FINITE WORDLENGTH DIGITAL FILTER DESIGN USING AN ANNEALING ALGORITHM

N. Benvenuto, M. Marchesi, G. Orlandi, F. Piazza, A. Uncini

Dipartimento di Elettronica ed Automatica
Università di Ancona
Ancona, Italy

ABSTRACT

Recently, a very versatile algorithm for the design of finite word length filters has been proposed. The algorithm makes use of a simulated annealing algorithm which searches over discrete values of the filter coefficients. It has very fast development time and produces filters with good performances, its only drawback is a quite high execution time. In this paper, a detailed description of the particular annealing algorithm is presented together with three examples of filter design.

1. INTRODUCTION

Usually, specifications for filters response are given in the frequency domain and criterion of the filter design is based on minimizing the maximum weighted ripple (i. e. minmax criterion). Althougth many algorithms are available for the design of discrete coefficients finite impulse response (FIR) digital filters, their memory and execution time requirements are usually very high (see extensive bibliography in reference [1,2,3,4]) Furthermore, all these algorithms have in common an high development time due to their complexity. On the other hand, many local search algorithms exist whose complexity and development time is quite low [4,5]. However, depending on the particular filter design, these techniques may produce filters with poor performance. In this situation we proposed a simulated annealing algorithm which has very fast development time and produces filters with good performance [6]. Its only drawback is a quite high execution time. However, it is our opinion that the real goal on designing a filter is to keep the development time low at a reasonable algorithm execution time. Another advantage of a global optimization technique as simulated annealing is that it is very versatile because it does not make use of the particular objective functional (say minmax) and can be used with multiple constraints.

In [6] the new technique is used to design digital FIR filters for a classical minimax criterion. It is also compared with existing local and global search procedures in terms of value of the final solution and execution time. Here, a detailed description of the annealing algorithm is given and three digital filter design cases are reported.

2. THE OPTIMIZATION ALGORITHM

Simulated Annealing (SA) is a powerful global optimization algorithm introduced for combinatorial optimization [7] and extended also to functions of continuous variables [8]. It is based on random moves, and has the ability to overcome local minima found on the way, toward a better minimum, with up-hill moves.

Since the number of values the variables (filter coefficients) can assume is finite, the design of digital filters with finite word length is a combinatorial problem. However, if the word length is long enough (at least 5-6 bits), the distribution of these values is dense and regular in a finite interval, and many analogies with the continuous case exist.

For this reason, to design digital filters with word length longer than 5 bits, we introduce a SA algorithm directly derived from the SA algorithm for functions of continuous variables [8]. However a proper discretization of the variables values and a better control of the computational effort during the search is needed. The advantage over a traditional combinatorial SA [7] is due to the fact that the new algorithm automatically adapts the search steps in the definition domain of the function, thus saving many function evaluations. It is worth noting that, while our combinatorial optimization algorithm has been derived by discretizing a continuous one, the result of the optimization is related in no way with the rounding of a solution found in a continuous domain.

The following description of the algorithm is based on the continuous algorithm reported in [8], and only the differences will be reported.

If b + 1 is the number of bits used to represent a variable (i.e. filter coefficient), the domain of the variable is given by:

$$D = \{ \pm \frac{2^b-1}{2^b}, \pm \frac{2^b-2}{2^b}, ..., 0\}.$$

Let x be a vector in D^M and x_i its generic i-th component. Let f(x) be the real-valued function to minimize:

$$f(x) : D^M \rightarrow R$$

The algorithm proceeds iteratively: from the starting point x_0 it generates a sequence of points $\{x_k\}$ that tends toward the minimum. The latest point added to the sequence, x_c, is called the current point. New candidate points x are generated at random around the current point along each coordinate direction h (h=1, ..., M), in turn, according to the following formula:

$$\mathbf{x} := \mathbf{x}_c + \mathrm{INT}\,(r\, s_h + 0.5)\, 2^{-b}\, \mathbf{e}_h$$

where r is a random number in the range [-1,1], e_h is the versor of h-th direction, s is the step vector, whose h-component assumes values between 2 and 2^{b-2} (see later) and INT() is the integer part operator. If the newly generated h-th component of x lies outside D, i.e. if:

$$|x_h| > \frac{2^b-1}{2^b}$$

then another trial point x is generated along the h-th coordinate until $x_h \in D$. With this method trial coordinate values are uniformly distributed in intervals of size 2 s_h 2^{-b}, centered around the h-th component of x_c. The generation of an admissible trial point x is called a move. A series of M moves along each coordinate direction is called a cycle.

