International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:5, No:12, 2011

IIR Filter design with Craziness based Particle Swarm Optimization Technique

Suman Kumar Saha, Rajib Kar, Durbadal Mandal, S. P. Ghoshal

***Abstract*—**This paper demonstrates the application of craziness based particle swarm optimization (CRPSO) technique for designing the 8th order low pass Infinite Impulse Response (IIR) filter. CRPSO, the much improved version of PSO, is a population based global heuristic search algorithm which finds near optimal solution in terms of a set of filter coefficients. Effectiveness of this algorithm is justified with a comparative study of some well established algorithms, namely, real coded genetic algorithm (RGA) and particle swarm optimization (PSO). Simulation results affirm that the proposed algorithm CRPSO, outperforms over its counterparts not only in terms of quality output i.e. sharpness at cut-off, pass band ripple, stop band ripple, and stop band attenuation but also in convergence speed with assured stability.

***Keywords*—**IIR Filter; RGA; PSO; CRPSO; Evolutionary Optimization Techniques; Low Pass (LP) Filter, Magnitude Response; Pole-Zero Plot; Stability.

I. INTRODUCTION

SIGNAL is the carrier of information which is germinated in almost every field of science and engineering, opens a fairly large scope of research. On the time dependence, it can be classified as, continuous time and discrete time signals. Mathematical manipulation of signal and data for digital signal processing (DSP) is built upon the platform of discrete time instants. Owing to the tremendous growth of microelectronics and enhancement of computational power, filter design in DSP has become an apex interest for many scholars for last few years. In signal processing, a filter is mainly used to extract the useful portion of interest and remove the unwanted portion such as noise, which could be generated due to unavoidable circumstances of the environment, from the input signal. Filters are broadly categorized as analog and digital ones, on the basis of filtering process and physical design approach. Analog filter uses electronic components such as, resistors, capacitors and op-amps to realize its effectiveness in the field of noise reduction, video signal enhancement, and graphic equalizer in hi-fi system and so on. Discrete component dependent design, prone to high component tolerance sensitivity, large physical size, poor accuracy and highly susceptible to thermal drift are the major retractions of analog filter implementation. Physical size, poor accuracy and highly susceptible to thermal drift are the major retractions of analog filter implementation. On the contrary, digital filter performs mathematical operation on a sampled, discrete timed signal to achieve the desired features with the help of a specially designed digital signal processor (DSP) chip or a processor used in a general purpose computer.

Suman Saha, Rajib Kar, Durbadal Mandal are with the Department of Electronics and Communication Engg., National Institute of Technology, Durgapur, West Bengal, 713029 (e-mail: rajibkarece@gmail.com).

S.P. Ghoshal is with the Department of Electrical Engg., National Institute of Technology, Durgapur, West Bengal, 713029. (spghoshalnitdgp@gmail.com)

Digital filters are broadly classified into two main categories namely; finite impulse response (FIR) filter and infinite impulse response (IIR) filter [1-2]. The output of FIR filter depends on present and past values of input, that is why, it is also called non-recursive filter. On the other hand the output of IIR filter not only depends on previous inputs, but also on the previous outputs with impulse responses continuing for infinite time, at least theoretically. Recursive means, feeding the output of some mathematical operation to the input of the same system for the calculation of the current output. For this purpose a large memory is required to store the previous outputs.

Hence, FIR filter realization is easier with the requirement of less memory space and design complexity. Ensured stability and linear phase response over wide frequency range are the additional advantages. On the hand, IIR filter distinctly meets the design specification of sharp transition band width, pass band ripple and stop band attenuation with ensured lower order as compared to FIR filter [3-4]. As a consequence, the lower order IIR filter with similar performance of FIR filter can be implemented with minimum number of digital multipliers and delay elements for hardware realization and minimum computational time for software implementation [2], [4]. Due to this challenging feature with wide field of applications, performances of IIR filters designed using various evolutionary algorithms are compared to find out the effectiveness of an algorithm.

In the conventional method, IIR filters of various types (Butterworth, Chebyshev and Elliptic) can be implemented by two methods. In the first case frequency sampling technique is adopted for Least Square Error [5] and Remez Exchange [6] process. In the second method, filter coefficients and minimum order are calculated for a prototype LP filter in analog domain, which are transformed to digital domain with bilinear transformation. This frequency mapping works well at low frequency, but in high frequency domain this method is liable to frequency warping [4].

IIR filter design is a highly challenging optimization problem. So far, gradient based classical algorithms such as steepest descent and quasi Newton algorithms have been used aptly for the design of IIR filters [7-8]. In general, these algorithms are very fast and efficient to obtain the optimum solution of the objective function for a unimodal problem. But the error surface (typically the mean square error between the desired response and estimated filter output) of IIR filter is multimodal, so global optimization techniques are required.

