1982-08

Finite Word Length FIR Filter Design Using Integer Programming Over a Discrete Coefficient Space

Yong, Ching Lim; Parker, Sydney R.; Constantinides, A.G.

IEEE

Lim, Yong, S. Parker, and A. Constantinides. "Finite word length FIR filter design using integer programming over a discrete coefficient space." IEEE Transactions on Acoustics, Speech, and Signal Processing 30.4 (1982): 661-664.
http://hdl.handle.net/10945/61414

precision elsewhere. The resulting MSE was computed based upon 1500 outputs in each case, and is compared to MSE_{ROM} and $\overline{MSE}_{ROM}$ in Table II. The norm $\|G\|_2$ was computed using a formula from [3]. To determine the accuracy of (4), the filters were simulated with $B = 20$ for all word lengths. The resulting MSE is compared to MSE_{TOT} in the last two columns in Table II.

The MSE's in Table II are proportional to 2^{-2B}, so to compare the predicted error with the observed error in units of bits, one should compute $\frac{1}{2}\log_2(\cdot)$ of the values in the table. Performing this computation mentally, we see that the predicted MSE's are generally within a small fraction of one bit of the observed. The greatest departure of predicted error from that observed is only about one bit. In addition, it is seen that MSE_{ROM} and $\overline{MSE}_{ROM}$ are about equally good predictors of the error due to finite word length ROM. Therefore, $\overline{MSE}_{ROM}$, which is easier to compute, will be preferable in practice.

REFERENCES

[1] D. C. Munson, Jr., and B. Liu, "Low-noise realizations for narrow-band recursive digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 41-54, Feb. 1980.

[2] A. Peled and B. Liu, "A new hardware realization of digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 456-462, Dec. 1974.

[3] E. I. Jury, *Theory and Application of the z-Transform Method*. Huntington, NY: Krieger, 1973, p. 298.

Finite Word Length FIR Filter Design Using Integer Programming Over a Discrete Coefficient Space

YONG CHING LIM, SYDNEY R. PARKER, AND A. G. CONSTANTINIDES

Abstract—**It is demonstrated that the improvement achieved by using integer programming over simple coefficient rounding in the design of finite impulse response (FIR) filters with discrete coefficients is most significant when the discrete coefficient space is the powers-of-two space or when a specification is to be met with a given coefficient word length by increasing the filter length. Both minimax and least square error criteria are considered.**

In a recent correspondence, Munson [1] has indicated that the design of finite word length finite impulse response (FIR) digital filters using rounded coefficient values is optimal for both a minimum time domain error norm, $\max|Y_n - \hat{Y}_n|$, and a minimum mean-square error norm, $E(Y_n - \hat{Y}_n)^2$ where Y_n is the output of an infinite precision coefficient version of

Manuscript received June 6, 1981; revised December 2, 1981 and February 17, 1982. This work was supported by the National Research Council, the Naval Postgraduate School, the Office of Naval Research, and the Naval Electronics Systems Command.

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

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

A. G. Constantinides is with the Department of Electrical Engineering, Imperial College, London, England.

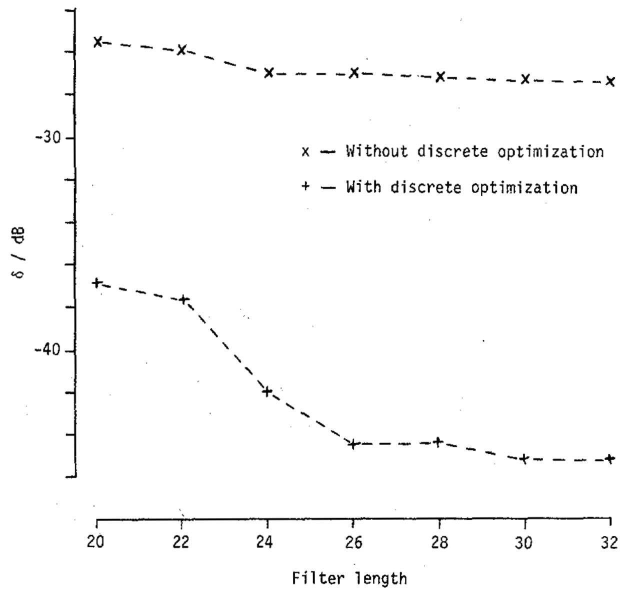


Fig. 1. A comparison between integer linear programming designs and simple coefficient rounding designs, using minimax criterion. The coefficient space is the nonuniformly distributed powers-of-two space.

