Particle Swarm Optimization with Quantum Infusion for the Design of Digital Filters

Bipul Luitel

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Particle Swarm Optimization with Quantum Infusion for the Design of Digital Filters

Bipul Luitel, *Student Member IEEE* and Ganesh Kumar Venayagamoorthy, *Senior Member IEEE*

Abstract— In this paper, Particle Swarm Optimization with Quantum Infusion (PSO-QI) has been applied for the design of digital filters. In PSO-QI, Global best (*gbest*) particle (in PSO star topology) obtained from Particle swarm optimization is enhanced by doing a tournament with an offspring produced by Quantum behaved PSO, and selecting the winner as the new *gbest*. Filters are designed based on the best approximation to the ideal response by minimizing the maximum ripples in passband and stopband of the filter response. PSO-QI, as is shown in the paper, converges to a better fitness. This new algorithm is implemented in the design of Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filter.

I. INTRODUCTION

A FILTER is a frequency selective circuit that allows a certain frequency to pass while attenuating the others.

Filters could be analog or digital. In contrast to analog filters which use electronic components such as transistor, resistor, capacitor etc. to perform the filtering operation, digital filters use digital processors which perform mathematical calculations on the sampled values of the signal in order to perform the filter operation. A computer or a dedicated digital signal processor may be used for implementing digital filters. Filters mostly find their use in communication for noise reduction, audio/video signal enhancement etc.

Traditionally, different techniques exist for the design of digital filters. Of these, windowing method is the most popular. In this method, ideal impulse response is multiplied with a window function. There are various kinds of window functions (Butterworth, Chebyshev, Kaiser etc.), depending on the requirements of ripples on the passband and stopband, stopband attenuation and the transition width. These various windows limit the infinite length impulse response of ideal filter into a finite window to design an actual response. But windowing methods do not allow sufficient control of the frequency response in the various frequency bands and other filter parameters such as transition width. Designer always has to compromise on one or the other of the design specifications. So, computational intelligence techniques have been implemented in the design of digital filters to design with better parameter control and to better approximate the ideal filter. Since population based stochastic search methods have proven to be effective in multidimensional nonlinear environment, all of the constraints of filter design can be effectively taken care of by the use of these algorithms.

Previously, computational intelligence based techniques such as particle swarm optimization (PSO) and genetic algorithms (GA) have been implemented in the design of digital filters. Use of PSO and GA in the design of digital filters is described in [1]. Use of differential evolution in the design of digital filters has been implemented in Storn's work [2], [3] and Karaboga's work [4]. Design of infinite impulse response (IIR) filters using PSO is described in [5]. Quantum behaved PSO (QPSO) and its application in filter design has been described in [6] and [7].

In this paper, swarm and quantum algorithms have been applied for the design of digital filters. It shows comparison of performance of PSO, QPSO and particle swarm optimization with quantum infusion (PSO-QI) in the design of FIR and IIR filters as two different cases. In PSO-QI, *gbest* particle obtained from PSO is enhanced by doing a tournament with the offspring obtained from QPSO applied on a randomly chosen particle, and selecting the winner as the new *gbest*. The following sections in the paper are arranged as follows: Digital filters are described in Section 2. In Section 3, PSO-QI algorithm is described and its application in digital filter design is described in Section 4. Results and discussion are given in Section 5 and conclusion in Section 6.

II. DIGITAL FILTERS

Digital filters can be FIR or IIR depending on whether the output of the filter at any given instance is dependent on only the current inputs or on both the current inputs and the past outputs, respectively. An FIR or the non-recursive filter can be described by the transfer function as:

$$H(z) = \sum_{i=0}^{N} a_i z^{-i} \tag{1}$$

and an IIR filter will have a transfer function as shown:

$$H(z) = \frac{\sum_{i=0}^{N} a_i z^{-i}}{1 + \sum_{i=1}^{M} b_i z^{-i}} \tag{2}$$

