FIR Filter Design Over a Discrete Powers-of-Two Coefficient Space

YONG CHING LIM, MEMBER, IEEE, AND SYDNEY R. PARKER, FELLOW, IEEE

Abstract—**FIR digital filters with discrete coefficient values selected from the powers-of-two coefficient space are designed using the methods of integer programming. The frequency responses obtained are shown to be superior to those obtained by simply rounding the coefficients. Both the weighted minimax and the weighted least square error criteria are considered. Using a weighted least square error criterion, it is shown that it is possible to predict the improvement that can be expected when integer quadratic programming is used instead of simple coefficient rounding.**

I. Introduction

RECENTLY, there has been much discussion regarding the use of integer programming to design FIR digital filters with discrete coefficients [1]-[6], [11], [12]. The discrete coefficient space discussed in [1]-[4] is the "integer space" or the "finite word length space" where each coefficient value is represented by a finite number of bits. The ability to produce an optimum finite word length solution has led to the use of integer programming for the design of FIR filters. However, integer programming suffers from the following disadvantages.

1) The optimum finite word length solution obtained by using integer programming saves only a few bits in coefficient word length when compared to the solution obtained by simple coefficient rounding.

2) The high computing cost required by integer programming limits the maximum size of the filter that can be designed optimally to a length of about 40. Computing cost increases exponentially with filter length.

On the other hand, integer programming becomes eminently useful when the discrete coefficient space is the powers-of-two space. A first attempt at using integer programming to design FIR filters whose coefficients are integer powers of 2 has been reported in [5]. Although the methods used in [5] are rather elementary and suboptimal, the results obtained are dramatic. The fundamental principle of integer programming is applicable for designing filters with either the integer or the powers-of-two coefficient space. However, there are significant differences between the two. These differences render most general-purpose integer programming packages unsuitable for designing filters with powers-of-two coefficient space.

In Sections II and III, we discuss the methods of integer programming with reference to integer linear programming and integer quadratic programming. Special attention is placed on transforming the filter design problem into a suitable mathematical programming problem, and on adapting general-purpose linear programming and quadratic programming packages to the design of filters with powers-of-two coefficient values.

In Section IV, we compare the performance of filters designed using integer programming to those designed by simple rounding of coefficient values. In each case, the discrete coefficient space is the powers-of-two space. The criteria of comparison are the weighted minimax criteria and the weighted least square criteria.

We also introduce the concept of discrete optimizability which serves as a measure of how much can be gained by performing discrete optimization compared to simple coefficient rounding. A problem is said to have a high discrete optimizability if much can be gained from the use of discrete space optimization techniques. In Section V, it is shown that there exists a set of eigenvalues and eigenvectors which provide valuable information as to the discrete optimizability of a design when the criterion for optimization is in the least square sense. In the minimax sense, *a priori* knowledge of the discrete optimizability is difficult to obtain. However, in Section VI, we present some insight into the discrete optimizability for the latter case.

II. Linear Phase FIR Filter Design Using Linear Programming and Quadratic Programming

The frequency response $H(\omega)$ of an FIR linear phase filter of length N may be expressed as a trigonometric function of frequency ω. Omitting the linear phase factor $e^{-j(N-1/2)\omega}$, the frequency response of a symmetrical impulse response filter with N odd is given by

$$H(\omega) = a(0) + 2 \sum_{n=1}^{(N-1)/2} a(n) \cos \omega n. \tag{1}$$

The impulse response $h(n)$ is related to $a(n)$ by

$$h(n) = a\left(\frac{N-1}{2} - n\right) = h(N-1-n). \tag{2}$$

Manuscript received November 20, 1981; revised September 29, 1982. This work was supported by the National Research Council, the Naval Postgraduate School, the Office of Naval Research, the Naval Electronics Systems Command, the Imperial College of the University of London, and the National University of Singapore.

Y. C. Lim is with the Department of Electrical Engineering, National University of Singapore, Singapore 0511.

S. R. Parker is with the Department of Electrical Engineering, Naval Postgraduate School, Monterey, CA 93940.

The relationship between the frequency response and the impulse response for antisymmetrical impulse response filters may be found in standard texts [7]. The filter design problem is one of obtaining a set of coefficients $a(n)$ such that $H(\omega)$ is the best approximation to some desired function $D(\omega)$, over a given frequency range, with respect to some optimality criterion. The frequency band in which $H(\omega)$ may take on any value is the transition band. The optimality criterion defines the class of mathematical programming problem which must be solved.

In the minimax sense, the value of $H(\omega)$ is subject to the constraints

$$H(\omega) \leqslant D(\omega) + \delta k(\omega)$$
$$H(\omega) \geqslant D(\omega) - \delta k(\omega) \tag{3}$$

where $\delta k(\omega)$ is the ripple to be minimized. Evaluating (3) on a dense frequency grid allows the optimization to be formulated as a linear programming problem [8], [9]. If only a finite number of bits Q are allowed for each coefficient value, the optimization problem reduces to determining the set of integer values for $a(n)$ which meet the following conditions:

$$H(\omega) \leqslant 2^Q D(\omega) + k(\omega)\delta$$
$$H(\omega) \geqslant 2^Q D(\omega) - k(\omega)\delta \tag{4}$$

with a minimum value of δ. This is an integer linear programming problem [9], [10] and may be solved by any good general-purpose integer linear programming package. The solution of (4) is a dual feasible but degenerate problem, and the integer linear programming package should anticipate its dual degeneracy. Experiences in using integer linear programming packages to solve (4) have been reported in [1]-[4]. The reason the finite coefficient word length design problem may be cast into the form of (4) is that by scaling by 2^{-Q}, the integer values of $a(n)$ obtained in the optimization process become finite word lengths. This follows because both the integer space and the finite word length space are evenly distributed. If the discrete space of $a(n)$ is nonuniformly distributed, such as

$$a(n) = 2^{-g(n)} \tag{5}$$

(where $g(n)$ is an integer), the optimization problem cannot be cast in the form of (4). Any attempt to scale the nonuniformly distributed space of (5) into the integer space is doomed to failure. Thus, general-purpose integer linear programming packages are not suitable for solving the problem unless specially modified. We defer the discussion of these modifications to Section III.