The shortfalls of classical optimization techniques for which they are inefficient for handling the global optimization problem are as follows: i) requirement of continuous and differentiable cost or objective function; ii) usually converges to the local optimum solution or revisits the same suboptimal solution; iii) incapable to search the large problem space; iv) requirement of the piecewise linear cost approximation (linear programming); v) highly sensitive to starting points when the number of solution variables are increased and as a result, the solution space also gets increased. So, it can be concluded that classical optimization techniques are only suitable for handling differentiable unimodal objective function with constricted search space. But the error surface of IIR filter is usually multimodal and non-differentiable. Different heuristic search algorithms are proposed for this purpose like Genetic Algorithm (GA), inspired by the Darwin's "Survival of the Fittest" strategy [9-11]; Simulated Annealing (SA) designed by the thermodynamic effects [12]; Artificial Immune Systems (AIS) mimics the biological immune systems [13]; Ant Colony Optimization (ACO) simulates the ants' food searching behavior [14]; Bacterial Foraging Algorithm (BFA) is based on food searching nature of bacteria [15-16]; and Particle Swarm Optimization (PSO) simulates the behavior of bird flocking or fish schooling [17-20] etc.

In this paper the capability of global searching and near optimum result finding features of GA, PSO and CRPSO are investigated thoroughly for solving IIR filter design problem. GA is a probabilistic heuristic search optimization technique developed by Holland [9]. The features such as multi-objective, coded variable and natural selection make this technique distinct and suitable for finding the near global solution of filter coefficients. On the other hand Particle Swarm Optimization (PSO) is an evolutionary algorithm developed by Eberhart *et al.* [21-22]. Several attempts have been taken to design digital filter with conventional PSO and its modified version [3], [23]. The key attraction of PSO is its simplicity in calculation and very less number of steps in algorithm. The limitations of the classical PSO are premature convergence and stagnation problem [24-25]. To overcome these problems, an improved version of PSO, called craziness based particle swarm optimization (CRPSO) technique is suggested by the authors for low pass IIR filter design.

CRPSO is a global search algorithm originated from PSO, mimics the particle behaviors of a swarm in a very closely manner. CRPSO has adopted the special features such as abrupt change of velocity; a craziness factor; and change of direction of flying towards an apparently non-promising area of food depending upon particle's mood enhances the usefulness of this algorithm towards the design of low pass IIR filter. The paper is organized as follows: Section II describes the filter design problem in hand; different evolutionary algorithms namely, RGA, PSO and CRPSO are discussed in section III; section IV consists of comprehensive and demonstrative sets of data and illustrations that articulate the usefulness of the present work in terms of results and discussion.

II. Low Pass IIR Filter Formulation

This section discusses the design strategy of IIR filter based on CRPSO. The input-output relation is governed by the following difference equation:

$$y(p)+\sum_{k=1}^{n}a_k y(p-k)=\sum_{k=0}^{m}b_k x(p-k)$$

where $x(p)$ and $y(p)$ are the filter's input and output, respectively, and $n \geq (m)$ is the order of the filter. The transfer function of the IIR filter is expressed as follows:

$$H(z)=\frac{\sum_{k=0}^{m}b_k z^{-k}}{1+\sum_{k=1}^{n}a_k z^{-k}} \tag{1}$$

Let $z=e^{j\Omega}$. Then, the frequency response of the IIR filter becomes

$$H(\Omega)=\frac{\sum_{k=0}^{m}b_k e^{-jk\Omega}}{1+\sum_{k=1}^{n}a_k e^{-jk\Omega}} \tag{2}$$

or,

$$\begin{aligned}H(\Omega)&=\frac{Y(\Omega)}{X(\Omega)}\\&=\frac{b_0+b_1e^{-j\Omega}+b_2e^{-j2\Omega}+\ldots+b_me^{-jm\Omega}}{1+a_1e^{-j\Omega}+a_2e^{-j2\Omega}+\ldots+b_ne^{-jn\Omega}}\end{aligned} \tag{3}$$

where $\Omega=2\pi\left(\frac{f}{f_s}\right)$ in [0, π] is the digital frequency; f is the analog frequency; and f_s is the sampling frequency. The commonly used approach for IIR filter design is to represent the problem as an optimization problem with the mean square error (MSE) as the cost function [23] shown in (4).

$$J(\omega)=E[e^2(p)]=E[(d(p)-y(p))^2] \tag{4}$$

where $d(p)$ and $y(p)$ are the filter's desired and actual responses, respectively.

The difference $e(p)=(d(p)-y(p))$ is the filter's error signal. The design goal is to minimize the cost function $J(\omega)$ with proper adjustment of coefficient vector ω represented as:

$$\omega=[a_0a_1\ldots a_nb_0b_1\ldots b_m]^T.$$

The cost function is usually expressed as the time averaged cost function defined by (5).

$$J_1(\omega)=\frac{1}{N_s}\sum_p\ [(d(p)-y(p))^2] \tag{5}$$

where N_s is the number of samples used for the computation of the cost function.

In this paper, a novel error fitness function has been adopted in order to achieve higher stop band attenuation and to have a good control on the transition width. The fitness function used in this paper is given in (6). Using (6), it is found that the proposed filter deign approach results in considerable improvement over other optimization techniques.

$$J_2(\omega) = \sum abs\left[abs\left(\left|H(\omega)\right| - 1\right) - \delta_p\right] + \sum \left[abs\left(\left|H(\omega)\right| - \delta_s\right)\right] \quad (6)$$

For the first term of (6), $\omega \in$ pass band including a portion of the transition band and for the second term of (6), $\omega \in$ stop band including the rest portion of the transition band. The portions of the transition band chosen depend on pass band edge and stop band edge frequencies.