the FIR filter and $\hat{Y}_n$ is the output of the finite precision coefficient version. Although the optimality criteria discussed in [1] are useful for certain applications, in many cases we are more concerned with optimality in the frequency domain, using a peak weighted error norm given by

$$\max M(\omega)|H(\omega) - \hat{H}(\omega)| \tag{1}$$

where $\hat{H}(\omega)$ is the finite precision coefficient filter frequency response, $M(\omega)$ is a weighting function, and $H(\omega)$ is the desired frequency response. The minimization of (1), subject to discrete constraints in the filter coefficient values, is an integer linear programming problem. It can be shown that the minimization of $E(d_n - Y_n)^2$, where d_n is the desired output, may be achieved by minimizing

$$\int_0^{\pi} W(\omega)|H(\omega) - \hat{H}(\omega)|^2 \, d\omega \tag{2}$$

where $W(\omega)$ is a weighting function equal to the input signal frequency power spectrum. The minimization of (2), subject to discrete constraints in the filter coefficient values, is an integer quadratic programming problem. Integer linear programming and integer quadratic programming are both capable of producing optimum finite word length designs, but are computationally extremely expensive. Hence, integer programmings are to be used only when their effects are significant. From our experience with a large number of designs, we have arrived at the conclusion that integer programming is most desirable for the following two cases.

Case 1: Integer programming is particularly useful when the space of allowable discrete coefficient values is nonuniformly distributed, such as the powers-of-two space. Fig. 1 shows a comparison of 14 low-pass filters, seven of which are designed using integer linear programming, while the remaining seven are obtained by rounding the coefficient values of the corresponding infinite word length designs. The passband and stopband have the same ripple weighting and the normalized

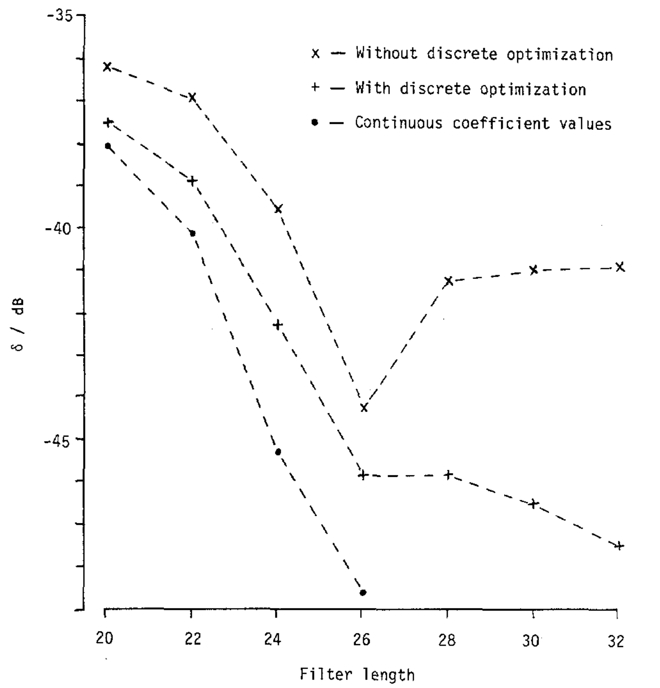


Fig. 2. A comparison between integer linear programming designs and simple coefficient rounding designs, using minimax criterion. The coefficient space is the uniformly distributed finite wordlength space. The coefficient word length is ten bits, including the sign bit.

