2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Design of Two-Dimensional IIR Digital Filters Using an Improved Quantum-behaved Particle Swarm Optimization Algorithm

Jun Sun, Wei Fang, Wei Chen and Wenbo Xu

Abstract—The design problem of two-dimensional recursive digital filters is generally reduced to a constraint minimization problem. Based on our previously proposed Quantum-Behaved Particle Swarm Optimization (QPSO) algorithm, inspired by quantum physics, we proposed an improved QPSO, called Diversity-Guided QPSO (DGQPSO) to obtain the solution of the design problem. The DGQPSO is implemented by controlling the diversity measure of the swarm to enhance global search ability of the QPSO. The results yield by DGQPSO in design problem show its superiority compared to other competitive optimization algorithm.

I Introduction

Two-dimensional (2-D) IIR filters play an important role in multidimensional digital signal processing (MDSP), since the mayor problems in MDSP can be comprehended by comparing 1-D signal with 2-D signal. They find extensive applications in the domain of denoizing of digital images, biomedical imaging and digital mammography, X-rays image enhancement, seismic data processing, etc. [1]-[3]. During last three decades, two-dimensional (2-D) filter design has received growing attention by researchers and practitioners. The most popular design methods for 2-D IIR filters are based either on an appropriate transformation of 1-D filter [2], [3], or on appropriate optimization techniques such as linear programming, Remex Exchange Algorithm, Non-linear Programming: Gradient Methods, Direct Search methods, Newton and Gauss-Newton Methods, Fletcher-Powell, and Conjugate Gradient [2]-[10]. However, most of the existing methods may result in an unstable filter [1], [3]-[10]. Thus many techniques have been adopted to resolve these instability problems, but the outcome is likely to be a system that has a very small stability margin and hence, not of much practical importance [11], [12].

Just as for 1-D IIR filter design, many modern heuristic methods have also been employed for 2-D IIR design problem, such as GA [11], [38], neural network (NN) [12] and particle swarm optimization (PSO) [13]. These techniques are able to find out better solution than those mentioned in the afore-paragraph. Our present research concentrates on the optimization approach for 2-D IIR filter design based on our previously proposed quantum-behaved particle swarm Optimization (QPSO) algorithm.

The QPSO algorithm, inspired by the principles of quantum mechanics, is a novel variant of the Particle Swarm Optimization (PSO) [14]-[16]. It has fewer parameters and stronger search capability than the PSO, as well as is easy to implement. In our previous work, QPSO has been shown to be more efficient in the design of FIR filters and adaptive IIR filters than the other heuristic techniques, such as PSO, GA. The contribution of this paper is to propose an improved QPSO (called Diversity-Guided QPSO) and use it for 2-D IIR filter design.

Jun Sun is with School of Information Technology, Jiangnan University, No.1800, Lihu Avenue, Wuxi, Jiangsu 214122, China. (Telephone: 0086-510-85912136; fax: 0086-510-85912136; e-mail: sunjun_wx@hotmail.com).

The rest of the paper is organized as follows. Section II describes the problem formulation of IIR filter design. The description of QPSO is given in Section III. In Section IV, a diversity-guided QPSO (DGQPSO) is proposed. Section V presents how to apply DGQPSO to the design problem. Section VI provides the experiment results generated by various optimization algorithms on 2-D IIR filter design. Section VII offers some conclusion.

II. Problem Formulation

The design task of 2-D recursive filters amounts to finding a transfer function $H(z_1, z_2)$ as in (1) such that the function $M(\omega_1, \omega_2) = H(e^{-j\omega_1}, e^{-j\omega_2})$ approximates the desired amplitude response, where the frequencies $\omega_1, \omega_2 \in [-\pi, \pi]$ and $z_1 = e^{-j\omega_1}, z_2 = e^{-j\omega_2}$.

For design purpose, the function $M(\omega_1, \omega_2)$ is equivalent to a class of nonsymmetrical half-plane (NSHP) filters, whose 2-D transfer function $H(z_1, z_2)$ is given by

$$H(z_1, z_2) = H_0 \frac{\sum_{i=0}^{N}\sum_{j=0}^{N} a_{ij} z_1^{\,i} z_2^{\,j}}{\prod_{k=1}^{N}(1 + q_k z_1 + r_k z_2 + s_k z_1 \cdot z_2)}, \quad a_{00} = 1 \tag{1}$$

The variables z_1 and z_2 can be interpreted as complex indeterminants in the discrete Laplace transform (z-transformation).

