# An Efficient Evolutionary Algorithm Applied to the Design of Two-dimensional IIR Filters


Swagatam Das
Electronics & Telecom Eng Dept.
Jadavpur University
Kolkata 700032, India
+(91) (33) 2528-2717
swagatamdas19@yahoo.co.in

Amit Konar
Electronics & Telecom Eng Dept.
Jadavpur University
Kolkata 700032, India
+(91) (33) 2416-2697
babu25@hotmail.com

Uday K. Chakraborty
Math and Computer Science Dept.
University of Missouri
St. Louis, MO 63121, USA
+1 (314) 516-6339
uday@cs.umsl.edu



## ABSTRACT

This paper presents an efficient technique of designing two-dimensional IIR digital filters using a new algorithm involving the tightly coupled synergism of particle swarm optimization and differential evolution. The design task is reformulated as a constrained minimization problem and is solved by our newly developed PSO-DV (Particle Swarm Optimizer with Differentially perturbed Velocity) algorithm. Numerical results are presented. The paper also demonstrates the superiority of the proposed design method by comparing it with two recently published filter design methods.



## Categories and Subject Descriptors

J.2 --- Electronics, engineering; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search --- *Heuristic methods*; G.1.6 [**Numerical Analysis**]: Optimization --- *Global optimization*; G.3 --- *Probabilistic algorithms*

## General Terms

Design, algorithms

## Keywords

Particle swarm optimization, differential evolution, IIR filter, genetic algorithm


## 1. INTRODUCTION

Two-dimensional zero-phase digital filters find an extensive application in the domain of biomedical imaging and digital mammography, X-rays image enhancement, seismic data processing, etc. [7], [13], [9]. The most popular design methods for 2-D IIR filters are based either on an appropriate transformation of 1D filter [13], [9] or on appropriate optimization techniques. One of the major problems underlying the design task is to satisfy the stability criterion for the filter transfer function. Although researchers have attempted to tackle the stability problem in a number of ways, most of these efforts resulted in a filter having a very small stability margin with hardly any practical importance [11]. The application of evolutionary computation techniques to the design of digital IIR filters can be traced to the work of Gorne and Schneider [6]. Chellapila *et al.* [1] used evolutionary programming in optimizing the coefficients of the transfer function of a one-dimensional IIR filter. In the present paper the design task of 2D recursive filters is formulated as a constrained optimization problem and a new optimization technique, called PSO-DV [2], is applied to solve the problem. In [2], we show that our PSO-DV algorithm outperforms both particle swarm optimization (PSO) [8] and differential evolution (DE) [12] on a set of benchmark functions. In the present paper numerical results show that the algorithm yields a better approximation to the transfer function as compared to the works presented in [11] and [10]. The proposed technique also satisfies the stability criterion which is presented as constraints to the minimization problem. Compared to a genetic algorithm-based method, the algorithm used here is easier to implement and requires fewer function evaluations to find an acceptable solution.

The remainder of this paper is organized as follows. Section 2 provides an overview of the filter design problem and its reformulation as a constrained minimization problem. In Section 3 the PSO-DV algorithm is presented briefly and its application to the present problem is described in Section 4. Section 5 presents the results of applying the proposed method to a specific design problem and also provides performance comparisons with two previous techniques [11], [10]. Conclusions are drawn in Section 6.



*GECCO'05,* June 25-29, 2005, Washington, DC, USA.



## 2. FORMULATION OF THE DESIGN PROBLEM

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

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

with $P_{00} = 1$. Also, let us assume that the user-specified amplitude response of the filter to be designed is $M_d$ which is obviously a function of digital frequencies $\omega_1$ and $\omega_2$ ($\omega_1, \omega_2 \in [0, \pi]$). Now the main design problem is to determine the coefficients in the numerator and denominator of equation (1) in such a fashion that $H(z_1, z_2)$ 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

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

where

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

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).