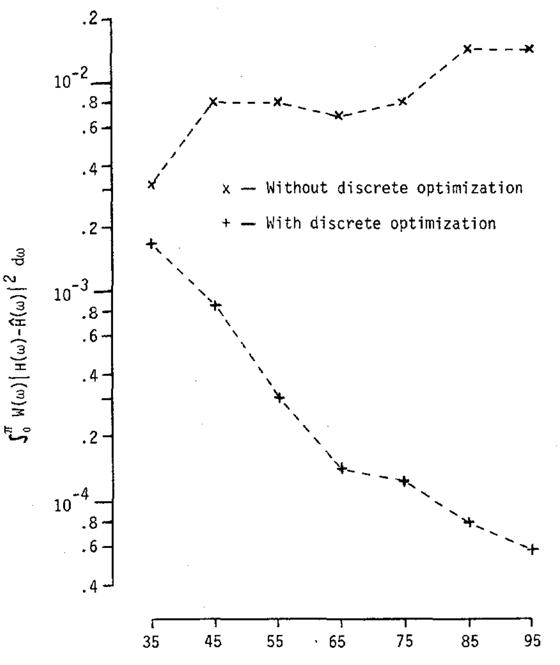


Fig. 3. A comparison between integer quadratic programming designs and simple coefficient rounding designs, using a least square criterion. The coefficient space is the nonuniformly distributed powers-of-two space.

cutoff frequencies are 0.1 and 0.2, respectively. δ is the peak weighted ripple. The passband gain, denoted by b, is fixed at the mean value of the passband ripple subject to the constraint $0.9 \leqslant b \leqslant 1.1$. The normalized peak weighted ripple δ/b is used as the performance measure criterion. Each of

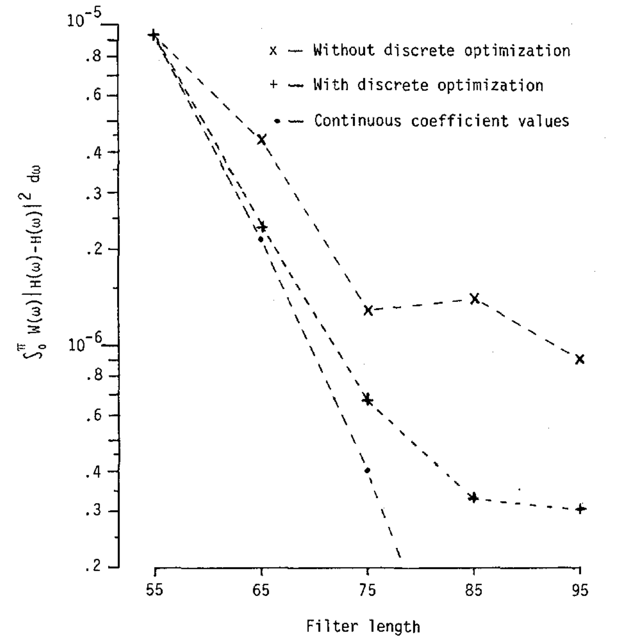


Fig. 4. A comparison between integer quadratic programming designs and simple coefficient rounding designs using least square criterion. The coefficient space is the uniformly distributed finite word length space. The coefficient word length is 14 bits, including the sign bit.

the coefficient values $h(n)$ is expressed as a sum or difference of two powers of two, i.e.,

$$h(n) = \sum_{i=1}^{2} S_i(n) \times 2^{g_i(n)} \tag{3}$$

where $S_i(n) = -1, 0, 1$ and $g_i(n)$ is an integer $\geqslant -10$.

In Fig. 1, for filter length $N = 32$, the integer programming design is 18 dB better than that obtained by rounding the infinite word length design.

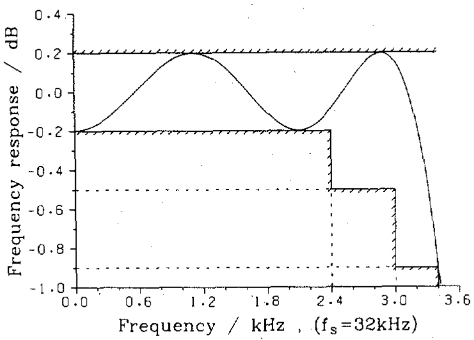
Fig. 2 shows a comparison of another 14 low-pass filters with the same specifications as above. Each coefficient value is represented by ten binary bits, including the sign bit. Comparing the results of Figs. 1 and 2, we note that the improvement achieved by integer linear programming for the uniformly distributed finite word length coefficient space is less impressive when compared to that for the nonuniformly distributed powers-of-two space. The computer time required for each of the above designs, using a simplex-based integer programming program, ranges from several seconds to several hundred seconds on an IBM 3033 computer.