It is a general practice to take $a_{00} = 1$ (by normalizing a_{ij}'s with respect to the value of a_{00}). The design task for 2-D filter at hand can be reduced to finding a transfer function $H(z_1, z_2)$ in (1) such that the frequency response $H(e^{-j\omega_1}, e^{-j\omega_2})$ approximates the desired amplitude response $M_d(\omega_1, \omega_2)$ as closely as possible. The approximation can be achieved by minimizing [11]-[13]:

$$J = J(a_{ij}, q_k, r_k, s_k, H_0) = \sum_{k_1=0}^{N_1}\sum_{k_2=0}^{N_2}[|M(\omega_1, \omega_2)| - M_d(\omega_1, \omega_2)]^p \tag{2}$$

where

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

and

$$\omega_1 = (\pi\pi/_1)k_1, \omega_2 = (\pi\pi/_2)k_2 \tag{4}$$

and p is an positive even integer (usually p= 2 or 4, 8). Equation (2) can be rewritten as

$$J = \sum_{k_1=0}^{K_1}\sum_{k_2=0}^{K_2}\left[\left|M\left(\frac{\pi k_1}{N_1}, \frac{\pi k_2}{N_2}\right)\right| - M_d\left(\frac{\pi k_1}{N_1}, \frac{\pi k_2}{N_2}\right)\right]^p \tag{5}$$

Here the prime object is to reduce the difference between the desired and actual amplitude responses of the filter at $N_1 \times N_2$ points. For bounded input bounded output (BIBO) stability, the prime requirement is that the z-plane poles of the filter transfer function should lie within the unit circle. Since the denominator contains only first-degree factors, we can assert the stability conditions, following as [11]-[13]:

$$|q_k + r_k| - 1 < s_k < 1 - |q_k - r_k|, \quad k = 1,2,\cdots,N \tag{6}$$

Thus, the design of 2-D recursive filters is equivalent to the following constrained minimization problem

$$\text{Minimize } J = \sum_{k_1=0}^{K_1}\sum_{k_2=0}^{K_2}\left[\left|M\left(\frac{\pi k_1}{K_1}, \frac{\pi k_2}{K_2}\right)\right| - M_d\left(\frac{\pi k_1}{K_1}, \frac{\pi k_2}{K_2}\right)\right]^p \tag{7}$$

$$\text{s.t.} \quad |q_k + r_k| - 1 < s_k \quad k = 1,2,\cdots,N \qquad s_k < 1 - |q_k - r_k| \quad k = 1,2,\cdots,N \tag{8}$$

where p=1 or 2, 4, 8 and K_1 and K_2 are positive integers.

In [11], Mladenov and Mastorakis tackle the design problem with neural networks, and Mastorakis attempts to solve it using a binary coded GA in [12]. Das et al. applied a Particle Swarm Optimization method to the design problem [13].

III. The QPSO Algorithm

The PSO algorithm is a population-based optimization technique originally introduced by Kennedy and Eberhart in 1995 [16] and [17]. A PSO system simulates the knowledge evolvement of a social organism, in which individuals (particles) representing the candidate solutions to the problem at hand fly through a multidimensional search space to find out the optima or sub-optima. The particle evaluates its position to a goal (objective function) at every iteration, and particles in a local neighborhood share memories of their "best" positions, and then use those memories to adjust their own velocities, and thus subsequent positions.

In the original PSO with m individuals, each individual is treated as a volume-less particle in the n-dimensional space, with the position vector and velocity vector of particle i at t^{th} iteration represented as $x_i(t) = [x_{i,1}(t), x_{i,2}(t), \cdots, x_{i,n}(t)]$ and $v_i(t) = [v_{i,1}(t), v_{i,2}(t), \cdots, v_{i,n}(t)]$. The particle moves according to the following equations

$$v_{i,j}(t+1) = w \cdot v_{i,j}(t) + c_1 \cdot r_{1,j}(t) \cdot [y_{i,j}(t) - x_{i,j}(t)] + c_2 \cdot r_{2,j}(t) \cdot [\hat{y}_j(t) - x_{i,j}(t)] \tag{9}$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \tag{10}$$

For i=1, 2…m, j=1, 2,…n, where c_1 and c_2 are called acceleration coefficients. Vector $y_i(t) = [y_{i,1}(t), y_{i,2}(t), \cdots, y_{i,n}(t)]$ is the best previous position (the position giving the best objective function value) of particle i called personal best (pbest) position, and vector $\hat{y}(t) = [\hat{y}_1(t), \hat{y}_2(t), \cdots, \hat{y}_n(t)]$ is the position of the best particle among all the particles in the population and called global best (gbest) position. The parameters $r_{1,j}(t)$ and $r_{2,j}(t)$ are two sequences of random numbers distributed uniformly in (0, 1). Parameter w is called inertia weight. Generally, the value of $v_{i,j}(t)$ is restricted in the interval $[-v_{max}, v_{max}]$. This inertia-weight PSO was introduced by Shi and Eberhart and is called Standard PSO [16].

