Design of Two-dimensional IIR Filters with Self-Organizing Hierarchical PSO Algorithm

Swagatam Das, Amit Konar and Uday K. Chakraborty

***Abstract--* This paper investigates a novel technique of designing 2-dimensional IIR digital filters using a recently-proposed extension of the canonical particle swarm optimization, namely Self-Organizing Hierarchical Particle Swarm Optimizer with Time Varying Acceleration Coefficients (HPSO-TVAC) [7]. The design task is reformulated as a constrained minimization problem and is solved by the HPSO-TVAC algorithm. Numerical results are provided. The paper also attempts to demonstrate the superiority of the present approach by comparing it with recently-published methods based on genetic algorithms and neural networks.**

***Index Terms*—Two-dimensional IIR filter, constrained optimization, particle swarm optimization, genetic algorithm**

I. Introduction

Two-dimensional zero-phase digital filters find an extensive application in the domain of digital image processing, biomedical imaging, digital mammography, X-rays image enhancement, seismic data processing, etc. [1]-[3]. Popular design methods for 2-D IIR filters use either an appropriate transformation of 1-D filter [2], [3] or some optimization technique. One of the major problems underlying the design task is to satisfy the stability criterion for the filter transfer function. There have in the past been several attempts at tackling the stability problem in various ways; however, most of these efforts resulted in a filter with a small stability margin [4]. In this paper the design task of a 2D recursive filter is formulated as a constrained optimization problem and a recent optimization technique [7] borrowed from the field of swarm intelligence is applied to solve it. Our numerical results show that the HPSO-TVAC algorithm yields a better approximation to the transfer function as compared to the works presented in [4] and [5]. This technique also satisfies the stability criterion which is presented as constraints to the minimization problem. Compared to a genetic algorithm (GA) approach, the algorithm used here is seen to require fewer function evaluations to find an acceptable solution.

II. The Problem

Let the general prototype 2-D transfer function for the digital filter be

$$H(z_1, z_2) = H_0 \frac{\sum_{i=0}^{N} \sum_{j=0}^{N} p_{ij} z^i z^j}{\prod_{k=1}^{N} (1 + q_k z_1 + r_k z_2 + s_k z_1 . z_2)} \quad \textbf{(1)}$$

with P_{00} =1. Also let us assume that the user-specified amplitude response of the filter to be designed is M_d, a function of digital frequencies ω_1 and ω_2 ($\omega_1, \omega_2 \in [0, \pi]$). The design problem is to determine the coefficients in the numerator and denominator of (1) in such a fashion that $H(Z_1 = e^{j\omega_1}, Z_2 = e^{j\omega_2})$ follows the desired response $M_d(\omega_1, \omega_2)$ as closely as possible. Such an approximation of the desired response can be achieved by minimizing

$$J(p_{ij}, q_k, r_k, s_k, H_0) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} [|M(\omega_1, \omega_2)| - M_d(\omega_1, \omega_2)]^b \quad \textbf{(2)}$$

where

$$M(\omega_1, \omega_2) = H(z_1, z_2) \Big|_{\substack{z_1 = e^{j\omega_1} \\ z_2 = e^{j\omega_2}}} \quad \textbf{(3)}$$

and

$$\omega_1 = (\pi / N_1) n_1 ;$$
$$\omega_2 = (\pi / N_2) n_2 ;$$

and b is an even positive integer (usually b = 2 or 4).

Here the goal is to reduce the difference between the desired and actual amplitude responses of the filter at $N_1.N_2$ points. Since the denominator contains only first-degree factors, we can assert the stability conditions, following [1]-[3], as

$$|q_k + r_k| - 1 < s_k < 1 - |q_k - r_k|, \quad k = 1, 2... N \quad \textbf{(4)}$$

Thus the design of a 2-D recursive filter is equivalent to the following constrained minimization problem:

Minimize J

$$= \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} [\left| M(\frac{\pi n_1}{N_1}, \frac{\pi n_2}{N_2}) \right| - M_d(\frac{\pi n_1}{N_1}, \frac{\pi n_2}{N_2})]^b \quad \textbf{(5)}$$

subject to the constraints imposed by (4).

In [4] the design problem has been solved with neural networks and in [5] it was solved using a GA. This paper presents a better solution using the HPSO-TVAC algorithm.