These parameters $a_0, a_1, a_2, ..., a_N, b_1, b_2, ..., b_M$ appearing in (1) and (2) are called the filter coefficients. These determine the characteristics of the filter. Various other filter parameters which come into picture are the stopband and passband normalized frequencies (ω_s, ω_p), the passband and stopband ripple (δ_p) and (δ_s), the stopband attenuation and the transition width. These parameters are mainly decided by the filter coefficients as is evident from transfer functions in (1) and (2). Significance of these parameters in actual filters with respect to ideal filter is illustrated in Fig. 1. In any filter design problem, some of these parameters are fixed while others are determined. In this paper, swarm and quantum optimization algorithms are applied in order to obtain the actual filter response as close as possible to the ideal response. By properly choosing the filter coefficients, we can design the filter according to our requirement.

IIR filters have both poles and zeros, where as FIR filters have only the zeros. Due to the fact that the poles are located at the origin, they lie within the unit circle and hence FIR filters are inherently stable. Also that FIR filters can be designed as linear phase, which makes them a better choice in phase sensitive applications. IIR filters can achieve much sharper transition region than FIR filters of the same order. Also, IIR filters require less memory and are computationally less complex for the same length of the filter. This makes IIR filters a better choice for hardware implementation. However, due to the feedback element present in it, IIR filter has chances of accumulating the rounding errors over summed iterations.
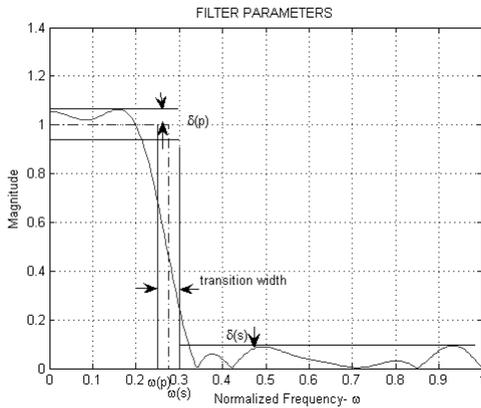

Fig. 1: Illustration of filter parameters.

III. PARTICLE SWARM OPTIMIZATION WITH QUANTUM INFUSION

A. Particle Swarm Optimization

Particle swarm optimization is an evolutionary algorithm developed by Eberhart and Kennedy in 1995 [8]. It is a

population based search algorithm and is inspired by the observation of natural habits of bird flocking and fish schooling. In PSO, a swarm of particles moves through a D dimensional search space. The particles in the search process are the potential solutions, which move around the defined search space with some velocity until the error is minimized or the solution is reached, as decided by the fitness function. The particles reach to the desired solution by updating their position and velocity according to the PSO equations. In PSO model, each individual is treated as a volume-less particle in the D-dimensional space, with the position and velocity of ith particle represented as:

$$X_i = (x_{i1}, x_{i2}, ..., x_{iD}) \qquad (3)$$

$$V_i = (v_{i1}, v_{i2}, ..., v_{iD}) \qquad (4)$$

$$V_{id} = w * V_{id} + c_1 * rand_1() * (P_{id} - X_{id}) + \\ c_2 * rand_2() * (P_g - X_{id}) \qquad (5)$$

$$X_{id} = X_{id} + V_{id} \qquad (6)$$

These particles are randomly distributed over the search space with initial position and velocity. They change their positions and velocity according to (5) and (6) where c_1 and c_2 are cognitive and social acceleration constants, $rand_1()$ and $rand_2()$ are two random functions uniformly distributed in the range of [0,1] and w is the inertia weight introduced to accelerate the convergence speed of PSO [8]. Vector $P_i = (P_{i1}, P_{i2}, ..., P_{iD})$ is the best previous position (the position giving the best fitness value) of particle i called the $pbest$, and vector $P_g = (P_{g1}, P_{g2}, ..., P_{gD})$ is the position of the best particle among all the particles in the population and is called the gbest. X_{id}, V_{id}, P_{id} are the dth dimension of vector of X_i, V_i, P_i.