Equation (2) can be restated as

$$J = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} [\left| M\left(\frac{\pi n_1}{N_1}, \frac{\pi n_2}{N_2}\right)\right| - M_d\left(\frac{\pi n_1}{N_1}, \frac{\pi n_2}{N_2}\right)]^b \quad (4)$$

Here the prime objective 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 [7], [13] and [9], as

$$|q_k + r_k| - 1 < s_k < 1 - |q_k - r_k|, \quad (5)$$

where k = 1, 2,...,N.
Thus the design of a 2-D recursive filter is equivalent to the following constrained minimization problem:

Minimize J

$$= \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} [\left| M\left(\frac{\pi n_1}{N_1}, \frac{\pi n_2}{N_2}\right)\right| - M_d\left(\frac{\pi n_1}{N_1}, \frac{\pi n_2}{N_2}\right)]^b \quad (6a)$$

subject to the constraints

$$|q_k + r_k| - 1 < s_k \qquad k = 1, 2,...,N$$

$$s_k < 1 - |q_k - r_k| \qquad k = 1, 2,...., N \quad (6b)$$

where $N_1$, $N_2$ and N are all positive integers. In [11], the design problem has been tackled with neural networks and the work in [10] attempts to solve it using a GA. In the present paper a much better solution has been obtained using an improved version of particle swarm optimization, called PSO-DV.

## 3. A BRIEF DESCRIPTION OF PSO-DV

In classical PSO [8], 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 $V_i = (v_{i1}, v_{i2}, v_{i3},...,v_{in})$. Positions and velocities are adjusted, and the function 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

$$\left.\begin{aligned} V_{id}(t+1) &= \omega.V_{id}(t) + C_1.\varphi_1.(P_{lid} - X_{id}(t)) + C_2.\varphi_2.(P_{gd} - X_{id}(t)) \\ X_{id}(t+1) &= X_{id}(t) + V_{id}(t+1) \end{aligned}\right\} \quad (7)$$

The variables $\varphi_1$ and $\varphi_2$ are random positive numbers, drawn from a uniform distribution and restricted to an upper limit $\varphi_{max}$ which is a parameter of the system. $C_1$ and $C_2$ are called acceleration constants whereas ω is called inertia weight. $P_{li}$ 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.

In the PSO-DV algorithm [2], for each particle i in the swarm two other distinct particles, say j and k (i ≠ j ≠ k), are selected randomly. The difference between their positional coordinates is taken as a difference vector **δ**:

$$\vec{\delta} = \overrightarrow{X_k} - \overrightarrow{X_j} \quad (8)$$

Then the d-th velocity component (1 < d < n) of the target particle i is updated as follows:

If ($rand_d$ (0, 1) < CR) Then

$$\left.\begin{aligned} &V_{id}(t+1) = \omega.V_{id}(t) + \beta.\delta_d + C_2.\varphi_2.(P_{gd} - X_{id}(t)) \\ &\text{Else } V_{id}(t+1) = V_{id}(t) \end{aligned}\right\} \quad (9)$$

where CR is the crossover probability, $\delta_d$ is the d-th component of the difference vector **δ** defined in equation (8) and β is a scale factor in [0, 1]. In essence the cognitive part of the velocity updating formula in equation (7) is replaced by the vector differential operator to produce some additional exploration capability. Obviously for CR < 1, some of the velocity components retain their old values. Now, a new trial location $Tr_i$ is created by adding this updated velocity to the previous position $X_i$:

$$\overrightarrow{Tr_i} = \overrightarrow{X_i}(t) + \overrightarrow{V_i}(t+1) \quad (10)$$

The particle is placed at this new location only if the coordinates of the location yield a better fitness value. So if we are seeking the minimum of an n-dimensional function $f(x_1, x_2,...,x_n) = f(\vec{X})$, then the target particle is relocated as follows:

$$\left.\begin{aligned} &\text{If } (f(\overrightarrow{Tr_i}) < f(\overrightarrow{X_i}(t))) \text{ then } \overrightarrow{X_i}(t+1) = \overrightarrow{Tr_i} \\ &\text{Else } \overrightarrow{X_i}(t+1) = \overrightarrow{X_i}(t) \end{aligned}\right\} \quad (11)$$

Therefore the particles either move to better positions in the search space or stick to their previous locations every time their velocities are changed. Thus the current location of a particle is actually the best location it has found so far. Unlike the classical PSO, here $P_{lid}$ always equals $X_{id}$. The cognitive part involving $|P_{lid} - X_{id}|$ is eliminated automatically in our algorithm.