B. The QPSO Algorithm

The main disadvantage is that the PSO algorithm is not guaranteed to be global convergent. The QPSO algorithm was developed and presented in conference papers such as [14], [15]. This section will present a complete concept and the parameter control method of the QPSO.

Trajectory analyses in [18] demonstrated the fact that convergence of the PSO algorithm may be achieved if each particle converges to its local attractor, $p_i = (p_{i1}, p_{i2}, \cdots p_{iD})$ defined at the coordinates

$$p_{i,j}(t) = \frac{c_1 r_{1,j}(t) y_{i,j}(t) + c_2 r_{2,j}(t) \hat{y}_j(t)}{c_1 r_{1,j}(t) + c_2 r_{2,j}(t)}, \quad 1 \le j \le n \tag{11}$$

In PSO, the acceleration coefficients c_1 and c_2 are generally set to be equal, i.e. $c_1 = c_2$, and thus $\varphi_j(t)$ will be a sequence of uniformly distributed random number within [0,1]. Consequently, equation (12) can be restated as

$$\varphi_j(t) = c_1 r_{1,j}(t) / [c_1(t) r_{1,j}(t) + c_2 r_{2,j}(t)] \tag{12}$$

equation (9). In PSO, the acceleration coefficients c_1 and c_2 are generally set to be equal, i.e. $c_1 = c_2$, and thus $\varphi_j(t)$ will be a sequence of uniformly distributed random number within [0,1]. Consequently, equation (12) can be restated as

$$p_{i,j}(t) = \varphi_j(t) \cdot y_{i,j}(t) + [1 - \varphi_j(t)] \cdot \hat{y}_j(t) \quad \varphi_j(t) \sim U(0,1) \tag{13}$$

It can be seen that p_i is a stochastic attractor of particle i that lies in a hyper-rectangle with y_i and $\hat{y}$ being two ends of its diagonal and moves following y_i and $\hat{y}$. In fact, when the particles are converging to their own local attractors, their personal best positions, local attractors and the global best positions will all converge to one point, leading the PSO algorithm to converge.

We hypothesize that the PSO system is a quantum system, with each particle in a quantum state formulated by wavefunction ψ. $|\psi|^2$ is the probability density function of the position of the particle. Inspired by analysis of convergence of the traditional PSO in [19], we assume that, at iteration t, particle i moves in n-dimensional space with a δ potential well at $p_{i,j}(t)$ on j^{th} dimension for $1 \le j \le n$. Correspondingly, the wavefunction at iteration t+1 is

$$\psi[x_{i,j}(t+1)] = \frac{1}{\sqrt{L_{i,j}(t)}} \exp\left[-\left|x_{i,j}(t+1) - p_{i,j}(t)\right| / L_{i,j}(t)\right] \tag{14}$$

and thus the conditional probability distribution function F is

$$F[x_{ij}(t+1)] = \exp[-2|x_{i,j}(t+1) - p_{i,j}(t)| / L_{i,j}(t)] \quad (15)$$

where $L_{i,j}(t)$ is the standard deviation of the double exponential distribution. Using Monte Carlo method, we can obtain the j^{th} component of position x_i at iteration t+1 as

$$x_{i,j}(t+1) = p_{i,j}(t) \pm \frac{L_{i,j}(t)}{2} \ln[1/u_{i,j}(t)] \quad u_{i,j}(t) \sim U(0,1) \quad (16)$$

where $u_{i,j}(t)$ is a random number uniformly distributed in (0, 1). The value of $L_{ij}(t)$ is evaluated by

$$L_{i,j}(t) = 2\alpha \cdot |C_j(t) - x_{i,j}(t)| \quad (17)$$

where C called Mean Best Position, is defined as the mean of the pbest positions of all particles. That is

$$C(t) = (C_1(t), C_2(t), \cdots, C_n(t)) = \left(\frac{1}{M}\sum_{i=1}^{M} y_{i,1}(t), \ \frac{1}{M}\sum_{i=1}^{M} y_{i,2}(t), \ \cdots, \ \frac{1}{M}\sum_{i=1}^{M} y_{i,n}(t) \right) \quad (18)$$

where M is the population size and y_i is the personal best position of particle i. Hence, the position of the particle updates according to the following equation:

$$x_{i,j}(t+1) = p_{i,j}(t) \pm \alpha \cdot |C_{i,j}(t) - x_{i,j}(t)| \cdot \ln[1/u_{i,j}(t)] \quad u_{i,j}(t) \sim U(0,1) \quad (19)$$

The parameter α in equation (17) and (19) is called Contraction-Expansion (CE) Coefficient, which can be tuned to control the convergence speed of the particle.

