or optimization design procedures that essentially assume an infinitely precise representation of the filter coefficients. As a consequence, the frequency response of the quantized (or digital) filter can deviate appreciably from the filter designed with infinitely precise coefficients. In fact, the quantized filter may in certain cases fail to meet initial specifications even though the unquantized filter does. As an example, consider the low-pass filter specification shown in Fig. 2. The desired filter should have a stopband attenuation of 80 dB, a ripple of 0.01 dB in the passband, and a transition ratio of 0.6. An optimum filter of length 49 designed with infinite precision coefficients using linear programming techniques had a stopband attenuation of 98.6 dB and a passband ripple of 0.002dB (see curve A in Fig. 3). However, rounding the filter coefficients to 12 bits resulted in a minimum stopband attenuation of

* The coefficient precision of the FIR filter can be made as accurately as required at the price of increasing hardware cost.

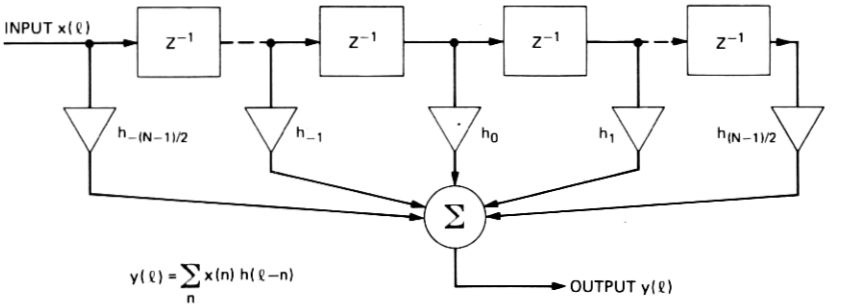

Fig. 1—FIR filter.

62 dB. This indicated a degradation in the stopband of approximately 32 dB. This result is shown as curve B in Fig. 3. We see from this example that coefficient inaccuracy can degrade the frequency response of band-select filters, especially in the band-reject region. The number of bits would have to be greater than 12 bits to meet the original specification.

A useful way of finding the appropriate number of bits is to use formulas that estimate or bound[20–22] the error magnitude in the frequency domain. These bounds are functions of the number of bits and length of the filter. In Section II we derive an upper bound on the error magnitude in the frequency domain. For the filter example given in Fig. 2, known existing bounds and our new bound indicate that 14 to 17 bits of coefficient wordlength would be required to achieve a stopband attenuation of 80 dB. These bounds only serve as a guide for

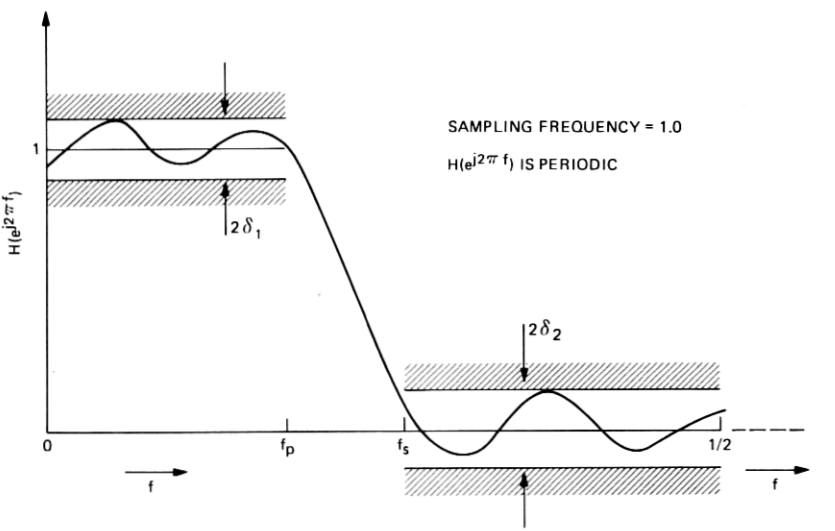

Fig. 2—LPF specification.

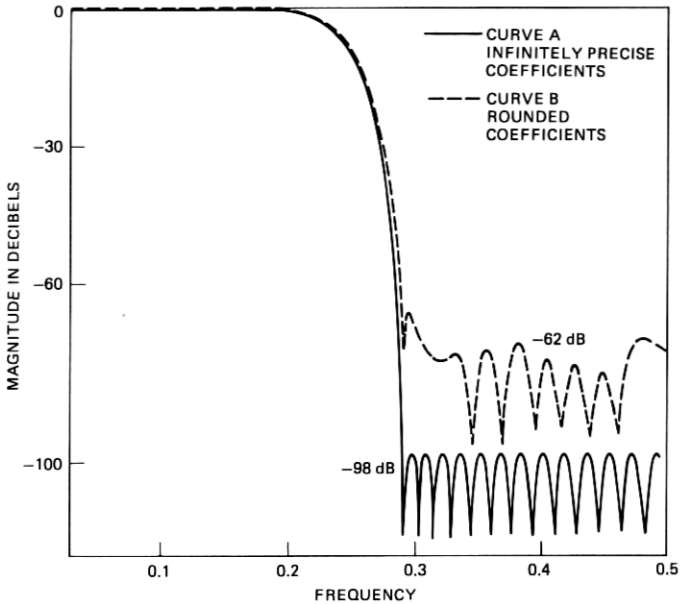

Fig. 3—Spectra of filter with infinitely precise and rounded coefficients.

the design of band-select filters. In an actual implementation, the frequency response is evaluated for the designed filter using different coefficient wordlengths to determine the least number of bits needed to meet the desired specification. The coefficient wordlengths used in a trial are in the proximity of the wordlength determined by the bounds. Though these bounds only serve as guides for the design of fixed FIR filters, they are extremely useful in establishing the coefficient wordlength requirements for adaptive FIR filters,[24] because for adaptive filters it is not possible to know a priori what the exact desired characteristics are.

For the filter example discussed earlier, an improvement of 2 dB in the stopband attenuation is achieved by using selected rounding (i.e., a mixture of rounding, truncation, or boosting) rather than rounding off the coefficients to 12 bits. Other quantization schemes such as pure truncation in either two's complement or sign-magnitude format similarly degraded filter performance (see Table I for comparison of the various rounding schemes). We display in Table II the filter coefficients for the infinitely precise filter, rounded filter, truncated filter, and a random-rounded filter where we show only half of the coefficients starting from the center coefficient. In Table III we find that truncation can produce a better result than rounding.

We develop, in Section III, techniques that include the effects of coefficient quantization in the design of FIR filters. We use mixed-

Table I—Rounded and truncated stopband ripples

	Infinite Precision	Bounding	Truncation 2'sC	Truncation S/M	Selected Rounding
Passband δ_1 in dB	0.0	0.01	0.01	0.01	0.005
Stopband δ_2 in dB	98	62	57	58	65

integer linear programming algorithms to select finitely precise coefficients for a FIR filter that approximates a given magnitude characteristic. These mixed integer routines allow the unknown coefficients to take on integer values while the stopband and passband ripples are allowed to be noninteger. The approximation is optimal in the given design sense, i.e., minimum absolute-weighted value of the error in the filter response. The errors in the approximation cannot be both equiripple and minimax, as in the case of infinite precision designs using linear programming, because of the finite precision wordlength constraints. However, other desired properties of FIR filters such as linear phase can still be preserved by constraining the resultant finitely precise coefficients to be even and symmetric.