A trial point x becomes the new current point if the Metropolis test is satisfied [7,8]. Namely, if

CH2673-2/89/0000-0861 $1.00 © 1989 IEEE

$$f(\mathbf{x}) \le f(\mathbf{x}_c)$$

it is always accepted. However, if

$$f(\mathbf{x}) > f(\mathbf{x}_c)$$

x is accepted with probability p where

$$p := \mathrm{EXP}(\frac{f(\mathbf{x}_c) - f(\mathbf{x})}{T})$$

T is a control parameter called temperature. If T > 0, it is possible that a trial point **x** with a higher value of *f* than the current point is accepted. Usually, the method starts with a "high" value of T, for which most trial points are accepted, and then gradually reduces T in order to focus on the minimum. The algorithm is controlled by many parameters. Some of them are the same of the algorithm described in [8] and will not be described in this paper.

Fig. 1 shows a Pascal-like diagram of the main steps of the minimization algorithm. It is composed of an initialization phase, a main loop which it is possible to escape from if a termination test has been satisfied, and four nested loops.

```
begin
  initialize parameters;
  while true do
  begin
    j := 1;
    while j <= $N_r$ do
    begin
      k := 1;
      while k <= $N_k$ do
      begin
        m := 1;
        while m <= $N_s$ do
        begin
          h := 1;
          while h <= M do
          begin
            search move;
            if Metropolis test passed then
            begin
              accept new point;
              compare with $x_{opt}$;
            end;
            h := h+1;
          end;
          m := m+1;
        end;
        step adjustment;
        k := k+1;
      end;
      compute <f>, <$f^2$>, A, $N_k$;
      if termination test passed then goto 1;
      T := $r_T$ * T;
      if T <= $T_{min}$ then goto 1;
      j := j+1;
    end;
    restart from $x_{opt}$;
  end;
1: return $x_{opt}$;
end;
```

Fig. 1 The proposed Simulated Annealing optimization algorithm.

2.1 Initialization and termination

The initialization phase consists of choosing the following parameters:

- $\mathbf{x}_0$: the starting point.
- $\mathbf{s}_0$: the starting step vector.
- T_0 : the starting temperature.
- T_{min} : the minimum temperature.
- r_T : the temperature reduction factor (usually 0.85).
- ε : the terminating criterion.
- N_s : the number of cycles between step adjustments.
- N_T : the minimum number of step adjustments between temperature reductions.

In the new algorithm, besides the classical termination test, the search stops if the temperature falls below a given minimum value T_{min}. A word of caution: when a new minimization problem is tackled it is better to set T_{min} to zero. As some experience is gained on the problem it is easy to understand at which temperature the optimum is "frozen" and the setting of T_{min} allows not to waste computation time at lower temperatures.

2.2 Step adjustments

The step components are periodically adjusted to follow the function behavior and not to waste function evaluations with search steps too big (in such case most of the trial points are rejected) or too small (the sequence tends too slowly toward the minimum). The criterion used to adjust the steps is to maintain a 1:1 ratio between accepted and rejected candidate points along each coordinate direction.

The formulae used for step adjustments are the same reported in [8], with the varying criterion c equal to 3. The only difference is that values of the components of the step vector s must belong to the interval $[2,2^{b-2}]$. If the general formulae yield values lower than 2 or greater than 2^{b-2}, limit values are assumed. In this way, one is guaranteed both that new trial points are likely to differ from the current point and that they are not very likely to lie outside the interval [-1,1].

2.3 Search cycles

In this algorithm, the number of search cycles between step adjustments, N_s, is constant, while the number of step adjustments, N_k, at a fixed temperature, T_k, is not. This parameter, that is crucial for a good sampling of the function to minimize without wasting computational effort, is dynamically varied according to a criterion similar to White's [9].

During the search at a fixed temperature T_{k-1}, the averages of the values of the function to minimize and of its square in the accepted points are computed. Let A be the quantity:

$$A := \frac{<f^2> - <f>^2}{T_{k-1}}$$

The number of steps adjustments at the successive temperature T_k, N_k, is then computed as follows [9]:

$$\begin{cases} N_k := N_T & \text{if } A \le 1 \\ N_k := N_T + 0.2\,(A - 1)\,(N_{max} - N_T) & \text{if } 1 < A < 6 \\ N_k := N_{max} & \text{if } A \ge 6 \end{cases}$$

Incidentally, if the distribution of the values of f in its definition domain is Gaussian with standard deviation *d*, A is an estimate of the ratio d / T_{k-1} [9].