If a particle gets stagnant at any point in the search space, the particle is shifted by a random mutation step to a new location. This technique helps escape local minima.

If $((X_i(t) = X_i(t+1) = X_i(t+2) = \ldots = X_i(t+N))$ and

$(f(\overrightarrow{X_i}(t+N)) \neq f^*))$

then for (r = 1 to n)

$$X_{ir}(t+N+1) = X_{min} + rand_r(0, 1)*(X_{max} - X_{min}) \quad (12)$$

where f* is the global minimum of the fitness function, N is the maximum number of iterations up to which stagnation can be tolerated and ($X_{max}$, $X_{min}$) define the legitimate bounds of the search space. The pseudo-code for this method, which we call

PSO-DV (Particle Swarm Optimization with Differentially perturbed Velocity), is presented below:

**Procedure PSO-DV**

**begin**

initialize population;

**while** stopping condition not satisfied do

**for** i = 1 to no_of_particles

evaluate fitness of particle;

update $P_{gd}$ ;

select two other particles j and k (i≠j≠k) randomly;

construct the difference vector as $\vec{\delta} = \overrightarrow{X_k} - \overrightarrow{X_j}$ ;

**for** d = 1 to no_of_dimensions

**if** $rand_d(0, 1) < CR$

$V_{id}(t+1) = \omega.V_{id}(t) + \beta.\delta_d + C_2.\varphi_2.(P_{gd}\text{-}X_{id}(t))$;

**else** $V_{id}(t+1) = V_{id}(t)$; **end if**

**end for**

create trial location as $\overrightarrow{Tr_i} = \overrightarrow{X_i}(t) + \overrightarrow{V_i}(t+1)$ ;

**if** ( $f(\overrightarrow{Tr_i}) < f(\overrightarrow{X_i}(t))$ ) **then** $\overrightarrow{X_i}(t+1) = \overrightarrow{Tr_i}$

**else** $\overrightarrow{X_i}(t+1) = \overrightarrow{X_i}(t)$ ; **end if**

**end for**

**for** i = 1 to no_of_particles

**if** $X_i$ stagnates for N successive generations **for** r = 1 to no_of_dimensions

$X_{ir}(t+1) = X_{min} + rand_r(0, 1)*(X_{max}-X_{min})$

**end for**

**end if**

**end for**

**end while**

**end**

# 4. APPLICATION OF THE ALGORITHM TO THE DESIGN PROBLEM

## 4.1 Converting the Problem to a Suitable Form

Without loss of generality let us assume N = 2. Then H ($Z_1$, $Z_2$) in equation (1) can be restated as:

$$H(z_1,z_2)=H_0\frac{p_{00}+p_{01}z_2+p_{02}z_2^2+p_{10}z_1+p_{20}z_1^2+p_{11}z_1z_2+p_{12}z_1z_2^2+p_{21}z_1^2z_2+p_{22}z_1^2z_2^2}{(1+q_1z_1+r_1z_2+s_1z_1z_2)(1+q_2z_1+r_2z_2+s_2z_1z_2)} \tag{13}$$