The two mixed-integer programming optimization methods we use are the well-known branch and bound algorithm[23] and the zero-one algorithm[26] using decomposition methods. We introduce several con-

Table II—Coefficient values

(Length of Filter = 49, Sampling Frequency = 1.0,
Passband Frequency = 0.16875, Stopband Frequency = 0.28125)

Infinite Precision	Rounding	Truncation 2'C	Truncation S/M	Selected Rounding
1826.28288	1826	1826	1826	1826
1276.55037	1277	1276	1276	1277
211.53000	212	211	211	212
−358.04334	−358	−359	−358	−358
−184.68103	−185	−185	−184	−185
146.50082	147	146	146	147
146.54836	147	146	146	147
−50.25278	−50	−51	−50	−51
−104.77742	−105	−105	−104	−105
3.25052	3	3	3	4
66.47289	66	66	66	67
15.85331	16	15	15	15
−36.39718	−36	−37	−36	−37
−19.16569	−19	−20	−19	−19
16.23998	16	16	16	17
15.11741	15	15	15	15
−5.01737	−5	−6	−5	−6
−9.24632	−9	−10	−9	−9
0.20740	0	0	0	1
4.45473	4	4	4	5
0.98694	1	0	0	1
−1.59912	−2	−2	−1	−2
−0.73767	−1	−1	0	−1
0.35331	0	0	0	0
0.28114	0	0	0	0

Table III—Stopband ripples for rounding and truncation

(Length of Filter = 33, Sampling Frequency = 1.0, Passband Frequency = 0.15, Stopband Frequency = 0.3, Infinitely Precise Coefficients Stopband Ripple = −79.12 dB)

No. of bits	Rounding (dB)	Truncation (dB)
12	−62.4	−62.7
10	−49.4	−47.0
8	−38.5	−38.9
6	−26.5	−29.6
4	−14.5	−14.8

straints to speed the convergence time of the optimization. These constraint techniques allow the designer to round the infinitely precise coefficients to the nearest M variable neighborhood, where M is the number of LSB bits which were allowed to vary. We especially concentrate on the simplest case of $M = 1$, i.e., the unit neighborhood. In addition, the constraints allow the designer the flexibility of fixing some coefficients while varying others. Although the use of our constraints reduced the convergence time of the optimization algorithm, the approximation error increased. The error (i.e., the deviation between the magnitude response of the ideal and quantized filters) in the approximation using either the unconstrained or constrained mixed-integer programming techniques is compared to the error introduced by either straight rounding or truncation of the coefficients. In some design examples, we obtained improvements of 7 dB in the stopband attenuation between the amplitude response of the optimized filter with finitely precise coefficients and that of the filter obtained by rounding the infinitely precise coefficients.

Section IV shows how to use zero-one integer formulation to design FIR filters. We compare the results of a zero-one integer design with the mixed-integer design discussed in Section III. Finally, we show how to design filters with powers of two coefficients.

II. THE TRANSFER FUNCTION

The transfer function of a linear-phase FIR filter of length N has the general form

$$H_a(z) = \sum_{n=-(N-1)/2}^{(N-1)/2} h_n z^{-n}, \tag{1}$$

where the length N is odd and $h_n = h_{-n}$. The coefficients h_n are real and can take on any value. The frequency response, $H_a(e^{j2\pi f})$ is obtained by evaluating $H_a(z)$ along the unit circle, $z = e^{j2\pi f}$,

$$H_a(e^{j2\pi f}) = e^{-j2\pi f(N-1)/2} \cdot \left[h_0 + 2 \sum_{k=1}^{(N-1)/2} h_k \cos(2\pi f k) \right], \tag{2}$$

where f is the normalized frequency variable and $(N-1)/2$ is the delay of the filter. The magnitude function, $H(e^{j2\pi f})$, is given by

$$H(e^{j2\pi f}) = h_0 + 2\sum_{k=1}^{(N-1)/2} h_k \cos(2\pi f k), \quad \text{for } N \text{ odd} \tag{3}$$

$$H(e^{j2\pi f}) = 2\sum_{k=1}^{N/2} h_k \cos(2\pi f k), \quad \text{for } N \text{ even.} \tag{4}$$

The distinction between N odd and N even is of considerable importance in the design and mechanization of FIR filters.[27] However, for the sake of simplicity, we assume hereafter that the length N is odd. We use the symbol h'_k to represent the infinitely precise coefficients.

In an actual implementation, the infinitely precise coefficients $\{h'_k\}$ are quantized to take on values $\{h_k\}$ which are integer multiples of the smallest quantizing step size, 2^{-B}, where B is the number of bits used for the implementation. The difference between the quantized and infinitely precise coefficient is defined by

$$\delta h_k = h'_k - h_k. \tag{5}$$

If the quantizing scheme is rounding, then

$$|\delta h_k| \leq 1/2 \cdot 2^{-B}, \tag{6}$$

and, for truncation,

$$|\delta h_k| \leq 2^{-B}. \tag{7}$$

If the relationship between changes in the desired frequency response $H(e^{j2\pi f})$ and the coefficients $\{h'_k\}$ are known, the degradation in performance of the quantized filter can be bounded. Such a relationship has the general form:

$$\frac{\delta H(e^{j2\pi f})}{\delta h_k} = \frac{\delta}{\delta h_k} \cdot \left[h'_0 + 2\sum_{k=1}^{(N-1)/2} h'_k \cos(2\pi f k) \right] \tag{8}$$

$$= 1 \qquad \text{if } k = 0 \tag{9}$$

$$= 2\cos(2\pi f k) \qquad \text{if } k \neq 0. \tag{10}$$

Using eq. (8), we obtain the frequency sensitivity function, $\Delta H(e^{j2\pi f})$, which is expressed in the following form:

$$\Delta H(e^{j2\pi f}) = \delta h'_0 + 2\sum_{k=1}^{(N-1)/2} \delta h'_k \cos(2\pi f k). \tag{11}$$

Assuming the largest change occurs in each coefficient, a worst case bound on $\Delta H(e^{j2\pi f})$ is given by

$$\Delta H(e^{j2\pi f}) \leq 2^{-(B+1)} \cdot \left[1 + 2\sum_{k=1}^{(N-1)/2} |\cos 2\pi f k| \right]. \tag{12}$$

