ACKNOWLEDGMENT

The authors wish to thank M. J. Britt for contributions to the initial design of the radix-2 processor (in particular the serial vector multiplier).

REFERENCES

[1] M. J. Corinthios, "The design of a class of fast Fourier transform computers," *IEEE Trans. Comput.*, vol. C-20, pp. 617–623, June 1971.
[2] ——, "A fast Fourier transform for high-speed signal processing," *IEEE Trans. Comput.*, vol. C-20, pp. 843–846, Aug. 1971.
[3] M. L. Groginsky and G. A. Works, "A pipeline fast Fourier transform," *IEEE Trans. Comput.*, vol. C-19, pp. 1015–1019, Nov. 1970.
[4] G. D. Bergland, "Fast Fourier transform hardware implementations—an overview," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 104–108, June 1969.
[5] R. C. Singleton, "A short bibliography on the fast Fourier transform," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 166–169, June 1969.
[6] M. C. Pease, "An adaptation of the fast Fourier transform for parallel processing," *J. Ass. Comput. Mach.*, vol. 15, pp. 252–264, Apr. 1968.

Linear Programming Design of IIR Digital Filters with Arbitrary Magnitude Function

LAWRENCE R. RABINER, MEMBER, IEEE, NANCY Y. GRAHAM, AND HOWARD D. HELMS, SENIOR MEMBER, IEEE

Abstract—**This paper discusses the use of linear programming techniques for the design of infinite impulse response (IIR) digital filters. In particular, it is shown that, in theory, a weighted equiripple approximation to an arbitrary magnitude function can be obtained in a predictable number of applications of the simplex algorithm of linear programming. When one implements the design algorithm, certain practical difficulties (e.g., coefficient sensitivity) limit the range of filters which can be designed using this technique. However, a fairly large number of IIR filters have been successfully designed and several examples will be presented to illustrate the range of problems for which we found this technique to be useful.**

INTRODUCTION

A LARGE NUMBER of techniques are available for designing infinite impulse response (IIR) digital filters [1], [2]. The techniques of impulse invariant design, bilinear transformation, and matching poles and zeros [3], for transforming a given analog filter to an "equivalent" digital filter are well known and widely used. These techniques, however, are limited in that they are generally applied only to the case of transforming standard analog filters—e.g., low-pass, bandpass, bandstop, or high-pass filters. When one is interested in designing a digital filter with a nonstandard frequency response, i.e., one which has not been exhaustively studied, then some algorithmic (as opposed to closed form) design procedure is generally used. Several frequency domain (e.g., [4]–[7]) and one time domain algorithmic design procedure [8] have been recently proposed for designing IIR filters. One difficulty with almost all of these procedures is that convergence of the optimization procedure that is used to design the filter is not guaranteed, and even when the procedure converges, the optimality of the resulting filter is also not guaranteed. In this paper a frequency domain IIR filter design procedure is discussed which uses linear programming techniques (the simplex algorithm) to choose filter coefficients to approximate an arbitrary magnitude characteristic. Theoretical convergence of the optimization algorithm is guaranteed, and the resulting filter can be shown to be optimal in the given design sense (e.g., minimum absolute weighted error). The optimization algorithm itself has been designed to minimize the number of applications of the simplex algorithm.

The next section presents the design procedure with a discussion of the practical aspects of implementing the method. Following this, several representative filter designs are given. Finally, some discussion is given as to practical limitations in using the method.

Manuscript received December 13, 1973.
L. R. Rabiner and N. Y. Graham are with Bell Laboratories, Murray Hill, N. J. 07974.
H. D. Helms is with Bell Laboratories, Holmdel, N. J. 07733.

THEORY

Let $H(z)$ be the transfer function of an IIR digital filter. Assume $H(z)$ has the form

$$H(z) = \frac{N(z)}{D(z)} = \sum_{i=0}^{m} b_i z^{-i} \Big/ \sum_{i=0}^{n} a_i z^{-i} \qquad (1)$$

where the numerator polynomial $N(z)$ is of mth degree, and the denominator polynomial $D(z)$ is of nth degree. The a_0 term in (1) can be set to 1.0 without any loss in generality. The magnitude response of the filter is obtained by evaluating (1) on the unit circle (i.e., for $z = \exp[j\omega]$),[1] and taking its magnitude, thus giving

$$|H(\exp[j\omega])| = \left|\frac{N(\exp[j\omega])}{D(\exp[j\omega])}\right|$$

$$= \left|\sum_{i=0}^{m} b_i \exp[-j\omega i] \Big/ \sum_{i=0}^{n} a_i \exp[-j\omega i]\right|. \quad (2)$$

In many frequency domain filter design problems it is desired that the magnitude of the resulting filter approximate a given magnitude function, $M(\exp[j\omega])$, to within a tolerance $G(\omega,\delta)$, where $G(\omega,\delta)$ is a monotically increasing function of δ for fixed ω.[2] Thus, the resultant approximation problem is to choose the filter coefficients (the a_i's and b_i's) to minimize the quantity δ consistent with that constraint inequality

$$\left|\left|\frac{N(\exp[j\omega])}{D(\exp[j\omega])}\right| - M(\exp[j\omega])\right| \leq G(\omega,\delta). \quad (3)$$

Inequality (3) is generally evaluated over a union of disjoint subintervals of the band $0 \leq \omega \leq \pi$.

