ELSEVIER

Signal Processing 80 (2000) 1591–1596

SIGNAL PROCESSING

www.elsevier.nl/locate/sigpro

Design of finite-word-length FIR filters with least-squares error

Yung-An Kao, Sau-Gee Chen*

Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu, Taiwan

Received 25 May 1998; received in revised form 5 January 2000

Abstract

This paper proposes a new algorithm for designing finite word length linear-phase FIR filters. The new algorithm produces finite-precision least-squares error (LSE) solutions with much reduced search time than the brute-force full search algorithm. It is different from the full search algorithm that tries all possible combinations directly. The new algorithm utilizes geometric properties of a hyper-space to pinpoint potential solutions in a much more restricted way. Accordingly, a much smaller search space is generated. © 2000 Elsevier Science B.V. All rights reserved.

Zusammenfassung

In dieser Arbeit wird ein neuer Entwurfsalgorithmus für linearphasige FIR-Filter bei endlicher Wortlänge vorgeschlagen. Der neue Algorithmus liefert LSE (kleinstes Fehlerquadrat)-Lösungen mit endlicher Genauigkeit bei sehr verkleinerter Suchdauer gegenüber der vollständigen Suche. Er ist verschieden von einer vollständigen Suche, die alle möglichen Kombinationen direkt ausprobiert. Der neue Algorithmus nutzt geometrische Eigenschaften eines Hyperraumes aus, um potentielle Lösungen in einer eingeschränkten Weise festzulegen. Dadurch wird ein viel kleinerer Suchraum erzeugt. © 2000 Elsevier Science B.V. All rights reserved.

Résumé

Nous proposons dans cet article un algorithme nouveau pour la conception de filtres FIR à phase linéaire en précision finie. Cet algorithme produit des solutions aux monidres carrés (LES) avec un temps de recherche bien plus réduit que l'approche de recherche exhaustive. II est différent de l'alogorithme de recherche exhaustive qui essaye directement toutes les combinaisons possibles. Cet algorithme utilise les propriétés géométriques d'un hyperspace pour mettre en évidence less solutions potentielles d'une manière beaucoup plus restrictive. De ce fait un espace de recherche beaucoup plus petit est généré. © 2000 Elsevier Science B.V. All rights reserved.

1. Introduction

In practice, filters are realized by fixed-point arithmetic. In designing finite word length or powers-of-two linear-phase FIR filters there are many algorithms based on integer programming [5] and the modified integer programming algorithms [4,6–8,10,11,13]. Solutions of these algorithms are found by searching the regions confined by some linear constraints subject to minimizing objective functions. The computation load of the linear/integer programming approach [5] is very

* Corresponding author.
E-mail address: sgchen@cc.nctu.edu.tw (Sau-Gee Chen).

PII: S0165-1684(00)00058-X

Nomenclature

ω	frequency in radian
$D(\omega)$	desired frequency response
ω_p	passband cutoff frequency
ω_s	stopband cutoff frequency
$h(n)$	filter coefficient
$\boldsymbol{h}_o$	optimal coefficient vector
$\boldsymbol{D}$	system matrix for solving $\boldsymbol{h}_o$
$\boldsymbol{C}$	vector for solving $\boldsymbol{h}_o$, i.e., $\boldsymbol{h}_o = \boldsymbol{D}^{-1}\boldsymbol{C}$
$\boldsymbol{h}_r$	Rounded vector of $\boldsymbol{h}_o$
$E(\boldsymbol{h})$	square error function due to coefficient vector $\boldsymbol{h}$
$\boldsymbol{h}_{od}$	optimal finite-precision coefficient vector producing the least-squares error
$\boldsymbol{h}_d$	finite-precision version of the coefficient vector $\boldsymbol{h}$
$h_d(n)$	finite-precision version of the coefficient $h(n)$
$h_l(n)$	lower bound of $h_d(n)$
$h_u(n)$	upper bound of $h_d(n)$
$S(n)$	number of all the candidate finite-precision coefficients of $h(n)$ for the full search algorithm
$L(n)$	number of all the candidate finite-precision coefficients of $h(n)$ for the new optimization algorithm