Another possible criterion for optimizing $a(n)$ is to reduce the output error power of Fig. 1 to a minimum. In Fig. 1, $v(n)$ and $e(n)$ are the input signal and error signal, respectively. $V(\omega)$ and $E(\omega)$ are the frequency spectrum of $v(n)$ and $e(n)$, respectively. It can be shown that

$$|E(\omega)|^2 = |V(\omega)|^2 \, |D(\omega) - H(\omega)|^2 . \tag{6}$$

Since

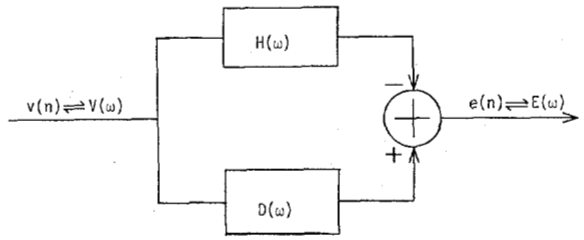


Fig. 1. $D(\omega)$ and $H(\omega)$ are the frequency responses of the ideal filter and actual filter, respectively. Minimizing $\sum_{n=0}^{\infty} e(n)^2$ may be achieved by minimizing $\int_0^{\pi} |v(\omega)|^2 \, |D(\omega) - H(\omega)|^2 \, d\omega$.

$$\sum_{n=0}^{\infty} e(n)^2 \propto \int_0^{\pi} |E(\omega)|^2 \, d\omega, \tag{7}$$

minimizing the output error power thus implies the minimization of

$$J = \int_0^{\pi} W(\omega) \, |D(\omega) - H(\omega)|^2 \, d\omega \tag{8}$$

where $W(\omega) = |V(\omega)|^2$. The factors $|D(\omega) - H(\omega)|^2$ and $W(\omega)$ may be interpreted as the frequency response error square and the frequency response error weighting function, respectively. $W(\omega)$ can be used to weigh the relative importance of the ripples in the passbands and attenuation bands. The integral of (8) may be approximated by a summation as follows:

$$J = \sum_i W(\omega_i) \, |D(\omega_i) - H(\omega_i)|^2 . \tag{9}$$

In (9), the constant of proportionality in approximating the integral by a summation is dropped for simplicity. Rewriting (1) in vector form,

$$H(\omega) = \boldsymbol{C}(\omega)^T \boldsymbol{a} \tag{10}$$

where

$$\boldsymbol{C}(\omega)^T = \left[1 \quad 2\cos\omega \quad 2\cos 2\omega \cdots 2\cos\frac{N-1}{2}\omega\right]$$
$$\boldsymbol{a}^T = \left[a(0) \quad a(1) \cdots a\left(\frac{N-1}{2}\right)\right].$$

Substituting (10) into (9) yields

$$J = \sum_i \{W(\omega_i) \, D(\omega_i)^2 - 2W(\omega_i) \, D(\omega_i) \, \boldsymbol{C}(\omega_i)^T \boldsymbol{a}$$
$$+ W(\omega_i) \boldsymbol{a}^T \boldsymbol{C}(\omega_i) \, \boldsymbol{C}(\omega_i)^T \boldsymbol{a}\} . \tag{11}$$

The minimization of J, subject to linear constraints on the elements of $\boldsymbol{a}$, is a quadratic programming problem. If the discrete coefficient space of $a(n)$ can be converted to the integer space by a linear scaling, then a general-purpose integer quadratic programming package may be used for the solution; otherwise, special treatment as discussed in Section III is necessary.

III. Discrete Programming Using a Simplex-Based General-Purpose Mathematical Programming Package

Linear programming and quadratic programming algorithms are suitable for minimizing or maximizing a linear objective function (linear programming) or a quadratic objective function (quadratic programming) subject to linear constraints. Coupling these mathematical programming packages with a suitable branch-and-bound algorithm [10] enables one to design filters with any discrete coefficient space. In this section, we discuss the basic principles of branch and bound algorithms, and present two useful variants for the filter design.

The first step in a branch-and-bound algorithm is to obtain a continuous coefficient (i.e., infinite precision coefficient value) design by using an appropriate general-purpose mathematical programming package. We designate this problem as P_0. The next step is to select a coefficient whose value is not a desired discrete value. Let this coefficient be $a(n)$. If $\lfloor a(n) \rfloor$ and $\lceil a(n) \rceil$ are two consecutive discrete levels such that

$$\lfloor a(n) \rfloor < a(n) < \lceil a(n) \rceil, \tag{12}$$

then, since the discrete value of $a(n)$ cannot fall between $\lfloor a(n) \rfloor$ and $\lceil a(n) \rceil$, two mathematical programming problems P_1 and P_2 may be generated by adding the constraints

$$a(n) \leqslant \lfloor a(n) \rfloor \tag{13a}$$

$$a(n) \geqslant \lceil a(n) \rceil, \tag{13b}$$

respectively, to the original problem P_0 as shown in Fig. 2. P_1 and P_2 are solved individually. Further branching may be performed on P_1 and P_2 to produce the subproblems P_3, P_4, P_5, and P_6. The branching process of Fig. 2 may be continued until the problem is completely solved. Further details may be found in [9] and [10]. The number of branchings required can be reduced substantially by removing those subproblems for which it can be predicted that an improved solution cannot be obtained. Two useful variants of the foregoing are now presented. A flowchart is shown in Fig. 3.

1) The isocost branch-and-bound search finds the optimum discrete solution by solving a minimum number of subproblems. After solving P_1 and P_2, they are compared and the better one is selected for branching. If P_1 is the better one, then after solving for P_3 and P_4; P_3, P_4, and P_2 are compared and the best one is selected for branching. The procedure continues by always selecting the best subproblem for further branching.

2) The depth-first branch-and-bound search approach finds a series of suboptimum discrete solutions with improving quality until the optimum solution is obtained. The algorithm is as follows. After solving P_0, speculate on P_1 and P_2. If a decision is made to solve P_1, then P_2 is saved for solution later. After solving P_1, speculate on P_3 and P_4; solve one of them and save the other. This procedure continues until a suboptimum discrete solution is obtained—hence the term "depth-first." After obtaining a discrete solution, reinitiate the depth-first search process from an unsolved subproblem. This process continues until the entire problem is solved. Often, some methods are incorporated to forecast the possible result of pursuing a branch. Unpromising branches are fathomed (i.e., terminated) to save computing cost. Compared to the isocost search, the depth-first search solves substantially more subproblems to obtain the optimum solution.