The above approximation problem is a nonlinear one in that the filter coefficients enter into the constraint equation nonlinearity. Although various techniques have been proposed for solving this nonlinear problem (e.g., [4] and [5]), a linear approximation problem can be defined by considering the magnitude squared function of the filter. From (1) we get the relation

$$H(z)H(z^{-1}) = \frac{N(z)N(z^{-i})}{D(z)D(z^{-1})} \quad (4)$$

$$= \left(\sum_{i=0}^{m} b_i z^{-i}\right)\left(\sum_{j=0}^{m} b_j z^{+j}\right) \Big/ \left(\sum_{i=0}^{n} a_i z^{-1}\right)\left(\sum_{j=0}^{n} a_j z^{+j}\right) \quad (5)$$

$$= \sum_{i=-m}^{m} c_i z^{-i} \Big/ \sum_{i=-n}^{n} d_i z^{-i} \quad (6)$$

where

$$c_i = c_{-i} \qquad i = 1,2,\cdots,m \quad (7)$$

$$d_i = d_{-i} \qquad i = 1,2,\cdots,n. \quad (8)$$

The magnitude squared function of the filter is obtained by evaluating (6) on the unit circle giving

$$|H(\exp[j\omega])|^2 = H(z)H(z^{-1})\,|_{z=\exp[j\omega]} = \frac{\hat{N}(\omega)}{\hat{D}(\omega)} \quad (9)$$

$$= \left[c_0 + \sum_{i=1}^{m} 2c_i \cos(\omega i)\right] \Big/ \left[d_0 + \sum_{i=1}^{n} 2d_i \cos(\omega i)\right]. \quad (10)$$

(Again the d_0 term in (10) can be set to 1.0 without loss of generality.) Equation (10) shows that the magnitude squared function of the filter is a ratio of trigonometric polynomials. It is also seen that both $\hat{N}(\omega)$, the numerator polynomial, and $\hat{D}(\omega)$, the denominator polynomial, are linear in the unknown filter coefficients $\{c_i\}$ and $\{d_i\}$. It is now shown how linear programming techniques can be used to determine the c_i's and d_i's such that $|H(\exp[j\omega])|^2$ approximates a given magnitude squared characteristic $F(\omega)$ where the peak weighted error of approximation is minimized—i.e., the weighted error is an equiripple function.

If we let $F(\omega)$ be the desired magnitude squared characteristic, then the approximation problem consists of finding the filter coefficients such that

$$-\epsilon(\omega) \leq \frac{\hat{N}(\omega)}{\hat{D}(\omega)} - F(\omega) \leq \epsilon(\omega) \quad (11)$$

where $\epsilon(\omega)$ is a tolerance function on the error which allows for unequal weighting of errors as a function of frequency. Since $F(\omega)$ and $\epsilon(\omega)$ are generally specified functions of frequency, (or depend on some parameter in a manner explained below), (11) can be expressed as a set of linear inequalities in the c_i's and d_i's by writing it in the form

$$\hat{N}(\omega) - \hat{D}(\omega)F(\omega) \leq \epsilon(\omega)\hat{D}(\omega)$$

$$-\hat{N}(\omega) + \hat{D}(\omega)F(\omega) \leq \epsilon(\omega)\hat{D}(\omega) \quad (12)$$

or

$$\hat{N}(\omega) - \hat{D}(\omega)[F(\omega) + \epsilon(\omega)] \leq 0 \quad (13)$$

$$-\hat{N}(\omega) + \hat{D}(\omega)[F(\omega) - \epsilon(\omega)] < 0. \quad (14)$$

The additional linear inequalities

$$-\hat{N}(\omega) \leq 0 \quad (15)$$

$$-\hat{D}(\omega) \leq 0 \quad (16)$$

completely define the approximation problem.

Thus, the question of whether or not there exists a digital filter with magnitude squared characteristic $F(\omega)$ and tolerance function $\epsilon(\omega)$ is equivalent to the question of whether or not there exists a set of filter coefficients satisfying the system of constraints defined by (13)–(16). The question can be answered by using linear programming techniques [9]. First, an auxiliary variable ν is subtracted from the left side of each constraint, forming the new set of constraints

$$\hat{N}(\omega) - \hat{D}(\omega)[F(\omega) + \epsilon(\omega)] - \nu \leq 0 \quad (17)$$

$$-\hat{N}(\omega) + \hat{D}(\omega)[F(\omega) - \epsilon(\omega)] - \nu \leq 0 \quad (18)$$

$$-\hat{N}(\omega) - \nu \leq 0 \quad (19)$$

$$-\hat{D}(\omega) - \nu \leq 0. \quad (20)$$

[1] The quantity ω is the normalized frequency variable (i.e., the sampling period, T, is assumed to be 1.0).