Swagatam Das is with the Electronics and Telecommunication Engineering Department, Jadavpur University, Kolkata 700032, India (telephone: +91(33)2528-2717, e-mail: swagatamdas19@yahoo.co.in).

Amit Konar is with the Electronics and Telecommunication Engineering Department, Jadavpur University, Kolkata 700032, India (telephone: +91(33)2416-2697, e-mail: babu25@hotmail.com).

Uday K. Chakraborty is with the Math & Computer Science Department, University of Missouri, St. Louis, MO 63121, USA (telephone: +1(314) 516-6339, e-mail: uday@cs.umsl.edu).

III A BRIEF OVERVIEW OF HPSO-TVAC ALGORITHM

In particle swarm optimization (PSO) [6] a population of particles is initialized with random positions X_i and velocities V_i, and a function, *f*, is evaluated, using the particle's positional coordinates as input values. In an n-dimensional search space, $X_i = (x_{i1}, x_{i2}, x_{i3} \ldots x_{in})$ and $V_i = (v_{i1}, v_{i2}, v_{i3} \ldots v_{in})$. Positions and velocities are adjusted, and an objective function is evaluated with the new coordinates at each time-step. The fundamental velocity and position update equations for the d-th dimension of the i-th particle in the swarm may be given as

$$\left.\begin{aligned} V_{id}(t+1) &= \omega . V_{id}(t) + C_1 . \varphi_1 . (P_{id} - X_{id}(t)) + C_2 . \varphi_2 . (P_{gd} - X_{id}(t)) \\ X_{id}(t+1) &= X_{id}(t) + V_{id}(t+1) \end{aligned}\right\} \quad \textbf{(6)}$$

The variables φ_1 and φ_2 are random positive numbers, drawn from a uniform distribution and restricted to an upper limit φ_{max} which is a parameter of the system. C_1 and C_2 are called acceleration constants whereas ω is called inertia weight. P_i is the best solution found so far by an individual particle while P_g represents the fittest particle found so far in the entire community.

Recently, Ratnaweera and Halgamuge [7] have suggested a novel parameter automation strategy for PSO where the acceleration coefficients are changed linearly with time as

$$C_1 = (C_{1f} - C_{1i})\frac{iter}{MAXITER} + C_{1i} \quad \textbf{(7)}$$
$$C_2 = (C_{2f} - C_{2i})\frac{iter}{MAXITER} + C_{2i}$$

where C_{1i}, C_{1f}, C_{2i}, and C_{2f} are constants, *iter* is the current iteration number and *MAXITER* is the number of maximum allowable iterations. The objective of this modification was to boost search over the entire search space during the early part of the optimization process and to encourage the particles to converge to the global optima at the end of the search. The authors referred to this as the PSO-TVAC method. They proposed another improvement, named *self-organizing hierarchical particle swarm optimizer* (HPSO-TVAC), that uses the 'time varying acceleration coefficients'. In that method the inertial velocity term is kept at zero and the modulus of the velocity vector is reinitialized to a random velocity, known as "re-initialization velocity", whenever the particle gets stagnant ($V_{id} = 0$) in some region of the search space. In this way a series of particle swarm optimizers are generated automatically inside the main particle system according to the behavior of the particles in the search space, until some stopping criterion is met. Following [7], in this paper the re-initialization velocity is kept proportional to the maximum allowable velocity V_{max}. The pseudocode for the HPSO-TVAC algorithm may be presented as follows:

```
begin
    initialize the population;
  while (stopping condition = false)
    do
     for (i = 1 to no_of_particles)
         evaluate the fitness := f(x)
         update Pid and Pgd ;
      for (d = 1 to no_of_dimensions)
        update the new velocity
        Vid = c1.rand1( ).(Pid - xid) + c2.rand2( ). (Pgd – xid);
      if (Vid = 0)
          if (rand3( ) < 0.5)
           Vid = rand4( ).V;
          else
            Vid = -rand5( ).V;
          end if
      end if
      Vid = sign(Vid).min(abs(Vid,Vmax);
          update the position;
          d := d+1;
       i := i+1;
   end do
 end
```

This algorithm has reportedly outperformed the classical PSO and some of its older variants in most cases when tested over a number of benchmark functions [7].