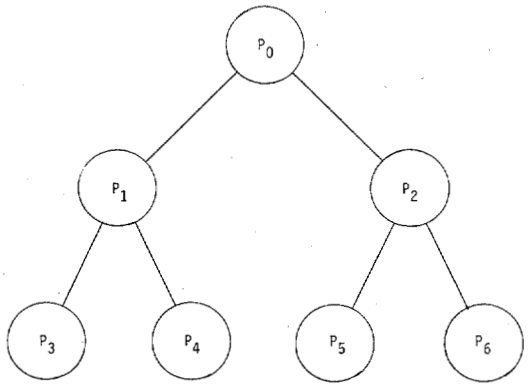


Fig. 2. A branch and bound tree.

At first glance, it appears as though the isocost search is preferred to the depth-first search for designing high-order filters. However, the depth-first search is often preferred for high-order filter design because the computing cost for the optimum design of a high-order filter is excessive. The depth-first search has the ability of producing a first suboptimal discrete solution quickly and then improving on the suboptimal solution with additional investment of computing resources. In practical cases, it is better to have a good suboptimal discrete solution rather than having no solution at all.

The computing cost required for discrete optimization depends on several factors including the following.

1) The efficiency of the linear programming and quadratic programming packages. This is important as they are the optimization tool.

2) The strategy for selecting an $a(n)$ for branching. The strategies used here are the general-purpose strategies reported in standard texts of integer programming.

3) The strategy used to speculate on the subproblems in a depth-first search procedure and the ability to forecast unpromising branches.

4) The quality of the suboptimal design if a suboptimal solution is acceptable.

5) The particular problem at hand. Computing cost required for designing two filters of the same filter length, but with slightly different sets of specifications may sometimes differ by as much as an order of magnitude.

IV. Examples

Fig. 4 presents a comparison of 22 low-pass filters, 11 of which are optimum discrete minimax designs, while the remaining 11 are obtained by rounding the coefficients of the

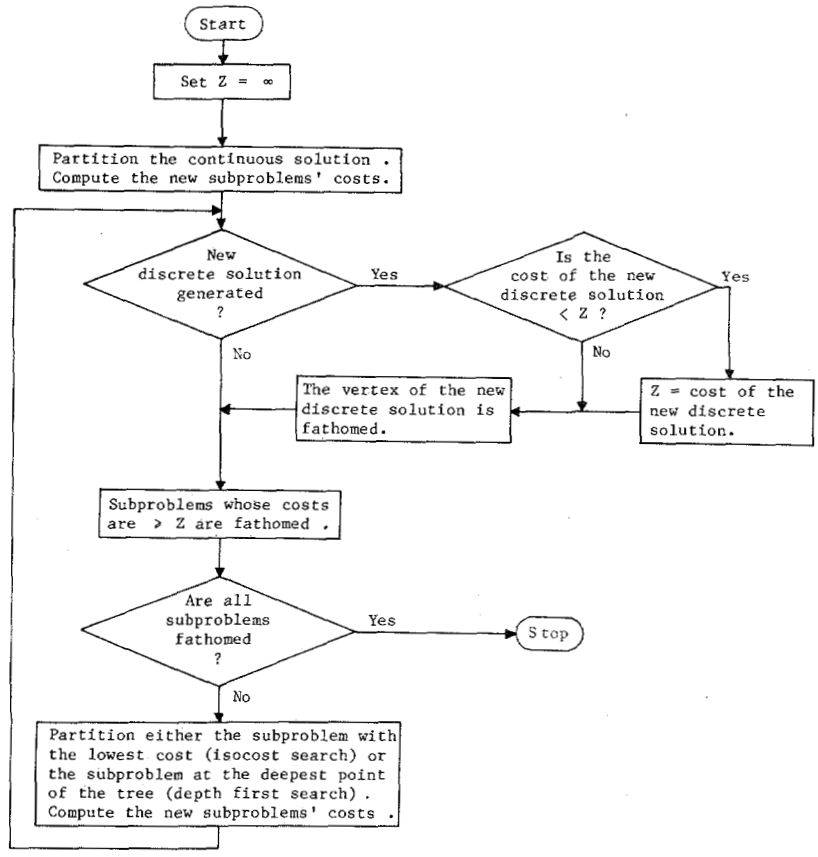


Fig. 3. A tree search flowchart.

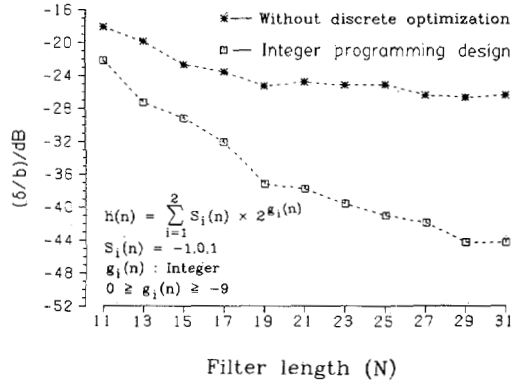


Fig. 4. A comparison between filters designed using integer programming and those obtained by rounding the coefficient values of the corresponding infinite wordlength design. The normalized peak weighted ripple δ/b is used as the performance criterion. Each of the coefficient values is expressed as a sum or difference of two powers-of-two.

corresponding infinite word length design. The passband and stopband have the same ripple weighting and the normalized cutoff frequencies are 0.15 and 0.25, respectively. δ is the peak weighted ripple. The passband gain denoted by b is fixed at the mean value of the passband ripple. The normalized peak weighted ripple δ/b is used as the performance measure criterion. Each of the coefficient values $h(n)$ is expressed as a sum or difference of two powers-of-two, which allows for a simple multiplierless implementation.