The above equation is the Chan and Rabiner deterministic bound.[20] We can improve this bound if we regard $\delta h_k'$ as a variable which can take on a maximum value of $2^{-(B+1)}$ rather than assume it is a constant. Then a bound on the frequency sensitivity function is obtained in the appendix as

$$\Delta H(e^{j2\pi f}) \leq 2^{-(B+1)} \cdot \sqrt{N^2 + \frac{N}{2} - \frac{1}{2}} \cdot W_N(f) \tag{13}$$

where

$$W_N(f) = \frac{(N-1) + [\sin(N2\pi f)/\sin(2\pi f)]^{1/2}}{(2N-1)} . \tag{14}$$

The weighting function, $W_N(f)$, takes on a value of 1 at $f = 0$ and $f = \frac{1}{2}$, i.e.,

$$W_N(0) = W_N(\tfrac{1}{2}) = 1$$

and

$$W_N(f) \leq 1 \qquad \text{for all other values of } f.$$

Therefore, a frequency-independent upper bound on $\Delta H(e^{j2\pi f})$ is given by

$$\Delta H(e^{j2\pi f}) \leq 2^{-(B+1)} \cdot \sqrt{N^2 + \frac{N}{2} - \frac{1}{2}} . \tag{15}$$

In the limit as $N \to \infty$

$$\lim_{N\to\infty} W_N(f) = \frac{1}{\sqrt{2}}, \qquad 0 < f < \frac{1}{2}$$

and an approximate bound on the frequency sensitivity function for large N becomes

$$\begin{aligned} \Delta H(e^{j2\pi f}) &\leq \frac{1}{\sqrt{2}} \cdot 2^{-(B+1)} \cdot \sqrt{N^2 + \frac{N}{2} - \frac{1}{2}} \\ &\leq \frac{1}{\sqrt{2}} \cdot 2^{-(B+1)} \cdot N. \end{aligned} \tag{16}$$

We refer to the above bound as the L-2 norm bound, and, in comparison to the Chan-Rabiner deterministic bound,[20] the L-2 norm bound is tighter by a factor of $1/\sqrt{2}$, or 3 dB. However, this deterministic bound is not as tight as the statistical bounds of Gersho et al[21] and Chan-Rabiner.[20] The basic assumptions of the statistical bounds is that the quantizing error in each coefficient is randomly distributed. In Fig. 4,

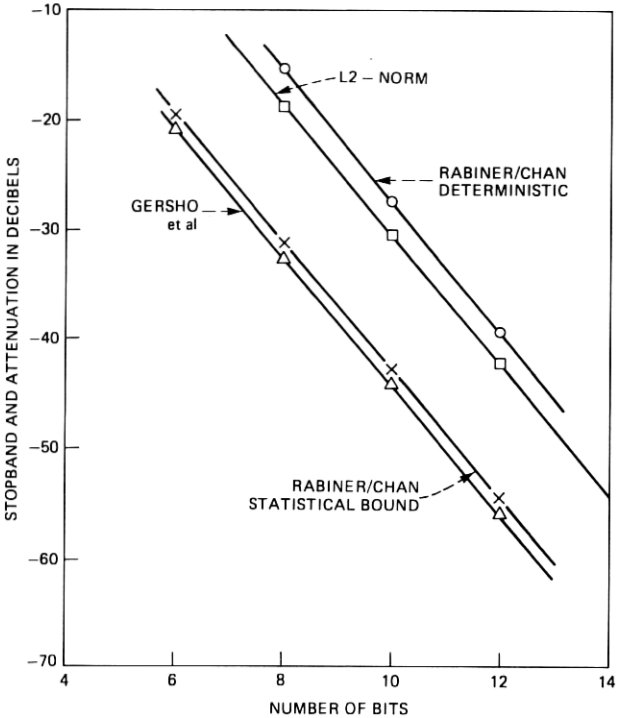

Fig. 4—Bounds on frequency sensitivity function.

we compare the stopband attenuation derived from these bounds, for a filter of length 33 with passband and stopband normalized frequencies of 0.15 and 0.3, respectively. We also display in Table IV the appropriate number of bits required for such a filter to have a stopband attenuation of at least 45 dB.

In Table IV, we find by using our technique that an optimized filter would require only 8 bits, while the bounds suggest wordlengths of 10 to 13 bits. This indicates that significant improvements can be achieved in filter performance if the filter coefficients are optimized; that is, by including the effects of quantization in the direct design of FIR filters. In the next section we present various techniques for designing FIR filters with finitely precise coefficients.

III. FORMULATION OF THE DESIGN PROBLEM

We now address the problem of finding an optimum set of coefficients that give the best approximation of $H(e^{j2\pi f})$ to a desired magnitude function $D(e^{j2\pi f})$ in the minimax sense. We denote the error in the approximation by

$$E(e^{j2\pi f}) = D(e^{j2\pi f}) - H(e^{j2\pi f}), \tag{17}$$

where $0 \leq f \leq \frac{1}{2}$. The desired magnitude function $D(e^{j2\pi f})$ is a real-valued function which for our purposes will be defined only at discrete set of frequencies $\{f_k\}$ where $k = 1, 2, \cdots, K$. The choice of the discrete set of points $\{f_k\}$ is of considerable importance. Equation (17) can be written in vector form as:

$$\mathbf{E} = \mathbf{D} - \mathbf{H}. \tag{18}$$

Rabiner showed that an optimal set of infinitely precise coefficients $\{h_i\}$ that best approximates $\mathbf{H}$ to $\mathbf{D}$ in the minimax sense is obtained by formulating the problem as a linear programming problem.[10]

3.1 Linear programming formulation

For simplicity of exposition, the linear programming problem is formulated for a low-pass filter that best approximates the desired magnitude characteristics shown in Fig. 2. The passband and stopband frequencies are f_p and f_s, respectively. The desired magnitude function $D(e^{j2\pi f})$ is represented mathematically as

$$\begin{aligned} D(e^{j2\pi f}) &= 1 \qquad \text{for } 0 \leq f_k \leq f_p \\ &= 0 \qquad \text{for } f_s \leq f_k \leq \tfrac{1}{2}. \end{aligned} \tag{19}$$

From Fig. 2, the constraints on the low-pass filter are:

$$\begin{aligned} 1 - \delta_1 \leq H(e^{j2\pi f}) \leq 1 + \delta_1 \qquad &\text{for } 0 \leq f_k \leq f_p \quad \text{passband} \\ H(e^{j2\pi f}) \leq \delta_2 \qquad &\text{for } f_s < f_k \leq \tfrac{1}{2} \text{ stopband.} \end{aligned} \tag{20}$$

The linear programming problem is to minimize

$$f(\delta_1, \delta_2) = C'X \tag{21}$$

subject to

$$\mathbf{AX} < \mathbf{D}_0 \tag{22}$$

$$\mathbf{L} \leq \mathbf{X} \leq \mathbf{U}, \tag{23}$$

where

$$C' = [c_1, c_2, 0, 0 \cdots 0] \tag{24}$$

is the transpose of an $[(N-1)/2] + 3$ vector. The coefficients c_1 and

Table IV—Number of bits required to achieve a stopband attenuation of 45 dB for the filter specified in Fig. 2