The error function given in (6) represents the generalized fitness function to be minimized using the evolutionary algorithms. The algorithms try to minimize this error and thus improve the filter performance. Unlike other error fitness functions as given in (4) and (5) which consider only the maximum errors, J_2 involves summation of all absolute errors for the whole frequency band, and hence, minimization of J_2 yields much higher stop band attenuation and lesser pass band ripples.

III. Evolutionary Techniques Employed

A. Real Coded Genetic Algorithm (RGA)

Standard Genetic Algorithm (also known as real coded GA) is mainly a probabilistic search technique, based on the principles of natural selection and evolution built upon the Darwin's "Survival of the Fittest" strategy. Each encoded chromosome that constitutes the population is a solution to the filter designing optimization problem. Chromosomes are constructed over some particular alphabet {0, 1}, so that chromosomes' values are uniquely mapped onto the real decision variable domain. Each chromosome is evaluated by a function known as fitness function, which is usually the fitness function or objective function of the corresponding optimization problem [9-11], [26].

Steps of RGA as implemented for the optimization of coefficient vector ω are as follows [19-20]:

Step 1: Initialize the real coded chromosome strings of n_p population, each consists of a set of numerator and denominator filter coefficients b_k and a_k, respectively. Size of the set depends on the number of filter coefficients for a particular order of the filter to be designed.

Step 2: Decoding the strings and evaluation of absolute error with cost function, $J_2(\omega)$.

Step 3: Selection of elite strings in order of increasing cost values from the minimum value.

Step 4: Copying the elite strings over the non-selected strings.

Step 5: Crossover and mutation generate offspring.

Step 6: Genetic cycle updating.

Step 7: The iteration stops when maximum number of cycles is reached. The grand minimum cost and its corresponding chromosome string having the desired optimal IIR LP filter coefficients are finally obtained.

B. Particle Swarm Optimization (PSO)

PSO is flexible, robust, population based stochastic search algorithm with attractive features of simplicity in implementation and ability to quickly converge to a reasonably good solution. Additionally, it has the capability to handle larger search space and non-differential objective function, unlike traditional optimization methods. Eberhart *et al.* [21-22] developed PSO algorithm to simulate random movements of bird flocking or fish schooling.

The algorithm starts with the random initialization of a swarm of individuals, which are known as particles within the multidimensional problem search space. In which each particle tries to move toward the optimum solution, where next movement is influenced by the previously acquired knowledge of particle's best and global best positions once achieved by the individual and the entire swarm. The features incorporated within this simulation are velocity matching of individuals with the nearest neighbor, elimination of ancillary variables and inclusion of multidimensional search and acceleration by distance. Instead of the presence of direct recombination operators, acceleration and position modification supplement the recombination process in PSO. Due to the aforementioned advantages and simplicity, PSO has been applied to different fields of practical optimization problem [17-18], [21], [27-30].

To some extent, IIR filter design with PSO is already reported in [3] and [30]. A brief idea about the algorithm for a D-dimensional search space with N particles that constitutes the flock is presented here. The i^{th} particle is described by a position vector as: $S_i = (s_{i1}, s_{i2}, \ldots, s_{iD})^T$ and velocity is expressed by: $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^T$, the best position that the i^{th} particle has reached previously: $pbest_i = (p_{i1}, p_{i2}, \ldots, p_{iD})^T$, and the group best is expressed as: $gbest = (p_{g1}, p_{g2}, \ldots, p_{gD})^T$. The maximum and minimum velocities are: $V_{max} = (v_{max1}, v_{max2}, \ldots, v_{maxD})^T$ and $V_{min} = (v_{min1}, v_{min2}, \ldots, v_{minD})^T$, respectively.

The basic steps of this algorithm are as follows:

Step 1: Initialize the swarm of N particles with random positions and velocities in D-dimensional search space with the ability of random movement for each particle in the entire search space.

Step 2: Compute the value with predefined cost function for the current position S_i of each particle.

Step 3: Each particle can remember its best position (pbest) which is known as cognitive information and that could be updated with each iteration.

Step 4: Each particle can also remember the best position the swarm has ever attained (gbest) which is called social information and could be updated in each iteration cycle.

Step 5: Velocity and position of the particle are modified according to equations (7), (8) and (9) [31].

$$V_i^{(k+1)} = w * V_i^{(k)} + C_1 * rand_1 * \{pbest_i^{(k)} - S_i^{(k)}\} + C_2 * rand_2 * \{gbest_i^{(k)} - S_i^{(k)}\} \quad (7)$$

$$V_i = V_{\max} \quad \text{for } V_i > V_{\max}$$
$$= V_{\min} \quad \text{for } V_i < V_{\min} \quad (8)$$

$$S_i^{(k+1)} = S_i^{(k)} + V_i^{(k+1)} \quad (9)$$

Step 6: The iteration stops when maximum number of cycles or any predefined stopping criteria is reached.

The positive constants C_1, C_2 are related with accelerations and $rand_1$, $rand_2$ lies in the range [0, 1]. The inertia weight w is a constant and chosen carefully to obtain fast convergence to optimum result. K denotes the iteration number.