B. Quantum Particle Swarm Optimization

Quantum behaved particle swarm optimization was introduced by Sun in 2004 [9]. According to the uncertainty principle, position and velocity of a particle in quantum world cannot be determined simultaneously. Thus QPSO differs from traditional PSO mainly in the fact that exact values of x and v cannot be determined. In quantum mechanics, a particle, instead of having position and velocity, has a wavefunction given by:

$$\psi(r,t) \qquad (7)$$

which has no physical meaning but its amplitude squared gives the probability measure of its position in any one dimension r at time t. The governing equation of quantum

mechanics is the Schrodinger's equation given by:

$$j\hbar \frac{\partial}{\partial t}\psi(r,t) = \hat{H}(r)\psi(r,t) \tag{8}$$

where H is a time-independent Hamiltonian operator given by:

$$\hat{H}(r) = -\frac{\hbar^2}{2m}\nabla^2 + V(r) \tag{9}$$

where $\hbar$ is Planck's constant, m is the mass of the particle and $V(r)$ is the potential energy distribution [10]. Based on the probability density function, a particle's probability of appearing in position x can be determined. Therefore in QPSO, a Delta-potential-well based probability density function has been used with center at point $P = (p_1, p_2, .., p_D)$ in order to avoid explosion and help the particles in PSO to converge [11]. Assuming a particle in one-dimensional space having its center of potential at P, normalized probability density function Q and distribution function D_f can be obtained [12]. Let $y=x-p$, then the form of this probability density function is given as follows and depends on the potential field the particle lies in:

$$Q(y) = \frac{1}{L}e^{-2|y|/L} \tag{10}$$

$$D_f(y) = \int_{-\infty}^{y} Q(y)dy = e^{-2|y|/L} \tag{11}$$

where the parameter L is the length of the potential field which depends on the energy intensity and is called the creativity or imagination of the particle that determines its search scope [11].

In QPSO, the search space and the solution space are two different spaces of different quality. So a mapping mechanism is necessary to interpret the position of a particle in solution space by looking at its position in quantized search space. This is called the collapse and is achieved by applying the Monte Carlo simulation. In [9], this has been described as follows:

Let s be any random number uniformly distributed between 0 and $1/L$. For a random number $u=rand(0,1)$, s is defined as:

$$s = \frac{1}{L}u \tag{12}$$

Now, equating (10) and (12), we get:

$$u = e^{-2|y|/L} \tag{13}$$

$$y = \pm\frac{L}{2}\ln(1/u) \tag{14}$$

The position equation is given as follows:

$$x = P \pm \frac{L}{2}\ln(1/u) \tag{15}$$

where the particle's local attractor point $P = (p_1, p_2, .., p_D)$ has coordinates given by the following equation where c_1 and c_2 are two uniformly distributed random numbers:

$$P_d = (c_1 p_{id} + c_2 p_{gd})/(c_1 + c_2) \tag{16}$$

and L can be evaluated as the distance between the particles' current position and point P as follows:

$$L = 2\beta.|P - x| \tag{17}$$

From (15) and (17), the new position of the particle is calculated as:

$$x = P \pm \beta.|P - x|.\ln(1/u) \tag{18}$$

The parameter β is the only parameter of the algorithm. It is called the creativity coefficient and is responsible for the convergence speed of the particle. The term u is a uniformly distributed random number. This Delta-Potential-well based quantum PSO is called the QDPSO. Then an improvement to it is brought by defining a mainstream thought or the Mean Best Position, *mbest*, defined in [12] as:

$$mbest = \frac{1}{M}\sum_{i=1}^{M}P_i = \left(\frac{1}{M}\sum_{i=1}^{M}p_{i1},....,\frac{1}{M}\sum_{i=1}^{M}p_{iD}\right) \tag{19}$$

where M is the size of the population, D is the number of dimensions and p_i is the *pbest* position of each particle. Now the positions update equation in (18) can be written as:

$$x = P \pm \beta.|mbest - x|.\ln(1/u) \tag{20}$$

The pseudocode for the QPSO algorithm is written as follows:

Initialize x, pbest and gbest of the particles.
Do
  For i from 1 to population size
    evaluate fitness
    If fitness (x)<fitness (pbest)
      pbest=x
    gbest=min(pbest)
Calculate mbest

For *d from 1 to dimension size*
    $r_1 = rand(0,1)$
    $r_2 = rand(0,1)$
    $P = (r_1 * p_{id} + r_2 * p_{gd})/(r_1 + r_2)$
    $r_3 = rand(0,1)$
    $L = \beta * abs(mbest - x_{id})$
    If *rand(0,1)>0.5*
       $x_{id} = P - L * ln(1/r_3)$
    else
       $x_{id} = P + L * ln(1/r_3)$
    end
While *termination criteria not met*

C. Particle Swarm Optimization with Quantum Infusion

In this paper, the QPSO has been modified in order to improve its performance. Here, the QPSO has been used to update and guide the *gbest* particle obtained from the traditional PSO. By doing this, the good features of both the algorithms, fast convergence obtained by PSO which is the rate of convergence for first few iterations, and the lower value of average error obtained by QPSO have been utilized and the performance has significantly improved, as is seen in the results and figures. After the position and velocity of the particles are updated using PSO, a randomly chosen particle is utilized to do the QPSO operation and thus create an offspring. The fitness of the offspring is evaluated and the offspring replaces the *gbest* particle of PSO only if it has a better fitness. Thus the *gbest* particle gets improved and pulled towards the solution over iterations. The QPSO algorithm has been infused into the PSO and hence the name particle swarm optimization with quantum infusion. This has been shown in the flowchart in Fig. 2.

IV. FILTER DESIGN USING PSO-QI

From (1), transfer function of the FIR filter can also be represented as:

$$\frac{y_n}{x_n} = a_0 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n} \qquad (21)$$

or from (2), an IIR filter can have the following transfer function:

$$\frac{y_n}{x_n} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_m z^{-m}} \qquad (22)$$

Now for (21), the numerator coefficient vector { $a_0, a_1, a_2,$ ...., a_N} is represented in N dimensions where as for (22), the numerator as well as denominator coefficient vector is { $a_0, a_1, a_2,$ ...., $a_N, b_0, b_1, b_2,$ ...., b_M} which is represented in $(N+M)$ dimensions. The particles are distributed in a D dimensional search space, where $D = N$ for FIR and $D = (N+M)$ for IIR filter. The

position of the particles in this D dimensional search space represents the coefficients of the transfer function. In each iteration, these particles find a new position, which is the new set of coefficients. Fitness of particles is calculated using the new coefficients. This fitness is used to improve the search in each iteration, and result obtained after a certain number of iterations or after the error is below a certain limit is considered to be the final result.
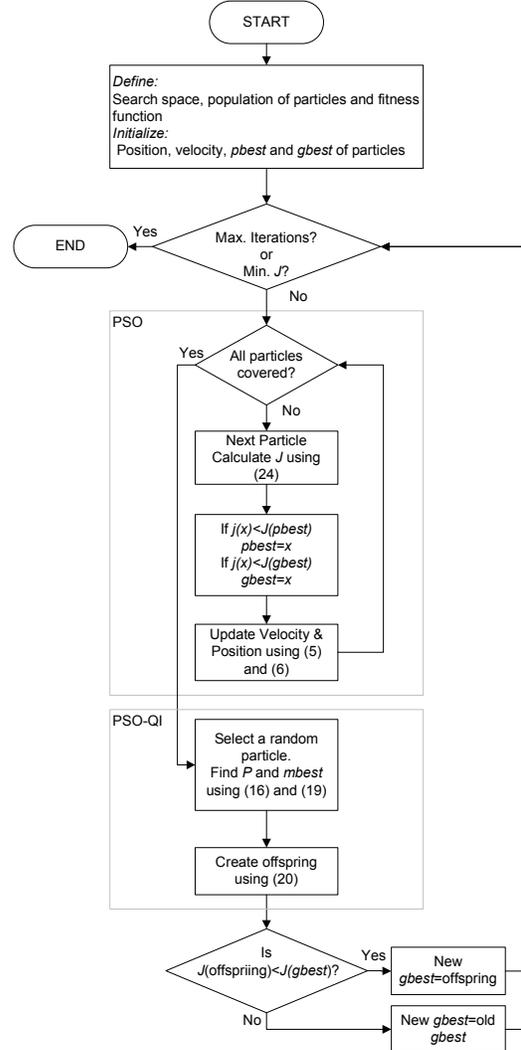