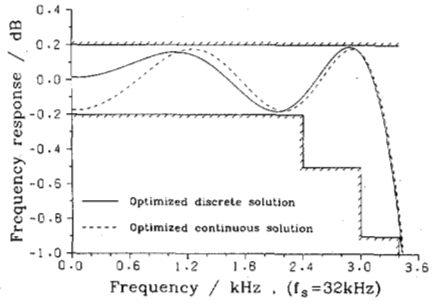
$$h(n) = \sum_{i=1}^{2} S_i(n) \times 2^{g_i(n)}$$

$$S_i(n) = -1, 0, 1 \tag{14}$$

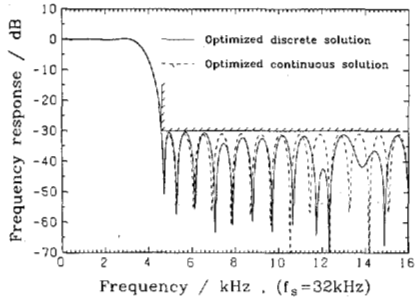
where $g_i(n)$ is an integer with $0 \geq g_i(n) \geq -9$. In Fig. 3, for a filter length $N = 31$, the integer programming design is 17 dB better than that obtained by rounding the infinite word length design.

Discrete space optimization is particularly useful when there is a frequency response specification with given tolerance to be met. For purposes of comparison, an infinite word length FIR linear phase filter, which meets a PCM channel CCITT specification with $N = 35$ and no flexibility for discrete space design, as used in [11], is considered. Fig. 5 shows a design meeting the specification by using integer linear programming with $N = 36$, and each coefficient value expressed as a sum or difference of two-powers-of-two. Thus, for nonmultiplex parallel arithmetic, each multiplier can be replaced by an adder (shifting may be achieved by wiring in a hardware implementation). Fig. 6 shows the frequency response of a filter with $N = 36$ obtained by rounding the coefficient values of the infinite word length design to a sum or difference of two powers-of-two. Obviously, this design fails to meet the specification. This specification cannot be met with this coefficient grid without discrete optimization for an arbitrarily large filter length.

In the above designs, the computer time on a CDC 6400 ranged from a few seconds to a few hundred seconds.



(a)



(b)

Filter length , N = 36

Passband gain = 664.9

Impulse response

h(0) =	2^3-2^0	= h(35)	h(9) =	2^2+2^0	= h(26)
h(1) =	2^0	= h(34)	h(10) =	-2^3-2^1	= h(25)
h(2) =	-2^2-2^1	= h(33)	h(11) =	-2^5+2^2	= h(24)
h(3) =	-2^3-2^0	= h(32)	h(12) =	-2^5+2^0	= h(23)
h(4) =	-2^3	= h(31)	h(13) =	-2^4+2^1	= h(22)
h(5) =	-2^2	= h(30)	h(14) =	2^4+2^3	= h(21)
h(6) =	2^3	= h(29)	h(15) =	2^6+2^4	= h(20)
h(7) =	2^4-2^1	= h(28)	h(16) =	2^7	= h(19)
h(8) =	2^4	= h(27)	h(17) =	2^7+2^5	= h(18)

(c)

Fig. 5. The frequency response of a discrete coefficient filter optimized using integer programming as well as an infinite wordlength filter with length $N = 36$ satisfying a PCM channel CCITT specification. Note the coefficient values shown have been scaled such that they consist of positive powers-of-two; a gain of 664.9 is defined as 0 dB. The coefficient values may, of course, be scaled by any integer powers-of-two.

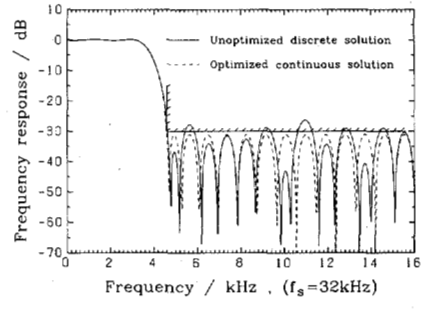


Fig. 6. The frequency response of an unoptimized discrete coefficient filter with length $N = 36$ obtained by rounding the coefficient values of the infinite wordlength design. Each of the coefficient values is expressed as a sum or difference of two powers-of-two. The allowed range of the powers-of-two is the same as in Fig. 5. Obviously, the response does not meet the specification.

In order to meet the specification with a filter whose coefficient values are an exact integer power-of-two, a cascaded design is considered. The optimum design of a cascaded discrete coefficient filter is a nonlinear process requiring excessive computing resources, even for small designs. We now present a suboptimal alternate method which uses moderate computing resources. We shall assume that two discrete coefficient filters F_1 and F_2, with frequency responses $H_1(\omega)$ and $H_2(\omega)$ and lengths N_1, and N_2, respectively, are to be cascaded. The resulting frequency response

$$H(\omega) = H_1(\omega) H_2(\omega) \tag{15}$$

is required to meet a given specification within a given tolerance. There are no specific rules for choosing N_1 and N_2, and so they must be obtained by the trial-and-error method. If $N_{\min}$ is the length of the minimum length infinite word length filter required to meet a given specification within a given tolerance, then a starting value of N_1 and N_2 may be

$$N_1 = N_2 \approx \tfrac{2}{3} N_{\min}.$$

Utilizing the fact that the optimization of $H_2(\omega)$ for a given $H_1(\omega)$, so that $H(\omega)$ is a best approximation to a desired function $D(\omega)$, is a linear optimization process, the following algorithm is presented.

Step 1: Design F_1 so that $H_1(\omega)$ best approximates $D(\omega)$.

Step 2: Design F_2 so that $H(\omega) = H_1(\omega) H_2(\omega)$ best approximates $D(\omega)$.

Step 3: Stop if possible.

Step 4: Redesign F_1 so that $H(\omega)$ best approximates $D(\omega)$.

Step 5: Stop if necessary.

Step 6: Go to Step 2.

The stopping criterion of the above algorithm is flexible, depending on the computing resources of the designer. The above algorithm may be repeated for several sets of N_1 and N_2, and the best result is selected. Fig. 7 shows the frequency response of a cascaded filter whose coefficient values are expressed as a power-of-two meeting the same specification.

Fig. 8 shows a comparison of 12 bandpass designs using a weighted least square response error criterion [see (8)], six of which are optimized (suboptimal results obtained by partial tree search) in the discrete coefficient space, and the remaining six are obtained by rounding the corresponding infinite word length coefficient values. The specification in normalized frequency is as follows.

Band 1: $D(\omega) = 0$, $W(\omega) = 1$, band edges = 0 and 0.05.