This algorithm is applied to design the IIR filter, in which real coded n_p (population size) vectors, each consists of filter coefficients, are considered and filter order determines the number of components in each vector.

C. Craziness based Particle Swarm Optimization (CRPSO)

The global search ability of above discussed conventional PSO is improved with the help of the following modifications. This modified PSO is termed as craziness based particle swarm optimization (CRPSO).

The velocity in this case can be expressed as follows [32]:

$$\begin{aligned} V_i^{(k+1)} &= r_2 * sign(r_3) * V_i^{(k)} + \\ &(1-r_2) * C_1 * r_1 * \{pbest_i^{(k)} - S_i^{(k)}\} + \\ &(1-r_2) * C_2 * (1-r_1) * \{gbest^{(k)} - S_i^{(k)}\} \end{aligned} \tag{10}$$

where r_1, r_2 and r_3 are the random parameters uniformly taken from the interval [0, 1]; and $sign(r_3)$ is a function defined as:

$$\begin{aligned} sign(r_3) &= -1 \quad \text{where } r_3 \le 0.05 \\ &= 1 \quad \text{where } r_3 > 0.05 \end{aligned} \tag{11}$$

The two random parameters $rand_1$ and $rand_2$ of (7) are independent. If both are large, both the personal and social experiences are over used and the particle is driven too far away from the local optimum. If both are small, both the personal and social experiences are not used fully and the convergence speed of the technique is reduced. So, instead of taking independent $rand_1$ and $rand_2$, one single random number r_1 is chosen so that when r_1 is large, $(1-r_1)$ is small and vice versa. Moreover, to control the balance between global and local searches, another random parameter r_2 is introduced. For birds' flocking for food, there could be some rare cases that after the position of the particle is changed according to (9), a bird may not, due to inertia, fly towards a region at which it thinks is most promising for food. Instead, it may be leading toward a region which is in opposite direction of what it should fly in order to reach the expected promising regions. So, in the step that follows, the direction of the bird's velocity should be reversed in order for it to fly back to the promising region. $sign(r_3)$ is introduced for this purpose. In birds' flocking or fish schooling, a bird or a fish often changes directions suddenly. This is described by using a ''craziness'' factor and is modelled in the technique by using a craziness variable. A craziness operator is introduced in the proposed technique to ensure that the particle would have a predefined craziness probability to maintain the diversity of the particles. Consequently, before updating its position the velocity of the particle is crazed by,

$$V_i^{(k+1)} = V_i^{(k+1)} + P(r_4) * sign(r_4) * v^{craziness} \tag{12}$$

where r_4 is a random parameter which is chosen uniformly within the interval [0, 1];

$v^{craziness}$ is a random parameter which is uniformly chosen from the interval $\left[v_i^{cr\_min}, v_i^{cr\_max}\right]$; and $P(r_4)$, $sign(r_4)$ are defined, respectively, as:

$$\begin{aligned} P(r_4) &= 1 \quad \text{when } r_4 \le P_{cr} \\ &= 0 \quad \text{when } r_4 > P_{cr} \end{aligned} \tag{13}$$

$$\begin{aligned} sign(r_4) &= -1 \quad \text{when } r_4 \ge 0.5 \\ &= 1 \quad \text{when } r_4 < 0.5 \end{aligned} \tag{14}$$

where P_{cr} is a predefined probability of craziness. The steps of CRPSO algorithm are as follows:

Step 1: Population is initialized for a swarm of n_p vectors, in which each vector represents a solution of filter coefficient values.

Step 2: Computation of initial cost values of the total population, n_P.

Step 3: Computation of population based minimum cost value i.e. the group best solution vector (gbest) and computation of the personal best solution vectors (pbest).

Step 4: Updating the velocities as per (10) and (12); updating the particle vectors as per (9) and checking against the limits of the filter coefficients; finally, computation of the updated cost values of the particle vectors and population based minimum cost value.

Step 5: Updating the pbest vectors, gbest vector; replace the updated particle vectors as initial particle vectors for step 4.

Step 6: Iteration continues from step 4 till the maximum iteration cycles or the convergence of minimum cost values are reached; finally, gbest is the vector of optimal IIR LP filter coefficients. The justifications of choosing the value of different CRPSO parameters are as follows:

Reversal of the direction of bird's velocity should rarely occur; to achieve this, $r_3 \le 0.05$ (a very low value) is chosen when $sign(r_3)$ will be -1 to reverse the direction. If P_{cr} is chosen less or, equal to 0.3, the random number r_4 will have more probability to become more than P_{cr}, in that case, craziness factor $P(r_4)$ will be zero in most cases, which is actually desirable, otherwise heavy unnecessary oscillations will occur in the convergence curve near the end of the maximum iteration cycles as referred to (9). $v^{craziness}$ is chosen very small (=0.0001) as shown in Table II. $r_4 \ge 0.5$ or, <0.5 is chosen to introduce equal probability of direction reversal of $v^{craziness}$ as referred to (12).

The design objective in this paper is to obtain the optimal combination of the IIR LP filter coefficients, so as to acquire the maximum stop band attenuation with the least transition width. Here lies the author's contribution that this design objective has been attained by the proposed CRPSO technique. The values of the parameters used for RGA, PSO and CRPSO techniques are given in Table II. The values of the parameters used for CLPSO technique are adopted from [36].