Chan-Rabiner Deterministic Bound	13 bits
L-2 Norm	12 bits
Chan-Rabiner Statistical Bound	10 bits
Gersho, Gopinath, and Odlyzko	10 bits
Actual Rounded Filter	10 bits
Empirical 6-dB Estimate	9 bits
Optimized Filter	8 bits

c_2 are the weights of δ_1 and δ_2, the passband and stopband ripples, respectively.

$$\mathbf{X} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ h'_1 \\ h'_2 \\ \cdot \\ h'_i \\ \cdot \\ h'_{(N-1)/2} \end{bmatrix} \tag{25}$$

is an $[(N-1)/2] + 3$ vector whose third to $[(N-1)/2] + 3$ elements are the filter coefficients. The first and second elements are the passband and stopband ripples. All the elements of $\mathbf{X}$ are real numbers with infinite precision. The matrix A is a $2K$ by $[[(N-1)/2] + 3]$ constraint matrix whose elements are either -1, 0, or $\pm\{2\cos 2\pi f_k i\}$, where i runs from 1 to $(N-1)/2$. D_0 is a $2K$ vector whose elements are either 1, 0, or -1. U and L are $([(N-1)/2] + 3)$ vectors that specify the upper and lower bounds on the vector $\mathbf{X}$. For a low-pass filter whose passband magnitude is normalized to unity, the magnitude of the elements of $\mathbf{U}$ and $\mathbf{L}$ are all less than unity.

3.2 Formulation of mixed integer linear programming problem

The decision variables, h'_i, obtained from the linear programming formulation above are the infinitely precise filter coefficients. When these variables are quantized to a fixed number of bits, the resulting solution is no longer optimal. To obtain an optimal solution, the effects of coefficient quantization should be included in the formulation of the problem. This is done by formulating the problem as a mixed-integer linear programming problem. In formulating a mixed-integer programming problem, the symbol B represents the number of bits, including the sign bit. We also scale the vectors D_0, U and L by 2^{B-1}. Substituting the scaled vectors into (22) and (23) results in the following mixed-integer linear programming formulation shown in summation notation:

$$\text{minimize } c_1\delta_1 + c_2\delta_2 \tag{26}$$

subject to

$$\left.\begin{aligned} -1.\delta_1 - 0.\delta_2 \pm h_0 \pm 2\sum_{i=1}^{\frac{N-1}{2}} h_i \cos 2\pi f_j i \le \pm 2^{-B+1} \\ 0 \le f_j \le f_p \ (i.e., \text{ passband}) \\ 0.\delta_1 - 1.\delta_2 \pm h_0 \pm 2\sum h_i \cos 2\pi f_j i \le 0 \\ f_s \le f_j \le f_{1/2} \ (i.e., \text{ stopband}) \end{aligned}\right\} \tag{27}$$

$$\alpha_1 \leq \delta_1 \leq \beta_1, \tag{28}$$

$$\alpha_2 \leq \delta_2 \leq \beta_2, \tag{29}$$

$$l_i \leq h_i \leq u_i, \qquad i = 0 \cdots \frac{N-1}{2}. \tag{30}$$

The decision variables δ_1 and δ_2 are real variables bounded by α_1, β_1, α_2, β_2. The rest of the decision variables $h_i's$ are integer variables bounded by integer constants less than or equal to 2^{B-1}. The problem as formulated in (26) through (30) is a mixed-integer linear programming problem. Commercial software packages are available for solving mixed-integer linear programming problems.[23] A particular software package we found useful was written by Kochman.[25] This program's interfaces are straightforward and can easily be integrated with other software packages to provide a complete system that is efficient in computation time and storage requirements. The additional software needed for efficient computation time and storage requirements are discussed later.

The integer variables that result from the mixed-integer linear programming problem defined above are the coefficients of a linear phase FIR filter. These coefficients are integers which lie in the range -2^{B-1} to 2^{B-1}. The actual binary bits used in the mechanization of the filter are the binary representation of the integer variables. We plot, in Fig. 5, the stopband attenuation for a filter of length 33 with passband and stopband frequencies of 0.15 and 0.3, respectively, using 12, 10, 8, 6, and 4 bits. These results were obtained using mixed-integer programming design. Also plotted in Fig. 5 are the stopband attenuation derived using the Gersho et al. bound,[21]* the rounded solution, and the 6 dB/bit heuristic estimate. For all the number of bits considered, the optimized filter increased the stopband attenuation by 5-7 dB. Improvements were also observed in the passband ripples. The difference in stopband attenuation between the optimized filter and the Gersho et al. bound for quantized coefficients is approximately 10 dB. These results clearly show the advantage of the optimized filter compared with direct quantization of the infinite-precision filter.

3.3 Computational efficiency

A mixed-integer linear programming problem is solved in two stages. First, the problem is optimized by considering all the integer variables as being continuous variables. The optimal solution obtained is called an optimal continuous solution. These are the infinitely precise filter coefficients $\{h_i'\}$. Second, a search is started from the optimal contin-

* We used the Gersho et al. bound because in our experience it is the tightest bound. The Gersho et al. bound predicts performance for quantization of conventional filter designs and not for the mixed-integer optimized filter.

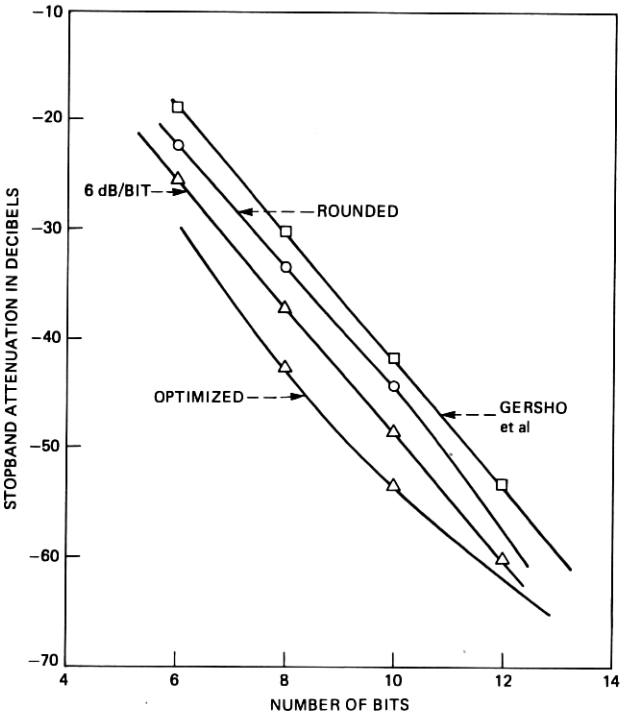

Fig. 5—Comparison of bounds, rounded and optimized solution.

uous solution obtained at the end of the first stage. The integer variables are then forced to take integer values using a "branch and bound" technique with heuristic rules. An optimal integer solution satisfying all the constraints and giving the best possible value to the objective function is searched for. Using an ordinary commercial mixed-integer programming software package on a filter design problem results in long computation times,* unless several improvements are added to the software package. The reasons for this are twofold. First, the constraint matrix A is unusually large and dense.† Second, simple commercial software packages employ simple, straightforward strategies. A straightforward strategy leads to a series of integer solutions tending toward the optimal integer solution. When an integer solution is found, it is not immediately known whether it is optimal. The search continues either until a better solution is found or until all the set of possible solutions is exhausted. For problems with many integer variables and relatively loose constraints, good solutions are

* This severely limits the length of filters than can be designed.