heavy, and it is intended for minimizing the min–max error norm. In implementation, all these algorithms need to sample the target filter spectrum for testing constraints, instead of ideally testing the whole continuous frequency band. This results in computation penalty, as well as error. The algorithms in [4,10] provided fast search algorithms to reduce computation time. Some of the local search algorithms [6,13] reduce search time, at the expense of performance. There are the effective but computation-intensive simulated annealing technique [1,2]. Simulated annealing methods require very intensive computation. The near least-squares error approaches [9,12] reduce computation time considerably, but only get the suboptimal solutions.

In summary, the existing algorithms either produce optimal results at the cost of intensive computation, or produce suboptimal results at a much reduced computation load. In this paper, we will propose a new LSE optimization algorithm for finite word length filters. For each coefficient, the new algorithm utilizes the geometric projection property of a hyper-space to locate potentially discrete solutions, subject to the LSE constraint. From these possible solutions, an efficient tree path search method is introduced to pinpoint the final optimal LSE solution. Doing this way, a much smaller search space than that of brute-force search algorithm is generated, and accordingly a much reduced search time.

2. The new algorithm

The new algorithm starts with the optimal infinite-precision LSE solution [3] to the given ideal response $D(\omega)$, where $D(\omega) = 1$, for $0 \leqslant \omega \leqslant \omega_p$, $D(\omega) = 0$, for $\omega_s \leqslant \omega \leqslant \pi$. Without loss of generality, we consider an N-tap, symmetric, zero-phase, odd-length filter, with the frequency response, $H(e^{j\omega}) = h(0) + 2\sum_{n=1}^{(N-1)/2} h(n)\cos(n\omega)$, $h(n) = h(-n)$. The optimal LSE solution $\boldsymbol{h}_o = [h_o(0)h_o(1)\cdots h_o((N-1)/2)]^T$ can be solved as $\boldsymbol{h}_o = \boldsymbol{D}^{-1}\boldsymbol{C}$ [3], by setting the gradient of the square-error cost function

$$E(\boldsymbol{h}) = \frac{p}{\omega_p}\int_0^{\omega_p} [D(\omega) - H(e^{j\omega})]^2 \, d\omega + \frac{s}{\omega_s}\int_{\omega_s}^{\pi} [H(e^{j\omega})]^2 \, d\omega$$

to zero, where $\boldsymbol{C}$ and $\boldsymbol{D}$ are vector and matrix, respectively, depending upon ω_p, ω_s, p and s.

Define $\boldsymbol{h}_r = \text{round}(\boldsymbol{h}_o 2^b)2^{-b}$, with $b+1$ the number of bits, we can get a square error $k = E(\boldsymbol{h}_r)$. Note that k is very close to the least-squares error $E(\boldsymbol{h}_o)$ and the error surface $E(\boldsymbol{h}) = k$ encloses h_o. As will be shown later in the simulations $\boldsymbol{h}_r$ is a good initial value for locating the optimal discrete $\boldsymbol{h}_{od}$ and occasionally $\boldsymbol{h}_{od} = \boldsymbol{h}_r$. Therefore, we can find some discrete coefficient vectors whose square errors are smaller than k if they are inside the error

surface $E(\boldsymbol{h}) = k$. On the other hand, $\boldsymbol{h}_r$ is the optimal discrete solution when there is no discrete coefficient vector inside the error surface $E(\boldsymbol{h}) = k$. The design problem then reduces to: how do we locate these discrete points which are inside the error surface $E(\boldsymbol{h}) = k$ in an efficient way? To solve the design problem, we will iteratively use a projection algorithm in finding potential discrete coefficients, in combination with an efficient tree-path search algorithm. The projection algorithm utilizes the geometric properties of an LSE surface. Before introducing the new algorithm, we first introduce the projection algorithm.