IV. Results and Discussions

A. Analysis of Magnitude Response of IIR LP Filters

Extensive simulation study has been performed for performance comparison of three algorithms namely, RGA, PSO and CRPSO for the 8th order IIR filter optimization problem. The design specification followed for all algorithms are given in Table I.

TABLE I
DESIGN SPECIFICATION OF IIR LP FILTER

Pass band ripple (δ_p)	Stop band ripple (δ_s)	Pass band normalized edge frequency (ω_p)	Stop band normalized edge frequency (ω_s)
0.001	0.0001	0.35	0.40

The values of the control parameters of RGA, PSO and CRPSO are given in Table II. Each algorithm is run for 30 times to get the best solution and the best results are reported in this paper. All optimization programs are run in MATLAB 7.5 version on core (TM) 2 duo processor, 3.00 GHz with 2 GB RAM.

Three aspects of the algorithms are investigated in this work namely, their accuracy, speed of convergence and stability. Fig. 1 shows the comparative gain plots in dB for the designed 8th order IIR LP filter obtained for different algorithms. Fig. 2 represents the comparative normalized gain plots for 8th order IIR LP filter. The best optimized numerator coefficients *(b_k)* and denominator coefficients *(a_k)* obtained after extensive simulation study are reported in Table III. It has been observed that maximum stop band attenuations 20.0 dB, 21.75 dB and 33.117 dB are obtained for RGA, PSO and CRPSO algorithms, respectively. Figs. 3-5 show the pole-zero plots for low pass 8th order IIR LP filter designed using RGA, PSO and CRPSO, respectively. A system is called stable and minimum phase only when its all poles and zeros, respectively are within the unit circle of z-plane. For designing the FIR filter, achieving these criterions is not a problem, but for IIR filters fulfilling these features, simultaneously is really a challenging job. Fig. 3 shows the pole-zero plot of 8th order IIR LP filter designed with RGA. In Fig. 4, pole-zero plot of 8th order IIR LP filter designed with PSO has been demonstrated. Fig. 5 shows the pole-zero plot of 8th order IIR LP filter designed with CRPSO. In this design approach, primarily, stability is assured with the location of poles within the unit circle shown in Figs. 3-5. So, stability condition assures that all 8th order IIR LP filters produce bounded output for bounded input (BIBO) without the fear of oscillation.

TABLE II
CONTROL PARAMETERS OF RGA, PSO AND CRPSO

Parameters	RGA	PSO	CRPSO
Population size	120	25	25
Iteration Cycle	100	600	500
Crossover rate	1	-	-
Crossover	Two Point Crossover	-	-
Mutation rate	0.01	-	-
Mutation	Gaussian Mutation	-	-
Selection	Roulette	-	-
Selection Probability	1/3	-	-
C_1	-	2.05	2.05
C_2	-	2.05	2.05
v_i^{min}	-	0.01	0.01
v_i^{max}	-	1.0	1.0
p_{cr}	-	-	0.3
$v^{craziness}$			0.00001

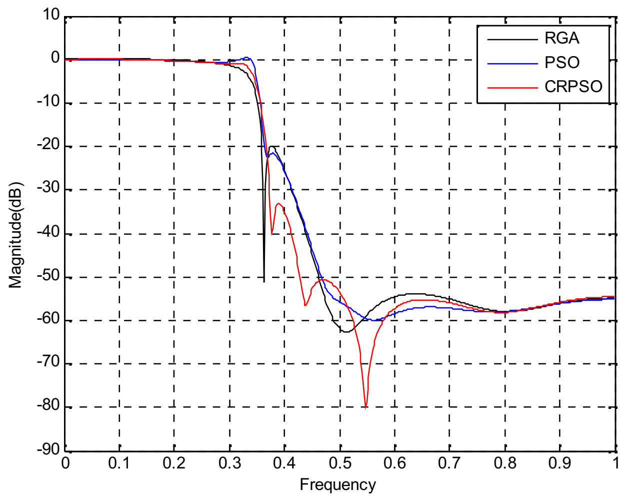

Fig. 1 Gain plots in dB for 8th order IIR LP filter using RGA, PSO and CRPSO.

International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:5, No:12, 2011

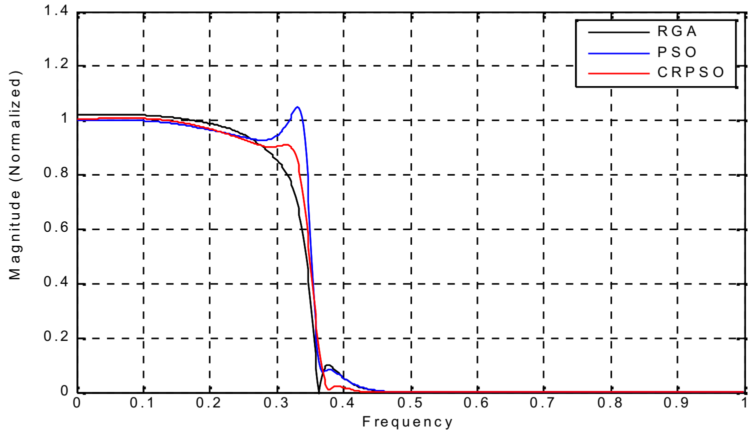

Fig. 2 Normalized gain plots for 8th order IIR LP filter using RGA, PSO and CRPSO