The value of N_0 is set to N_T. The value of N_{max} is usually set to 300, while N_T depends both on the problem and on the dimension M of the search space and usually varies from 20 to 60.

Fig. 2 shows a typical annealing curve (solid line) and the corresponding behavior of A (dashed line) on designing a FIR filter using minmax criterion. In abscissae there is the temperature T in logarithmic scale. The solid line represents the average of *f* for the accepted points at a given temperature. At the lowest values of T the annealing curve exhibits the characteristic plateau that denotes the random sampling of *f*. Values of A are lower than 1, and the number of function evaluations is the lowest possible (this justifies also the random appearance of the curve). When the annealing takes place and the curve decreases, the value of A rises and allows to perform many more function evaluations. Eventually, the annealing curve reaches its final plateau close to the value of the minimum found. Correspondingly, the value of A decreases and allows again not to waste function evaluations when the minimum has been found. In this particular case the algorithm stops because T_{min} was reached.

The variation of N_k allowed to save more than 50% of function evaluations and to improve reliability with respect to the original algorithm.

2.4 Restart from the optimum

In the original algorithm [8], after each temperature reduction, the current point $\mathbf{x}_c$ was set equal to the optimum point reached. This choice was justified by the fact that the functions considered were continuous, with many local minima, and defined over possible unbounded domains. This approach led to better results with respect to the original SA algorithm where the sequence $\{\mathbf{x}_k\}$ was not altered. The presented algorithm deals with a different problem: the minimization of a function of discrete variables defined in quite limited domains, althougth sufficiently dense.

For this reason, we tried many variations of the algorithm, and found that best results were reached when $\mathbf{x}_c$ was set equal to the optimum value every N_r temperature reductions. A typical value of N_r was 20.

3. RESULTS

The presented algorithm was applied to the finite wordlength FIR filters design problem. Some results have already been published and demonstrate to be competitive with more traditional design techniques [6].

Table I shows further results on designing linear phase FIR filter by using a minmax criterion. Here b ranges from 6 to 9 while the number of filter coefficients, N, ranges from 21 to 32. (In case III the stopband ripple is weighted 5 times the passband ripple.) In the first part of Table I we also report the stopband and passband ripples obtained by using other approaches reported in literature. It is seen that at a cost of a relatively long computation time, the SA approach yields comparable or better performance than other "optimum" approaches.

It is worth noting that in the presented SA algorithm when a single coefficient value changes during the search, the cost function evaluation takes into account only the contribution of the changed coefficient, thus saving a lot of computations. Such a modification is not specifical for filter design problems, but can be applied to all the problems in which the cost function is a sum of the contributes of the various parameters. The tests were run on a CDC Cyber 810 computer with NOS 2 operating system.

Recently, the application of a well tuned and adaptive SA algorithm to many design problems of digital filter design has been reported [10,11]. The only FIR filter design case there reported seems comparable with our results, taking into account the fact that the performance ratio between their and our computer is at least 4-5 or more, depending on the precision used.

ACKNOWLEDGMENT

The authors would like to thank Prof. G.M. Cortelazzo who stimulated this work by providing reference [5], and Prof. S. Ridella who generously provided the computer facility. The research has been partially founded by MPI.

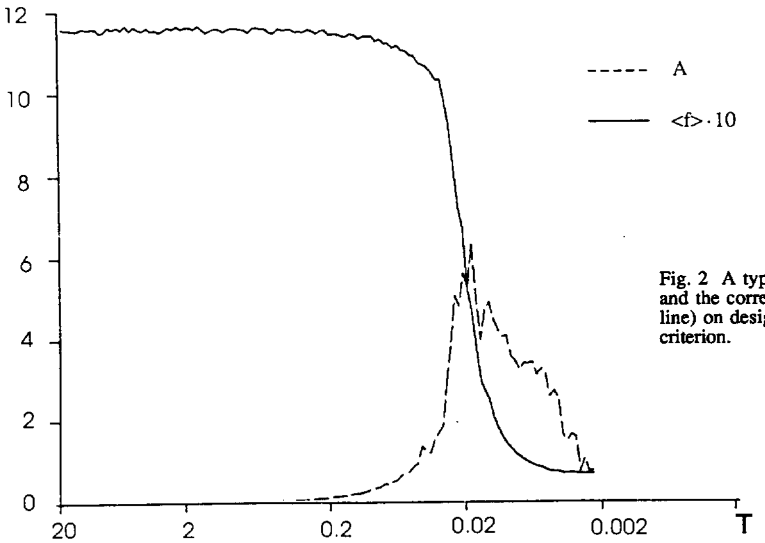

Fig. 2 A typical annealing curve (solid line) and the corresponding behavior of A (dashed line) on designing a FIR filter using minmax criterion.