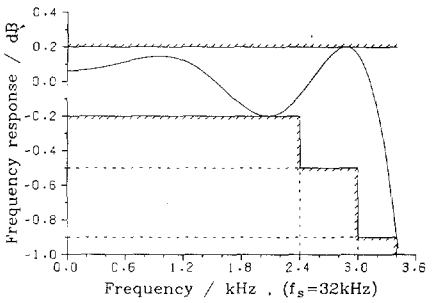
Band 2: $D(\omega) = 1$, $W(\omega) = 10$, band edges = 0.1 and 0.3.

Band 3: $D(\omega) = 0$, $W(\omega) = 1$, band edges = 0.35 and 0.5.

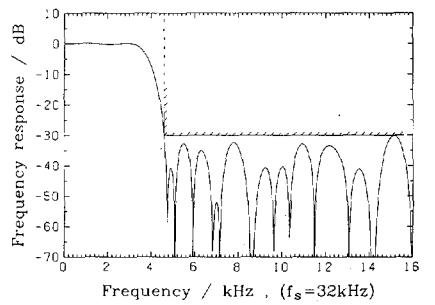
Each coefficient value $h(n)$ is expressed as a sum or difference of two powers-of-two as in (14), with the exception that the range of $g_i(n)$ is such that

$$-1 \geqslant g_i(n) \geqslant -14. \tag{16}$$

It is evident from Fig. 8 that an integer programming design (even when suboptimal) produces significant improvement in performance over simple rounding of coefficient values. The computer time required to design the filter with length $N = 95$ was 10 s on an IBM 3033 computer using a special-purpose program recently developed for high-order discrete coefficient FIR filter designs [12].



(a)



(b)

Filter lengths , N_1 = 24 , N_2 = 22

Passband gain = 38210

Impulse response of F_1		Impulse response of F_2	
h(0) =	2^0 = h(23)	h(0) =	2^2 = h(21)
h(1) =	-2^5 = h(22)	h(1) =	2^0 = h(20)
h(2) =	2^3 = h(21)	h(2) =	-2^0 = h(19)
h(3) =	2^5 = h(20)	h(3) =	-2^2 = h(18)
h(4) =	2^5 = h(19)	h(4) =	-2^2 = h(17)
h(5) =	-2^4 = h(18)	h(5) =	-2^3 = h(16)
h(6) =	-2^5 = h(17)	h(6) =	2^0 = h(15)
h(7) =	-2^5 = h(16)	h(7) =	2^3 = h(14)
h(8) =	2^2 = h(15)	h(8) =	2^4 = h(13)
h(9) =	2^5 = h(14)	h(9) =	2^5 = h(12)
h(10) =	2^6 = h(13)	h(10) =	2^5 = h(11)
h(11) =	2^6 = h(12)		

(c)

Fig. 7. Two filters F_1 and F_2 of length N_1 and N_2, respectively, are cascaded to meet the PCM channel specification. The coefficient values of F_1 and F_2 may be scaled by any powers-of-two. The passband gain is then scaled by the product of the scaling factors for F_1 and F_2.

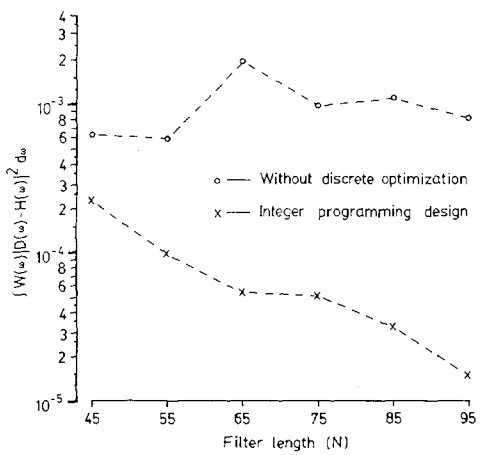


Fig. 8. A comparison between filters designed by integer programming and those obtained by rounding the coefficient values of the infinite wordlength design. The criteria of performance is the mean square error $\int W(\omega)\,(H(\omega) - D(\omega))^2\,d\omega$. Each of the coefficient values is expressed as a sum or difference of two powers of 2. Note rounding the coefficient values does not guarantee a design with a longer filter length to be at least as good as one with a shorter filter length.

V. Discrete Optimizability of a Weighted Least Square Design

In this section, we present a method for predicting how much can be gained by performing discrete optimization over simple coefficient rounding. The basic principle is as follows. After the continuous coefficient design is obtained, fix a coefficient at some other value away from its optimum value. This will cause an increase in the error measure. Reoptimize all the other coefficient values to compensate for the effect of fixing the above coefficient value away from its optimum value. The error measure will decrease by an amount, which is a measure of the discrete optimizability. Unfortunately, this change in the error measure cannot be obtained without performing discrete optimization. However, a measure of the change in the other coefficient values, in order to compensate for fixing some coefficient value away from its optimal value, can be obtained without performing discrete optimization. We shall use this change to serve as a measure on the discrete optimizability. Let

$$\boldsymbol{a} = \boldsymbol{a}_{\text{opt}} + \boldsymbol{\theta} \tag{17a}$$

where $\boldsymbol{a}_{\text{opt}}$ is the optimum set of $\boldsymbol{a}$ which minimizes (11). $\boldsymbol{a}_{\text{opt}}$ satisfies the equation

$$\left[\sum_i W(\omega_i)\,\boldsymbol{C}(\omega_i)\,\boldsymbol{C}^T(\omega_i)\right]\boldsymbol{a}_{\text{opt}} = \sum_i W(\omega_i)\,\boldsymbol{C}(\omega_i)\,D(\omega_i) \tag{17b}$$

and

$$J_{\text{opt}} = \sum_i W(\omega_i)\,D^2(\omega_i) - \boldsymbol{a}_{\text{opt}}^T\left[\sum_i W(\omega_i)\,\boldsymbol{C}(\omega_i)\,\boldsymbol{C}^T(\omega_i)\right]\boldsymbol{a}_{\text{opt}}. \tag{17c}$$

J_{opt} is the value of J when $\boldsymbol{a} = \boldsymbol{a}_{\text{opt}}$. Let

$$\boldsymbol{C} = \sum_i \boldsymbol{C}(\omega_i)\,\boldsymbol{C}(\omega_i)^T\,W(\omega_i). \tag{18}$$