[2] It should be noted that the function $G(\omega,\delta)$ is generally determined as soon as $M(\exp[j\omega])$ is specified by the designer, as will be seen in the examples later in the paper.

The objective function

$$z = \nu \tag{21}$$

is chosen to be minimized under the constraints of (17)–(20).[3] Clearly a solution to constraints (13)–(16) exists if and only if the minimum value of z under constraints (17)–(20) is zero. If the minimum value of ν is 0, then a solution exists to the approximation problem and the filter coefficients may be obtained directly as the output of the linear programming routine. If $\nu > 0$, then no solution to the approximation problem exists, and either $F(\omega)$, or $\epsilon(\omega)$, or both must be modified in order to obtain a solution.

To illustrate the above procedure, consider the design of a low-pass filter. If we let δ be the peak approximation error in the stopband, and $K\delta$ (K is a constant expressing the ratio of passband to stopband ripple) be the peak approximation error in the passband, then the magnitude function for an equiripple error approximation is as shown in Fig. 1 (a). The quantity δ is unknown and is to be minimized in the ultimate design program. (Of course in this case the resulting filter is an elliptic filter which can be readily designed in closed form, but we are only using this as an example of how to apply the design technique.) The passband cutoff frequency is $\omega_p = 2\pi F_p$ and the stopband cutoff frequency is $\omega_s = 2\pi F_s$. The magnitude squared function of the filter is the square of the response in Fig. 1 (a) and is shown in Fig. 1 (b). This magnitude squared function can be viewed as a weighted equiripple approximation of the function $F(\omega)$ [shown in Fig. 1 (c)], with peak approximation error $\epsilon(\omega)$ [shown in Fig. 1 (d)], defined by

$$\begin{aligned} F(\omega) &= 1 + K^2\delta^2 && 0 \le \omega \le \omega_p \\ &= \delta^2/2 && \omega_s \le \omega \le \pi \end{aligned} \tag{22}$$

$$\begin{aligned} \epsilon(\omega) &= 2K\delta && 0 \le \omega \le \omega_p \\ &= \delta^2/2 && \omega_s \le \omega \le \pi. \end{aligned} \tag{23}$$

It is easily verified that $|H(\exp[j\omega])|^2$ of Fig. 1 (b) is less than or equal to $F(\omega) + \epsilon(\omega)$ and greater than or equal to $F(\omega) - \epsilon(\omega)$ in both the passband and the stopband.

To determine the smallest value of δ such that the filter approximation problem has a solution (i.e., to find the δ of the elliptic filter) an iterative procedure must be used since δ enters into the design constraints in a nonlinear manner. If we let δ^* denote the minimum value of δ for which the approximation problem has a solution, then δ^* satisfies the inequality

$$0 < \delta^* \le \frac{1}{K+1} \tag{24}$$

since the sum of passband ripple ($K\delta^*$) and stopband ripple (δ^*) must be less than or equal to 1.0 because otherwise the passband and stopband would no longer be well defined. Based on (24), a binary search may be used to locate δ^*, as in the following procedure. (This procedure is different from the usual binary search in that the search is performed on the log of δ instead of δ itself.)

Step 1: Let δ_+^i and δ_-^i denote initial upper and lower bounds for δ^*. For example, choose $\delta_+^i = (K+1)^{-1}$, $\delta_-^i = 10^{-8}$ (since $\delta_-^i < 10^{-8}$ is unrealistic). Initialize δ at $(\delta_+^i\delta_-^i)^{1/2}$, the geometric mean of the *initial* upper and lower bounds. (Note that the geometric mean of two quantities is equal to the arithmetic mean of the logs of these quantities.)

Step 2: Solve the linear programming problem (17)–(21) with this value of δ. If $z = 0$ a solution to the approximation problem exists and $\delta^* < \delta$. In this case set $\delta_+ = \delta$. Otherwise no solution exists for this value of δ and $\delta < \delta^*$; in which case set $\delta_- = \delta$.

Step 3: Set $\delta = (\delta_+\delta_-)^{1/2}$ and repeat Step 2. This procedure is iterated until a predetermined accuracy criterion in locating δ^* is satisfied.