For IIR filter design, group delay is the function of normalized frequency due to which different frequency components undergo different amounts of phase shift. And the degree of severity increases as the distances of zeros are increased away from the unit circle. With this view point, it is observed from Fig. 3 that two zeros with same radii of 1.0932 are out side the unit circle along with two zeros of radii 1.0002 are almost on the unit circle and rest of the four zeros are within the unit circle. From Fig. 4 it is noticed that four zeros constitute two groups with radii 1.0829 and 1.0180, respectively and stay outside the unit circle apart from the rest of the four zeros which are within the unit circle. Fig. 5 explores that four zeros are within the unit circle like others with the uniqueness of rest of the zeros which are almost on the unit circle with radii 1.0081 and 1.0069.

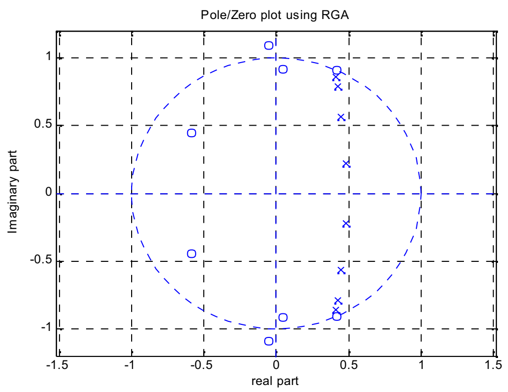

Fig. 3 Pole-Zero plot of 8th order IIR LP filter using RGA

It can be concluded that positions of four zeros out of eight are almost on the unit circle of z-plane for CRPSO design approach which gives the best group delay response among the algorithms. But it fails to acquire the designation of minimum phase system due to violating the condition of existence of all zeros within the unit circle.

TABLE III
OPTIMIZED COEFFICIENTS AND PERFORMANCE COMPARISON OF CONCERNED ALGORITHMS

Algorithms	Num_Coeff (b_k)	Den_Coeff (a_k)	Max. stop Band Attenuation (dB)
RGA	0.0167 0.0059 0.0434 0.0234 0.0451 0.0302 0.0277 0.0120 0.0092	0.9996 -3.5213 7.1631 -9.4231 8.7904 -5.7905 2.6429 -0.7583 0.1066	20.0
PSO	0.0165 0.0060 0.0423 0.0237 0.0454 0.0286 0.0275 0.0122 0.0073	0.9996 -3.5201 7.1638 -9.4233 8.7894 -5.7906 2.6430 -0.7593 0.1072	21.5683
CRPSO	0.0169 0.0054 0.0424 0.0228 0.0456 0.0285 0.0275 0.0115 0.0092	0.9990 -3.5209 7.1621 -9.4221 8.7896 -5.7908 2.6431 0.7587 0.1072	33.1170

Gain plots and Tables IV and V also explore that the proposed IIR filter design approach using CRPSO attains the highest stop band attenuation and smallest stop band and pass band ripples with a little increase in the transition width as compared to those produced by RGA and PSO algorithms. Gain plots also show better response at stop band region for CRPSO as compared to RGA and PSO. Table IV also shows that CRPSO yields lesser mean stop band attenuation, lesser variance and lesser standard deviation. Luitel *et al.* reported the design of 9th order IIR filter using PSO and PSO-QI in [33] and approximate attenuations of 22dB and 27dB, respectively, have been achieved in [33]. In this paper, maximum attenuation obtained for PSO is almost the same achieved with lower order filter.

International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:5, No:12, 2011

TABLE IV
STATISTICAL DATA FOR STOP BAND ATTENUATION (dB) FOR 8TH ORDER IIR LP FILTER

Algorithm	Maximum	Mean	Variance	Standard Deviation
RGA	-20.0000	-42.9281	263.0129	16.2177
PSO	-21.5683	-44.5499	264.6049	16.2667
CRPSO	-33.1170	-48.3590	80.5940	8.9774

TABLE V
QUALITATIVELY ANALYZED DATA FOR 8TH ORDER IIR LP FILTER

Algorithm	Pass band ripple (normalized)			Stop band ripple (normalized)			Transition Width (normalized)
	Maximum	Minimum	Average	Maximum	Minimum	Average	
RGA	1.0214	0.9198	0.9706	0.1000	7.3286×10^{-4}	5.0366×10^{-2}	0.0341
PSO	1.0500	0.9280	0.9890	0.0835	1.0000×10^{-3}	4.2250×10^{-2}	0.0216
CRPSO	1.0086	0.9029	0.9558	0.0221	1.0000×10^{-4}	1.1100×10^{-2}	0.037

B. Comparative effectiveness and convergence profiles of RGA, PSO and CRPSO

The effectiveness of an algorithm is measured in terms of the requirement of iteration cycles for achieving the optimum result with minimum error fitness value or the minimum value of cost function. In order to compare the algorithms' convergence speeds, Fig. 6 shows the variation of the cost (error) values with iteration cycles for the CRPSO, PSO and RGA based IIR filter designs. From Fig. 6 it can be concluded that the proposed algorithm CRPSO obtains the minimum cost (error value) with lesser number of iteration cycles as compared to PSO and RGA. It is also noticed that the proposed algorithm, CRPSO, has the faster rate of convergence in terms of sharp reduction in error function value shown in the abovementioned figure, compared to the rest of the error function curves obtained by RGA and PSO algorithms for obtaining the optimum results.