IV. APPLICATION OF THE ALGORITHM TO THE DESIGN PROBLEM

A. CONVERTING THE PROBLEM TO A SOLUBLE FORM

Without loss of generality let us assume N = 2 in equation (1). Now if we substitute Z_1 and Z_2 as in (3), then $M(\omega_1, \omega_2)$ can be expressed in a compact form as

$$M(\omega_1, \omega_2) = H_0 \frac{N_R - jN_I}{(D_{1R} - jD_{1I}).(D_{2R} - jD_{2I})} \quad \textbf{(8)}$$

where

$$\begin{aligned} N_R &= p_{00} + p_{01}f_{01} + p_{02}f_{02} + p_{10}f_{10} + p_{20}f_{20} + p_{11}f_{11} + p_{12}f_{12} + p_{21}f_{21} + p_{22}f_{22} \\ N_I &= p_{00} + p_{01}g_{01} + p_{02}g_{02} + p_{10}g_{10} + p_{20}g_{20} + p_{11}g_{11} + p_{12}g_{12} + p_{21}g_{21} + p_{22}g_{22} \\ D_{1R} &= 1 + q_1 f_{10} + r_1 f_{01} + s_1 f_{11} \\ D_{1I} &= q_1 g_{10} + r_1 g_{01} + s_1 g_{11} \\ D_{2R} &= 1 + q_2 f_{10} + r_2 f_{01} + s_2 f_{11} \\ D_{2I} &= q_2 g_{10} + r_2 g_{01} + s_2 g_{11} \end{aligned} \quad \textbf{(9a)}$$

with $f_{xy} = \cos(x\omega_1 + y\omega_2)$
$g_{xy} = \sin(x\omega_1 + y\omega_2)$ **(9b)**
and x, y = 0, 1, 2.

Hence the actual magnitude may be written as

$$|M(\omega_1, \omega_2)| = H_0 \sqrt{\frac{(N_R^2 + N_I^2)}{(D_{1R}^2 + D_{1I}^2)(D_{2R}^2 + D_{2I}^2)}} \quad \textbf{(10)}$$

Now let us consider a specific example of the design problem where the user specification for the desired circular symmetric low pass filter response may be given as

$$M_d(\omega_1, \omega_2) = \begin{cases} 1, & \text{if } \sqrt{\omega_1^2 + \omega_2^2} \le 0.04\pi \\ 0.5, & \text{if } 0.04\pi \le \sqrt{\omega_1^2 + \omega_2^2} \le 0.08\pi \\ 0, & \text{otherwise.} \end{cases} \quad \textbf{(11)}$$

Also from (4) the constraints may be put in a continuously differentiable form as

$$\begin{aligned} &-(1+s_k) < (q_k + r_k) < (1+s_k) \\ &-(1-s_k) < (q_k - r_k) < (1-s_k) \\ &(1+s_k) > 0 \\ &(1-s_k) > 0 \end{aligned} \qquad \textbf{(12)}$$

In this example problem we select b = 2, N_1 = 50, and N_2 = 50. Thus the constrained minimization task becomes:

Minimize J

$$= \sum_{n_1=0}^{50}\sum_{n_2=0}^{50}\left[\left|M\left(\frac{\pi n_1}{50},\frac{\pi n_2}{50}\right)\right| - M_d\left(\frac{\pi n_1}{50},\frac{\pi n_2}{50}\right)\right]^2$$

(13)

subject to constraints imposed by (12) with k = 1,2.

B. PARTICLE REPRESENTATION

To apply the HPSO-TVAC algorithm to the problem formulated in (13) we need to represent each trial solution as a particle in a multi-dimensional search space. Since p_{00} is always set to 1 in (1), the present problem has 14 dimensions. A particle has 14 positional coordinates represented by the vector

$$X = (p_{01}, p_{02}, p_{10}, p_{11}, p_{12}, p_{20}, p_{21}, p_{22}, q_1, q_2, r_1, r_2, s_1, s_2, H_0)^T$$

(14)

C. OTHER PARAMETER SETTING

Through simulations on well-known numerical benchmarks (the De Jong functions and some others) we find that a very good range for the acceleration coefficient C_1 is from 0.35 to 2.4 and that for C_2 is from 2.4 to 0.35 in equation (6). For the HPSO algorithm we have used a time-varying re-initialization velocity which was set to decay from V_{max} to $0.1V_{max}$ during the search. A constant population size of 40 particles has been maintained throughout the simulation.