Fig. 2: Flowchart for filter design using PSO-QI

Different kinds of fitness functions have been used in different literature. An error function given by (23) is the approximate error used in Parks-McClellan algorithm for filter design.

$$E(w) = G(w)[H_d(e^{jw}) - H(e^{jw})] \qquad (23)$$

where $G(w)$ is the weighting function used to provide different weights for the approximate errors in different frequency bands, $H_d(e^{jw})$ is the frequency response of the

desired filter and $H(e^{jw})$ is the frequency response of the approximate filter [2].

Now the error to be minimized is defined as:

$$J = \underset{w \le w_p}{Max}(|E(\omega)| - \delta_p) + \underset{w \ge w_s}{Max}(|E(\omega)| - \delta_s) \quad (24)$$

where δp and δs are the ripples in the passband and stopband, and ω_p and ω_s are passband and stopband normalized cut off frequencies respectively. The algorithms try to minimize this error and thus increase the fitness.

V. Studies and Results

Two different cases have been studied. The error shown in the results is the average of 50 trial runs. The magnitude and gain plots are for any random trial. In Case I, an FIR filter has been designed. In Case II an IIR filter has been designed. The values of PSO parameters used in the study are based on the best parameters report in literature [13]. Alternative values of PSO parameters were also considered in the study. Studies carried out with fixed inertia weight of 0.8 instead of having it linearly decreasing showed that PSO gets stuck very early in the search process, whereas no significant impact was seen with on PSO-QI. This further supports its effectiveness against PSO. The specifications of the filter and the number of filter coefficients are taken from literature [1, 6] in order to evaluate the performance of PSO-QI.

The parameters of the filter to be designed are as follows:
- Passband ripple (δ_p) = 0.1
- Stopband ripple (δ_s) = 0.01
- Passband normalized cutoff frequency (ω_p) = 0.45
- Stopband normalized cutoff frequency (ω_s) = 0.55

The parameters of the algorithm are as follows:
- β = linearly increasing from 0.5 to 1
- w = linearly decreasing from 0.9 to 0.4
- c_1, and c_2 = 2
- Population size = 25
- Number of iterations = 500
- Number of trials = 50

Case I: FIR Filter
Dimension of a particle = Number of filter coefficients in (21) = 20

Case II: IIR Filter
Dimension of a particle = Number of filter coefficients in (22) = 20
Number of numerator coefficients = 10
Number of denominator coefficients = 10

Based on (24), the error graph for Case I is shown in Fig. 3. It clearly shows that PSO-QI performed much better than PSO in terms of the fitness. The magnitude and the gain plot for the designed FIR filter are shown in Figs. 4 and 5 respectively. This shows significant improvement on the ripples at the passband and stopband by the use of PSO-QI. The similar results for IIR filter designed in Case II are shown in Figs. 6, 7 and 8. From the figures, it is seen that PSO could better approximate the filter coefficients in Case II than in Case I. It is also observed that the IIR filters showed sharper cut-off than FIR filters at transition band. The comparison of the values obtained from both the algorithms for the two cases are summarized in Table 1. The minimum, maximum and average values of ripples in the passband and stopband for each case have been tabulated. Standard deviation of the minimum values is also shown in the table. The lower values of standard deviation show that PSO-QI is more consistent and always converges to a much lower error. The lower values of average error also confirm the effectiveness of PSO-QI for the given problem. The results also show that PSO-QI allows better control of filter parameters than traditional PSO. However, the time required for one run by PSO-QI is almost twice as that taken by PSO.