2.1. The projection algorithm

Given a hyper-ellipse described by $E(\boldsymbol{h}) = k$, if there exits a discrete coefficient vector enclosed by the hyper-ellipse, then the coefficient vector will produce a square error smaller than k. From geometric point of view, to find all the potential finite-precision solutions of a particular coefficient $h(m)$, one can project the hyper-ellipse onto the $h(m)$ axis. This results in a line segment enclosed by $h(m) = h_l(m)$ and $h(m) = h_u(m)$, $h_l(m) < h_u(m)$. All the discrete $h(m)$ points within the line segment potentially lead to a smaller square error than k. Obviously, the projection process is done by locating two surfaces tangent to the hyper-ellipse. Geometrically, the projection is required to be tangent to the hyper-ellipse, and parallel to all $h(n)$ axes, $n = 0, \ldots, (N-1)/2$ and $n \neq m$, but perpendicular to the $h(m)$ axis. Hence, the two tangent points must satisfy the condition that $\partial E(\boldsymbol{h})/\partial h(n) = 0$, $n = 0, \ldots, (N-1)/2, n \neq m$. The condition results in a set of $(N-1)/2$ linear equations. From these equations, the coefficients $h(0), \ldots, h((N-1)/2)$ excluding $h(m)$ can be solved in terms of $h(m)$. That is, they can be solved as $\boldsymbol{h}_{om} = (\boldsymbol{D}_m)^{-1}\boldsymbol{C}_m$, where $\boldsymbol{h}_{om}$ is the coefficient vector excluding $h(m)$, $\boldsymbol{D}_m$ is the system matrix of the set of linear equations, and $\boldsymbol{C}_m$ is a vector whose elements are composed of linear combinations of $h(m)$ and constants. Since the tangent points are on the hyper-ellipse, we can substitute all $h(n)$'s, which are all linear functions of $h(m)$, $n = 0, \ldots, (N-1)/2, n \neq m$, into the quadratic hyper-ellipse function $E(\boldsymbol{h}) = k$. As a result, we have a quadratic equation of $h(m)$ whose roots are $h_l(m)$ and $h_u(m)$, which are the end points of the projected line segment of the hyper-ellipse. In between these two points there are $S(m)$ discrete values of $h(m)$.

2.2. The new finite-precision LSE algorithm based on the projection algorithm

Assume that the finite-precision solutions $h_d(0), \ldots, h_d(m-1)$ for coefficients $h(0), \ldots, h(m-1)$ have been temporarily found and fixed in a manner as described in the following treatment for $h_d(m)$ of $h(m)$ similar to the projection method introduced before, then all the potential finite-precision LSE solutions $h_d(m)$'s for $h(m)$ can be found by setting the gradient of $E(\boldsymbol{h})$ to zero as

$$\frac{\partial E(\boldsymbol{h})}{\partial h(n)} = 0, \quad n = m+1, \ldots, (N-1)/2,$$

which results in a set of $(N-1)/2 - m$ linear equations. From these equations, the coefficients $h(m+1), \ldots, h((N-1)/2)$ can be solved in terms of $h(m)$. By plugging these solutions into equation $E(\boldsymbol{h}) = k$, one can solve two real roots $h_l(m)$ and $h_u(m)$ of $h(m)$, $h_l(m) < h_u(m)$. In between these two points there are $L(m)$ discrete values of $h(m)$. By combining the projection algorithm iteratively with an efficient tree search scheme, one has the following optimization algorithm.

2.2.1. The optimization process of the new algorithm

Step 1. Solve the infinite-precision LSE solution $\boldsymbol{h}_o$.

Step 2. Get $\boldsymbol{h}_r$ by directly rounding $\boldsymbol{h}_o$. Let $m = 0$, $E_{min} = k = E(\boldsymbol{h}_r)$ and let the optimal discrete solution $\boldsymbol{h}_{od} = \boldsymbol{h}_r$.