Thus the equation (19) is the fundamental iterative equation of the particle's position for the QPSO. Moreover, unlike the PSO, the QPSO needs no velocity vectors for particles at all, and also has fewer parameters to control (only one parameter α except population size and maximum iteration number), making it easier to implement. The experiment results on some well-known benchmark functions show that the QPSO described by the following procedure has better performance than the PSO [14]-[16].

In the QPSO, the parameter α must be set as $\alpha < 1.782$ to guarantee convergence of the particle [16]. In most cases, it can result in good performance to make α decrease linearly from α_0 to α_1 ($\alpha_0 < \alpha_1$) over the running of the QPSO. That is α can be adjusted according to

$$\alpha = (\alpha_0 - \alpha_1) \times (MAXITER - t) / MAXITER + \alpha_1 \quad (20)$$

where MAXITER is the maximum number of iterations and MAXITER is the number of current iteration.

IV. The Proposed Diversity-Guided QPSO

In a PSO system, with the fast information flow between particles due to its collectiveness, diversity of the particle swarm declines rapidly, leaving the PSO algorithm with great difficulties of escaping local optima. In the QPSO, although the search scope of an individual particle at each iteration is the whole feasible solution space of the problem, diversity loss of the whole population is also inevitable. In this paper, we propose a Diversity-Guided QPSO (DGQPSO) in this paper.

The diversity in DGQPSO is measured by average Euclidean distance from the particle's current position to their centroid swarms, namely

$$\text{diversity}[X(t)] = \frac{1}{m \cdot |A|} \cdot \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{n} [x_{i,j}(t) - \bar{x}_j(t)]^2} \quad (21)$$

where

$$X(t) = [x_1(t), x_2(t), \cdots, x_m(t)] \quad (22)$$

and

$$\bar{x}_j(t) = \frac{1}{m} \sum_{i=1}^{m} x_{i,j}(t) \quad (23)$$

In (21), |A| is the length of longest the diagonal in the search space, and n is the dimensionality of the problem. Hence, we may guide the search of the particles with the diversity measures when the algorithm is running.

At the beginning of the search, the diversity of the particle swarm in QPSO is high after initialization. With the development of evolution, the convergence of the particle makes the diversity be declining, which, in turn, is enhancing the local search ability (exploitation) but weakening the global search ability (exploration) of the algorithm. At early or middle stage of the evolution, the declination of the diversity is necessary for the particle swarm to search effectively. However, after middle or at later stage, the particles may converge into such a small region that the diversity of the swarm is very low and further search is impossible. At that time, if the particle with global best position is at local optima or sub-optima, premature convergence occurs.

To avoid the premature convergence and improve the performance of the QPSO, we propose a DGQPSO, in which a low bound for diversity$[X(t)]$ is set to prevent the diversity from constantly declining. The procedure of the algorithm is as follows. After initialization, the algorithm is running in convergence mode that is realized by varying from 1.0 to 0.5 on the course of running. This control method of the parameter is also adopted in the original QPSO and can result in good performance of QPSO generally. On the course of evolution, if the diversity measure of the swarm declines to below the low bound d_{low}, the particles will explode to increase the diversity until it is larger than d_{low}.

We propose a method of exerting the following mutation operation on the particle with global best position if the once diversity measure is smaller than d_{low}.

$$\hat{y}_j(t) = \hat{y}_j(t) + \gamma \cdot |A| \cdot \varepsilon, \quad \varepsilon \sim N(0,1), \ (j = 1,2,\cdots,n) \quad (24)$$

where ε is a random number with standard normal distribution N(0,1), γ is a user-specified parameter. When the mutation operation is exerted, the displacement of the global best particle will make increase value of $|\hat{y}_j(t) - y_{i,j}(t)|$. Thus, the position C will be pulled away from its original position by the displaced global best particle, which, in turn, enlarges the gaps between particles' current position and the position C, consequently making particles' search scope extended and resulting in the gain of diversity$[X(t)]$. The DGQPSO is outlined as follows.