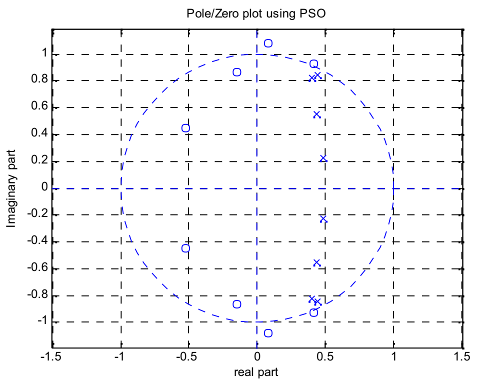

Fig. 4 Pole-Zero plot of 8th order IIR LP filter using PSO

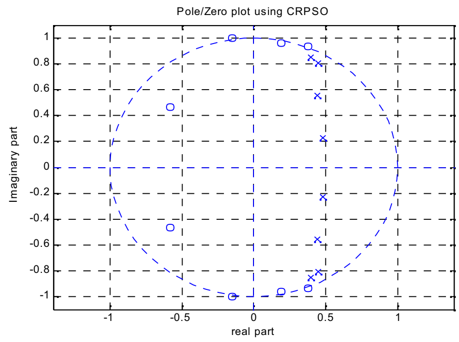

Fig. 5 Pole-Zero plot of 8th order IIR LP filter using CRPSO

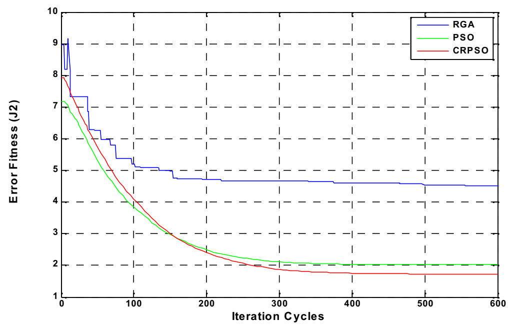

Fig. 6 Convergence profiles for CRPSO, PSO and RGA in case of 8th order low pass IIR filter

Table VI is gives the convergence profile data for the algorithms RGA, PSO and CRPSO applied for the design of 8th order IIR LP filter. From these data a platform is obtained on which it can be argued that the proposed algorithm CRPSO outperforms the rest algorithms in terms of minimum error, fastest convergence speed with least number of iteration cycles.

TABLE VI
CONVERGENCE PROFILE DATA FOR RGA, PSO AND CRPSO FOR 8TH ORDER LOW PASS IIR FILTER

Algorithms	Minimum Error Value	Iteration Cycles	Convergence Speed (per cycle)
RGA	4.5150	600	7.4400×10^{-3}
PSO	2.0300	600	8.5633×10^{-3}
CRPSO	1.7290	600	10.3183×10^{-3}

V. CONCLUSION

In this paper, a recently developed algorithm CRPSO, as a much improved version of PSO has been considered for realization of 8th order low pass IIR filter. Simulation studies show better performance of the proposed algorithm CRPSO over RGA and PSO in terms of magnitude response, convergence speed and stability which ensure the potential of proposed algorithm to handle similar filter design problem.

REFERENCES

[1] A. V. Oppenheim and R. W. Buck, *Discrete-Time Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1999.

[2] J. G Proakis and D. G. Manolakis, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[3] S. Das and A. Konar, "A swarm intelligence approach to the synthesis of two-dimensional IIR filters," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 8, pp. 1086-1096, April 2007.

[4] Z. M. Hussain, A. Z. Sadik and P. O' Shea, *Digital Signal Processing- An Introduction with MATLAB Applications*. New York: Springer-Verlag, 2011.

[5] R. K. Livesley, *Mathematical methods for Engineer*. Ellis Horwood Limited, West Sussex, 1989.

[6] L.B. Jackson and G. J. Lemay, "A simple remez exchange algorithm to design IIR filters with zeros on the unit circle," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, NM, USA, vol. 3, pp. 1333-1336, 1990.

[7] A. Antoniou, *Digital Signal Processing: Signals, Systems and Filters*. U.S.A.: McGraw Hill, 2006.

[8] W. S. Lu and A. Antoniou, "Design of digital filters and filter banks by optimization: a state of the art review," in *Proc. European Signal Processing Conf.*, vol. 1, pp. 351-354, Tampere, Finland, Sep. 2000.

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: Univ. Michigan Press. 1975.

[10] D. T. Pham and D. Karaboga, *Intelligent Optimization Techniques, Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. New York: Springer-Verlag, 2000.

[11] Z. Michalewics, *Genetic Algorithm + Data Structures = Evolution Programs*. 2nd ed. New York: Springer – Verlag, 1994.