Now if we substitute $Z_1$ and $Z_2$ as in (3), then M ($\omega_1$, $\omega_2$ ) can be expressed as:

$$M(\omega_1,\omega_2)=H_0[\frac{(p_{00}+p_{01}f_{01}+p_{02}f_{02}+p_{10}f_{10}+p_{20}f_{20}+p_{11}f_{11}+p_{12}f_{12}+p_{21}f_{21}+p_{22}f_{22})}{V}$$
$$-\frac{j(p_{00}+p_{01}g_{01}+p_{02}g_{02}+p_{10}g_{10}+p_{20}g_{20}+p_{11}g_{11}+p_{12}g_{12}+p_{21}g_{21}+p_{22}g_{22})}{V}] \tag{14a}$$

with $f_{xy} = \cos(x\omega_1 + y\omega_2)$

$g_{xy} = \sin(x\omega_1 + y\omega_2)$

and x, y = 0, 1, 2 (14b)

with

$$V = [(1 + q_1f_{10} + r_1f_{01} + s_1f_{11}) - j(q_1g_{10} + r_1g_{01} + s_1g_{11})] * [(1 + q_2f_{10} + r_2f_{01} + r_2f_{11}) - j(q_2g_{10} + r_2g_{01} + s_2g_{11})] \tag{14c}$$

From (14a) we may put M($\omega_1$, $\omega_2$ ) in a compact form as

$$M(\omega_1,\omega_2) = H_0\frac{N_R - jN_I}{(D_{1R} - jD_{1I}).(D_{2R} - jD_{2I})} \tag{15}$$

where

$$\begin{aligned}
N_R &= p_{00}+p_{01}f_{01}+p_{02}f_{02}+p_{10}f_{10}+p_{20}f_{20}+p_{11}f_{11}+p_{12}f_{12}+p_{21}f_{21}+p_{22}f_{22}\\
N_I &= p_{00}+p_{01}g_{01}+p_{02}g_{02}+p_{10}g_{10}+p_{20}g_{20}+p_{11}g_{11}+p_{12}g_{12}+p_{21}g_{21}+p_{22}g_{22}\\
D_{1R} &= 1+q_1f_{10}+r_1f_{01}+s_1f_{11}\\
D_{1I} &= q_1g_{10}+r_1g_{01}+s_1g_{11}\\
D_{2R} &= 1+q_2f_{10}+r_2f_{01}+s_2f_{11}\\
D_{2I} &= q_2g_{10}+r_2g_{01}+s_2g_{11}
\end{aligned} \tag{16}$$

Hence the actual magnitude may be written as:

$$|M(\omega_1,\omega_2)| = H_0\sqrt{\frac{(N_R^2 + N_I^2)}{(D_{1R}^2 + D_{1I}^2)(D_{2R}^2 + D_{2I}^2)}} \tag{17}$$

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:

$$\begin{aligned}
M_d(\omega_1,\omega_2) &= 1, \text{ if } \sqrt{\omega_1^2 + \omega_2^2} \le 0.04\pi\\
&= 0.5, \text{ if } 0.04\pi \le \sqrt{\omega_1^2 + \omega_2^2} \le 0.08\pi\\
&= 0, \text{ otherwise.}
\end{aligned} \tag{18}$$

Also from (5) the constraints may be put in a continuously differentiable form as:

$$\begin{aligned}
&-(1+s_k) < (q_k + r_k) < (1+s_k)\\
&-(1-s_k) < (q_k - r_k) < (1-s_k)\\
&(1+s_k) > 0\\
&(1-s_k) > 0
\end{aligned} \tag{19}$$

Now in this problem we select b = 2, $N_1$ = 50 and $N_2$ = 50. Finally the constrained minimization task becomes:

Minimize J

$$= \sum_{n_1=0}^{50}\sum_{n_2=0}^{50}[\left|M(\frac{\pi n_1}{50},\frac{\pi n_2}{50})\right| - M_d(\frac{\pi n_1}{50},\frac{\pi n_2}{50})]^2 \tag{20a}$$

subject to the constraints imposed by (19) with k = 1, 2.