$\boldsymbol{C}$ is a symmetrical positive definite matrix. Substituting (17a)-(17c) and (18) into (11) and noting that $\boldsymbol{a}_{\text{opt}}^T\boldsymbol{C}\boldsymbol{\theta} = \boldsymbol{\theta}^T\boldsymbol{C}\boldsymbol{a}_{\text{opt}}^T$ yields

$$J = J_{\text{opt}} + \boldsymbol{\theta}^T\boldsymbol{C}\boldsymbol{\theta}. \tag{19}$$

The basic principle of obtaining the discrete optimizability of a filter design problem can be illustrated by using a two-coefficient example. Let

$$\boldsymbol{a} = [a(0)\;\; a(1)]^T \tag{20a}$$

$$\boldsymbol{a}_{\text{opt}} = [a_{\text{opt}}(0)\;\; a_{\text{opt}}(1)]^T. \tag{20b}$$

Substituting (20) into (19), the contour for a given value of J is an ellipse as shown in Fig. 9 where $J_2 > J_1$. Consider Fig. 9(a). Moving the value of $a(1)$ from $a_{\text{opt}}(1)$ to $a'(1)$ while keeping $a(0)$ fixed at $a_{\text{opt}}(0)$ increases J from J_{opt} to J_2. If $a(1)$ is fixed at $a'(1)$, an optimization process will move the value of $a(0)$ from $a_{\text{opt}}(0)$ to $a'(0)$ to minimize J. Now consider Fig. 9(b) and (c). Moving the value of $a(1)$ from $a_{\text{opt}}(1)$ to $a'(1)$ increases J to J_2. Fixing the value of $a(1)$ at $a'(1)$, it can be seen that no optimization process is able to select another value of $a(0)$ to reduce the value of J because the optimum value of $a(0)$ for any value of $a(1)$ is $a_{\text{opt}}(0)$. The effect of

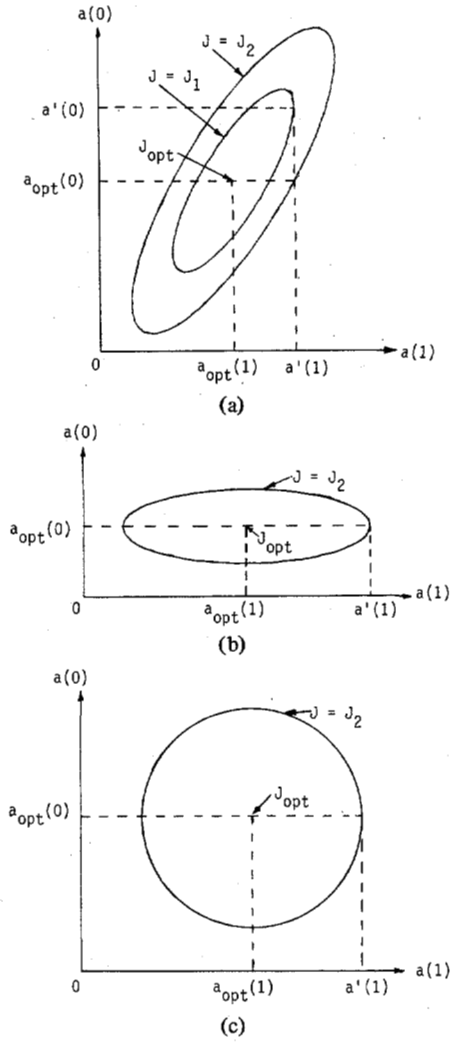


Fig. 9. Elliptical contours of J.

moving the value of $a(1)$ from its optimum value cannot be partially compensated by changing the value of $a(0)$! From the observation of Fig. 9, we draw the conclusion that a filter design problem has low discrete optimizability if

1) the principal axes of the elliptical contours of J have roughly equal length or if

2) the principal axes of the ellipse are almost parallel to the axes of $a(0)$ and $a(1)$.

In general, (19) may be written as

$$J = J_{\text{opt}} + \boldsymbol{\theta}^T \boldsymbol{M}\boldsymbol{S}\boldsymbol{M}^T\boldsymbol{\theta}. \tag{21}$$

$\boldsymbol{S}$ is the spectral matrix (a diagonal matrix whose elements are the eigenvalues of $\boldsymbol{C}$) and $\boldsymbol{M}$ is the normalized modal matrix (a matrix whose columns are the normalized eigenvectors of $\boldsymbol{C}$). Equation (21) shows that the square roots of the eigenvalues of $\boldsymbol{C}$ are the lengths of the principal axes of the multidimensional ellipsoid. The elements of the normalized eigenvectors of $\boldsymbol{C}$ are the direction cosines of the principal axes. Hence, we can draw a general conclusion that the filter design problem has low discrete optimizability if

1) the eigenvalues of $\boldsymbol{C}$ are roughly equal or if

2) a large portion of the elements of $\boldsymbol{M}$ are either almost zero or almost 1.

A special case of the filter design problem is

$$W(\omega) = 1. \tag{22}$$

In this special case, $\boldsymbol{C}$ is a diagonal matrix and the values of all of the elements of $\boldsymbol{M}$ are either 1 or 0. This means that the principal axes of the ellipsoid are parallel to the axes of $\boldsymbol{a}$, and hence, the discrete optimum solution is the rounded coefficient solution.

VI. Discrete Optimizability of a Minimax Design

The basis of the filter design depends upon how much the cost function [8]-[10] and the coefficients will change in the next iteration when one of the coefficient values is incremented. Experience has shown that this information is insufficient for determining (even qualitatively) the discrete optimizability of the filter design problem. However, our experience has shown that if the discrete coefficient space is the powers-of-two space, then the problem has high discrete optimizability.

In the minimax sense, there are no general contours of ellipsoids such as those shown in Fig. 9. The contours are multidimensional polygons specific to each filter design problem. To present an idea of the shape of these polygons, we choose a simplified problem whose specifications are shown below.

Filter type: low-pass; symmetrical impulse response; length = 3.

Band edges: band 1: 0 and 0.1; band 2: 0.3 and 0.5 } normalized frequency.

Peak passband ripple = peak stopband ripple = δ.