Fig. 3 shows a comparison of 14 bandpass filters, seven of which are designed using integer quadratic programming, while the remaining seven are obtained by rounding the coefficient values of the corresponding infinite word length designs. The specifications in normalized frequency are

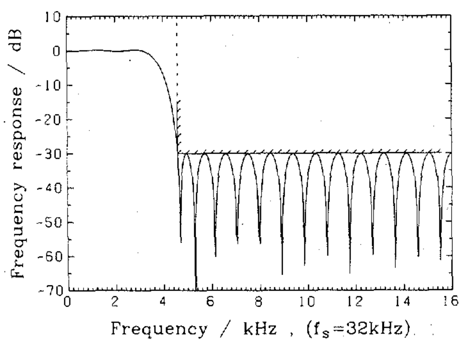
band 1: $H(\omega) = 0$
$W(\omega) = 1$
band edges = 0 and 0.1

band 2: $H(\omega) = 1$
$W(\omega) = 10$
band edges = 0.15 and 0.3

band 3: $H(\omega) = 0$
$W(\omega) = 1$
band edges = 0.35 and 0.5. (4)



(a)



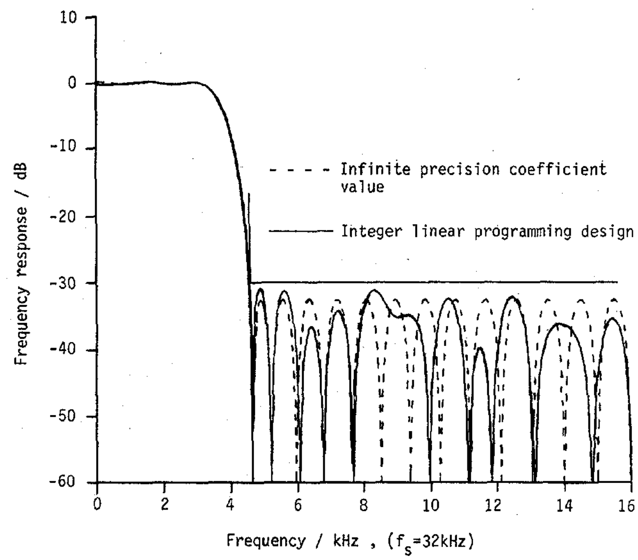
(b)

Fig. 5. A CCITT specification. The minimum length linear phase filter meeting this set of specifications has length $N = 35$.

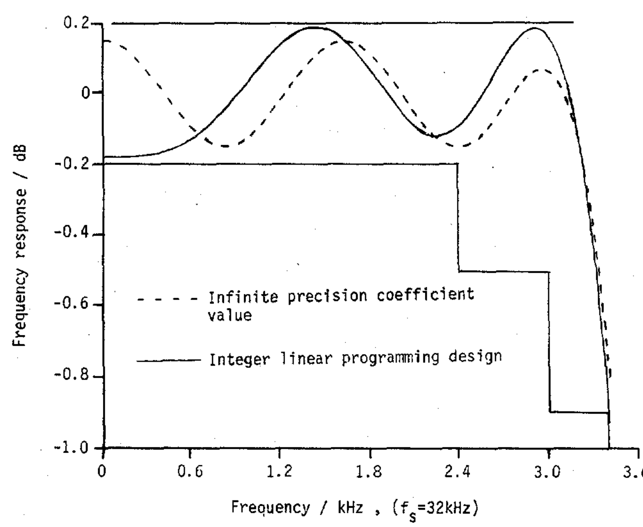
Filter length , N = 38
Passband gain = 1.052 x 2^7
Impulse response

h(0) = 1 = h(37)	h(10) = 1 = h(27)
h(1) = 1 = h(36)	h(11) = -3 = h(26)
h(2) = 0 = h(35)	h(12) = -6 = h(25)
h(3) = -1 = h(34)	h(13) = -6 = h(24)
h(4) = -2 = h(33)	h(14) = -3 = h(23)
h(5) = -2 = h(32)	h(15) = 5 = h(22)
h(6) = 0 = h(31)	h(16) = 16 = h(21)
h(7) = 1 = h(30)	h(17) = 26 = h(20)
h(8) = 3 = h(29)	h(18) = 32 = h(19)
h(9) = 3 = h(28)	