Since the objective is to minimize this cost function and the true minimum is at the origin, to make the error function values (usually less than 1) more sensitive, we take the square root of the modulus of each individual term above, instead of taking squares,

while applying the PSO-DV algorithm. Thus for PSO-DV our fitness function becomes:

Minimize J

$$= \sum_{n_1=0}^{50}\sum_{n_2=0}^{50}\sqrt{|| M(\frac{\pi n_1}{50},\frac{\pi n_2}{50}) | - M_d(\frac{\pi n_1}{50},\frac{\pi n_2}{50}) |} \qquad (20b)$$

However, other competitive processes use the cost function (20a).

## 4.2 Particle Representation

In order to apply the PSO-DV algorithm to the problem formulated in (20) 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 dimensionality of the present problem is 14 and each particle has 14 positional coordinates represented by the vector

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

All these 14-dimensional particles have 14 components in their velocity vector.

## 4.3 Population Initialization and Population Size

Each space-coordinate of a particle was initialized with a random floating point number whose absolute value was kept below 3.00. Following Eberhart and Shi [4], the maximum allowable velocity $V_{max}$ for each particle was limited to the upper value of the dynamic range of search, i.e., $|V_{max}| = |X_{imax}| = 3.00$. Eberhart and Shi also showed that the population size has hardly any effect on the performance of the PSO method. It is quite common in PSO research to limit the number of particles in the range 20 to 60. Van den Bergh and Engelbrecht [14] have shown that though there is a slight improvement of the optimal value with increasing swarm sizes, a larger swarm increases the number of function evaluations necessary to converge to an error limit. We maintain a constant population of 40 particles throughout the runs of the PSO-DV algorithm.

## 4.4 Other Control Parameter Setting

Through simulations carried out with numerical benchmarks we find that an excellent value for the acceleration-coefficient $C_2$ is 2.4 in equation (9). For the PSO-DV algorithm, scale factor $\beta$ is kept at a value of 0.78 and the cross-over constant CR has been set to 0.9. For comparison against the work done in [10] we run a simple GA [5] to solve the same problem. The GA parameter set-up is given in Table 1.

**Table 1. GA Parameter Set-up**

| Name of the Parameter | Value |
|---|---|
| Maximum number of chromosomes per generation | 250 |
| No. of bits per gene | 32 |
| Mutation probability | 0.05 |
| Part of genetic materials interchanged during cross-over | 12 |
| Maximum number of children from each pair of parents | 10 |
| Maximum no. of iterations | 500 |

## 4.5 Handling the Constraints

To handle the constraints we followed the method in [3] as follows: a) any feasible solution is preferred to any infeasible solution; b) between two feasible solutions, the one with a better objective function value is preferred; c) between two infeasible solutions, the one having a smaller constraint violation is preferred. To tackle the constraints presented in (19) we start with a population of around 200 particles with randomly initialized positional coordinates. Out of these, 40 particles were selected, space-coordinates of which obey the constraints imposed by (19). If more than 40 particles are initially found to obey the constraints, obviously the selection takes into account the initial fitness value of these particles. During the run of the program, the globally best particle was sorted not only on the basis of its fitness value in the swarm but also depending on whether or not it obeyed the constraints. That is, if a particle in course of its movement through the search space yields the lowest fitness value found so far, its position will be memorized as the globally best position by all other members in the swarm only if it satisfies the constraints.

# 5. RESULTS OF SIMULATION

Fig. 1 shows the desired amplitude response of the filter to be designed. In the present work 50 independent runs of the PSO-DV algorithm were taken and the maximum permissible error limit was achieved within 400 iterations on average.

We took the average value of the best particle positions found in these 50 runs. The vector X found in this way is given by

$$X = [0.1796, 1.3554, 0.4902, 0.5430, -0.3281, 0.2897, -2.0832, 1.7903, -0.9586, -0.9813, -0.8623, -0.8215, 0.8521, 0.8212, 0.0004]^T$$

Therefore the filter transfer function may be given by equation (21) (equation 21 appears at the top of the next page).