† The constraint matrix A is very sparse for problems found in the fields of investment, capital budgeting, and production planning. Most commercially available, mixed-integer programming packages were designed for these types of problems.

quickly found, but a long computation is necessary either to improve them slightly or to prove their optimality.

To alleviate the above problems, we introduced several improvements in our software package to enhance the computational efficiency of the algorithm. For example, the designer can interrupt the search after an integer solution has been found and change the bounds on the passband and stopband ripples. This decreases the computation time and storage requirements. Another important consideration is the order in which integer variables are processed during the search for integer solutions. This order is called "priority order" of integer variables. Solution times vary significantly with the priority order chosen. Integer variables should be processed according to their importance in the model, the most important ones being processed first. Usually the integer variables are processed in the order* in which they appear in the decision vector **X**. We found in most filter design examples that the smaller coefficients are the most sensitive. Slight changes in their least significant bits produced correspondingly larger changes in the stopband ripple. In a particular example, (see Table II), a change in the 23rd coefficient for the rounded filter from one to zero produced a 5-dB change in the stopband ripple. A change from one to zero represents just a change in the least significant bit. The 23rd coefficient is shown in a rectangular box in Table II. Therefore, eqs. (26) through (30) are rearranged so that the smaller coefficients appear in the leading rows of the decision vector **X**. Another facility in the software package is the option to freeze some integer variables. This fixes the activities of the integer variables to their current integer values in the matrix, thus allowing for post-optimal studies of the solution. This facility is essential for very long length filters, in particular, filters of length greater than 63. The computation time required to obtain either an optimal or a good solution increases with the length of the filter. For very long length filters, a good solution is obtained by dividing the original integer programming problem into suboptimal integer problems using the following algorithm.

Step 1: (Initialization) Divide the integer coefficients into two sets, S_1 and S_2. Fix the coefficients in S_1 at their rounded values. Vary the coefficients in S_2 for a suboptimal integer solution. Go to Step 2.

Step 2: Fix the coefficients in S_2 obtained from the suboptimal integer solution. Vary the coefficients in S_1 for a new suboptimal solution. Go to Step 3.

Step 3: Fix the coefficients in S_1 obtained from the current subop-

* This is the case except when other priority capabilities have been included in the software package.

timal integer solution in Step 2. Vary the integer coefficients in S_2 for a new suboptimal solution. Go to Step 4.

Step 4: Test whether the current suboptimal solution is desirable. If not, go to Step 2, or else terminate.

Such a heuristic procedure eventually produces a fairly good integer solution. This solution is only a local optimum.

3.4 M variable neighborhood

For the mixed-integer linear programming problem formulated in (26) through (30), the decision variables take on values ranging from -2^{-B+1} to 2^{-B+1}. Such a large range represents loose constraints which generally requires long computation times for filter optimization problems. The computation time can be shortened by tightening the constraints or bounds on the decision variables. This restricts the decision variables within the neighborhood of the global continuous solution. This concept is referred to as the M-variable neighborhood. The integer solution obtained using this technique is a local optimum. The elements of $\mathbf{U}_m$ and $\mathbf{L}_m$ define the M-variable neighborhood:

$$\mathbf{U}_m = \left\{ h_i' + M, i = 0, \cdots \frac{N-1}{2} \right\}, \tag{31}$$

$$\mathbf{L}_m = \left\{ h_i' - M, i = 0, \cdots \frac{N-1}{2} \right\}, \tag{32}$$

where h_i, is the ith infinitely precise filter coefficient obtained from the first part of the optimization (h_i' is the scaled coefficient). Thus, the decision variables $\{h_i\}$ can only take on values that are M units or less from the infinitely precise coefficient h_i'. The special case when $M = 1$ is referred to as the unit neighborhood. The local optimal solution produced by the unit neighborhood is equivalent to the best quantizing* scheme for the continuous solution. We plot in Figure 6 the stopband attenuation for a filter designed using the unit neighborhood technique. This curve is labeled in Fig. 6 as best rounding. The length of the filter used was 33 and the passband and stopband frequencies were 0.15 and 0.3, respectively. In these cases, 12, 10, 8, 6, and 4 bits were used. Also plotted in Fig. 6 are the stopband attenuation curves for the rounded filter and the global optimized filter.† We find in Fig. 6 an improvement of approximately 3 dB in stopband attenuation for the best rounded filter compared to the stopband attenuation of the simple rounded filter. The optimized filter has at least 5 to 7 dB

* The quantizing scheme used in this context includes a mixture of rounding, truncation or boosting, where boosting is the opposite of truncation. In using this scheme, each coefficient is rounded up or down in the optimal way.

† For the global optimized filter, $M = 2^{B-1}$.

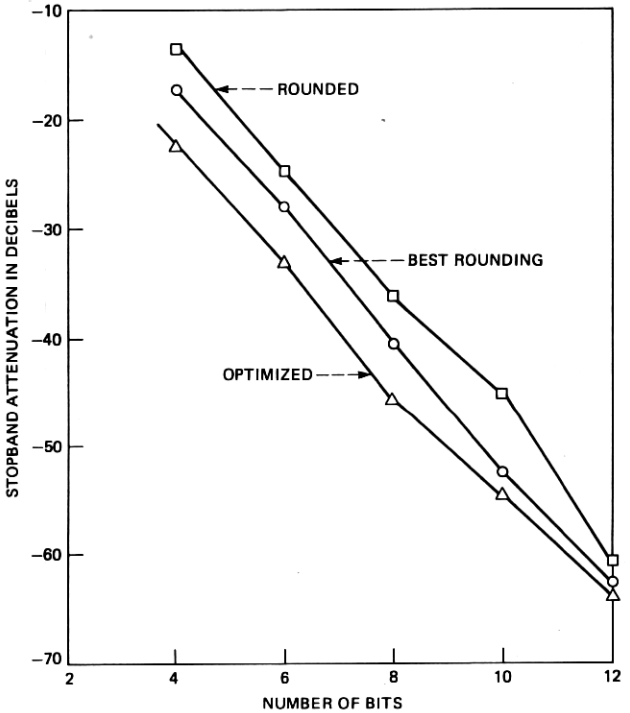

Fig. 6—Stopband attenuation for rounded, best rounded and optimized filters.

improvement in stopband attenuation compared to the simple rounded filter. Improvements were also observed in the passband ripples for the optimized and best rounded filters. Table V displays the filter coefficients for the best rounded, simple rounded, and optimized filters.