Objective: minimize δ.

From the above specifications, the frequency response $H(\omega)$ is given by

$$H(\omega) = a(0) + 2a(1)\cos\omega.$$

A set of inequalities suitable for solution by linear programming may be formulated on a dense grid of frequencies. For simplicity, we shall consider the inequalities on the following four grid points.

$$\omega_0 = 0$$

$$\omega_1 = 0.1 \times 2 \times \pi$$

$$\omega_2 = 0.3 \times 2 \times \pi$$

$$\omega_3 = \pi.$$

The optimum solution is

$$a(0) = 0.3618$$

$$a(1) = 0.2764$$

$$\delta = 0.1910 = \delta_{\text{opt}}.$$

The contours for the given values of δ are shown in Fig. 10. If the value of $a(1)$ is moved to 0.4, without further optimization δ becomes 0.4382. Fixing $a(1) = 0.4$, an optimizing algorithm will move the value of $a(0)$ to 0.5, reducing δ to 0.3.

In the weighted least square sense, the contour of J for $J > J_{\text{opt}}$ is an ellipse whose principal axes bear a fixed ratio and have a constant direction cosine independent of the value of J. In the minimax sense, the contours of δ do not have a fixed geometrical shape. The particular geometry of the contour of δ depends on the value of δ. Fig. 11 shows a few contours of δ for the example of Fig. 10. It can be seen from

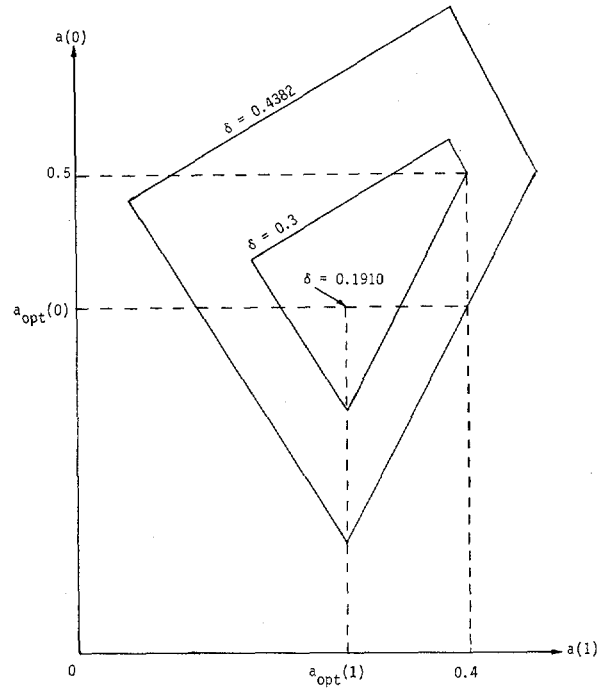


Fig. 10. Polygonal contours of δ.

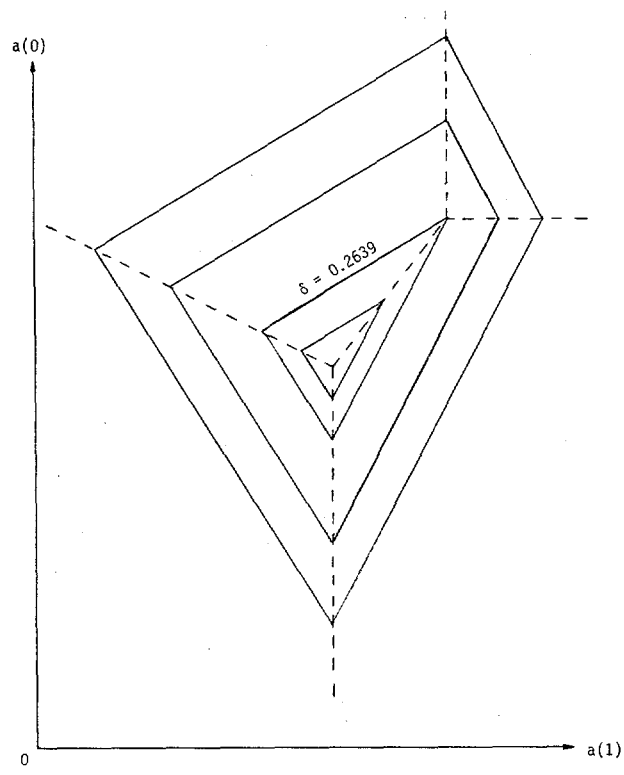


Fig. 11. Contours of δ for the example of Fig. 10. The contour is a triangle for $\delta_{opt} < \delta < 0.2639$. It is a quadrangle for $\delta > 0.2639$.

Fig. 11 that the contour changes from a triangle to a quadrangle when δ crosses the value 0.2639.

VII. Conclusion

The methods of integer linear programming and integer quadratic programming are particularly useful for designing FIR filters with the powers-of-two coefficient grid. The result obtained is significant when compared to simple rounding of coefficient values. The computing cost for the optimum design of a high-order discrete coefficient filter is generally very high. However, good suboptimal results may be obtained with affordable computing cost by coupling a suitable linear programming algorithm or quadratic programming algorithm with a depth-first search algorithm. The construction of a high-efficiency optimization algorithm for designing high-order discrete coefficient filters is an area for future research. The eigenvalues and eigenvectors of C [see (11)] provide useful *a priori* knowledge as to how much can be gained by using discrete optimization techniques over simple rounding of coefficient values. In the minimax case, there is no easy *a priori* approach for determining optimizability.

References

[1] F. Grenez, "Reduction of coefficient wordlength for FIR linear phase digital filters," in *Proc. 1978 European Conf. Circuit Theory and Design.*

[2] Y. Chen, S. M. Kand, and T. G. Marshall, "The optimal design of CCD transversal filter using mixed integer programming technique," in *Proc. 1978 IEEE Int. Symp. Circuits and Syst.*

[3] F. Grenez, "A comparison of the direct and cascade structures for non-recursive digital filters, with regard to the bit-multiplier product," in *Proc. 1978 Int. Conf. Digital Signal Processing.*

[4] D. M. Kodek, "Design of optimal finite wordlength FIR digital filters using integer programming techniques," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 304-308, June 1980.