The corresponding amplitude response is presented in Fig. 2. For the purpose of comparison, in Figures 3 and 4 we present the amplitude response obtained by using the methods in [11] and [10], respectively. A closer look at these figures reveals that the PSO-DV algorithm yields a better approximation to the desired response compared to the works presented in [11] or [10]. In addition, our method takes considerably less time to find the solution. The ripple in the stop-band of Fig. 2 is much less than that in Fig. 3 or Fig. 4.

Fig. 5 presents the performance curves of PSO-DV and GA applied to the same design problem. This figure presents fitness function values in log scale versus the number of error function evaluations. The curves of Figure 5 show that PSO-DV is the preferred choice when a reasonably good solution is required in a limited time.

$$H(z_1, z_2) = 0.0004 \frac{1 + 0.1796 z_2 + 1.3554 z_2^2 - 4902 z_1 + 0.543 z_1^2 - 0.3281 z_1 z_2 + 0.2897 z_1 z_2^2 - 2.0832 z_1^2 z_2 - 1.7903 z_1^2 z_2^2}{(1 - 0.9586 z_1 - 0.8623 z_2 + 0.8521 z_1 z_2)(1 - 0.9813 z_1 - 0.8215 z_2 + 0.8212 z_1 z_2)} \quad (21)$$

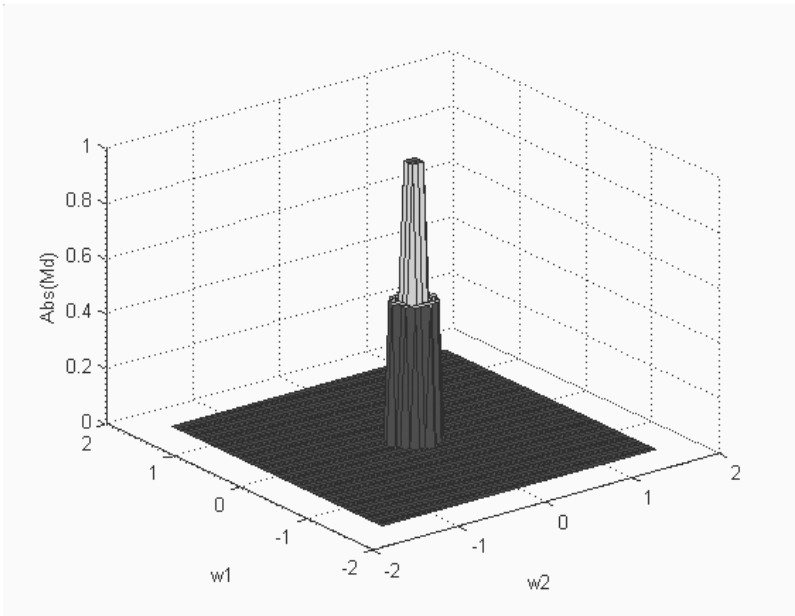


**Figure 1. Desired amplitude response $|M_d(\omega_1, \omega_2)|$ of the 2-D filter.**

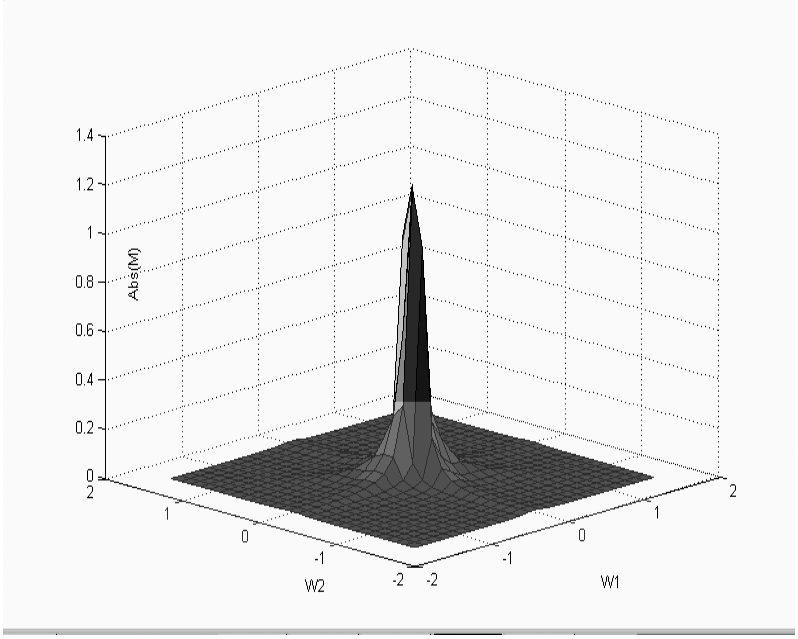


**Figure 2. Amplitude response $|M(\omega_1, \omega_2)|$ of the 2-D filter using PSO-DV.**

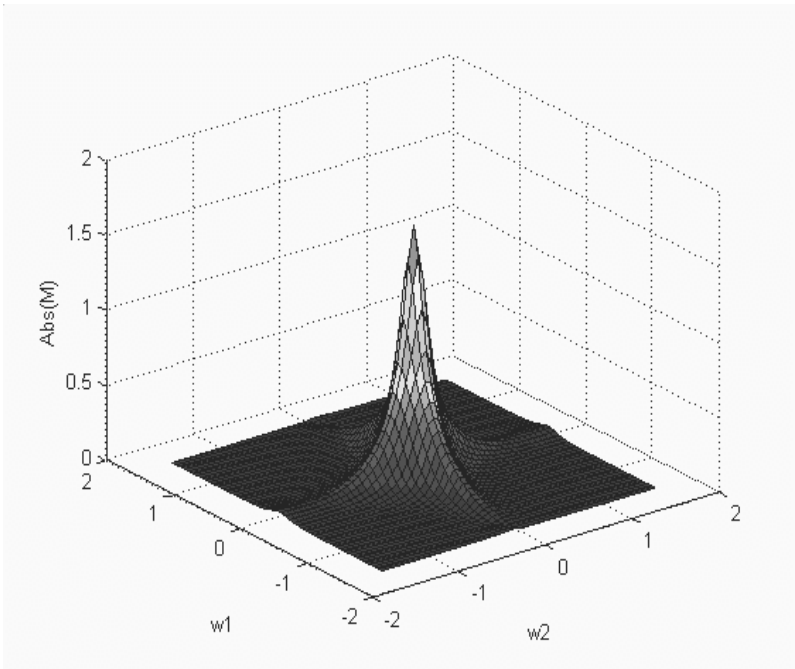