It should be noted that, when δ^* is small ($<10^{-2}$) (as in practical filter design problems), choosing δ to be the geometric mean of the upper and lower bounds for δ^* will result in a smaller number of iterations required to achieve relative accuracy in locating δ^* than required by the usual

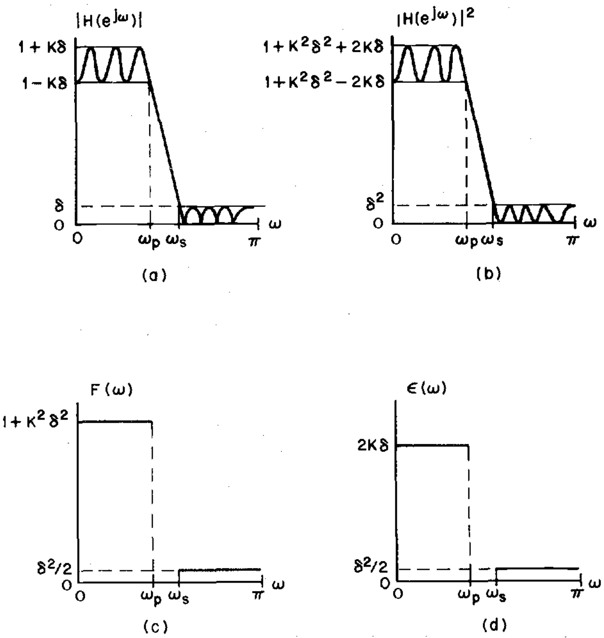


Fig. 1. Specifications for designing an equiripple error low-pass filter. (a) Bounds on the approximation. (b) Bounds on the square of the approximation. (c) Function obtained by averaging the upper and lower bounds on the square of the approximation. (d) Error bounds obtained by subtracting Fig. 1(c) from Fig. 1(b).

[3] The linear programming problem defined above (by constraints (17)–(20) and objective function $z = \nu$) may be solved by a straightforward application of the simplex (or revised simplex) algorithm. However, since the number of constraints is generally much larger than the number of filter coefficients, it is of course much more efficient to apply the simplex algorithm to the dual linear programming problem. Furthermore, since the range of values of filter coefficients, for the examples of interest here, is between -1 and $+1$, a change of variables to $c_i' = c_i + 1$, $d_i' = d_i + 1$ is performed before application of the simplex algorithm, which does not allow variables to assume negative values.

TABLE I
RESULTS ON THE DESIGN OF SEVERAL LOW-PASS FILTERS

Filter No.	n	m	F_p	F_s	K	$-20 \log_{10} \delta^*$	$-20 \log_{10} \delta_+$	No. of Iterations	Total Run Time (s)
1	4	4	0.30	0.35	5.8	37.9	37.8	11	40.6
2	4	4	0.15	0.18	2.0	29.8	29.9	11	43.5
3	4	4	0.10	0.15	12.0	43.1	42.9	11	44.2
4	4	4	0.10	0.14	6.5	38.7	38.6	11	46.9
5	4	4	0.10	0.13	3.4	34.0	33.9	11	48.2
6	4	4	0.10	0.12	1.7	28.5	28.4	11	50.6
7	6	6	0.20	0.23	8.5	48.3	48.4	11	134.3
8	6	6	0.20	0.25	71.9	61.6	61.8	11	141.7

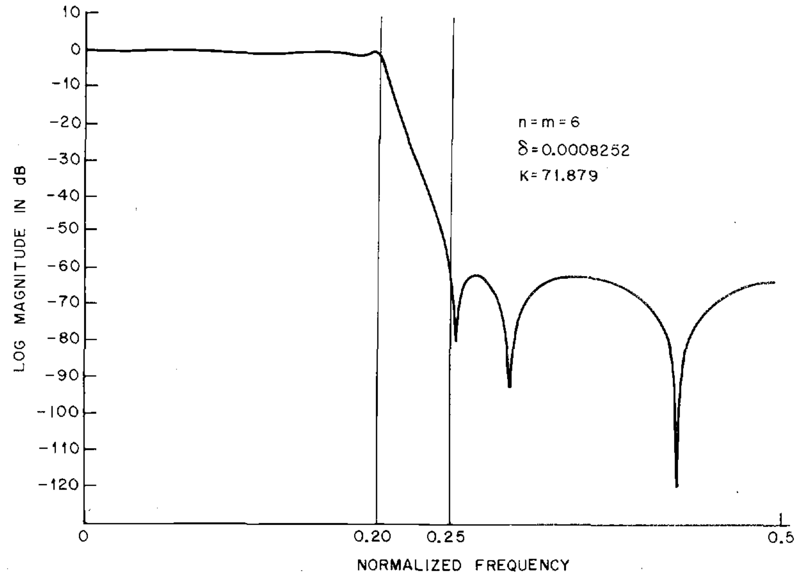


Fig. 2. Log magnitude response of a sixth order low-pass filter designed using linear programming methods.

binary search in which δ is chosen to be the arithmetic mean of the upper and lower bounds. In particular, the number n of iterations required for a maximum relative error p (where p is defined as

$$p = \frac{\delta_+^f - \delta_-^f}{\delta_-^f} \tag{25}$$

with δ_+^f and δ_-^f being the final upper and lower bounds determined in step 3 above) can be shown to be of the form

$$n = \left[\log_2 \left(\frac{\log_{10}(\delta_+^i) - \log_{10}(\delta_-^i)}{\log_{10}(\delta_+^f) - \log_{10}(\delta_-^f)}\right)\right] + 1 \tag{26}$$

$$= \left[\log_2 \left(\frac{\log_{10}(\delta_+^i) - \log_{10}(\delta_-^i)}{\log_{10}(1+p)}\right)\right] + 1 \tag{27}$$

where $[y]$ denotes the largest integer $\leq y$.