TABLE I
PASSBAND AND STOPBAND RIPPLES (WITH 500 ITERATIONS)

		PSO		PSO-QI	
		Case I	Case II	Case I	Case II
Time (s)	Avg.	16.617	16.187	31.223	31.568
	Min.	15.781	16.900	30.031	30.219
Error	Avg.	0.311	0.208	0.043	0.060
	Min.	0.015	0.010	0.022	0.003
	Std.	0.477	0.154	0.016	0.043
Passband (δ_p)	Avg.	0.203	0.156	0.101	0.153
	Min.	0.099	0.098	0.095	0.091
	Max.	1.377	0.442	0.124	0.808
	Std.	0.251	0.080	0.005	0.148
Stopband (δ_s)	Avg.	0.217	0.164	0.056	0.087
	Min.	0.025	0.020	0.035	0.013
	Max.	1.010	0.662	0.123	0.793
	Std.	0.297	0.123	0.017	0.109

Figs. 9, 10 and 11 show the comparison of PSO, QPSO and PSO-QI in terms of error, magnitude and gain plot respectively for an FIR filter. This experiment is carried out for 1500 iterations because of the slow convergence of QPSO. Although QPSO seems to converge better than PSO in greater number of iterations, PSO-QI shows much better performance in terms of lowest value of error achieved as well as the rate of convergence, which is its ability to reach that value in less number of iterations. Another study is carried out to test the performance of PSO while running it

for as much time as is taken by PSO-QI. As PSO takes one half of the time taken by PSO-QI, it is run for twice the number of iterations. The error graph thus obtained is shown in Fig. 12. It shows that there is no significant change in error after a certain time and thus performance of PSO-QI is still better.
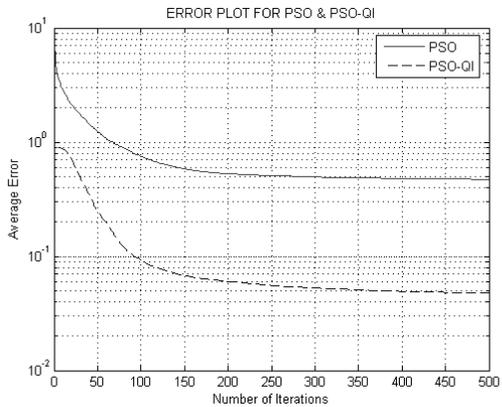

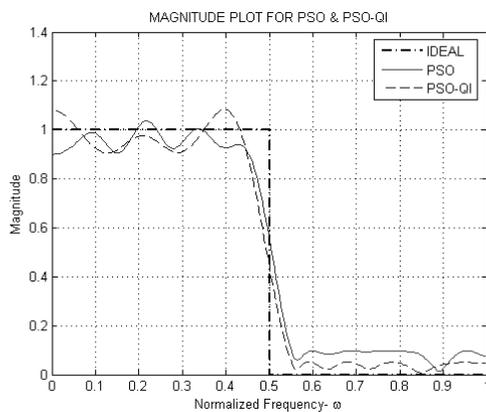Fig. 3: Error graph for the FIR filter designed in Case I.

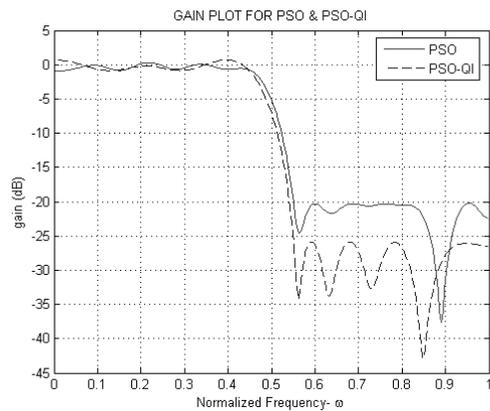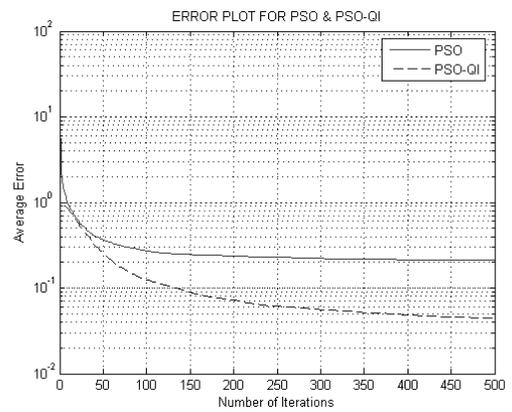Fig. 4: Magnitude plot for the FIR filter designed in Case I.