REFERENCES

[1] Y.C. Lim and S.R. Parker, "Discrete Coefficients FIR Digital Filter Design upon a LMS Criteria", *IEEE Trans. Circuits Syst.*, Vol. CAS-30, pp. 723-739, Oct. 1983.

[2] V.B. Lawrence and A.C. Salazar, "Finite Precision Design of Linear-phase FIR Filters", *Bell. Syst. Tech. J.*, Vol. 59, pp. 1575-1598, Nov 1980..

[3] D.M. Kodek, "Design of Optimal Finite Wordlength FIR Digital Filters Using Integer Programming Techniques", *IEEE Trans. Acoustics Speech Signal Processing*, Vol. ASSP-28, pp.304-308, June 1980.

[4] D. Kodek and K.Steiglitz, "Comparison of Optimal and Local Search Methods for Designing Finite Wordlength FIR Digital Filters", *IEEE Trans. Circuits Syst.*, Vol. CAS-28, pp. 28-32, Jan. 1981.

[5] E.J. Diethorn and D.C. Munson Jr., "Finite Word Length FIR Digital Filter Design Using Simulated Annealing", *Proc. IEEE 1986 Int. Symp. on Circuits and Systems*, San Jose California, May 1986.

[6] N. Benvenuto and M. Marchesi, "Digital Filters by Simulated Annealing", *IEEE Trans. Circuits Syst.*, to appear in the January issue.

[7] S. Kirkpatrick, C.D. Gelatt Jr. and M.P. Vecchi, "Optimization by Simulated Annealing", *Science*, Vol. 220, pp. 671-680, May 1983.

[8] A. Corana, M. Marchesi, C. Martini and S. Ridella, "Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing Algorithm", *ACM Trans. Mathematical Software*, Vol. 13, pp. 262-280, Sept. 1987.

[9] S.R. White, "Concepts of Scale in Simulated Annealing", Proc. IEEE ICCD'84, New York, pp. 646-651, October 1984.

[10] F. Catthoor, J. Vandewalle and H. De Man, "SAMURAI: a General and Efficient Simulated-Annealing Schedule, with Fully Adaptive Annealing Parameters", *Integration, the VLSI J.*, vol. 6, pp. 147-174, 1988 .

[11] F. Catthoor, H. De Man and J. Vandewalle, "Simulated-Annealing-Based Optimization of Coefficient and Data Word-Lengths in Digital Filters", *Int. J. Circuit Theory Appl.*, vol. 16, pp. 371- 390, 1988.

[12] R.K. Patney and S.C. Dutta Roy, "Design of Linear-Phase FIR Filters Using Pseudo-Boolean Methods", *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 255-260, Apr. 1979.

Table 1. Performance of the simulated annealing approach on three filter design cases.

Design Case	I [12]	II [3]	III [12]
N	24	21	32
M	12	11	16
b	9	6	9
bandpass edges	[0,0.08]	[0,0.2]	[0.2,0.354]
stopband edges	[0.16,0.5]	[0.25,0.5]	[0,0.11],[0.415,0.5]
bandpass ripple	0.0167	0.0555	0.044
stopband ripple	0.0143	0.0637	0.0078
T_0	2	2	2
N_s	40	20	40
CPU (s)	8390	3517	12459
bandpass ripple	0.0156	0.0636	0.033
stopband ripple	0.0126	0.0555	0.00651
coefficients * 2^b	(119,97,59,21,-7,-20, -17,-8,0,7,6,3)	(29,20,3,-6,-3,3, 2,-1,-2,1,2)	(147,-151,-57,53,-3,29, 2,-25,6,-7,3,9,-7,3,2,-4)
hstart * 2^b	(256,128,64,32,0,...,0)	(36,16,8,0,...,0)	(0,0,...,0)