DGQPSO Algorithm:

Step 0: Initialize particles with random position; set the personal position of each particle as $y_i(0) = x_i(0)$;

Step 1: For t=1 to MAXITER, execute the following steps;

Step 2: Calculate the objective function value of each particle's current position and pbest position and determine the gbest position $\hat{y}(t)$;

Step 3: Calculate the mean best position C among the particles and the value of α as

$$\alpha = (\alpha_0 - \alpha_1) \times (MAXITER - t) / MAXITER + \alpha_1;$$

Step 4: Measure the diversity according to formula (24). If $diversity[X(t)] < d_{low}$, the population will be in explosion mode and execute from step through 5 to 6. Or else go to step 7;

Step 5: For each component of gbest position $\hat{y}_j(t)$, implement the mutation operation described in (24).

Step 6: Update the objective function value (fitness value) and return to step 4;

Step 7: Update the position of each particle according to equation (19);

Step 8: Execute the steps 1 through 7.

V. Applications to 2-D Filter Design

A. A Design Example

Without loss of generality, we assume N=2 and then the transfer function (1) can be rewritten as

$$H(z_1,z_2) = H_0 \frac{a_{00} + a_{01}z_2 + a_{02}z_2^2 + a_{10}z_1 + a_{20}z_1^2 + a_{11}z_1z_2 + a_{12}z_1z_2^2 + a_{21}z_1^2z_2 + a_{22}z_1^2z_2^2}{(1+q_1z_1+r_1z_2+s_1z_1z_2)(1+q_2z_1+r_2z_2+s_2z_1z_2)} \quad (25)$$

Now if we substitute z_1 and z_2 in (3), then $M(\omega_1,\omega_2)$ can be

$$M(\omega_1,\omega_2) = H_0 \left[\frac{a_{00} + a_{01}f_{01} + a_{02}f_{02} + a_{10}f_{10} + a_{20}f_{20} + a_{11}f_{11} + a_{12}f_{12} + a_{21}f_{21} + a_{22}f_{22}}{D} - \frac{j[a_{00} + a_{01}g_{01} + a_{02}g_{02} + a_{10}g_{10} + a_{20}g_{20} + a_{11}g_{11} + a_{12}g_{12} + a_{21}g_{21} + a_{22}g_{22}]}{D} \right] \quad (26)$$

with

$$f_{pq} = \cos(i\omega_1 + j\omega_2)$$
$$g_{pq} = \sin(i\omega_1 + j\omega_2) \quad p, q=0,1,2 \quad (27)$$

and

$$D = [(1 + q_1f_{10} + r_1f_{01} + s_1f_{11}) - j(q_1g_{10} + r_1g_{01} + s_1g_{11})] \cdot [(1 + q_2f_{10} + r_2f_{01} + s_2f_{11}) - j(q_2g_{10} + r_2g_{01} + s_2g_{11})] \quad (28)$$

Hence, the compact form of $M(\omega_1,\omega_2)$ is

$$M(\omega_1,\omega_2) = H_0 \frac{A_R - jA_I}{(B_{1R} - jB_{1I})(B_{2R} - B_{2I})} \quad (29)$$

where

$$A_R = a_{00} + a_{01}f_{01} + a_{02}f_{02} + a_{10}f_{10} + a_{20}f_{20} + a_{11}f_{11} + a_{12}f_{12} + a_{21}f_{21} + a_{22}f_{22}$$
$$A_I = a_{00} + a_{01}g_{01} + a_{02}g_{02} + a_{10}g_{10} + a_{20}g_{20} + a_{11}g_{11} + a_{12}g_{12}$$
$$B_{1R} = 1 + q_1f_{10} + r_1f_{01} + s_1f_{11}$$
$$B_{1I} = q_1g_{10} + r_1g_{01} + s_1g_{11}$$
$$B_{2R} = 1 + q_2f_{10} + r_2f_{01} + s_2f_{11}$$
$$B_{2I} = q_2g_{10} + r_2g_{01} + s_2g_{11} \quad (30)$$

The actual magnitude is

$$|M(\omega_1,\omega_2)| = H_0 \sqrt{\frac{(A_R^2 + A_I^2)}{(B_{1R}^2 + B_{1I}^2)(B_{2R}^2 + B_{2I}^2)}} \quad (31)$$

Now we adopted the same example of the design problem as that considered in [11] and [12], where the user-specification for the desired circular symmetric low-pass filter response is given as

$$M_d(\omega_1,\omega_2) = \begin{cases} 1 & \sqrt{\omega_1^2 + \omega_2^2} \le 0.08\pi \\ 0.5 & 0.08\pi < \sqrt{\omega_1^2 + \omega_2^2} \le 0.12\pi \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

A continuous differentiable form of the constraints can be obtained from (6) in the form

$$-(1+s_k) < (q_k + r_k) < (1+s_k)$$
$$-(1-s_k) < (q_k - r_k) < (1-s_k)$$
$$(1+s_k) > 0$$
$$(1-s_k) > 0 \quad (33)$$

Choosing the values p=2, and $N_1 = 50$ and $N_2 = 50$, the corresponding constrained optimization problem (7) becomes

$$\text{Minimize } J = \sum_{k_1=0}^{50}\sum_{k_2=0}^{50}\left[\left|M\left(\frac{\pi k_1}{50},\frac{\pi k_2}{50}\right)\right| - M_d\left(\frac{\pi k_1}{50},\frac{\pi k_2}{50}\right)\right]^2 \quad (34)$$

subject to the constraints imposed by (33) with k=1,2.