(a)



(b)



(c)

Fig. 6. A filter designed using integer linear programming with 8-bit coefficient value (including sign bit) meeting the specifications of Fig. 5.

The coefficient grid is the same as (3). In Fig. 3, it can be seen the error measure using quadratic programming is always reduced as the filter length is increased. If the discrete coefficient values are obtained by simple coefficient rounding, there is no guarantee that the error measure will not increase with increasing filter length. This fact can also be observed in Fig. 2.

Fig. 4 shows a comparison of ten filters. The specifications are given in (4). Each coefficient value is represented by 14 binary bits, including the sign bit. Comparing the results in Figs. 3 and 4, we note that the improvement achieved by integer quadratic programming for the uniformly distributed finite word length coefficient space is less impressive when compared to that of the nonuniformly distributed powers of two.

The computer time required to design the filter of 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 [2].

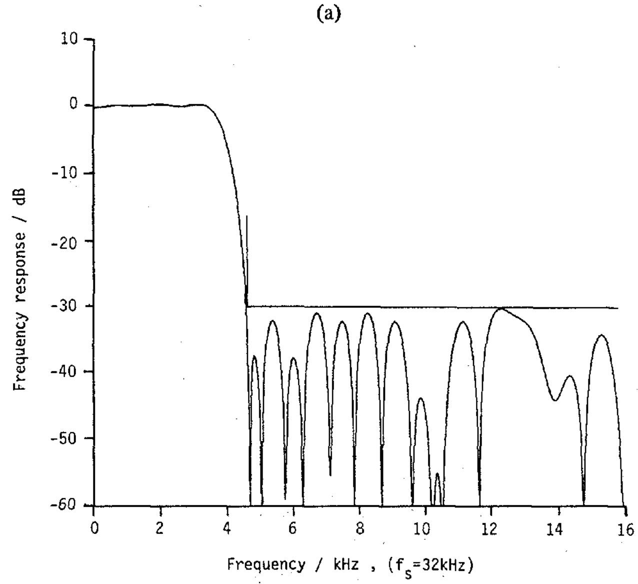
Case 2: Integer programming is very useful when there is a frequency response specification to be met with a given tolerance limit using a fixed coefficient word length. In this type of design problem, the only variable is the filter length. Since simple coefficient rounding does not guarantee that the higher order filter will be at least as good as a lower order one, the chance of meeting a given set of specifications for a given word length by increasing the filter length is sometimes remote. However, integer programming will produce a design (if there is one) meeting the given set of specifications. Fig. 5 shows an infinite word length FIR linear phase filter with a minimum length of $N = 35$, meeting a pulse code modulation (PCM) channel CCITT specification. Imposing an 8-bit constraint on the coefficient word length, the minimum length filter which meets the specifications of Fig. 5 using integer linear programming is $N = 38$. Its frequency response is shown in Fig. 6.

The coefficient values of the finite word length filter ob-

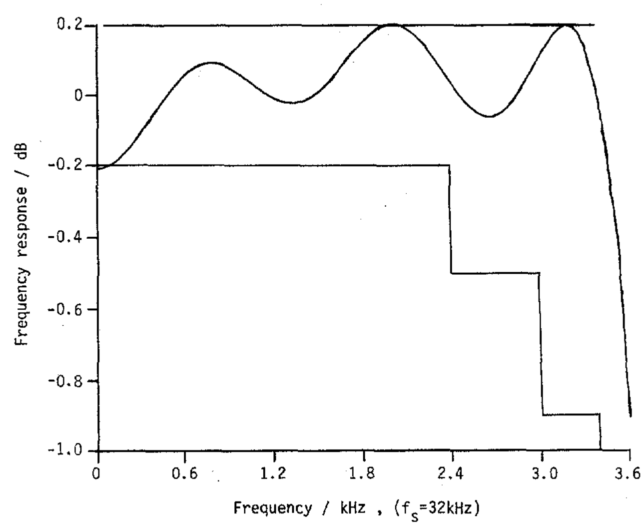
Filter length , N = 45
Passband gain = 0.9916 x 2^7
Impulse response