Step 3. Find all the $L(m)$ potential discrete values $h_d(m)$ of $h(m)$ using the projection algorithm. Reset the index $j(m)$ (of the candidate discrete values) of $h(m)$ to $j(m) = 0$. Note that all the coefficients $h(0), \ldots, h(m-1)$ here have been replaced with some discrete values in the cost function $E(\boldsymbol{h})$.

Step 4. Let $j(m) = j(m) + 1$. Replace $h(m)$ with its $j(m)$th discrete value in $E(\boldsymbol{h}_t)$, where $\boldsymbol{h}_t$ is the coefficient vector consisting of $h_d(0), \ldots, h_d(m)$ obtained in the previous steps, while

discrete values of $h(m+1), \ldots, h((N-1)/2)$ remain to be determined.

Step 5. Cases:

(i) $h(m)$ is not the last coefficient and at least one of the discrete values of $h(m)$ has not been tested (that is, $j(m) \leqslant L(m)$), let $m = m + 1$ and go to Step 3.

(ii) $h(m)$ is the last coefficient and at least one of the discrete values of $h(m)$ has not been tested (that is, $j(m) \leqslant L(m)$), then the bottom level is reached and a complete discrete vector $\boldsymbol{h}_t$ is obtained, do the operations: $\boldsymbol{h}_{od} = \boldsymbol{h}_t$ and $E_{min} = E(\boldsymbol{h}_t)$ if $E(\boldsymbol{h}_t) < E_{min}$, go to Step 4.

(iii) Here, all the discrete values of $h(m)$ have been tested (that is, $j(m) > L(m)$). Let $m = m - 1$, go to Step 4 if $m \geqslant 0$ (regardless of whether $h(m)$ is the last coefficient or not), otherwise go to step 6.

Step 6. All the $\boldsymbol{h}_t$'s have been searched and the LSE solution is obtained, end the optimization process.

According to simulations, most of search paths did not go to the bottom coefficient level, because in most cases the projection algorithm produces null discrete solutions in higher levels. This property greatly reduces the optimization time.

3. Simulations

A low-pass filter design problem is simulated. All the simulations were performed on UltraSPARC using MATLAB 5.1. Here, the filter length N is varied from 19 to 51 (where N is an odd number), $\omega_p = 0.4\pi$, $\omega_s = 0.5\pi$, $p = 0.5, s = 0.5$, and wordlength = 12 bits. The detail simulation data is summarized in Table 1, where the mark '$*$' indicates the cases when $\boldsymbol{h}_{od} = \boldsymbol{h}_r$. In the table, we only list the numbers of all possible solutions for the full search algorithm, because the computation times of full search algorithm greatly increase with N and far exceed those required by non-full search algorithms. The full search algorithm is also based on the projection algorithm defined in subsection 2.1 of Section 2. Specifically, there are $S(0)S(1)\cdots S((N-1)/2-1)S((N-1)/2)$ coefficient vectors to be simulated. The number of combinations increases exponentially with filter length.

Table 1
Speed and LSE comparisons between new algorithm and the Shyu and Lin algorithm [12], and the full search algorithm