B. Representation of the Particle

In order to apply the QPSO and DGQPSO algorithms to the design problem formulated in equation (35), we must need to represent each trial solution as a in a multi-dimensional space. Since a_{00} is always set to 1 in (1), the dimensionality of the present problems is 15 and each particle has 15 positional coordinates represented by the vector

$$X = (a_{01}, a_{02}, a_{10}, a_{11}, a_{12}, a_{20}, a_{21}, a_{22}, q_1, q_2, s_1, s_2, r_1, r_2, H_0)^T \quad (35)$$

In QPSO, the solution of the problem to be resolved is expressed by the particle's position. Then the vector X represents the position of particles in the algorithm. Each coordinate of the vector is real number encoding. In handling the constraints, we select the same method used in [11].

IV. Experiments and Results

We run four population-based optimization algorithms, namely QPSO, DGQPSO, PSO in [13] and [19], and the binary encoded GA suggested by Mastorakis et al. [11] on the design of circular symmetric zero-phase low pass filter according to the user specification summarized in equation (40). All the algorithm discussed here have been developed in MATLAB 7.0 platform on a Pentium IV, 1.9 GHz PC, with 1 MB cache and 512 MB of main memory in Windows Professional 2000 environment. The graphs and figures have been obtained using MATLAB 7.0.

A. Parameter Configurations

As suggested in [11] and [12], we selected the initial value of the parameters of the vector in equation (34) randomly from the interval (-3, 3). In cases of QPSO and DGQPSO, we vary the CE coefficient linearly from 1.0 to 0.5. The inertia weight in the cases of PSO is decreased linearly from 0.9 to 0.4. We employ 20 particles respectively for either of PSO, QPSO and DGQPSO. Other parameter configurations, used in all experiments, have been shown in Table 1.

B. Simulation Results

To judge the accuracy of the algorithms, we firstly run all of them (except the neural network based on reported in [12]) for 40,000 FEs (Function Evaluations). Each algorithm is run independently for 30 times. The best values of the parameters obtained and mean best values of J2 over 30 runs (along with standard deviations) have been reported in Table 2.

A closer look at Fig. 2-7 reveals that the QPSO and DGQPSO algorithm proposed by us, particularly the latter, yield better approximations of desired response as compared to PSO and GA and NN methods in [12]. The ripples in the stopband of Fig. 7 are much lesser as compared to Fig.2-6.

Table 1. Parameter configurations for the competitor algorithms

	Parameter	Value
GA	Popsize m	250
	No. of bits	32
	Mutation probability	0.05
	Part of genetic material interchanged during cross-over	12
	Maximum number of children from each pair of parents	10
PSO	Popsize m	20
	inertia weight w	0.9 to 0.4
	c_1, c_2	2
	V_{max}	3
QPSO	Popsize	20
	CE coefficient α	1.0 to 0.5
DGQPSO	Popsize m	20
	CE coefficient α	1.0 to 0.5
	d_{low}	0.0005

Table 2. The parameters of 2D filter obtained by different algorithms

	NN	GA	PSO	QPSO	DGQPSO
a_{01}	1.8922	1.8162	1.3357	-0.83657	-0.5993
a_{02}	-1.2154	-1.1060	-2.7052	-1.862	-1.9587
a_{10}	0.0387	0.0712	-2.656	-1.6195	-0.0447
a_{11}	-2.5298	-2.5132	0.93592	2.994	1.6032
a_{12}	0.3879	0.4279	0.36258	0.07868	-0.6160
a_{20}	0.6115	0.5926	-1.0322	-1.7383	-2.4065
a_{21}	-1.4619	-1.3690	1.0865	0.54907	-0.3271
a_{22}	2.5206	2.4326	-0.81088	-0.67628	0.8995
q_1	-0.8707	-0.8662	-0.06855	-0.53496	-0.2269
q_2	-0.8729	-0.8907	-0.8659	-0.82565	-0.9187
r_1	-0.8705	-0.8531	-0.55581	-0.34733	-0.4113
r_2	-0.8732	-0.8388	-0.08978	-0.87021	-0.8874
s_1	0.7756	0.7346	-0.27798	-0.01655	-0.2572
s_2	0.7799	0.8025	0.06095	0.72768	0.8304
H_0	0.0010	0.0009	-0.0061	0.002146	-0.0015
J_2	--	18.9614 $\pm$ 4.7974	16.9224 $\pm$ 2.4654	13.9032 $\pm$ 1.9924	12.8112 $\pm$ 1.8260