D. HANDLING THE CONSTRAINTS

The constraint-handling method [8] is as follows: a) a feasible solution is preferred to an infeasible solution; b) among two feasible solutions, the one having a better objective function value is preferred; c) between two infeasible solutions, the one having smaller constraint violation is preferred. To tackle the constraints in (12) we start with a population of 200 particles with randomly initialized positional coordinates. Out of these, 40 particles are selected, space-coordinates of which obey the constraints imposed by (12). If more than 40 particles are initially found to obey the constraints, selection takes care of these particles. During a run of the program, the globally best particle is sorted not only on the basis of its lowest fitness value in the swarm but also depending on whether or not it obeys the constraints, i.e., if a particle in the course of its motion yields the lowest fitness value found so far, its position is memorized as the globally best position by all other members in the swarm only if it satisfies the constraints.

E. GA PARAMETER SETUP

To examine the relative performance of the HPSO technique we minimize the same fitness function (given in (13)) using a simple genetic algorithm as proposed in [5]. Chromosomes are initialized randomly. Each parameter or gene in a chromosome is encoded as a 32-bit binary string. In every iteration the chromosomes are subjected to the usual crossover and mutation operations, and members failing to satisfy the constraints are eliminated. From the remaining members, candidates for the next generation are selected according to their fitness values.

V. RESULTS OF THE SIMULATION AND COMPARISON

Fig. 1 shows the desired amplitude response of the filter to be designed. In the present work 50 trials of the HPSO-TVAC algorithm were run and the maximum permissible error limit was achieved within 400 iterations on average.

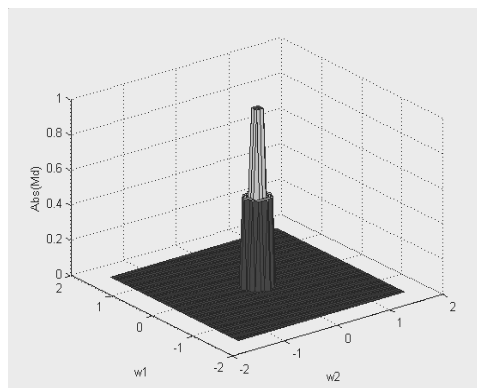

Fig. 1 Desired amplitude response $|M_d(\omega_1, \omega_2)|$ of the 2-D filter

The average value of the best particle positions found in these 50 trials is given by

$$X = [0.1596, 1.2154, 0.4302, 0.5630, -0.3282, 0.2889, -2.0931, 1.7801, -0.9366, -0.9833, -0.8653, -0.8115, 0.8551, 0.8214, 0.0005]^T$$

The corresponding amplitude response is shown in Fig. 2. The amplitude responses obtained by the methods suggested in [4] and [5] are shown in Figs. 3 and 4, respectively. Compared to the results presented in [4] and [5], the HPSO-TVAC algorithm used in this paper yields a better approximation to the desired response and also takes less time to find an acceptable solution. The ripple in the stop-band of Fig. 2 is much less as compared to that in Figs. 3 and 4.

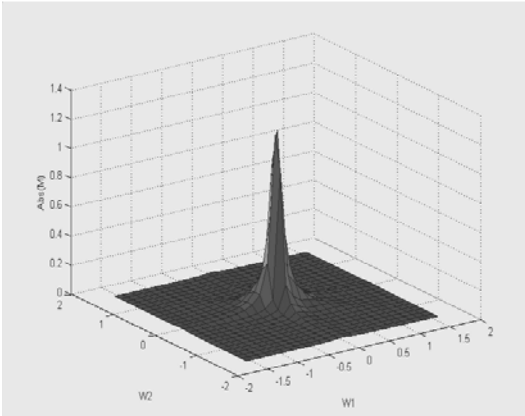

Fig. 2 Amplitude response $|M(\omega_1, \omega_2)|$ of the 2-D filter using HPSO-TVAC