Theoretically the value of δ^* may be bounded as tightly as desired by using a large number of iterations thereby reducing the tolerance (difference between δ_+ and δ_-) at much as desired. In practice, a relative error of $p = 0.01 =$ 1 percent on the deltas is sufficient for most problems.

The values of c_i and d_i associated with the final value of δ_+ are used in the polynomials in z (6), which are then factored. A z-transform, which is stable, and minimum phase is obtained by retaining only zeros and poles which lie inside or on the unit circle. (If zeros lie on the unit circle, only half of the pairs are retained.)

The extension of the above procedure to other types of filters other than low-pass filters is straightforward and will not be discussed here. In the next section we present examples of several filters which were designed using the above iterative procedure.

FILTER EXAMPLES

In order to test out the procedure a number of filters were designed. Table I gives the results for a set of 8 low-pass filters. The data in this table correspond to the 8 low-pass filters designed by Swanton [10] using linear programming methods in a sequence of individual numerator and denominator optimization iterations. These data are for low-order filters (4th or 6th order). The values for F_p and F_s are the corresponding filter cutoff frequencies where $F_p = \omega_p/2\pi$ and $F_s = \omega_s/2\pi$. The quantity $20 \log_{10} \delta^*$ is the theoretical stopband attenuation for the elliptic filter and $20 \log_{10} \delta_+$ is the actual attenuation for the filter designed using the linear programming technique described above. Table I also gives the number of iterations on the delta's (for a 1 percent tolerance on δ) and the overall run time on a Honeywell 6000 Computer. It is seen from Table I that the resulting filters meet approximately the same specifications as the equivalent elliptic filter. To further illustrate these results, Fig. 2 shows the log magni-

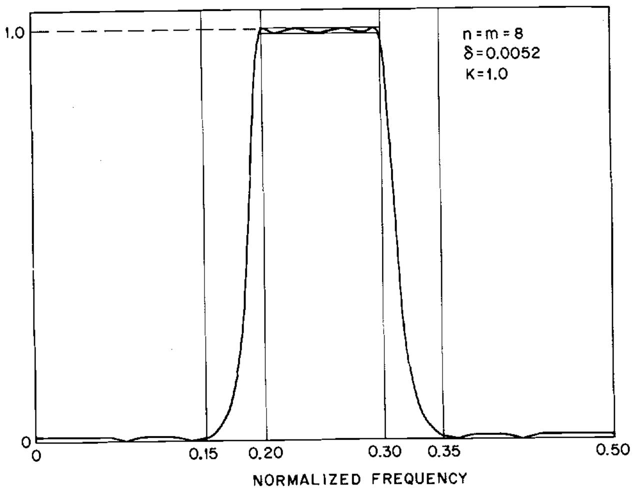


Fig. 3. Magnitude response of an eighth order bandpass filter designed using linear programming techniques.

tude response of the eighth filter in the table which achieved a $\delta = \delta_+$ of 0.0008144 or a stopband attenuation of 61.8 dB.

As seen in Table I the run time for the simple 4th order examples was about 45 s whereas for the two 6th order examples it was about 138 s. These times could be significantly reduced by an improved initial guess of δ, or by relaxing the convergence criterion (the 1 percent tolerance on the deltas). These were not done for the examples in Table I in order to verify that the procedure would converge without any starting information.

In addition to low-pass filters, several 3, 4 and 5 band filters, and wide-band differentiators have been designed with this procedure. Figs. 3–7 show the frequency responses of some typical filters which were designed. Fig. 3 gives an example of a standard bandpass filter of eighth order (i.e., $n = m = 8$). This filter took 11 iterations to achieve the desired 1 percent accuracy, and required 187 s of processor time. The value of K was 1.0 giving a value of δ of 0.0051. The two filter stopbands were from 0.0 to 0.15 and from 0.35 to 0.5. Fig. 3 shows the magnitude response of the filter including the tolerances in each of the bands and the filter band edges.

Fig. 4 shows the frequency response of a 4 band filter with a stopband followed by 2 disjoint passbands, followed by another stopband. The filter is of eighth order and the tolerances in all the bands were the same. It required 11 iterations for convergence of the delta to within 1 percent and took 238 s of computation. Fig. 4(a) shows the log magnitude response of the filter; Fig. 4 (b) shows the positions of the poles and zeros in the z-plane; and Fig. 4(c) shows the group delay response of this filter. The pole and zero positions of the filter are obtained by factoring the denominator and numerator polynomials of the magnitude squared function of the filter and assigning poles inside the unit circle to the resulting denominator, and zeros inside or on the unit circle to the resulting numerator. (Generally the zeros of the magnitude squared function will be on the unit circle in pairs. Thus one of each pair of zeros on the unit circle is assigned to the resulting filter.)