[5] Y. C. Lim and A. G. Constantinides, "Linear phase FIR digital filter without multipliers," in *Proc. 1979 IEEE Int. Symp. Circuits and Syst.*, pp. 185-188.

[6] —, "New integer programming scheme for non-recursive digital filter design," *Electron. Lett.*, pp. 812-813, Dec. 1979.

[7] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall.

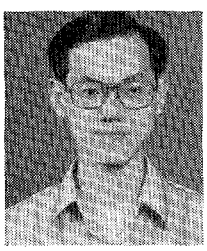
[8] G. Hadley, *Linear Programming.* Reading, MA: Addison-Wesley.

[9] A. Land and S. Powell, *Fortran Codes for Mathematical Programming.* New York: Wiley.

[10] R. S. Garfinkel and G. L. Nemhauser, *Integer Programming.* New York: Wiley.

[11] Y. C. Lim, S. R. Parker, and A. G. Constantinides, "Finite wordlength FIR filter design using integer programming over a discrete coefficient space," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 661-664, Aug. 1982.

[12] Y. C. Lim and S. R. Parker, "A discrete coefficient FIR digital filter design based upon an LMS criteria," in *Proc. 1982 IEEE Int. Symp. Circuits and Syst.*, pp. 796-799.



Yong Ching Lim (S'80-M'80-S'80-M'82) was born in Malaysia. He received the ACGI and B.Sc. degrees in 1977, and the DIC and Ph.D. degrees in 1980, all in electrical engineering from Imperial College, University of London, London, England.

From 1980 to 1982 he was a National Research Council Research Associate at the Naval Postgraduate School, Monterey, CA. He joined the Department of Electrical Engineering, National University of Singapore in 1982. His research interests include Chinese speech processing, system modeling, finite word length digital filter design, adaptive filtering, applications of microprocessors, active *RC* filters, and switch capacitor filters.

Dr. Lim is a member of Eta Kappa Nu.

Sydney R. Parker (S'43-M'45-SM'51-F'75) received the B.E.E. degree from City College of New York, New York, NY, and the M.S. and Sc.D. degrees from the Stevens Institute of Technology, Hoboken, NJ.

From 1956 to 1965 he was a Professor of Electrical Engineering at City College of New York, and in 1965 he became Professor at the University of Houston, Houston, TX. In 1966 he joined the Faculty of the Naval Postgraduate School, Monterey, CA, and in 1970 became Department Chairman, leaving in 1975 to become Dean of Engineering at Rutgers University, New Brunswick, NJ. He returned to the Naval Postgraduate School in 1976. He has served as a consultant for Rockwell International and is currently a consultant for the U.C. Lawrence Livermore Laboratory. He has been an Editor for Pergamon Press, Macmillan, Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS, and Co-Editor of a Joint Special Issue on Adaptive Signal Processing of the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. He is currently Co-Editor of the journal *Circuits, Systems and Signal Processing* (Birkhauser-Boston).

Dr. Parker was President of Eta Kappa Nu (1980-1981) and is a member of Tau Beta Pi and Sigma Xi. He was awarded the NPGS Sigma Xi Research Award in 1977. He was President of the IEEE Circuits and Systems Society in 1974, and served on the Engineering Education and Accreditation Committee of ABET (formerly ECPD) from 1975 to 1980.

Deconvolution of Nonstationary Seismic Data Using Adaptive Lattice Filters

A. K. MAHALANABIS, SENIOR MEMBER, IEEE, SURENDRA PRASAD, MEMBER, IEEE, AND K. P. MOHANDAS

Abstract—**This paper examines the results of the application of two lattice algorithms to the problem of adaptive deconvolution on nonstationary seismic data. A comparative study of the deconvolution performance of the recently proposed gradient lattice and least-squares lattice algorithms is made with the help of experiments on simulated and real seismic data. We show that the gradient lattice algorithm is computationally superior, but it suffers from a possible slow rate of convergence, while the least-squares lattice has better convergence properties and is more robust numerically. We also show that both algorithms can yield equally good deconvolution results with a moderate amount of computation. Finally, we indicate that a modified deconvolved output, derived as a linear combination of the forward and backward residuals, improves the performance without involving any additional computational burden.**

I. INTRODUCTION

DECONVOLUTION of seismic data with a time varying operator has been shown to be very effective in removing multiples and reverberations which render the seismic trace nonstationary. Griffiths, Smolka, and Trembly [1] have applied a modified form of the LMS adaptive filter originally developed by Widrow *et al.* [4], while Prasad and Mahalanabis [2] have studied and compared the performance of the LMS adaptive filter, the adaptive lattice filter, and the adaptive Kalman filter identifier. A comparative study of these methods [2] indicates that the adaptive lattice filter has the same order of computational complexity as the adaptive LMS filter, but that it has a faster convergence rate. The rate of convergence has a direct bearing on the effectiveness of deconvolution for time varying or nonstationary seismic data. As discussed in [2], the reason for the slow convergence of the LMS algorithm lies in the typically large eigenvalue spread of the signal correlation matrix. On the other hand, the convergence of the adaptive gradient lattice filter is independent of the eigenvalue spread. This happens due to the fact that the lattice configuration (Fig. 1) carries out an approximate stage-by-stage orthogonalization of the input data, which permits an independent choice for the adaptation constant for each stage.

The above-mentioned adaptive lattice algorithm, however, still suffers from an arbitrariness in the choice of a suitable value for the displacement constant. This choice is dictated by the contradictory requirements of a fast convergence rate on the one hand and a low maladjustment noise (in the steady state) on the other. We consider here the use of a recursive least-squares algorithm [5] for updating the lattice filter coefficients, which avoids the need for choosing an arbitrary displacement constant. This algorithm is mathematically equivalent to the adaptive Kalman algorithm and requires computations of the order of N. Thus, it combines the desirable computational and numerical advantages of the gradient lattice with the convergence properties of the adaptive Kalman algorithm. The convergence rate, the computational

Manuscript received August 26, 1981; revised September 30, 1982 and December 20, 1982.

A. K. Mahalanabis is with the Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18105.

S. Prasad is with the Department of Electrical Engineering, Indian Institute of Technology, New Delhi, India.

K. P. Mohandas is with the Department of Electrical Engineering, Regional Engineering College, Calicut, Kerala, India.

0096-2518/83/0600-0591$01.00