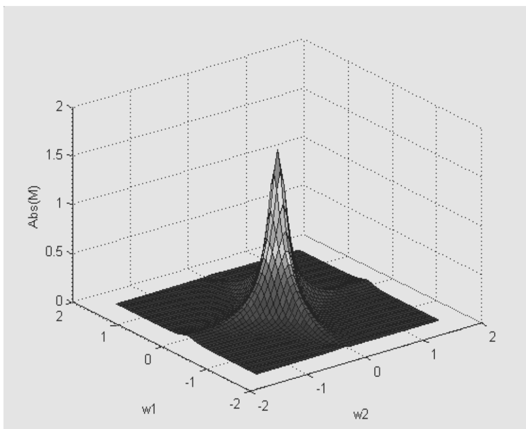

Fig. 3 Amplitude response $|M(\omega_1, \omega_2)|$ of the 2-D filter using GA

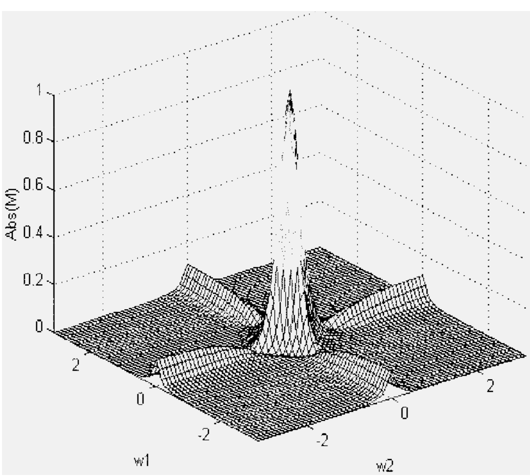

Fig. 4 Amplitude response $|M(\omega_1, \omega_2)|$ of the 2-D filter using the method in [5]

Fig. 5 shows the performance curves of the HPSO-TVAC algorithm and the GA (fitness function value in log scale versus number of function evaluations). This shows HPSO-TVAC to be the better choice when a reasonably good solution is required in a limited time.

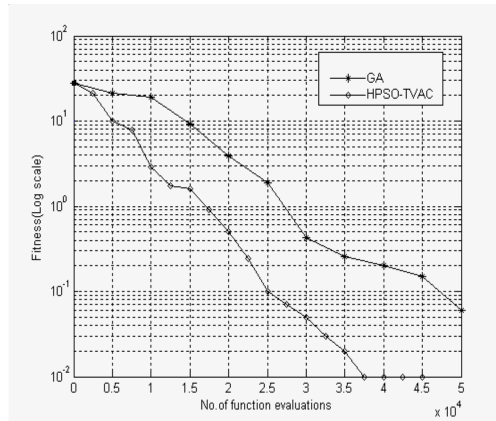

Fig. 5 Performance Comparison for GA and HPSO-TVAC algorithms

VI. CONCLUSION

In this paper a problem of great practical importance has been solved using one of the recent modifications of the classical PSO algorithm. Specifically, the HPSO-TVAC algorithm [7] has been successfully applied to the design of 2-D zero-phase recursive filters. The filter thus obtained has a reasonably good stability margin. We have checked the stability criterion as constraints to the minimization task. The present approach leads to a simpler filter since, in practice, we have to realize a factorable denominator. Compared to the genetic [5] and neural [4] methods published earlier, the algorithm used here yields a better design in a considerably short time.

VII. REFERENCES

[1] Kaczorek, T., Two-Dimensional Linear Systems. Berlin, Germany: Springer-Verlag (1985).

[2] Tzafestas, S. G. (Ed.), Multidimensional Systems, Techniques and Applications. New York: Marcel Dekker (1986).

[3] Lu, W. S., Antoniou, A., Two-Dimensional Digital Filters. NewYork: Marcel Dekker (1992).

[4] Mladenov, V., Mastorakis, N., Design of two-dimensional recursive filters by using neural networks, IEEE Trans. Neural Networks. Vol.12. 585–590 (2001).

[5] Mastorakis, N., Gonos, I. F., Swamy, M. N. S., Design of two-dimensional recursive filters using genetic algorithms, IEEE Trans. Circuits and Systems. Vol.50. 634–639 (2003).

[6] Kennedy, J., Eberhart, R., Particle swarm optimization, Proc. IEEE Int. Conf. Neural Networks, (1995) 1942-1948.

[7] Ratnaweera, A., Halgamuge, K. S., Self organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. on Evolutionary Computation (2004) 8(3): 240-254.

[8] Deb, K., An efficient constraint handling method for genetic algorithms, Computer Methods in Applied Mechanics and Engineering, 186(2-4) (2000) 311-338.