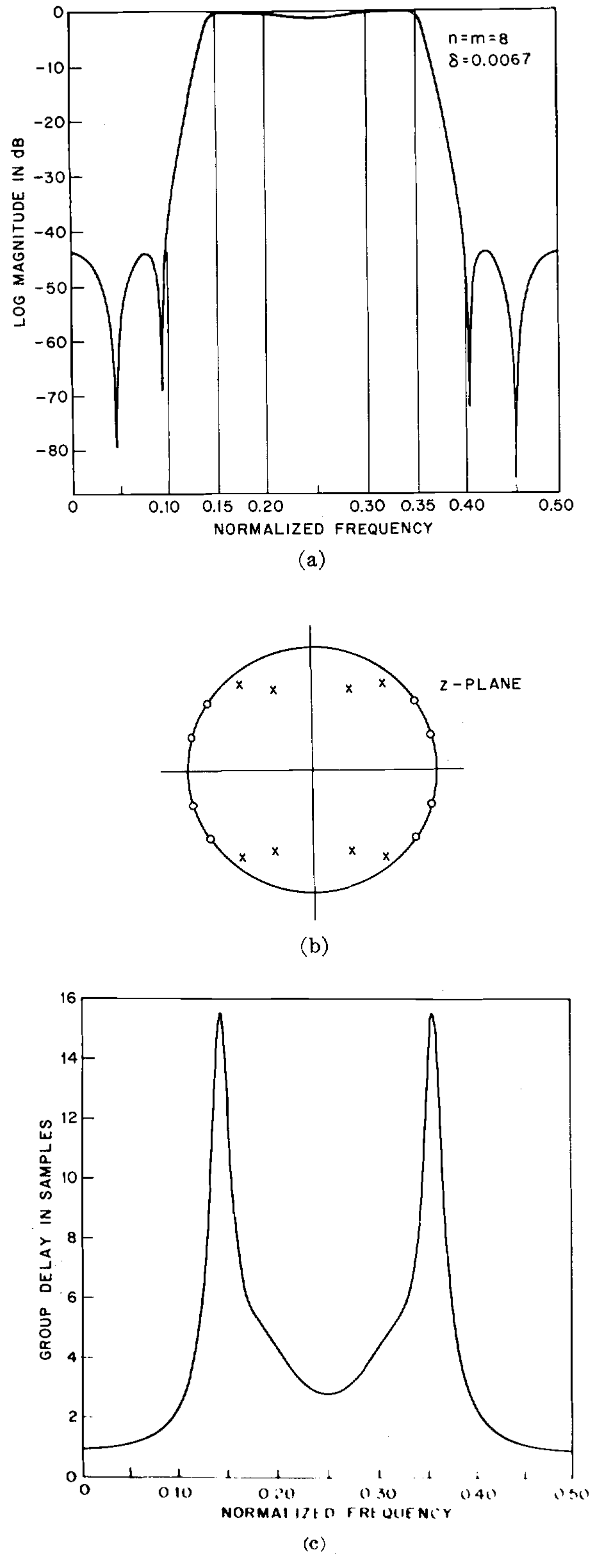


Fig. 4. Log magnitude response, z-plane pole–zero diagram, and group delay response of an eight order-four band filter.

Fig. 5 shows an example of a 5 band filter with 2 passbands and 3 stopbands. An eight order filter gave a delta of 0.035 with equal weighing in each of the five bands. The design procedure took 11 iterations and required 224 s to design.

Fig. 6 shows another 5 band filter with the arbitrary specifications:

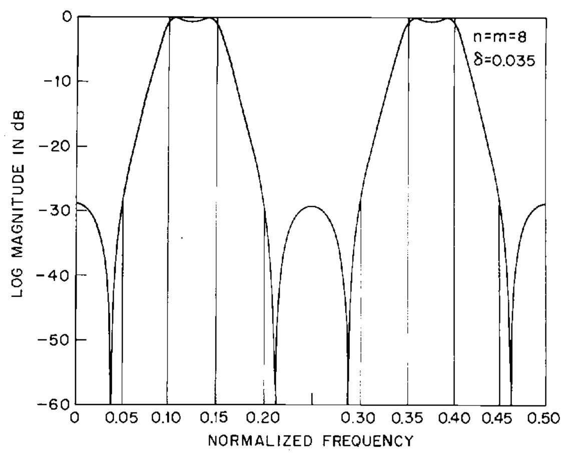
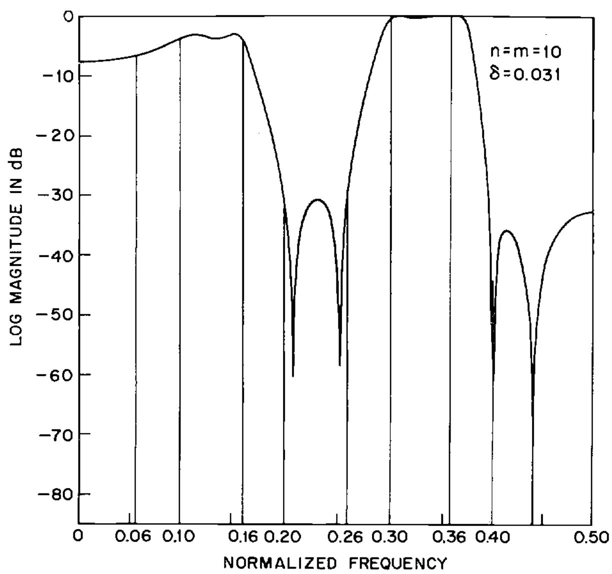
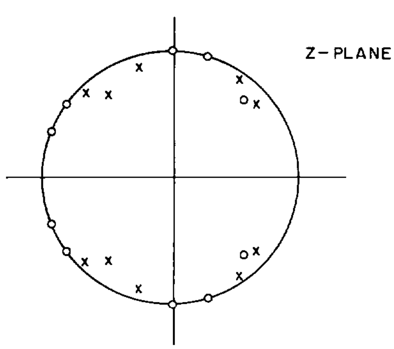


Fig. 5. Log magnitude response of an eight order filter with 5 alternating stopbands and passbands.



(a)



(b)

Fig. 6. Log magnitude response and z-plane pole–zero positions of a tenth order five band filter with arbitrary magnitude specifications.

$$
\begin{aligned}
|H(\exp[j2\pi f])| &= 0.5 && 0.00 \le f \le 0.06 \\
&= 0.75 && 0.10 \le f \le 0.16 \\
&= 0.0 && 0.20 \le f \le 0.26 \\
&= 1.0 && 0.30 \le f \le 0.36 \\
&= 0.0 && 0.40 \le f \le 0.50
\end{aligned}
$$

and with equal tolerances on the error in each band. The resulting filter was of 10th order and had a δ of 0.0303. Even though the filter met specifications in all the bands, the magnitude squared function blew up between the fourth and fifth bands because of a pole which was on the unit circle. The simple expedient of moving the pole slightly inside the unit circle gave a stable filter whose log magnitude response and pole–zero positions are plotted in Fig. 5. It should be noted that the presence of a pole on the unit circle in one of the unconstrained regions of the frequency scale is not a violation of the design procedure and is perfectly acceptable to the optimization procedure. Of course the resulting filter cannot tolerate such a situation.

Finally, Fig. 7 shows the error response of a differentiator of fourth degree designed using the technique. In this case the error criterion was a minimum relative error criterion. The resulting value of δ was 0.00000763 when the desired band for differentiation was from 0 to 0.45. (In this case δ represents the peak relative error for the differentiator.) This example required 11 iterations and took 51 s of computation time.

DISCUSSION

The preceding examples have shown that the linear programming method does, in many cases, give reasonable solutions for IIR filters which approximate arbitrary magnitude specifications with arbitrary weighing of the error function. In this section we discuss what we believe are the practical limitations of this technique.

One of the major difficulties with the proposed method is that one is forced to work with magnitude squared characteristics to solve for the filter coefficients. Kaiser [11] has shown that an extreme coefficient sensitivity problem exists for sharp cutoff filters when implemented in the direct form. This coefficient sensitivity is aggravated by using magnitude squared functions rather than the magnitude function itself. Thus the results derived by Kaiser, along with our own practical experience indicate the following.

1) Sharp cutoff filters are difficult to design with this procedure. Thus, if the width of a transition band is small, the coefficient sensitivity will make the procedure unstable.

2) High-order filters are difficult to design. Filters with order greater than about 12 cannot readily be designed with double precision arithmetic on a 36 bit word length computer since the high order polynomial coefficients are extremely sensitive to small changes in the filter specifications.

3) Filters with deltas on the order of 10^{-5} or less cannot generally be designed even with double precision arithmetic since, a tolerance of the magnitude function on the order of $\le 10^{-5}$ in a band implies a tolerance of the magnitude squared function on the order of $\le 10^{-10}$ in that band. The attainment of such a high degree of numerical precision is of course limited by the precision capabilities of the computer. Experience has shown that the linear programming routine required to solve for the filter coefficients must be implemented in double precision