IIV. Conclusions

In the paper, the design of 2-D recursive filters is attempted by the proposed GDQPSO, which is a novel population based search technique using a diversity control method. It has stronger global search ability and more robust than QPSO and other methods. For the same example, our proposed DGQPSO outperforms those methods presented in [11]-[13] and QPSO because it can find a better approximation of the filter's system function.

References

[1] T. Kaczorek, "Two-dimensional linear system," Springer, Berlin, Germany, 1985.
[2] S.G. Tzafestas (Editor), "Multidimensional system, techniques and applications," Marcel Dekker, New York, 1986.
[3] W.S. Lu, A. Antoniou, "Two-dimensional digital filters," Marcel Dekker, New York, 1992.
[4] G.A. Maria, M.M. Fahmy, "An lp design technique for two-dimensional digital recursive filters," IEEE Trans. Aoust., Speech, Signal Processing, vol. 22, pp. 15-21, Feb. 1974.
[5] C. Charalambous, "Design of 2-dimensional circularly-symmetric digital filters," Proc. Inst. Elect. Eng., pt. G, vol. 129, pp. 47-54, Apr. 1982.
[6] P.K. Rajan, M.N.S. Swamy, "Quadrantal symmetry associated with two-dimensional digital transfer functions," IEEE Trans. Cirucuits Syst., vol. 29, pp. 340-343, June 1983.
[7] T. Lassko, S. Ovaska, "Design and implementation of efficient IIR notch filters with quantization error feedback," IEEE Trans. Instrum. Meas. vol. 43, pp. 449-456, June 1994.
[8] C.-H. Hsieh, C.-M. Kuo, Y.-D. Jou, Y.-L. Han, "Design of two-dimensional FIR digital filters by a two-dimensional WLS technique," IEEE Trans. Circuits Syst. II, wol. 44, pp. 348-412, May 1997.
[9] M. Daniel, A. Willsky, "Efficient implementations of 2-D noncausal IIR filters," IEEE Trans. Circuits Syst. II, vol. 44, pp. 549-563, July1997.
[10] W.-P. Zhu, M.O. Ahmad, M.N.S. Swamy, "A closed-form solution to the least-square design problem of 2-d linear-phase FIR filters," IEEE Trans. Circuits Syst. II, vol, 44, pp. 1032-1039, Dec. 1997.
[11] N. E. Mastorakis, I. F. Gonos, and M. N. S. Swamy, "Design of two-dimensional re cursive filters using genetic algorithms," IEEE Trans. on Circuits and Systems I-Fundamental Theory and Applications, vol. 50, no. 5, pp. 634–639, 2003.
[12] V.M. Mladenow, N.E. Mastorakis, "Design of two-dimensional digital recursive filters by using neural networks," IEEE Trans. On Neural Networks, Vol.12, pp. 585-590, 2001.
[13] S. Das, A. Konar, U.K. Chakraborty, "An efficient evolutionary algorithm applied to the design of two-dimensional IIR filters," in Proc. Genetic and Evolutionary Computation Conference (GECCO200), pp. 2157-2163, Washington, DC, USA, 2005
[14] J. Sun, B. Feng and W.B. Xu, "Particle Swarm Optimization with Particles Having Quantum Behavior," in Proc. of Congress on Evolutionary Computation, pp.326-331, 2004.
[15] J. Sun, Wenbo Xu and Bin Feng, "A Global Search Strategy of Quantum-behaved Particle Swarm Optimization," in IEEE conference on Cybernetics and Intelligent Systems, pp.111-116, 2004.
[16] Y. Shi, R.C. Eberhart, "A Modified Swarm optimizer," in Proc. IEEE int
[17] J. Kennedy, R. C. Eberhart, "Particle swarm optimization," Proc. IEEE Int'l Conference on Neural Networks, IV. Piscataway, NJ: IEEE Service Center, 1995, pp. 1942-1948.
[19] M. Clerc, "The Swarm and Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization," in Proc. Congress on Evolutionary Computation, pp. 1951-1957, 1999.