Fig. 5: Gain plot for the FIR filter designed in Case I.

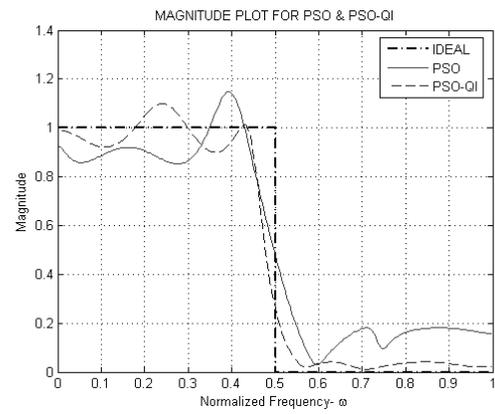Fig. 6: Error graph for the IIR filter designed in Case II.

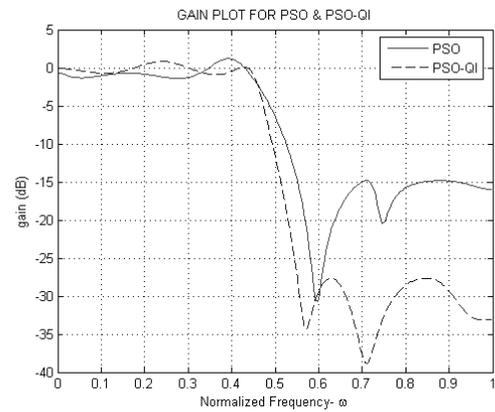Fig. 7: Magnitude plot for the IIR filter designed in Case II.

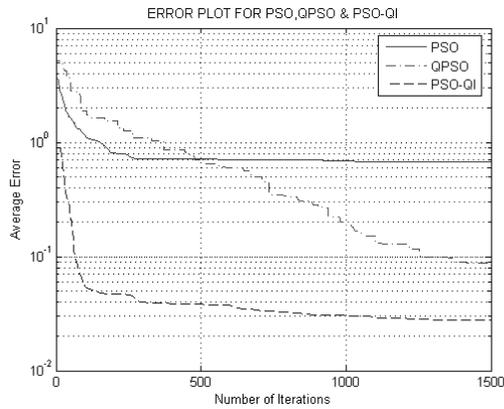Fig. 8: Gain plot for the IIR filter designed in Case II.

ERROR PLOT FOR PSO,QPSO & PSO-QI

Fig. 9: Error graph showing the comparison of PSO, QPSO and PSO-QI.

MAGNITUDE PLOT FOR PSO, QPSO & PSO-QI

Fig. 10: Magnitude plot showing the comparison of PSO, QPSO and PSO-QI.

GAIN PLOT FOR PSO, QPSO & PSO-QI
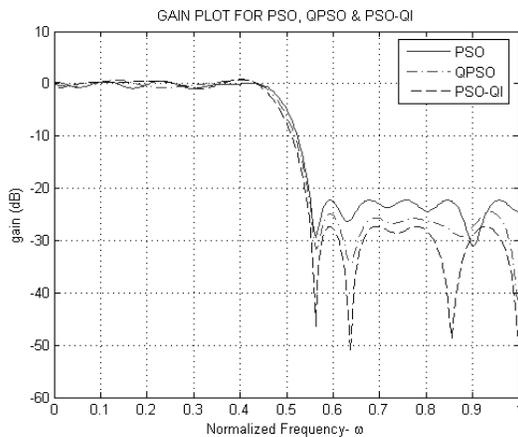
Fig. 11: Gain plot showing the comparison of PSO, QPSO and PSO-QI.