One can select any neighborhood; the larger the value of M, the closer the local optimal solution approaches the global optimal solution. A global optimum is also a local optimum with respect to any neighborhood containing the global optimum. Thus, enumeration of all local optima, with respect to all neighborhoods, may produce an acceptable solution. Sometimes iterations over a few neighborhoods would be sufficient to obtain a good solution. The neighborhood technique reduces the computation time, search space and storage requirements.

The unit neighborhood technique is extremely useful for very long length filters. These filters are extremely difficult to design using integer programming in a reasonable amount of computer time. For example, a 12-bit filter of length 63 was designed using the unit neighborhood technique. An improvement of 3 dB was observed in the stopband attenuation compared to the simple rounded filter. Improve-

ments were also observed in the passband. These results are displayed in Table VI.

IV. ZERO-ONE FORMULATION

Zero-one integer programming[26] is a special case of the integer linear programming problem formulated in (26) through (30). The decision variables for zero-one programming are restricted to two values, 0 or 1. In our experience, zero-one integer programming packages exhibit faster run time for digital filter design applications. The decision variables are the actual binary bits of the filter coefficients. The decision variables in eqs. (26) through (30) can be either positive or negative integers. To formulate a zero-one problem, the decision variables are transformed to take on values in the range $0 \rightarrow 2^B$, thus eliminating the sign variable. The new variables are:

$$\hat{h}_i = h_j + 2^{B-1}. \tag{33}$$

$\hat{h}_i$ is expressed in terms of binary values as:

$$\hat{h}_i = \sum_{q=0}^{B-1} h_{iq} 2^q. \tag{34}$$

Substituting eqs. (33) and (34) into eqs. (26) through (30), we have

$$\text{minimize } f(\hat{\delta}_1, \hat{\delta}_2) = c_1\hat{\delta}_1 + c_2\hat{\delta}_2 \tag{35}$$

subject to

$$-\delta_i \pm \sum_{q}^{B-1} h_{0q}2^q \pm 2 \sum_{i=1} \left(\sum_{q}^{B-1} h_{iq}2^q \right) \cos 2\pi f_j i \leq \pm 2^{B-1} \pm g(f_j)$$

$$0 \leq f_j \leq f_p \tag{36}$$

$$-\delta_2 \pm \sum_{q}^{B-1} h_{0q}2^q \pm 2 \sum_{i=1} \left(\sum_{q}^{B-1} h_{iq}2^q \right) \cos 2\pi f_j i \leq \pm g(f_j), \tag{37}$$

$$f_s \leq f_j \leq f_{1/2}, \qquad \hat{\alpha}_n \leq \hat{\delta}_n \leq \hat{\beta}_n, \tag{38}$$

where $n = 1$, or 2 and $\hat{\alpha}_n$ and $\hat{\beta}_n$ are the bounds on the stopband and passband ripples.

$$h_{iq} = 0 \text{ or } 1. \tag{39}$$

In eqs. (36) and (37), $g(f_j)$ is given by

$$g(f_j) = 2 \cos\left(\frac{N+1}{2} \pi f_j\right) \cdot \frac{\sin\left(\frac{N+1}{2} \pi f_j\right)}{\sin \pi f_j} - 1 \quad 0 < j < K \tag{40}$$

Table V—Coefficients for rounded, best rounded, and optimized filters

(Length of Filter = 33, Sampling Frequency = 1.0, Passband Frequency = 0.15, Stopband Frequency = 0.3)

Infinite Precision	Rounding	Best Rounding	Optimized
117.03603	117	118	105
79.50871	80	80	77
9.84019	10	9	20
−22.41649	−22	−23	−17
−8.16686	−8	−8	−15
9.50543	10	10	3
5.94591	6	6	9
−3.88934	−4	−4	2
−3.75361	−4	−4	−4
1.31050	1	1	−2
2.01544	2	2	1
−0.28597	0	0	1
−0.89044	−1	−1	0
−0.00239	0	0	0
0.30452	0	0	0
0.02607	0	0	0
−0.07076	0	0	0

and

$$g(f_j = 0) = N + 2. \tag{41}$$

By using relationship (40) and (41), we ensure that the coefficients are positive. Equations (35) through (39) are solved using a zero-one, mixed-integer, linear programming software package.

Equations (35) to (39) are written in matrix form as:

$$\min C'X \tag{42}$$

subject to:

$$A_z X \leq D_0, \tag{43}$$

$$0 \leq X \leq 1. \tag{44}$$

The constraint matrix A_z is very large and dense. Furthermore, it has no structure. For a 12-bit filter of length 49, the number of columns of A_z is $(25 \times 12 + 2)$. Allowing for at least one degree of freedom for each coefficient, there must be twice as many rows (frequency points) as columns. Hence, for this example the A_z matrix is a 604-by-302 matrix. Inverting this large dense matrix several times, as is often done in integer linear programming, is costly and very sensitive to numerical errors. To alleviate this difficulty, we implemented the following scheme. First we solved the problem as a linear programming problem and then rounded each coefficient to its nearest binary value. We partitioned the vector **X** into two parts. One part of **X** contained the least $(B - q)$ bits. The idea was to fix the q most significant bits and allow only the least $B - q$ bits to vary. Second, we separated the

matrix A_z into A_q and A_{Bq} accordingly, so that the resulting integer minimization problem is:

$$\min C'X \tag{45}$$

subject to

$$A_{Bq}X_{Bq} < (D_0 - A_qX_q). \tag{46}$$

The new constraint matrix A_{Bq} is smaller. For the example discussed earlier, by varying only the last five bits, the constraint matrix A_{Bp} becomes a 254-by-127 matrix. The constraint matrix A_{Bq} can be reduced further in size since the vector X_{Bq} need not contain all the $(B - q)$ least significant bits of each coefficient, $\hat{h}_i$. The larger coefficients may be $(B - q)$ bits while the smaller end coefficients have only one or two bits contained in the modified X_{Bq} matrix.

4.1 Filters with powers of two coefficients

Usually filters with powers of two coefficients can be mechanized easily in hardware by simple shift operations, since the binary representation of such coefficients has only one nonzero bit. The filter coefficients are constrained to be zero or powers of two by adding the following constraint to (36) through (39):

$$\sum_{q=0}^{B-1} h_{iq} = 1 \tag{47}$$

for all $i = 0 \cdots (N-1)/2$. FIR linear phase filter with powers of two coefficients can be designed using eqs. (35) through (39) and eq. (47), together with a zero-one integer optimization software package.