h(0) = -1 = h(44)	h(12) = 3 = h(32)
h(1) = -1 = h(43)	h(13) = 3 = h(31)
h(2) = 0 = h(42)	h(14) = 0 = h(30)
h(3) = 1 = h(41)	h(15) = -4 = h(29)
h(4) = 1 = h(40)	h(16) = -6 = h(28)
h(5) = 1 = h(39)	h(17) = -6 = h(27)
h(6) = 0 = h(38)	h(18) = 0 = h(26)
h(7) = -1 = h(37)	h(19) = 9 = h(25)
h(8) = -2 = h(36)	h(20) = 20 = h(24)
h(9) = -2 = h(35)	h(21) = 29 = h(23)
h(10) = 0 = h(34)	h(22) = 32 = h(22)
h(11) = 2 = h(33)	

(a)



(b)



(c)

Fig. 7. A filter obtained by simple coefficient rounding.

tained by simple coefficient rounding depends on the infinite word length prototype. There are an infinite number of infinite word length filters capable of meeting the specification of Fig. 5. Our method of choosing the infinite word length prototype is as follows. For any given filter length, an infinite word length prototype is designed using linear programming to meet the given specification. The coefficient values are then rounded to their nearest discrete values and the frequency response is examined. If the resulting finite word length design fails the specification in the stopband, but clears the specification in the passband, then another infinite word length prototype filter with a smaller stopband ripple but a larger passband ripple, keeping the filter length constant, is designed (using linear programming), and vice versa. The band edges of the infinite word length prototype may also be manipulated. Using the above method, over 100 infinite word length prototypes for filter length $\leqslant 50$ were designed, and their coefficient values were rounded to their nearest discrete values. None of the designs meets the specification of Fig. 5. The best result obtained is one with $N = 45$, which fails the specification very slightly in the passband. Its frequency response is shown in Fig. 7. In Fig. 7, it appears as though the passband edge of the infinite precision prototype may be shifted to make room for the passband ripples. However, our attempts to exploit this situation have not produced useful results.

References

[1] D. C. Munson, "On finite wordlength FIR filter design," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, p. 329, Apr. 1981.

[2] 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.*

A Note on Polynomial Transform Error Analysis

S. PRAKASH AND V. V. RAO

Abstract—**In this correspondence, a fixed-point error analysis is given for polynomial transforms computed using two's complement arithmetic. The results are extended for computing the mean-square error of a two-dimensional discrete Fourier transform (DFT) computed by polynomial transform technique. Also, an earlier result derived by Nussbaumer [1] for comparing the rms error/rms result of the polynomial transform and the fast Fourier transform (FFT) has been modified.**

I. Introduction

Polynomial transforms (PT's) are found to be computationally efficient for computing multidimensional discrete Fourier transforms (DFT's), and single and multidimensional convolutions [1]-[5]. Since they are computed without multiplications, a radix-2 PT can be expected to have less roundoff error than the corresponding FFT. A fixed-point error analysis of the polynomial transform has been included in a recent paper [1] by Nussbaumer. Some work has also been reported by these authors on error analysis of convolution using PT technique [6].

In [1] it is stated that for fixed-point implementation (one's complement or sign magnitude), the rms error/rms result of a radix-2 PT is 1.7 times less than the corresponding FFT. But

Manuscript received July 21, 1981; revised October 14, 1981 and March 16, 1982.

The authors are with the Department of Electrical Engineering, Indian Institute of Technology, Madras 600 036, India.

0096-3518/82/0800-0664$00.75