PSO AND PSO-QI RUN FOR EQUAL TIME

Fig. 12: Error graph for PSO run for the same amount of time as PSO-QI.

VI. CONCLUSION

Digital filters, both FIR and IIR are designed using two different algorithms are compared in the paper. Of these, PSO-QI showed much better results than traditional PSO. Although it took longer for the algorithm to converge, it found much better solution than PSO and QPSO. The results are not tabulated for QPSO because of the higher number of iterations and the results are clear from the figures. Hence, it can be concluded that swarm and quantum algorithms can be effectively used in digital filter design, and PSO-QI is a better choice. It is evident from the figures and results how the best features of two algorithms can be extracted and performance can be improved by hybridization of these algorithms. However, there is more room for improving the algorithm. Some experiments carried out by changing some of the parameters of the algorithm, such as replacing *mbest* and x_{id} by *gbest* and *pbest* particles in the QPSO algorithm gave better results, which is not within the scope of this paper. However, this could be problem specific and needs further research. Also, instead of simply evolving the *gbest* particle, a whole population of offspring could be created and allowed to replace the parents in next iteration depending on their fitness. This also continues to be the authors' future work. Modification of fitness function for the filter design to incorporate transition width, and thus allow for a much wider choice of parameter trade-off based on the designer's requirements also remains to be explored.

REFERENCES

[1] J. I. Ababneh, M. H. Bataineh "Linear phase FIR filter design using particle swarm optimization and genetic algorithms," *Digital Signal Process* (2007), doi: 10.1016/j.dsp.2007.05.011.

[2] R. Storn, "Differential Evolution Design of an IIR Filter", in *Proc. Of IEEE Int. Conf. Evolutionary Computation*, May 1996, pp. 268-273.

[3] R. Storn, "Designing Nonstandard Filters with Differential Evolution," *Signal Processing Magazine*, *IEEE*, vol. 22, Jan. 2005, pp. 103-106.

[4] N. Karaboga, Digital IIR Filter Design Using Differential Evolution

Algorithm, EURASIP Journal on Applied Signal Processing 2005:8, pp. 1269-1276

[5] D. J. Krusienski, W. K. Jenkins, "Particle Swarm Optimization for Adaptive IIR Filter Structures," in *Cong. On Evolutionary Computation*, June 2004, vol. 1, pp. 965-970.

[6] W. Fang, J. Sun, W. Xu, J. Liu, "FIR Digital Filters Design Based on Quantum-behaved Particle Swarm Optimization", in Proc. Of the First Intl. Conf. on Innovative Computing, Information and Control

[7] W. Fang, J. Sun, W. Xu, "Analysis of Adaptive IIR Filter Design Based on Quantum-behaved Particle Swarm Optimization", Sixth World Congress on Intelligent Control and Automation, vol. 1, 2006, pp. 3396-3400

[8] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, R. G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems", *in IEEE Trans. on Evolutionary Computation*, Accepted for future publication.

[9] J. Sun, B. Feng, W. Xu, "Particle Swarm Optimization with Particles having Quantum Behavior", Congress on Evolutionary Computation, vol. 1, June 2004, pp. 325-331

[10] S. M. Mikki and A. A. Kishk, "Quantum Particle Swarm Optimization for Electromagnetics", IEEE Trans. On Antennas and Propagation, vol. 54, Oct. 2006

[11] J. Sun, W. Xu, B. Feng, "Global search strategy of quantum-behaved particle swarm optimization", IEEE Conf. on Cybernetics and Intelligent Systems, vol. 1, Dec 2004, pp. 111-116

[12] J. Sun, W. Xu, B. Feng, "Adaptive Parameter Control for Quantum-behaved Particle Swarm Optimization on Individual Level", IEEE Intl. Conf. on Systems, Man and Cybernetics, vol 4, Oct. 2005, pp. 3049-3054

[13] R. C. Eberhart, Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", Proceedings of the Congress on Evolutionary Computation, vol.1, Jul. 2000, pp. 84-88.