V. SUMMARY OF COMPUTATIONAL RESULTS

To illustrate the effectiveness of integer optimization, we considered a 33-tap low-pass filter. The passband and stopband frequencies were 0.15 and 0.30, respectively, and the normalized sampling frequency was 1.0. The filter with infinitely precise coefficients designed using linear

Table VI—Stopband and passband ripples
(Length of Filter = 63, Sampling Frequency = 1.0,
Passband Frequency = 0.1875, Stopband Frequency = 0.2625,
Infinitely Precise Coefficients Stopband Ripple = −79.17 dB,
Infinitely Precise Coefficients Passband Ripple = 0.001 dB)

No. of bits	Rounding (dB)	Truncation (dB)	Best Rounding (dB)
	Stopband Ripple		
12	−56.9	−54.4	−60.4
	Passband Ripple		
12	0.01	0.01	0.004

Table VII—Stopband ripples

(Length of Filter = 33, Sampling Frequency = 1.0, Passband Frequency = 0.15, Stopband Frequency = 0.3, Infinitely Precise Coefficients Stopband Ripple = −79.12 dB)

No. of bits	Rounding (dB)	Truncation (dB)	Optimized Integer Solution (dB)	Best Rounding (dB)	Varying 5 Last bits (dB)
12	−62.4	−62.7	−66.2	−65.3	−66.2
10	−49.4	−47.0	−55.9	−54.4	−55.9
8	−38.5	−38.9	−47.2	−42.1	−47.2
6	−26.5	−29.6	−33.8	−29.6	−33.8
4	−14.5	−14.8	−23.4	−16.9	−23.4

programming had a stopband attenuation of −79 dB, the passband ripple was 0.001 dB, and the coefficients were represented in sign-magnitude format. We studied the effect of rounding or truncating the coefficients to either 12, 10, 8, 6, or 4 bits. The results for the stopband and passband ripples are shown in columns 2 and 3 of Tables VII and VIII, respectively. For the 8- and 6-bit filters, truncating the coefficients was better than rounding.

Optimized filters were designed using the mixed-integer technique for filter wordlengths of 12, 10, 8, 6, or 4 bits. In each case, we had improvements in the stopband attenuation of between 5 and 7 dB. Improvements were observed in the passband. These results are shown in column 4 of Tables VII and VIII for the stopband and passbands, respectively. We also designed filters using the unit neighborhood scheme (i.e., best rounding) for the different filter wordlengths. Compared to the roundoff filters we found improvements of 3 to 5 dB in the stopband performance. The zero-one mixed-integer formulation was used for designing several filters. We used the partitioning-of-variables technique discussed earlier to reduce computation and storage requirements. Here we found that varying the last two binary bits produce negligible improvement over roundoff solutions. However, we found that varying the last five bits of either the 12-, 10-, 8-, or 6-bit filters in the zero-one integer programming design produced the same

Table VIII—Passband ripples

(Length of Filter = 33, Sampling Frequency = 1.0, Passband Frequency = 0.15, Stopband Frequency = 0.3, Infinitely Precise Coefficients Passband Ripple = 0.001 dB)

No. of bits	Rounding (dB)	Truncation (dB)	Optimized Integer Solution (dB)	Best Rounding (dB)	Varying 5 Last bits (dB)
12	0.005	0.005	0.003	0.004	0.003
10	0.02	0.03	0.01	0.02	0.01
8	0.06	0.12	0.06	0.06	0.06
6	0.36	0.33	0.14	0.17	0.14
4	0.39	0.90	0.25	0.30	0.25

Table IX—Computation time

Infinite precision solution	1.8 seconds
Best rounding	9.6 seconds
Optimized solution	790 seconds
Proof of optimality	1560 seconds
Zero-one optimization varying 5 bits	370 seconds

stopband attenuation and filter coefficients as the optimized filter discussed earlier.

5.1 Computation time

We display in Table IX the CPU time used for the design of a 12-bit FIR filter of length 33. The normalized passband and stopband frequencies were 0.15 and 0.3, respectively. The stopband and passband frequencies are displayed in Tables VII and VIII, respectively. The filter was designed on the IBM 370 using mixed-integer optimization techniques. The CPU time used to obtain the continuous solution (i.e., the infinitely precise coefficients) was 1.8 seconds. It took 9.6 seconds to obtain the best rounded filter and 790 seconds to obtain optimized integer filter coefficients. It took 1560 seconds, twice as long, to exhaust all the possible solutions. These long computation times were shortened using the zero-one mixed-integer optimization technique and varying only the least five significant bits of the coefficients. The CPU time used was 370 seconds.

VI. CONCLUSIONS

New techniques for designing minimax linear-phase FIR filters with finitely precise coefficients have been presented. These techniques generate a number of possible solutions, including that of simple rounding or truncation, and select the best finitely precise coefficients from this set. In this way, significant improvement in filter performance is gained over methods that simply round off or truncate the infinitely precise coefficients. In all design examples considered, our techniques increased the stopband attenuation by at least 7 dB, as well as reduced the passband ripple, compared to techniques that simply round off the infinitely precise coefficients.

It is difficult to use integer optimization to design long filters, since computation time as well as storage requirements are excessive unless specialized techniques are employed. The computation time and storage requirements were considerably reduced by using zero-one integer optimization with constraints on the binary bits. This technique is recommended for designing optimum FIR filters with limited-precision coefficients. A simplified version of our method chooses the best rounding scheme for quantizing infinitely precise coefficients to a fixed

number of bits. The design of a 63-tap filter using the simplified scheme improved the stopband attenuation by 3 dB.

VII. ACKNOWLEDGMENTS

We gratefully acknowledge our debt to G. Kochman and T. Moore for making available, and assisting us in the use of, mixed-integer programming packages. We also acknowledge useful discussions with J. F. Kaiser.

APPENDIX

Derivation of Frequency Sensitivity Function

To prove the upper bound on the frequency sensitivity function given by (13), we regard δh_k as a variable which can take on a maximum of $2^{-(B+1)}$. Therefore, the maximum change in the frequency sensitivity response, $\Delta H(e^{j2\pi f})$ is given by:

$$\Delta H(e^{j2\pi f}) = \max_{|\delta h'_k| \le 2^{-(B+1)}} \left| \sum \frac{dH(e^{j2\pi f})}{dh'_k} \cdot \delta h'_k \right|$$

$$= \max_{|\delta h'_k| \le 2^{-(B+1)}} \cdot \left| \left(\delta h'_0 + \sum_{k=1}^{(N-1)/2} 2\delta h'_k (\cos 2\pi f k) \right) \right| \quad (48)$$

$$= \max_{|\delta h'_k| \le 2^{-(B+1)}} \cdot \left| \left(\delta h'_0, \delta h'_1, \ldots, \delta h' \frac{N-1}{2} \right) \cdot \left(1, 2\cos 2\pi f, \ldots, 2\cos 2\pi f \left(\frac{N-1}{2} \right) \right) \right|. \quad (49)$$

The above equation is the magnitude of an inner product which we denote by

$$\Delta H(e^{j2\pi f}) = \max_{|\delta h'_k| \le 2^{-(B+1)}} \cdot |((\delta \mathbf{h}', F(\mathbf{f})))| \quad (50)$$

$$\le \max_{\|\delta h'\| \le \sqrt{(N+1)/2} \cdot 2^{-(B+1)}} \cdot |((\delta \mathbf{h}', F(\mathbf{f})))|. \quad (51)$$