Filter length	Square error			Computation time (s)		No. of all possible solutions for the full search algorithm
	New algorithm	Algorithm of [12]	Due to $\boldsymbol{h}_r$	New algorithm	Algorithm of [12]	
*19	9.6157e − 4	9.6157e − 4	9.6157e − 4	2.100e − 1	1.800e − 1	3.2000e + 1
21	4.3243e − 4	4.3243e − 4	4.3244e − 4	3.200e − 1	2.000e − 1	4.6080e + 3
23	4.3077e − 4	4.3077e − 4	4.3086e − 4	1.630e + 0	2.300e − 1	3.1104e + 5
25	2.0633e − 4	2.0634e − 4	2.0637e − 4	1.560e + 0	2.700e − 1	3.4992e + 5
*27	1.8500e − 4	1.8500e − 4	1.8500e − 4	5.600e − 1	3.100e − 1	1.8662e + 5
29	1.0632e − 4	1.0632e − 4	1.0636e − 4	1.860e + 0	3.500e − 1	3.7791e + 7
31	7.7723e − 5	7.7723e − 5	7.7734e − 5	8.900e − 1	4.000e − 1	1.0078e + 7
*33	5.6581e − 5	5.6581e − 5	5.6581e − 5	1.080e + 0	4.900e − 1	1.7916e + 8
35	3.3187e − 5	3.3187e − 5	3.3221e − 5	2.110e + 0	5.700e − 1	3.2249e + 9
37	2.9730e − 5	2.9730e − 5	2.9806e − 5	7.900e + 0	6.400e − 1	5.6435e + 11
39	1.5046e − 5	1.5103e − 5	1.5127e − 5	2.847e + 1	7.200e − 1	2.4079e + 13
*41	1.4977e − 5	1.4977e − 5	1.4977e − 5	5.170e + 0	8.100e − 1	9.2096e + 12
43	7.3998e − 6	7.4425e − 6	7.4665e − 6	2.600e + 1	9.100e − 1	8.2591e + 14
45	7.1164e − 6	7.1331e − 6	7.2659e − 6	2.274e + 2	1.020e + 0	2.8400e + 18
47	3.9814e − 6	3.9957e − 6	4.0026e − 6	2.529e + 1	1.140e + 0	2.5142e + 17
49	3.2726e − 6	3.3374e − 6	3.3374e − 6	3.657e + 1	1.260e + 0	5.8078e + 19
51	2.2629e − 6	2.2955e − 6	2.3479e − 6	2.301e + 2	1.370e + 0	1.0061e + 23

To compare the new optimal algorithm with the existing efficient (however, non-optimal) algorithms, we simulated the fast but suboptimal algorithm by Shyu and Lin [12] (which we consider the most efficient algorithm in the literature), using the same design example. Parameter L in [12] is set to 3. The frequency responses of new algorithm and the algorithm of [12] are shown in Fig. 1 for $N = 39$. As shown, for smaller N, Shyu and Lin's algorithm can obtain the same optimal results as those of the new algorithm in most cases, within shorter time duration than those of the new algorithm. However, for larger N, Shyu and Lin's algorithm fail to locate the optimal solutions. Also, notice that, for off-line and fixed-coefficient applications, filter design time is generally not an issue as long as one can find the optimal solution within an allowable amount of time. This argument puts the new algorithm in a more appealing position than the highly cost-effective (but suboptimal) algorithm of [12].

Table 1 also shows the square errors due to $\boldsymbol{h}_r$. As shown, $\boldsymbol{h}_r$'s are good initial values for locating the optimal $\boldsymbol{h}_{od}$'s, that give square errors close to the LSE's produced by $\boldsymbol{h}_{od}$'s. In some cases $\boldsymbol{h}_r$ is equal to $\boldsymbol{h}_{od}$. In this situation, the new algorithm can solve the optimal solution very quickly. As can be seen, the square errors generally reduce and computation times increase with the increasing filter length. To roughly compare the min-max approach [5], we also simulated the example of [5] with the specifications: $\omega_p = 0.4\pi$, $\omega_s = 0.5\pi$, $N = 21$, and wordlength $= 6$ bits. In this case, $\boldsymbol{h}_{od} = \boldsymbol{h}_r$ and $E(\boldsymbol{h}_{od}) = E(\boldsymbol{h}_r) = 7.7119e-4$ which is predictably smaller than the square error $E(\boldsymbol{h}_{min\text{-}max}) = 12.8863e-4$ due to the min-max solution $\boldsymbol{h}_{min\text{-}max}$ from [5]. On the other hand, the max error due to $\boldsymbol{h}_r$ is 0.1094, which is also predictably larger than the min-max error 0.0711 due to $\boldsymbol{h}_{min\text{-}max}$. For other design examples, similar comparison results can be concluded as this one.