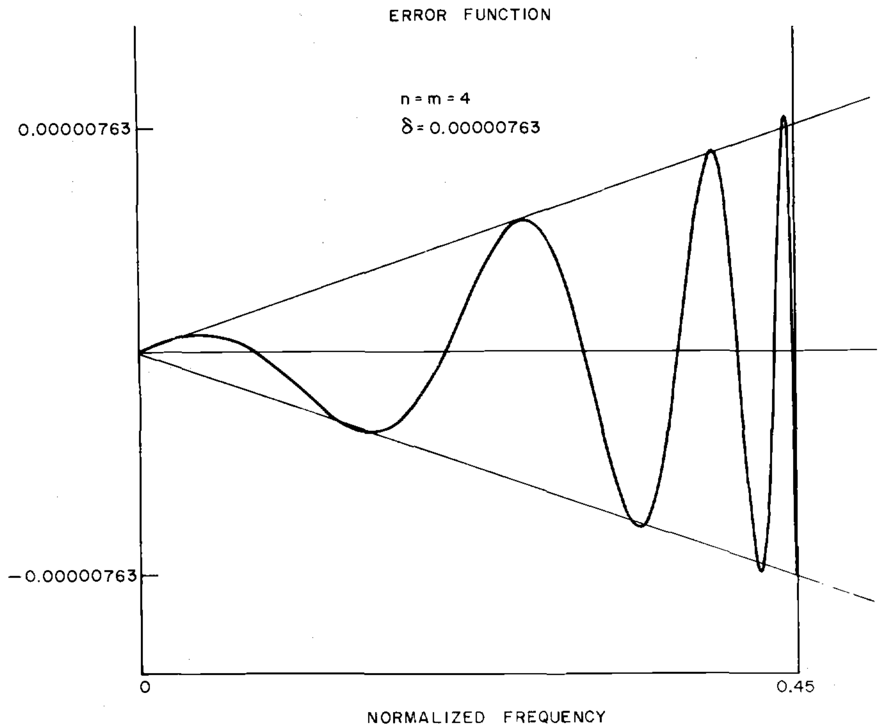


Fig. 7. Error function of a fourth order differentiator with minimum relative error over the band $0 \leq f \leq 0.45$.

arithmetic in order to obtain sufficient accuracy for deltas in the range $10^{-1} > \delta \geq 10^{-5}$, whereas for deltas $< 10^{-5}$, sufficient accuracy is almost impossible to consistently obtain even with double precision arithmetic on a 36-bit computer.

4) Experience has also shown that for filter design problems with ≥ 8 degrees of freedom in numerator and denominator polynomials, the number of pivots within a given application of the simplex algorithm may be exceedingly large, say > 400.

Fortunately, despite the above practical limitations, there is a large class of problems where the linear programming technique can be used to advantage. One of the key properties here is the guaranteed theoretical convergence and the clear statement of optimality of the resulting approximation. Also, compilers which permit arbitrary precision arithmetic may circumvent the above limitations, although at the cost of greatly increased computation time.

SUMMARY

A new technique for designing IIR filters which can approximate arbitrary magnitude specifications was presented. The technique sets up a linear programming problem which is solved iteratively for the best approximation to the given specifications. Several examples of filters designed using this technique were given and the ultimate limitations of the procedure were discussed.

ACKNOWLEDGMENT

The authors wish to thank J. E. Heller of Bell Laboratories, Whippany, N. J. and W. L. Roach of Bell Laboratories, Holmdel, N. J. for their comments and suggestions.

REFERENCES

[1] J. F. Kaiser, "Digital filters," in *System Analysis by Digital Computer*, F. F. Kuo and J. F. Kaiser, Ed. New York: Wiley, 1966.

[2] B. Gold and C. M. Rader, *Digital Processing of Signals*. New York: McGraw-Hill, 1969.

[3] R. M. Golden, "Digital filter synthesis by sampled-data transformation," *IEEE Trans. Audio Electroacoust.*, vol. AU-16, pp. 321–329, Sept. 1968.

[4] K. Steiglitz, "Computer-aided design of recursive digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, pp. 123–129, June 1970.

[5] A. G. Deczky, "Synthesis of recursive digital filters using the minimum *p*-error criterion," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 257–263, Oct. 1972.

[6] P. Thajchayapong and P. J. Rayner, "Recursive digital filter design by linear programming," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 107–112, Apr. 1973.

[7] F. X. Brophy and A. C. Salazar, "Recursive digital filter synthesis in the time domain," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-22, pp. 45–56, Feb. 1974.

[8] C. S. Burrus and T. W. Parks, "Time domain design of recursive digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, pp. 137–141, June 1970.

[9] H. L. Loeb, "Algorithms for Chebyshev approximations using the ratios of linear forms," *J. Soc. Ind. Appl. Math.*, vol. 8, pp. 458–465, Sept. 1960.

[10] D. Swanton, "Linear programming design of recursive digital filters," M.Sc. Thesis, McGill Univ., Mar. 1973.

[11] J. F. Kaiser, "Some practical considerations in the realization of linear digital filters," in *Proc. 3rd Annu. Allerton Conf. on Circuit and System Theory*, 1965, pp. 621–623.