[12] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.

[13] J. D. Farmer, N. H. Packard and A. S. Perelson, "The immune system, adaptation and machine learning," in *Proc. 5th Annu. Int. Conf. Physica D: Nonlinear Phenomena*, North – Holland, Amsterdam, 1986, vol. 22, Issues 1-3, pp. 187-204.

[14] M. Dorigo, V. Maniezzo and A. Colorni, "The ant system: optimization by a colony of cooperative agents," *IEEE Trans. on Sys., Man and Cybernetics* - Part B, vol. 26, no.1, pp. 29-41, 1996.

[15] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms," *IEEE Transactions on Systems, Man and Cybernetics*- Part B, vol. 34, no. 1, pp. 539-557, 2004.

[16] D. H. Kim, A. Abraham and J. H. Cho, "A hybrid genetic algorithm and bacterial foraging approach for global optimization," *Information Sciences*, vol. 177, pp. 3918-3937, 2007.

[17] T. Y. Sun, C-C. Liu, T-Y. Tsai, and S-T. Hsieh, "Adequate determination of a band of wavelet threshold for noise cancellation using particle swarm optimization," in *Proc. Evolutionary Computation, 2008,* Hong Kong, China, 1-6 June, pp. 1168-1175.

[18] W. Yao, S. Chen, S. Tan and L. Hanzo, "Particle swarm optimization aided minimum bit error rate multi-user transmission," in *Proc. IEEE Int. Conf. on Communications,* Germany, pp. 1-5, 2009.

[19] D. Mondal, S. P. Ghosal and A. K. Bhattacharya, "Radiation pattern optimization for concentric circular antenna array with central element feeding using craziness based particle swarm optimization," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 20, no. 5, pp. 577-586, John Wiley and sons, Inc., Sept. 2010.

[20] D. Mandal, S. P. Ghoshal and A. K. Bhattacharya, "Application of evolutionary optimization techniques for finding the optimal set of concentric circular antenna array," *Expert Systems with Applications, (Elsevier),* vol. 38, pp. 2942-2950, 2010.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization", in *Proc. IEEE Int. Conf. on Neural Network,* vol. 4, pp. 1942-1948, Australia 1995.

[22] R. Eberhart and Y. Shi, "Comparison between genetic algorithm and particle swarm optimization," in *Proc. 7th Annu. Conf. Evolutionary Computation,* San Diego. 2000.

[23] W. Fang, J. Sun and W. Xu, "A mutated quantum-behaved particle swarm optimizer for digital IIR filter design," *EURASIP Journal on Advances in Signal Processing,* Article ID-367465, pp. 1-7, 2009.

[24] S. H. Ling, H. H. C. Iu, F. H. F. Leung and K. Y. Chan, "Improved hybrid particle swarm optimized wavelet neural network for modeling the development of fluid dispensing for electronic packaging," *IEEE Trans. Ind. Electron.,* vol. 55, no. 9, pp. 3447-3460, Sep. 2008.

[25] B. Biswal, P. K. Dash and B. K. Panigrahi, "Power quality disturbance classification using fuzzy c-means algorithm and adaptive particle swarm optimization," *IEEE Trans. Ind. Electron.,* vol. 56, no. 1, pp. 212-220, Jan. 2009.

[26] N. E. Mastorakis, I. F. Gonos, and M. N. S. Swamy, "Design of two dimensional recursive filters using genetic algorithms," *IEEE Transaction Circuits and Systems 1- Fundamental Theory and Applications,* vol. 50, issue 5, pp. 634-693, May 2003.

[27] A. Ratnaweera, S. K. Halgamuge and H. C. Watson, "Self organizing hierarchical particle swarm optimizer with time varying acceleration coefficients," *IEEE Trans. Evolutionary Computational*, vol. 8, no. 3, pp.240-255, 2004.

[28] S. M. Guru, S. K. Halgamuge and S. Fernando, "Particle swarm optimizers for cluster formation in wireless sensor networks," in *Proc. Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing,* Melbourne, pp. 319-324, 2005.

[29] J. Sun, W-B Xu and J. Liu, "Training RBF neural network via quantum-behaved particle swarm optimization," in *Proc. ICONIP 2006*, Hong Kong, China, 3-6 Oct. pp. 1156-1163, 2006.

[30] H-M. Feng, "Self-generation RBFNs using evolutional PSO learning," Neuro *Computing,* vol. 70, nos. 1-3, pp. 41-251, 2006.

[31] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimization and intelligence: Advances and Applications," *Information Science Reference,* Hershey, New York, 2010.

[32] D. Mandal, S.P. Ghoshal, and A. K. Bhattacharjee, "Radiation Pattern Optimization for Concentric Circular Antenna Array With Central Element Feeding Using Craziness Based Particle Swarm Optimization," *International Journal of RF and Microwave Computer-Aided Engineering,* John Wiley & Sons, Inc., vol. 20, Issue. 5, pp. 577-586, Sept. 2010.

[33] B. Luitel and G. K. Venayagamoorthy, "Particle swarm optimization with quantum infusion for the design of digital filters," *IEEE Swarm Intelligence Symposium,* St. Louis MO USA, pp. 1-8, Sep. 2008.