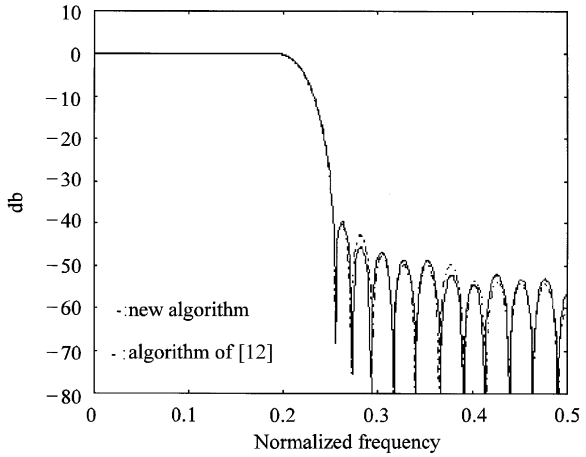

Fig. 1. Frequency response comparison, filter length = 39, word length = 12 bits.

4. Conclusion

An efficient finite-precision filter optimization algorithm generating LSE results is proposed. It is different from the brute-force search algorithm that tries all possible combinations directly. The new algorithm utilizes geometric properties of a hyper space to pinpoint potential solutions in a much more restricted way, and accordingly a much smaller search space is generated. The future work is to extend the algorithm to weighted LSE filters and 2-D filters.

References

[1] N. Benvenuto, M. Marchesi, Digital filters design by simulated annealing, IEEE Trans. Circuits Systems 36 (March 1989) 459–460.

[2] N. Benvenuto, M. Marchesi, A. Uncini, Applications of simulated annealing for the design of special digital filters, IEEE Trans. Signal Process. 40 (February 1992) 323–332.

[3] S. Burrus, A.W. Soewito, R.A. Gopinath, Least squared error FIR filter design with transition bands, IEEE Trans. Signal Process. 40 (6) (June 1992) 1327–1340.

[4] B. Jaumard, M. Minoux, P. Siohan, Finite precision design of filters using a convexity property, IEEE Trans. Acoust. Speech, Signal Process. 36 (3) (March 1988) 407–411.

[5] D.M. Kodek, Design of optimal finite word length FIR digital filters using integer programming technique, IEEE Trans. Acoust. Speech, Signal Process. ASSP-28 (3) (June 1980) 304–308.

[6] D.M. Kodek, K. Steiglitz, Comparison of optimal and local search methods for designing finite word length FIR digital filters, IEEE Trans. Circuits Systems CAS-28 (1) (January 1981) 28–32.

[7] Y.C. Lim, S.R. Parker, Finite word length FIR filter design using integer programming over a discrete coefficient space, IEEE Trans. Acoust. Speech, Signal Process. ASSP-30 (4) (August 1982) 661–664.
[8] Y.C. Lim, S.R. Parker, FIR filter design over a discrete powers-of-two coefficient space, IEEE Trans. Acoust. Speech, Signal Process. ASSP-31 (3) (June 1983) 583–591.
[9] Y.C. Lim, S.R. Parker, Discrete coefficient FIR digital filter design based upon an LMS criteria, IEEE Trans. Circuits Systems CAS-30 (October 1983) 723–739.
[10] J.P. Marques, A new design method of optimal finite word length linear phase FIR filters, IEEE Trans. Acoust. Speech, Signal Process. ASSP-31 (4) (August 1983) 1032–1034.
[11] H. Samueli, An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients, IEEE Trans. Circuits Systems 36 (7) (July 1989) 1044–1047.
[12] J.J. Shyu, Y.C. Lin, A new approach to the design of discrete coefficient FIR digital filters, IEEE Trans. Signal Process. 43 (January 1995) 310–314.
[13] Q. Zho, Y. Tadokoro, A simple design of FIR filters with powers-of-two coefficients, IEEE Trans. Circuits Systems 35 (May 1988) 566–570.