**Figure 3. Amplitude response $|M(\omega_1, \omega_2)|$ of the 2-D filter using GA**

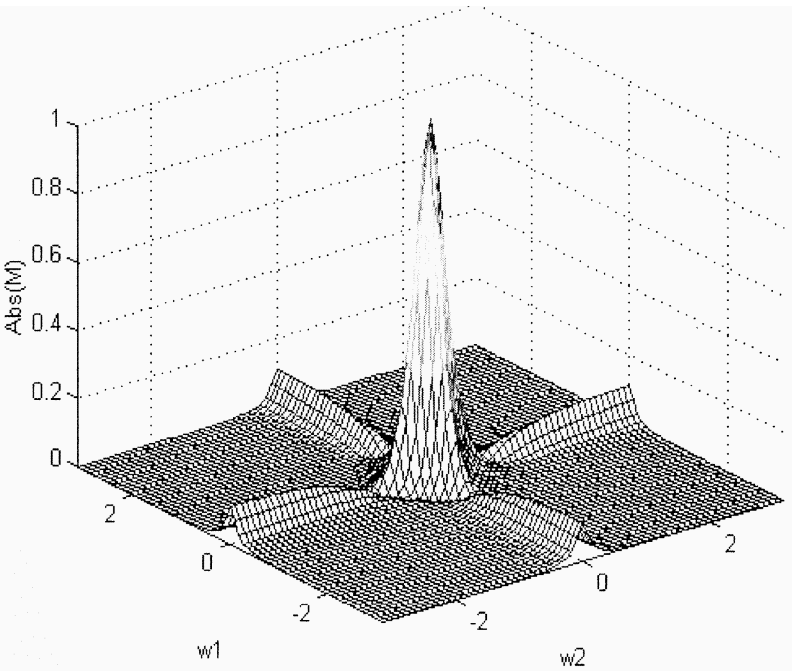


**Figure 4. Amplitude response $|M(\omega_1, \omega_2)|$ of the 2-D filter using the method in [7].**

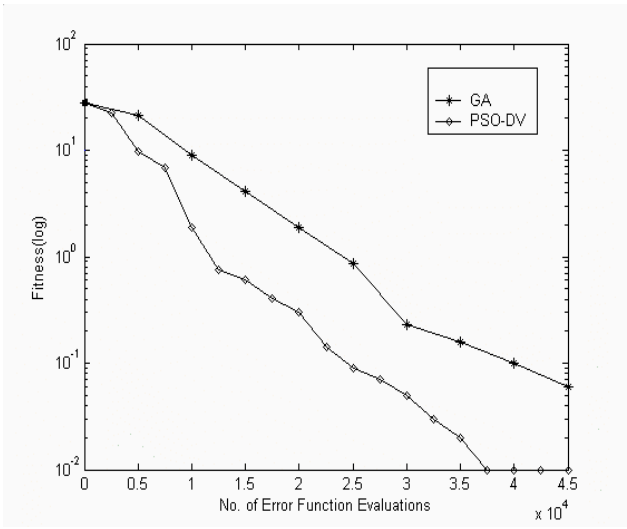


**Figure 5. Performance comparison of GA and PSO-DV algorithms.**

# 6. CONCLUSION

In this paper a new optimization algorithm (developed by modifying the classical PSO) has been applied to a real-world problem of designing 2-D zero-phase recursive filters. The filter thus obtained has a reasonably good stability margin (we have incorporated the stability criteria as constraints to the minimization task). Our method leads to a simpler filter since in practice we have to realize a factorable denominator. Compared to the methods described in [11] and [10], which, to our knowledge, are the most recent and the best-known methods to date, the algorithm used here yields a better design in considerably less time.

# 7. ACKNOWLEDGMENTS

Partial support of UGC-sponsored projects on i) *AI and Expert Systems* and ii) *Excellence Program in Cognitive Science* is acknowledged. We are grateful to five anonymous reviewers for their helpful comments.

# 8. REFERENCES

[1] Chellapilla, K., Fogel, D.B., Rao, S. S. Optimizing IIR Filters using Evolutionary Programming, In *Proceedings of the Adaptive Distributed Parallel Computing Symposium*, Dayton, OH (1996), pp. 252-258.

[2] Das, S., Konar, A., Chakraborty, U. K. Improving particle swarm optimization with differentially perturbed velocity, to appear in *ACM-SIGEVO Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2005)*, Washington D.C., June, 2005.

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

[4] Eberhart, R. C., Shi, Y. Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of IEEE International Congress on Evolutionary Computation,* Vol. 1. (2000) 84-88.

[5] Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley, (1989).

[6] Gorne, T., Schneider, M. Design of digital filters with evolutionary algorithms, *International Conference on Artificial Neural Networks and Genetic Algorithms* (1993), Ref. No. 155-1, pp 1-7.

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

[8] Kennedy, J., Eberhart, R. Particle swarm optimization, In *Proceedings of IEEE International conference on Neural Networks*. (1995) 1942-1948.

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

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

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

[12] Storn, R., Price, K. Differential evolution – a simple and efficient heuristic for global continuous spaces, *Journal of Global Optimization,* 11(4) (1997) 341–359.

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

[14] van den Bergh, F., Engelbrecht, P. A. Effects of swarm size on cooperative particle swarm optimizers. *Proc. GECCO-2001*, San Francisco, CA, (2001) 892-899.