Using Schwartz's inequality, the above expression simplifies to

$$\Delta H(e^{j2\pi f}) \le \sqrt{(N+1)/2} \cdot 2^{-(B+1)} \cdot \| F(\mathbf{f}) \|, \quad (52)$$

when

$$\delta \mathbf{h}' = \sqrt{(N+1)/2} \cdot 2^{-(B+1)} \cdot \frac{F(\mathbf{f})}{\| F(\mathbf{f}) \|}. \quad (53)$$

Writing eq. (52) in terms of cosines we obtain

$$\Delta H(e^{j2\pi f}) \le \sqrt{(N+1)/2} \cdot 2^{-(B+1)} \cdot \left[1 + 4 \sum_{k=1}^{(N-1)/2} \cos^2(2\pi f k) \right]^{1/2},$$

$$\Delta H(e^{j2\pi f}) \le \sqrt{(N+1)/2}\cdot 2^{-(B+1)} \cdot \left[1 + 2\sum_{k=1}^{(N-1)/2}(1+\cos 4\pi f k)\right]^{1/2}. \quad (54)$$

Using the fact that

$$\frac{1}{2} + \sum_{k=1}^{(N-1)/2} \cos 2\pi f k = \frac{\sin N\pi f}{2\sin \pi f}, \quad (55)$$

$$\Delta H(e^{j2\pi f}) \le \sqrt{(N+1)/2}\cdot 2^{-(B+1)} \cdot \left[(N-1) + \frac{\sin N2\pi f}{\sin 2\pi f}\right]^{1/2} \quad (56)$$

$$\le 2^{-(B+1)}\cdot\sqrt{(2N-1)(N+1)/2} \cdot \left[\frac{(N-1)}{(2N-1)} + \frac{\sin N2\pi f/\sin 2\pi f}{(2N-1)}\right]^{1/2} \quad (57)$$

$$\le 2^{-(B+1)}\sqrt{N^2 + \frac{N}{2} - \frac{1}{2}}\cdot W_n(f), \quad (58)$$

where

$$W_N(f) = \left[\frac{(N-1)}{(2N-1)} + \frac{\sin N2\pi f/\sin 2\pi f}{(2N-1)}\right]^{1/2}.$$

REFERENCES

1. K. V. Mina, V. B. Lawrence, and J. J. Werner, "Digital Techniques for Communication Signal Processing," IEEE Communications Society Magazine, *16,* No. 1 (January 1978), pp. 18–22.
2. S. L. Freeny et al, "System Analysis of a TDM-FDM Translator/Digital A-Type Channel Bank," IEEE Trans. Commun., *COM-19,* No. 6 (December 1971).
3. H. G. Alles et al, "Digital Signal Processing in Telephone Switching," Conference Record, ICC 1977. Minneapolis.
4. R. W. Lucky, "Automatic Equalization for Digital Communication," B.S.T.J., *44,* No. 4 (April, 1965), pp. 547–588.
5. S. B. Weinstein, "Echo Cancellation in the Telephone Network," IEEE Commun., *15,* No. 1 (January 1977).
6. L. R. Rabiner and R. Schafer, *Digital Processing of Speech Signals*, Englewood Cliffs, N.J.: Prentice-Hall, 1978.
7. A. Oppenheim and R. Schafer, *Digital Signal Processing*, Englewood Cliffs, N.J.: Prentice-Hall, 1977, pp. 450–500.
8. L. R. Rabiner, "The Design of Finite Impulse Response Digital Filters Using Linear Programming Techniques," B.S.T.J., *51,* No. 6 (July–August 1972), pp. 1177–1198.
9. H. D. Helms, "Digital Filters with Equiripple or Minimax Responses," IEEE Trans. Audio and Electroacoustic, *AU-19,* No. 1 (March 1971).
10. F. Brglez, "Sampling Rate and Word Length Minimization in FIR Digital Filters," International Symposium on Circuits and Systems, New York, May 1978.
11. T. W. Parks and J. H. McClellan, "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase," IEEE Trans. Circuit Theory, *CT-19* (March 1972), pp. 189–194.
12. E. Hofstetter et al, "A New Technique for the Design of Nonrecursive Digital

Filters," Proc. 5th Annual Princeton Conference Information Science Systems, 1971, pp. 63–72.
13. J. F. Kaiser and R. W. Hamming, "Sharpening the Response of a Symmetric Nonrecursive Filter by Multiple Use of the Same Filter," IEEE Trans. Acoustics Speech and Signal Processing, *ASSP-25,* No. 5 (October 1977), pp. 415–422.
14. O. Hermann and W. Schussler, "On the Accuracy Problem in the Design of Nonrecursive Digital Filter," Arch Elek. Uber., *24* (1970), pp. 525–526.
15. A. Gersho, "Charge Transfer Filtering," Proc. IEEE (February 1979).
16. S. M. Kang et al, "An Optimal Design of Split-Electrode CCD Transversal Filters," Record International Symposium on Circuits and Systems, New York, May 1978.
17. Intel MCS-85/MCS-86 Users Manual, 1978.
18. Advanced Micro Devices, The AM2900 Family Data Book, 1978.
19. W. Nicholson et al, "The S2811 Signal Processing Peripheral," Wescon, 1978.
20. D. Chan and L. Rabiner, "Analysis of Quantization Errors in the Direct Form for Finite Impulse Response Digital Filters," IEEE Trans. Audio and Electroacoustics, *AU-21* (August 1973), pp. 354–366.
21. A. Gersho, B. Gopinath, and A. M. Odlyzko, "Coefficient Inaccuracy in Transversal Filtering," B.S.T.J. *58,* No. 10 (December 1979), pp. 2301–2316.
22. U. Heute, "Necessary and Efficient Expenditure for Non-Recursive Digital Filters in Direct Structures," European Conf. on Circuit Theory and Design, IEEE Conf. Pub., No. 116 (July 1974), pp. 13–19.
23. *IBM Program Product Manual*—Mathematical Programming System Extended (MPSX) Mixed Integer Programming (MIP) Program Description, Second Edition, August 1973.
24. R. D. Gitlin and S. B. Weinstein, "On the Required Tap-Weight Precision for Digitally Implemented, Adaptive, Mean-Squared Equalizers," B.S.T.J., *58,* No. 2 (February 1979), pp. 301–321.
25. G. Kochman, "Computer Programs for Decomposition in Integer Programming," Stanford Tech. Report, No. 71, September 1976.
26. T. G. Moore, "Solving Large 0-1 Problems," ORSA/TIMS, Miami Beach, November 1976.
27. L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Englewood Cliffs, N.J.: Prentice-Hall, 1975.
28. J. R. Boddie et al, "A Digital Signal Processor for Telecommunications Applications," Digest of ISSCC 1980, February 1980.