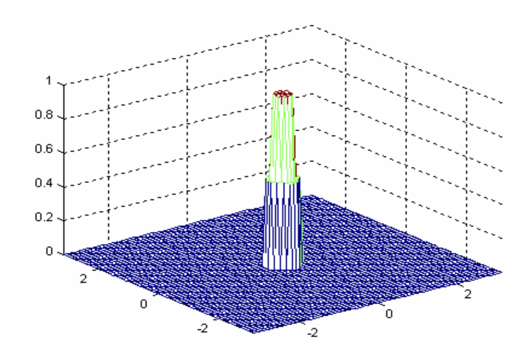

Fig. 2. Desired amplitude response $|M_d(\omega_1,\omega_2)|$ of 2D filter

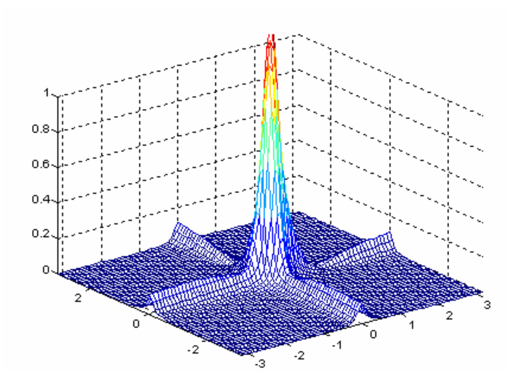

Fig. 3. Amplitude response $|M(\omega_1,\omega_2)|$ of 2D filter using NN in [12]

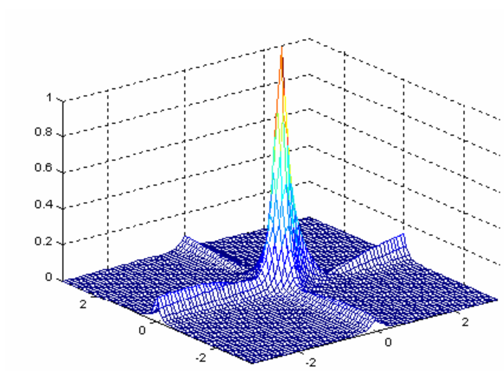

Fig. 4. Amplitude response $|M(\omega_1,\omega_2)|$ of 2D filter using GA in [11]

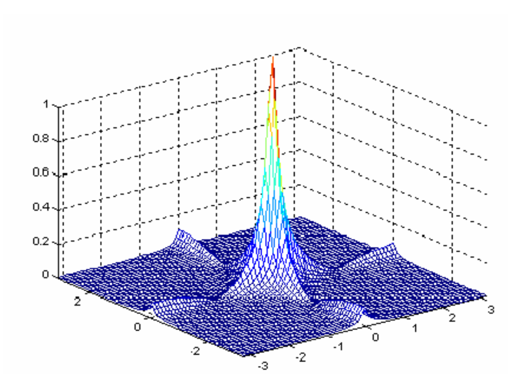

Fig. 5. Amplitude response $|M(\omega_1,\omega_2)|$ of 2D filter using PSO

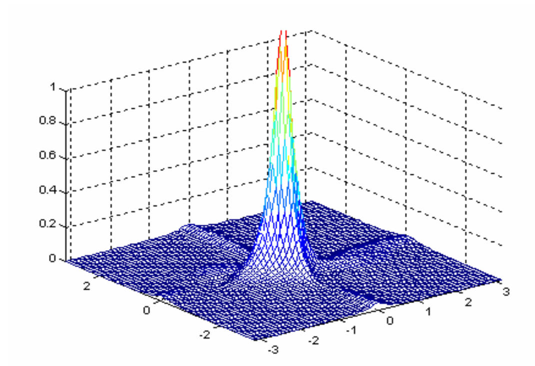

Fig. 6. Amplitude response $|M(\omega_1,\omega_2)|$ of 2D filter using QPSO

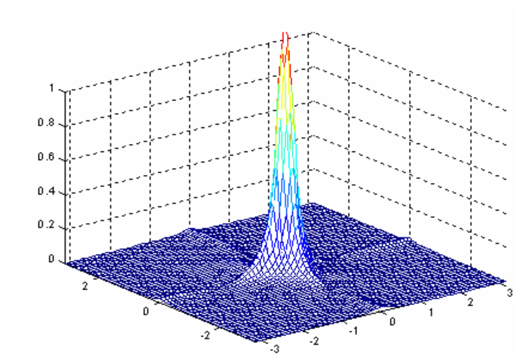

Fig. 7. Amplitude response $|M(\omega_1,\omega_2)|$ of 2D filter using DGQPSO